Reconstruction and Applications of Collective Storylines from Web Photo Collections

Gunhee Kim

September 2013
CMU-CS-13-125

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Eric P. Xing, Chair
Takeo Kanade
Christos Faloutsos
Antonio Torralba, MIT

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

This work is supported by NSF IIS-1115313, NSF IIS-0713379, NSF DBI-0640543, AFOSR FA9550010247, ONR N000140910758, and Google to Eric P. Xing. Some parts of this research are also supported by MURI N00014-07-1-0747 to Takeo Kanade.

Keywords: Computer Vision, Machine Learning, Optimization

Abstract

Widespread access to photo-taking devices and high speed Internet has combined with rampant social networking to produce an explosion in picture sharing on Web platforms. In this environment, new challenges in image acquisition, processing, and sharing have emerged, creating exciting opportunities for research in computer vision and multimedia data mining. In this dissertation, we explore one of these interesting problems, *the reconstruction of collective storylines* as an efficient but comprehensive structural summary of ever-growing big image data shared online.

More specifically, the goal of this dissertation can be summarized as follows. *Given large-scale online image collections and associated meta-data, we aim to create the collective storylines by jointly inferring the temporal trends and the overlapping contents of image collections. We also explore novel computer vision and data mining applications taking advantage of the reconstructed photo storylines*.

In order to achieve the proposed research objective, we develop the required technologies from three research directions, which are (1) understanding of temporal trends of image collections, (2) discovery of overlapping contents across image collections, and (3) reconstruction and applications of collective photo storylines. The first direction of the work addresses the problems of understanding what topics are popular when by whom in the image collections, while the second line of the work studies the approaches for detecting salient and recurring contents across the image collections in the form of bounding boxes or pixel-wise segmentations. Finally, based upon the results of the work in the first two directions, we propose the reconstruction algorithms of branching storyline graphs, and explore their promising applications at the intersection of computer vision and multimedia data mining.

Acknowledgments

First of all, I must thank my advisor Eric P. Xing for his great advise and guidance through my Ph.D. study. He always challenged me to pursue big practical problems with in-depth technical specialties, which ended up achieving much better work than I initially could.

I also want to thank my thesis committee members. Takeo Kanade was my initial advisor who taught me the fundamentals to be a professional researcher. Christos Faloutsos provided me with key insights to bridge between computer vision and data mining since when I was a Master student. Antonio Torralba allowed me to work in his group as a visiting researcher, where I could publish two initial papers that became the base of this dissertation.

I owe much to collaborators and co-authors; I would particularly thank Li Fei-Fei, who invited me to her Stanford vision lab during spring semester 2011. It was a three-month visit, which was long enough for me to develop my thesis ideas further under the collaboration with her and her students.

I thank all the CMU SAILING lab members for the wonderful research comments they provided: Seunghak Lee, Bin Zhao, Ankur Parikh, Qirong Ho, Kyung-Ah Sohn, Amr Ahmed, Jun Zhu, Le Song, Kriti Puniyani, Mladen Kola, Junming Yin, Chong Wang, Pengtao Xie, Bin Shu, and Kumar Avinava Dubey.

I am grateful to many of CMU faculties, staff members, and friends, who made my time at CMU smoother and more enjoyable. I am very sorry that they are too many to name here. In addition, I have to admit that I learned a lot about basic knowledge and research skills from great CMU courses, talks, and seminars such as VASC seminars, ML Lunch seminars, and the computer vision misc reading group.

Finally, I would like to thank my parents, my brother and his family, grandma, and aunt for their many years of love, support, and encouragement.

Contents

1 Introduction **1**
1.1 Background and Motivation 1
1.2 Thesis Statement 5

2 Survey of Related Work **8**

I Understanding Temporal Trends of Image Collections **13**

3 Analyzing Dynamic Behaviors of Web Photo Sets **16**
3.1 Introduction 16
3.2 Network Construction by Sequential Monte Carlo 17
3.2.1 Image Description and Similarity Measure 17
3.2.2 Problem Statement 18
3.2.3 Network Construction using Sequential Monte Carlo 18
3.3 Experiments 22
3.3.1 Evaluation setting 22
3.3.2 Results on Evolution of Subtopics 22
3.3.3 Comparison with Text Analysis 24
3.3.4 Temporal Association for classification 25
3.4 Summary 27

4 Time-Sensitive Image Retrieval and Prediction **28**
4.1 Introduction 28
4.2 Problem Formulation 31
4.2.1 Image Description 31
4.2.2 User Description 32
4.3 Multivariate Point Processes 33
4.4 Temporal Modeling of Photo Streams 34
4.4.1 Models of Temporal Behaviors 34
4.4.2 Model Selection 37
4.4.3 Regularized Multi-Task Regression 38
4.4.4 Optimization for Parameter Learning 39
4.5 Time-Sensitive Image Retrieval 40

4.5.1 Predictive Ranking . . . 40
4.5.2 Personalization . . . 41
4.5.3 Computation time . . . 41
4.6 Experiments . . . 42
4.6.1 Evaluation Setting . . . 42
4.6.2 Quantitative Results . . . 43
4.6.3 Qualitative Results . . . 44
4.7 Summary . . . 45

II Discovering Overlapping Contents of Image Collections 48

5 Unsupervised Detection of Regions of Interests (ROI) 51
5.1 Introduction . . . 51
5.2 ROI Candidates and Description . . . 52
5.3 Iterative Detection of Regions of Interest . . . 53
5.3.1 Similarity Networks and Link Analysis Techniques . . . 53
5.3.2 Overview of Algorithm . . . 54
5.3.3 Hub Seeking with Centrality and Diversity . . . 55
5.3.4 ROI Refinement . . . 56
5.3.5 Scalability Setting . . . 56
5.4 Experiments . . . 57
5.4.1 Performance Tests . . . 57
5.4.2 Scalability Tests . . . 59
5.5 Summary . . . 60

6 Diversity Ranking, Image Segmentation, and Cosegmentation 62
6.1 Introduction . . . 62
6.1.1 Background . . . 63
6.2 Submodularity and Diffusion . . . 64
6.2.1 Optimization on Anisotropic Diffusion . . . 64
6.2.2 Diversity ranking and clustering . . . 66
6.3 Image CoSegmentation . . . 68
6.3.1 Segmentation of a Single Image . . . 68
6.3.2 Cosegmentation . . . 69
6.4 Experiments . . . 72
6.4.1 Results on Figure-ground Cosegmentation . . . 72
6.4.2 Results on Scalable Cosegmentation . . . 74
6.5 Summary . . . 75

7 Multiple Foreground Cosegmentation 76
7.1 Introduction . . . 76
7.2 Problem Formulation . . . 78

7.2.1 Foreground Models . . . 79
7.2.2 Region Assignment . . . 79
7.3 Tractable Multiple Foreground Cosegmentation . . . 81
7.3.1 Tree-Constrained Region Assignment . . . 81
7.3.2 Generating Candidate Sets . . . 82
7.3.3 Tractable Region Assignment . . . 82
7.3.4 The MFC Algorithm . . . 85
7.4 Experiments . . . 86
7.4.1 Results over FlickrMFC Dataset . . . 86
7.4.2 Results over ImageNet Dataset . . . 89
7.5 Summary . . . 91

III Reconstruction and Applications of Photo Storylines 92

8 Jointly Aligning and Segmenting Multiple Photo Streams 95
8.1 Introduction . . . 95
8.2 Problem Formulation . . . 97
8.2.1 Input and Output . . . 97
8.2.2 Overview of Algorithm . . . 97
8.3 Alignment of Photo Streams . . . 98
8.3.1 Image Description . . . 98
8.3.2 Image Similarity Measure . . . 98
8.3.3 Pairwise Photo Stream Alignment . . . 99
8.3.4 Multiple Photo Stream Alignment . . . 99
8.4 Large-Scale Cosegmentation . . . 100
8.4.1 Building Image Graphs . . . 100
8.4.2 Scalable Cosegmentation . . . 101
8.4.3 Analysis of Algorithm . . . 104
8.5 Experiments . . . 104
8.5.1 Results on Alignment . . . 105
8.5.2 Results on Segmentation . . . 106
8.5.3 Preliminary Results on Photo Storylines . . . 108
8.6 Summary . . . 109

9 Reconstructing Photo Storyline Graphs 111
9.1 Introduction . . . 111
9.2 Problem Formulation . . . 113
9.3 Estimating Photo Storyline Graphs . . . 114
9.3.1 Optimization . . . 115
9.3.2 Incorporating Side Information . . . 117
9.4 Experiments . . . 117
9.4.1 Evaluation Setting . . . 117

9.4.2 Results on Storyline graphs . . . 120
9.5 Summary . . . 123

10 Visualizing Brand Associations from Web Photos **125**
10.1 Introduction . . . 125
10.2 Problem Formulation . . . 128
10.2.1 Image Data Crawling . . . 128
10.2.2 Overview of Algorithm . . . 128
10.3 Exemplar Detection/Clustering and Brand Localization . . . 129
10.3.1 Image Description . . . 129
10.3.2 Image Similarity Measure . . . 129
10.3.3 Constructing K-Nearest Neighbor Graphs . . . 130
10.3.4 Exemplar detection and clustering . . . 131
10.3.5 Brand Localization via Cosegmentation . . . 132
10.4 Embedding Brand Association Maps . . . 133
10.5 Experiments . . . 136
10.5.1 Results on Brand Association Maps . . . 136
10.5.2 Results on Clustering . . . 137
10.5.3 Results on Brand Localization . . . 138
10.5.4 Correlations between Image data and Sales Data . . . 140
10.6 Summary . . . 143

IV Conclusion **144**

11 Discussion **146**
11.1 Key Observations and Contributions . . . 146
11.2 Future Directions . . . 148
11.3 Conclusion . . . 150

Bibliography **151**

Chapter 1

Introduction

The prevalence of digital cameras and smartphones with ubiquitous high-speed Internet connection produces an explosion of pictures being uploaded, shared and communicated online, across websites, platforms and social networks. This new phenomenon poses challenges and opportunities in the research of computer vision and multimedia data mining. This dissertation aims to explore the solutions to several interesting problems that emerge from large-scale online image collections contributed by general public. In this introduction, we begin with describing some notable challenges and opportunities that constitute the motivation of our research, and derive the thesis statement that we would like to achieve through our research.

1.1 Background and Motivation

Recent technical progresses in photo-taking devices, Internet connection, and social networking have changed the ways of image acquisition, processing, and sharing. We here summarize several important characteristics of them as background and motivation of our research.

A picture is a memory frozen in time.

The starting point of this thesis is to view fast growing Web image collections as the socially aggregated pictorial records of general users' experiences. That is, photos are the records of personal experiences for specific time and places with their own stories. This is a reasonable assumption because people usually take pictures on their memorable moments, and the Web is a popular medium to share the photos with their friends and families.

Therefore, it becomes important to understand the temporal, contextual, and episodic meanings of the pictures beyond their semantic meanings, which are traditional subjects of computer vision research. This challenge can be better understood with an intuition from a widely accepted nomenclature of human and social memory studies [Halbwachs, 1992; Tulving, 1972]. We illustrate three important types of memories in the image domain, *semantic*, *episodic*, and *collective* memory, with an example of the topic keyword *gun* in Fig.1.1. First, the *semantic memory* accounts for the general meaning and concept-based knowledge of the word. That is, in semantic memory, a *gun* is a portable weapon. Fig.1.1.(a) shows some example pictures of the gun in semantic memory, which are sampled from ImageNet [Deng et al., 2009]. Second, the *episodic memory* is an autobiographical record of a personal experience associated with time, place and other contextual knowledge.

Figure 1.1: The word *gun* in different memory types. (a) In *semantic memory*, a gun is a portable weapon. Images are obtained from the synset *gun* (n03467984) of ImageNet [Deng et al., 2009]. (b) A gaming scene can be a piece of *episodic memory* associated with a gun for a particular user. (c) The recent Sandy Hook Elementary School shooting accident in Connecticut may build a collective memory shared by American people in December 2012. The photo streams in (b) and (c) are downloaded from Flickr with the query word *gun*. We also present the information about owners, locations, and timestamps.

For example, as shown in Fig.1.1.(b), for a particular person, a gun may be mainly perceived as a controller for video games, and thus a gaming scene can be a trace of his *episodic memory* associated with the gun. Finally, if we zoom out the memory into a social scale, another interesting memory type is the *collective memory* that is constructed, shared, passed by a coherent group of people [Halbwachs, 1992]. For example, the recent school shooting accident in Connecticut in December 2012 comprises a piece of *collective memory* for people in the United States, as shown in Fig.1.1.(c).

From such memory perspective, most previous work in computer vision has mainly focused on the semantic memory tasks. For example, the scene and object parsing has been a core problem in computer vision [Chum and Zisserman, 2007; Felzenszwalb et al., 2010; Kim et al., 2008a,b; Lazebnik et al., 2006; Liu et al., 2009a]; its goal is to recognize and segment general objects or scenes in images. In this pipeline, one may first identify a fixed-number of object categories to be detected, train object models for each category using training data, and finally apply the learned detectors to assign object labels to the regions of the input images. As a result of this parsing, each image is simply interpreted as a spatial layout of semantic objects and scenes. However, very little research has been done for episodic and collective memories in computer vision, falling short of delivering personalized or socially aggregated understanding of images. As highly connected information society has come, we believe episodic and collective interpretation of the images becomes more interesting and anticipating, which is one important motivation of this thesis.

Pictures are not alone.

One interesting characteristic of today's photo taking is that it is hard to imagine completely isolated pictures. Taking a picture is so easy and cheap, which leads people to usually take a series

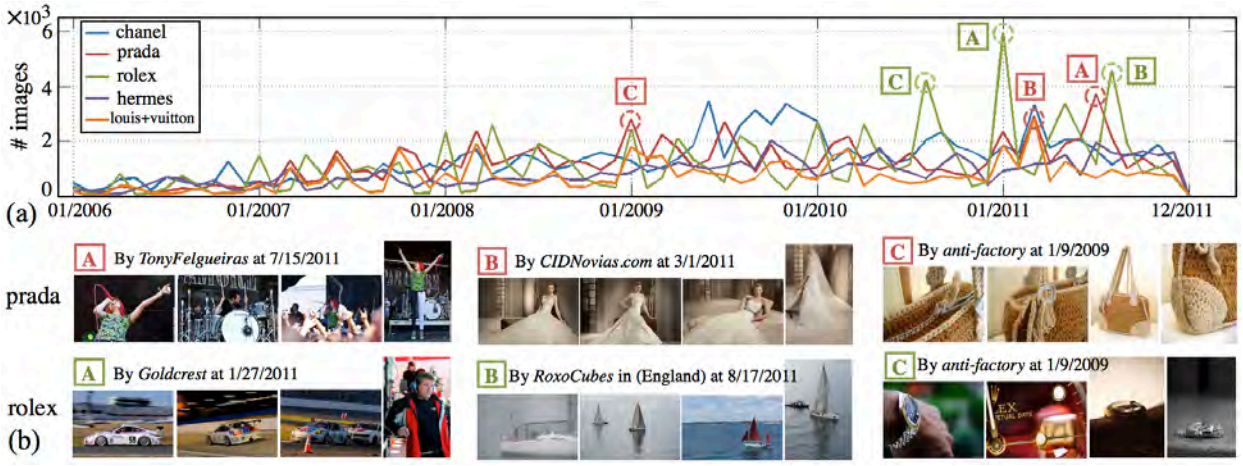


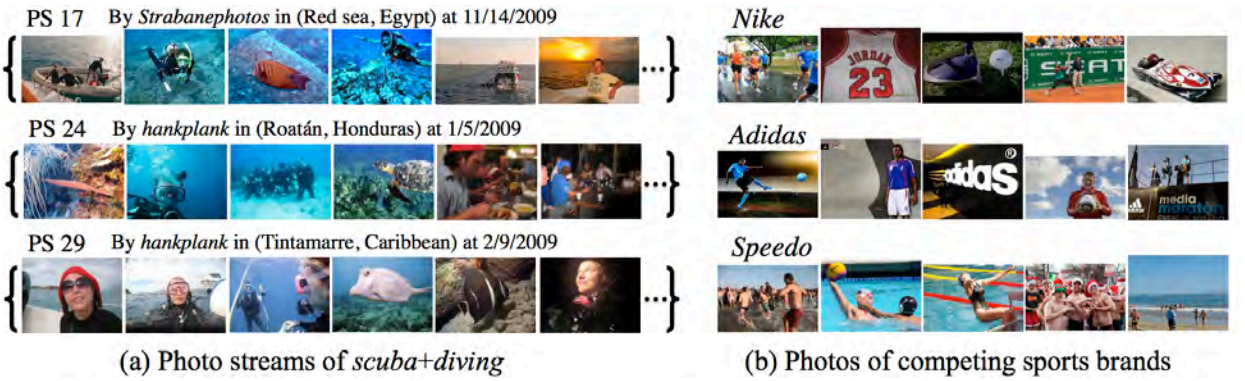
Figure 1.2: Images on the Web change over time. (a) We present the variations of Flickr image volumes per month from 2006 to 2011 for five luxury brands: {*Chanel*, *Prada*, *Rolex*, *Hermes*, *Louis+Vuitton*}. We mark top three peaks of the *Prada* and the *Rolex*. (b) We sample one photo stream per peak for the *Prada* and the *Rolex*. The main themes of photo streams are severely variable: *concert*, *wedding dress*, and *bag* for the *Prada*, and *car racing*, *yacht*, and *watch* for the *Rolex*.

of pictures for their memorable moments. Moreover, nowadays photo-taking devices are not only intended for image recording but also for editing and communication. Thus, photographers can easily check, edit, and share their series of pictures with optionally adding some tags or comments. This new technology convenience results in that many online pictures are grouped as a photo set and are associated with additional contexts or meta-data surrounding the pictures. We call such a set of pictures as a *photo stream*, which is roughly defined as a set of photos taken in sequence by a single photographer for a single event within a short range of time (*e.g.* a single day). As an example, the photo stream in Fig.1.1.(b) shares overlapping contents including the same persons, objects, and background, since it is taken in series. In addition, it accompanies several meta-data, such as the owner ID, timestamps, or GPS information. We can freely download millions of photo streams for any topics from the Web.

Consequently, in this thesis, we seek the techniques to take advantage of such additional information to solve challenging computer vision problems. From the fact that images are taken in sequence, we use the signal of recurring objects or scenes to identify salient contents across the image set. From the fact that metadata are available, we leverage personal, spatial, and temporal information to understand the context of images.

Pictures change over time.

The topical patterns of Web image corpora evolve over time. As an example, Fig.1.2.(a) shows the variations of Flickr image volumes for five luxury brands per month from 2006 to 2011. The popularities of five brands, which can be roughly estimated by the number of shared pictures on Flickr, show a lot of rises and falls on the timeline. This visualization is similar to Google trends, which summarize the temporal variation of search volumes of keywords. In addition to the image



(a) Photo streams of *scuba+diving* (b) Photos of competing sports brands

Figure 1.3: Motivation for the storyline reconstruction from community photos. (a) We show three sampled photo streams of the *scuba+diving*. Although they are taken by various users at different time and places, they share common storylines (*e.g.* wearing equipment, riding a boat, underwater diving, dinner, and so on). (b) We show sampled images from three competing sports brands: *Nike*, *Adidas*, and *Speedo*. Various personal experiences are associated with each brand differently, and such storylines can be used for a wide range of e-commerce applications such as brand evaluation and online multimedia advertisement.

volumes, the image contents also evolve over time according to the changes of information flows or people's interests at different time points, as shown in Fig.1.2.(b). We sample one photo stream for the *Prada* and the *Rolex* topic at the top three peak months. Even though they are retrieved with the same keyword, the main themes of photo streams are severely diverse, for example, *a concert*, *wedding dress*, and *a bag* in the *Prada*, and *car racing*, *yacht*, and *a watch* in the *Rolex*. Such extreme diversity of the Web images is one of well-known challenges in the Internet vision research. We believe such diversity is largely contributed by human's rich notion of similarities and associations, some of which can be disambiguated by understanding of temporal trends of Web image collections. Therefore, in this thesis, we explore the discovery of such topical evolution in Web images to address challenging computer vision problems including image classification and retrieval.

Every picture is a part of story.

We believe that one important challenge in recent Web-oriented computer vision research is to infer collective storylines from millions of photo streams, and to discover the relations between the reconstructed storylines and the photo streams of individual users. The reconstructed photo storylines can be used as an efficient but comprehensive structural summary of large-scale and ever-growing online pictorial data, which have led to an *information overload* problem; users are often overwhelmed by the flood of pictures, and struggling to grasp various activities, events, and stories of the pictures taken by even their close friends.

In this dissertation, we explore the two applications of photo storylines, which can be better understood with examples of Fig.1.3, even though they are just a tip of iceberg for the impact of this research.

First, many topics of interest usually consist of a sequence of activities or events recurred across

the photo streams. Some typical examples include outdoor recreational activities, holidays, and sports events. Fig.1.3.(a) shows three photo streams of the *scuba+diving* as an outdoor recreational activity. Although they are independently captured by various users at different time and places, they are likely to share common storylines, such as wearing equipment, riding a boat, underwater diving, dinner, and so on. The construction of such photo storylines can potentiate a variety of applications. For example, if a family decides to go to a scuba diving trip, they can make a plan by previewing what other people usually do. After the trip, they can also review the similarities and differences of their trip compared to others, and fill missing parts of their photo sets by others' pictures.

Second, another interesting example includes the comparison between the photo storylines of competing brands. Fig.1.3.(b) shows sampled images from three competing sports brands: *Nike*, *Adidas*, and *Speedo*. With widespread availability of digital cameras and smartphones, people can freely take pictures on any memorable moments, which include experiencing or purchasing products they like. In addition, many online tools enable people to easily share, comment, or bookmark the images of products that they wish to buy. Hence, from large-scale online pictures of the brands over social networking sites, we can infer the meaningful threads of stories associated with the brands. The reconstructed storylines can reveal how people perceive the brands, what products people particularly like, and what typical interactions take place between users and products in natural social contexts. Consequently, the research on the storylines can lead a wide variety of potential benefits, ranging from content-based image retrieval to online multimedia advertisement.

Therefore, in this thesis, we develop algorithms to automatically summarize and visualize a large set of pictures in the form of storylines, which can characterize various branching narrative structure associated with the topic in an efficient but comprehensive way. Even though the definition of storylines differs according to literature, we refer the storyline to a series of events that have *chronological* or *causal* relations. Its more rigorous definition will be developed throughout this dissertation.

1.2 Thesis Statement

The thesis statement can be summarized as follows:

> *Given large-scale online image collections and associated meta-data, we aim to create the collective storylines by jointly inferring the temporal trends and the overlapping contents of image collections. We also explore novel computer vision and data mining applications taking advantage of the reconstructed photo storylines.*

Consequently, this dissertation consists of three parts: (i) understanding of temporal trends of image collections (Part I), (ii) discovery of overlapping contents across image collections (Part II), and (iii) reconstruction and applications of collective photo storylines (Part III). Metaphorically, the projects in Part I and Part II attempt to simultaneously *see the forest for the trees* and *see the trees for the forest*. The trees and forest analogically correspond to individual images and the collection of images, respectively. The underlying idea is that it is mutually rewarding to understand the overall trends of the collection, and the contents of individual photos. Discovering the temporal

and contextual changes of the photo collection can tell what topics are popular when by whom. Subsequently, it helps interpret the contents of images more accurately. In the reverse direction, understanding the contents of images can reveal statistically dominant visual information, which can help reconstruct the overall trends more effectively. Finally, Part III proposes the reconstruction algorithms of branching storyline graphs, and explores several applications of collective photo storylines based upon the results of the work in Part I and Part II. In this dissertation, we mainly focus on the storylines about outdoor recreational activities, holidays, sports events, and competing brands, but their usefulness is not limited but promising in a wide range of other applications.

Table 1.1 summarizes the outline of this thesis. In order to achieve the proposed thesis statement, we have completed the following projects.

Understanding Temporal Trends of Web Image Collections (Part I). The main objective of the algorithms in this part is to model the temporal trends of large-scale image collections, and help solve existing or novel computer vision problems as follows.

- (Chapter 3) We study the temporal evolution of topics in Flickr image collections. This research enables us to detect subtopic outbreak detection, and improve image classification performance by using temporal context.
- (Chapter 4) We develop an approach for leveraging time and optionally user information to improve image search quality. We then extend the proposed time-sensitive image retrieval method into solving a Web image prediction problem, in which given a query word and a future time point, we predict the images that are likely to appear on the Web.

Discovering Overlapping Contents of Image Collections (Part II). The goal of the algorithms in this part is to discover salient contents of individual images in the form of bounding

Part I – Understanding Temporal Trends	**Part II – Discovering overlapping image contents**
• Analyzing Dynamic Behaviors of Web Photo Sets [Kim et al., 2010] (Chapter 3) • Time-Sensitive Image Retrieval and Prediction [Kim et al., 2012; Kim and Xing, 2013b] (Chapter 4)	• Unsupervised Detection of Regions of Interests (ROI) [Kim and Torralba, 2009] (Chapter 5) • Diversity Ranking, Image Segmentation, and Cosegmentation [Kim et al., 2011] (Chapter 6) • Multiple Foreground Cosegmentation [Kim and Xing, 2012] (Chapter 7)
Part III – **Reconstruction and Applications of Photo Storylines**	
• Jointly Aligning and Segmenting Multiple Web Photo Streams [Kim and Xing, 2013a] (Chapter 8) • Reconstructing Photo Storyline Graphs [Kim and Xing, 2014a] (Chapter 9) • Visualizing Brand Associations from Web Photos [Kim and Xing, 2014b] (Chapter 10)	

Table 1.1: Thesis outline.

boxes or pixel-wise segmentation, by detecting the recurring regions that are shared across the image collections.

- (Chapter 5) We present a fast and scalable method to detect rectangular regions of interest (ROI), by searching for statistical dominance of object signals from cluttered large-scale Web images without any labels.
- (Chapter 6) We develop a diffusion-based optimization framework that is applicable to a wide range of computer vision problems. We show that the proposed optimization lead to an efficient and effective solutions to diversity ranking, single-image segmentation, and cosegmentation.
- (Chapter 7) We propose an approach to multiple foreground cosegmentation as a less restrictive and more practical cosegmentation algorithm so far, aiming to be directly applicable to the Web photo streams of general users.

Reconstruction and Applications of Photo Storylines (Part III). The objective here is to integrate the algorithms in previous two parts, and address the discovery of collective photo storylines and their uses for interesting Web applications as follows.

- (Chapter 8) As a first technical step to detect collective storylines, we propose an approach to jointly aligning and segmenting large-scale Web photo streams of different users.
- (Chapter 9) We address an approach for reconstructing branching storyline graphs as a structural summary of large-scale photo streams. Our optimization algorithm can estimate sparse time-varying directed graphs directly from photo streams with optionally other side information such as friendship graphs.
- (Chapter 10) We develop a novel methodology to visualize brand associations as the storylines of competing brands from image collections contributed by general public. Our algorithm can also automatically identify the regions that are associated most with brands in the images.

Note that most chapters of this thesis have been published in [Kim et al., 2012; Kim and Torralba, 2009; Kim and Xing, 2012, 2013b, 2014b, 2013a; Kim et al., 2011, 2010; Kim and Xing, 2014a]. In order to facilitate further research in this area, we provide Matlab codes, demos, and image data at our webpage (`http://www.cs.cmu.edu/~gunhee`).

Chapter 2

Survey of Related Work

In this chapter, we review closely related previous research that grounds the work of this dissertation, and clarify important uniqueness of our work compared to them. We categorize the survey of literature into three main directions as presented in the thesis statement.

Understanding Temporal Trends of Web Image Collections

Temporal context in computer vision: We here review important previous work using temporal cues for image analysis. The importance of temporal context has long been recognized in neuroscience research [Becker, 1999; Sinha et al., 2006; Wallis and Bulthöff, 2001]. Much recent research has supported that the *temporal association* (*i.e.* liking temporally close images) is an important mechanism to recognize objects and generalize visual representation.

In computer vision, Paletta *et al.* [Paletta et al., 2000] use a POMDP framework for the modeling of temporal context to disambiguate the object hypotheses. In [Boutell et al., 2005], an HMM-based temporal context model is proposed to solve scene classification problems. The timing information is also used to organize personal photo albums.

As the Internet vision emerges as a promising research area in computer vision, time information starts to be used to assist visual tasks for Web applications. Surprisingly, however, the dynamics or temporal context of Web images has not yet been studied a great deal, contrary to that such study for Web text data has been one of active research areas in data mining and machine learning communities [Blei and Lafferty, 2006; Wang and McCallum, 2006]. We briefly review some notable examples using the timestamps associated with the images for visual tasks. Cao *et al.* [Cao et al., 2008] develop an annotation method for personal photo collections, in which the timestamps are used for better correlation discovery between the images. Li *et al.* [Li et al., 2009] propose a landmark classification that leverages temporal information as a constraint to reduce misclassification. Quack *et al.* [Quack et al., 2008] also use the timestamps as an additional feature for the object and event retrieval of online images. Kalogerakis *et al.* [Kalogerakis et al., 2009] present a method to geolocate a sequence of images taken by a single individual. Temporal constraints between the images in sequence are used as a strong prior to improve the geolocation accuracy.

Image retrieval and reranking: Recently, image reranking has been actively studied to improve text-based image search by leveraging visual or user feedback information [Cui et al., 2008; Jing and Baluja, 2008; Liu et al., 2011; Morioka and Wang, 2011; Wang et al., 2011; Yang and Han-

jalic, 2010]. Most image reranking methods have exploited three sources of information, which are human-labeled training data [Yang and Hanjalic, 2010], user relevance feedback [Cui et al., 2008; Wang et al., 2011], and pseudo-relevance feedback [Morioka and Wang, 2011]. Given an image database retrieved by text-based search, the user relevance feedback approach asks a user to select a query image to clarify her search intent. The pseudo-relevance feedback approach assumes the top images retrieved by text-based search as pseudo-positive examples and bottom ranked images as pseudo-negative examples. Once the training data are obtained, almost all existing methods learn ranking models relying on the semantic meaning of a query word and the feature-wise image similarity. The uniqueness of our work beyond previous work is that we additionally emphasize the temporal trends and user history associated with the images.

Prediction of user behaviors on the Web: Based on the fact that contents and user behaviors on the Web change over time, building predictive models for them has been a promising direction in Web study [Amodeo et al., 2011; Jin et al., 2010; Radinsky et al., 2008, 2012]. Some notable examples of prediction include future likely news [Radinsky et al., 2008], peaks of topics in the New York Times corpus [Amodeo et al., 2011], periodicity and surprise detection of queries, URLs, and query-URL pairs [Radinsky et al., 2012], and the distribution of consumer products [Jin et al., 2010]. In this thesis, we perform the prediction of images that are likely to appear at future time points, which is a novel application of user behavior prediction in the image domain.

Discovering Overlapping Contents of Image Collections

Unsupervised localization: The unsupervised localization addresses the problem of localizing objects in images without any supervision [Ahuja and Todorovic, 2007; Fergus et al., 2005; Kim et al., 2008a; Russell et al., 2006; Sivic et al., 2005; Winn and Jojic, 2005]. They automatically identify repetitive visual contents across the input dataset, and learn their appearance models. One main limitation of most previous work is that they do not scale up to large-scale Web datasets. We focus on overcoming this limitation in the second part of this dissertation.

Detection of regions of interest (ROI): The ROI detection identifies the regions of objects that may interest users in cluttered images [Bosch et al., 2007; Chum et al., 2007; Quattoni and Torralba, 2009; Liu et al., 2007]. The ROI detection can be used as an important building block in a variety of computer vision problems, including object modeling for recognition [Bosch et al., 2007; Chum et al., 2007; Russakovsky and Ng, 2010], indoor scene description [Quattoni and Torralba, 2009], segmentation prior [Lempitsky et al., 2009], and image thumbnailing [Marchesotti et al., 2009]. In chapter 5 of this thesis, we propose a fast and scalable ROI detection method by searching for recurring object signals from large-scale Web images.

Online image collections: The goal of online image collection methods is to collect relevant images from highly noisy data retrieved by text keywords from the Web [Collins et al., 2008; Deng et al., 2009; Li et al., 2007; Schroff et al., 2007]. Previous work can be classified according to what additional metadata are used to decide which images are acceptable or not. Examples include user-labeled seed images [Li et al., 2007; Collins et al., 2008], texts and HTML tags [Schroff et al., 2007], and human assistance by Amazon Mechanical Turk [Deng et al., 2009]. In this thesis, we explore the methods that identify relevant regions of objects in images without any supervision.

Image segmentation and cosegmentation: Image segmentation has long been considered as

a complicated visual task that demands a tight integration of high-level and low-level processes in vision [Ullman, 2000]. It involves not only the grouping of pixels based on low-level structure of color and texture, but also the high-level knowledge about the objects of interest. Recently, cosegmentation has been actively studied in image segmentation research [Batra et al., 2011; Hochbaum and Singh, 2009; Joulin et al., 2010, 2012; Kim and Xing, 2012; Kim et al., 2011; Mukherjee et al., 2011; Rother et al., 2006; Vicente et al., 2010, 2011]; in this setting, the high-level signal is implicitly provided as recurring objects (or foregrounds) in multiple images, out of which the repeating objects are jointly segmented. Cosegmentation has a wide potential in web-scale applications. For example, it can guide an interactive image editing by suggesting popular regions in the image database [Batra et al., 2010; Rother et al., 2006], or summarize personal photo collections by automatically segmenting highly co-occurring object instances such as persons or dogs [Joulin et al., 2010]. In this thesis, we propose novel scalable cosegmentation algorithms that can be applicable to general users' photo streams in chapter 6 and chapter 7, in which we will clarify key uniqueness of our methods

Combinatorial optimization for object detection: Recently, combinatorial optimization techniques have been popularly used in object detection research. Some notable examples include branch-and-bound schemes for efficient subwindow search [Lampert et al., 2009], a Steiner tree based selection of object candidate regions [Russakovsky and Ng, 2010], and the maximum-weight connected subgraph for the detection of non-boxy objects [Vijayanarasimhan and Grauman, 2011]. The main purpose of these methods is to efficiently enumerate candidate regions to which object classifiers are applied. For the cosegmentation algorithm in chapter 7, we exploit welfare maximization in combinatorial auction [Cramton et al., 2005] to efficiently solve the image cosegmentation problem.

Reconstruction and Applications of Photo Storylines

Story structure: The structure of story has been studied much in psychology because it plays an important role in memory tasks and human development [Mandler and Johnson, 1977; Trabasso and Broek, 1985]. Building the story structure is a well-known psychological mechanism facilitating memory and recall; higher structured stories tend to be more easily memorized or recalled. In the developmental psychology, it is supported that as children get older, they use more sophisticated story structure when telling their experiences. A story is usually conceived as a sequence of events that are causally and temporally related one another. Thus, trees or graphs have been a common model for story representation [Mandler and Johnson, 1977; Trabasso and Broek, 1985; Riedl and Young, 2006].

Storylines from text data: In the recent research of web mining, much work has been done to extract diverse threads of stories from online text collections such as news articles and scientific papers [Ahmed et al., 2011; Gillenwater et al., 2012; Shahaf and Guestrin, 2010; Shahaf et al., 2012]. However, Web image collections have not been explored much yet. In [Wang et al., 2012a], images are jointly used with texts to generate storylines; however, only primitive image features are used, and more importantly, the algorithm is tested with a cleaned small dataset of 355 images. On the other hand, we leverage large-scale online images to reconstruct storylines. The details can be found in chapter 8 and chapter 9.

Storylines from landmark photo collections: The landmark photos taken by a number of tourists have been one of most favorable uses of community photos in computer vision research. Each photo stream is a collection of photos about one tourist's experience, and they are used for various purpose in computer vision, including 3D models of landmarks in Photosynth [Snavely et al., 2010], geolocation of tourists' image sequences [Kalogerakis et al., 2009], world-scale landmark recognition in [Zheng et al., 2010], and image-based location estimation [Chen and Grauman, 2011]. One contribution of our research is to broaden the applicability of community photos to the storyline reconstruction for outdoor recreational activities and competing brands.

Storyline detection and segmentation in videos: The storyline mining has been studied much in video analysis, including sports videos [Gupta et al., 2009] and News videos [Misra et al., 2010]. Especially, the work of [Gupta et al., 2009] proposes a storyline model that is formed by graph grammar to model causal relationships between visual grounds. However, since videos usually contain only a small number of specified actors in a single scene, the model can take advantage of strong spatio-temporal constraints and synchronized captions, which are not available in community photo collections contributed by millions of general public.

Event detection from Web data: There has been several important previous work using community photos for event detection. Rattenbury *et al*. [Rattenbury et al., 2007] are interested in place and event detection by analyzing temporal and spatial distributions of tags' usages associated with photos. Jin *et al*. [Jin et al., 2010] leverage Flickr images to discover and predict social trends in the areas of politics, economics, and marketing. They consider image uploading and downloading as implicit votes for the subjects of the images. Sing *et al*. [Singh et al., 2010] propose a method called *social pixels* to visualize spatio-temporal phenomena in Twitter and Flickr posts.

Free associations: The free association is a well-known psychological technique in which given a cue, a human subject recalls the list of words that comes to mind without any editing or censoring [Nelson et al., 2004, 2005]. The use of free association was originally pioneered by Sigmund Freud for the purpose of Psychotherapy; given a question, patients write down whatever thoughts come to mind, from which therapists learn more about how patients think and feel. Since then, the free association has been used for other practical cognitive studies.

In Psychology, the free association technique has been used for mapping the association links between lexical concepts, in order to study memory tasks such as recall and recognition [Nelson et al., 2004, 2005]. In NLP research, Vickrey *et al*. [Vickrey et al., 2008] develop an online word game that exploits the free association technique, in order to obtain labels of semantic relationships between pairs of words (*e.g*. *wing* is a part of *dragon*). Their results show that the qualities of semantic relations mined by the free associations is higher than those of other methods.

Another interesting use of free associations is measuring brand associations in marketing [Chen, 2001; Danes et al., 2010; Till et al., 2011]. In this technique, subjects are asked to freely answer their feelings and thoughts about a given brand name. (*e.g*. What comes to mind when you think of *Nike*?) Our work in chapter 10 is also based on this *free association* idea, because we view the Web photos tagged with a brand name by anonymous users as their candid pictorial impressions to the brand.

Analysis of product images: Recently, with the exploding interests in electronic commerce, computer vision techniques have widely applied to analyze product images for commercial appli-

cations. Some notable examples include the product image search and ranking [Jing and Baluja, 2008], the logo and product detection in natural images [Gao et al., 2009; Kang et al., 2012; Kleban et al., 2008; Sanyal and Srinivasan, 2007], the attribute discovery in product images [Berg et al., 2010], and clothing parsing in fashion photos [Yamaguchi et al., 2012]. One important problem in this thesis is to extract and visualize the core concepts of the brands from extremely diverse online pictures in chapter 10. Our work differs from most of past research, which has focused on detecting a fixed number of specified product models or logos in the images. In our work, it is important to mine the visual topics that do not explicitly contain the products but reflect general public's thoughts, feelings, or experiences over the brands (*e.g.* sponsored yacht competition scenes in the *Rolex* image set).

Prediction of users' economic behaviors using Web data: Recently, many studies have demonstrated that Web data generated by general users can predict their economic behaviors in real world. For example, Choi and Varian [Choi and Varian, 2012] have found that Web search volumes obtained from GOOGLE TRENDS are often correlated with various economic indicators, such as automobile sales, unemployment claims, and consumer confidence. Goel *et al.* [Goel et al., 2010] also show that online search queries can forecast consumers' near-future behaviors in box-office revenues, video game sales, and the ranks of songs on music charts. Similarly, Bordino *et al.* [Bordino et al., 2012] study the correlation between trading volumes of stocks and their daily query volumes. In this thesis, we study the visualization of brand associations from online pictures contributed by general public; our work is novel in that any analysis or predication about brands has not been explored before.

Part I

Understanding Temporal Trends of Image Collections

Part I – Understanding Temporal Trends of Image Collections

In this part, we discuss the methods to understand the temporal trends of large-scale image collections, which are gathered by querying topic keywords from photo sharing sites such as Flickr. Here we do not address any sub-image level analysis such as object detection or segmentation, which will be the main theme of the next part.

This part consists of two chapters. First, we propose a nonparametric approach to modeling and analysis of topical evolution in image collections. With experiments on more than 9 millions of images of 47 topics from Flickr, we show that our method successfully perform the subtopic outbreak detection to point out when the topical contents of images rapidly change, and improve image classification performance using temporal context. We also show that the images can often be a more reliable source of information than tag texts to detect topical evolution.

Second, we investigate a time-sensitive image retrieval problem, in which given a query keyword, a query time point, and optionally user information, we retrieve the most relevant and temporally suitable images from the database. Inspired by recently emerging interests on query dynamics in information retrieval research, our time-sensitive image retrieval algorithm can infer users' implicit search intent better and provide more engaging and diverse search results according to temporal trends of Web photos. Furthermore, we extend our algorithm to address the *Web image prediction* problem of predicting the images that are likely to be popular on the Web at any given future time point.

Chapter 3

Analyzing Dynamic Behaviors of Web Photo Sets

3.1 Introduction

Suppose that we download millions of images retrieved by the query term *apple* from Flickr, and distribute them on the timeline according to their associated timestamps. The apple topic consists of various subtopics (*e.g*. *fruit*, *logo*, *laptop*, *tree*, and *iPhone*), and their popularity changes over time. In Fig.3.1, we choose the central images of five subtopics of the *apple* and measure the similarity changes with the image samples at each time step. As *Google trends* reveal the popularity variation of a query term in the search volumes, we can easily obtain the affinity changes of each subtopic in the *apple* image set. The *fruit apple* photos occur relatively stationary on the timeline, whereas the *iPhone* photos show bursty occurrences according to specific events such as the release of new models or Steve Jobs' talks.

The main objectives of this work are as follows. First, we propose a nonparametric approach to modeling and analysis of temporal evolution of topics in Web image collections (section 3.2.3). Second, we show that understanding image dynamics is useful to solve novel problems such as the subtopic tracking and the subtopic outbreak detection (section 3.3.2). Third, we present that the images can be a more reliable and delicate source of information to detect topical evolution than

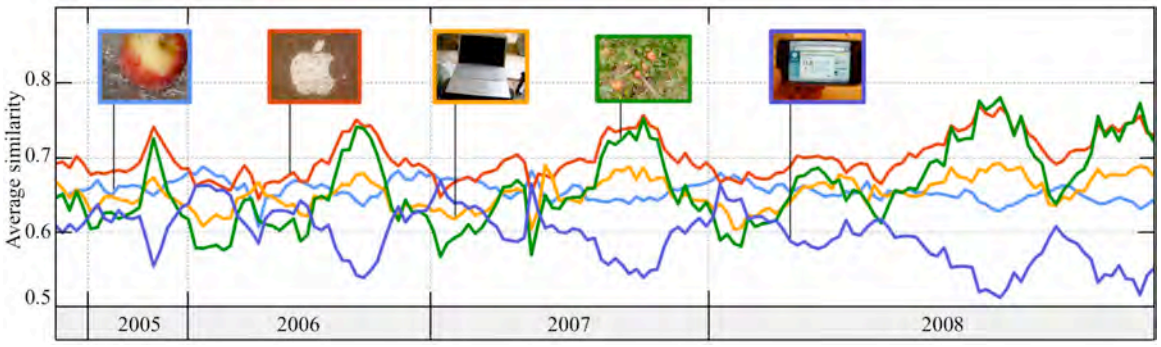


Figure 3.1: A *Google trends*-like visualization of the subtopic evolution in the *apple* Flickr images. Here we consider five subtopics of the *apple*: *fruit* (blue), *logo* (red), *laptop* (orange), *tree* (green), and *iphone* (purple). We choose the central images of each subtopic, and measure their average similarity with image samples at each time step. The *fruit* subtopic is stable along the timeline, whereas the *iphone* subtopic highly fluctuates.

associated tag texts (section 3.3.3). Finally, we show that the classification performance can be improved using the *temporal association* inspired by human vision studies [Becker, 1999; Sinha et al., 2006; Wallis and Bulthöff, 2001] (section 3.3.4).

Our approach is motivated by the recent success of the nonparametric methods [Liu et al., 2009a; Torralba et al., 2008] that are powered by large databases. Instead of using sophisticated parametric topic models [Blei and Lafferty, 2006; Wang and McCallum, 2006], we represent the images with timestamps in the form of a *similarity network* [Kim and Torralba, 2009], in which vertices are images and edges connect the temporally close and visually similar images. Thus, our approach is able to perform diverse temporal analysis without solving complex inference problems. For example, a simple information-theoretic measure of the network can be used to detect subtopic outbreaks, which point out when the topical evolution speed abruptly changes. The *temporal context* is also easily integrated with the classifier training in a framework of the Metropolis-Hastings algorithm.

The network generation is based on the sequential Monte Carlo (*i.e.* particle filtering) [Arulampalam et al., 2002; Isard and Blake, 1998], in which, the posterior (*i.e.* subtopic distribution) at a particular time step is represented by a set of weighted image samples. We track the similar subtopics (*i.e.* clusters of images) in consecutive posteriors along the timeline, and create edges between them. Our sampling based representation is practically beneficial for the following reasons. First, since we deal with unordered natural images on the Web, any Gaussian or linearity assumption does not hold and the multiple peaks of distributions are unavoidable. Second, we can easily control the tradeoff between accuracy and speed by managing the number of samples and parameters in the transition model. Third, our algorithm is easily parallelizable by running multiple sequential Monte Carlo trackers with different initialization and parameters. Finally, our approach is also scalable and fast; the run time is linear with the number of images.

For evaluation, we download more than 9 millions of images of 47 topics from Flickr. Most standard datasets in computer vision research [Everingham et al., 2010; Russell and Torralba, 2009] have not yet considered the importance of temporal context. While several datasets have recently introduced *spatial contexts* as a fundamental cue to recognition [Russell and Torralba, 2009], the support for the temporal context has been still largely under-addressed. Our experiments clearly show that the proposed approach is successful to model the temporal behaviors of large-scale image collections, and leverage them to achieve several novel or existing computer vision problems better.

3.2 Network Construction by Sequential Monte Carlo

3.2.1 Image Description and Similarity Measure

Each image is represented by two types of descriptors, which are spatial pyramids of SIFT visual words [Liu et al., 2008] and HOG [Bosch et al., 2007]. We use the codes provided by the original authors. A dictionary of 200 visual words is formed by applying K-means to randomly selected SIFT descriptors [Liu et al., 2008]. Every pixel of an image is densely assigned to the nearest visual word in the dictionary. Then visual words are binned using a two-level spatial pyramid. The oriented gradients are computed by Canny edge detection and Sobel mask [Bosch et al., 2007]. The HOG descriptor is then discretized into 20 orientation bins in the range of [0°,180°]. Finally, the

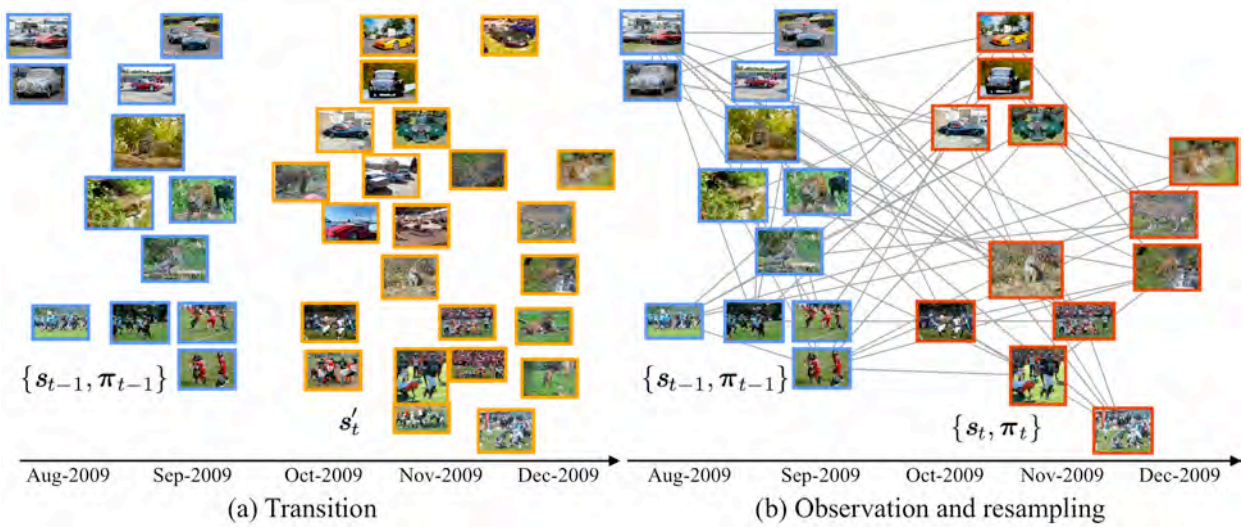


Figure 3.2: An overview of the SMC based network construction for the *jaguar* topic. The subtopic distribution at each time step is represented by a set of weighted image samples (*i.e.* posterior) $\{\boldsymbol{s}_t, \boldsymbol{\pi}_t\}$. In this example, the posterior $\boldsymbol{s}_{t-1}$ consists of image samples of the *animal*, *car*, and *football* subtopics. (a) The transition model generates new posterior candidates $\boldsymbol{s}'_t$ from $\boldsymbol{s}_{t-1}$. (b) The observation model discovers $\boldsymbol{\pi}'_t$ of $\boldsymbol{s}'_t$ and the resampling step obtains $\{\boldsymbol{s}_t, \boldsymbol{\pi}_t\}$ from $\{\boldsymbol{s}'_t, \boldsymbol{\pi}'_t\}$. Finally, the network is built by similarity matching between two consecutive posteriors $\boldsymbol{s}_{t-1}$ and $\boldsymbol{s}_t$.

HOG descriptors are binned using a three-level spatial pyramid. The similarity measure between a pair of images is the cosine similarity, which is calculated by the dot product of a pair of L_2 normalized descriptors.

3.2.2 Problem Statement

The input of our algorithm is a set of images $\mathcal{I} = \{I_1, \ldots, I_N\}$ and associated timestamps $\mathcal{T} = \{T_1, \ldots, T_N\}$. The main goal is to generate an $N \times N$ sparse similarity network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ by using the Sequential Monte Carlo (SMC) method. Each vertex in $\mathcal{V}$ is an image in the dataset. The edge set $\mathcal{E}$ is created between the images that are visually similar and temporally close within a certain interval that is defined by the *transition model* of the SMC tracker (Section 3.2.3). The weight set $\mathcal{W}$ is discovered by the similarity between descriptors of images (Section 3.2.1). For sparsity, each image is connected to its k-nearest neighbors with $k = a \log N$, where a is a constant (*e.g.* a =10).

3.2.3 Network Construction using Sequential Monte Carlo

Algorithm 1 summarizes the proposed SMC based Network construction. For better readability, we follow the notation of the *condensation* algorithm [Isard and Blake, 1998]. The output of each iteration of the SMC is the conditional subtopic distribution (*i.e.* posterior) at every step, which is approximated by a set of images with weights denoted by $\{\boldsymbol{s}_t, \boldsymbol{\pi}_t\} = \{s_t^{(i)}, \pi_t^{(i)}, i = 1, \ldots, M\}$ where M is the number of image samples. As a notation convention, we use superscripts

Algorithm 1: The SMC based network generation

Input: (1) A set of images $\mathcal{I}$ sorted by timestamps $\mathcal{T}$. (2) Start time T_0 and end time T_e. (3) Posterior size M. (4) Parameters for *drift*: $(\triangle M_\mu, \sigma^2)$.

Output: Network $\boldsymbol{G}$.

[*Initialization*]

1: Draw $s_0^{(i)} \sim N(T_0, \tau^2(T_0, 2M/3))$, $\pi_0^{(i)} = 1/M$ for $i = 1, \ldots, M$.

while $\mu_t < T_e$, *(*$\mu_0 = T_0$ *and* $\mu_t = \mu_{t-1} + \tau(\mu_{t-1}, \triangle M_\mu)$*).* **do**

  [*Transition*]

  foreach $s_{t-1}^{(i)} \in \mathbf{s}_{t-1}$ *starting with* $\mathbf{x}^{(i)} = \emptyset$ **do**

    repeat

      2: Draw $x \sim N(x; \mu_t, \sigma^2) \times \Gamma(x; \alpha_{t-1}^{(i)}, \beta_{t-1}^{(i)})$, where $\alpha_{t-1}^{(i)} \propto 1/\pi_{t-1}^{(i)}, \beta_{t-1}^{(i)} = \mu_t/\alpha_{t-1}^{(i)}$.

      3: $\boldsymbol{x}^{(i)} \leftarrow x$ with probability of $w(s_{t-1}^{(i)}, x)$.

    until $|\mathbf{x}^{(i)}| = m_i = 2M \times \pi_{t-1}^{(i)}$. *Then,* $\mathbf{s}'_t \leftarrow \mathbf{x}^{(i)}$.;

  [*Observation*]

  4: Compute self-similarity graph $\boldsymbol{W}_t$ of $\boldsymbol{s}'_t$. Row-normalize $\boldsymbol{W}_t$ to $\widetilde{\boldsymbol{W}}_t$.

  5: Get stationary distribution $\boldsymbol{\pi}'_t$ by solving $\boldsymbol{\pi}'_t = \widetilde{\boldsymbol{W}}_t^T \boldsymbol{\pi}'_t$ with $\|\boldsymbol{\pi}'_t\|_1 = 1$.

  [*Resampling*]

  6: Resample $\{\boldsymbol{s}_t, \boldsymbol{\pi}_t\}_{i=1}^M$ from $\{\boldsymbol{s}'_t, \boldsymbol{\pi}'_t\}$ by *systematic sampling*. Normalize $\boldsymbol{\pi}_t$.

  7: $\boldsymbol{G} \leftarrow \boldsymbol{W}_t(\boldsymbol{s}_t, \boldsymbol{s}_t), \boldsymbol{W}_{t-1,t}(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t)$. Then convert $\boldsymbol{G}$ into a k-NN graph.

to denote the image numbers and subscripts to denote the iterations. Note that our SMC does not explicitly solve the *data association* during the tracking. In other words, we do not assign a subtopic membership to each image in $\boldsymbol{s}_t$. However, it can be easily obtained later by applying clustering to the subgraph of $\boldsymbol{s}_t$.

Fig.3.2 shows a downsampled example of a single iteration of the posterior estimation. At every iteration, the SMC generates a new posterior $\{\boldsymbol{s}_t, \boldsymbol{\pi}_t\}$ by running *transition*, *observation*, and *resampling* steps.

The image data are severely unbalanced on the timeline. (*e.g.* There are only a few images within a month in 2005 but a large number of images within even a week in 2008). Thus, in our experiments, we bin the timeline by the number of images instead of a fixed time interval. (*e.g.* The timeline may be binned by every 3000 images rather than by a month). The function $\tau(T_i, m)$ denotes the timestamp of the m-th image later from the image at time T_i.

Initialization

We first manually decide the starting time T_0 for the SMC tracker. The initialization step then samples the initial posterior $\boldsymbol{s}_0$ from the prior at T_0, which is set by a Gaussian distribution $N(T_0, \tau^2(T_0, 2M/3))$ on the timeline. It means that $2M$ numbers of images around T_0 have nonzero probabilities to be selected as one of $\boldsymbol{s}_0$. The initial $\boldsymbol{\pi}_0$ is uniformly set to $1/M$.

Transition Model

The transition model generates posterior candidates $\boldsymbol{s}'_t$ rightward on the timeline from the previous $\{\boldsymbol{s}_{t-1}, \boldsymbol{\pi}_{t-1}\}$ (See Fig.3.2.(a) for an example). Each image $s^{(i)}_{t-1}$ in $\boldsymbol{s}_{t-1}$ recommends m_i numbers of images that are similar to itself as the members of candidates set $\boldsymbol{s}'_t$. A more weighted image $s^{(i)}_{t-1}$ recommends more images for $\boldsymbol{s}'_t$. ($\sum_i m_i = 2M$ and $m_i \propto \pi^{(i)}_{t-1}$). At this stage, we generate $2M$ candidates (*i.e.* $|\boldsymbol{s}'_t| = 2M$), and the observation and resampling steps reduce it to be $|\boldsymbol{s}_t| = M$ while computing weights $\boldsymbol{\pi}_t$.

Similarly to the condensation algorithm [Isard and Blake, 1998], the transition consists of deterministic *drift* and stochastic *diffusion*. The *drift* describes the transition tendency of the overall $\boldsymbol{s}'_t$ (*i.e.* how far the $\boldsymbol{s}'_t$ is located from the $\boldsymbol{s}_{t-1}$ on the timeline). The *diffusion* assigns a random transition of an individual image. The *drift* and the *diffusion* are modeled by a Gaussian distribution $N(\mu_t, \sigma^2)$ and a Gamma distribution $\Gamma(\alpha, \beta)$, respectively. The final transition model is the product of these two distributions [Hinton, 2002] as follows.

$$P^{(i)*}_t(x) = N(x; \mu_t, \sigma^2) \times \Gamma(x; \alpha^{(i)}_{t-1}, \beta^{(i)}_{t-1}) \tag{3.1}$$

where the asterisk of $P^{(i)*}_t(x)$ in Eq.(3.1) means that it is not normalized. Renormalization is not required since we will use *importance sampling* to sample images on the timeline with the target distribution (See the next subsection with Fig.3.3 for the detail).

In Eq.(3.1), the mean μ_t of $N(\mu_t, \sigma^2)$ is updated at every iteration by $\mu_t = \mu_{t-1} + \tau(\mu_{t-1}, \triangle M_\mu)$ where $\triangle M_\mu$ is the control parameter for the speed of the tracking. The higher $\triangle M_\mu$, the further $\boldsymbol{s}'_t$ is located from $\boldsymbol{s}_{t-1}$ and the fewer iterations are executed until completion. The variance σ^2 of $N(\mu_t, \sigma^2)$ controls the spread of $\boldsymbol{s}'_t$ along the timeline. A higher σ^2 results in a $\boldsymbol{s}'_t$ that includes images with a longer time range.

A Gamma distribution $\Gamma(\alpha, \beta)$ is usually used to model the time required for α occurrences of events that follow a Poisson process with a constant rate β. In our interpretation, given an image stream, we assume that the occurrence of images of each subtopic follows the Poisson process with β. Then, $\Gamma(\alpha^{(i)}_{t-1}, \beta^{(i)}_{t-1})$ of Eq.(3.1) indicates the time required for the next α images that have the same subtopic with $s^{(i)}_{t-1}$ in the image stream. Based on this intuition, $\alpha^{(i)}_{t-1}$ for each $s^{(i)}_{t-1}$ is adjusted; a smaller $\alpha^{(i)}_{t-1}$ is chosen for the image $s^{(i)}_{t-1}$ with higher $\pi^{(i)}_{t-1}$ since the similar images to a more weighted $s^{(i)}_{t-1}$ are likely to occur more frequently in the dataset. The mean of Gamma distribution of each $s^{(i)}_{t-1}$ is aligned with the mean μ_t of $N(\mu_t, \sigma^2)$. Therefore, we set $\beta^{(i)}_{t-1} = \mu_t / \alpha^{(i)}_{t-1}$ given that the mean of Gamma distribution is $\alpha\beta$.

The main reason to adopt the *product model* rather than the *mixture model* in Eq.(3.1) is as follows. The *product model* only has a meaningful probability for an event when none of its component distribution has a low probability. (*i.e.* if one of two distributions has zero probability, their product does as well). It is useful in our application because the product with the Gaussian of the *drift* sets almost zero probability for the images outside 3σ from μ_t, which can prevent the sampled images from severely spreading along the timeline.

In sum, for each $s^{(i)}_{t-1}$, we sample an image x by the distribution of Eq.(3.1), which constrains the position of x on the timeline. In addition, x is required to be visually similar to its recommender $s^{(i)}_{t-1}$. Thus, x is accepted with probability of $w(s^{(i)}_{t-1}, x)$, which is the cosine similarity between the

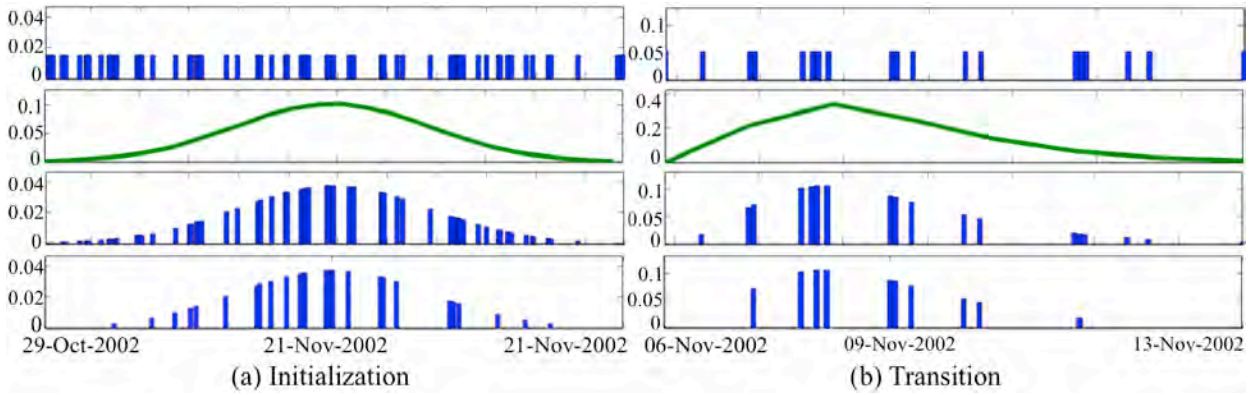


Figure 3.3: An example of sampling images on the timeline during (a) the initialization and (b) the transition. From top to bottom: The first row shows the image distributions along the timeline. The images are regarded as the samples ($\{x^{(r)}\}_{r=1}^{R}$) from a proposal distribution $Q^*(x)$. They are equally weighted (*i.e.* $Q^*(x^{(r)}) = 1$). The second row shows the target distribution $P^*(x)$. (*e.g.* Gaussian in (a) and the product of Gaussian and Gamma in (b)). The third row shows the image samples weighted by $P^*(x^{(r)})/Q^*(x^{(r)})$. The fourth row shows the images chosen by *systematic sampling* [Arulampalam et al., 2002].

descriptors of $s_{t-1}^{(i)}$ and x. This process is repeated until m_i number of samples are accepted for each $s_{t-1}^{(i)}$. Algorithm 1 summarizes the major steps of the transition model.

Sampling Images with Target Distribution

During the initialization and the transition, we sample a set of images on the timeline from a given target distribution $P^*(x)$. (*e.g.* Gaussian in the initialization and the product of Gaussian and Gamma in the transition). Fig.3.3 shows an example of our *importance sampling* method [MacKay, 2002], which is particularly useful for our transition model since there is no closed form of the product of Gaussian and Gamma distributions and its normalization is not straightforward.

Observation Model

The goal of the observation model is to compute weights $\boldsymbol{\pi}'_t$ for $\boldsymbol{s}'_t$. First, we obtain the similarity matrix $\boldsymbol{W}_t$ of $\boldsymbol{s}'_t$ by computing pairwise cosine similarity between the images of $\boldsymbol{s}'_t$. Then, $\boldsymbol{\pi}'_t$ is the stationary distribution of $\boldsymbol{W}_t$ by solving $\boldsymbol{\pi}'_t = \widetilde{\boldsymbol{W}}_t^T \boldsymbol{\pi}'_t$ with $\|\boldsymbol{\pi}'_t\|_1 = 1$, where $\widetilde{\boldsymbol{W}}_t$ is row-normalized from $\boldsymbol{W}_t$ so that $\widetilde{w}_{ij} = w_{ij} / \sum_k w_{ik}$.

Resampling

The final posterior $\{\boldsymbol{s}_t, \boldsymbol{\pi}_t\} = \{s_t^{(i)}, \pi_t^{(i)}\}_{i=1}^{M}$ is resampled from $\{\boldsymbol{s}'_t, \boldsymbol{\pi}'_t\}$ by running the *systematic sampling* [Arulampalam et al., 2002] on $\boldsymbol{\pi}'_t$. Then $\boldsymbol{\pi}_t$ is normalized so that its sum is one. The network $\boldsymbol{G}$ stores $\boldsymbol{W}_t(\boldsymbol{s}_t, \boldsymbol{s}_t)$ and $\boldsymbol{W}_{t-1,t}(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t)$ (*i.e.* the similarity matrix between two consecutive posteriors $\boldsymbol{s}_{t-1}$ and $\boldsymbol{s}_t$). As discussed in section 3.2.2, each vertex in $\boldsymbol{G}$ is connected to only its k-nearest neighbors.

3.3 Experiments

3.3.1 Evaluation setting

Table 3.1 shows 47 topics of our Flickr dataset. The topic name is identical to the query word. We downloaded all the images retrieved by the query word from Flickr (*i.e.* all the images retrieved when a query word is typed in Flickr's search box without any option change). For the timestamp, we use the *date_taken* field of each image that Flickr provides.

We generate the similarity network of each topic by using the proposed SMC based tracking. The runtime is $O(NM)$ where M is the size of posterior and $M \ll N$ (*i.e.* $1000 \leq M \leq 5000$ in our experiments). The network construction is so fast that, for example, it took about 4 hours for the *soccer* topic with $N = 1.1 \times 10^6$ and $M = 5,000$ in our matlab implementation on a single PC. The analysis of the network is also fast since most network analysis algorithms depend on the number of nonzero elements, which is $O(N \log N)$.

3.3.2 Results on Evolution of Subtopics

Fig.3.4 shows the subtopic evolution examples of two topics, the *big+ben* and the *korean*. As discussed in previous section, the SMC tracker iteratively generates the posterior sets $\{\boldsymbol{s}_0, \ldots, \boldsymbol{s}_L\}$. For each $\boldsymbol{s}_t$, we discover five clusters from each posterior by applying spectral clustering to the subgraph $\boldsymbol{G}_t(\boldsymbol{s}_t, \boldsymbol{s}_t)$. Obviously, each topic shows its own intrinsic dynamic behaviors. Some topics such as the *big+ben* are stationary and coherent whereas others like the *korean* are highly diverse and variant.

Subtopic Outbreak Detection

The subtopic outbreak detection is an important task in Web mining since it reflects the change of information flows and people's interests. We perform the outbreak detection by calculating an information-theoretic measure of link statistics of the network. Note that the consecutive posterior

Nation	*brazilian* (119,620), *jewish* (165,760), *korean* (254,386), *swedish* (94,390), *spanish* (322,085)
Place	*amazon* (160,008), *ballpark* (340,266), *big+ben* (131,545), *grandcanyon* (286,994), *pisa* (174,591), *wall+street* (177,181), *white+house* (241,353)
Animal	*butterfly+insect* (69,947), *cardinals* (177,884), *giraffe+zoo* (53,591), *jaguar* (122,615), *leopard* (121,061), *lobster* (144,596), *otter* (113,681), *parrot* (175,895), *penguin* (257,614), *rhino* (96,799), *shark*(345,606)
Object	*classic+car* (265,668), *keyboard* (118,911), *motorbike* (179,855), *pagoda* (128,019), *pedestrian* (112,116), *sunflower* (165,090), *television* (157,033)
Activity	*picnic* (652,539), *soccer* (1,153,969), *yacht* (225,508)
Abstract	*advertisement* (84,521), *economy* (61,593), *emotion* (119,899), *fine+art* (220,615), *horror* (157,977), *hurt* (141,249), *politics* (181,836)
Hot topic	*apple* (713,730), *earthquake* (65,375), *newspaper* (165,987), *simpson* (106,414), *starbucks* (169,728), *tornado* (117,161), *wireless* (139,390)

Table 3.1: 47 topics of our Flickr dataset. The numbers in parentheses indicate the numbers of downloaded images per topic. 9,751,651 images are collected in total.

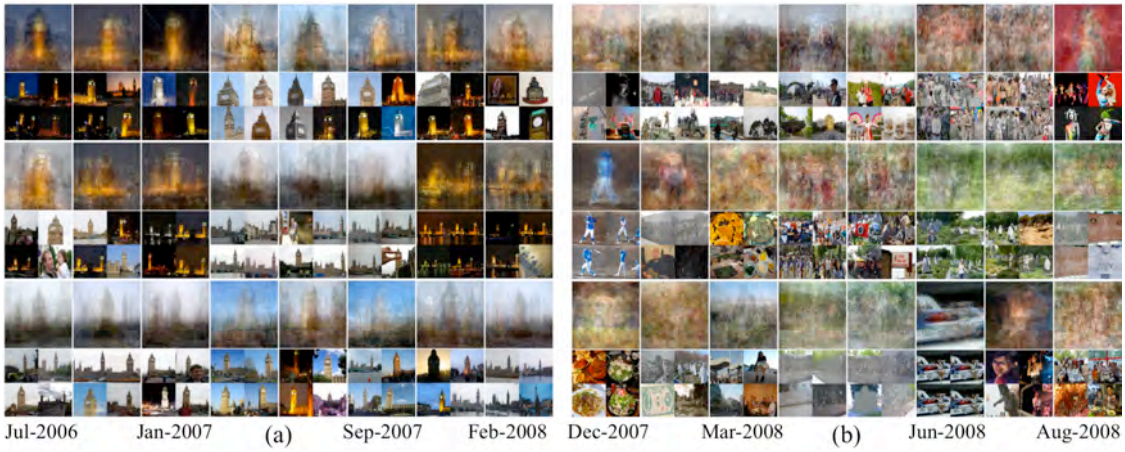


Figure 3.4: The subtopic evolution of (a) the *big+ben* and (b) the *korean* topic. In each column, we show top three out of five clusters of each $\boldsymbol{s}_t$ with the average images in the top, and four highest ranked images in the cluster in the bottom. The *big+ben* is relatively stationary and coherent, whereas the *korean* is dynamic and includes diverse subtopics such as *sports*, *food*, *events*, *buildings*, and the *Korean war memorial park*.

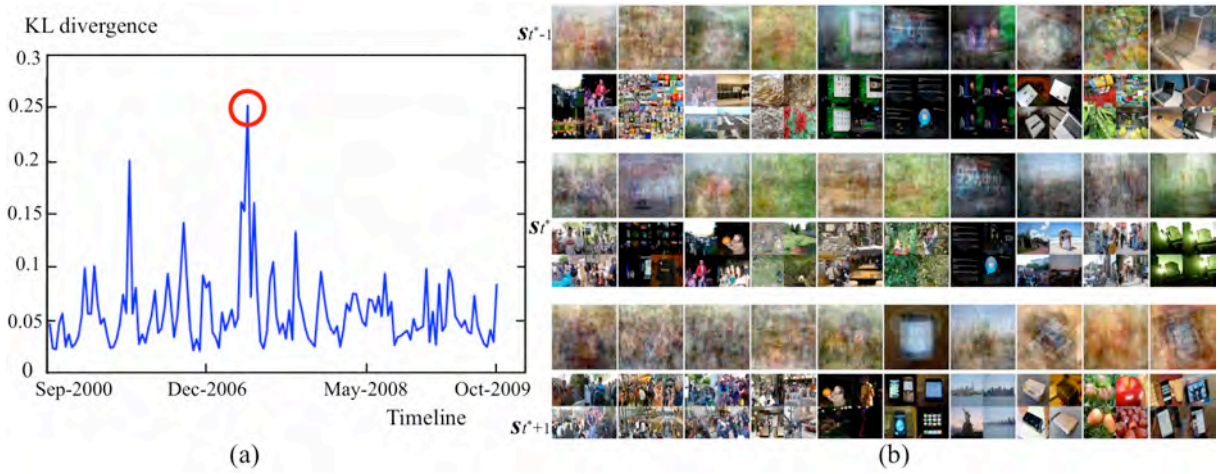


Figure 3.5: An example of the subtopic outbreak detection. (a) The variation of KL divergences for the *apple* topic. The highest peak is observed at step $t^* = 63$ ([May-2007, Jun-2007] with the median of 11-Jun-2007). (b) The subtopic changes around the highest peak. We show ten subtopics (*i.e.* image clusters) of $\boldsymbol{s}_{t^*-1}$, $\boldsymbol{s}_{t^*}$, and $\boldsymbol{s}_{t^*+1}$ from top to bottom. In each set, the first row shows the average images of top 15 images, and the bottom row shows top four highest ranked ones of each subtopic. Several clusters of *Steve Jobs' presentation* appear in $\boldsymbol{s}_{t^*-1}$ and $\boldsymbol{s}_{t^*}$, but vanish in $\boldsymbol{s}_{t^*+1}$. Rather, the *crowds in the street* (*i.e.* cluster $1 \sim 4$) and the *iphone* (*i.e.* cluster 6, 8, 10) newly emerge in $\boldsymbol{s}_{t^*+1}$.

sets are linked in our network. (*i.e.* $\boldsymbol{s}_{t-1}$ is connected to $\boldsymbol{s}_t$, which is linked to $\boldsymbol{s}_{t+1}$). The basic idea of our outbreak detection is that if the subtopic distributions at step $t-1$ and $t+1$ are different each other, then the degree distribution of $\boldsymbol{s}_t$ to $\boldsymbol{s}_{t-1}$ ($\boldsymbol{f}_{t,t-1}$) and that of $\boldsymbol{s}_t$ to $\boldsymbol{s}_{t+1}$ ($\boldsymbol{f}_{t,t+1}$) are dissimilar as well. Both $\boldsymbol{f}_{t,t-1}$ and $\boldsymbol{f}_{t,t+1}$ are $|\boldsymbol{s}_t| \times 1$ histograms, each element of which is the sum of edge weights of a vertex in $\boldsymbol{s}_t$ with $\boldsymbol{s}_{t-1}$ and $\boldsymbol{s}_{t+1}$, respectively. In order to measure the difference between $\boldsymbol{f}_{t,t-1}$

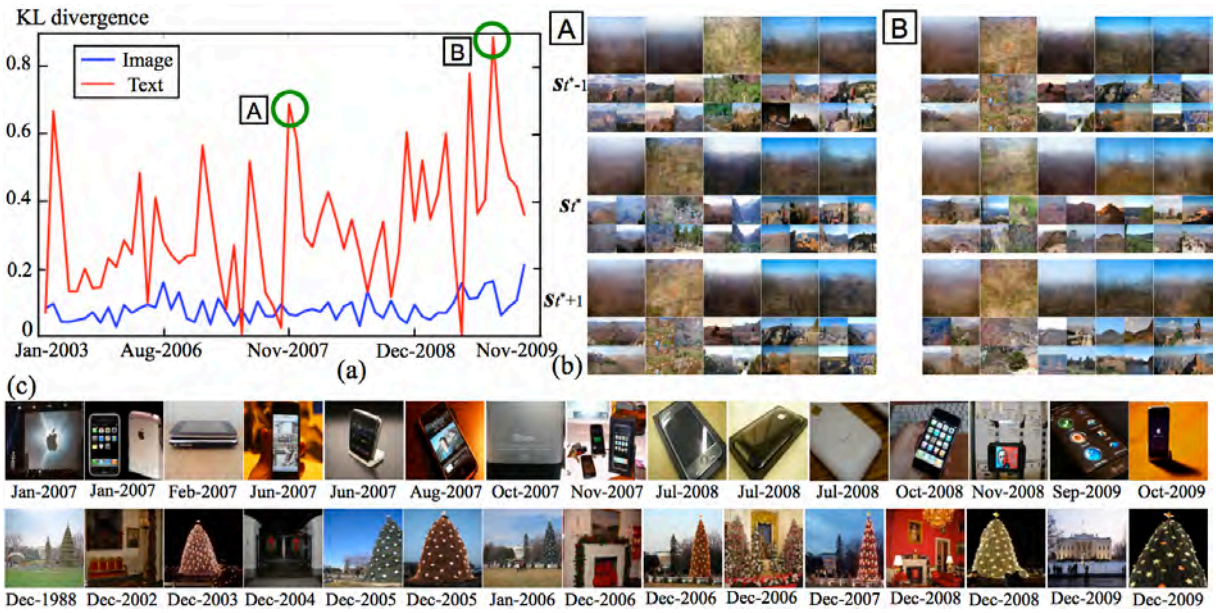


Figure 3.6: The comparison between the outbreak detection results using images and associated text tags. (a) The variation of the KL divergence for the *grandcanyon* topic. The KL divergences of images are stationary on the timeline whereas those of texts highly fluctuate. (b) The subtopic changes around the two highest peaks **A** (05-Nov-2007) and **B** (16-Aug-2009). We show five subtopics of $\boldsymbol{s}_{t^*-1}$, $\boldsymbol{s}_{t^*}$, and $\boldsymbol{s}_{t^*+1}$, in which very little visual variation is observed. (c) We show 15 selected images of two groups tagged by *apple+new+iphone* (first row) and *whitehouse+christmas* (second row) in a chronological order. We observe the gradually upgraded appearance of the *iphone*.

and $\boldsymbol{f}_{t,t+1}$, we use *Kullback-Leibler* (KL) divergence:

$$D_{KL}(\boldsymbol{f}_{t,t+1} \,\|\, \boldsymbol{f}_{t,t-1}) = \sum_{i \in \boldsymbol{s}_t} \boldsymbol{f}_{t,t+1}(i) \log \frac{\boldsymbol{f}_{t,t+1}(i)}{\boldsymbol{f}_{t,t-1}(i)} \tag{3.2}$$

where a higher $D_{KL}(\boldsymbol{f}_{t,t+1} \,\|\, \boldsymbol{f}_{t,t-1})$ indicates a higher subtopic variation from $\boldsymbol{s}_{t-1}$ to $\boldsymbol{s}_{t+1}$.

Fig.3.5.(a) shows an example of KL divergence changes along 142 steps of the *apple* tracking. The peaks of the KL divergence indicate the radical subtopic changes from $\boldsymbol{s}_{t-1}$ to $\boldsymbol{s}_{t+1}$. We observed the highest peak at step $t^* = 63$, where $\boldsymbol{s}_{t^*}$ is distributed in [May-2007, Jun-2007]. Fig.3.5.(b) represents ten subtopics of $\boldsymbol{s}_{t^*-1}$, $\boldsymbol{s}_{t^*}$, and $\boldsymbol{s}_{t^*+1}$, which are significantly different one another.

3.3.3 Comparison with Text Analysis

In this section, we empirically compare the image-based topic analysis with the text-based one. One may argue that similar topical evolution can be also detected by analyzing the associated texts with images. However, our experiments show that the associated texts cannot fully characterize the information from the images. First of all, 13.7% of images in our dataset have no text tags. It may be reasonable because the Flickr is a photo-sharing site, and thus its users care less about text annotations. As a more compelling justification, we perform the outbreak detection task in previous

section using images and their associated tags. Using the same algorithm, the only difference between the two tests is the features: the spatial pyramids of SIFT and HOG for images and term frequency histograms for texts. Fig.3.6.(a) shows an example of outbreak detection using images and texts for the *grandcanyon* topic, which is one of the most stationary and coherent topics in our dataset (*i.e.* no matter when the images are taken, the majority of them are taken for the scenes of the *Grand Canyon*). The image-based analysis successfully detects its intrinsic stationary behavior. However, the plot for text tags highly fluctuates mainly because tags are subjectively assigned by different users with little consensus. Such mismatch between images and associated texts is a well-known noise source of Web image search.

Another important advantage of image-based temporal analysis is that it can convey more delicate information that is hardly captured by text descriptions. Fig.3.6.(b) shows two typical examples from the *apple* and the *white+house*. When a new *iphone* is released, the emergence of the *iphone* subtopic can be detected via both images and texts. However, the images can reveal more intuitively the upgraded appearance, new features, and visual context around the new events.

3.3.4 Temporal Association for classification

As studied in neuroscience research [Sinha et al., 2006; Wallis and Bulthöff, 2001], humans can perceive and remember better the temporally connected visual information rather than the discontinuous one. Inspired by this study, we perform a preliminary test about the effect of training using temporal consistency on the image classification task. Presumably, the subtopics that consistently appear along the timeline are likely to be more closely connected to the first meaning of the topic rather than the ones that are observed during only a short period of time. For example, the *fruit apple* is likely to steadily exist in the *apple* image set, which may be a more representative subtopic of the *apple* rather than a specific model of an early *Mac* computer. In this experiment, we compare the classification performance between using two different training sets for the extremely diverse Flickr images. The first training set is constructed by selecting the images that are temporally and visually associated, and the other set is randomly chosen without using any temporal information.

Since our similarity network links temporally close and visually similar images, the dominant subtopics and their central images correspond to large clusters and hub nodes of the graph, respectively. Therefore, we choose the images with high stationary probabilities as temporally and visually strengthened images, given that the stationary probability is a popular centrality measure of the node for the graph analysis. However, our network may not be complete for this purpose in that we only connect the images in the local neighborhood of temporal space. In order to cope with such incompleteness, we generate training sets by the Metropolis-Hasting (MH) algorithm as follows.

We first compute the stationary probability $\boldsymbol{\pi}_G$ of the network $\boldsymbol{G}$ by solving $\boldsymbol{\pi}_G = \widetilde{\boldsymbol{G}}^T \boldsymbol{\pi}_G$ with $\|\boldsymbol{\pi}_G\|_1 = 1$, where $\widetilde{\boldsymbol{G}}$ is the row-normalized $\boldsymbol{G}$. Since a general suggestion for a starting point in the MH algorithm is to begin around the modes of the distribution, we start from an image s_o with the highest $\boldsymbol{\pi}_G(s)$. Then, from a current image s, we sample a next candidate image s^* using a proposal distribution $q(s_1, s_2)$ that is based on a random surfer model as follows.

$$\alpha = \min\left(\frac{\boldsymbol{\pi}_G(s^*)q(s^*, s_{t-1})}{\boldsymbol{\pi}_G(s_{t-1})q(s_{t-1}, s^*)}, 1\right) \text{ where } q(i,j) = \lambda\widetilde{w}_{ij} + (1-\lambda)\boldsymbol{\pi}_G(j). \tag{3.3}$$

In Eq.3.3, the candidate s^* is chosen by following an outgoing edge of the s_{t-1} in the network $\widetilde{\boldsymbol{G}}$ with probability λ, but randomly resampling it according to the $\boldsymbol{\pi}_G$ with probability $1-\lambda$. A larger λ observes more the local link structure while a smaller λ relies on $\boldsymbol{\pi}_G$ more. The candidate s^* is accepted with probability α in Eq.(3.3) where $\widetilde{w}_{ij}$ is the element (i,j) of $\widetilde{\boldsymbol{G}}$. We repeat this process until the desired numbers of training samples are selected.

For binary classification tests, we generate the positive training set of each topic in two different ways. We sample 256 images by using the above MH method (called *Temporal* training) and randomly choose the same number of images (called *Random* training). For the negative training images, we randomly draw 256 images from the other topics of Flickr dataset. For the test sets, we use the images retrieved from Google Image Search by querying the same query words in Table

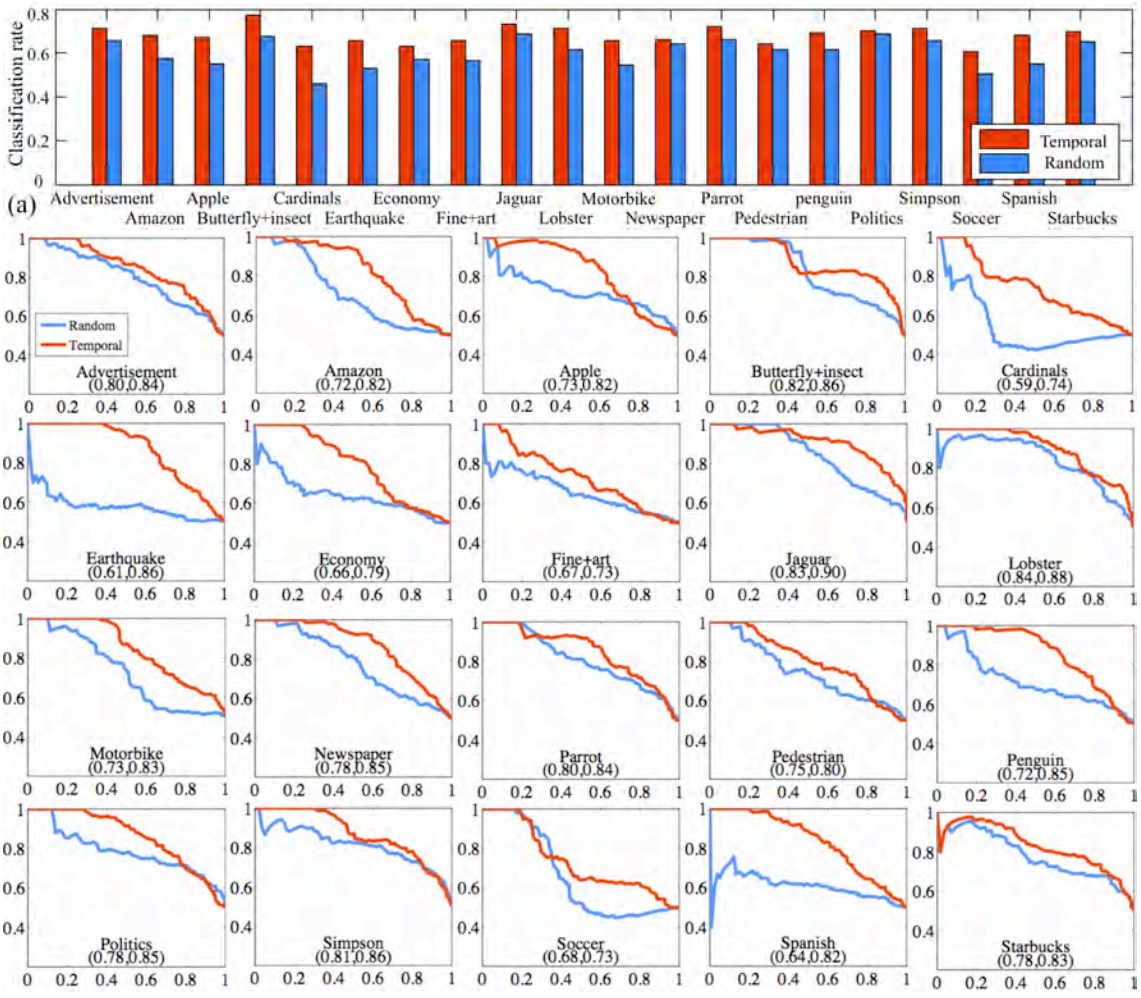


Figure 3.7: Comparison between the binary classification performance between the *Temporal* training and the *Random* training. (a) Classification accuracies of selected 20 topics. (b) Corresponding Precision-Recall curves. The number (n,m) underneath the topic name indicates the average precision of (*Random*, *Temporal*).

3.1. The Google Image Search provides relatively clean images in the highest ranking. Since we would like to test whether the temporally associated samples are better generalization of the topic, the Google images are more suitable for the test sets of this experiment than the images from the noisy Flickr dataset. Finally, the positive test set of each topic is the 256 top-ranked Google images of the topic, and the negative test set is Google images that are randomly selected from the other topics. Note that in each run of experiment, only the positive training samples are different between *Temporal* and *Random* tests. As the binary classifier, we use the 128 nearest neighbor voting [Torralba et al., 2008], because it is one of the simplest classifiers and thus it can show the effects of training sets more directly. We repeat experiments ten times, and report the mean scores.

Fig.3.7 summarizes the comparison of classification performance between the *Temporal* and the *Random* training. Fig.3.7.(a) shows the classification rates for the selected 20 topics. The accuracies of *Temporal* training are higher by 8.05% than that of *Random* training on average. Fig.3.7.(b) presents the corresponding precision-recall curves, which show that the *temporal association* significantly improves the confidence of classification. The *Temporal* training is usually better than the *Random* training in performance, but the improvement is limited in some topics. In severely variant topics (*e.g.* *advertisement* and *starbucks*), the temporal consistency is hardly captured. In excessively stationary and coherent topics (*e.g.* *butterfly+insect* and *parrot*), the random sampling is also acceptable.

3.4 Summary

We present a nonparametric approach for modeling and analysis of the dynamic behaviors of Web image collections. A sequential Monte Carlo based tracker is proposed to capture the subtopic evolution in the form of the similarity network between images. In addition to the newly developed framework, the major empirical contributions and observations of this chapter are as follows.

- We perform the subtopic outbreak detection, which points out when the topical contents of image sets rapidly change.
- We show that the images can be a more reliable and delicate source of information to detect the topical evolution than tag texts.
- We show that training using the temporal association can improve image classification performance especially for extremely diverse Web images.

Chapter 4

Time-Sensitive Image Retrieval and Prediction

4.1 Introduction

As digital images are gaining popularity as a form of communicating information online, image search and retrieval has become an indispensable feature in our daily Web uses. Most commercial Web image search engines such as Bing, Google, and Yahoo largely rely on the text-based approach [Cui et al., 2008], in which given a query keyword, relevant pictures are retrieved and ranked by matching textual information of images such as surrounding texts, titles, or captions. Although the text-based image search has been successful as an effective and scalable image retrieval approach, it suffers from ambiguous and noisy results due to the mismatch between images and their surrounding texts. Moreover, it is still limited to correctly exploit visual contents of images and identify implicit or explicit search intent of a user.

In this chapter, we study one additional aspect to improve image search quality: *temporal dynamics of image collections*. In other words, given Web image collections associated with keywords of interest, we aim at identifying their characteristic temporal patterns of occurrences on the Web, and leveraging them to improve search relevance at a query time. This problem is closely related to one recent emerging research in information retrieval: *exploring the temporal dynamics of Web queries to improve search relevance* [Dakka et al., 2008; Kulkarni et al., 2011; Metzler et al., 2009; Radinsky et al., 2012]. Many queries are time-sensitive; the popularity of a query and its most relevant documents change over time. For example, a statistical analysis of Web query logs in [Metzler et al., 2009] reported that more than 7% of queries have implicitly temporal intents (*e.g.* miss universe, Olympics). Moreover, many of them are connected to the events that have occurred with predictable periodicity. This new area of research has cast a variety of interesting research questions, for example, identifying search terms that are sensitive to time, and reranking documents according to the query time. However, much of previous work has targeted at the search of text documents such as blogs and news archives by analyzing the query log data; the time-sensitive Web image retrieval has yet received little attention.

With our experiments on more than seven millions of Flickr images, we have found three good reasons why the discovery of *temporal* patterns in Web image collections is beneficial to existing image retrieval systems. We present them with a query example of the *cardinal* in Fig.4.1. *First, knowing when search takes place is useful to infer users' implicit search intents*. Fig.4.1.(a) shows the top ten images retrieved by Google and Bing image search engines. Seemingly, they are reasonable because the *cardinal* usually refers to the red bird in America. However, the term

Figure 4.1: Overview of time-sensitive Web image ranking and retrieval with a query example of the *cardinal*. (a)-(b) Top ten images retrieved by Google/Bing and Flickr search engines at 7/31/2012. (c) The results of our time-sensitive image retrieval for two query time points in winter and summer. (d) The result of our personalized image retrieval for a designated time and user.

cardinal is polysemous; it is also the names of popular sports teams (*e.g.* the American football and the baseball team). Therefore, some of *cardinal* queries in summer and winter are likely to be associated with the baseball and football team, respectively, according to the scheduled seasons of the sports.

Second, the timing suitability can be used as a complementary attribute to relevance. Fig.4.1 illustrates two such cases: One is that, as shown in Fig.4.1.(a), due to explosion of images shared on the Web, there are redundantly relevant images to popular queries like the *cardinal*. Timing suitability would be a good complementary ranking attribute to improve diversity or break ties between almost equally relevant images. The other case is that, in Fig.4.1.(b), the actual user images in the photo-sharing site Flickr are extremely diverse, and thus it is still very challenging to rank those images in any meaningful order. As shown in Fig.4.1.(c), the query time information can help obtain a more focused search output, which may include the images about *a cardinal bird in snowy field* in winter, but the images of *baby cardinals or eggs* in summer.

Third, temporal information is synergetic in personalized image retrieval. If a query word has a broad range of concepts, its dominant usages vary much according to users. Our experiments show that once we can identify a user's preference, image retrieval can be further specific since the term usages of individual users are relatively stationary. For example, as shown in Fig.4.1.(d), if

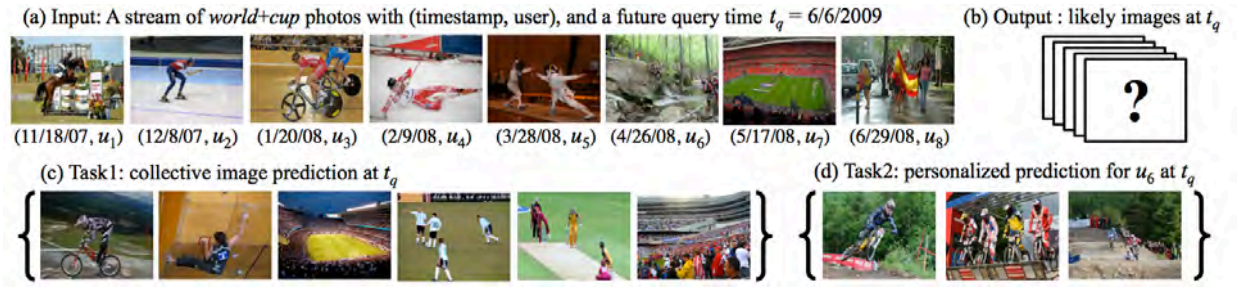


Figure 4.2: (a) Given an image sequence of *world+cup* up to 12/31/2008, can we guess what images are likely to occur at a future time point t_q=6/6/2009? (c) Collective image prediction. The *world+cup* usually refers to the soccer event, so a soccer scene can be a reasonable guess. However, the actual Web images are diverse because they reflect different users' experiences and preferences. (d) Personalized image prediction for user u_6. A user's unique angle of seeing the topic can make the prediction more focused.

a user took or searched *cardinal* pictures a lot for a basketball team last winter, he tends to do the same this winter as well.

Problem Statement: As an input, we gather a large-scale pool of Web images along with metadata (*e.g.* timestamps, owners) by querying Q topic keywords from a text-based image search engine. We use raw Flickr images since our time-sensitive image retrieval is more interesting for extremely diverse general users' photos rather than sufficiently cleaned-up Google or Bing images. In this work, our objective is two-fold: Our first goal is to automatically model the temporal properties of each topic keyword, because every topic is not necessarily time-sensitive, and has its own characteristic temporal behaviors. The second goal is to leverage the learned temporal models to rank the images in database according to temporal suitability when a topic keyword and a query time are given. We also address the *personalized* time-sensitive image ranking, which is customized image retrieval for a designated user.

Web image prediction: In real scenarios, the query time is usually *now* (*i.e.* the time when the search takes place), but we assume that it can be *any time even in future* for generality. However, if the query time is in future, we have to learn users' photo-taking patterns and extrapolate likely images for the future query time. We call such time-sensitive image retrieval with a future query time point as the *image prediction* task. For better understanding, Fig.4.2 shows the image prediction task with an example of the *world+cup* query. Suppose that we download the image database from Flickr for the *world+cup* keyword up to $12/31/2008$. The objective here is to estimate what would be the most likely pictures that are taken in a future query time, for example, $6/6/2009$, and retrieve images similar to them from the database. As Fig.4.2.(c) has shown, the pictures actually taken at $6/6/2009$ and shared on Flickr are not necessarily about the best possible world cup pictures (if the definition of *best* is even possible). Instead, they are the pictures that not only reflect the semantic meaning of the keyword, but also people's intends at that given moment of time. Furthermore, if a user cue is supplemented, the image prediction becomes highly personalized as shown in Fig.4.2.(d), given that individual users have their own preferences and photo-taking styles. Although the term *world+cup* usually refers to the international soccer event, it is also commonly used in other international sports and competitions (*e.g.* ski, skate, bicycle, or

horse riding, as shown in Fig.4.2.(a)), which are usually held periodically. With the majority of Web photos now coming from hundreds of millions of general users with different experiences and preferences, the contents of images that are associated even with the same keyword can be highly variable according to who took the pictures when.

Proposed method: Our objectives are accomplished by a unified statistical model: regularized multi-task regression on multivariate point process. We view an observed image stream as an instance of multivariate point process, which is a stochastic process that consists of a series of random events occurring at points in time and space [Daley and Vere-Jones, 2003]. Then, we automatically test what temporal models or their combinations are the best to describe the image occurrence behaviors, and formulate a regression problem to learn the historical relations between image occurrence probabilities and various temporal factors or covariates that influence them (*e.g.* seasons, dates, and other external events). From the learned models, we can easily compute the ranking scores of images for any given time point. For a more accurate ranking, We explore the idea of multi-task learning to incorporate multiple types of image representation. Consequently, our algorithm offers several important advantages for large-scale image retrieval as follows: (i) *Flexibility*: The image occurrence on the Web is correlated with a wide range of factors or covariates (*e.g.* season, time, user preference, and other external events). We can easily build a set of parametric models to capture any number of possible temporal behaviors of image collections, and automatically choose the most statistically suitable ones (Section 4.4.1). (ii) *Optimality*: We can achieve a globally optimal or approximate solution to the learning of temporal models (Section 4.4.4). (iii) *Scalability*: The learning is performed offline once, and the online query step is very fast. Both processes run in a linear time with most parameters such as time steps and the number of image descriptors (Section 4.5.3). (iv) *Retrieval accuracy*: We perform experiments on more than seven millions of Flickr images over a wide range of 30 topic keywords. We demonstrate that our image retrieval algorithm outperforms other candidate methods including Ranking SVM [Joachims, 2002], a PageRank-based image retrieval [Jing and Baluja, 2008; Kim et al., 2010] and a generative author-time topic model [Rosen-Zvi et al., 2004] (Section 4.6).

4.2 Problem Formulation

We assume that each of input Flickr images is assigned to topic keywords, timestamp, and owner ID. In addition to such meta-data from Flickr, we extract two types of information modalities: image description and user description.

4.2.1 Image Description

In this work, we extract four different image descriptors because no single descriptor can completely capture various contents of an image, and thus leveraging multiple descriptors is a widely accepted common practice in recent computer vision research. The four descriptors that are explained below can be classified into two low-level (SIFT and HOG) and two high-level descriptors (Tiny and Scene), all of which are extracted by using publicly available codes[1].

[1] We use following codes: (SIFT) at http://www.vlfeat.org, (HOG) at http://www.cs.brown.edu/∼pff/latent, (Tiny) and (Scene) at http://people.csail.mit.edu/jxiao/SUN/.

Color SIFT (SIFT): We densely extract HSV color SIFT on a regular grid at steps of 4 pixels. We form 300 visual words by applying K-means to randomly selected SIFT descriptors. The nearest word is assigned to every SIFT, and binned using a three-level spatial pyramid.

HOG2x2 (HOG): We also use the histogram of oriented edge (HOG) feature, inspired by its recent success in object detection research [Felzenszwalb et al., 2010]. We extract HOG descriptors on a regular grid at steps of 8 pixels by following the method called HOG2x2 in [Xiao et al., 2010].

Tiny Image: Inspired by [Torralba et al., 2008], we resize each image to a 32×32 tiny color image, and use RGB pixel values as features. This approach not only reduces image dimensionality to be computationally feasible, but also is discriminative enough to convey high-level statistics of the image.

Scene description: Since a large portion of Web images contain scenes, the scene classifier outputs can be a meaningful high-level description of an image. SUN database [Xiao et al., 2010] is an extensive dataset of 397 scene categories. As a scene descriptor, we compute the scores of linear one-vs-all SVM classifiers for 397 scene categories using Hog2x2 features, by following the classification benchmark protocol in [Xiao et al., 2010].

Visual clusters: Since all the above descriptors except (Scene) are high-dimensional (*e.g.* 6,300 of (SIFT)), they are down-sampled further by the soft-assignment idea. For each descriptor type k, we construct $L_k(=300)$ *visual clusters* by applying K-means to randomly sampled image descriptors. Then, an image I is assigned to r-nearest visual clusters for each descriptor type with the weights of an exponential function $\exp(-d^2/2\sigma^2)$, where d is the distance between the descriptor and the visual cluster and σ is a spatial scale. Consequently, an image I is described by four L_1 normalized vectors with only r nonzero weights, which are denoted by $\{\boldsymbol{h}_k(I)\}_{k=1}^4$ with dimensions of $[L_k]_{k=1}^4 = [300\ 300\ 300\ 374]$. We also let $L = \sum L_k = 1274$.

4.2.2 User Description

Clustering users and measuring similarity between users are important for personalization in collaborative filtering [Das et al., 2007]. Its basic assumption is that similar users are likely to share common photo taking and search behaviors. For clustering users, we use the pLSA (Probabilistic latent semantic analysis) clustering as proposed in Google News personalization [Das et al., 2007]. We first choose a fixed number of top users who have uploaded images most, and compute an L-dimensional histogram for each user where each bin represents the count of images belonging to the corresponding visual cluster. In pLSA, the distribution of visual cluster v in user u_i's images, $p(v|u_i)$, is given by the following generative model:

$$p(v|u_i) = \sum_{z \in \mathcal{Z}} p(v|z)p(z|u_i). \tag{4.1}$$

The latent variable $z \in \mathcal{Z}$ represents the cluster of user propensity. Thus, $p(z|u_i)$ is proportional to the fractional membership of user i to cluster z. We use $p(\mathbf{z}|u_i)$ as the descriptor of user u_i. The user clustering can be done by grouping users with the same $z^* = \mathrm{argmax}_z\, p(\mathbf{z}|u_i)$ or run K-means on the user descriptors $p(\mathbf{z}|u_i)$. The user similarity is calculated by histogram intersection on the user descriptors.

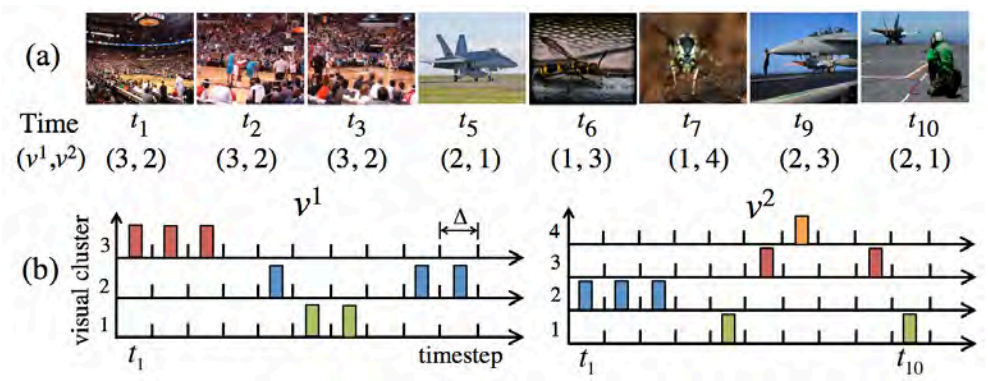


Figure 4.3: A multivariate point process for a short image stream of the *hornet*. (a) Each image is assigned to a timestamp and visual clusters of two different descriptors ($K = 2$, $L_1 = 3$, $L_2 = 4$). (b) The image stream is modeled by two multivariate discrete-time point processes.

4.3 Multivariate Point Processes

In this section, we discuss the mathematical background of multivariate point process for modeling Web photo streams. Fig.4.3 shows a toy example for a short image stream of the *hornet*. Suppose that we extract K image descriptors from each image, and for each descriptor, we cluster the images into L_k visual clusters (In this example, $K = 2, L_1 = 3, L_2 = 4$). Intuitively, one can easily construct K multivariate point processes as shown in Fig.4.3.(b). For simplicity, we first assume that the occurrence of each visual cluster is independently modeled. Hence, the point process of Fig.4.3 can be regarded as a single multivariate point process with $L = L_1 + L_2 = 7$. In section 4.4.3, we will consider an extended multi-task framework with considering correlations between different descriptors.

Intensity functions: Since the intensity function can completely define a point process [Daley and Vere-Jones, 2003], we first introduce its definition. Formally, a multivariate point process can be described by a counting process $\boldsymbol{N}(t) = (N^1(t), \cdots, N^L(t))^T$ where $N^l(t)$ is the total number of observed images assigned to visual cluster l in the interval $(0, t]$. Then, $N^l(t + \Delta) - N^l(t)$ represents the number of images in a small interval Δ. By letting $\Delta \to 0$, we obtain the *intensity function* at t, which is the infinitesimal expected occurrence rate of visual cluster l at time t [Daley and Vere-Jones, 2003]:

$$\lambda^l(t) = \lim_{\Delta \to 0} \frac{P[N^l(t+\Delta) - N^l(t) = 1]}{\Delta}, \; l \in \{1, \ldots, L\}. \tag{4.2}$$

Generalized Linear Model: We assume that the intensity function $\lambda^l(t)$ is represented by the covariates that influence the occurrence of visual cluster l. We define the parametric form of $\lambda^l(t_i|\boldsymbol{\theta}^l)$ as the exponential of a linear summation of the functions f_j^l of the covariates x_j with a parameter vector $\boldsymbol{\theta}^l = (\theta_1^l, \cdots, \theta_J^l)$:

$$\log \lambda^l(t_i|\boldsymbol{\theta}^l) = \sum_{j=1}^{J} \theta_j^l f_j^l(x_j), \;\; l \in \{1, \ldots, L\}. \tag{4.3}$$

Data likelihood: Suppose that we partition the interval $(0, T]$ by a sufficiently large number M (*i.e.* $\Delta = T/M$) so that in each time bin Δ only one or zero image occurs. Then, we can denote the sequence of images up to T by $N^l_{1:M} = n^l_1 \cdots n^l_M$ with $n^l_i \in \{0, 1\}$. It is shown in [Truccolo et al., 2005] that the likelihood of such a point process along with λ^l of Eq.(4.3) is identical to that of the *Poisson regression*. Therefore, the log-likelihood of an observed image sequence is

$$\ell(N^l_{1:M}|\boldsymbol{\theta}^l) = \sum_{i=1}^{M} \big(n_i \lambda^l(t_i|\boldsymbol{\theta}^l) - \exp(\lambda^l(t_i|\boldsymbol{\theta}^l)) - \log n_i!\big). \tag{4.4}$$

L_1 **regularized likelihood**: Although numerous factors or covariates can be plugged in Eq.(4.3), each visual cluster is likely to depend on only a small subset of them. Hence, it is important to detect a few strong covariates by encouraging a sparse estimator of $\boldsymbol{\theta}^l$ for each visual cluster l. This approach is also practical because we usually do not know what factors are important beforehand; we safely include as many candidate factors as possible, and then choose only a few covariates for each visual cluster via MLE learning. Therefore, we introduce *Lasso* penalty [Tibshirani, 1996] into the likelihood of Eq.(4.4) with a regularization parameter μ controlling sparsity level:

$$\ell_L(N^l_{1:M}|\boldsymbol{\theta}^l) = \ell(N^l_{1:M}|\boldsymbol{\theta}^l) - \mu \sum_{j=1}^{J} |\theta^l_i|. \tag{4.5}$$

4.4 Temporal Modeling of Photo Streams

Our first objective is to identify the temporal properties of a given image stream. This goal is achieved via the learning of temporal models as follows. We first represent the image stream with a multivariate point process $\{N^l_{1:M}\}^L_{l=1}$ as described in previous section. Then, we define multiple models for $\lambda^l(t_i|\boldsymbol{\theta}^l)$ by enumerating all possible temporal factors that influence the image occurrences (section 4.4.1). Finally, for each occurrence data $N^l_{1:M}$ of visual cluster l, we select a subset of most statistically plausible models (section 4.4.2), and learn the parameters $\boldsymbol{\theta}^{l*}$ of the models to discover which factors are actually contributing (section 4.4.4). Note that the whole processes above can be automatically performed.

4.4.1 Models of Temporal Behaviors

In this section, we enumerate a set of models for the intensity functions, each of which is designed to capture a particular temporal property. Thanks to the flexibility of our framework, one can freely add or remove such models according to the characteristics of image topics unless they contradict the definition of Eq.(4.3). In this work, we construct two groups of models: *temporal attributes* and traditional *time series*.

Temporal attributes: Humans' time perception and photo taking and search behaviors are not only continuous on time but also driven by temporal attributes. For example, *zoo* photos may be more frequently taken in weekend rather than in weekdays, or *ski* images appear more often in

January than in June. Therefore, we build a set of intensity function models for temporal attribute-driven covariates as follows.

$$\log \lambda_y^l(t_i|\boldsymbol{\alpha}^l) = \alpha_0^l + \sum_{y=Y_s}^{Y_t} \alpha_y^l I_y(t_i) \tag{4.6}$$

$$\log \lambda_m^l(t_i|\boldsymbol{\beta}^l) = \beta_0^l + \sum_{t=1}^{12} \beta_t^l g(t_i - t) \tag{4.7}$$

$$\log \lambda_d^l(t_i|\boldsymbol{\gamma}^l) = \gamma_0^l + \sum_{t=1}^{12} I_t(t_i) \sum_{d=1}^{31} \gamma_{i,d}^l I_d(t_i) \tag{4.8}$$

$$\log \lambda_w^l(t_i|\boldsymbol{\zeta}^l) = \zeta_0^l + \sum_{w\in\{M,\ldots,S\}} \zeta_w^l I_w(t_i) \tag{4.9}$$

$$\log \lambda_h^l(t_i|\boldsymbol{\eta}^l) = \eta_0^l + \sum_{h\in\mathcal{H}} \eta_h^l I_h(t_i). \tag{4.10}$$

In equations, λ_y^l, λ_m^l, λ_d^l, λ_w^l, and λ_h^l are the models of intensity functions for years, months, days, weekdays (from Monday to Sunday), and holidays[2], whose lists are denoted by $\mathcal{H}$. The parameter set to be learned comprises $\{\boldsymbol{\alpha}^l, \boldsymbol{\beta}^l, \boldsymbol{\gamma}^l, \boldsymbol{\zeta}^l, \boldsymbol{\eta}^l\}$. $I_y(t_i)$ is an indicator function that is 1 if the year of t_i is y, and 0 otherwise (*e.g.* $I_y(t_i) = 1$ if $y = 2008$ and t_i= 6/3/2008). Similarly, $I_w(t_i)$ and $I_h(t_i)$ are indicators for week and holidays. For month covariates, we use Gaussian weighting $g(t_i - t) \propto \exp(-(t_i - t)^2/\sigma)$, which leads that if an image occurs in May, for example, some contributions are also given to nearby months like April and June, assuming that images smoothly change on the timeline.

Here, our models are mainly built based on calendric temporal attributes, but the models driven by other textual or social factors (*e.g.* news articles) can be supplemented.

An example: Fig.4.4 is a toy example of the *shark* topic to intuitively show how the intensity function models are used for fitting observed image streams. This example illustrates the intensity function models for years (λ_y^l of Eq.(4.6)) and months (λ_m^l of Eq.(4.7)), where the parameter set comprises seven $[\alpha_y^l]_{y=2003}^{2009}$ and twelve $[\beta_m^l]_{m=1}^{12}$. Fig.4.4.(a) shows four sampled images from three visual clusters, each of which approximately corresponds to *sea tour*, *ice hockey*, *diving in aquarium*. Fig.4.4.(b) presents their actual occurrence sequences. Fig.4.4.(c)-(d) show the learned intensity functions λ_y^l and λ_m^l. Most of intensity functions for years roughly increase every year because the number of uploaded photos in Flickr grows yearly. The rates decrease in 2009 because the *shark* dataset is gathered up to mid 2009. Interestingly, the visual clusters show different monthly behaviors in Fig.4.4.(d). λ_m^1 has a higher intensity value (*i.e.* more frequently occurred) in June, λ_m^2 peaks around January, and λ_m^3 is stationary all year long. This result is reasonable because sea tours are popular in summer, the ice hockey season takes place during winter, and visiting aquarium is favored regardless of season. The learned intensity functions can be used for a simple time-sensitive image retrieval. For example, if the month of the query time t_q is January, then $\lambda_m^2(t_q) \gg \lambda_m^3(t_q) > \lambda_m^1(t_q)$. Hence, we can rank the images of v^2 (*i.e.* the *ice hockey*) as the highest.

[2]We use the lists at http://vpcalendar.net/Holiday_Dates/.

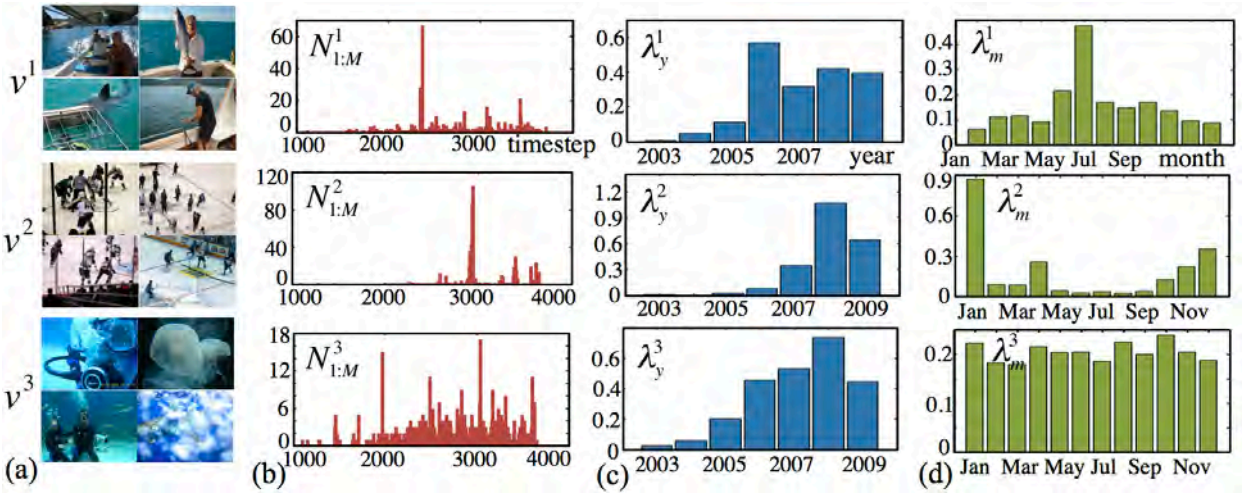


Figure 4.4: Examples of intensity function models of years and months for three visual clusters (VC) of the *Shark*: v^1(*sea tour*), v^2(*ice hockey*), v^3(*diving in aquarium*). (a) Four images sampled from each VC. (b) Observed image occurrences. (c)-(d) Estimated intensity functions for years and months. λ^1_m and λ^2_m have different image occurrence rates peaked in summer and winter, respectively. λ^3_m is stationary along the timeline.

Autoregression: The other group of temporal models is based on autoregression, which is one of most popular models for the analysis of time series. We present an example in Fig.4.5 for better understanding. We assume that the occurrence of each visual cluster is affected by its own history in Eq.(4.11), and the history of other visual clusters in Eq.(4.12). The first history model is represented by a linear autoregressive process:

$$\log \lambda^l_a(t_i|\boldsymbol{\phi}^l) = \phi^l_0 + \sum_{p=1}^{P_d} \phi^l_{dp} \Delta N^l_{i-dp} + \sum_{p=1}^{P_w} \phi^l_{wp} \Delta N^l_{i-wp} + \sum_{p=1}^{P_m} \phi^l_{mp} \Delta N^l_{i-mp} \tag{4.11}$$

where ΔN^l_{i-dp} denotes the occurrence counts of visual cluster l during $[t_i - dp, t_i)$, and d is the time window width. In Eq.(4.11), we use three different time windows: $d = 1$ day, and w =1 week, and $m = 1$ month. That is, λ^l_a is modeled by three different time-scaled (daily, weekly, and monthly) regressors whose orders are P_d, P_w, and P_m, respectively. The history model can capture the dynamic behavior of a visual cluster. As shown in Fig.4.5.(c), the learned parameters of v^1 (top) and v^2 (middle) show the typical patterns for yearly periodic behaviors, whereas the parameters of v^3 (bottom) are biphasic, which indicates a bursty occurrence.

The second correlation model represents the influence from the history of other visual clusters. Its mathematical form is almost identical to that of Eq.(4.11):

$$\log \lambda^l_c(t_i|\boldsymbol{\psi}^l) = \psi^l_0 + \sum_{c=1, c\neq l}^{L} \left(\sum_{q=1}^{Q_d} \psi^{lc}_{dq} \Delta N^l_{i-dq} + \sum_{q=1}^{Q_w} \psi^{lc}_{wq} \Delta N^l_{i-wq} + \sum_{q=1}^{Q_m} \psi^{lc}_{mq} \Delta N^l_{i-mq} \right). \tag{4.12}$$

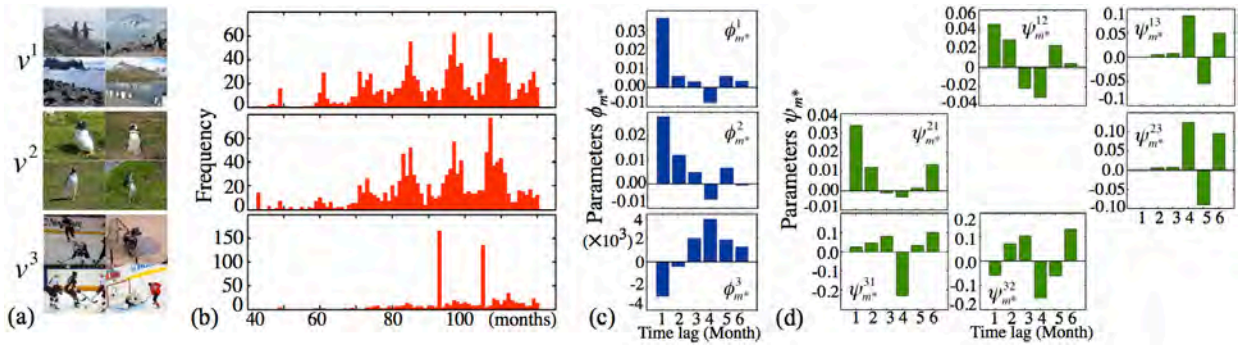


Figure 4.5: Examples of the *Penguin* topic for the learned parameters of history and correlation components. Visual clusters are {*penguins in landscape*, *penguins on grass*, *ice hockey team*}. (a) Four images sampled from each visual cluster. (b) Observed occurrence data. For simplicity, we only consider the granularity of month. The v^1 and v^2 are strongly synchronized and periodically peaked in summer, whereas the v^3 has two high peaks in winter. (c)-(d) Learned parameters of history and correlation components, respectively.

The parameter set consists of $(L-1) \times (Q_d + Q_w + Q_m) + 1$ number of ψ in the full model. This correlation component is quite useful for the actual prediction in the Flickr dataset; we observe that there are strong correlations between visual clusters, and thus the existence or absence of a particular visual cluster gives a strong clue for others' prediction. The learned parameters ψ_{m*} in Fig.4.5.(d) clearly capture the correlations observed in Fig.4.5.(b). For example, the subfigures of ψ^{12}_{m*} and ψ^{21}_{m*} in Fig.4.5.(d) show that the occurrence of v^1 and v^2 are highly synchronized, whereas the subfigures of ψ^{13}_{m*} and ψ^{23}_{m*} illustrate the occurrence of v^3 precedes those of v^1 and v^2 by about four months. For fast computation, instead of using the full pairwise model, we can learn the correlations with respect to some selected most frequent visual clusters.

4.4.2 Model Selection

In previous section, we introduce rather exhaustive seven temporal models from Eq.(4.6) to Eq.(4.12). However, the occurrence of each visual cluster does not necessarily depend on all the above models. For example, the occurrence of the *ice hockey* visual cluster v^2 of Fig.4.4 can be explained sufficiently well by the month intensity function model λ^l_m while other models may not be required any further. Therefore, we perform a model selection procedure, to choose a subset of temporal models by removing the ones with little or no predictive information. Mathematically, the parameter for visual word l can be defined by concatenating the parameters of seven temporal models $\boldsymbol{\theta}^l = [\boldsymbol{\alpha}^l, \boldsymbol{\beta}^l, \boldsymbol{\gamma}^l, \boldsymbol{\zeta}^l, \boldsymbol{\eta}^l, \boldsymbol{\phi}^l, \boldsymbol{\psi}^l]$, most of which will be zeros.

Algorithm 2 summarizes the overall procedure of our model selection. It is based on the well-known *greedy forward selection* scheme, in which we keep increasing models one by one by adding at each step the one that increases the goodness-of-fit score the most, until any further addition does not increase the score. As the goodness-of-fit test, we use Kolmogorov-Smirnov (KS) test using time-rescaling theorem [Brown et al., 2001], which is one of most popular approaches for statistical model assessment in point process literature. The KS statistic is a quantitative measure for the agreement between a learned intensity function and actual image occurrence data. This

Algorithm 2: Model selection for each visual cluster

Input: (a) A set of intensity function models in Eq.(4.6)– (4.12): $\Lambda^l = \{\lambda_y^l, \lambda_m^l, \lambda_d^l, \lambda_w^l, \lambda_h^l, \lambda_a^l, \lambda_c^l\}$.
(b) $N_{1:M}^l$: Occurrence data of visual cluster l.

Output: The best intensity function model λ^{l*} with learned parameter set $\boldsymbol{\theta}^{l*}$.

1: Define $\boldsymbol{\theta}_i \leftarrow$ param_est $(\lambda_i, N_{1:M}^l)$ to be the function that computes the MLE solution of parameters for a given λ_i and $N_{1:M}^l$. This will be discussed in section 4.4.4.
2: For λ_a^l and λ_c^l, decide the AR orders using AIC measure: $AIC(P) = -2\log \ell(N_{1:M}^l|\boldsymbol{\theta}_t) + 2P$ where P is the total number of parameters.
foreach $\lambda_i \in \Lambda^l$. **do**
  3: Compute $\boldsymbol{\theta}_i \leftarrow$ param_est $(\lambda_i, N_{1:M}^l)$.
  4: Compute KS static d_i by applying time rescaling theorem to the learned $\lambda_i(\boldsymbol{\theta}_i)$ and $N_{1:M}^l$.
5: Sort λ_i in an increasing order. Let o to be this order. Initialize $\lambda^{l*} = \text{argmin}_{\lambda_i \in \Lambda^l} d_i$ and $d^{l*} = \min d_i$.
repeat
  foreach $\lambda_i \in \Lambda^l$ *and* $\lambda_i \notin \lambda^{l*}$ *in the order of* o. **do**
    6: Set $\lambda_t = \lambda_i^{l*} \cdot \lambda_i$. $\boldsymbol{\theta}_t \leftarrow$ param_est $(\lambda_t, N_{1:M}^l)$.
    7: Compute KS static d_t as done in step **4**.
    if $d_t < d^{l*}$ **then** $\lambda^{l*} \leftarrow \lambda_t$, $d^{l*} = d_t$, and $\boldsymbol{\theta}^{l*} = \boldsymbol{\theta}_t$.
until λ^{l*} *is not updated*;

value is a distance metric, and thus a smaller value indicates a better model. In step 2 of Algorithm 2, the orders of the autoregressive models, λ_a^l of Eq.(4.11) and λ_c^l of Eq.(4.12), are decided by Akaike's information criterion (AIC). We choose the order parameters that lead to the smallest AIC, implying that the approximate distance between the model and the true process generating the data is the smallest. In practice, this step is important because temporal behaviors of visual clusters can operate at different time scales (*i.e.* monthly, weekly, or daily).

4.4.3 Regularized Multi-Task Regression

Until now, each visual cluster is independently modeled and learned without considering which description it is derived from. In order to fully take advantage of any arbitrary number of image descriptions, we introduce the idea of multi-task learning [Chen et al., 2011; Liu et al., 2009b], in which multiple related tasks are jointly learned by analyzing data from all of the tasks at the same time. This framework is powerful when the multiple tasks of interest are *different enough* to be specified by separate models, but are at the same time *similar enough* to be jointly learned.

We treat each descriptor as a *task*. Since each descriptor characterizes an image from a different perspective, it should be separately expressed. However, at the same time, it is likely that the descriptors from the same image share enough correlation that makes simultaneous learning beneficial. For example, suppose that a large portion of images of visual cluster 35 of HOG are also assigned to visual cluster 27 of Scene descriptors. It indicates that these two visual clusters are highly correlated, and thus are likely to share common covariates affecting their occurrences. Algorithm 3 discovers the set of frequently co-occurred visual cluster pairs as $\mathcal{E}$, where

Algorithm 3: Build the correlation set $\mathcal{E}$.

Input: (1) A set of images $\mathcal{I}$, each of which is assigned to the closest visual clusters of K descriptors.
Output: The correlation set $\mathcal{E}$.

1: Initialize $K(K-1)/2$ number of co-occurrence matrices $\mathcal{C}$, where $\boldsymbol{C}_{ab} \in \mathcal{C}$ is an $(L_a \times L_b)$ zero matrix between descriptor a and b.
foreach $I \in \mathcal{I}$. *Let visual cluster of* I *be* $(v_1, \ldots, v_K)$ **do**
  foreach $a, b \in \{1, \ldots, K\}$ *with* $a \neq b$ **do**
    2: $\boldsymbol{C}_{ab}(v_a, v_b) \leftarrow \boldsymbol{C}_{ab}(v_a, v_b) + 1$.
foreach $a, b \in \{1, \ldots, K\}$ *with* $a \neq b$ **do**
  3: $C_{ab} = \mathsf{row\_normalize}(C_{ab}) + \mathsf{column\_normalize}(C_{ab})$.
3: Select top R highest edges (v_a, v_b) from $\mathcal{C}$. The weight of a pair is $r_{ab} \propto \boldsymbol{C}_{ab}(a, b)/|\mathcal{I}|$. Set $\mathcal{E} \leftarrow (v_a, v_b, r_{ab})$.

$e = (v_a, v_b, r_{ab}) \in \mathcal{E}$ consists of three tuples: a pair of visual clusters v_a and v_b with correlation weight $r_{ab} > 0$. We can model this dependency structure across multiple tasks (*e.g.* the correlations between the visual clusters of different image descriptors) by introducing regularization term $\Omega(\boldsymbol{\Theta}_E)$ to the log-likelihood:

$$\mathcal{L} = \sum_{l \in \mathcal{E}} \ell(N^l_{1:M} | \boldsymbol{\theta}^l_k) - \Omega(\boldsymbol{\Theta}_E) \tag{4.13}$$

$$\Omega(\boldsymbol{\Theta}_E) = \mu \sum_{l \in \mathcal{E}} \|\boldsymbol{\theta}^l_k\|_1 + \nu \sum_{(a,b) \in \mathcal{E}} r_{ab} \sum_{j=1}^{J} |\theta^a_j - \theta^b_j|. \tag{4.14}$$

The regularization term $\Omega(\boldsymbol{\Theta}_E)$ consists of two different types of penalties, which are the *Lasso* penalty [Tibshirani, 1996] and *graph-guided fusion* penalty [Chen et al., 2011]. μ and ν are regularization parameters that control sparsity and fusion levels. The overall effect of *graph-guided fusion* penalty is that each subgraph of visual clusters in $\mathcal{E}$ tends to share common relevant covariates, and the degree of commonality is proportional to the correlation strength r_{ab}.

4.4.4 Optimization for Parameter Learning

The goal of parameter learning is to obtain the MLE solution $\boldsymbol{\theta}^{l*}$ that maximizes the likelihood with respect to an intensity function model λ^l and an observed image sequence $N^l_{1:M}$ for all $l = 1, \ldots, L$. Alternatively, if we explicitly represent the descriptor k as subscript, the set of parameters is denoted by $\boldsymbol{\Theta}^* = \{\boldsymbol{\Theta}^*_1, \ldots, \boldsymbol{\Theta}^*_K\}$ where $\boldsymbol{\Theta}^*_k = \{\boldsymbol{\theta}^{1*}_k, \cdots, \boldsymbol{\theta}^{L_k*}_k\}$ is the set of learned parameters for all visual clusters of descriptor k. We have introduced three likelihoods with different regularizations, which are optimized differently. First, the likelihood of Eq.(4.4) with no regularization term reduces to that of Poisson regression, and the globally-optimal solution can be attained by an iteratively reweighted least square algorithm [Daley and Vere-Jones, 2003]. Second, for the likelihood of Eq.(4.5) with the Lasso penalty, the globally-optimal MLE solution can be achieved by using the cyclical coordinate descent in [Friedman et al., 2010]. Finally, for the likelihood of Eq.(4.13) with the graph-guided fusion penalty, we obtain an approximate MLE solution by modifying the

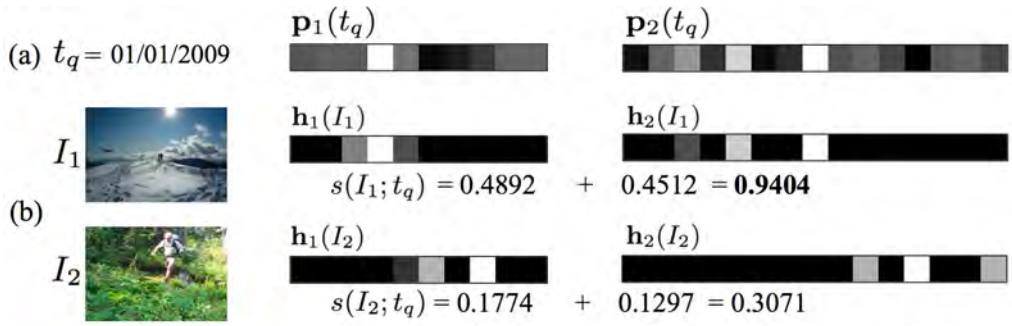


Figure 4.6: A toy example of computing ranking scores of two *mountain+camping* images I_1 and I_2 for $t_q = (01/01/2009)$ with ($K = 2$, $L_1 = 10$, $L_2 = 15$). (a) Two membership vectors $\boldsymbol{p}_1(t_q)$ and $\boldsymbol{p}_2(t_q)$ are computed from the learned intensity functions. (b) Two descriptor vectors $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are extracted from each image, and the ranking scores $s(I_1;t_q)$ and $s(I_2;t_q)$ are computed by Eq.(4.15). I_1 has a higher ranking value (0.9404) than I_2 (0.3071) for the t_q.

Proximal-gradient method [Chen et al., 2011], which is a scalable first-order method (*i.e.* using only gradient) with a fast convergence rate. We extend this method that was originally developed for linear regressions to be applicable to the regularized Poisson regressions.

4.5 Time-Sensitive Image Retrieval

In this section, we discuss our second goal, which is to perform image ranking using the learned temporal models.

4.5.1 Predictive Ranking

Computing intensity functions: As a result of optimization, we have the learned parameters of all visual clusters of all K descriptors: $\boldsymbol{\Theta}^* = \{\boldsymbol{\Theta}_1^*, \ldots, \boldsymbol{\Theta}_K^*\}$.

In the retrieval step, given a query time t_q, we first obtain $\boldsymbol{\Lambda}(t_q|\boldsymbol{\Theta}^*) = \{\boldsymbol{\Lambda}_1(t_q|\boldsymbol{\Theta}_1^*), \ldots, \boldsymbol{\Lambda}_K(t_q|\boldsymbol{\Theta}_K^*)\}$, which is the set of intensity functions of all visual clusters of all K descriptors for t_q. ($|\boldsymbol{\Lambda}(t_q|\boldsymbol{\Theta}^*)| = \sum_{k=1}^{K} L_k$). Each $\lambda_k^l(t_q|\boldsymbol{\theta}_k^{l*}) \in \boldsymbol{\Lambda}_k(t_q|\boldsymbol{\Theta}_k^*)$ is computed by gathering covariate values for t_q, and plugging them along with learned $\boldsymbol{\theta}_k^{l*}$ into Eq.(4.3). Here, let us remind that $\lambda_k^l(t_q|\boldsymbol{\theta}_k^{l*}) \propto P(N_k^l(t_q + \Delta) - N_k^l(t_q)|N_{1:M}^l)$[3]. That is, the intensity function of a visual cluster at t_q is proportional to its occurrence probability at t_q. Therefore, for each $k \in K$, we can define a membership vector: $\boldsymbol{p}_k(t_q) = \boldsymbol{\Lambda}_k(t_q|\boldsymbol{\Theta}_k^*)/\|\boldsymbol{\Lambda}_k(t_q|\boldsymbol{\Theta}_k^*)\|_1 (\in \mathbb{R}^{L_k \times 1})$, where each $p^l \in \boldsymbol{p}_k(t_q)$ is the membership probability that an image occurred at t_q belongs to visual cluster l of descriptor k.

Ranking: The next step is to compute the ranking score of any given image I for t_q. We use the idea of continuous error-correcting output codes (ECOC) [Crammer and Singer, 2002]. We first extract K image descriptors $\{\boldsymbol{h}_k(I)\}_{k=1}^K$ by the feature extraction methods in section 4.2.1.

[3] It can be easily shown by that λ_k^l is an infinitesimal expected occurrence rate at t_q and a series of images is modeled by a sequence of conditionally independent Bernoulli trials during the derivation of the likelihood function of Eq.(4.4).

Then, the ranking score of image I at t_q is defined by the histogram intersection[4]

$$s(I; t_q) = \sum_{k=1}^{K} \| \min(\boldsymbol{h}_k(I), \boldsymbol{p}_k(t_q)) \|_1. \tag{4.15}$$

Fig.4.6 illustrates a toy example of computing ranking scores for two images of the *mountain+camping* with $K = 2, L_1 = 10, L_2 = 15$. Fig.4.6.(a) shows two membership vectors $\boldsymbol{p}_k(t_q)$ that are computed from the learned intensity functions for t_q=(01/01/2009), and Fig.4.6.(b) illustrates four descriptor vectors for I_1 and I_2. The $\boldsymbol{p}_k(t_q)$ are more similar to the descriptors $\boldsymbol{h}_k(I_1)$ of image I_1 (*snowy mountain*) than $\boldsymbol{h}_k(I_2)$ of I_2 (*tracking in woods*), and thus image I_1 is ranked higher.

The computation of our ranking score is very fast; the histogram intersection requires only element-wise min operations between K vector pairs. It is also easy to organize the descriptor vectors of images in the database by using any data structure such as trees or hashes for fast retrieval.

4.5.2 Personalization

The key idea of personalization is, given a query user u_q, to assign more weights to the pictures taken by u_q and similar users to u_q during learning. In a normal setting, one image occurrence is equally counted by one for $N^l_{1:M}$ (See an example in Fig.4.4.(b)). However, for personalization, the images by u_q and the users in the same user cluster with u_q are weighted by larger values so that model fitting is more biased to their images. We implement the personalization by using the locally weighted learning framework [Atkeson et al., 1997], which is a form of lazy learning for a regression to adjust the weighting of data samples according to a query.

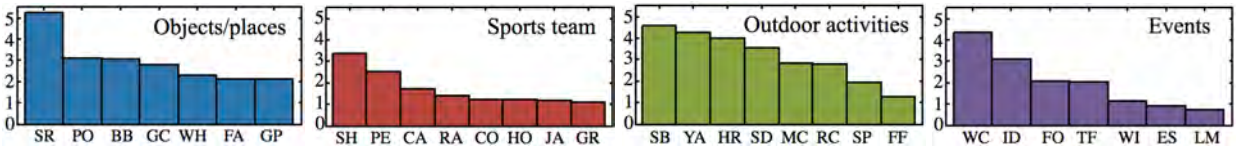
In order for personalization to be done offline, we exploit this idea at the user cluster level. Suppose that there are Z user clusters as a result of pLSA based user clustering in section 4.2.2. We then compute $Z \times Z$ pairwise user similarity matrix $\mathbf{U}$ by $\mathbf{U}(x, y) = \sqrt{\exp(-(\boldsymbol{u}_x - \boldsymbol{u}_y)^2/\sigma)}$, where $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ are the user descriptors of cluster centers of U_x and U_y, respectively[5]. We separately learn the personalized model for each user cluster U_z, in which the weights of image occurrences are adjusted by $\mathbf{U}(z, *)$ (*i.e.* the z-th row of $\mathbf{U}$). That is, if the owner of an image I is in user cluster U_x, then the occurrence of image I is reweighted by $\mathbf{U}(z, x)$. At the query stage, given a query user u_q, we identify the user cluster to which u_q belongs, and then use the pre-computed learned model of that cluster.

4.5.3 Computation time

Learning: The learning step performs offline only once. The learning time without the graph-guided fusion penalty is $O(L|T|J)$ while that with the fusion penalty is $O(L|T|J^2)$ where $|T|$ is the number of time steps (*e.g.* discretized by day), J is the number of covariates, and $L = \sum_{k=1}^{K} L_k$. For a linear model, our Matlab implementation takes about less than one hour to learn the model for the 553K of *world+cup* images with $L = 1,050$, $|T| = 2,000$, and $J = 32$.

[4] In the ECOC terminology, Eq.(4.15) means that the histogram intersection is chosen as the decoding metric.

[5] We use the Gaussian kernel function for user weighting.



SR(spider), PO(potato), BB(blackberry), GC(grandcanyon), WH(white +house), FA(fine+art), GP(grape), SH(shark), PE(penguin), CA(cardinal), RA(raptor), CO(coyote), HO(hornet), JA(jaguar), GR(grizzly), SB(snowboarding), YA(yacht), HR(horse+riding), SD(scuba+diving), MC(mountain+camping), RC(rock+climbing), SP(safari+park), FF(fly+fishing), WC(world+cup), ID(independence+day), FO(formula+one), TF(tour+de+france), WI(wimbledon), ES(easter+sunday), LM(london+marathon)

Figure 4.7: 30 topics of our Flickr dataset. The topic words are classified into four categories. The total numbers of images and users are (7,592,426, 1,434,749). The Y-axis is the number of images ($\times 10^5$).

Querying: At the online querying stage, computing the intensity functions for a given query time t_q (and optionally a user u_q) runs in $O(LJ)$, and calculating the ranking scores of N images takes $O(LN)$. The overall querying step takes less than 0.5 second with $N = 1$K in the same experiment. Querying is fast enough to run online, but it can be also pre-computed offline, for example, processing queries for next one year (365 days) can be done within a couple of hours.

4.6 Experiments

We evaluate the performance of our time-sensitive image retrieval algorithm using Flickr datasets.

4.6.1 Evaluation Setting

Datasets: Fig.4.7 summarizes our dataset that consists of more than seven million images of 30 topics from Flickr. We download all images that are retrieved by topic names as search keywords from Flickr without any filtering. The *date_taken* field of each image provided by Flickr is used for the timestamp.

Tasks: We first divide each image set into training and test set by time $T_T = T_e - (1$ year) where T_e is the end time point of the dataset. That is, for each topic, the test set consists of the images in the last one year of the database, and the training set $\mathcal{I}_B$ comprises the other images, which are used to learn the image occurrence patterns.

Our tasks for experiments are similar to those of other image ranking and retrieval papers [Cui et al., 2008; Liu et al., 2011; Wang et al., 2011] except that time suitability of retrieved images is the key performance index to be evaluated. We perform the image retrieval task as follows; a topic name and a query time point $t_q > T_T$ are given. That is, t_q is a *future* time point with respect to training data since T_T is the time threshold that separates the training set from the test set. The images that are actually taken in t_q are the positive test set $\mathcal{I}_P$. The negative test set $\mathcal{I}_N$ is gathered by randomly selecting the same number images outside of [$t_q \pm 3$ months] from the test set. The algorithm is supposed to rank the test images $\mathcal{I}_P \cup \mathcal{I}_N$ from which average precisions are computed. The personalized retrieval is the same except that a query user u_q is specified at the test. u_q is randomly chosen from a set of users who have at least 100 images in both training and test sets. For each topic, we randomly generate 36 t_q test cases (*i.e.* three random choices per month) for normal retrieval, and 20 (t_q, u_q) test pairs on average for personalized retrieval. That is, we examine more than $1,500$ test instances in total to evaluate the performance of our algorithm.

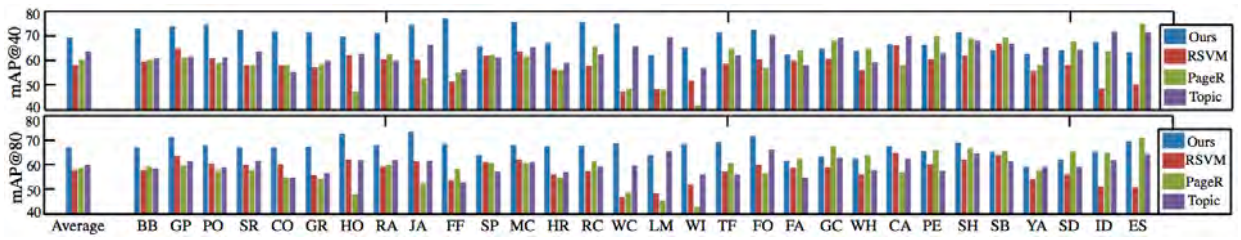


Figure 4.8: Quantitative comparison of image retrieval between our method and three baselines (RSVM, PageR, Topic) using mAP@40(top) and mAP@80(bottom) metrics. The average performances for mAP@(40,80) in the left-most bar set are ours: (69.1%,66.7%), RSVM: (58.0%,57.6%), PageR: (60.1%,58.6%), Topic: (63.5%,59.7%).

Baselines: The time-sensitive image retrieval is relatively new, and thus there are few existing methods to be compared. Hence, we select and adapt three baselines from popular image ranking methods for quantitative comparison with our algorithm. Below we summarizes the baselines, each of which is denoted by (RSVM) [Joachims, 2002], (PageR) [Jing and Baluja, 2008; Kim et al., 2010], and (Topic) [Rosen-Zvi et al., 2004]. In the personalized retrieval, the locally weighted learning is also applied to all the baselines.

- Ranking SVM(RSVM) [Joachims, 2002]: We obtain pseudo-relevant and pseudo-irrelevant training data by sampling images from the training set $\mathcal{I}_T$ based on their timestamps. The pseudo-relevant images are randomly sampled from Normal distributions whose mean are the same dates (m/d) of t_q in previous years. The pseudo-irrelevant images are randomly chosen from the images whose timestamps are outside [date(m/d) of $t_q \pm$ 3 months] at every year. Then, we learn the Ranking SVM using the code provided by the authors of [Joachims, 2002].

- PageRank-based model(PageR) [Hsu et al., 2007; Kim et al., 2010]: Given the same training data above, we build a similarity graph between training and test data by using HOG and SIFT features, and compute ranking scores using the *random walk with restart* [Tong et al., 2006] (*i.e.* a query-specific PageRank).

- Author-Time Topic Model(Topic) [Rosen-Zvi et al., 2004]: We modify the Author-Topic model [Rosen-Zvi et al., 2004] to jointly model users, months, and visual clusters of images. Using the same training data above, we estimate the subtopic distribution of each month and the subtopic assignments of visual clusters, from which we compute the ranking scores of test images for t_q.

4.6.2 Quantitative Results

Fig.4.8 and Fig.4.9 show the quantitative comparison of normal and personalized image retrieval between our approach and three baselines, respectively. We report the mean average precision at top 40 and 80 ranked images, which are denoted by mAP@40 and mAP@80. In each figure, the leftmost bar set is the average performance of 30 topics, and the results of all 30 topics follow.

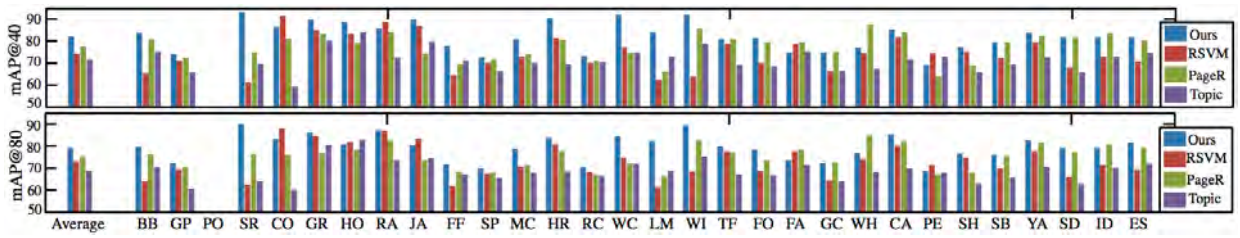


Figure 4.9: Quantitative comparison of personalized image retrieval between our method and three baselines using mAP@40(top) and mAP@80(bottom) metrics. The average performances for mAP@(40,80) in the left-most bar set are ours: (82.1%,79.2%), `RSVM`: (74.3%,72.8%), `PageR`: (77.4%,75.0%), `Topic`: (71.4%,68.7%).

Our algorithm significantly outperformed all the competitors in most topic classes for both tasks. In the average accuracy of normal retrieval, our mAP@40(80) values are higher by 5.6% (8.0%) points than the best baseline (`Topic`). In the average accuracy of personalized retrieval, our method also outperforms the best baseline (`PageR`) by 4.7% (4.2%) points for mAP@40(80). The personalized retrieval is more accurate to rank the images than the normal one, because knowing the user at query time provides a strong clue to correctly narrow down the search space.

4.6.3 Qualitative Results

Fig.4.10 shows some examples of retrieval comparison between our method in the top row and the best baseline in the bottom row. We illustrate top eight ranked images by each method, along with the average images of top 100 images to show the mean statistics of the two output sets. In these examples, our method reports fewer false positives (*i.e.* the images with red boundaries) than the best baselines.

As another qualitative result, Fig.4.11 shows the prediction power of our algorithm for unseen future images. Fig.4.11.(a) illustrates retrieval results for the *independence+day* at four t_q from different months. In each set, the top row shows five images that are sampled out of ten highest ranked predicted images for t_q, and the bottom row presents their best-matched actually taken images. The matched pairs are obtained from one-to-one correspondences by feature-wise distances. If the matched pairs are similar each other, it means that our algorithm can predict unseen future images very well. The *independence+day* is national holidays for many countries with different dates. Hence, according to four different t_q, we can observe various views of the events in different countries. For example, the second t_q (top-right) of Fig.4.11.(a) is near to the US independence day; the high ranked images show its common storyline: parades, parties with children, and fireworks at night. They are distinctive with the scenes in the Independence day of India (bottom-left) and an African country (bottom-right) of Fig.4.11.(a). The *mountain+camping* in Fig.4.11.(b) is a relatively stationary topic, in which the majority of pictures contain mountains, rock faces, and climbers. However, our ranking method can correctly capture the seasonal variation in the scenes, activities, and people's outfits. We can make similar observations in the *white+house* and the *wimbledon* topics as well, as shown in Fig.4.11.(c)–(d).

Fig.4.12 illustrates examples of the importance of personalization with four topic keywords.

Figure 4.10: Comparison of eight top-ranked images for normal image retrieval in (a)–(c) and for personalized image retrieval in (d)–(f), between our method in the top row and the best baseline in bottom row. The pictures with red boundaries are false positives. We also present average image of top 100 retrieved images in the left-most of each row.

For example, the *raptor* topic in Fig.4.12.(a) shows the variation of the term usages including a basketball team, a fighter aircraft, an eagle, and an ice hockey team, most of which are seemingly irrelevant to its first semantic meaning as a dinosaur. Each user perceives the term *raptor* narrowly for his or her interests, which are relatively stationary and predictable once they are learned. Other examples in Fig.4.12, including *grizzly*, *jaguar*, and *hornet*, also show that this personal variation is quite common in online user photo sets, and it can be correctly treated once the user history is learned.

Our experimental results conclude that some topics follow periodical patterns that are predictable, and our algorithm can enhance the image retrieval quality according to the temporal trends. Specifically, our method is successful for polysemous topics that show strong annual or periodic trends (*e.g.* sports related topics such as the *shark* and the *hornet*), and event topics that many people share but experience in different ways (*e.g.* outdoor activities such as *mountain+camping*). Moreover, we observe that the time-sensitive personalization is promising for image retrieval when a query keyword has a broad range of concepts, which are differently recognized according to people's thoughts and interests. Although the personalized search has been studied much in text retrieval research, our results reveal that images can convey more subtle information about user preferences that are hardly captured by texts.

4.7 Summary

In this chapter, we propose an approach for time-sensitive image ranking and retrieval using multi-task regression on multivariate point processes. With experiments on more than seven millions of

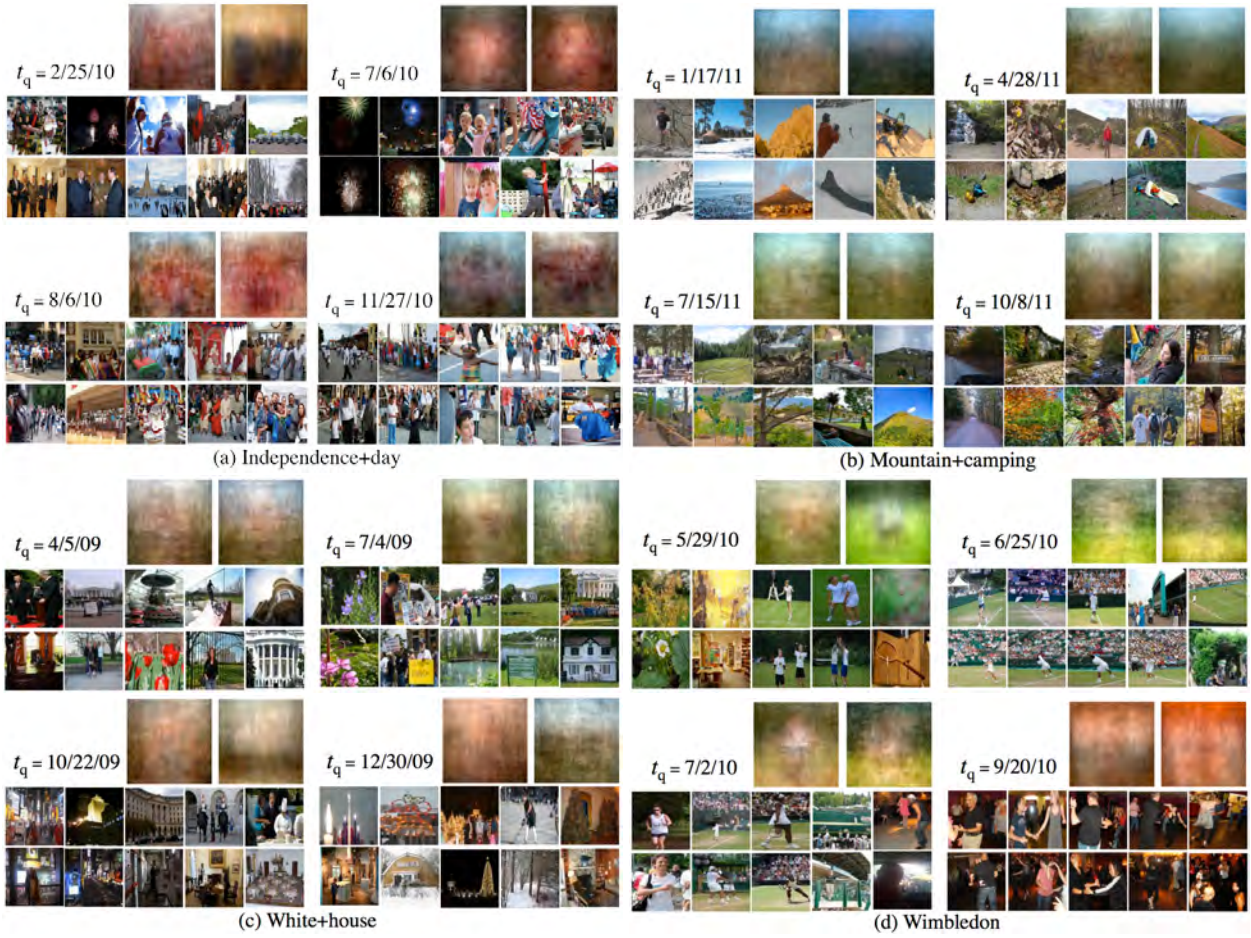


Figure 4.11: Examples of normal image prediction at four t_q in different months for the topics of (a) *independence+day*, (b) *mountain+camping*, (c) *white+house*, and (d) *wimbledon*. In all sets, we first find one-to-one correspondences between the estimated images $\mathcal{I}_e$ and the actual images $\mathcal{I}_+$ by the L2 measure, and then sample five image pairs per month. The first row shows the estimated images by our method, and the second row depicts their matched actual images. We also present the average images of top 100 estimated images (left) and their best-matched actual images (right).

Flickr images for 30 topic keywords, we show the superiority of the proposed approach over other candidate methods. The main contributions of this chapter can be summarized as follows.

- We develop an approach for time and optionally user sensitive image ranking and retrieval. To the best of our knowledge, our work is the first attempt so far on such retrieval algorithms that leverage the temporal aspects of large-scale Web photo collections.
- We design our image retrieval algorithm using multi-task regression on multivariate point processes. Although the point process models have been employed for analysis of neural spiking activities [Truccolo et al., 2005], and for event detection in video [Prabhakar et al., 2010], no attempt has been made for image retrieval and ranking so far. Consequently, our algorithm can automatically select and learn stochastic temporal models while satisfying

Figure 4.12: Examples of personalized image prediction at four different (t_q, u_q) pairs for the topics of (a) *raptor*, (b) *grizzly*, (c) *jaguar*, and (d) *hornet*. In all sets, we first find one-to-one correspondences between the estimated images $\mathcal{I}_e$ and the actual images $\mathcal{I}_+$ by the `L2` measure, and then sample five image pairs per month. The first row shows the estimated images by our method, and the second row depicts their matched actual images. We also present the average images of top 100 estimated images (left) and their best-matched actual images (right). Although the images are associated with the same keyword, their contents extremely vary according to users' interests.

a number of key challenges of Web image ranking, including flexibility, scalability, and retrieval accuracies.

Part II

Discovering Overlapping Contents of Image Collections

Part II – Discovering Overlapping Contents of Image Collections

Now we turn our attention to the unsupervised algorithms to detect the recurring contents of individual images across large-scale image sets. The online photos shared by general users are extremely diverse, but at the same time, they often share overlapping contents, which are likely to be statistically meaning visual information. Our methods in this part aim to quickly detect such repeating contents from the image set in the form of bounding boxes or pixel-wise segmentations.

This part consists of three chapters. First, we propose a fast and scalable alternating optimization technique to detect regions of interest (ROIs) in cluttered Web images without any supervision. The proposed approach discovers highly probable regions of object instances by iterating the following two functions: (i) finding the exemplar set (*i.e.* a small number of highly ranked reference ROIs) across the dataset and (ii) refining the ROIs of each image with respect to the exemplar set.

Second, we propose *CoSand*, a distributed cosegmentation approach for a highly variable large-scale image collection. The segmentation task is modeled by temperature maximization on anisotropic heat diffusion. We show that our method takes advantage of a strong theoretic property in that the temperature under linear anisotropic diffusion is a submodular function; therefore, a greedy algorithm guarantees at least a constant factor approximation to the optimal solution for temperature maximization. Our theoretic result is successfully applied to scalable cosegmentation as well as diversity ranking and single-image segmentation.

Third, we address a challenging image cosegmentation problem called multiple foreground cosegmentation (MFC), which concerns a realistic scenario in general users' photo sets over which a finite number of foregrounds repeatedly occur, but only an unknown subset of them is presented in each image. Our method builds on an iterative scheme that alternates between a foreground modeling module and a region assignment module, both of which are highly efficient and scalable. In particular, our approach is flexible enough to integrate any advanced region classifiers for foreground modeling, and our region assignment employs a combinatorial auction framework that enjoys several important properties for the large-scale cosegmentation such as optimality guarantee and linear complexity.

Chapter 5

Unsupervised Detection of Regions of Interests (ROI)

5.1 Introduction

In this chapter, based on our previous unsupervised object modeling and detection studies [Kim et al., 2008a,b], we explore the problem of detecting *regions of interest* (ROI) from large-scale image collections without relying on any supervision (Fig.5.1). We define the regions of interest as highly probable rectangular regions of object instances in the images. The extraction of ROIs is extremely helpful for recognition and Web user interfaces. For example, comparative studies in [Bosch et al., 2007; Chum and Zisserman, 2007] show that the ROI detection is useful to learn more accurate object models, which lead to nontrivial improvement of classification and localization performance. In the recognition of indoor scenes [Quattoni and Torralba, 2009], the local regions that contain objects may have special meaning to characterize the scenes. In addition, ROI detection can be exploited as a useful building block in addressing several computer vision problems, including segmentation prior design [Lempitsky et al., 2009], and image thumbnailing [Marchesotti et al., 2009]. As a user interaction tool, many Web applications allow a user to attach notes on user-specified rectangular regions in a cluttered image (*e.g.* Flickr and Facebook). Our algorithm can ease this cumbersome annotation by automatically suggesting the regions that a user may be interested in.

Our solution to the problem of unsupervised ROI detection is inspired by alternating optimization, which is one of widely used heuristics where optimization over two sets of variables is not straightforward, but optimization with respect to one while keeping the other fixed is much easier and solvable. This approach has been successful to solve a wide range of problems such as K-means, Expectation-Maximization, and Iterative Closest Point algorithm [Besl and McKay, 1992].

Figure 5.1: Detection of regions of interest (ROIs). Given a Web-sized dataset, our algorithm detects bounding boxed ROIs that are statistically significant across the dataset in an unsupervised manner. The yellow boxes are groundtruth labels, and the red and blue ones are ROIs detected by the proposed method.

The unsupervised ROI detection can be thought of as a chicken-and-egg problem between (1) finding exemplars of objects in the dataset and (2) localizing object instances in each image. If class-representative exemplars are given, the detection of objects in images is solvable (*i.e.* a conventional *detection* or *localization* problem). Conversely, if object instances are clearly annotated beforehand, the exemplars can be easily obtained (*i.e.* a *modeling* or *ranking* problem).

Given an image set, we first assume that each image itself is the best ROI (*i.e.* the most confident object region). Then a small number of highly ranked ones among the selected ROIs are chosen as exemplars (called *hub seeking*), which serve as references to refine the ROIs of each image (called *ROI refinement*). We repeat these two updates until convergence. The two steps are formulated as ranking in two different similarity networks of ROI hypotheses by link analysis. The *hub seeking* corresponds to finding a central and diverse hub set in a network of the selected ROIs (*i.e.* inter-image level). The *ROI refinement* is the ranking in a bipartite graph between the hub sets and all possible ROI hypotheses of each image (*i.e.* intra-image level).

The main advantages of our approach are summarized as follows. First, the proposed method is extremely simple and fast, with compelling performance. Our approach shows superior results over a state-of-the-art unsupervised localization method [Russell et al., 2006] for the PASCAL 06 dataset. We proposed a simple heuristic for scalability to make the computation time linear with the data size without severe performance drop. For example, the localization of about 200K images took only 4.5 hours with naive matlab implementation on a single PC equipped with Intel Xeon 2.83 GHz CPU (once image oversegmentation and feature extraction were done). Second, our approach is *dynamic* thanks to the *evolving network* representation. At every iteration, new ROI hypotheses are added and trivial ones are removed from the network while reusing a large portion of previously computed information. Third, unlike most previous work, our approach requires neither human annotation, meta-data, nor initial seed images. Finally, we evaluate our approach with a challenging Flickr dataset of up to 200K images. Although some work [Torralba et al., 2008] in image retrieval uses millions of images, this work has a different goal from ours. The objective of image retrieval is to quickly index and search the nearest images to a given query. On the other hand, our goal is to localize objects in every single image of a dataset without supervision.

5.2 ROI Candidates and Description

The input to our algorithm is a set of images $\mathcal{I} = \{I_1, \ldots, I_N\}$, where N is the size of the image set. The first task is to define a set of ROI hypotheses $\mathcal{R} = \{R_1, \ldots, R_N\}$ from the image set $\mathcal{I}$. Ideally, the set of ROI hypotheses $R_a = \{r_{a1}, \ldots, r_{am}\}$ of an image I_a enumerates all plausible bounding boxes, and at least one of them is supposed to be a good object annotation. Fig.5.2 shows the procedure of ROI hypotheses generation. Given an image, 15 segments are extracted by Normalized cuts [Shi and Malik, 2000]. The minimum rectangle to enclose each segment is defined as initial ROI hypotheses. Since the over-segmentation is unavoidable in most cases, the combinations of the initial hypotheses are also considered. We first compute pairwise minimum paths between the initial hypotheses using the Dijkstra algorithm. Then the bounding boxes to enclose those minimum paths are added to the set of ROI hypotheses. Finally, a largely overlapped pair of ROIs is merged if $\frac{r_{ai} \cap r_{aj}}{r_{ai} \cup r_{aj}} > 0.8$. Note that the hypothesis set always includes the image

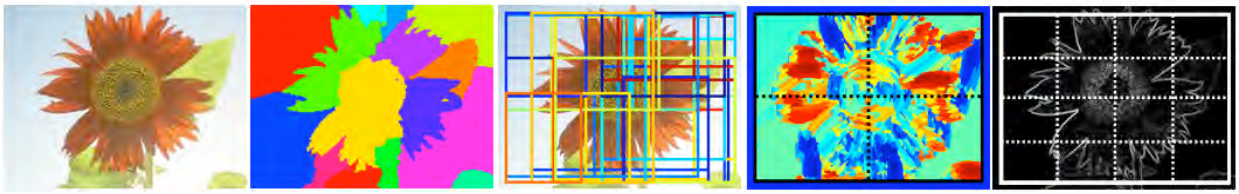


Figure 5.2: An example of ROI extraction and description. From left to right: (a) An input image. (b) 15 segments. (c) 43 ROI hypotheses. (d) Distribution of visual words. (e) Edge gradients.

itself as the largest candidate, and the average set size is about 50. One drawback of this simple heuristics is that it cannot detect under-segmented objects. Thus, the granularity of ROI candidates is a trade-off with computation speed.

Each ROI hypothesis is represented by two types of descriptors, which are spatial pyramids of SIFT visual words [Quattoni and Torralba, 2009] and HOG [Bosch et al., 2007]. As usual, the visual words are generated by vector quantization to randomly selected SIFT descriptors. K-means is applied to form a dictionary of 200 visual words. A visual word is assigned to each pixel of an image by finding nearest cluster center in the dictionary, and then binned using a two-level spatial pyramid. The oriented gradients are computed by Canny edge detection, and then the HOG descriptor is discretized into 20 orientation bins in the range of [0°,180°] by following [Bosch et al., 2007]. The pyramid level is up to three. The similarity between a pair of ROIs is measured by cosine similarity, which is simply calculated by dot product of two L_2 normalized histograms. Here both descriptors are equally weighted.

5.3 Iterative Detection of Regions of Interest

5.3.1 Similarity Networks and Link Analysis Techniques

All inferences in our approach are based on the link analysis of k-nearest neighbor similarity network between ROI hypotheses. The similarity network is a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V}$ is the set of vertices that are ROI hypotheses. $\mathcal{E}$ and $\mathcal{W}$ are edge and weight sets discovered by the similarity measure in the previous section. Each vertex is only connected to its k-nearest neighbors with $k = a \cdot \log |\mathcal{V}|$ [von Luxburg, 2007], where a is a constant set to 10. It results in a sparse network, which is more advantageous in terms of computational speed and accuracy. It guarantees that the complexity of network analysis is $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$ at worst. The network is row normalized so that the edge weight from note i and j indicates the probability of a random surfer jumping from i to j. The link analysis technique we use is the *PageRank* algorithm [Brin and Page, 1998]. Given a similarity matrix $\boldsymbol{G}$ (*i.e.* the adjacency matrix of $\mathcal{G}$), it computes the same length of *PageRank* vector $\boldsymbol{p}$, which assigns a ranked score to each vertex of the network. Intuitively, the *PageRank* scores of the network of ROI hypotheses are indices of the goodness of hypotheses.

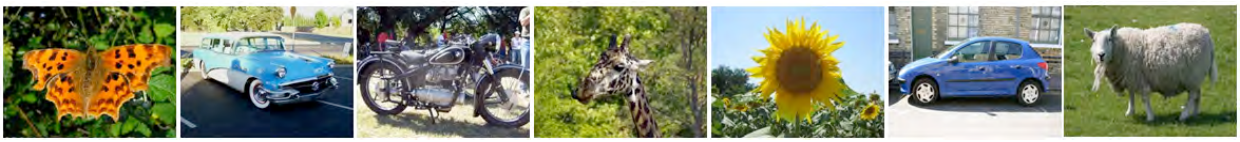


Figure 5.3: Examples of exemplars (*i.e.* hub images). The pictures illustrate highest-ranked images in 10,000 randomly selected images from five objects of our Flickr datasets and all {train+val} images from two objects of the PASCAL06.

Algorithm 4: The Algorithm

Input: The set of ROI hypotheses $\mathcal{R}$ for the input image set $\mathcal{I}$.
Output: The set of selected ROIs $\mathcal{S}^*(\subset \mathcal{R})$ and the exemplar set $\mathcal{H}^*(\subset \mathcal{S}^*)$ when converged at T.

1: $\mathcal{S}^{(0)} \leftarrow$ largest ROI hypothesis in each image.
while $\mathcal{S}^{(t-1)} \neq \mathcal{S}^{(t)}$ *or maximum iterations are not reached yet.* **do**
  2: Generate k-NN similarity network $\boldsymbol{G}^{(t)}$ of $\mathcal{S}^{(t)}$.
  3: $\mathcal{H}^{(t)} \leftarrow$ Hub seeking$(\boldsymbol{G}^{(t)})$, where the hub set $\mathcal{H}^{(t)} \subset \mathcal{S}^{(t)}$
  foreach $I_a \in \mathcal{I}$ *unless ROI selection of* I_a *is not changed for several consecutive times* **do**
    4: $s_a^{(t)} \leftarrow$ ROI refinement$(\mathcal{H}^{(t)}, R_a)$, where $s_a^{(t)}$: ROI selection of I_a, R_a: ROI hypothesis set of I_a.
    5: $\mathcal{S}^{(t)} \leftarrow \mathcal{S}^{(t)} \cup s_a^{(t)} \backslash s_a^{(t-1)}$.

Algorithm 5: Hub seeking function

Input: (1) Network $\boldsymbol{G}^{(t)}$. (2) Window size: d.
Output: (1) Hub set $\mathcal{H}^{(t)}$.

1: Compute *PageRank* vector $\boldsymbol{p}$ of $\boldsymbol{G}^{(t)}$.
foreach *vertex* $v \in \mathbf{G}^{(t)}$ **do**
  2: Find the neighbor set of v: $\mathcal{N}_v = \{u|$ max reachable probability from v to $u > d\}$.
  3: Find local maxima node of v: $\boldsymbol{m}(v) = \arg\max_u \boldsymbol{p}(\mathcal{N}_v)$ where $u \in \mathcal{N}_v$.
  4: $\mathcal{H}^{(t)} \leftarrow v$ if $v = \boldsymbol{m}(v)$.

Algorithm 6: ROI refinement function

Input: (1) Hub set $\mathcal{H}^{(t)}$. (2) R_a, ROI hypotheses of I_a.
Output: (1) The selected ROIs $s_a^{(t)}(\subset R_a)$.

1: Generate k-NN self-similarity matrix $\boldsymbol{W}_i$ of R_a and k-NN similarity matrix $\boldsymbol{W}_o$ between R_a and $\mathcal{H}^{(t)}$. Both of them are row-normalized.
2: Generate augmented bipartite graph $\boldsymbol{W} = \begin{pmatrix} \alpha \boldsymbol{W}_i & (1-\alpha)\boldsymbol{W}_o \\ \boldsymbol{W}_o^{\mathrm{T}} & \boldsymbol{0} \end{pmatrix}$.
3: Compute *PageRank* vector $\boldsymbol{p}$ of $\boldsymbol{W}$. **4**: $s_a^* = \arg\max_{r_{aj}} \boldsymbol{p}(r_{aj})$ where $r_{aj} \in R_a$.

5.3.2 Overview of Algorithm

Algorithm 4 summarizes the proposed algorithm. The main input is the set of ROI hypotheses $\mathcal{R}$ generated by the method of section 5.2. The output is the set of selected ROIs $\mathcal{S}^*(\subset \mathcal{R})$. In each image, usually one or two, and rarely more than three, of the most promising ROIs are chosen.

The basic idea of our approach is to jointly optimize the ROI selection of each image and the exemplar detection among the selected ROIs. Exemplars correspond to hubs in our network representation. We begin with images themselves as an initial set of ROI selection $\mathcal{S}^{(0)}$ (**Step 1**). Even though this initialization may be poor for many images, highly ranked hubs among the ROIs are

likely to be much more reliable. They are detected by the function Hub seeking (**Step 3**). Then, the hub sets are exploited to refine the ROIs of each image by the function ROI refinement (**Step 4**). In turn, those refined ROIs are likely to lead to a better hub set at next iteration. The alternating iterations of those two functions are expected to reach a convergence for not only the best ROI selection of each image but also the most representative ROIs of the data set as the exemplar set. Fig.5.4.(c) shows an example of refining ROI selections at every iteration. Although our algorithm forces to select at least one ROI for each image, the *PageRank* vector by ROI refinement can indicate the confidence of each ROI, which can be used to filter out wrongly selected ROIs later. Conceptually, both functions share a similar ranking problem to select a small subset of highly ranked nodes from the input networks of ROI hypotheses. They will be discussed in the following subsections in detail.

Inherently, a good initialization is essential for alternating optimization. Our key assumption here is as follows: *Provided that the similarity network is built from a sufficiently large number of images, the hub images are likely to be good references*. This is based on the finding of our previous work [Kim et al., 2008a]. If each visual entity votes for others that are similar to itself, this democratic voting can reveal the dominant statistics of the image set. The more repetitive visual information may get more similarity votes, which can be easily and quickly discovered as hubs by link analysis. Fig.5.3 supports this argument in our dataset. Although images in our datasets are highly variable, majority of pictures are taken from canonical views. Therefore, top-ranked images of our dataset clearly show the objects in the center with a significant size. Obviously, they are excellent initialization candidates.

Since we deal with discrete patches from unordered natural images on the Web, it is extremely difficult to analytically understand several important behaviors of our algorithm such as convexity, convergence, sensitivity to initial guess, and quality of our solution. One widely used assumption in the optimization with image patches is linearity with small incremental displacement (*e.g.* AAM [Cootes et al., 2001]). However, it is not the case in our problem and causes severe computation increase. These issues may be open challenges for the optimization of large-scale image analysis.

5.3.3 Hub Seeking with Centrality and Diversity

The goal of this step is to detect a hub set $\mathcal{H}^{(t)}$ from $\mathcal{S}^{(t)}$ by analyzing the network $\boldsymbol{G}^{(t)}$. The main criteria are *centrality* and *diversity*. In other words, the selected hub set should be not only highly ranked but also diverse enough not to lose various aspects of the dataset. To meet this requirement, we design the hub seeking inspired by Mean Shift [Comaniciu and Meer, 2002]; given data points, the algorithm creates a fixed-radius window at each point. Then each window iteratively moves into the direction of the maximum increase in the local density function until it reaches a local maximum. Those local maxima become the modes, and the data points that converge to the same maxima are clustered.

The proposed algorithm 5 works in the same manner. For each vertex, we define the search window in the form of maximum reachable probability d (**Step 2**). The window covers the vertices whose maximum reachable probability is larger than d. For example, given $d = 0.1, w_{ij} = 0.6, w_{jk} = 0.2$, the probability of vertices i to k is $0.6 \times 0.2 = 0.12 > d$. Thus, k is considered

inside the search window of i. For the density function, we use the *PageRank* vector, whose values are proportional to the vertex degrees if the graph is symmetric and connected [Zhou et al., 2004]. In **Step 3**, we compute the vector $\boldsymbol{m}$ that assigns the local maximum vertex within the window of each vertex. If $v = \boldsymbol{m}(v)$, the v is a local maximum, and it is added to $\mathcal{H}^{(t)}$. Additionally, we can easily perform the clustering from $\boldsymbol{m}$. For each node, the search window keeps moving the maximum direction indicated by $\boldsymbol{m}$ until it reaches the local maximum. Then the nodes that converge to the same maxima are clustered.

5.3.4 ROI Refinement

Formally, this step is to define a nonparametric function for each image $f_a : R_a \rightarrow R^+$ (positive real number) with respect to the hub set $\mathcal{H}^{(t)}$. Then the hypothesis with maximum ranked value is chosen as the best ROI. In order to solve this problem, we first construct an augmented bipartite graph $\boldsymbol{W}$ between the hub set $\mathcal{H}^{(t)}$ and all possible ROIs R_a as shown in **Step 2** of Algorithm 3 (see Fig.5.4(a)). For better understanding, let us first consider a pure bipartite graph with $\alpha = 0$. Then the matrix $\boldsymbol{W}$ represents the similarity voting between the ROI candidates and the hub set. If the *PageRank* vector $\boldsymbol{p}$ of $\boldsymbol{W}$ is computed, then $\boldsymbol{p}(R_a)$ summarizes the relative importance of each ROI hypothesis with respect to the $\mathcal{H}^{(t)}$, which is exactly what we require. Rather than a pure bipartite graph ($\alpha = 0$), we augment it by nonzero α. Fig.5.4.(b) explains the effects of α. The left image shows the result of $\alpha = 0$. Even though the red hypothesis is the maximum, several hypotheses near the dark gray car have significant values. With nonzero $\alpha = 0.1$, those hypotheses are allowed to augment each other, so the maximum ROI is changed to a hypothesis on the car. In terms of link analysis, if a random surfer visits nodes of ROI hypotheses (R_a), it jumps to other hypotheses with probability α or other hubs with $1 - \alpha$. Since the nearby hypotheses share large portions of rectangles, they have higher similarity, which results in more votes for nearby hypotheses.

5.3.5 Scalability Setting

The bottleneck of our approach is the **Step 3** of Algorithm 4. The network generation requires quadratic computation of cosine similarity of $\mathcal{S}^{(t)}$. In order to bound the computational complexity, we limit the maximum number of images to be considered each run of Algorithm 4 by constant number N. N should be small enough not to suffer from computational burden. Simultaneously, it should be large enough to successfully detect the meaningful statistics from an extremely variable dataset. (In experiments, N is set to 10,000.) If the dataset size $|\mathcal{I}| > N$, we randomly sample N images from $\mathcal{I}$ and construct initial consideration set $\mathcal{I}_c \subset \mathcal{I}$. Algorithm 4 is applied to the image set $\mathcal{I}_c$ to obtain $\mathcal{S}_c^*$. Then we generate new $\mathcal{I}_c$ by sampling from unvisited images of $\mathcal{I}$. In order to reuse the result of $\mathcal{S}_c^*$ for the new $\mathcal{I}_c$, we sample $x\%$ of N from previous $\mathcal{S}_c^*$ based on the *PageRank* values of the network $\boldsymbol{G}^*$ of $\mathcal{S}_c^*$. In other words, the highly ranked (*i.e.* highly confident) ROIs in the previous step are reused to expedite the convergence of next iteration. We iterate the above strategy until all images are examined. This simple heuristic allows our technique to analyze an extremely large dataset in a linear time without significant performance drop.

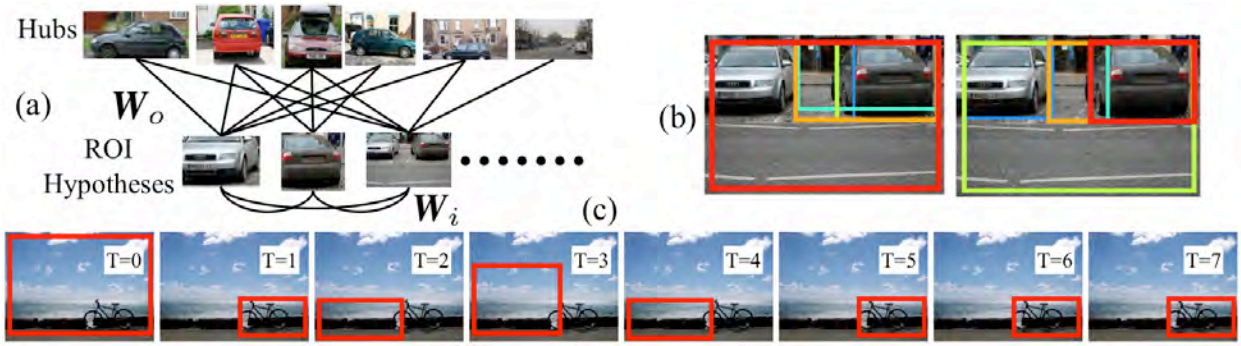


Figure 5.4: (a) An example of a bipartite graph between the hub set and ROI hypotheses of an image. The similarity between hubs and hypotheses is captured by W_o and the affinity between hypotheses by W_i. The hub set is sorted by *PageRank* values from left and right. The values of leftmost and rightmost are 0.0081 and 0.0024, respectively. The hub set successfully covers various views of the *car* class. (b) The effect of the augmented bipartite graph. The left image is with $\alpha = 0$ and the right with $\alpha = 0.1$. The ranking of hypotheses is represented by jet colormap from red (high) to blue (low). In the left, the weights from the red box to the blue one are (0.052, 0.050, 0.049, 0.049, 0.049); in the right, (0.060, 0.060, 0.059, 0.059, 0.057). (c) An example of ROI evolution. At $T = 0$, the selected ROI is an image itself and is converged to the real object after $T = 5$.

5.4 Experiments

We evaluate our approach with two different experiments, (1) performance tests with PASCAL VOC 2006[1] and (2) scalability tests with Flickr images. The PASCAL dataset provides groundtruth labels, so our approach is quantitatively evaluated and compared with other approaches. Using Flickr dataset, we examine the scalability of our method in a real-world problem. The images are collected by a query that consists of one object word and one context word. We downloaded images of the objects {*butterfly*+*insect*(69,990), *classic*+*car*(265,731), *motorcycle*+*bike*(106,590), *sunflower*(165,235), *giraffe*+*zoo*(53,620)}. The numbers in parentheses are dataset sizes.

5.4.1 Performance Tests

The input of our algorithm consists of unlabeled images, which may include a single object (called as *weakly supervised*) or multiple objects (called *unsupervised*). For *unsupervised* cases, we perform not only localization but also classification according to object types. The PASCAL 06 dataset is challenging so that only very rare previous work has used it for unsupervised localization. For comparison, we ran publicly available code of one of the state-of-the-art techniques proposed by Russell et al in the identical setting with ours.

The PASCAL dataset consists of {train+val+test}. However, our approach requires only images as an input, and thus all of the {train+val+test} images are used without discrimination between them. Note that our task is an image annotation not a learning problem that requires training and test steps. The performance is evaluated by following the protocol of PASCAL evaluation:

[1]The dataset is available at http://www.pascal-network.org/challenges/VOC/.

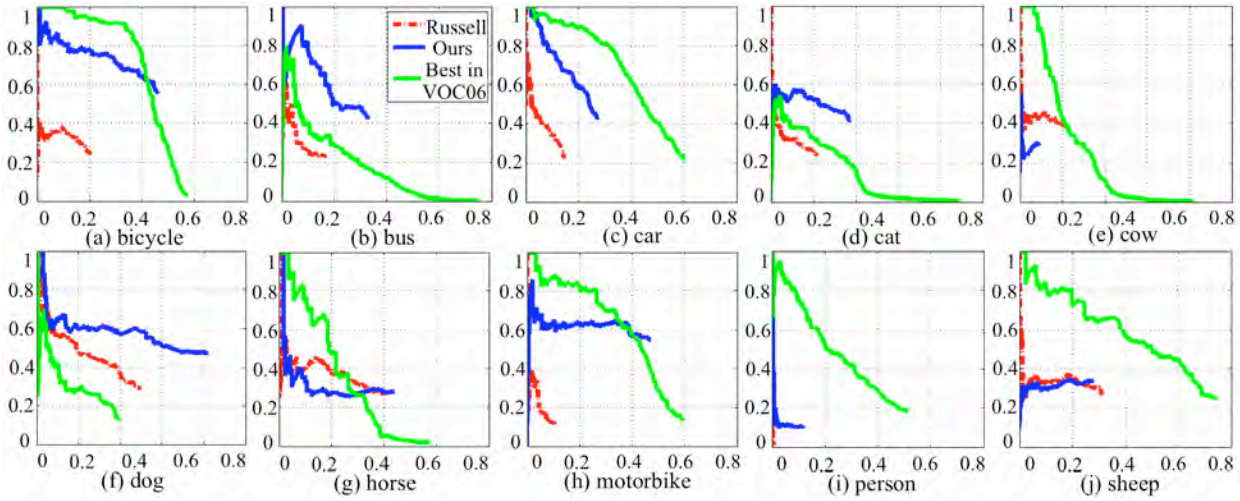


Figure 5.5: Results of weakly supervised localization. PR curves for the {test} sets of all objects in the PASCAL 06 dataset for ours (blue), [Russell et al., 2006] (red), and the best of VOC06 (green). Note that our localization and that of [Russell et al., 2006] are unsupervised, whereas the VOC06 localization is supervised. (X-axis: recall; Y-axis: precision).

(1) The accuracies are measured from only the {test} set. In practice, there is very little performance difference between analysis of all {train+val+test} and {test} only. (2) The detection is considered correct if the overlap between the prediction and ground truth exceeds 50%.

Weakly supervised localization. Fig.5.5 shows the detection performance as Precision-Recall (PR) curves. For [Russell et al., 2006], we iterate experiments by changing the number of topics from two to six, and report the best results. For fair comparison between our results and [Russell et al., 2006], we select only the single best bounding box in each image. We also present the best result of each object in VOC06 competition. Strictly speaking, it is not a fair comparison because the experimental setup of VOC06 competition is supervised while ours are unsupervised. However, we include them as references to show how closely our approach can reach the best supervised methods in VOC 06 for the localization. Although the performance varies according to objects, our approach significantly outperformed [Russell et al., 2006] except in *cow*. Promisingly, the performances of our approach for *bicycle* and *motorbike* are comparable, and those for *bus*, *cat*, and *dog* objects are superior to the bests of the supervised methods in VOC06.

Unsupervised classification and localization. Here we evaluate how well our approach works for unsupervised classification and localization tasks (*i.e.* images of multiple objects are given without any annotation). Since both our method and [Russell et al., 2006] aim at sub-image level classification and detection, we first find out the most confident region of each image, and run the LDA clustering for [Russell et al., 2006] and spectral clustering [Shi and Malik, 2000] for our method. Fig.5.6 shows ROC curves as the evaluation of classification by following the VOC06 protocol. We also show the best of the VOC06 submissions for supervised classification as reference. As shown in Fig.5.6.(a)−(c), our method and [Russell et al., 2006] present similar ROC performance. In other words, both methods are quite good at ranking for classification. However, the classification rates of our method are better by about 10% for both 3-object and 4-object

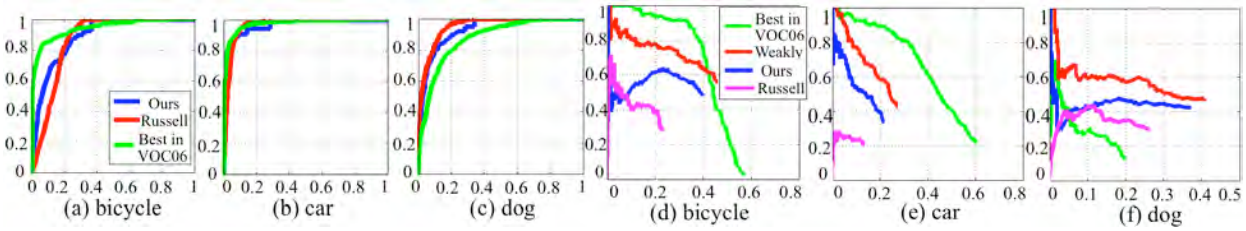


Figure 5.6: Results of unsupervised classification and localization. (a)−(c) ROC curves for the {test} set of $\{bicycle, car, dog\}$ for ours (blue), [Russell et al., 2006] (red), and the best of VOC06 (green). The AUCs of ours, [Russell et al., 2006], and the best of VOC06 are as follows; *bicycle*: (0.892, 0.869, 0.948), *car*: , and *dog*: (0.932, 0.954, 0.876), respectively. (X-axis: false positive rates, Y-axis: true positive rates). (d)−(f) PR curves for unsupervised localization of ours (blue) and [Russell et al., 2006] (magenta). As references, we also represent the results of our weakly supervised localization (red) and the best of VOC 06 (green). (X-axis: recall, Y-axis: precision).

cases. (Ours: **69.08**%; [Russell et al., 2006]: 59.05% for $\{bicycle, car, dog\}$. Ours: **59.51**%; [Russell et al., 2006]: 50.99% for $\{bicycle, car, dog, sheep\}$.) We show the unsupervised localization performance as PR-curves in Fig.5.6.(d)−(f). As references, we also represent the results of our weakly supervised experiments and the bests of VOC 06 for corresponding objects. We observe a nontrivial performance drop because the unsupervised setting is more challenging than the weakly supervised one due to the classification errors and distraction by other objects in the dataset.

5.4.2 Scalability Tests

It is an open question how to evaluate the results of a large number of Web images that have no ground-truth. For a quantitative evaluation, we manually annotated 0.5% randomly selected images of datasets, and they are used as limited but approximate indices of performance measures. According to the data sizes used in experiments, we randomly pick $x\%$ from the annotated set and $(100 - x)\%$ from the non-annotated set. The x is $\{20, 10, 5, 1, 0.5, 0.5\}$ for the dataset sizes of $\{500, 5K, 10K, 50K, 100K, 200K\}$.

Weakly supervised localization. One interesting question we address here is how performances and computation times vary as a function of data sizes. The experiments are repeated ten times for each dataset size, and the median (*i.e.* fifth-best) performance scores are reported. Similarly to previous tests, we select only the single best ROI per image. As shown in Fig.5.7, the performances of 500 images highly fluctuate, but those of the dataset sizes above 5K are stable. As dataset sizes increase, a small performance improvement is observed. Since the maximum number of images at each execution of the algorithm is bounded by $N(= 10,000)$, the computation times are linear to the number of images, and the performances of the data sizes above N are similar one another.

Perturbation tests. Here we test the goodness of selected ROIs from a different view: robustness of ROI detection against random network formation. For example, given an image I_a, we can generate 100 sets of 200 randomly selected images including I_a. If the ROI selection for I_a is repetitive across 100 different sets, we can say the ROI estimator for I_a is confident. This procedure is similar to *bootstrapping* or *cross-validation*.

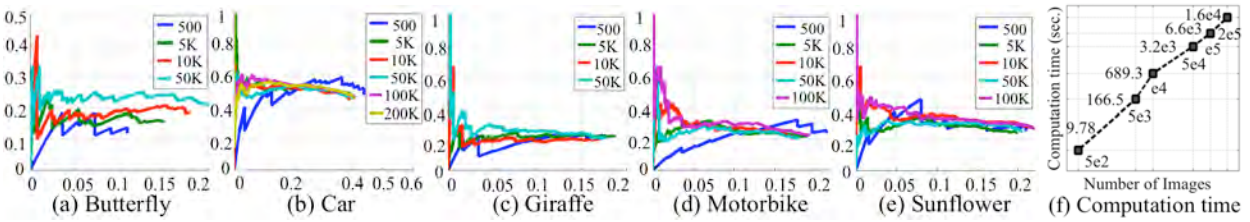


Figure 5.7: Weakly supervised localization. (a) PR curves for five objects of our Flickr dataset by varying dataset sizes from 500 to 200K. (b) The log-log plot between the number of images and the computation time for the *car* object. The slope of each range is $\{1.23, 2.05, 0.95, 1.05, 1.28\}$ from left to right, respectively.

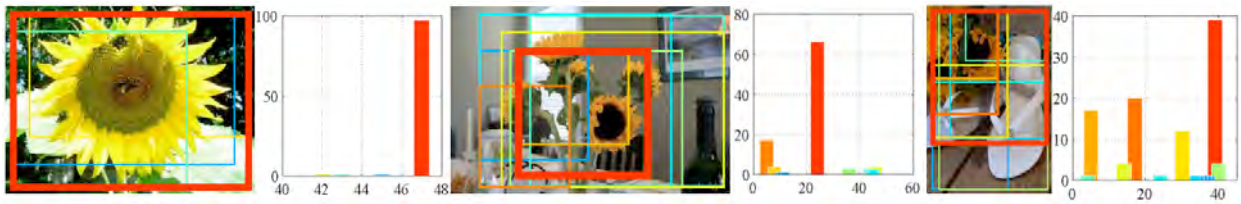


Figure 5.8: Examples of perturbation tests. The histograms summarize how many times each ROI is selected in 100 random sets. The frequencies of ROIs are represented in the images by the thickness of bounding boxes and the jet colormap from red (high) to blue (low). From left to right, the entropies of the distributions are $\{0.2419, 1.6846, 2.4331\}$, respectively. (X-axis: ROI hypotheses; Y-axis: frequencies).

Fig.5.8 shows some examples of the perturbation tests. The histogram indicates how many times each ROI hypothesis is chosen among 100 random sets. From the left image to the right, one can see the increase of the difficulty of ROI detection. A peak is observed for the obvious left image, but the distribution is wider for the challenging right image. The entropy of the distribution in the caption of Fig.5.8 can be an index of the measure of difficulty or the confidence of the estimator for the image.

More localization examples. Fig.5.9 shows more examples of localization by our approach. The third row illustrates some typical examples of failure. Frequently co-occurred objects can be detected instead, such as insects on flowers, many different animals in the zoo, and persons everywhere. Another common case of failure is that our approach sometimes detects small multiple instances or a part of an object as a single ROI (*e.g.* a giraffe's face instead of the whole body).

5.5 Summary

We develop an alternating optimization approach for scalable unsupervised ROI detection by analyzing the statistics of similarity links between ROI hypotheses. The main contributions of this chapter can be summarized as follows.

- We propose an alternating optimization approach based on iterative link analysis. The unsupervised ROI detection is achieved by alternating between solving two sub-problems: (i) finding exemplars of objects in the dataset and (ii) localizing object instances in each image.

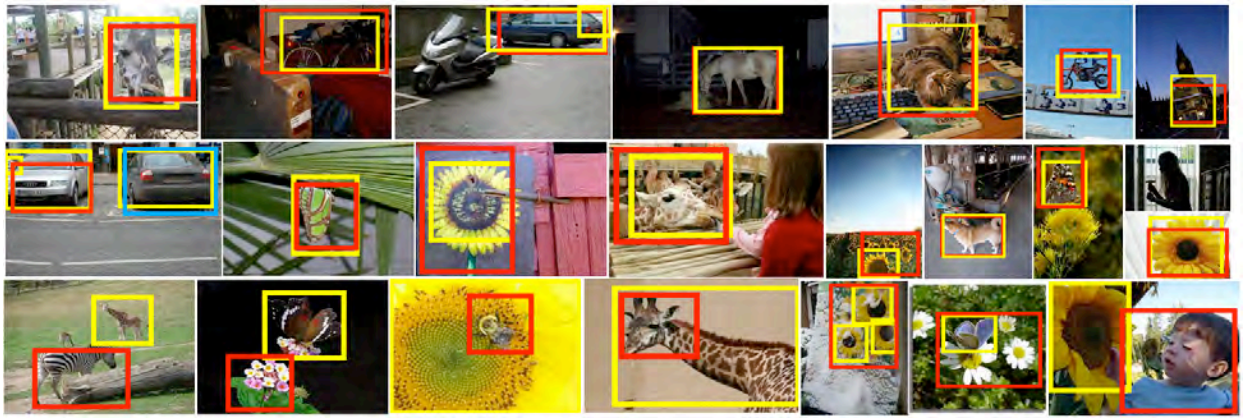


Figure 5.9: More examples of the ROI discovery. The first and second rows represent successful detections, and the third row illustrates some typical failures. The yellow boxes are groundtruth labels, and the red and blue ones are ROIs detected by the proposed method.

- This idea enables the ROI detection to be extremely simple and fast, with compelling performance on both PASCAL06 and Flickr datasets. (*e.g.* The ROI detection takes only 4.5 hours for about 200K images on a single machine).
- Unlike most previous work, our approach requires neither human annotation, meta-data, nor initial seed images. We take advantage of statistical consistency in the very large image sets.

Chapter 6

Diversity Ranking, Image Segmentation, and Cosegmentation

6.1 Introduction

In this chapter, we investigate an optimization problem connected to the anisotropic diffusion, which is potentially useful for efficiently solving a wide range of computer vision problems such as image segmentation [Zhang et al., 2010], optical flow estimation [Bruhn et al., 2005], and image smoothing [Weickert, 1998]. This optimization problem can be summarized in a single sentence as follows: *given a system under heat diffusion and finite K heat sources, where should one place all the sources in order to maximize the temperature of the system?* In terms of image segmentation, the optimization corresponds to finding the K segment centers that maximize the segmentation confidence of every pixel in the image[1]. (*e.g.* the ideal segmentation is that every pixel has confidence one to be clustered with one of K segment centers).

Since a naive combinatorial approach to this optimization is NP-hard, we seek a much more efficient and scalable approximate solution by taking advantage of a strong theoretical property known as *submodularity* underlying our problem. We first prove that, the *temperature*, which is to be optimized in our problem, is a *submodular function* if the system is under anisotropic diffusion. It is a well-known beneficial property of submodular functions that one can achieve at least a constant factor of the optimal by a greedy algorithm, which iteratively chooses K locations that maximize the marginal gain of the temperature. Such a greedy solution is particularly promising for tasks on large-scale image collections.

For better understanding of the proposed optimization, we first show that our method is able to effectively solve the diversity ranking [Zhu et al., 2007], which is to re-rank the items to reduce redundancy while maintaining their centrality. Intuitively, in order to maximize the temperature of the system with limited sources, the sources should be located in the center-of-gravity regions that are densely connected to other elements of the system with high conductivity. At the same time, the sources should be sufficiently distant from one another to have a broad and balanced coverage of the system. These two objectives are universal to a wide variety of machine learning and pattern recognition problems. Then, this temperature optimization idea is extended into the segmentation

[1] We use the following terminological correspondences between temperature maximization and image segmentation: *temperature* $\equiv$ *segmentation confidence*, *heat sources* $\equiv$ *segment centers*, *conductance or diffusivity* $\equiv$ *similarity between feature vectors of pixels*.

Work	Models / Algorithms	M	K
Ours	Diffusion/ Submodularity	$\geq 10^3$	Any
[Joulin et al., 2010]	Discriminative clustering	≤ 30	2
[Mukherjee et al., 2011]	MRF+ Rank-1 global / Iterative opt.	≤ 20	2
[Hochbaum and Singh, 2009]	MRF+Reward global / Graph Cuts	2	2
[Rother et al., 2006]	MRF+L1 global / Trust Region GC	2	2
[Vicente et al., 2010]	Boykov-Jolly / Dual Decomposition	2	2

Table 6.1: Comparison between our approach and existing unsupervised cosegmentation methods. Models and optimization algorithms are summarized. Let M and K denote the number of images and segments, respectively. Most previous work has mainly focused on binary *figure-ground* segmentation of small-sized image sets.

of a single image and the cosegmentation problem, in which largely co-occurred regions in the image set are jointly segmented out. We name our cosegmentation algorithm as CoSand, standing for **CoS**egmentation via **an**isotropic **d**iffusion.

6.1.1 Background

We first briefly review some background research that related to our work.

Anisotropic diffusion: The heat diffusion is represented by a partial differential equation called *heat equation*, which describes how the distribution of heat (or temperature variation) changes to achieve an equilibrium state in the system. It has been a successful technique in image processing and computer vision; notable examples include image segmentation [Zhang et al., 2010], optical flow estimation [Bruhn et al., 2005], and image smoothing [Weickert, 1998]. In these applications, the temperature corresponds to various objectives, which are the clustering confidence in segmentation, the optical flow in motion analysis, or the RGB value in image smoothing. In this chapter, we focus on image segmentation, but our optimization is also easily extendible to those problems such as large-scale edge-preserving image smoothing or layered motion segmentation in video.

Submodular optimization: In recent years, submodular optimization has emerged as a useful optimization tool in a variety of machine learning problems such as active learning, structure learning, clustering, and ranking [Krause and Guestrin, 2008; Leskovec et al., 2007]. The submodular function is characterized as a *diminishing return* property that states that, the marginal gain of adding an element to a smaller subset of $\mathcal{S}$ is higher than that of adding it to a larger subset of $\mathcal{S}$. Some typical submodular functions explored in machine learning include a cut function in a graph and the entropy and the information gain of Gaussian random variables [Krause and Guestrin, 2008]. To the best of our knowledge, our work is the first to address submodular optimization on diffusion in physics[2].

Cosegmentation: Since we already survey the existing work of image cosegmentation, we here present Table 6.1 to summarize the comparison of our work and other unsupervised cosegmentation

[2] *Diffusion* is a heavily overloaded term that is used with different meanings in diverse fields. Here it refers to *diffusion in physics* that is described by a partial differential equation such as heat diffusion or electric current.

methods. Simply put, cosegmentation is the problem of jointly segmenting each of M images into K different regions. Our approach is unique in terms of M and K. Most previous work has dealt with binary *figure-ground* segmentation (K=2) of small sized image sets (mostly M=2 but $M \leq 30$ in [Joulin et al., 2010]). On the other hand, our algorithm is able to segment a large-scale dataset with any arbitrary K. We tested with $M \geq 10^3$ images in our experiments, but a more scalable setup is also applicable. That is, the magnitude of the dataset sizes in our experiments exceeds those of previous work by orders of magnitude. The optimization methods for cosegmentation in most previous work, except [Joulin et al., 2010], are based on the graph-cut algorithm. Hence, it is not straightforward for them to be extended to arbitrary K-way cuts. In theory, the method of [Joulin et al., 2010] can perform cosegmentation with $K>2$, but it was not evaluated in the paper. On the other hand, our algorithm can attain a constant factor approximation to the optimum with any arbitrary K. The computation time is at worst linear with K.

In addition, our approach is easily parallelizable; most computations occur independently on individual images, and then an integration step quickly merges all outputs from individual images into a coherent cosegmentation result. Our approach also supports the automatic selection of K and robustness against a wrong choice of K, which will be shown in experiments of Section 6.4.

6.2 Submodularity and Diffusion

6.2.1 Optimization on Anisotropic Diffusion

We begin with a general theory of anisotropic diffusion [Weickert, 1998]. Let Ω denote the domain of a system and x be a point in $\Omega \in \mathbb{R}^d$ ($x \in \Omega$). Since we are usually interested in discrete systems (*e.g.* images or graphs), let us assume that Ω is a discrete set of points[3]. The $u(x,t)$ is the temperature at position x at time t and $D(x)$ is a $d \times d$ positive symmetric tensor called the *diffusion tensor*. The *linearity* of diffusion indicates that D is not a function of u or ∇u. The *anisotropy* means that the flux $-D(x)\nabla u(x,t)$ and the gradient $\nabla u(x,t)$ are not parallel in an image domain. The diffusion equation of such a system is as follows:

$$\frac{\partial u(x,t)}{\partial t} = \mathrm{div}\big(D(x)\nabla u(x,t)\big). \tag{6.1}$$

Our optimization problem is that of maximizing the sum of temperature of the system that is under anisotropic diffusion by choosing the locations of K heat sources. Formally,

$$\begin{aligned} \max \quad & \int_{x\in\Omega} u(x,t)dx \\ \text{s.t.} \quad & \frac{\partial u(x,t)}{\partial t} = \mathrm{div}\big(D(x)\nabla u(x,t)\big) \\ & u(g) = 0,\ u(s) = 1 \ \text{ for } s \in \mathcal{S} \subset \Omega,\ |\mathcal{S}| \leq K \end{aligned} \tag{6.2}$$

[3]It is not difficult to obtain the corresponding results of following arguments for the continuous (*i.e.* Ω and t: continuous) and semi-discrete (*i.e.* Ω: discrete, t: continuous) cases [Weickert, 1998].

where we assume that the temperature of environment (*i.e.* outside of the system Ω) is zero (*i.e.* $u(g) = 0$), and the source temperature is one at any time (*i.e.* $u(s) = 1$)[4].

For physical analogy, one may imagine a metal plate in open air, and its temperature is to be maximized with K point heat sources. Without loss of generality, we explicitly decompose the heat flux at every point into two parts - a flux within the system and a dissipation flux to out of the system. Let $z(x)$ be a positive scalar diffusivity to the environment at x, and then the dissipation heat loss is $-z(x)(u(x)-u(g))$. If $z(x) = 0$ for $\forall x \in \Omega$, the system is *insulated*. From now on, we assume that $-D(x)\nabla u(x,t)$ solely contributes to the diffusion within the system.

In order to efficiently solve the optimization of Eq.(6.2) for arbitrary K, we first prove that the temperature under the linear anisotropic diffusion is submodular.

Theorem 1 (Submodularity on Anisotropic Diffusion). Suppose that the system undergoes *linear anisotropic diffusion*. Let $u(x,t;\mathcal{S})$ be the temperature at position x at time t when identical heat sources are attached to $\mathcal{S}(\subset \Omega)$. Then, the following statements hold for $\forall x \in \Omega, \forall t \in [0,\infty]$.

(T1) $u(x,t;\emptyset) = 0$
(T2) $u(x,t;\mathcal{S})$ is *nondecreasing* and *submodular*.

Proof. Here we consider the discrete case where time and space are discretized; it is not difficult to draw the same conclusion for the continuous case. Without loss of generality, we assume that the source temperature is one and the environment temperature is zero. Then, the temperature can be interpreted as a *probability*. During the proof we drop t in the notation because the following arguments always hold for any t.

Note that the system is under *linear anisotropic* diffusion, which means that the system Ω and the diffusivity $D(x)$ including the dissipation diffusivity $z(x)$ are invariant for any t.

(T1) $u(x;\emptyset) = 0$ is obvious because without a source the system has zero temperature (*i.e.* the same temperature with that of environment).

(T2) $u(x;\mathcal{A})$ is *nondecreasing* (*i.e.* $u(x;\mathcal{A}) \leq u(x;\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$) because the temperature of the system is always higher with more heat sources. Physically, it means the *energy conservation law*.

The $u(x;\mathcal{A})$ is *submodular* if Eq.(6.3) holds for all placements $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and a new source $s \in \mathcal{V}\backslash\mathcal{B}$:

$$u(x;\mathcal{A}\cup\{s\}) - u(x;\mathcal{A}) \geq u(x;\mathcal{B}\cup\{s\}) - u(x;\mathcal{B}). \tag{6.3}$$

We shall prove the submodularity of u by induction on the distance $d(x,s)$. The induction proof consists of two steps, which are (a) **base step** showing that Eq.(6.3) holds for $d(x,s)$=0, and (b) **induction step** showing that if Eq.(6.3) holds for $d(x,s) \leq r$, then it is true for $d(x,s) \leq r+\delta r$ with a small $\delta r > 0$ as well.

(a) **Base step**: For x with $d(x,s) = 0$ (*i.e.* $x = s$), $u(x;\mathcal{A}\cup\{s\}) - u(x;\mathcal{A}) \geq u(x;\mathcal{B}\cup\{s\}) - u(x;\mathcal{B})$ because (i) $u(s;\mathcal{A}\cup\{s\}) = u(s;\mathcal{B}\cup\{s\}) = 1$ and (ii) $u(s;\mathcal{A}) \leq u(s;\mathcal{B})$ since $u(x;\mathcal{A})$ is nondecreasing for all $x \in \mathcal{V}$.

[4]Here we consider only *Dirichlet* boundary conditions.

(b) **Induction step**: Suppose that for all x with $d(x,s) \leq r$, Eq.(6.3) holds. We need to show that Eq.(6.3) is true for all x' with $d(x',s) = r + \delta r$ with a small $\delta r > 0$ as well.

If the system undergoes diffusion, as shown in Eq.(6.4), the temperature at point x is represented by the weighted sum of the temperatures of its neighbors $\mathcal{N}(x)$ [Tschumperle and Deriche, 2005; Weickert, 1998]. It is based on the physical fact that the heat diffusion is driven by thermal non-equilibrium and converges to local energy balance.

$$u(x) = \sum_{p \in \mathcal{N}(x)} g(p)u(p) \quad \text{for } \forall x \in \Omega \tag{6.4}$$

where $p \in \mathcal{N}(x)$ is a point of the neighbor set of x and $g(p)$ is a Kernel function describing how much the temperature at p ($u(p)$) contributes to the temperature at x ($u(x)$). $g(p)$ is the function of the diffusivity and the distance between p and x[5]. Therefore, $g(p)$ is invariant for any t under the *linear anisotropy* assumption (*i.e.* the system and the diffusivity are fixed for any t).

For a position x' with $d(x',s) = r + \delta r$, $\mathcal{N}(x')$ can be divided into two sets $\mathcal{P} = \{p | p \in \mathcal{N}(x'), d(p,s) \leq r\}$ and $\mathcal{Q} = \{q | q \in \mathcal{N}(x'), d(q,s) > r\}$. Therefore, $u(x'; \mathcal{A} \cup \{s\}) - u(x'; \mathcal{A}) \geq u(x'; \mathcal{B} \cup \{s\}) - u(x'; \mathcal{B})$ holds by Eq.(6.4) and induction hypotheses of (i) $u(p; \mathcal{A} \cup \{s\}) - u(p; \mathcal{A}) \geq u(p; \mathcal{B} \cup \{s\}) - u(p; \mathcal{B})$ for all $p \in \mathcal{P}$ and (ii) $u(q; \mathcal{A} \cup \{s\}) = u(q; \mathcal{A})$ and $u(q; \mathcal{B} \cup \{s\}) = u(q; \mathcal{B})$ for all $q \in \mathcal{Q}$. ■

Let $U(t; \mathcal{S}) = \int_{x \in \Omega} u(x, t; \mathcal{S}) dx$ be the temperature sum of the system at t. Intuitively, $U(t, \mathcal{S})$ is also submodular since it is the sum of submodular functions [Krause and Guestrin, 2008]. Theorem 2 below states that a simple greedy algorithm achieves a near optimal solution for the maximization of a submodular function.

Theorem 2 ([Nemhauser et al., 1978]). Let u be a submodular, nondecreasing set function and $u(\emptyset) = 0$. Then, the greedy algorithm finds a set $\mathcal{S}_G$ such that $u(\mathcal{S}_G) \geq C \cdot \max_{|\mathcal{S}| \leq K} u(\mathcal{S})$ where $C = (1 - 1/e) \approx 0.632$.

6.2.2 Diversity ranking and clustering

For better understanding of the above diffusion formulation, let us first examine a simple case – diversity ranking in a graph. Diversity ranking [Zhu et al., 2007] aims to re-rank items to reduce redundancy while maintaining their centrality, which is highly relevant to the goal of segmentation. In the next section, we extend this idea into the cosegmentation problem.

Suppose the following; (1) The system Ω is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. (2) We are interested in the steady state (*i.e.* when $t \to \infty$), thus we can drop t in our notation. (3) The diffusivity (*i.e.* conductance) is defined by Gaussian similarity between the features of vertices:

[5] The simplest discrete form of Eq.(6.4) with a 2D regular grid is $u(i,j) = \big(u(i-1,j) + u(i+1,j) + u(i,j-1) + u(i,j+1)\big)/4$ with $x = (i,j)$. In this case, $\mathcal{N}(x) = \{(i,j-1), (i,j+1), (i-1,j), (i+1,j)\}$ and $g(p) = 1/4$ for $\forall p \in \mathcal{N}(x)$. In a more accurate discretization [Tschumperle and Deriche, 2005], the Gaussian Kernel is used: $g(p) = \exp\big(-(x-p)^T D(p)(x-p)/\sigma_p\big)$ where σ_p is a normalization constant so that $\sum_{p \in \mathcal{N}(x)} g(p) = 1$.

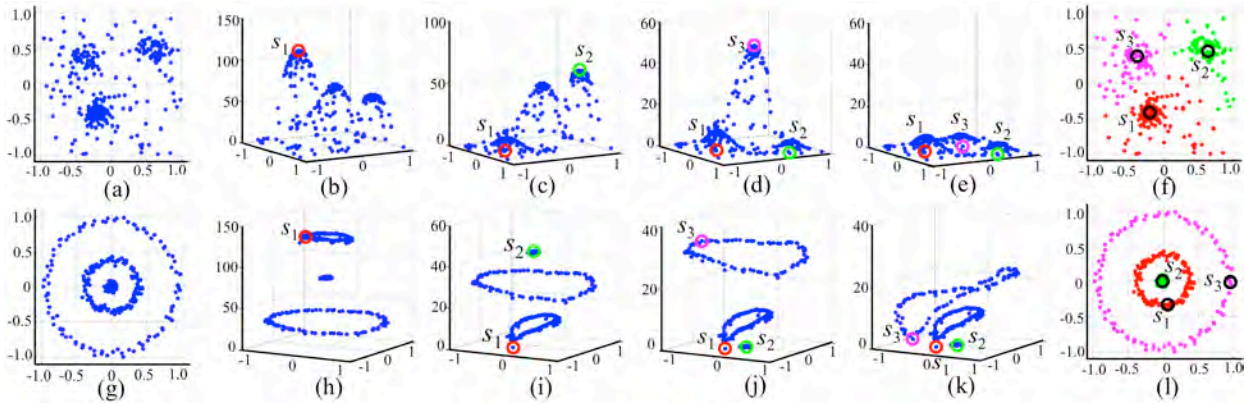


Figure 6.1: Two toy examples of diversity ranking. The data points are randomly generated from three Gaussian distributions in (a) and three co-centric circles in (g). In (b)-(e) and (h)-(k), the marginal temperature gain of each point $U(\mathcal{S} \cup \{x\}) - U(\mathcal{S})$ is shown along z-axis. $s_k (\in \mathcal{S})$ are iteratively selected by solving Eq.(6.7). Once a point is selected, the marginal gains of its neighbors largely drop because they already get high temperatures. In (f)(l),final three clusters are shown. The clustering from $\mathcal{S}$ will be discussed in Algorithm 7.

$$d_{xy} = \begin{cases} \exp(-\beta ||\mathbf{g}(x) - \mathbf{g}(y)||^2), & \text{if } (x, y) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

where $\mathbf{g}(x)$ is the feature vector at node $x \in \mathcal{V}$. (4) The dissipation conductance at a vertex x is constant in time, denoted by z_x. That is, each node x is connected to an environment node g with conductance of z_x. With these assumptions, diffusion reduces to the famous random walk model [Grady, 2006] or Gaussian random fields [Zhu et al., 2003]. The optimization problem in Eq.(6.2) grounds to a more specific form below[6]:

$$\begin{aligned} \max \ & \sum_{x \in \mathcal{V}} u(x) \\ \text{s.t. } & u(x) = \frac{1}{a_x} \sum_{(x,y) \in \mathcal{E}} d_{yx} u(y) \ \text{ for } a_x = \sum_{(x,y) \in \mathcal{E}} d_{yx} + z_x \\ & u(g) = 0, \ u(s) = 1 \ \text{ for } s \in \mathcal{S} \subset \mathcal{V}, \ |\mathcal{S}| \leq K \end{aligned} \tag{6.6}$$

where a_x is the degree of x. In terms of *random walks*, the optimization of Eq.(6.6) corresponds to *selecting K nodes as absorbing nodes to maximize the sum of absorbing probabilities of a random walker in a given network $\mathcal{G}$*. In terms of *linear electric circuits*, the first constraint of Eq.(6.6) is the *Kirchhoff equation*, and the problem is locating K voltage sources to maximize the electric potential of the circuit.

[6]Refer to [Grady, 2006; Zhang et al., 2010] for the derivation from Eq.(6.2) to Eq.(6.6).

Since the objective $u(x; \mathcal{S})$ is submodular, we can obtain a near-optimal solution by a greedy algorithm, which starts with an empty $\mathcal{S}$ and iteratively adds the item s_k that maximizes the marginal temperature gain, $U(\mathcal{S}_{k\text{-}1} \cup \{s_k\}) - U(\mathcal{S}_{k\text{-}1})$, as shown in Eq.(6.7). The details of the greedy algorithm will be discussed in Section 6.3.

$$s_k = \underset{x \in \mathcal{V}}{\operatorname{argmax}}\, U(\mathcal{S}_{k\text{-}1} \cup \{x\}) - U(\mathcal{S}_{k\text{-}1}) \quad \text{where } U(\mathcal{S}_k) = \sum_{x \in \mathcal{V}} u(x; \mathcal{S}_k) \tag{6.7}$$

The dissipation conductance z is a parameter to control trade-off between centrality and diversity. With a larger z, the heat loss to the environment is larger as well, and only the neighbors within a shorter range of a source will get high temperatures. Hence, a point to be closer to the already ranked set $S_{k\text{-}1}$ is likely to be chosen as a next s_k.

Fig.6.1 shows two toy examples of diversity ranking and clustering. Here, the location of a point is used as the feature $\mathbf{g}(i)=[x \;\; y]^T$ to compute the similarity of Eq.(6.5). Therefore, a closer point pair (i, j) has a larger diffusivity d_{ij}. In the first example of three Gaussian distributions (Fig.6.1.(a)-(f)), our intuition tells that the center point in the largest blob should be selected as the first item s_1, and it actually has the highest marginal gain in Fig.6.1.(b). In the next iteration, since the points near s_1 already have high temperature, the second choice to maximize the marginal gain should be not only distant enough from s_1 (diversity) but also densely linked by other points with high diffusivity d_{ij} (centrality), which is s_2 in Fig.6.1.(c). In sum, s_k is chosen as the most central but distant enough from already selected items $\mathcal{S}_{k\text{-}1}$.

In the second example of three co-centric circles (Fig.6.1.(g)-(l)), one interesting behavior is that among the points in each circle, the point at the opposite side of the circle to the selected point has the highest marginal gain. Thus, if the fourth s_4 is chosen in Fig.6.1.(k), it is the exact opposite of s_3 in the circle. That is, the largest circle in Fig.6.1.(l) will be divided as two exact half circles with K=4.

This algorithm may seem to be similar to the Grasshopper algorithm [Zhu et al., 2007], a greedy algorithm for diversity ranking. However, the objective function is different, and our main contribution over [Zhu et al., 2007] is that our method is not *ad-hoc*, but a constant-factor approximation based on the submodularity.

6.3 Image CoSegmentation

In this section, we present our scalable cosegmentation algorithm. Below, we begin with the segmentation of a single image to illustrate the basic behavior of the algorithm.

6.3.1 Segmentation of a Single Image

The segmentation of a single image aims to find K segment centers to maximize the sum of segmentation confidence of every pixel in an image. This can be achieved via the following procedure.

Building the intra-image graph of an image: For faster computational speed, we first extract superpixels from an image as shown Fig.6.2.(b). Any edge-preserving superpixel methods can be applied, and TurboPixels [Levinshtein et al., 2009] is used in our implementation. Then we build

Figure 6.2: An example of segmenting a single image. (a) An input image. (b) 1000 super-pixels and colored evaluation locations $\mathcal{L}$. (c) Image segmentation with red boundaries. (d)-(g) Color-coded segmentation outputs by ranging K from 2 to 8. As K increases, the following regions are detected in turn: {*sky*, *tree*, *wall* (center), *roof* (left), *windows* (left), *building* (left), and *trash container*}.

the intra-image graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{D}_i)$ where the vertex set $\mathcal{V}_i$ is the set of superpixels and the edge set $\mathcal{E}_i$ connects all pairs of adjacent superpixels. Let N_i denote the number of superpixels of an image i. In each superpixel, 3-D CIE Lab color and 4-D texture features[7] are extracted. The diffusivity $\mathcal{D}_i$ is computed by Gaussian similarity in Eq.(6.5) on the features of superpixels. The adjacency matrix $\boldsymbol{G}_i$ of $\mathcal{G}_i$ is a sparse $N_i \times N_i$ matrix, in which the number of nonzero elements of each superpixel is the same with the number of its neighbors. In most cases, it is less than 10.

Construction of evaluation set: In the diversity ranking discussed earlier, we compute the marginal gain at every datapoint to find the maximum (Fig.6.1). However, this search is inefficient since the actual distinctive regions in an image are usually much fewer than N_i. For example, in Fig.6.2, there are a lot of *sky* superpixels and there is little difference in the segmentation results no matter which *sky* superpixel is chosen as a segment center. Thus, we first run agglomerative clustering on $\boldsymbol{G}_i$ to find out the set of evaluation points $\mathcal{L}_i$. ($|\mathcal{L}_i| \leq 100$ in our experiments). The marginal gain is only computed at $\mathcal{L}_i$. That is, segment centers are limited to be placed in a subset of $\mathcal{L}_i$. (*i.e.* $\mathcal{S}_i \subset \mathcal{L}_i \subset \mathcal{V}_i$ in the third constraint of Eq.(6.8)). Fig.6.2.(b) shows an example of $\mathcal{L}_i$ as colored superpixels.

Basic behavior of segmentation: In summary, our segmentation algorithm greedily selects the largest and most coherent regions. As shown in Fig.6.2.(d), the sky is first chosen with K=2. As K increases, the regions of the tree, the house in the center, and the building in the left are chosen in the decreasing order of their sizes and coherence in Fig.6.2.(d)-(g). This desirable trend comes from the greedy nature of our algorithm. This behavior is quite helpful for automatic selection of K. We can keep increasing K until the detected segment is not significant any more (*i.e.* temperature increase by adding a new source is not significant any more). As iteration goes, we re-use the previous results of a lower K, which significantly reduce the computation time (*e.g.* the lazy greedy approach in [Leskovec et al., 2007]).

6.3.2 Cosegmentation

The input of cosegmentation is an image set $\mathcal{I}$ and the number of segments K. The optimization formulation for cosegmentation in Eq.(6.8) is an extension of that of the diversity ranking (Eq.(6.6)).

[7] http://www.robots.ox.ac.uk/∼vgg/research/texclass/.

$$\max \sum_{i\in\mathcal{I}}\sum_{x\in\mathcal{V}_i} u_i(x) \tag{6.8}$$

$$\text{s.t. } u_i(x) = \frac{1}{a_x}\sum_{(x,y)\in\mathcal{E}_i} d_{yx}u_i(y) \quad \text{for } a_x = \sum_{(x,y)\in\mathcal{E}_i} d_{yx} + z_x$$

$$u_i(g) = 0, \quad u_i(s_{ik}) = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk}))$$

$$\text{where } s_{ik} \in \mathcal{S}_i \subset \mathcal{L}_i \subset \mathcal{V}_i, |\mathcal{S}_i| \leq K, \quad \text{for } \forall i \in \mathcal{I}.$$

The objective in Eq.(6.8) is the sum of temperature (*i.e.* segmentation confidence) of every image in the dataset. Thus, it encourages each image to be segmented as K largest and most coherent regions that are nevertheless content-wise diverse with respect to one another. In order to enforce inter-image similarity between chosen clusters, the second constraint of Eq.(6.8) is introduced. The $f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk}))$ is an increasing function of the feature affinity between the k-th sources of an image i (s_{ik}) and an image j (s_{jk}). More visually similar the features of s_{ik} and s_{jk} are, a higher value $f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk}))$ has. It is intuitive that the system temperature is linear with the source temperature. (*e.g.* if the source temperature is halved, then the temperatures of all points in the system are halved as well). Hence, the second constraint pushes the k-th source placement of image i to be similar to its corresponding placement in other images of $\mathcal{N}(i)$, which is the neighborhood image set of i to be jointly cosegmented. If $\mathcal{N}(i) = \mathcal{I}\backslash i$, then each image is cosegmented with respect to all the other images in $\mathcal{I}$. Meanwhile, the affinity function f controls how strongly the inter-image similarity is imposed. If $f(\mathbf{g}(s_{ik}), \mathbf{g}(s_{jk}))$ is constant, the optimization of Eq.(6.8) reduces to independent segmentation of each image. Otherwise, if it is a fast increasing function, the inter-image similarity is highly weighted. We use the Gaussian similarity in Eq.(6.5) for f.

Algorithm 7 presents the greedy algorithm to solve Eq.(6.8). Note that Algorithm 7 is easily parallelizable. All steps except step **5** can be computed individually in each image. The computation complexity of step **5** is $O(|\mathcal{I}||\mathcal{N}|)$[8].

Once we obtain K source placement $\mathcal{S}_i$ for each image, the segmentation is straightforward. Here we use the method of [Grady, 2006], which is summarized in step **7**-**8** of Algorithm 7. It first calculates $(N_i - K)\times K$ matrix $\mathbf{X}$ in which $\mathbf{X}(j,k)$ is the probability that a random walker starting at an unselected j-th point (*i.e.* $x_j \in \mathcal{V}_i\backslash\mathcal{S}_i$) reaches the k-th source points. Then, we cluster the superpixels that share the same source point as the most probable destination.

Fig.6.3 shows an example of our cosegmentation on three MSRC cow images with K=4. Since our algorithm can handle arbitrary K, the brown and black cows and the river in the first image can be detected as individual clusters.

Optimality: The constant factor approximation of our algorithm is guaranteed if the element with the maximum marginal gain is chosen in each round (step **5**). In diversity ranking and single-

[8] In our Matlab implementation, the major independent computation, step **3**-**4**, took about 2 second per image of 1,000 superpixels. Step **5** took about 6-8 minutes for 1000 images with full dependency (*i.e.* $|\mathcal{I}|$=1000, $|\mathcal{N}|$=999). The other steps took much less than 1 second per image.

Algorithm 7: CoSand Cosegmentation.

Input: (1) Intra-image matrix $\boldsymbol{G}_i$ for all $I_i \in \mathcal{I}$. (2) Number of segments K. (3) Evaluation set size $|\mathcal{L}|$.

Output: Cluster centers $\mathcal{S}_i$ and segmented images for $I_i \in \mathcal{I}$.

1: **foreach** $I_i \in \mathcal{I}$ **do** $\mathcal{S}_i \leftarrow \emptyset$ **end**
2: **foreach** $I_i \in \mathcal{I}$ **do** $\mathcal{L}_i \leftarrow \mathsf{AggloClust}(\boldsymbol{G}_i, |\mathcal{L}|)$ **end**
while $|\mathcal{S}_i| \leq K$ **do**
  foreach $I_i \in \mathcal{I}$ **do**
    foreach $l_j \in \mathcal{L}_i$ **do**
      3: Solve $\mathbf{u} = \mathbf{L}_i\mathbf{u}$ where $\mathbf{L}_i$ is the Laplacian of $\mathbf{G}_i$ and $\mathbf{u}$ is an $N_i \times 1$ vector with the constraints of $\mathbf{u}(\{\mathcal{S}_i \cup l_j\}) = 1$ and $\mathbf{u}(g) = 0$.
      4: Obtain the gain $\Delta U_i(l_j) = |\mathbf{u}|_1$ (l-1 norm of $\mathbf{u}$).
  5: Solve the energy maximization by belief propagation
  $E(l) = \sum_{i \in \mathcal{I}} \Delta U_i(l_i) \big(\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} f(\mathbf{g}(l_i), \mathbf{g}(l_j)) \big)$. $\{s_1, \cdots, s_{\mathcal{I}}\} \leftarrow \mathrm{argmax}_{l_1, \cdots, l_{\mathcal{I}}} E(l)$.
  6: **foreach** $I_i \in \mathcal{I}$ **do** $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup s_i$ **end**
foreach $I_i \in \mathcal{I}$ **do**
  7: Compute $(N_i - K) \times K$ matrix $\mathbf{X}$ by solving $\mathbf{L}_u\mathbf{X} = -\mathbf{B}^T\mathbf{I}_s$ where if we let $\mathcal{X}_i = \mathcal{V}_i \backslash \mathcal{S}_i$, $\mathbf{L}_u = \mathbf{L}_i(\mathcal{X}_i, \mathcal{X}_i)$, $\mathbf{B} = \mathbf{L}_i(\mathcal{S}_i, \mathcal{X}_i)$, and $\mathbf{I}_s$ is a $K \times K$ identity matrix.
  8: A superpixel $v_j (\in \mathcal{V}_i)$ is clustered $c_j = \mathrm{argmax}_k \mathbf{X}(j, k)$.

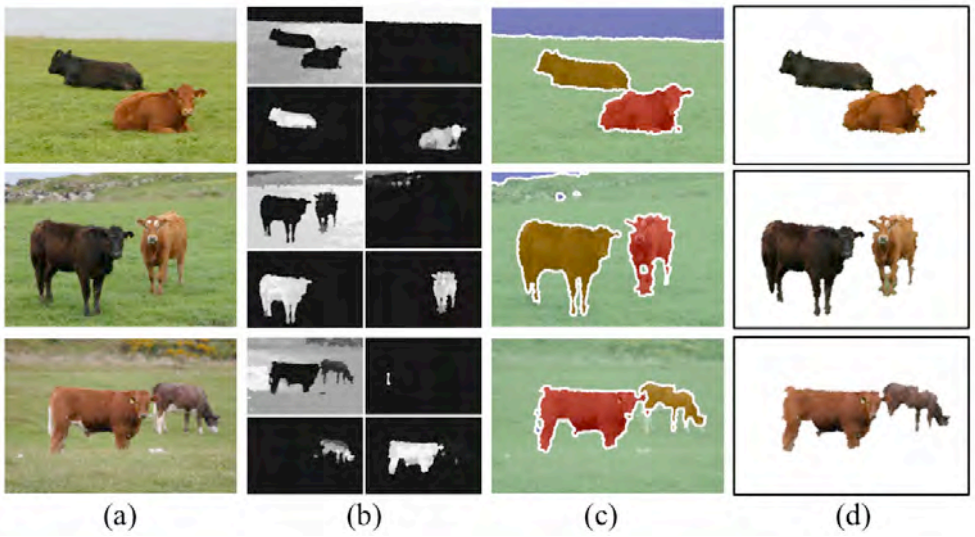


Figure 6.3: An example of cosegmentation on MSRC cow images (M=3, K=4). (a) Input images. (b) Likelihood of each segment from white (high) to black (low). (c) Color-coded cosegmentation outputs. (d) The 3rd and 4th segments from input images.

image segmentation, we compute the exact solution for this step. However, we use belief propagation, which is an approximate maximization, for a large-scale cosegmentation with full dependency. In practice, this relaxed solution is good enough to obtain a high-quality segmentation.

A more scalable setting: In practice, a large-scale image set is likely to contain various noisy information as well. If heterogeneous images are cosegmented, then the results would be worsen than those of individual image segmentation. Thus, one can first decompose $\mathcal{I}$ into disjoint sets $\mathcal{I} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_O$ so that each subset $\mathcal{I}_o$ consists of similar images. Then, Algorithm 7 can be applied to each $\mathcal{I}_o$ separately. This decomposition can be done by the proposed diversity ranking and clustering of Eq.(6.6) on the similarity graph of $\mathcal{I}$, which can be constructed by applying Gaussian similarity to image descriptors (*e.g.* dense SIFT or GIST).

6.4 Experiments

We evaluate our approach with two different experiments: (1) figure-ground segmentation with a pair of images (M=2 and K=2), and (2) scalability tests with a large number of images ($M \sim 1000$). The figure-ground tests are performed to quantitatively compare our method with other state-of-the-art cosegmentation techniques that are only applicable in this setting. The scalability tests evaluate how well our algorithm works with real-world data.

6.4.1 Results on Figure-ground Cosegmentation

In the figure-ground tests, we use MSRC dataset [Winn et al., 2005], which provides 30 pixel-wise labeled images per object. Two recent cosegmentation methods, [Hochbaum and Singh, 2009]

Class (%)	Ours	(Co-Seg)	(DC)
Aeroplane	**37.6** ± 10.6	25.6 ± 9.9	26.5 ± 7.9
Bike	**68.4** ± 12.6	66.8 ± 13.9	58.4 ± 11.6
Bird	**57.0** ± 18.2	30.4 ± 19.3	50.3 ± 19.2
Car	**57.7** ± 9.4	55.8 ± 16.6	52.5 ± 13.3
Cat	73.1 ± 12.2	**75.9** ± 16.9	65.6 ± 13.9
Chair	**64.4** ± 12.6	62.2 ± 21.8	61.6 ± 15.4
Cow	66.1 ± 18.5	**72.4** ± 11.9	67.3 ± 11.9
Dog	**55.5** ± 3.9	47.7 ± 18.9	48.3 ± 22.9
Face	**78.5** ± 11.4	72.1 ± 18.4	60.9 ± 12.0
Flowers	**75.6** ± 2.2	70.0 ± 14.44	71.6 ± 16.4
Sheep	69.2 ± 16.6	43.7 ± 19.3	**70.5** ± 16.1
Sign	**68.7** ± 12.9	58.8 ± 17.9	64.1 ± 17.5
Tree	**67.6** ± 1.1	60.2 ± 13.0	60.8 ± 13.1

(a)

Class (%)	Ours	(MNcut)	(LDA)
Ban spider	**48.6** ± 24.1	35.3 ± 13.0	32.4 ± 10.0
Hognose	**55.3** ± 22.0	47.2 ± 17.0	44.7 ± 17.1
Coral	**79.3** ± 20.1	66.4 ± 22.0	52.6 ± 14.7
St Bernard	**68.2** ± 21.3	50.5 ± 13.7	45.7 ± 12.3
Basenji	**58.8** ± 23.1	46.3 ± 15.8	42.2 ± 14.9
Tabby	**67.2** ± 22.1	51.3 ± 16.6	49.6 ± 14.6
Jaguar	**67.8** ± 21.0	50.2 ± 14.7	49.4 ± 14.5
Lion	**63.6** ± 22.4	50.7 ± 17.7	47.6 ± 16.8
Starfish	**50.2** ± 25.9	41.6 ± 18.7	40.1 ± 16.4
Polecat	**58.3** ± 21.5	47.6 ± 15.7	44.7 ± 13.4
Badger	**51.6** ± 24.6	43.0 ± 17.9	41.3 ± 16.3
Orangutan	**61.3** ± 26.0	49.5 ± 19.8	48.0 ± 18.3
Guenon	**58.8** ± 24.8	47.8 ± 16.9	46.4 ± 16.2

(b)

Table 6.2: Cosegmentation accuracies. (a) Comparison between our method and baselines (Co-Seg) [Hochbaum and Singh, 2009] and (DC) [Joulin et al., 2010] for figure-ground cosegmentation for 100 random pairs of images per object from the MSRC dataset. (b) Comparison between our method and baselines (MNcut)[Cour et al., 2005] and (LDA)[Russell et al., 2006] for scalable cosegmentation with 13 selected synsets from the ImageNet dataset. Synset Wordnet IDs = {Ban spider (n01773549), Hognose snake (n01729322), Coral (n09256479), St Bernard (n02109525), Basenji (n02110806), Tabby (n02123045), Jaguar (n02128925), Lion (n02129165), Starfish (n02317335), Polecat (n02443114), Badger (n02447366), Orangutan (n02480495), Guenon monkey (n02484975)}.

Figure 6.4: Four cosegmentation examples on the MSRC dataset. (a) Pairs of input images. (b) Our cosegmentation results with K=8. The cosegmented pairs are presented by the same colors. Some segments are too small to be shown. (c) Figure-ground segmentation results that are induced from the eight pairs of cosegments.

and [Joulin et al., 2010], are compared using the original authors' implementation with the default parameter setting[9]. We run [Hochbaum and Singh, 2009], [Joulin et al., 2010], and our method on randomly generated 100 pairs in each class.

Unlike the others, the method of [Hochbaum and Singh, 2009] requires priori labels of foreground (*fg*) and background (*bg*) RGB colors. In order to obtain labels, we fist identify the *fg* and *bg* regions of each image from the groundtruth. Then, we apply K-means to the RGB space of *fg* and *bg* pixels to compute three cluster centers each, which are used as labels (*i.e.* total 6 *fg* and 6 *bg* RGB labels in each pair). These labels can be regarded as strong supervision, but they were used because the performance of [Hochbaum and Singh, 2009] was highly sensitive to the labels.

Since our method is not designed to aim at figure-ground segmentation, we add an additional step to generate the binary segmentation results. Our approach iteratively chooses large and coherent regions across input images in a bottom-up way. Thus, if the foreground object consists of several distinct regions, it is likely to segment them into multiple regions. For binary segmentation, we first safely cosegment a pair of images with a large K (K=8 in our experiments). Then, we apply Normalized cuts to the similarity graph of eight pairs of cosegments to obtain two balanced and discriminative partitions. We observed that our approach showed excellent performance for detecting a moderate number of cosegments but the final figure-ground segmentation accuracy was dependent much on this binarization.

Table 6.2.(a) summarizes the segmentation accuracies on the random test pairs of MSRC dataset. The accuracy is measured by the intersection-over-union metric that is a standard in PASCAL challenges (*i.e.* For each image, $Ac = \frac{GT_i \cap R_i}{GT_i \cup R_i}$). We observed that our method outperformed both [Hochbaum and Singh, 2009] and [Joulin et al., 2010] in most objects of the MSRC dataset. Our algorithm was also significantly faster than both competitors; it took less than 10 seconds for a pair of images with a [320×213] dimension, 750 superpixels, and K=8.

[9]Codes are available at [Hochbaum and Singh, 2009]: http://www.biostat.wisc.edu/∼vsingh/, [Joulin et al., 2010]: http://www.di.ens.fr/∼joulin/.

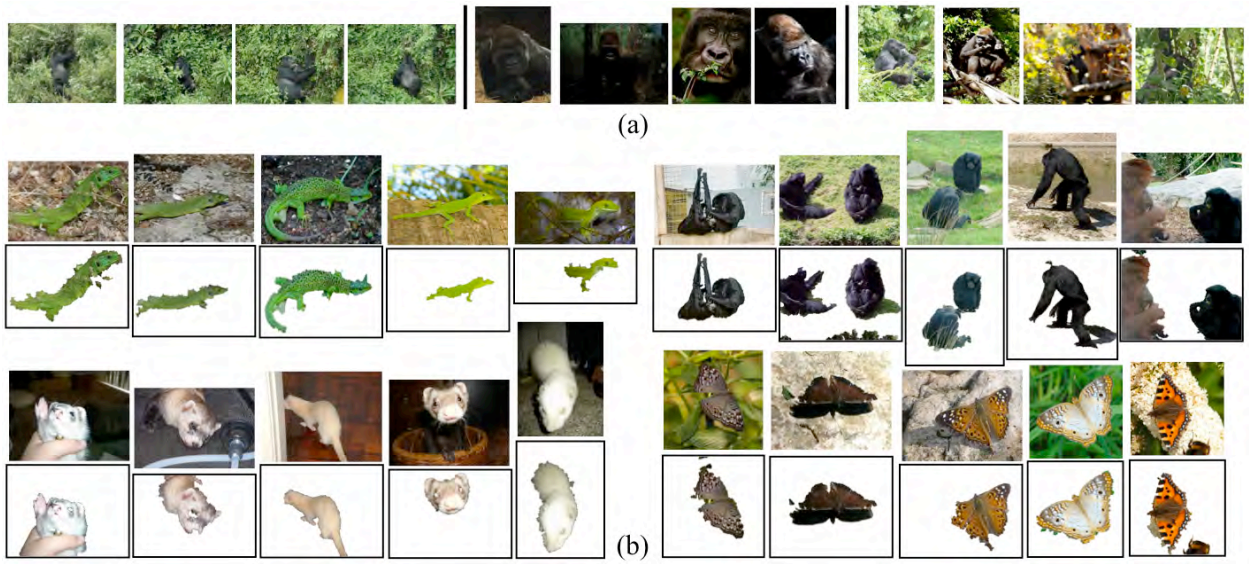


Figure 6.5: Examples of scalable cosegmentation on the ImageNet dataset. (a) Decomposition of the *Gorilla* Synset by the proposed diversity ranking and clustering. Three cluster centers and their three closest images are shown. (b) Examples of cosegmentation on *green lizard*, *siamang*, *ferret*, and *nymphalid butterfly*. In each set, 20∼60 images are simultaneously cosegmented and five selected images are shown.

6.4.2 Results on Scalable Cosegmentation

For scalability tests, we use ImageNet[10] [Deng et al., 2009]. We compute segmentation accuracies by using its bounding box annotations. The bounding boxes may not be a perfect groundtruth for segmentation evaluation, but in practice it is difficult to obtain pixel-wise labels for large-scale datasets.

We compare our algorithm to (`MNcut`) [Cour et al., 2005] and the method of [Russell et al., 2006], which are publicly available[11]. As a baseline, the (`MNcut`) [Cour et al., 2005] independently segments each image with K=2 and the *fg* and *bg* are assigned so that the segmentation accuracy is maximized. For [Russell et al., 2006], we apply the algorithm several times by changing the number of topics from two to eight, and the best results are reported. Note that most previous cosegmentation methods including [Hochbaum and Singh, 2009] and [Joulin et al., 2010] cannot run well with a large number of images. ([Joulin et al., 2010] reported that their algorithm took between 4 and 9 hours for 30 images).

For ImageNet tests, we select 50 synsets that provide bounding box labels. We randomly select up to 1000 images per synset. Since the ImageNet images are too diverse to be jointly cosegmented at once, we first split each synset into 100 disjoint sets $\mathcal{I} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_{100}$ by our diversity ranking and clustering. Then, our cosegmentation is separately applied into each $\mathcal{I}_o$. This decomposition is much more favorable for the performance. We tested a simultaneous cosegmentation of all 1,000

[10] http://www.image-net.org/challenges/LSVRC/2010.

[11] Codes are available at [Cour et al., 2005]: http://www.seas.upenn.edu/∼timothee, [Russell et al., 2006]: http://www.cs.washington.edu/homes/ bcr/projects/mult_seg_discovery/

images with full dependency, in which both accuracy and speed were much worse.

Fig.6.5.(a) shows an example of synset decomposition. A single synset has several different aspects, which were successfully detected by our diversity ranking and clustering. Table 6.2.(b) shows the segmentation accuracies for 13 selected synsets. Our algorithm significantly outperformed the two competitors by more than 10%. Our algorithm took 60-70 minutes for 1,000 images on a single machine. Note that this computation time can be significantly reduced by parallelization as discussed in section 6.3.2.

Fig.6.4 and Fig.6.5.(b) show some examples of cosegmentation on the MSRC and ImageNet datasets. We made two interesting observations here: (i) Our method can easily segment multiple instances in the images. (ii) Our algorithm is robust against an incorrect selection of K. In the duck example of the second column of Fig.6.4, the best choice of K would be four, but a faulty guess with $K{=}8$ did little harm. The four significant segments are successfully detected (*e.g.* three ducks and grass) and the other four overestimated segments were trivially selected as tiny dots.

6.5 Summary

In summary, the main contributions of this chapter are as follows.

- We propose a diffusion-based optimization framework that is applicable to a wide range of computer vision problems. In this work, we show that our optimization leads to an effective solution to diversity ranking, single-image segmentation, and cosegmentation.
- We prove that the *temperature* of a linear anisotropic diffusion system, which corresponds to many important objectives in computer vision tasks, including the cosegmentation score concerned in this work, is a submodular function. This is a new result that widens the applicability of submodular optimization in computer vision research.
- We present CoSand, a distributed cosegmentation exploiting the submodularity of our diffusion-inspired segmentation objective. As compared in Table 6.1, our approach has some unique benefits including compelling performance over previous methods, superior scalability, and a desirable ability of automatically deciding the number of segments.

Chapter 7

Multiple Foreground Cosegmentation

7.1 Introduction

In this chapter, we discuss the cosgementation algorithm in the same line with the previous chapter. However, we aim at more practical approach to be applicable to the photo sets of general users by overcoming some limitations from which existing cosegmentation methods still suffer. The arguably most limiting one is that every input image would need to contain all the foregrounds for the cosegmentation algorithms to be applicable. Fig.7.1 shows a typical example that violates this condition. This is an *apple+picking* photo stream downloaded from Flickr, and it follows an ordinary photo-taking pattern of a general photographer: a series of pictures about a specific event are taken; the number of objects in a photo stream is finite, but they do not appear in every single image. For example, in Fig.7.1, two girls, one baby, and an apple bucket repeatedly appear in the photo stream, but each image includes only an unknown subset of them. Such a *content-misaligned* set of images would not be correctly addressed by existing cosegmentation algorithms. The objective functions in most existing methods were built on the assumption that all input images contain the same objects, without explicitly considering the cases where foregrounds irregularly occur across the images. In order to apply a traditional cosegmentation method to such a photo set, a user is required to first divide her photo stream into several groups so that each group contains only photos that have the same foregrounds. This manual preprocessing can be cumbersome,

Figure 7.1: Motivation for multiple foreground cosegmentation. (a) Input images are 20 photos of an *apple+picking* photo stream of Flickr. Two girls, one baby, and an apple bucket repeatedly occur in the images, but only a subset of them is shown in each image. (b) The first row shows the color-coded cosegmentation output in which the same colored regions are identified as the same foreground. The second row shows the segmented foregrounds.

especially when the number of photos is very large (*e.g.* hundreds or more).

In this chapter, we propose a combinatorial optimization method, MFC, for cosegmentation that does not suffer from the aforementioned restriction. It allows irregularly occurring multiple foregrounds with varying contents to be present in the image collection, and directly cosegment them. More precisely, we consider the following task:

Definition 1 (Multiple Foreground Cosegmentation). *The multiple foreground cosegmentation (MFC) refers to the task of jointly segmenting K different foregrounds $\mathcal{F}=\{\mathcal{F}^1, \cdots, \mathcal{F}^K\}$ from M input images, each of which contains a different unknown subset of K foregrounds.*

Given the number of foregrounds K and an input image set, our approach automatically finds the most frequently occurring K foregrounds across the image set. Optionally, a user may select the example foregrounds of interest in a couple of images in the form of bounding boxes or pixel-wise annotations. Subsequently, our algorithm segments out every instance of K foregrounds in the input image set.

More specifically, our approach is based on an iterative optimization procedure that alternates between two subtasks: *foreground modeling*, and *region assignment*. Given an initialization for the regions of K foregrounds, the foreground modeling step learns the appearance models of K foregrounds and the background, which can be accomplished by using any existing advanced region classifiers or their combinations. During the region assignment step, we allocate the regions of each image to one of K foregrounds or the background. This is done via a combinatorial auction style optimization algorithm; every foreground and the background bid the regions along with their values of how much the regions are relevant to them. These values are computed by the learned foreground models. Finally, an optimal solution (*i.e.* the allocation of the regions that maximizes the overall value) is achieved in $O(MK)$ time, by leveraging the fact that the candidate regions bidden by foregrounds and the final region assignment can be represented by subtrees of a connectivity graph of regions in the image space. Iteratively, after the region assignment, each foreground model is updated by learning from the newly assigned segments (i.e., regions) to the foreground.

The concept of such an iterative segmentation scheme has been used in some previous work such as [Kim and Torralba, 2009] and [Rother et al., 2004]. But the allowance of arbitrary classifiers and their combinations to be plugged in during foreground modeling, and the use of a linear-time algorithm motivated by combinatorial auction for region assignment make our method unique and far more efficient and flexible than earlier ones.

We test our method on a newly created benchmark dataset, called FlickrMFC, with pixel-level groundtruth. Each group consists of photos from a Flickr photo stream taken by a single user, and contains a finite number of subjects that irregularly appear across the images. Our experiments in Section 7.4 show that our approach successfully solves the multiple foreground cosegmentation in a scalable way. Moreover, the cosegmentation accuracies are compelling over the state-of-the-art techniques [Joulin et al., 2010; Kim et al., 2011; Russell et al., 2006] on our novel FlickrMFC dataset and the standard ImageNet dataset [Deng et al., 2009].

To conclude the introduction, we present Table 7.1 that summarizes the comparison of our work with previous cosegmentation methods, as done in previous chapter. Our approach has several important features that are beneficial for the cosegmentation of general users' photo sets. Our algorithm is able to handle a large M for scalability and an arbitrary K for highly variable contents

Methods	M	K+1	MFC	Hetero-FG
Ours (MFC)	$\geq 10^3$	Any	O	O
SO [Kim et al., 2011]	$\geq 10^3$	Any	X	X
UGC [Rother et al., 2006; Vicente et al., 2010]	2	2	X	O
SGC [Batra et al., 2011; Mukherjee et al., 2011] [Hochbaum and Singh, 2009]	≤ 30	2	X	O
DC [Joulin et al., 2010]	≤ 30	2	X	O

Table 7.1: Comparison of our algorithm with previous cosegmentation methods. M and K denote the number of images and foregrounds, respectively. *MFC* indicates whether an algorithm is designed to solve the MFC problem in Definition 1. *Hetero-FG* means whether an algorithm can identify a heterogeneous object (*e.g.* a person) as a single foreground. (SO: submodular optimization, UGC: Graph-cuts (unsupervised), SGC: Graph-cuts (supervised), DC: Discriminative clustering).

of user images. This advantage is also shared with *CoSand* [Kim et al., 2011] in previous chapter, but the key differences are as follows. First, the *CoSand* is a bottom-up approach that relies on only low-level color and texture features, whereas our technique can be merged with any region classification algorithms. Second, the *CoSand* cannot model a heterogeneous object that consists of multiple distinctive regions (*e.g.* a person) as a single foreground. It can be a limitation to be used for consumer photos because they are likely to contain persons as subjects, which are often required to be segmented as a single foreground. However, our approach does not suffer from these issues. In conclusion, our approach can correctly account for multiple foreground cosegmentation in Definition 1, which has not been explicitly addressed by the optimization methods of most previous work [Batra et al., 2011; Hochbaum and Singh, 2009; Joulin et al., 2010; Kim et al., 2011; Mukherjee et al., 2011; Rother et al., 2006; Vicente et al., 2010], as shown in Table 7.1.

7.2 Problem Formulation

We denote the set of input images by $\mathcal{I} = \{I_1, \cdots, I_M\}$. According to Definition 1, we are interested in segmenting out K different foregrounds $\mathcal{F}=\{\mathcal{F}^1, \cdots, \mathcal{F}^K\}$ from all images in $\mathcal{I}$, each with an unknown subset of $\mathcal{F}$. Our algorithm deals with two different scenarios. In the *unsupervised* scenario, a user solely inputs the number K, and our algorithm automatically infers K distinctive foregrounds that are most dominant in $\mathcal{I}$. In the *supervised* scenario, a user can provide bounding-box or pixel-wise annotations for K foregrounds of interest in some selected images.

In our approach, we break the MFC problem defined above into two subproblems, which we solve iteratively: *foreground modeling* and *region assignment*. Foreground modeling learns the appearance models of K foregrounds or background, and region assignment allocates the regions of each image to one of K foregrounds or the background. Intuitively, given a solution to one of the two subproblems, the other is solvable. From an initial region assignment, one can learn $K+1$ foreground models, which in turn improve region assignment in every image. These two processes alternate until achieving a converging solution.

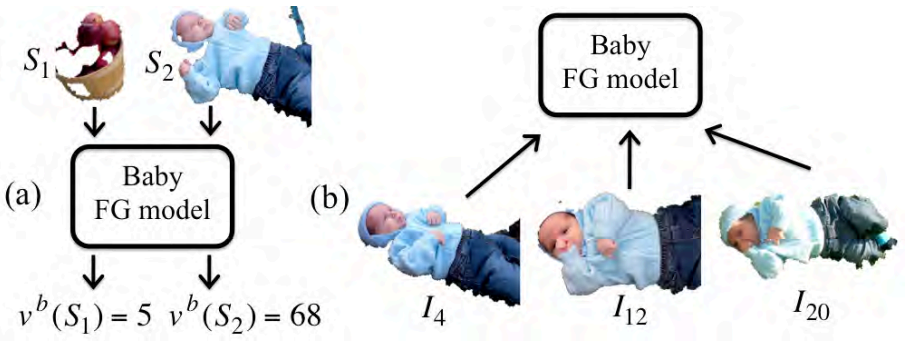


Figure 7.2: An example of the *baby* foreground (FG) model. (a) A FG model is a parametric function that maps any region to a value to the foreground. (b) After the region assignment, the FG model is updated by learning from the segments assigned to the FG.

7.2.1 Foreground Models

Without loss of generality, we define the k-th foreground (or the background) model as a parametric function $v^k : \mathcal{S} \rightarrow \mathbb{R}$ that maps any region $S \in \mathcal{S}$ in an image to its fitness value to the k-th foreground (*i.e.* how closely the region is relevant to the k-th foreground). If an image I_i is over-segmented as $\mathcal{S}_i$, then $v^k : 2^{|\mathcal{S}_i|} \rightarrow \mathbb{R}$ takes any subset $S \subset \mathcal{S}_i$ as input and returns its value to the k-th foreground. During the region assignment, each foreground model assesses how fit a region (or a set of regions) to the foreground, as shown in Fig.7.2.(a). During the foreground modeling, each foreground model is updated by learning from the segments allocated to the foreground, as shown in Fig.7.2.(b).

One important objective of our approach is to enable adaptability to any choice or combination of foreground models as plug-ins. Any classifiers or ranking algorithms can be used as foreground models so long as they can evaluate a region and be updated by learning from the assigned regions. (If we view the foreground model as a classifier, the former is a testing step and the latter is a training step). In this work, we use two different foreground models - the Gaussian mixture model (GMM) (*i.e.* Boykov-Jolly model [Boykov and Jolly, 2001; Rother et al., 2004]) and spatial pyramid matching (SPM) with linear SVM [Lazebnik et al., 2006]. The former has been a popular appearance model in cosegmentation [Batra et al., 2011; Vicente et al., 2010], and the latter is one of baselines for object classification and detection. Table 7.2 summarizes the region descriptors, model parameters, learning methods, and region valuation of the two foreground models. For both GMM and SPM models, we follow the algorithms proposed in the original papers [Boykov and Jolly, 2001; Rother et al., 2004] and [Lazebnik et al., 2006]. In experiments, the final region score is computed by $v^k(S) = \alpha \cdot v^k_{GMM}(S) + (1 - \alpha) \cdot v^k_{SPM}(S)$ by changing α from 0 to 1. Note that thanks to our flexible definition of the foreground model, the simple SPM model can be replaced by the state-of-the-arts deformable part models [Felzenszwalb et al., 2010] for better performance.

7.2.2 Region Assignment

Given the foreground models, the region assignment is performed on individual images separately. The goal of this step is to divide $\mathcal{S}_i$ (*i.e.* the segment set of each image I_i) into disjoint subsets

	GMM	SPM
Region features	A set of RGB colors extracted at every pixel of region S.	A spatial pyramid $h(S)$ (2 levels, 200 visual words of gray/HSV SIFT). The minimum rectangle enclosing S is used as the based pyramid.
Model and learning	A Gaussian mixture with C components. Parameters $\theta^k = \{\pi_c^k, \mu_c^k, \sigma_c^k\}_{c=1}^C$ are the prior probability, mean, and covariance. The standard EM is used for learning.	A linear SVM is learned using $\mathcal{F}^k$ as positive data and randomly chosen regions from other foregrounds or background as negative data.
$v^k(S)$	The mean log-likelihood of the RGB descriptors of S to the k-th learned GMM models.	$v^k(S) = \sum_{t=1}^T y_t \alpha_t K(h(S), h(t))$ where $h(t)$ is the histogram of training region t, $y_t \in \{+1, -1\}$ is the positive/negative label, $K(\cdot,\cdot)$ is the histogram intersection kernel, α_t is the weight of the support vector for t, and T is the number of training regions.

Table 7.2: Description of two foreground models – GMM and SPM models.

of foregrounds $\mathcal{F}_i^k$ ($k = \{1, \cdots, K\}$) and background (For notational simplicity, we use $\mathcal{F}_i^{K+1}$ for background). Since all foregrounds do not appear in every image, some foregrounds ($\mathcal{F}_i^k$) are empty sets.

Naively, we may distribute each segment $s \in \mathcal{S}_i$ to one of $\mathcal{F}_i^k$ that has the maximum value $v^k(s)$ for it. However, in image segmentation, the value of a segment bundle (*i.e.* a subset of $\mathcal{S}_i$) can be worth more than or less than the sum of values of individual segments. For example, suppose that a black patch is the most valuable to the *cow* foreground. But, if the black patch is combined with a skin-colored patch, this bundle would be more valuable to the *person* foreground than to the *cow* foreground.

Consequently, the region assignment reduces to finding a disjoint partition $\mathcal{S}_i = \bigcup_{k=1}^{K+1} \mathcal{F}_i^k$ with $\mathcal{F}_i^k \cap \mathcal{F}_i^l = \emptyset$ if $k \neq l$, to maximize $\sum_{k=1}^{K+1} v_k(\mathcal{F}_i^k)$. More formally, it corresponds to the integer program (IL) problem below:

$$\begin{aligned} \max \quad & \sum_{k=1}^{K+1} \sum_{S \subseteq \mathcal{S}_i} v^k(S) x^k(S) \\ \text{s.t.} \quad & \sum_{k=1}^{K+1} \sum_{s \in S, S \subseteq \mathcal{S}_i} x^k(S) \leq 1, \quad \forall s \in \mathcal{S}_i, \\ & x^k(S) \in \{0, 1\} \end{aligned} \tag{7.1}$$

where variables $x^k(S)$ describe the allocation of bundle S to k-th foreground $\mathcal{F}_i^k$. (*i.e.* $x^k(S) = 1$ if and only if the k-th foreground takes the bundle S). The first constraint checks whether the assignment is feasible; any segment $s \in S_i$ cannot be assigned more than once.

The region assignment in Eq.(7.1) requires to check all possible subset $S \subseteq \mathcal{S}_i$. Unfortunately, there are $2^{|\mathcal{S}_i|}$ possible subsets, so enumerating them is infeasible. It is proven in [Cramton et al., 2005] that Eq.(7.1) is identical to the weighted set packing problem, and thus it is NP-complete and inapproximable.

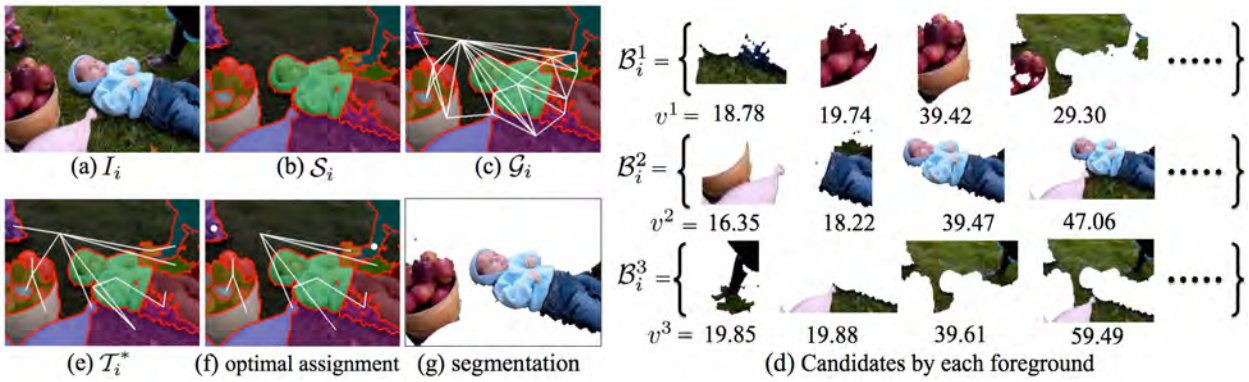


Figure 7.3: An example of region assignment with *apple bucket* and *baby* foregrounds (FG) and background (BG). (a) An input image I_i. (b) Segment set $\mathcal{S}_i$. (c) Adjacency graph $\mathcal{G}_i$. (d) The set of FG candidates $\mathcal{B}_i$ that are submitted by two FGs and BG. Each candidate is a subtree of $\mathcal{G}_i$, associated with its value. (e) The most likely tree $\mathcal{T}_i^*$ given $\mathcal{B}_i$. (f) The optimal assignment is a forest of subtrees in $\mathcal{B}_i$. (g) The segmentation of two FGs.

7.3 Tractable Multiple Foreground Cosegmentation

In this section, we propose a tractable MFC method that iteratively solves the two subproblems defined in the previous section. The foreground modeling is straightforward, but the region assignment is intractable. Hence, we here focus on developing a polynomial time algorithm to solve the region assignment by taking advantage of structural properties that are commonly observed in the image space.

7.3.1 Tree-Constrained Region Assignment

Given the $K{+}1$ foreground models, the region assignment module progresses as follows. First, each image I_i is oversegmented as $\mathcal{S}_i$ as shown in Fig.7.3.(b). Any segmentation algorithm can be used, and we apply the submodular image segmentation [Kim et al., 2011] to each image. Given the segment set $\mathcal{S}_i$ of image I_i, each foreground in $\mathcal{F}$ creates a set of *foreground candidates* $\mathcal{B}_i^k = \{B_1^k, \cdots, B_n^k\}$, where every candidate is a tuple $B_j^k = \langle k_j, C_j, w_j \rangle$, where k_j is the index of the foreground that submits candidate j, $C_j \subseteq \mathcal{S}_i$ is a bundle of segments and w_j is its value $w_j{=}v^k(C_j)$ (See an example in Fig.7.3.(d)). In this step, we allow each foreground to submit as many candidates as it is willing to take (Section 7.3.2). Finally, solving the region assignment in Eq.(7.1) corresponds to choosing some feasible foreground candidates among all submitted $\mathcal{B}_i{=}\{\mathcal{B}_i^1, \cdots, \mathcal{B}_i^{K+1}\}$ in order to maximize the overall values[1] (Section 7.3.3).

There are two possible approaches to make the region assignment problem in Eq.(7.1) tractable: putting a restriction on value function v^k or a restriction on generating foreground candidates $\mathcal{B}_i$.

[1]Our region assignment is closely related to combinatorial auction [Cramton et al., 2005] with following terminological correspondences: Given a set of segments (items) $\mathcal{S}_i$, $K{+}1$ foreground models (bidders or buyers) submit a set of foreground candidates (package bids) $\mathcal{B}_i$. The region assignment in Eq.(7.1) is commonly referred to a *Winner determination problem* or a *Welfare problem* in combinatorial auction literature.

We explore the latter approach (*i.e.* restriction on $\mathcal{B}_i$) because one of our design goals is to enable flexible choice of foreground models. (*e.g.* it is hard to define any regularity constraints on the output scores of the SPM model for arbitrary segment bundles). In the following sections, we will discuss how to achieve the tractability.

Assumption: We assume that a foreground instance in an image is represented by a set of *adjacent* segments. A pair of segments is considered as adjacent if its minimum spatial distance in an image is less than or equal to ρ. This is a reasonable assumption because most foregrounds of interest occupy connected regions in an image. Our approach allows multiple instances (*e.g.* several apple buckets in an image), which are regarded as multiple connected regions.

Suppose that we build an adjacency graph $\mathcal{G}_i{=}(\mathcal{S}_i, \mathcal{E}_i)$ where every segment is a vertex and $(s_l, s_m) \in \mathcal{E}_i$ if $\min d(s_l, s_m) \leq \rho$ (*e.g.* $\rho{=}5$ pixels) for all $s_l, s_m{\in}\mathcal{S}_i$ (See an example in Fig.7.3.(c)). Then, any connected regions in the image can be represented by subtrees of $\mathcal{G}_i$, and thus the final region assignment $\{\mathcal{F}_i^1, \cdots, \mathcal{F}_i^{K+1}\}$ should be a forest (*i.e.* set) of subtrees (See an example in Fig.7.3.(f)). Consequently, without loss of generality, we restrict any foreground candidate $B_i \in \mathcal{B}_i$ to be a subtree of the $\mathcal{G}_i$, and our goal of region assignment is to select some B_i that are not only feasible but also maximize the objective of Eq.(7.1).

7.3.2 Generating Candidate Sets

In this section, we discuss how each foreground generates a set of foreground candidates $\mathcal{B}_i^k$, each of which is a subtree of $\mathcal{G}_i$ (*i.e.* generating candidates in Fig.7.3.(d) from $\mathcal{G}_i$ of Fig.7.3.(c)). In this step, each foreground does not care for the winning chances of its proposals by competing the ones submitted by the other foreground models.

Given the adjacency graph $\mathcal{G}_i$, each foreground samples highly valued subtrees as candidates $\mathcal{B}_i^k$ by using beam search with v^k as a heuristic function and a beam width D [Russell and Norvig, 2009] (*e.g.* $D{=}10$ in our tests). Algorithm 8 summarizes this process. We start with all unit segments $\forall s \in \mathcal{S}_i$ to be added to $\mathcal{B}_i^k$. In every round, we enumerate all subtrees that can be obtained by adding one edge from previous candidates. The beam width D specifies the maximum number of subtrees to be retained at each round. We only keep top D highly valued subtrees as $\mathcal{B}_i^k$ without consuming too much time on poorly valued ones (See step 3 of Algorithm 8). In practice, this beam search selects good and sufficiently many candidates, because each foreground usually occupies only a part of an image. The computation time of this step per foreground is at most $O(D|\mathcal{S}_i|^2)$, and the number of foreground candidates $|\mathcal{B}_i|$ is at most $(D|\mathcal{S}_i|)$.

7.3.3 Tractable Region Assignment

Given $\mathcal{B}_i$, we are ready to solve Eq.(7.1) by choosing some feasible candidates among $\mathcal{B}_i$. For a tractable solution, we first introduce a theorem in [Sandholm and Suri, 2003], which is reformulated to be fit to our context as follows.

Theorem 3 ([Sandholm and Suri, 2003]). *Dynamic programming is able to solve Eq. (7.1) in $O(|\mathcal{B}_i||\mathcal{S}_i|)$ worst time if every candidate in $\mathcal{B}_i$ can be represented by a connected subgraph of a tree $\mathcal{T}_i^*$.*

Algorithm 8: Build candidates $\mathcal{B}_i^k$ from $\mathcal{G}_i$ by beam search.

Input: (1) Adjacency graph $G_i = (\mathcal{S}_i, \mathcal{E}_i)$. (2) Value function v^k of the k-th foreground model. (3) D: Beam width.

Output: k-th foreground candidates $\mathcal{B}_i^k$.

1: Set the initial open set to be $\mathcal{O} \leftarrow \forall s \in \mathcal{S}_i$. $\mathcal{B}_i \leftarrow \forall s \in \mathcal{S}_i$.

for $i = 1$ **to** $|\mathcal{S}_i| - 1$ **do**

  foreach $o \in \mathcal{O}$ **do**

    2: Enumerate all subgraphs $\mathcal{O}_o$ that can be obtained by adding an edge to o. $\mathcal{O} \leftarrow \mathcal{O}_o$ and $\mathcal{O} \leftarrow \mathcal{O} \backslash o$.

  3: Compute values $v_o \leftarrow v^k(o)$ for all $o \in \mathcal{O}$ and remove o from $\mathcal{O}$ if it is not top D highly valued. $\mathcal{B}_i \leftarrow \mathcal{O}$.

Theorem 3 suggests a linear-time algorithm for region assignment, if $\mathcal{B}_i$ can be organized as a tree. In the foreground candidate set $\mathcal{B}_i$, each $B_i \in \mathcal{B}_i$ is a subtree but its aggregation $\mathcal{B}_i$ may not. Therefore, we reject some $\mathcal{B}_i$ that cause cycles but are not highly valued, because the final solution is a forest of candidate subtrees. The pruned $\mathcal{B}_i$ is denoted by $\mathcal{B}_i^*$. Now we discuss how to obtain $\mathcal{T}_i^*$ and $\mathcal{B}_i^*$ from $\mathcal{B}_i$.

Inferring the tree from the candidate set: Given candidate set $\mathcal{B}_i$ (*i.e.* a set of subtrees) of image i, our objective here is to infer the most probable tree $\mathcal{T}_i^*$. It can be formulated as the following maximum likelihood estimation (MLE) in a similar way to tree structure learning (*e.g.* Chow-Liu tree [Chow and Liu, 1968]).

$$\mathcal{T}_i^* = \underset{\mathcal{T} \in \mathcal{T}(\mathcal{G}_i)}{\operatorname{argmax}} P(\mathcal{B}_i|\mathcal{T}) \tag{7.2}$$

where $P(\mathcal{B}_i|\mathcal{T})$ is the data likelihood of the given $\mathcal{B}_i$ and $\mathcal{T}(\mathcal{G}_i)$ is the set of all possible spanning trees on $\mathcal{G}_i$. The probability of a candidate set $\mathcal{B}_i$ in tree $\mathcal{T}$ is

$$P(\mathcal{B}_i|\mathcal{T}) = \prod_{B_l \in \mathcal{B}_i} P(B_l|\mathcal{T}) \tag{7.3}$$

where we assume the conditional independence between candidates given the tree $\mathcal{T}$. The probability of observing a candidate in a particular tree structure is defined as:

$$P(B_l|\mathcal{T}) = \prod_{(u,v) \in \mathcal{T}} \exp\left(P_{B_l}(u,v)\right) = \prod_{(u,v) \in \mathcal{T}} \exp\left(P(u,v) \cdot \delta((u,v) \in B_l)\right) \tag{7.4}$$

where the $P(u,v)$ is the probability of an edge between u and v and $\delta((u,v) \in B_l)$ is an indicator whether the edge (u,v) is in the B_l or not. Hence, from Eq.(7.3) and Eq.(7.4), the log-likelihood $\mathcal{L}$ is defined as follows.

$$\mathcal{L} = \sum_{B_l \in \mathcal{B}_i} \log P(B_l|\mathcal{T}) = \sum_{B \in \mathcal{B}_i} \sum_{(u,v) \in \mathcal{T}} \delta((u,v) \in B_l) \log P(u,v) \tag{7.5}$$

Algorithm 9: Infer the most probable $\mathcal{T}_i^*$ from $\mathcal{B}_i$

Input: (1) Candidate set $\mathcal{B}_i$ ($B_l = \langle k_l, C_l, w_l \rangle$ where C_l is a subtree of $\mathcal{G}_i$ and w_l is the value to its foreground).

Output: (1) Candidate tree $\mathcal{T}_i^*$ and (2) Pruned $\mathcal{B}_i^*$ ($\subset \mathcal{B}_i$).

1: Set $\mathbf{A}$ be an $N \times N$ zero matrix where $N = |\mathcal{S}_i|$. Set $\mathcal{B}_i^* \leftarrow \emptyset$.
foreach $B_l = \langle k_l, C_l, w_l \rangle \in \mathcal{B}_i$ **do**
  foreach $s \in \mathcal{C}_l$ **do**
    foreach $t \in \mathcal{C}_l, t \neq s$ **do** $\mathbf{A}(s,t) \leftarrow \mathbf{A}(s,t) + w_l$ **end**
2: Let $\mathcal{T}_i^*$ be the maximum spanning tree of $\mathbf{A}$.
foreach $B_l = \langle k_l, C_l, w_l \rangle \in \mathcal{B}_i$ **do**
  if all edges $(u,v) \in C_l$ is in $\mathcal{T}_i^*$ **then** $\mathcal{B}_i^* \leftarrow B_l$. **end**

Note that the sample proportions are the maximum likelihood estimates of the parameters of discrete distributions.

$$\widehat{P}_{ML}(u,v) = \widetilde{P}(u,v) \propto \exp\Big(\frac{\sum_{B_l \in \mathcal{B}_i} w_l \delta((u,v) \in B_l)}{\sum_{B_l \in \mathcal{B}_i} w_l}\Big) \tag{7.6}$$

Therefore, from Eq.(7.5) and Eq.(7.6), the maximum likelihood is as follows by maximizing over the parameters for a fixed structure:

$$\mathcal{L}^* = \sum_{(u,v) \in \mathcal{T}} \sum_{B \in \mathcal{B}_i} \delta((u,v) \in B_l) \log \widetilde{P}(u,v) \propto \sum_{(u,v) \in \mathcal{T}} \sum_{B_l \in \mathcal{B}_i} w_l \delta((u,v) \in B_l). \tag{7.7}$$

As shown in Eq.(7.7), the likelihood of a tree is the sum of the weight values associated with the edges of each candidate $B_l \in \mathcal{B}_i$. According to [Chow and Liu, 1968], we see that the maximum likelihood tree is a maximal spanning tree (MST). The above whole steps are summarized in Algorithm 9, which computes the most likely tree $\mathcal{T}_i^*$ given $\mathcal{B}_i$. Once we obtain $\mathcal{T}_i^*$, we retain only the candidates $\mathcal{B}_i^*$ ($\subset \mathcal{B}_i$) that are subgraphs of $\mathcal{T}_i^*$.

It is easy to see that Algorithm 9 runs in $O(|\mathcal{B}_i||\mathcal{S}|^2)$ time. Algorithm 9 first generates a complete undirected graph over $\mathcal{S}_i$ in which each edge (u,v) has the sum of values of $B_l \in \mathcal{B}_i$ such that $(u,v) \in B_l$. Its running time is $O(|\mathcal{B}_i||\mathcal{S}|^2)$. Then, the maximum spanning tree is obtained in $O(|\mathcal{S}|^2)$, and the final pruning step is performed in $O(|\mathcal{B}_i||\mathcal{S}|)$ at worst.

As another interpretation, we can also easily prove that Algorithm 9 minimizes the values of rejected $B_l \in \mathcal{B}_i$ under the constraint of the tree structure as shown in

$$\mathcal{T}_i^* = \operatorname*{argmin}_{\mathcal{T} \subset \mathcal{T}(\mathcal{G}_i)} \sum_{B_l \in \mathcal{B}_i, B_l \not\subset \mathcal{T}_i} v(B_l). \tag{7.8}$$

Search Algorithm: According to Theorem 3, given the $\mathcal{B}_i^*$ that are organized in the tree $\mathcal{T}_i^*$, the optimal solution to Eq.(7.1) can be achieved in $O(|\mathcal{B}_i^*||\mathcal{S}_i|)$ by Algorithm 10. We implement the dynamic programming (DP) based search algorithm by modifying the CABOB algorithm [Sandholm and Suri, 2003].

Algorithm 10: Solve region assignment (Eq.(7.1)) from $\mathcal{B}_i^*$

Input: (1) The pruned candidate set $\mathcal{B}_i^*$ (for each $B_l = \langle k_l, C_l, w_l \rangle$ where k_l is the index of the foreground, C_l is a sub- tree of $\mathcal{G}_i$ and w_l is the value to its foreground). (2) T_i^*.

Output: (1) Foreground assignment $\{\mathcal{F}_1^k, \cdots, \mathcal{F}_1^{K+1}\}$.

1: Randomly choose the root r of the tree T_i^*.

2: Assign each node of T_i^* a *level*, which is its distance from the root r. (*e.g.* The level of r is 0).

3: Compute the level of each candidate B_l in $\mathcal{B}_i^*$, where $level(B_l)$ is the smallest level of any item in B_l.

foreach *A node i in T_i^* in a decreasing order of level* **do**

4: Let $\mathcal{C}_i$ be the set of candidates B_l in $\mathcal{B}_i$, each of which includes i and whose level is the same as the level of i.

5: Let $opt(i)$ be the optimal solution for the problem considering only those candidates that contain items in the subtree below i.

6: Compute $opt(i)$ recursively as follows

$$opt(i) = \max\left(\max_{B_l \in \mathcal{C}_i}\left(w_l + \sum_{j \in \mathcal{U}_B} opt(j)\right), \sum_{j \in ch(i)} opt(j)\right) \tag{7.9}$$

where w_l is the value of candidate B_l and $ch(i)$ is the children nodes of i. $\mathcal{U}_B$ be the set of the roots of the forest of subtrees that are obtained by removing all nodes in B_l from T_i^*, while ignoring the subtree containing r.

7: The final solution is a set of candidates $\mathcal{B}_i^{opt}$ associated with $opt(r)$. Finally, $\mathcal{F}_i^K \leftarrow \forall B_l \in \mathcal{B}_i^{opt}$ where k_l of B_l is k.

7.3.4 The MFC Algorithm

Algorithm 11 summarizes the overall algorithm. We repeat the two main procedures, foreground modeling and region assignment, until the objective score of a new region assignment in Eq.(7.1) does not increase or the iterations reach a pre-defined number. Since we deal with free-formed patches of natural images and consider the foreground models as black boxes, it is difficult to analytically understand the convergence property. However, if we use only the GMM model as our foreground model, our algorithm is guaranteed to converge at least to a local minimum [Rother et al., 2004].

The initializations for region assignment are different between supervised and unsupervised settings. In the supervised scenario, we initialize the foreground models from the foreground regions labeled by users: $\mathcal{A} = \{\mathcal{A}^1, \cdots, \mathcal{A}^K\}$ where $\mathcal{A}^k$ is the regions annotated as the k-th foreground. In the unsupervised setting, we apply the diversity ranking method of [Kim et al., 2011] to the similarity graph of $\mathcal{S} = \{\mathcal{S}_1, \cdots, \mathcal{S}_M\}$ to discover the most repeated K regions that are diverse with respect to one another. Note that in the unsupervised setting, commonality of the regions is favored. Hence, when we apply the unsupervised cosegmentation to the images like Fig.7.3, it is unavoidable to detect grass regions as one of K foregrounds because it is dominant across the input images.

Algorithm 11: Multiple foreground cosegmentation

Input: (1) Input image set $\mathcal{I}$. (2) Number of foregrounds (FGs) K. (3) (In supervised case) annotations $\mathcal{A} = \{\mathcal{A}^1, \cdots, \mathcal{A}^K\}$.

Output: Foregrounds $\mathcal{F}_i = \{\mathcal{F}_i^1, \cdots, \mathcal{F}_i^K\}$ for all $I_i \in \mathcal{I}$.

Initialization

foreach $I_i \in \mathcal{I}$ **do**

  1: Oversegment I_i to $\mathcal{S}_i$ and build adjacency graph $\mathcal{G}_i = (\mathcal{S}_i, \mathcal{E}_i)$ where $(s_l, s_m) \in \mathcal{E}_i$ if $\min d(s_l, s_m) \leq \rho$.

if *unsupervised* **then**

  2: Apply diversity ranking of [Kim et al., 2011] to the similarity graph of $\mathcal{S}=\bigcup_{i=1}^{M} S_i$ to find K regions $\mathcal{A}=\{\mathcal{A}^1, \cdots, \mathcal{A}^K\}$ that are highly repeated in $\mathcal{S}$ and diverse with respect to one another.

3: Set $\mathcal{F} \leftarrow \mathcal{A}$.

Iterative Optimization

/ Stopping condition. */*

We stop the iteration if a new region assignment does not increase the objective value (*i.e.* $\sum_{i=1}^{M}\sum_{k=1}^{K+1} v^k(\mathcal{F}_i^k)$ from Eq.(7.1)).

/ Foreground Modeling (Any methods can be used). */*

foreach $k = 1{:}K$ **do**

  1: Learn GMM and SPM FG models from $\mathcal{F}^k$ (See Table 7.2).

/ Region assignment */*

foreach $I_i \in \mathcal{I}$ **do**

  foreach $k = 1{:}+1K$ **do**

    2: Generate FG candidates $\mathcal{B}_i^k$ by Alg.8 as a set of $B_i^k = \langle k_j, C_j, w_j \rangle$, where k_j is the foreground index, $C_j \subseteq \mathcal{S}_i$ is a subtree of $\mathcal{G}_i$, and $w_j = v^k(C_j)$.

  3: Compute the most probable candidate tree $\mathcal{T}_i^*$ and pruned $\mathcal{B}_i^*$ by Eq.(7.2) from $\mathcal{B}_i = \bigcup_{k=1}^{K+1} \mathcal{B}_i^k$.

  4: Obtain $\mathcal{F}_i$ to solve region assignment in Eq.(7.1) by using Algorithm 10 on $\mathcal{B}_i^*$.

7.4 Experiments

We evaluate the proposed MFC algorithm using the FlickrMFC dataset and the ImageNet dataset [Deng et al., 2009].

7.4.1 Results over FlickrMFC Dataset

Datasets: We introduce a new dataset called FlickrMFC for the benchmark purpose of multiple foreground cosegmentation. They are sampled images from Flickr photo streams, each of which is taken by a single user for a specific event in a single day. Hence, a fixed number of subjects (or foregrounds) frequently occur across the photo stream, but an unknown subset of them appears in every single image. We also provide hand-labeled pixel-level groundtruths.

The FlickrMFC consists of 14 groups, each of which contains 12∼20 images. Table 7.3 summarizes some key information about the collected groups including the number of images and the

Group	M	K	Foreground names
apple+picking	20	6	apple+bucket, baby, boy+blue, girl+blue, girl+red, pumpkin
baseball+kids	18	5	ball, boy+black, boy+gray, coach, glove
butterfly+blossom	18	8	beetle, butterfly+orange, butterfly+tiger, butterfly+yellow, flower+pink, flower+red, ladybug, leaf
cheetah+ safari	20	5	cheetah, eagle, elephant, lion, monkey
cow+pasture	20	5	cow+black, cow+brown, man+blue, man+red+cap, truck
dog+park	20	4	dog+black, dog+brown, dog+white, woman
dolphin+aquarium	18	3	killer+whale, dolphin+gray, seal
fishing+alaska	18	5	flower, man+gray, man+white, salmon, woman+gray
gorilla+zoo	18	4	boy, girl, gorilla+black, orangutan+brown
liberty+statue	18	4	boat+blue, boat+red, empire+state+building, liberty+statue
parrot+zoo	18	5	hand, parrot+green, parrot+red, parrot+white, parrot+yellow
stonehenge	20	5	bird, cow+black, cow+white, person, stonehenge
swan+zoo	20	3	flower+yellow, swan+black, swan+gray
thinker+Rodin	17	4	sculpture+thinker, sculpture+venus, van+gogh, woman

Table 7.3: Summary of 14 groups of *FlickrMFC* dataset. M and K denote the number of images and foregrounds, respectively.

Figure 7.4: The FlickrMFC dataset. We show all images of four groups (*apple+picking+fall*, *cow+pasture*, *stonehenge+england*, and *parrot+zoo+bird*) from top to bottom. In each group, the first row shows input images, and the second row illustrates hand-labeled pixel-level groundtruths.

description of foregrounds. The group names are identical to the search keywords that are used for image downloading from Flickr. Fig.7.4 shows all images of four selected groups.

Baselines: As baselines, we use one LDA-based unsupervised localization method [Russell et al., 2006] (LDA) and two cosegmentation algorithms: CoSand [Kim et al., 2011] (COS) and discriminative clustering method [Joulin et al., 2010] (DC). Since the two cosegmentation methods are not intended to handle irregularly appearing multiple foregrounds, we first manually divide the

Figure 7.5: Examples of multiple foreground cosegmentation on selected groups of FlickrMFC dataset. We sampled 5∼7 images per group. Each set presents input images, color-coded cosegmentation output, and segmented foregrounds, from top to bottom. The color bars below each set indicate which foregrounds are assigned to colored regions.

images into several subgroups so that the images of each subgroup share the same foregrounds. If an image contains multiple foregrounds, it belongs to multiple subgroups. Then, we apply the methods to each subgroup separately to segment out the common foreground. This is an exact scenario where a conventional cosegmentation is applied to the image sets of multiple foregrounds. The (LDA) [Russell et al., 2006] was not originally developed for cosegmentation, but it can segment multiple object categories without any annotation input. We use the source codes provided by original authors.

Results: Our algorithm is tested in both supervised (MFC−S) and unsupervised (MFC−U) settings. In (MFC−S), we randomly choose 20% of input images (*i.e.* 2∼4 images) to obtain annotated labels for the foregrounds of interest. For the unsupervised algorithms, (MFC−U) and (LDA), it is hard to know the best K beforehand. Thus, we run them by changing K from two to eight, and report the best results.

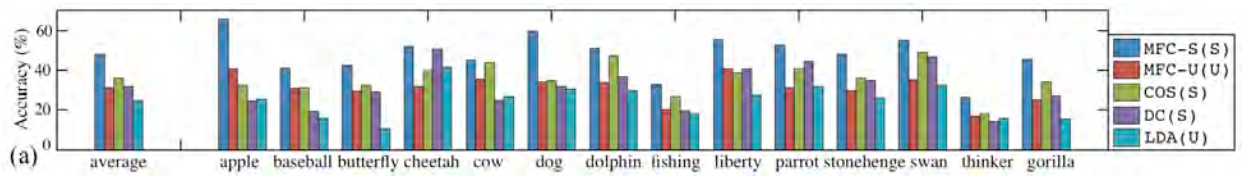


Figure 7.6: Comparison of segmentation accuracies between our supervised (MFC−S) and unsupervised (MFC−U) approaches and other baselines (COS, DC, LDA) for the FlickrMFC dataset. The S and U indicate whether any annotation input is required (S) or not (U).

Fig.7.6 summarizes the segmentation accuracies on the 14 groups of the FlickrMFC dataset. In the figure, the leftmost bar set is the average performance on 14 groups. The accuracy is measured by the intersection-over-union metric ($\frac{GT_i \cap R_i}{GT_i \cup R_i}$), the standard metric of PASCAL challenges. We observed that the performance of our (MFC−U) is slightly worse than (COS) and (DC) by $2\sim3\%$. Note that (COS) and (DC) are applied to the images of each separate subgroup that shares the same foregrounds. It allows the algorithms to know what foregrounds exist in the images beforehand, which is a strong supervision. On the other hand, (MFC−U) is a completely unsupervised; it is applied to the entire dataset without splitting. Our supervised (MFC−S) algorithm, even with a very small number of labeled images, significantly outperformed the competitors by more than 11% over the best of baselines (COS).

Fig.7.5 shows some examples of cosegmentation from six groups of the FlickrMFC dataset. In each set, we show input images, color-coded cosegmentation output, and segmented foregrounds from top to bottom. The same colored regions in the second row are identified as the same foregrounds, and the meanings of the colors are described below each set. We made several interesting observations in these examples: First of all, our algorithm correctly treated the multiple foreground cosegmentation in Definition 1. In Fig.7.5.(a), two girls, a boy, a baby, an apple bucket, pumpkins are intended foregrounds, which are irregularly presented in each image. This is a challenging situation for traditional cosegmentation methods, but our algorithm could successfully segment the foregrounds. As shown in the fourth image of Fig.7.5.(d), some input images include no foregrounds, which were successfully identified as well. One main source of errors in our experiments was the similarly looking regions; for example, in the first image in Fig.7.5.(a), the face region of the *girl+red* is allocated to the *girl+blue* foreground (depicted in red), which makes sense in that the two foregrounds are the girls with similar skin and hair colors but their main difference lies in their clothes.

7.4.2 Results over ImageNet Dataset

Dataset: ImageNet [Deng et al., 2009] may not be a perfect dataset for the evaluation of multiple foreground segmentation because each image contains only a single object class with a significant size. Instead, the main objectives of the evaluation with ImageNet [Deng et al., 2009] are to show (i) the scalability of our method, and (ii) the performance evaluation for the single foreground cosegmentation as a simplified task.

Baselines: We follow the experiment setting of [Kim et al., 2011] in order to compare our segmentation performance with those of (COS) [Kim et al., 2011], (LDA) [Russell et al., 2006],

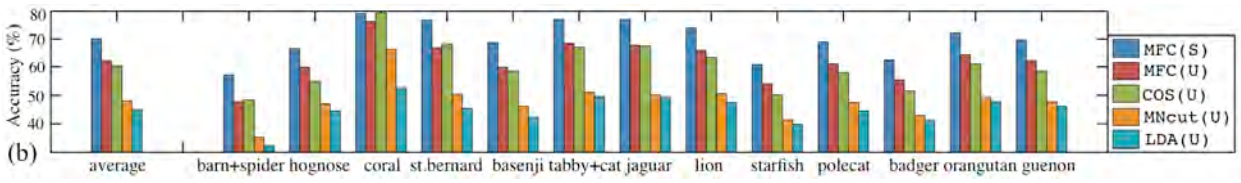


Figure 7.7: Comparison of segmentation accuracies between our approach and other baselines for the ImageNet dataset.

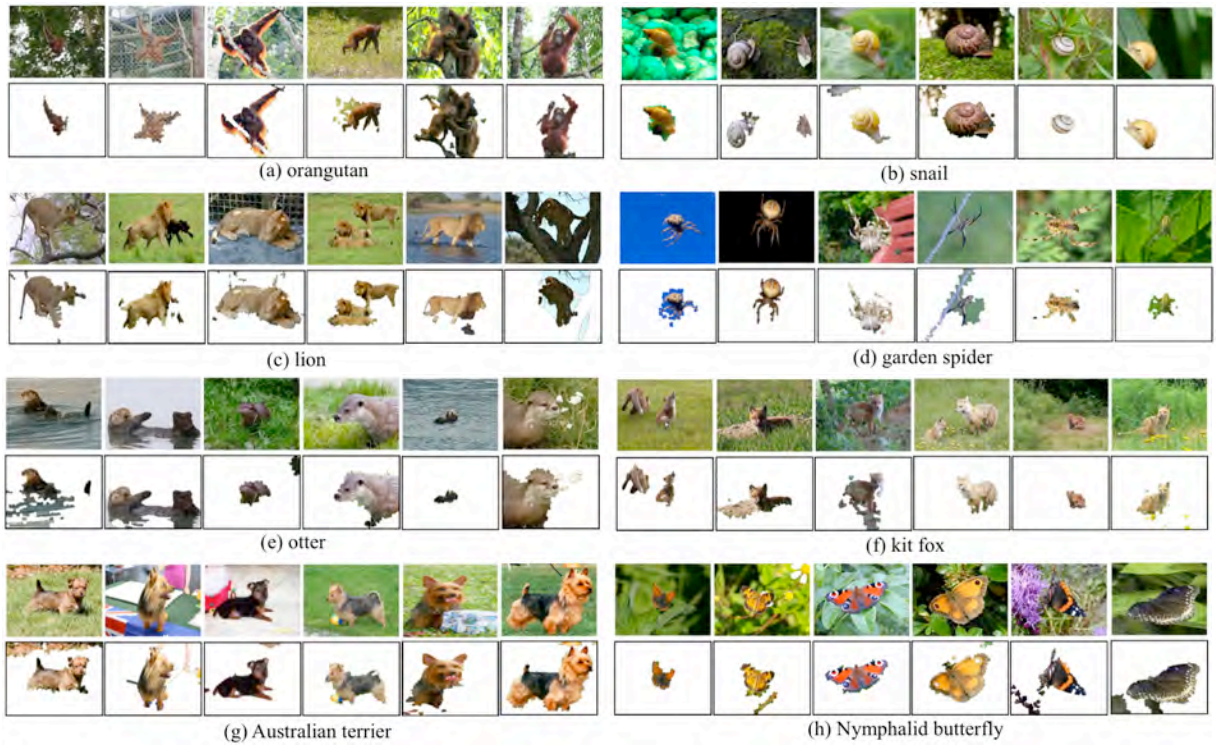


Figure 7.8: Examples of scalable cosegmentation on the ImageNet dataset. We sample six images from each synset. In each set, the first row shows input images, and the second row illustrates segmented foregrounds.

and MNcut [Cour et al., 2005] that are reported in [Kim et al., 2011]. We select 50 synsets that provide bounding box labels, and apply our technique to 1000 randomly selected images per synset in both supervised (MFC−S) and unsupervised (MFC−U) ways. In (MFC−S), the foreground models are initialized from the labels of 50 randomly chosen images. Finally, we compute segmentation accuracies by using the provided bounding box annotations.

Results: Fig.7.7 shows the segmentation accuracies for 13 selected synsets. The accuracies of (MFC−U) and (MFC−S) are higher than those of the best baselines (COS) by more than 3% and 8%, respectively. As discussed before, our algorithm is linear to M and it took about 20 min for 1,000 images on a single machine.

Fig.7.8 illustrates six segmented images for following synsets: *orangutan*, *snail*, *lion*, *garden+spider*, *otter*, *kit+ fox*, *Australian terrier*, and *Nymphalid butterfly*. Most ImageNet images

contain only a single object class with a significant size, and thus we here sample some challenging images that contain a relatively small foreground in cluttered background. The results show that the proposed approach has been also successful to the single foreground cosegmentation, which is a simper task than the multiple foreground cosegmentation defined in this chapter.

7.5 Summary

In summary, the main contributions of this chapter can be summarized as follows.

- We develop an approach to multiple foreground cosegmentation, which is a less restrictive and more practical cosegmentation so far, in order to be directly applicable to general users' photos. To the best of our knowledge, such cosegmentation tasks remain an under-addressed topic in the computer vision literature.
- We formulate the proposed cosegmentation as an iterative combinatorial auction in which image regions are optimally allocated to one of foregrounds or background to maximize the total relevance values of the regions to the assigned foreground or background. Our approach is flexible enough to be integrated with any advanced region classification algorithms, and achieves an optimal solution to region assignment in $O(MK)$, where M and K are the number of images and foregrounds, respectively.

Part III

Reconstruction and Applications of Photo Storylines

Part III – Reconstruction and Applications of Photo Storylines

In this part, we discuss the discovery of collective photo storylines and their potential uses for several interesting Web applications. This part consists of three chapters.

First, we propose an approach to jointly aligning and segmenting uncalibrated multiple photo streams of outdoor recreational activities, as a first technical step to detect the collective storylines. The alignment task discovers the matched images between different photo streams, and the image segmentation task parses the images into multiple meaningful regions to facilitate the image understanding. We integrate the two tasks so that solving one task helps enhance the performance of the other. To this end, we design scalable message-passing based optimization framework to jointly achieve both tasks for the whole input image set at once.

Second, we investigate an approach for reconstructing storyline graphs from large-scale photo collections, and optionally other side information such as friendship graphs. The storyline graphs can be used as an effective structural summary that visualizes various events or activities recurring across the input photo sets, which otherwise are too overwhelming for users to grasp any underlying big picture. We formulate the storyline reconstruction problem as an inference of sparse time-varying directed graphs, and develop an optimization algorithm that achieves a number of key challenges of Web-scale storyline reconstruction, including global optimality, linear complexity, and easy parallelization.

Third, we propose to leverage large-scale online photo collections contributed by the general public, for the analysis of brand associations, given that photos are gaining popularity as an important information modality on the Web. More specifically, we aim to jointly achieve the following two visualization tasks in a mutually-rewarding way: (i) detecting and visualizing core visual concepts associated with brands, and (ii) localizing the regions of brand in images.

Chapter 8

Jointly Aligning and Segmenting Multiple Photo Streams

8.1 Introduction

The work in this chapter is closely related to the main theme of this dissertation: building collective photo storylines from the photo streams of millions of users, and discovering the relations between the reconstructed storylines and the photo streams of individual users. As a first technical step to achieve this goal, in this chapter, we propose a method to jointly perform *alignment* of multiple photo streams and *cosegmentation* of aligned images, as shown in Fig.8.1. In the alignment step, images of different photo sets are matched based on visual contents and associated meta-data. The alignment is a core task to build a big picture of storylines from a large number of fragmented photo streams of individual users. In the cosegmentation step, the aligned images are segmented together in order to facilitate image understanding such as pixel-level classification in the images. It is important to note that solving these two tasks are *mutually rewarding*. The main challenge of cosegmenting multiple photo streams is that the Web images by general users are too diverse to segment all at once. Jointly segmenting images with no commonality, which contradicts the basic assumption of cosegmentation, could be worse than individually segmenting each image. Therefore, the alignment step fills in the role of enabling grouping of images that share sufficient commonality, which provides a high-level clue for cosegmentation. Conversely, once we parse each image into multiple segments, image matching, a basic operation for the photo stream alignment can be improved. We can iterate these two steps in multiple rounds.

In our approach, photo stream alignment and image cosegmentation are achieved in a similar way. For the alignment, we first establish a sparse graph that connects similar photo streams to be aligned together as a Markov random field. Then, we perform belief propagation to jointly align all photo steams at once. Likewise, for image cosegmentation, we build a graph linking the coherent images that are beneficial to be segmented together, based on the output of the alignment step. Then, we perform cosegmentation of the entire image set all at once under the guidance of the graph by a message-passing style optimization.

For evaluation, we collect about 1.5 millions of images of 13 thousands of photo streams regarding 15 outdoor recreational activities from Flickr. Our experiments in Section 8.5 demonstrate that our approach outperforms other candidate methods on both photo stream alignment and image cosegmentation.

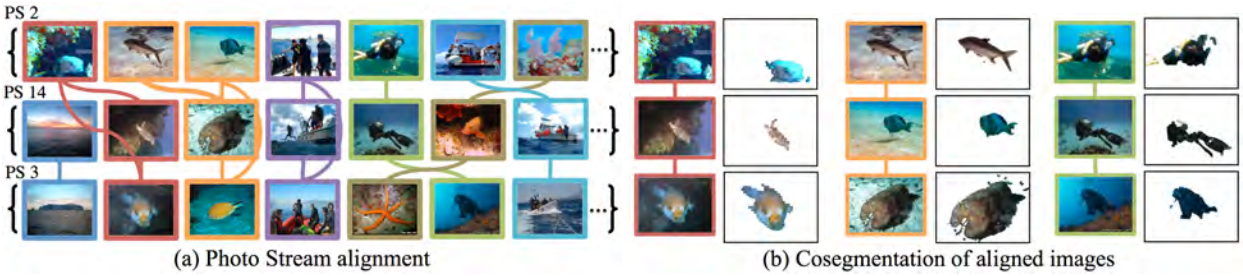


Figure 8.1: Motivation for jointly aligning and segmenting multiple photo streams with an example of three photo streams of *scuba+diving*. The input is any number of photo streams of an activity class that are taken by various users at different time and places. The outputs are two fold: (a) Photo stream alignment. The images in different photo streams are matched (as shown in the same colors). (b) Image cosegmentation. The shared regions in the aligned images are jointly segmented. Photo stream info. $=$ {PS2: u_1 at 10/19/2008 (Cayman Islands), PS3: u_2 at 03/19/2005 (Phuket, Thailand), PS14: u_3 at 08/27/2008 (Cozumel, Mexico)}.

Our problem involves segmenting aligned photo streams together. It resembles the cosegmentation problem [Batra et al., 2011; Joulin et al., 2012; Kim and Xing, 2012; Kim et al., 2011; Rother et al., 2006; Vicente et al., 2011], in which the objective is to jointly segment recurring objects (or foregrounds) that are shared in multiple images. Since we already survey the recent literature about cosegmentation in previous chapters, we briefly discuss the unique features of our work comparing to the large body of previous cosegmentation research. First, we focus on segmentation of unordered multiple web photo streams. The cosegmentation of Flickr photo streams was discussed in the MFC method in chapter 7 [Kim and Xing, 2012], but it was applied to at most 20 images that are manually selected out of hundreds of pictures of a single Flickr photo stream. In contrast, here we can handle an arbitrary number of uncalibrated Web photo streams by closing the loop between segmentation and photo stream alignment. Second, in our experiments, we perform scalable segmentation with more than 100K images of 1K photo streams, which exceeds those of previous work by two orders of magnitude. To our knowledge, the largest dataset sizes in previous work are about 1K [Kim and Xing, 2012; Kim et al., 2011].

Another interesting thread of research related to our work is *large-scale image alignment*. Image alignment has been one of fundamental tasks in a variety of computer vision problems. Recently, with the explosion of pictures available online, image alignment has become a key building block to solve several large-scale novel problems. Some notable examples include the reconstruction of 3D models of landmarks [Snavely et al., 2010], the localization of tourists' photos [Chen and Grauman, 2011], spatio-temporal reconstruction of time-varying 3D city models [Schindler and Dellaert, 2010], and nonparametric object recognition and scene parsing [Liu et al., 2009a]. However, their objectives of the image alignment are quite different from ours, which is to integrate with a subsequent image segmentation to infer common storylines of outdoor activities. As far as we know, [Yang et al., 2011] is one of the very few papers that involve the alignment of multiple photo streams. However, their algorithm is tested with relatively small datasets (*i.e.* 12 classes with less than 10 photo streams per class) compared to ours (*i.e.* 15 outdoor activities with 1K photo streams per activity) by orders of magnitude. More importantly, they did not explore any sub-image level analysis; no image segmentation is performed.

8.2 Problem Formulation

In this section, we describe the problem definition and the overview of our solution to the problem.

8.2.1 Input and Output

The input of our algorithm is the set of photo streams of a particular activity denoted by $\mathcal{P} = \{P^1, \cdots, P^L\}$, where L is the number of input photo streams. Each photo stream is a set of photos taken in sequence by a single photographer within a certain period of time, which is set to a single day in this work. Without loss of generality, we assume that each photo stream is sorted by taken time. We also use $\mathcal{I} = \{I_1, \cdots, I_N\}$ to denote the whole image set without distinguishing the membership of photo streams. As a notation convention, we use superscripts to denote photo stream numbers and subscripts to denote image numbers.

Another input is related to the segmentation task; a user can provide the maximum number of foregrounds of interest per image K. Then, our algorithm automatically identifies K most dominant regions that are distinctive one another from the image and its aligned neighbors[1]. The background is defined as all the other regions that are not included in any of K foregrounds. For notational simplicity, we interchange the term background and foreground $K+1$.

The output of our algorithm is two-fold. The first output for the alignment is the set of correspondences between the images of different photo streams. If we represent each image as a vertex and each correspondence as an edge, the output can be summarized as an L-partite graph. The second output for the segmentation is assigning every pixel of each image to one of K foregrounds or background.

8.2.2 Overview of Algorithm

Our approach alternates between solving the two target tasks, photo stream alignment and image cosegmentation. Given a large set of uncalibrated photo streams, we first build a nearest neighbor similarity graph that connects the photo streams to be aligned (see section 8.3.4). We formulate the alignment of the whole photo streams as an energy minimization problem, which can be solved by belief propagation on the graph. Its detailed procedure will be explained in section 8.3.3 and 8.3.4. As a result of the alignment, we can obtain the correspondences between the images of different photo streams, from which we establish an image graph connecting the similar images that are likely to share common foregrounds (see section 8.4.1). We perform large-scale cosegmentation for all images at once under the guidance of the image graph in a message-passing way, which will be discussed in section 8.4.2. The segmentation of images can enhance the similarity measurement between images, which subsequently contributes to a better photo stream alignment. This will be justified in section 8.3.2 with an intuitive example. Finally, we can return to the photo stream alignment step with the new segmentation-based image similarity.

[1] In segmentation literature, it is called an *unsupervised* setting. A user may provide some foreground examples in the form of bounding-boxes or pixel-wise annotations, which is called a *supervised* setting. In this work, we focus on the unsupervised case because it is more challenging. Also, it is trivial to adapt our approach to the supervised setting.

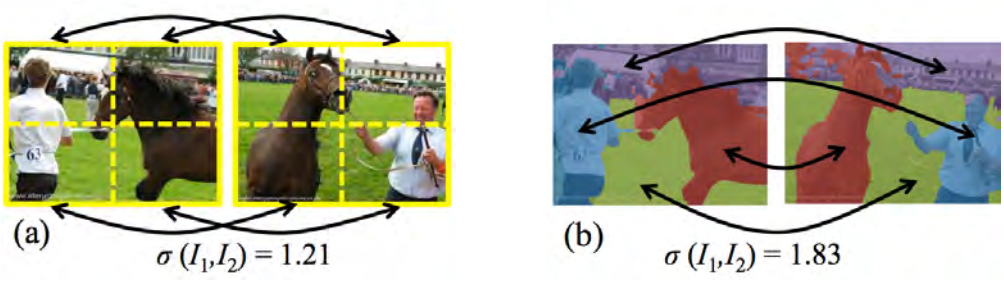


Figure 8.2: The benefit of segmentation for measuring image similarity. In this example, the same objects appear in different locations with different poses across the image pair. (a) When images are not yet segmented, we compute the image similarity using the spatial pyramid histograms on the whole images. (b) Once images are segmented, we find the best assignment between the segments of two images, and compute the mean of segment similarities.

8.3 Alignment of Photo Streams

We begin with our image description and similarity measure, and then discuss the proposed alignment algorithm.

8.3.1 Image Description

We use the dense feature extraction with vector quantization, which is one of standard methods in recent computer vision research. We densely extract two features from each image: HSV color SIFT and histogram of oriented edge (HOG) feature on a regular grid at steps of 4 and 8 pixels, respectively. Then, we form 300 visual words for each feature type by applying K-means to randomly selected descriptors. Finally, the nearest word is assigned to every node of the grid. As image and segment descriptors, we build L_1 normalized spatial pyramid histograms to count the frequency of each visual word in multiple levels of regular grids.

8.3.2 Image Similarity Measure

It is vital to design a reliable similarity metric between images for an accurate alignment of photo streams. Our alternating approach is based on the assumption that the segmentation is helpful to enhance the measurement of image similarity. Fig.8.2 shows a typical example of such intuition where the same objects appear in different locations with different poses across the images. When images are not segmented yet, the image similarity is calculated from two-level spatial pyramid histograms on the whole images, which are not robust against location and pose variations. However, this issue can be largely alleviated even with an imperfect segmentation. Given the segment sets of two images I_1 and I_2, denoted by $\mathcal{F}_1$ and $\mathcal{F}_2$, we first solve the linear assignment problem (*i.e.* finding the best assignment between the segments of two images), and then compute the mean of total similarity values as an image similarity metric. Formally, given a similarity metric between segments $\sigma_s : \mathcal{F}_1 \times \mathcal{F}_2 \rightarrow \mathbb{R}$, the image similarity σ is defined by

$$\sigma(I_1, I_2) = \max \left(\sum_{s \in \mathcal{F}_1} \sigma_s(s, f_s(s)) \right) / M \tag{8.1}$$

where $f_s : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ is a bijection and M is the number of segments. We use as σ_s the histogram intersection on the spatial pyramid histograms of the segments.

8.3.3 Pairwise Photo Stream Alignment

For a better understanding, our discussion starts from the alignment of a pair of photo streams P^1 and P^2. That is, the objective is to establish the correspondences between two photo streams through image matching. Our alignment objective is formulated based on the MRF energy function that has been applied to many computer vision problems such as deformable image matching [Shekhovtsov et al., 2007] and SIFT flow [Liu et al., 2009a]. Its strength lies in its flexibility to easily incorporate various energy terms related to alignment. It is of particular interest for our applications since we can easily incorporate various energy terms related to the alignment using any meta-data associated with the images.

The goal of alignment is to find a matching $f : P^1 \rightarrow P^2 \cup \{\emptyset\}$ where $\emptyset$ is the null, meaning that if $f(p_i) = \emptyset$ for an image $p_i \in P^1$, p_i has no correspondence in P^2. Let $\widehat{p_i} \in P^2 \cup \{\emptyset\}$ denote the matched image to $p_i \in P^1$. The pairwise alignment is performed by minimizing the energy function as follows.

$$\begin{aligned} E(P^1, P^2) = &- \sum_{p_i \in P^1} \sigma(p_i, \widehat{p_i}) + \sum_{p_i \in P^1} \eta \min(|t(p_i) - t(\widehat{p_i})|, \tau) \\ &+ \sum_{(p_i, p_j) \in \Delta} \rho \, \sigma(p_i, p_j) \min(|t(\widehat{p_i}) - t(\widehat{p_j})|, \nu) \end{aligned} \tag{8.2}$$

where τ and ν are the thresholds for truncated L_1 norms, and η and ρ are term weights. We let $t(p_i)$ be the timestamp of image p_i. The $\sigma(p_i, \widehat{p_i})$ is the image similarity between p_i and $\widehat{p_i}$. We let $\sigma(p_i, \emptyset) = 0$ and $t(\emptyset) = \infty$, which means that if $\min_{p_j \in P^2} \sigma(p_i, p_j) < \eta\tau + \rho\nu$, then p_i matches no image in P^2. The Δ contains the entire temporal neighborhood in a photo stream (*i.e.* $(p_i, p_j) \in \Delta$ means $|t(p_i) - t(p_j)| \leq \delta$). The first term accounts for the maximization of image similarity between the matched pairs, and the second term penalizes the time difference between the matched pairs. It is useful for tie-breaking of equally visually similar pairs using temporal information. The third one is the smoothness term to encourage that the matched images to the neighbors in P^1 are also neighbors in P^2. This regularization is more strongly imposed for a pair of images that are more visually similar by weighting $\sigma(p_i, p_j)$. The optimization of Eq.(8.2) can be achieved by using the belief propagation [Felzenszwalb and Huttenlocher, 2006; Liu et al., 2009a].

8.3.4 Multiple Photo Stream Alignment

We extend the pairwise alignment of Eq.(8.2) to that of an arbitrary number of photo streams $\mathcal{P}$. One naive approach may be to incrementally combine pairwise alignments starting from the most

similar photo stream pair and progressing to the most distant one. However, this approach has two significant drawbacks [Crandall et al., 2011]. First, it tends to be computationally intensive. Second, more importantly, this method does not treat all photo streams equally, which may lead to local minima according to the order of consideration.

To circumvent these issues, we jointly align all photo streams at once after constructing a graph between photo streams $\mathcal{G}_P = (\mathcal{P}, \mathcal{E}_P)$. For each photo stream $P^i \in \mathcal{P}$, we first find a set of photo streams that are sufficiently overlapped on timeline (*i.e.* the photo streams P^j such that (# of images of P^j within the time range of P^i)/ (total # of images P^j) $\geq \gamma$). Among them, we obtain K_P-nearest neighbors in terms of visual similarity, which is calculated using the idea of Naive-Bayes Nearest-Neighbor [Boiman et al., 2008] as follows. Given two photo streams P^i and P^j, for each image $p \in P^i$, we obtain the first nearest neighbor in P^j denoted by $\text{NN}(p)$. Then, the similarity from P^i to P^j is computed by $\sum_{p \in P^i} \|\sigma(p, \text{NN}(p))\|^2$. Finally, $\mathcal{E}_P$ includes all pairs of nearest neighbor photo streams.

The objective of multiple photo stream alignment reduces to find a matching $f : P^i \rightarrow P^j \cup \{\emptyset\}$ for all pairs $(P^i, P^j) \in \mathcal{E}_P$, which can be accomplished by minimizing

$$E = \sum_{(P^i, P^j) \in \mathcal{E}_P} E(P^i, P^j) \tag{8.3}$$

where $E(P^i, P^j)$ is defined by Eq.(8.2). The optimization can be achieved by the belief propagation on the graph of photo streams $\mathcal{G}_P$, in such a way that we repeat a pairwise alignment of previous section by following the edges of $\mathcal{E}_P$ until convergence.

8.4 Large-Scale Cosegmentation

In this section, we explain our algorithm to construct an *image graph* and jointly segment the whole image set.

8.4.1 Building Image Graphs

For large-scale cosegmentation, we establish an *image graph* $\mathcal{G}_I = (\mathcal{I}, \mathcal{E}_C)$ where $\mathcal{I}$ is the set of images of all photo streams, and $\mathcal{E}_C$ is the set of edges that connect the images that share enough commonality to be segmented together. The edge set consists of two groups: $\mathcal{E}_C = \mathcal{E}_B \cup \mathcal{E}_W$ where $\mathcal{E}_B$ defines the edges between the images of different photo streams while $\mathcal{E}_W$ connects the images within the same photo stream. $\mathcal{E}_B$ is trivially obtained from the output of photo stream alignment; simply, all correspondences of image pairs are added to $\mathcal{E}_B$. $\mathcal{E}_W$ is useful for cosegmentation because the images in the same photo stream are consecutively taken by the same camera, and thus they are likely to share common objects and scenes. In order to define $\mathcal{E}_W$, we find K_W-nearest neighbors for each image I_i among its temporal neighborhood in the same photo stream, which includes all images I such that $|t(I) - t(I_i)| \leq \delta$. In our experiments, δ is set to 2 hours.

8.4.2 Scalable Cosegmentation

We begin with some basic ingredients of our cosegmentation algorithm. We first oversegment every image of $\mathcal{I}$ by using the submodular image segmentation in chapter 6 [Kim et al., 2011]. Let $\mathcal{S}_i$ denote the set of oversegments of image I_i. Then, the goal of segmentation reduces to finding an optimal disjoint partition $\mathcal{S}_i = \bigcup_{k=1}^{K+1} \mathcal{F}_i^k$ with $\mathcal{F}_i^k \cap \mathcal{F}_i^l = \emptyset$ if $k \neq l$, where $\mathcal{F}_i^k$ denotes the regions of foreground k in image I_i.

MFC algorithm: In our approach, we select the MFC in chapter 7 [Kim and Xing, 2012] as our base cosegmentation algorithm, since it is scalable and has been successfully tested with Flickr user images. More specifically, we exploit two procedures of the MFC algorithm as our basic operations: *foreground modeling* and *region assignment* steps. The foreground models retain the appearance models of K foregrounds and the background. Formally, the k-th foreground model is defined as a parametric function $v^k : 2^{|\mathcal{S}_i|} \rightarrow \mathbb{R}$ that takes any subset $S \subset \mathcal{S}_i$ as input and returns its value to foreground k (*i.e.* how closely region S is relevant to foreground k). Each foreground model is learned from the regions that are allocated to the foreground after the region assignment step. Therefore, the foreground model can be accomplished by using any region classifiers or their combinations. In this work, we use the Gaussian mixture model (GMM) on the RGB color and HSV SIFT spaces. Thus, $v^k(S)$ is defined as the mean log-likelihood of the descriptors of S to the k-th learned GMM model [Rother et al., 2004].

The role of the region assignment step is, given a set of learned foreground models $\{v^k\}_{k=1}^{K+1}$, to discover the optimal partition of $\mathcal{S}_i$ into $\{\mathcal{F}_i^k\}_{k=1}^{K+1}$ that maximizes the overall allocation values. We let c_i denote one such partition instance of image I_i. Generally, the set partition problem is NP-complete, but the region assignment of the MFC can solve it in a very efficient way by using combinatorial auction idea. We do not discuss its details, which can be found in [Kim and Xing, 2012]. Instead, we denote the region assignment procedure by $\{\mathcal{F}_i^k\}_{k=1}^{K+1} = \mathsf{RegAss}(\mathcal{S}_i, \{v^k\}_{k=1}^{K+1})$. In the following, we use the abbreviated notation of $\{v\}$ for $\{v^k\}_{k=1}^{K+1}$.

Message Passing based Cosegmentation: The basic idea of our large-scale cosegmentation is to iteratively perform foreground modeling and region assignment based on image graph $\mathcal{G}_I$. We view the image graph $\mathcal{G}_I$ as a MRF with hidden variables corresponding to the partition c_i of each image I_i. Consequently, we formulate the cosegmentation of whole image set $\mathcal{I}$ as the following energy maximization:

$$D(\mathcal{I}; \mathcal{G}_I) = \alpha \sum_{I_i \in \mathcal{I}} \psi\left(c_i; \{v\}\right) + \sum_{(I_i, \mathcal{N}_i) \in \mathcal{E}_C} \phi\left(c_i; \{v_{\mathcal{N}_i}\}\right) \tag{8.4}$$

where $\mathcal{N}_i$ denotes the neighborhood of image I_i in image graph $\mathcal{G}_I$, and α is a term weight. $\{v\}$ and $\{v_{\mathcal{N}_i}\}$ indicate the global and local foreground models, respectively. Both of them are implemented by the same region classifiers (*e.g.* GMM models), and only difference is the training data; $\{v_{\mathcal{N}_i}\}$ is learned from the regions of foregrounds only in $\mathcal{N}_i$, whereas $\{v\}$ is obtained without imposing such local restriction.

The objective of Eq.(8.4) consists of a unary term ψ and a pairwise term ϕ; it means that c_i is achieved by searching for the best partition not only for $\{v\}$ in the unary term ψ but also for $\{v_{\mathcal{N}_i}\}$ in the pairwise term ϕ. For a partition c_i of $\mathcal{S}_i$ into $\{\mathcal{F}_i\}$, the unary term is defined as the sum of assignment scores by $\{v\}$:

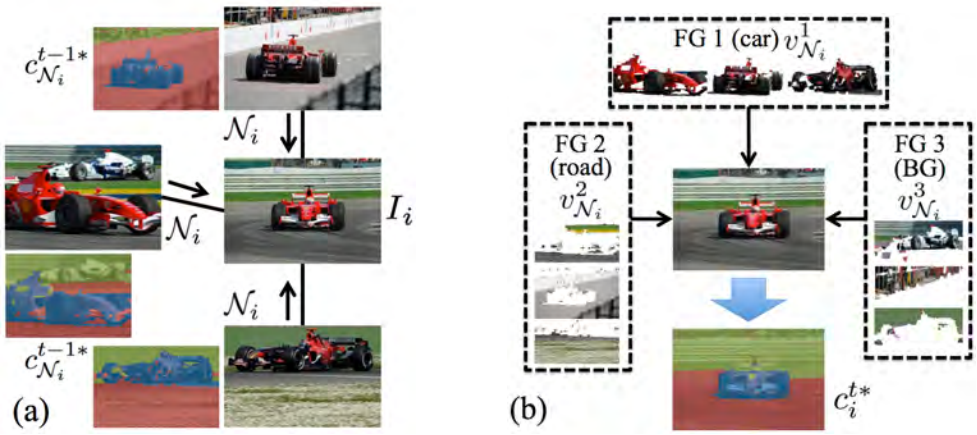


Figure 8.3: An intuition of our message-passing based cosegmentation at round t. (a) We show an image I_i to be segmented, and its three neighbors $\mathcal{N}_i$ in the image graph $\mathcal{G}_I$. We also present color-coded partitions of best beliefs of $\mathcal{N}_i$ at $t-1$, denoted by $c_{\mathcal{N}_i}^{t-1*}$. (b) The message passing from $\mathcal{N}_i$ to I_i at round t ends up performing the region assignment for I_i by using the foreground models $\{v_{\mathcal{N}_i}\}$ learned from $c_{\mathcal{N}_i}^{t-1*}$. As a result, we obtain the partition of the best belief of image I_i at t, denoted by c_i^{t*}.

$$\psi\left(c_i;\{v\}\right)=\sum_{k=1}^{K+1} v^k(\mathcal{F}_i^k). \tag{8.5}$$

The pairwise term $\phi\left(c_i;\{v_{\mathcal{N}_i}\}\right)$ is defined as the exact same form of Eq.(8.5) only except replacing $\{v\}$ by $\{v_{\mathcal{N}_i}\}$.

Optionally, the unary term ψ can be reasonably ignored by setting α to 0, if it is hard to define a single set of globally applicable foreground models. For example, the *person* foregrounds are ubiquitous in all photo sets but their appearances can severely vary in different photo sets. In this case, using only local models may be more robust.

Messages and beliefs: The energy maximization in Eq.(8.4) can be solved by the belief propagation, which proceeds by iteratively computing new *messages* for each edge in graph $\mathcal{G}_I$. Using the max-product algorithm (*i.e.* equivalently, the min-sum algorithm with negative log probabilities), the message from $\mathcal{N}_i$ to I_i at round t is defined by [Felzenszwalb and Huttenlocher, 2006]

$$m_{\mathcal{N}_i\to I_i}^t(c_i)=\max_{c_{\mathcal{N}_i}}\left(\phi\left(c_i;\{v_{\mathcal{N}_i}\}\right)+\psi\left(c_{\mathcal{N}_i};\{v\}\right)+\sum_{s\in\mathcal{N}(\mathcal{N}_i)\setminus I_i} m_{s\to\mathcal{N}_i}^{t-1}(c_{\mathcal{N}_i})\right) \tag{8.6}$$

where $\mathcal{N}(\mathcal{N}_i)\setminus I_i$ denotes the neighbors of $\mathcal{N}_i$ except I_i. According to Eq.(8.6), the message computation involves the search for the best $c_{\mathcal{N}_i}$ (*i.e.* the partitions of neighbors) for every possible c_i. It results in an exponential explosion of the search space, which is largely unnecessary in practice. Therefore, we introduce an assumption that is reasonable for image cosegmentation as follows. *The best partitions $c_{\mathcal{N}_i}$ for the message $m_{\mathcal{N}_i\to I_i}^t(c_i)$ at round t is the same with those of the best beliefs of $\mathcal{N}_i$ at round $t-1$.*

Algorithm 12: Jointly aligning and segmenting multiple Web photo streams.

Input: (1) A set of photo streams $\mathcal{P} = \{\mathcal{P}\}_{l=1}^{L}$ (interchangeably denoted by $\mathcal{I}$) with timestamps. (2) *Unsupervised*: Number of foregrounds K. *Supervised*: labeled foreground examples in some selected images.

Output: (1) As result of photo stream (PS) alignment L-partite matching graph $(\mathcal{I}, \mathcal{E}_B)$, where $\mathcal{E}_B$ include all matched pairs of images. (2) Image segmentation $\{\mathcal{F}_i\}$ for all $I_i \in \mathcal{I}$.

1: Perform feature extraction (section 8.3.1) and oversegmentation (section 8.4.2) for all $I_i \in \mathcal{I}$.

repeat

2: Define the image similarity σ: At round 1, use the histogram intersection on the two-level pyramid histograms. After that, when images are segmented, use Eq.(8.1) instead.

3: Build a photo stream graph $\mathcal{G}_P = (\mathcal{P}, \mathcal{E}_P)$, where $\mathcal{E}_P$ is the set of all pairs of nearest PS, using the method of section 8.3.4 (*i.e.* For each PS, find K_P nearest ones by using NBNN method).

4: Run multiple photo stream alignment by solving Eq.(8.3) on $\mathcal{E}_N$ (section 8.3.4). The output is an L-partite graph $(\mathcal{I}, \mathcal{E}_B)$.

5: Build an image graph $\mathcal{G}_I = (\mathcal{I}, \mathcal{E}_C)$ using the method of section 8.4.1.

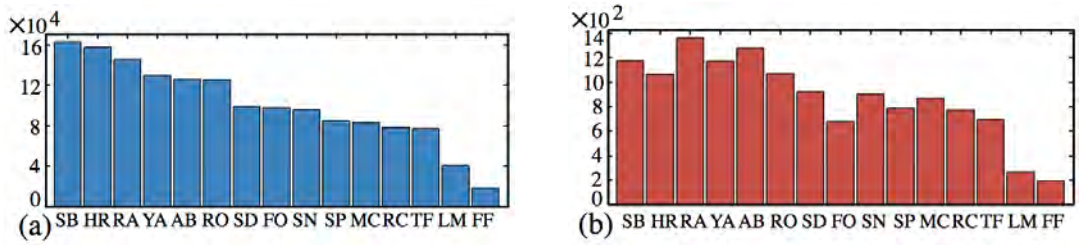
6: Run large-scale cosegmentation by solving Eq.(8.4) as described in section 8.4.2. The output is the image segmentation $\{\mathcal{F}_i\}$ for all $I_i \in \mathcal{I}$.

until $\mathcal{E}_B$ *converges or maximum iterations reach*;

Fig.8.3 shows an intuitive example of how our message passing works with this assumption. Fig.8.3.(a) shows the image I_i to be segmented at round t and its three neighbors $\mathcal{N}_i$ in image graph $\mathcal{G}_I$. We also illustrate the color-coded partitions of the best beliefs of $\mathcal{N}_i$ at round $t-1$, which are denoted by $c_{\mathcal{N}_i}^{t-1*}$. As shown in Fig.8.3.(b), when we compute the message $m_{\mathcal{N}_i \to I_i}^{t}(c_i)$, the assumption allows us to simply learn foreground models $\{v_{\mathcal{N}_i}\}$ from $c_{\mathcal{N}_i}^{t-1*}$ of Fig.8.3.(a), and to evaluate each possible c_i. By running $\{\mathcal{F}_i\} = \mathsf{RegAss}(\mathcal{S}_i, \{v_{\mathcal{N}_i}\})$, we can obtain the partition c_i^{t*} (*i.e.* the partition of the best belief of I_i at round t) as a result, which is shown in Fig.8.3.(b).

Consequently, the implementation of our message-passing based cosegmentation is straightforward; at every round, we iteratively segment each image I_i by using the learned foreground models from the partitioned regions of its neighbors $\mathcal{N}_i$ at previous round. Then, the segmented image I_i is subsequently used to learn the foreground models for its neighbors' segmentation. That is, we iteratively run foreground modeling and region assignment steps by following the edges of image graph $\mathcal{G}_I$.

Initialization: In order to proceed our iterative cosegmentation algorithm, we need initial image partitions as starting points of belief propagation. In the supervised scenario, we trivially begin from the labeled images. In an unsupervised setting, we apply the diversity ranking method of [Kim et al., 2011] to image graph $\mathcal{G}_I$ to discover a small number of central images and their neighbors. Then, the unsupervised version of MFC algorithm in [Kim and Xing, 2012] initially segments the images of each group, from which message passing begins.



SB: *surfing+beach*, HR: *horse+riding*, RA: *rafting*, YA: *yacht*, AB: *air+ballooning*, RO: *rowing*, SD: *scuba+diving*, FO: *formula+one*, SN: *snowboarding*, SP: *safari+park*, MC: *mountain+camping*, RC: *rock+climbing*, TF: *tour+de+france*, LM: *london+marathon*, FF: *fly+fishing*.

Figure 8.4: Our Flickr datasets of 15 outdoor recreational activities. The number of images and photo streams are shown in (a) and (b), respectively. The dataset sizes are (1,514,976, 13,157) in total.

8.4.3 Analysis of Algorithm

We summarize the overview of our approach in Algorithm 12. The step **2–3** describe the alignment of multiple photo streams, and the step **4–6** outline the large-scale cosegmentation. We can iterate running these two major procedures until the output converges or maximum iterations reach.

The core procedures of our approach are the two belief propagation (BP) techniques for alignment and cosegmentation. The alignment BP works on the graph of photo streams while the cosegmentation BP runs on the image graph. Generally, the BP algorithm runs in $\mathcal{O}(T|\mathcal{E}|)$ where T is the number of iterations and $|\mathcal{E}|$ is the number of edges. Since we use only sparse KNN graphs where each vertex is connected to a constant number of neighbors, the alignment BP runs in $\mathcal{O}(TL)$ and the cosegmentation BP does in $\mathcal{O}(TN)$ where L and N are the number of photo streams and images, respectively. Moreover, the BP algorithm has been studied much for parallelization [Gonzalez et al., 2009], which can further improve the speed of our algorithm.

8.5 Experiments

We evaluate the proposed approach from two technical perspectives: the photo stream alignment in section 8.5.1 and the image cosegmentation in section 8.5.2.

Flickr Dataset: Fig.8.4 summarizes our Flickr dataset that consists of 1,514,976 images of 13,157 photo streams for 15 outdoor recreational activity classes. Flickr is one of the best image sources to test our algorithm since a large number of photo streams of different users are freely available with rich associated meta-data. We use the class names as search keywords, and download all the photo streams that contain more than 50 images. We use all pictures of each photo stream without any filtering. For a quantitative segmentation evaluation, we manually annotate 100 images per class, from which we obtain approximate performance measures of algorithms. Although the labeled images are relatively few compared to dataset sizes, in practice the sampled annotation is widely adopted in standard large-scale benchmark datasets such as ImageNet [Deng et al., 2009].

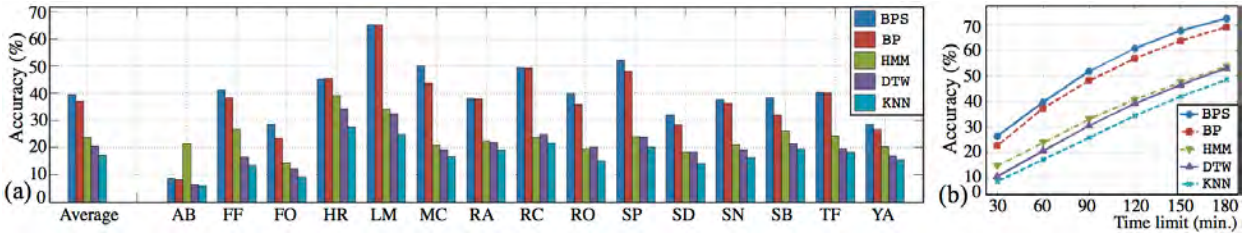


Figure 8.5: Comparison of temporal localization between our methods (BPS) and (BP) and the baselines (HMM), (DTW), and (KNN). In (a), we show the accuracies of all algorithms for 15 outdoor activity classes with $\epsilon = 60$ minutes. In (b), we show the variation of average localization accuracies by changing time thresholds ϵ from 30 minutes to 180 minutes. The acronyms of activities are referred to Fig.8.4.

8.5.1 Results on Alignment

Tasks: The performance of photo stream alignment is evaluated by a *temporal localization* task. It is inspired by the studies of geolocation estimation [Chen and Grauman, 2011; Kalogerakis et al., 2009], whose goal is to estimate the geolocations of individual pictures for a given sequence of a tourist's photos. We carry out our experiments similarly only except that the geolocation is replaced by the timestamp. We first randomly select 80% of photo streams of each class as training set and the others as test set. Then, the goal is to estimate the timestamps of all the images of the test photo streams by aligning them with training photo streams whose timestamps are known. Such temporal localization task is also important to achieve our ultimate goal, the picture-based storyline construction, which requires correctly locating each photo stream on the timeline to relate it with other photo streams.

Baselines: For the alignment tests, we compare our algorithm with four baselines. As one of the simplest baselines, the (KNN) performs image matching by using only image similarity. We also choose two alternatives of image sequence alignment. The (HMM) is the hidden Markov model method that has been widely applied for localizing tourists' photo sets [Chen and Grauman, 2011; Kalogerakis et al., 2009]. The (DTW) is dynamic time warping, one of most popular algorithms for multiple sequence alignment [Rath and Manmatha, 2003]. Our algorithm is tested in two different ways, according to whether image segmentation is in a loop or not. The (BP) does not exploit the image segmentation output whereas the (BPS) is our fully geared approach. That is, this comparison can justify the usefulness of our alternating approach between alignment and segmentation. In Table 8.1, we elaborate the application of our algorithms and baselines for the experiments.

Quantitative results: To compare the performances of algorithms, we use the similar evaluation metric to those of image geolocalization research [Chen and Grauman, 2011; Kalogerakis et al., 2009]. Given the estimated timestamps of all test images by each algorithm, we compute the percentage of images for which the estimated timestamps are within ϵ minutes of the ground-truths. Fig.8.5.(a) reports the accuracy rates of our algorithms and baselines across 15 activity classes with $\epsilon = 60$ minutes. The leftmost bar set is the average performance of 15 classes. Our algorithm significantly outperforms all the baselines in most classes. The average accuracy of our method (BPS) is 39.1%, which is notably higher than 23.7% of the best baseline (HMM). Fig.8.5.(b) compares the average accuracies of all algorithms according to different ϵ values from 30 to 180 minutes. In all ranges of ϵ, our (BPS) consistently outperforms the best baseline (HMM) by 17.1%

Method	Description
(BP) / (BPS)	Overall, our alignment algorithms are applied as described in this chapter. However, the alignment objective of Eq.(8.2) assumes that the timestamps of all photo streams are known, which is not the case for the test images in our experiments. Therefore, we use ordering information instead of the time information for Eq.(8.2). Our algorithm (BP) and (BPS) differ from each other according to whether the segmentation is in a loop or not. The (BP) does not use the image segmentation output, so we compute the image similarity from two-level spatial pyramid histograms on the whole images. On the other hand, for the (BPS), we run one complete loop of alignment and cosegmentation, and repeat the alignment again using the segmentation-based image similarity metric of Eq.(8.1).
(KNN)	For each test photo stream P^t, we first find K_T closest photo streams $\mathcal{P}^r$ from the training set by using the NBNN method in section 8.3.4. Then, for each test image $p \in P^t$, we search for K_p visually nearest images from $\mathcal{P}^r$. Finally, as an estimated timestamp of p, we compute the average of timestamps of K_p retrieved nearest images.
(DTW)	For each test photo stream P^t, we find K_T closest photo streams $\mathcal{P}^r$, as done in the (KNN). Then, we perform the pairwise alignment between P^t and each photo stream in $\mathcal{P}^r$ by using the *dynamic time warping* algorithm. Finally, as an estimated timestamp of p, we compute the average of timestamps of the images that are matched to p.
(HMM)	For each test photo stream P^t, we find K_T closest photo streams $\mathcal{P}^r$, as done in the (KNN). We run K-means clustering to the descriptors of randomly chosen images from $\{P^t \cup \mathcal{P}^r\}$, in order to define observation alphabets. By assigning the closest alphabet to each image, we represent each photo stream as a sequence of alphabets. Then, we run Baum-Welch algorithm to estimate the most likely set of HMM parameters, including the state transition matrix, the observation probability matrix, and the initial probabilities. Next, we carry out the Viterbi algorithm to find the single best state sequence for each photo stream. That is, all images in $\{P^t \cup \mathcal{P}^r\}$ are assigned to most probable state IDs. Finally, as an estimated timestamp of p, we compute the average timestamps of the training images that share the same state ID with p.

Table 8.1: Application of our algorithms ((BP) and (BPS)) and three baselines ((KNN), (DTW), and (HMM)) for the alignment evaluation.

points on average. Moreover, the accuracies of (BPS) is higher than those of (BP) by 3.6% points on average, which supports that segmentation can improve alignment.

8.5.2 Results on Segmentation

Tasks: The task of image cosegmentation is to identify frequently recurring foregrounds in the image set. The accuracy is measured by the intersection-over-union metric $(GT_i \cap R_i)/(GT_i \cup R_i)$, where GT_i is the groundtruth of image i and R_i is the estimated regions by an algorithm. It is also a standard metric in PASCAL challenge. We compute the average value of this metric from all annotated images.

Baselines: We select three baselines of unsupervised segmentation methods that can discover multiple objects from a large-scale dataset (*i.e.* at least more than tens of thousands of images). The (LDA) [Russell et al., 2006] is an LDA-based unsupervised localization method, and the

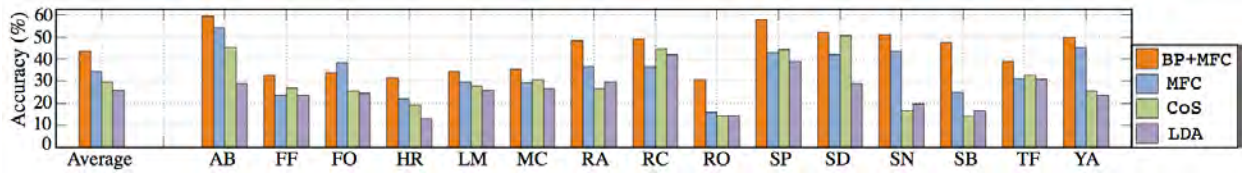


Figure 8.6: Cosegmentation accuracies between our method (BP+MFC) and the baselines (MFC), (COS), and (LDA) for 15 outdoor activities classes. The leftmost bar set shows the average accuracies. The acronyms of activities are referred to Fig.8.4.

(COS) [Kim et al., 2011] is a state-of-art cosegmentation algorithm based on submodular optimization. The (LDA) is applied to each photo stream separately. We also test the MFC algorithm (MFC) without involving the alignment step; this comparison can quantify the contribution of alignment to cosegmentation. For (COS) and (MFC), we cluster the images into multiple subgroups by K-means on visual features, and apply the methods to each subgroup independently. This decomposition improves not only segmentation accuracy but also computation speed. We run our method and all the baselines in an unsupervised manner (*i.e.* without any seed labels) for a fair comparison. Since it is hard to know the best K beforehand (*e.g.* multiple foregrounds may exist in an image), we repeat each method by changing K from one to five, and report the best results. In all baselines, we use the source codes provided by the original authors.

Quantitative results: Fig.8.6 compares the segmentation performance between our method and the three baselines. In almost all classes, the accuracies of our algorithm (BP+MFC) are far better than those of the best baselines. Especially, our average accuracy is 43.5%, which is significantly higher than 34.3% of the best baseline (MFC), which indicates that our alignment step is more successful than simple clustering such as K-means for cosegmenting extremely diverse Web user images.

Segmentation examples: Fig.8.7 shows some selected examples of cosegmentation. We observe that the subjects and their appearances are severely variable even in the images that are collected with the same keyword. For example, in the *safari+park* class, tens of different animals occur, and in all classes, people are ubiquitously shown with different appearance, poses, and clothes. Moreover, a single class may include multiple other activities; for example, the *mountain+camping* class contains the pictures of skiing, trekking, fishing, rock climbing, and hunting. Evidently, for the analysis of Web user images, it is extremely hard to pre-define the objects of interest and learn the classifiers beforehand. In contrast, our approach is greatly successful to quickly align a large-scale image set and segment out common regions in an unsupervised and bottom-up way, which can be a useful function for various Web applications.

Fig.8.8 illustrates some typical examples of failure. If the foreground consists of several distinctive regions, they can be split into multiple parts (*e.g.* multi-colored balloons and persons in the first two examples of Fig.8.8). Especially, persons may need special treatment because they are ubiquitous in almost all topics and highly variable according to clothing. The output in the other examples of Fig.8.8 is relatively reasonable but not perfect. In some cases, the foreground and background regions can be merged as a single segment if they are visually similar one another. One possible future direction to overcome these issues may be using more sophisticated region

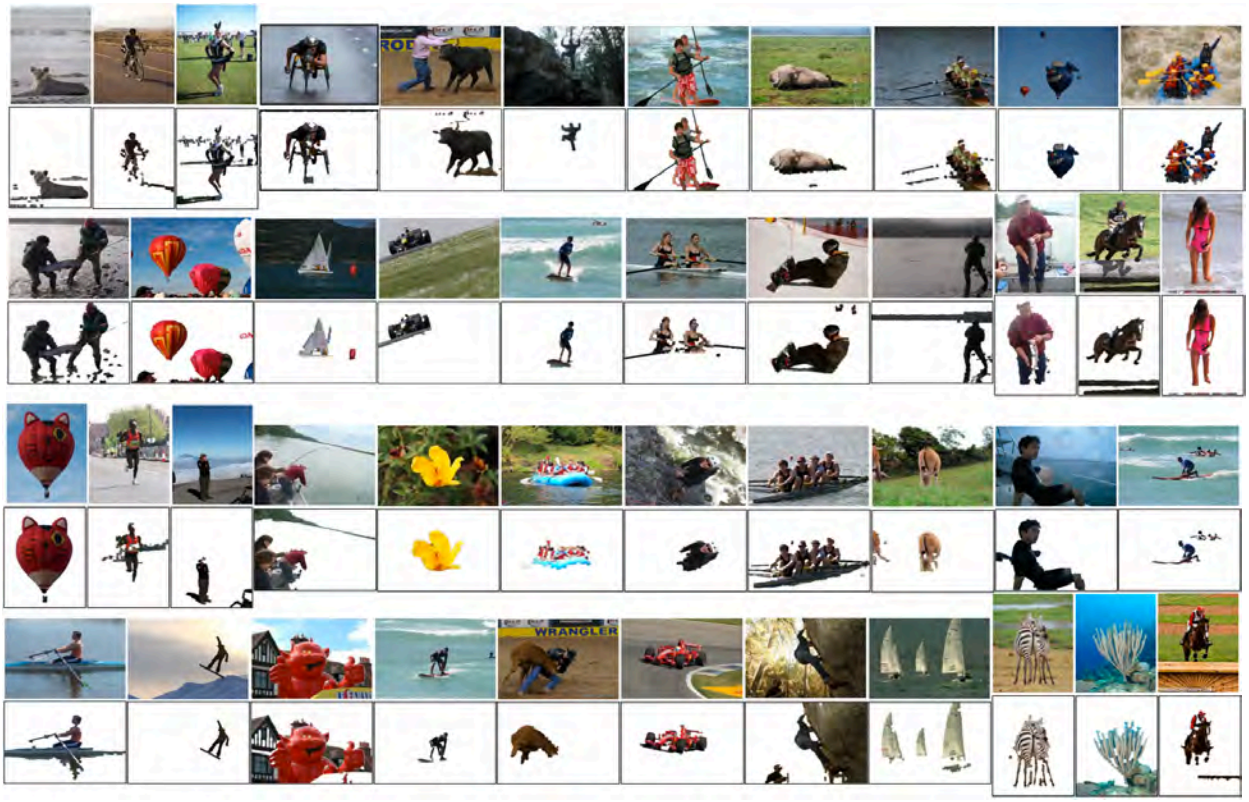


Figure 8.7: Cosegmentation examples of the Flickr outdoor recreational activity dataset.

Figure 8.8: Four typical failure cases of cosegmentation.

classifiers as foreground models. Since our approach can be regarded as an unsupervised bottom-up approach, it can by synergetic to integrate with the learned region classifiers that can provide high-level knowledge about the objects of interest.

8.5.3 Preliminary Results on Photo Storylines

In this section, we present very preliminary results of photo storyline construction, some of which are shown in Fig.8.9. We create these examples using the similar method as described in [Kim et al., 2010]. As we discussed in section 8.4.1 of the main draft, we build an image graph $\mathcal{G}_I = (\mathcal{I}, \mathcal{E}_C)$ to facilitate large-scale cosegmentation from the output of photo stream alignment. We first apply the *affinity propagation* [Frey and Dueck, 2007] to the image graph, in order to detect exemplars and clusters in the graph. Then, we find top five highest ranked clusters in every hour on the timeline. In order to compute the ranking values of clusters, we first obtain the stationary

distribution of each node (*i.e.* image) by applying PageRank algorithm to the image graph $\mathcal{G}$. Then, we compute the ranking scores of clusters as the sum of stationary distribution of the nodes in each cluster, which means the portion of time that a random walker traversing the graph stays in the cluster. Finally, each picture in Fig.8.9 is drawn by averaging the 30 nearest neighbors of each exemplar.

8.6 Summary

We propose a scalable approach to jointly aligning and segmenting multiple uncalibrated Web photo streams of different users. The empirical results assured that our approach can be key components to achieve our ultimate goal: inferring collective photo storylines from Web images. To conclude this chapter, we summarize the main contributions of this work as follows.

- We propose an approach to jointly aligning and segmenting large-scale Web photo streams of different users. Compared to previous cosegmentation research, our approach can handle any number of uncalibrated photo streams. Compared to existing image alignment research, our work can widen its applicability for reconstructing collective storylines from multiple photo streams by closing the loop with cosegmentation in a mutually rewarding way.
- We propose large-scale alignment and cosegmentation algorithms that jointly work on the whole dataset by using message-passing based optimization. The algorithms are scalable; they run in a linear time with the number of photo streams and images, respectively.
- In experiments, we evaluate the proposed approach with our new Flickr dataset of 16 outdoor activities. Our largest experiments run on more than 100K images of 1K photo streams, which exceed those of previous work by orders of magnitude. We also show the superiority of our approach over other candidate methods for both tasks.

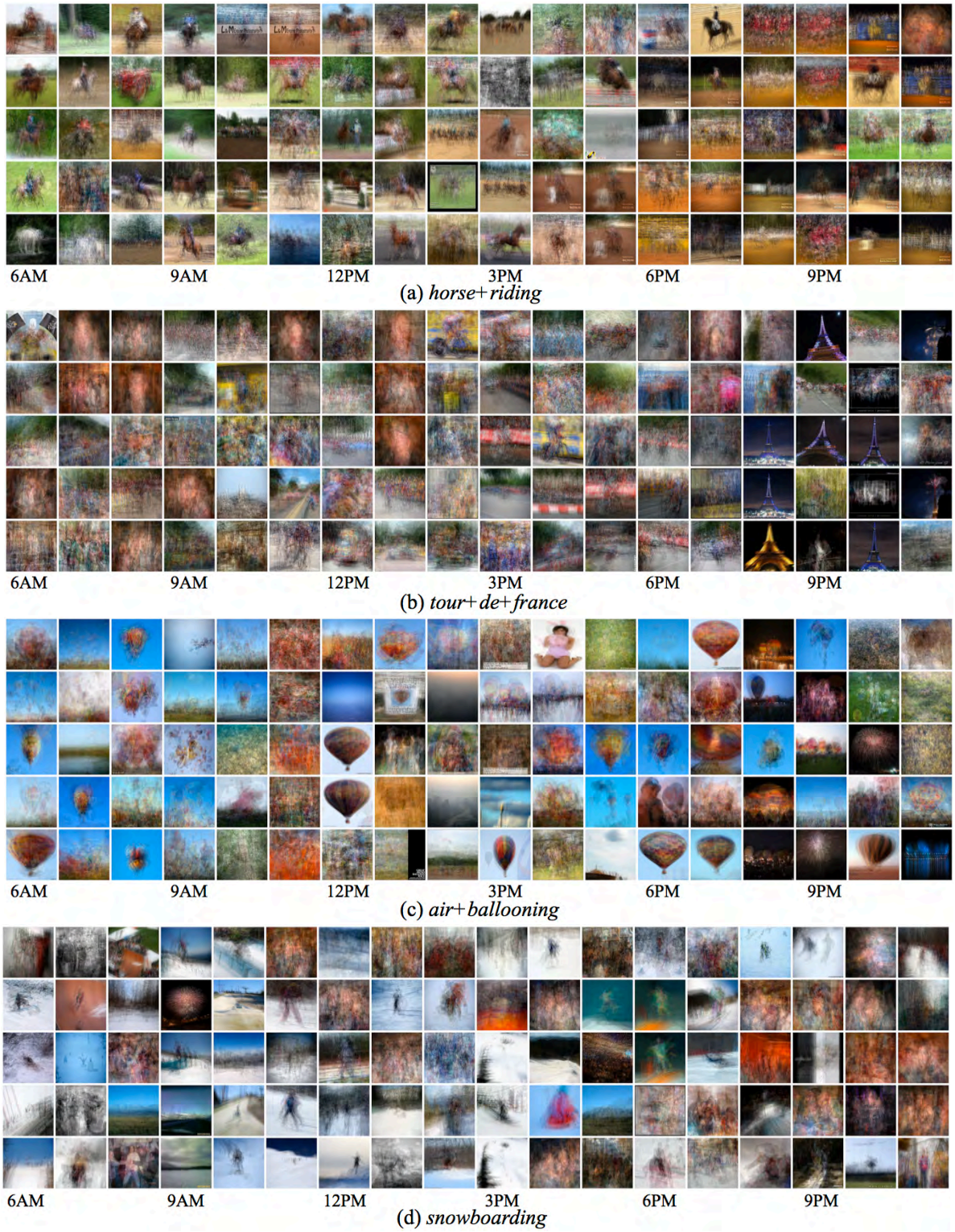


Figure 8.9: Examples of preliminary photo storyline reconstruction for three selected activity classes. Top five highest ranked image clusters are shown at every hour on the timeline. Each picture is the average of top 30 highest ranked images in each cluster.

Chapter 9

Reconstructing Photo Storyline Graphs

9.1 Introduction

The widespread access to photo-taking devices and high speed Internet has combined with rampant social networking to produce an explosion in picture sharing on web platforms. Such large-scale and ever-growing pictorial data have led to an *information overload* problem; users are often overwhelmed by the flood of pictures, and struggling to grasp various activities, events, and stories of the pictures taken by even their close friends. It is becoming increasingly more difficult but necessary to automatically summarize a large set of pictures in an efficient but comprehensive way.

In this chapter, as shown in Fig.9.1, we investigate an approach for *inferring storyline graphs* from a large set of photo streams contributed by multiple users for a particular topic (*e.g.* *independence+day*), of which a photo stream is a set of images that are taken in sequence by a single photographer within a fixed period of time (*e.g.* one day). A storyline usually refers to a series of events that have *chronological* or *causal* relations, which are commonly represented by a directed graph [Mandler and Johnson, 1977; Riedl and Young, 2006]. Likewise, our goal is to infer such

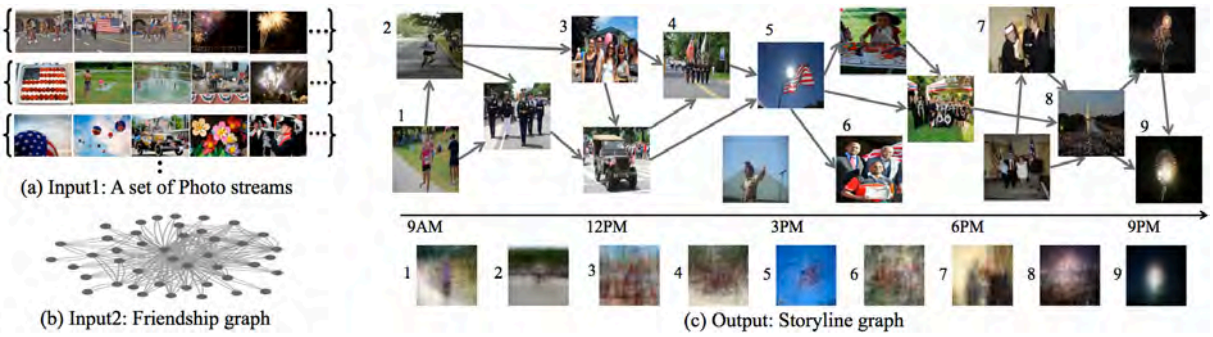


Figure 9.1: Motivation for reconstructing storyline graphs from large-scale Web photo streams with an *independence+day* example. The input is two-fold: (a) A set of photo streams that are independently taken by multiple users at different time and places, and (b) optionally a friendship graph. (c) The output is the storyline graph as a structural summary. The vertices are the exemplars of image clusters, and the edges connect sequentially recurring nodes across photo streams. We show the average images of nine selected node clusters in the bottom.

directed storyline graphs from a large set of photo streams automatically. Conceptually, the vertices in the graph correspond to dominant image clusters across the dataset, and the edges connect the vertices that sequentially recur in many photo streams. Its more rigorous definition will be developed throughout this chapter.

The representation of storyline graphs conveys several unique advantages as a structural summary of image database as follows. First, many topics of interest usually consist of a sequence of activities or events repeated across the photo streams. Some typical examples include recreational activities, holidays, and sports events. For example, various events and activities in the *independence+day* are captured by millions of people across the U.S as the sets of photo streams, which are likely to share common storylines: parades in the morning, barbeque parties in the afternoon, and fireworks at night. Such storylines can be described better by a graph of images rather than a set of independently retrieved images. Second, the storyline graph can characterize various branching narrative structure associated with the topic. The photo stream of a single user usually consists of a single linear thread of story as an image sequence on timeline. On the other hand, by aggregating multiple photo streams of different users, our algorithm can reveal various possible threads of storylines, which help users understand the underlying big picture surrounding the topic.

Note that our objective differs from the *private* storyline [Obrador et al., 2010], which is a summary of a single user's photo albums only. In this scenario, the face identification is important so that the storyline lays out in the center of herself or her close friends. Even though such private storyline is also interesting and demanding, we here explore the reconstruction of *collective* storyline graphs by leveraging all available photo sets of multiple users without identifying any particular actors. In addition, we also discuss *weakly personalized* storyline graphs, in which we leverage a friendship graph so that we weight more on the photo streams of a particular user's close friends.

We formulate the reconstruction of storyline graphs as an inference problem of sparse time-varying directed graphs. Our approach is based on the TV-DBN algorithm [Song et al., 2009], which was originally proposed to infer time-varying dynamic Bayesian networks from non-stationary biological time series such as gene expression and EEG signals. We significantly extend the TV-DBN algorithm so that we infer a directed graph from image database represented by multiple descriptors along with different types of side information. Consequently, our method enjoys several intuitively appealing properties such as optimality guarantee, linear complexity, easy parallelization, and asymptotic consistency.

For evaluation, we collect more than 3.3 millions of Flickr images of 42 thousands of photo streams for 24 topic classes. Qualitatively, we first illustrate the examples of the storyline graphs, in order to show that the proposed algorithm effectively summarizes and visualizes millions of photo streams, which are too overwhelming for a human to grasp any underlying big picture. Then, we quantitatively demonstrate that the reconstructed storylines help solve two *image sequential prediction* tasks: (i) predicting next likely pictures given a short sequence of pictures, and (ii) filling in missing parts of a photo stream. In our experiments, our approach outperforms other candidate methods such as an HMM-based model and a graph-based temporal topic modeling of chapter 3 [Kim et al., 2010]. We choose these two tasks as indirect ways to evaluate the resultant storylines due to two practical reasons. First, the storyline reconstruction of large-scale Web images is a novel problem, and thus no groundtruth of storylines is available. Second, the two prediction tasks

are directly connected to the *photo recommendation* applications, which can be regarded as one of foremost uses of storylines. For example, if a user is about to start his own *snowboarding* trip, our algorithm can preview the pictures of the most likely storylines reconstructed from the photo sets of other users, including his friends, who have experienced the *snowboarding* before. This is analogous to the Amazon's function of *Customers Who Bought This Item Also Bought*.

9.2 Problem Formulation

The input of our algorithm is two-fold. The first input is the set of photo streams of a particular topic. It is denoted by $\mathcal{P} = \{P^1, \cdots, P^L\}$, where L is the number of photo streams. Each photo stream, $P^l = \{I_1^l, \cdots, I_{N^l}^l\}$, is a set of sequential images taken by a single photographer within a period of time $[0, T]$, which is set to one day in our experiments. Therefore, the resultant storyline graph is defined in the range of $[0, T]$. Each image I_i^l is associated with owner ID u^l and timestamp t_i^l. The second input is the friendship graph $\mathcal{G}_F = (\mathcal{U}, \mathcal{E}_F)$, which is a weighted symmetric graph. The vertex set is the set of users, and the edge weights indicate the degrees of friendship.

Since the image set is too large and much of images are highly overlapped, it is inefficient to build a storyline graph over individual images. Preferentially, the vertices of storyline graphs correspond to the clusters of images that recur in the image set. We implement such *image clusters* by using the idea of encoding and decoding of neural coding [Olshausen and Field, 1997]. Conceptually, the *encoding* represents each image by a small set of codewords. Then the storyline graph is created over the codewords. The *decoding* can transform the graph over the codewords into the graph over images.

We perform the image encoding as follows. Each image I is first applied by J different image classifiers, each of which assigns a classification score $\mathbf{v}_j \in \mathbb{R}^{C_j}$ to image I. We defer the details of our J classifiers to section 9.4.1. By concatenating J scores, an image I is described by a vector $\mathbf{v}$, where $|\mathbf{v}| = \sum_{j=1}^{J} C_j$. Then we run the dictionary learning for sparse coding [Mairal et al., 2009], in order to jointly learn the dictionary of D codewords, and represent an image I by a linear combination of r best codewords while minimizing the reconstruction error. Finally, each image I is associated with a vector $\mathbf{x} \in \mathbb{R}^D$ with r nonzero elements.

The storyline graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as follows. Each node in the vertex set $\mathcal{V}$ corresponds to a codeword (*i.e.* $|\mathcal{V}| = D$), and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ includes directed edges between them. In our approach, we let the storyline graph be *sparse* and *time-varying* [Kolar et al., 2010; Song et al., 2009]. The sparsity is encouraged in order to avoid an unnecessarily complex storyline graph in which any images can follow any images. The time-varying graph means that we allow $\mathcal{E}^t$ to smoothly change over time in $t \in [0, T]$. It is based on the fact that the popular transition between image codewords can vary over time; for example, in the *scuba+diving* class, the *underwater* images may be followed by *lunch* images around noon but *sunset* images in the evening.

Consequently, the output of our algorithm is a set of storyline graphs $\{\mathbf{A}^t\}$ for $t \in [0, T]$, where $\mathbf{A}^t$ is the adjacency matrix of $\mathcal{G}^t$. Although we can compute $\mathbf{A}^t$ at any point t, in practice, we uniformly split $[0, T]$ into τ time points (*e.g.* every 30 minutes), at which the $\mathbf{A}^t$ is estimated.

Finally, the decoding step retrieves the most suitable images for the transitions between (sets of) codewords for a given $\mathbf{A}^t$ at time t. We adopt the approach of continuous error-correcting

output codes (ECOC) [Crammer and Singer, 2002], with the histogram intersection as the decoding metric. A codeword or its combination of $\mathbf{A}^t$ can be represented by $\mathbf{h} \in \mathbb{R}^D$, and thus we can rank images near t by calculating $\sum_{d=1}^{D} \min(\mathbf{h}_d, \mathbf{x}_d)$ (*i.e.* sum of the element-wise minimum).

9.3 Estimating Photo Storyline Graphs

By following the general procedure of the graph learning, we first perform *structure learning* to discover the topology of the storyline graph, and then carry out *parameter learning* while fixing the topology of the graph. Mathematically, the former is to identify the nonzero elements of $\{\mathbf{A}^t\}$, and the latter is to estimate their associated weights.

For statistical tractability and scalability, our algorithm builds on four assumptions about photo streams that are reasonable in practice. Three of them are introduced in the following, and the fourth one is presented later in this section. (A1) All photo streams are assumed to be taken independently of one another. (A2) We employ the k-th order Markovian assumption between the consecutive images in the photo stream. (A3) The graph is sparse and varies smoothly across time.

As a result of image encoding, each image I_i is associated with a descriptor vector $\mathbf{x}_i \in \mathbb{R}^D$. Thus, we denote a photo stream by $P^l = \{(\mathbf{x}_1^l, t_1^l), \ldots, (\mathbf{x}_{N^l}^l, t_{N^l}^l)\}$. We begin our model by deriving the likelihood $f(\mathcal{P})$ of an observed set of photo streams $\mathcal{P} = \{P^1, \cdots, P^L\}$. Based on the assumption (A1) and (A2), the likelihood $f(\mathcal{P})$ is defined as follows[1].

$$f(\mathcal{P}) = \prod_{l=1}^{L} f(P^l), \quad \text{where } f(P^l) = f(\mathbf{x}_1^l, t_1^l) \prod_{i=2}^{N^l} f(\mathbf{x}_i^l, t_i^l | \mathbf{x}_{i-1}^l, t_{i-1}^l) \tag{9.1}$$

where $f(\mathbf{x}_i^l, t_i^l | \mathbf{x}_{i-1}^l, t_{i-1}^l)$ is the conditional likelihood of consecutive occurrence from image $\mathbf{x}_{i-1}^l$ at time t_{i-1}^l to $\mathbf{x}_i^l$ at t_i^l in the photo stream l. Our forth assumption is imposed on the transition model. (A4) The codewords of $\mathbf{x}_i^l$ are conditional independent one another given $\mathbf{x}_{i-1}^l$. In other words, the transition likelihood factors over individual codewords: $f(\mathbf{x}_i^l, t_i^l | \mathbf{x}_{i-1}^l, t_{i-1}^l) = \prod_{d=1}^{D} f(x_{i,d}^l, t_i^l | \mathbf{x}_{i-1}^l, t_{i-1}^l)$. As a simple transition model $f(\mathbf{x}_i^l, t_i^l | \mathbf{x}_{i-1}^l, t_{i-1}^l)$, we use a *linear dynamics model*

$$\mathbf{x}_i^l = \mathbf{A}_e \mathbf{x}_{i-1}^l + \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \tag{9.2}$$

where $\boldsymbol{\epsilon}$ is a vector of Gaussian noise with zero mean and variance σ^2. In order to encode temporal information between t_{i-1}^l and t_i^l into $\mathbf{A}_e \in \mathbb{R}^{D \times D}$, we use two parametric rate models, the *exponential* and the *Rayleigh* model, which have been widely used to represent temporal dynamics of diffusion networks [Rodriguez et al., 2011]. With $\Delta = t_i^l - t_{i-1}^l$, the (x, y) element a_{xy} of $\mathbf{A}_e$ is defined as follows.

$$a_{xy} = \begin{cases} \alpha_{xy} \exp(-\alpha_{xy} \Delta) & \text{(Exponential)} \\ \alpha_{xy} \Delta \exp(-\alpha_{xy} (\Delta^2 / 2)) & \text{(Rayleigh)} \end{cases} \tag{9.3}$$

[1] Here we use the first-order Markovian assumption for simplicity of our discussion. Extending to the k-th order Markovian assumption is straightforward, and will be discussed later.

where α_{xy} is the transmission rate from codeword x to y. Since we are interested in time-varying graphs, the α_{xy} is a function of time t^l_{i-1}. But, for simplicity, we here let α_{xy} stationary, and its dynamics will be discussed in next section. Note that $\alpha_{xy} \geq 0$. As $\alpha_{xy} \to 0$, the consecutive occurrence from codeword x to y is very unlikely. By plugging Eq.(9.3) into Eq.(9.2), and letting $\mathbf{A} = \{\alpha_{xy} \exp(\alpha_{xy})\}_{D \times D}$, the transition model of Eq.(9.2) reduces to

$$\mathbf{x}^l_i = g_i \mathbf{A} \mathbf{x}^l_{i-1} + \boldsymbol{\epsilon}, \quad \text{where } g_i = \begin{cases} \exp(-\Delta) & \text{(Exponential)} \\ \Delta \exp(-(\Delta^2/2)) & \text{(Rayleigh)} \end{cases} \tag{9.4}$$

From Eq.(9.4), we can express the transition likelihood in the form of Gaussian distribution: $f(x^l_{i,d}, t^l_i | \mathbf{x}^l_{i-1}, t^l_{i-1}) = \mathcal{N}(x^l_{i,d}; g_i \mathbf{A}_{d*} \mathbf{x}^l_{i-1}, \sigma^2)$, where $\mathbf{A}_{d*}$ denotes the d-th row of the matrix $\mathbf{A}$. Finally, the log-likelihood $\log f(\mathcal{P})$ in Eq.(9.1) can be written

$$\log f(\mathcal{P}) = \sum_{l=1}^{L} \sum_{i=2}^{N^l} \sum_{d=1}^{D} \left(-\frac{N^l}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} (x^l_{i,d} - g_i \mathbf{A}_{d*} \mathbf{x}^l_{i-1})^2 \right) \tag{9.5}$$

9.3.1 Optimization

In this section, we discuss the optimization method to discover nonzero elements of $\mathbf{A}^t$ for any $t \in [0, T]$, by maximizing the log-likelihood of Eq.(9.5). One difficulty here is that for a fixed t, the learning data (*i.e.* images occurring at a particular t) may be scarce, and thus the estimator may suffer from extremely high variance. To overcome this difficulty, we take advantage of the assumption (A3), which allows to estimate $\mathbf{A}^t$ by re-weighting the observation data near t accordingly. Furthermore, to make the estimation problem trivially parallelizable, we adopt the assumption (A4), which let us to separately perform an optimization for each codeword d ($d = 1, \ldots, D$). (This approach is known as *neighborhood selection* in graph inference literature [Meinshausen and Bühlmann, 2006]). As a result of the two assumptions, we iteratively solve the following optimization problem D times:

$$\widehat{\mathbf{A}}^t_{d*} = \operatorname{argmin} \sum_{l=1}^{L} \sum_{i=2}^{N^l} w^t(i) (x^l_{i,d} - g_i \mathbf{A}^t_{d*} \mathbf{x}^l_{i-1})^2 + \lambda \|\mathbf{A}^t_{d*}\| \tag{9.6}$$

where $w^t(i)$ is the weighting of an observation of image I_i in photo stream l at time t. That is, when the timestamp of image I_i (*i.e.* t^l_i) is close to t, $w^t(i)$ is large so that the observation contributes more on the graph estimation at t. Naturally, we can define

$$w^t(i) = \frac{K_h(t - t^l_i)}{\sum_{l=1}^{L} \sum_{i=2}^{N_l} K_h(t - t^l_i)}, \quad \text{where } K_h(u) = \frac{\exp(-u^2/2h^2)}{\sqrt{2\pi}h} \tag{9.7}$$

where $K_h(u)$ is a Gaussian symmetric nonnegative kernel function and h is the kernel bandwidth.

In Eq.(9.6), we include ℓ_1-regularization for a sparse graph structure, where λ is a parameter that controls the sparsity of $\widehat{\mathbf{A}}^t_{d*}$. This approach not only avoids overfitting but also is practical

Algorithm 13: Inferring the topology of storyline graphs.

Input: (1) A set of photo streams $\mathcal{P} = \{P^1, \ldots, P^L\}$. (2) A friendship graph $\mathcal{G}_F = (\mathcal{U}, \mathcal{E}_F)$ (3) ℓ_1 penalty λ.

Output: (1) Time-varying storyline graph $\{\mathbf{A}^1, \ldots, \mathbf{A}^T\}$.

1: Randomly initialize $\mathbf{A}^0$.
foreach $d = 1 \ldots, D$ **do**
  foreach $t = 1 \ldots T$ **do**
    2: Initialize $\mathbf{A}^t_{d*} \rightarrow \mathbf{A}^{t-1}_{d*}$.
    3: Compute $w^t(i)$ using Eq.(9.8) for the personalized storyline or Eq.(9.7) otherwise.
    4: Scale $\tilde{x}^{l\,(d)}_{i+1} \leftarrow \sqrt{w^t(i)} x^{l\,(d)}_{i+1}$, $\tilde{\mathbf{x}}^l_i \leftarrow \sqrt{w^t(i)}\mathbf{x}^l_i$ for all $i = 1, \ldots, M_l - 1$ and $l = 1, \ldots, L$
    while $\mathbf{A}^t_{d*}$ *does not converge.* **do**
      foreach $j = 1 \ldots, D$ **do**
        5: $S_j \leftarrow \sum_{l=1}^{L} \frac{2}{T} \sum_{t=1}^{T} (\sum_{k \neq j} \mathbf{A}^t_{dk} \tilde{x}^{l\,(d)}_{i+1} - \tilde{x}^t_d) \tilde{x}^{t-1}_j$. $b_j \leftarrow \frac{2}{T} \sum_{t=1}^{T} (\tilde{x}^{t-1}_j)^2$.
        6: $\mathbf{A}^t_{dj} \leftarrow (\text{sign}(S_j - \lambda)\lambda - S_j)/b_j$, if $|S_j| > \lambda$, otherwise 0.

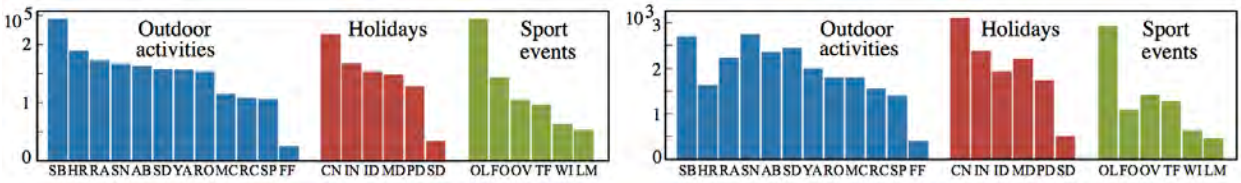
because the branches of storylines at each node are simple enough to be easily understood. Consequently, our graph inference reduces to iteratively solving a weighted standard ℓ_1-regularized least square problem, whose global optimum solution can be attained by highly scalable techniques such as shooting algorithm [Fu, 1998]. Algorithm 13 summarizes the overall procedure. Since the graphs smoothly change over time, we can use the warm start for further speedup; $\mathbf{A}^1$ is used as an initialization for $\mathbf{A}^2$.

It is straightforward to extend the above optimization to the k-th order Markovian assumption. Simply, Eq.(9.4) is extended to an autoregressive model with the k-th order (*i.e.* $\mathbf{x}^l_i = \sum_{q=1}^{k} g_i(q)\mathbf{A}(q)\mathbf{x}^l_{i-q} + \boldsymbol{\epsilon}$), and the square loss function of Eq.(9.6) is changed accordingly.

The graph inference can be performed in a linear time with respect to all parameters, including the number of images and the number of codewords D. Our MATLAB code takes less than one hour to obtain the set of $\{\mathbf{A}\}$ for 245K images of the *surfing+beach* topic with $D = 1,000$ and $\tau = 40$.

We can prove the asymptotic statistical consistency of the graph estimation procedure in Algorithm 13, which guarantees that true graph can be discovered as the number of data points increases indefinitely [Song et al., 2009]. Its detailed proof can be found in [Kolar and Xing, 2013].

Once $\{\mathbf{A}\}$ is discovered, the *parameter learning* updates the associated weights of nonzero entries of each $\mathbf{A}^t \in \{\mathbf{A}\}$, while unchanging zero elements. Since the structure of each graph is known and observations are independent one another from (A1) and (A4), we can trivially solve the maximum likelihood estimation of $\widehat{\mathbf{A}}^t$, which is similar to that of the transition matrix of k-th Markovian chains. For example, the MLE of $\widehat{\mathbf{A}}^t_{xy}$ with the first-order Markovian assumption is the fraction of observed transitions from x to y at time step t.



Outdoor activities (12): SB (*surfing+beach*), HR (*horse+riding*), RA (*rafting*), SN (*snowboarding*), AB (*air+ballooning*), SD (*scuba+div- ing*), YA (*yacht*), RO (*rowing*), MC (*mountain+camping*), RC (*rock+climbing*), SP (*safari+park*), FF (*fly+fishing*). **Holidays** (6): CN (*chinese +new+year*), IN (*inauguration*), ID (*independence+day*), MD (*memorial+day*), PD (*st+patrick+ day*), ES (*easter+sunday*). **Sports events** (6): OL (*olympic+london*), FO (*formula+one*), OV (*olympic+vancouver*), TF (*tour+de+france*), WI (*wimbledon*), LM (*london+marathon*).

Figure 9.2: The Flickr datasets of 24 classes of three categories. The number of images and photo streams are shown in (a) and (b), respectively. The dataset sizes are (3,320,080, 42,744) in total.

9.3.2 Incorporating Side Information

When side information is available such as a friendship graph, GPS data, and other types of temporal information, we can customize the storyline graphs accordingly. For example, given a particular user u_q, the storyline graph can be recast by weighting more the photo streams of u_q's neighbors in the friendship graph $\mathcal{G}_F$. Another example is a season-specific storyline graph, given that the popular activities or events of outdoor activities (*e.g. fly+fishing*) would change much from summer to winter. We utilize the *product kernel* as a unified framework to incorporate such side information for graph inference. For example, if a particular user u_q and a month m_q is given, the weighting function of Eq.(9.6) is replaced by

$$w^t(i, u_q, m_q) = \frac{K_h(t - t_i^l)K_s(m_q - m_i^l)K_u(\rho(u_q, u^l))}{\sum_{l=1}^{L}\sum_{i=2}^{N^l} K_h(t - t_i^l)K_s(m_q - m_i^l)K_u(\rho(u_q, u^l))} \tag{9.8}$$

where $\rho(u_q, u_i^l)$ is the distance between user u_q and u_i^l in the friendship graph. As the user distance, we use the inverse of the score of *random walk with restart* [Sun et al., 2005]. Consequently, the kernel weighting technique in Eq.(9.8) is flexible; we can easily extend the product kernel by including other continuous side information to enforce the smooth variation effect.

9.4 Experiments

In our experiments, we qualitatively present some examples of reconstructed storyline graphs, and quantitatively evaluate its usefulness to perform two sequence prediction tasks.

9.4.1 Evaluation Setting

Flickr Dataset: Fig.9.2 summarizes our Flickr dataset that consists of about 3.3M of images of 42K photo streams for 24 classes, which are classified into three categories: outdoor recreational activities, holidays, and sports events. We use the topic names as search keywords and download all queried photo streams of more than 30 images with correct timestamps and user information.

Since Flickr does not officially provide any friendship graphs between users, we indirectly build from user information. We crawl the list of groups each user is a member of, using the Flickr API. Then we connect a pair of users if they are the members of the same group. Of the friendship graph $\mathcal{G}_F = (\mathcal{U}, \mathcal{E}_F)$, the edge weight indicates the number of groups that both users join together.

Image description: In our experiments, we use four different image description methods to capture various visual information of an image. We first extract three types of image features denoted by (`SIFT`), (`HOG`), and (`Tiny`). The (`SIFT`) and the (`HOG`) indicates the three-level spatial pyramid histograms of HSV color SIFTs and HOG features, respectively. We use the code provided by [Xiao et al., 2010]. The (`Tiny`) is the RGB values of 32×32 resized tiny images as proposed in [Torralba et al., 2008]. Using the soft vector quantization, for each of three feature types, we construct $C_j(= 300)$ image clusters by applying K-means to randomly sampled image features, and then each image I is assigned to the c nearest image clusters with Gaussian weighting. In addition, we also use the scores of linear one-vs-all SVM classifiers for 397 scene categories of the SUN dataset [Xiao et al., 2010], which can convey a meaningful high-level description of an image since much of Web images contain scenes. Finally, we apply the dictionary learning in section 9.2 over the four descriptor vectors. Note that our graph inference algorithm is independent on the numbers and types of image description methods.

Tasks: The quantitative evaluation on the reconstructed storyline graphs is inherently difficult because the storyline graphs have no available groundtruth and the evaluation by human labelers can be subjective. Therefore, we instead demonstrate that the storylines reconstructed by the proposed algorithm can improve the performance of the two *image sequence prediction* tasks over other candidate methods. The two tasks are (I) predicting next likely images and (II) filling in missing parts of a photo stream. These two tasks are chosen based on the assumption that one foremost practical use of storylines should be the *photo recommendation*. For example, if we have a pictorial summary of what people usually do during *fly+fishing* from millions of images, we can recommend a part of their experiences to a user who is about to start her own *fly+fishing*.

For experiments, we first randomly select 80% of photo streams of each class as a training set and the others as a test set. For the task (I), we randomly divide each test photo stream into two disjoint parts. Then, given the first part of the photo stream and next 20 query time points $\mathbf{t}_q = \{t_{q1}, \ldots, t_{q20}\}$, we predict the likely images at $\mathbf{t}_q$. Likewise, for the task (II), we randomly crop out a portion of images in the middle of each test photo stream. Then, our goal is to predict the likely images for the missing part given its time points $\mathbf{t}_q$. We also perform the tests for weakly personalized storyline graphs; the tests are the same only except that a pair of query user and month (u_q, m_q) is specified. In this setting, test photo streams to be predicted are taken by user u_q at m_q, and the algorithms can leverage month data of photo streams and a friendship graph. Consequently, we examine more than 20K test instances in total to evaluate the performance of our algorithm.

The performance measures of both tasks are obtained as follows. Obviously, the actual image at each $t_q \in \mathbf{t}_q$ that is removed from the test photo stream is a positive test image I_P (*i.e.* groundtruth). We randomly sample 10 images from the other test photo streams as negative test images $\mathcal{I}_N$. The goal of each algorithm is to assign scores to $I_P \cup \mathcal{I}_N$, from which average precisions are computed. Ideally, the algorithm is supposed to rank I_P the first against distracting $\mathcal{I}_N$.

Method	Description
Our algorithm	We build the storyline graph by applying the proposed algorithm to the training photo streams only. As a result of, we obtain a set of $\{\mathbf{A}^t\}$ at each time point t, which can be regarded as a transition matrix between codewords. Let P_q be a test photo stream, which consists of known images $\mathcal{I}_g$ and unknown images $\mathcal{I}_q$ to be estimated. We also build the transition matrix $\mathbf{A}_g$ using the $\mathcal{I}_g$. For each query image $I_q \in \mathcal{I}_q$ of the P_q, we choose the $\mathbf{A}^t$ that is the closest to t_q, and perform the inference algorithms using both $\mathbf{A}_t$ and $\mathbf{A}_g$. We use the forward algorithm for the task (I) and the forward-backward algorithm with EM iterations for the task (II), since the observations in the middle of photo streams are missing.
(PAGE)	This is a Page-Rank based image retrieval without using any structural information. For each $I_q \in \mathcal{I}_q$ of the P_q, we first sample the training images $\mathcal{I}_{t_q}$ whose timestamps are within $[t_q \pm \delta]$. Then, we build a similarity graph between $\mathcal{I}_{t_q} \cup \mathcal{I}_g \cup I_q \cup \mathcal{I}_N$, and compute ranking scores $I_q \cup \mathcal{I}_N$ using the PageRank algorithm.
(HMM)	For each test photo stream P_q, we first find the training photo streams that are sufficiently overlapped with P_q on timeline. Then, we apply the HMM learning to estimate the most likely set of HMM parameters, including the state transition matrix, the observation probability matrix, and the initial probabilities. Similarly to our algorithm, we can use the forward algorithm for the task (I) and the forward-backward algorithm with EM iterations for the task (II).
(NET)	The basic idea of temporal topic modeling for Web images in [Kim et al., 2010] is to distribute the images on the timeline, and build a large similarity graph by connecting visually similar and temporally close images. We first build such an image graph, and then count which codewords of images consequently occur near the query time point $\mathbf{t}_q$, from which we can compute the transition matrix between codewords, denoted by $\mathbf{A}_{net}^{\mathbf{t}_q}$. Next, we can run the same inference algorithm with the (HMM) to perform the two prediction tasks.

Table 9.1: Application of our algorithm and three baselines (PAGE), (HMM), and (NET).

Our approach and Baselines: We simply outline the underlying rationale of our algorithm and three baselines in the following. The details of their application are summarized in Table 9.1. The three baselines that we compare with our method are as follows. The (Page) is a Page-Rank based image retrieval without using any structural information. It is compared to show that the importance of sequential structure modeling. The (HMM) is an HMM based method that has been popularly applied for modeling tourists' sequential photo sets [Chen and Grauman, 2011; Kalogerakis et al., 2009]. The (NET) is a temporal topic modeling method for Web image collections [Kim et al., 2010]. The (HMM) and the (NET) were not originally developed for the storyline reconstruction, but they are appealing candidate methods to visualize the topic evolution of image collections and perform the sequence prediction tasks. In our algorithm, we build the storyline graphs by applying the proposed algorithm to the training photo streams only. Likewise, (HMM) and (NET) learn their own models from the same training data. For the two prediction tasks, all of them use similar forward-backward algorithm with EM iterations, as shown in Table 9.1.

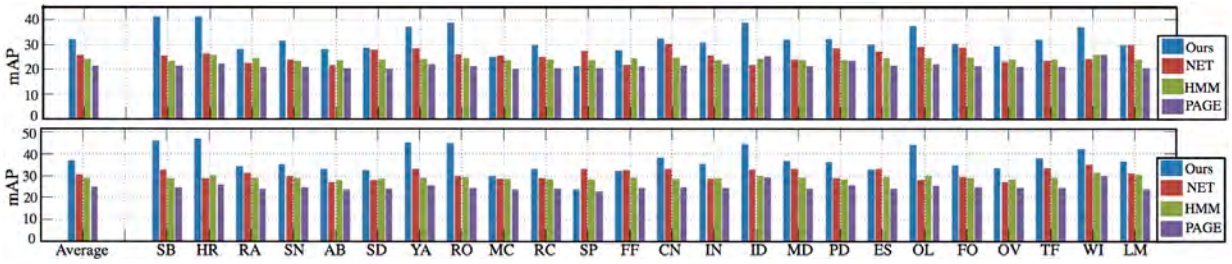


Figure 9.3: Comparison between our method and three baselines for the task (I) in the top (*i.e. predicting likely next images*) and the task (II) in the bottom (*i.e. filling in missing parts*). The average mAP (%) in the left-most bar set are ours (32.0, 36.8), `NET` (25.6, 30.4), `HMM` (23.9, 28.8), `Page` (21.4, 24.5).

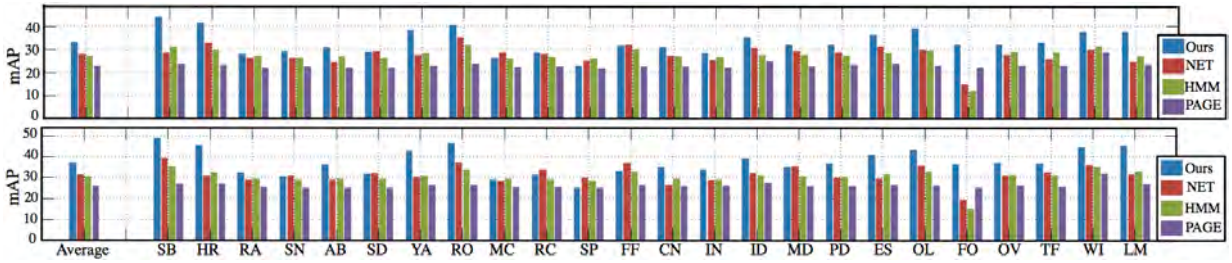


Figure 9.4: Comparison of the *weakly supervised* prediction for the task (I) in the top (*i.e. predicting likely next images*) and the task (II) in the bottom (*i.e. filling in missing parts*). The average mAP (%) in the left-most bar set are ours (33.1, 37.3), `NET` (27.8, 31.5), `HMM` (27.4, 30.4), `Page` (23.0, 26.2).

9.4.2 Results on Storyline graphs

Quantitative Results: Fig.9.3 and Fig.9.4 show the quantitative comparison between our method and three baselines for the normal and weakly personalized prediction, respectively. In each figure, we report the results of task (I) in the top and task (II) in the bottom. The leftmost bar set is the average performance of 24 classes, and the mean average precision (mAP) of all 24 classes follow. Our algorithm significantly outperforms all the competitors in most topic classes for the both tasks. In the average accuracy of normal prediction, our mAP values are higher by 6.4% and 6.3% points than the best baseline (`NET`) for the task (I) and task (II), respectively. In the average accuracy of weakly personalized prediction, our method also outperforms the best baseline (`NET`) by 5.3% and 5.9% points for the two tasks. We observe that the performance of the (`Page`) is the worst since it does not take advantage of any structural information. In almost all algorithms, the accuracies of the task (II) are higher than those of the task (I), since we can leverage the given data of test photo streams before and after the missing part. Interestingly, the weakly personalized prediction leads only a slight increase of prediction accuracies. It may be because the photo-taking styles of the users in neighborhood are not always similar one another, given that our friendship graph is built from users' Flickr group memberships, and thus most of them are likely to be professional photographers.

Examples of storyline graphs: Fig.9.5 shows two examples of storyline graphs for the *fly+fishing* class. We present them on the time horizon of 12 hours, although we can freely choose the tempo-

Figure 9.5: Examples of *weakly personalized* storyline graphs for the *fly+fishing* class with two different users. We show the central images of 12 selected clusters in the bottom.

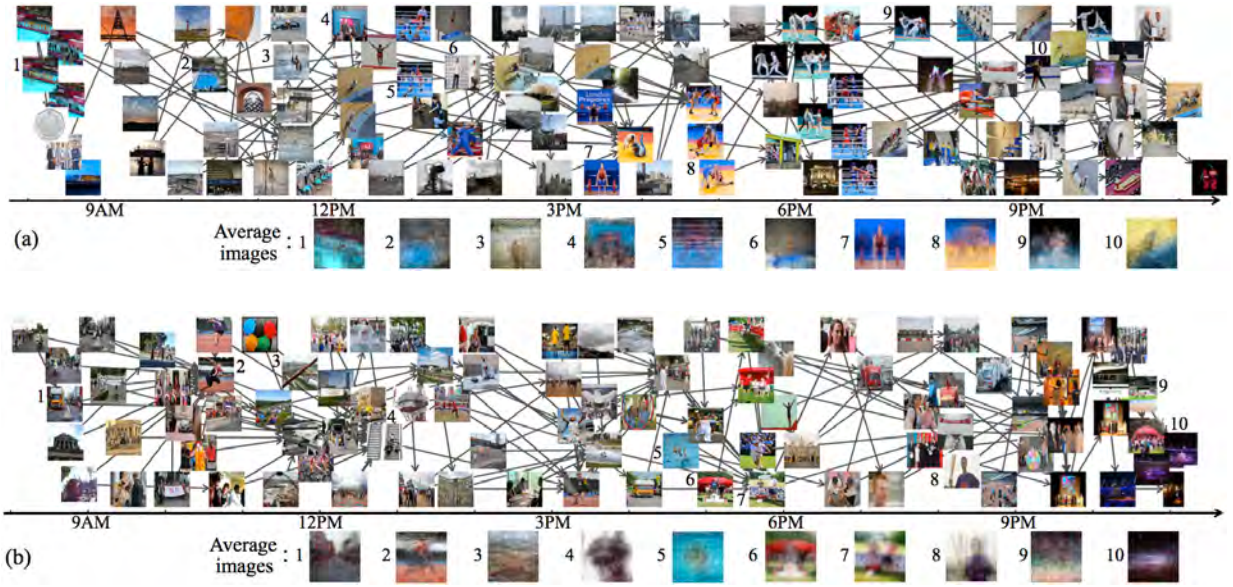


Figure 9.6: Examples of storyline graphs for the *olympic+london* class with two different months: *August* in the top and *May* in the bottom.

ral granularity to zoom in or out the storylines. We first select a fixed number of the most dominant image clusters, each of which is represented by its exemplar (*i.e.* cluster center). A fixed number of edges are chosen among the strongest edges between the selected image clusters from the $\mathbf{A}^t$ at each corresponding time point. The two storyline graphs in Fig.9.5 are *weakly personalized* ones for two different users at two different months. Interestingly, they share similar objects, activities,

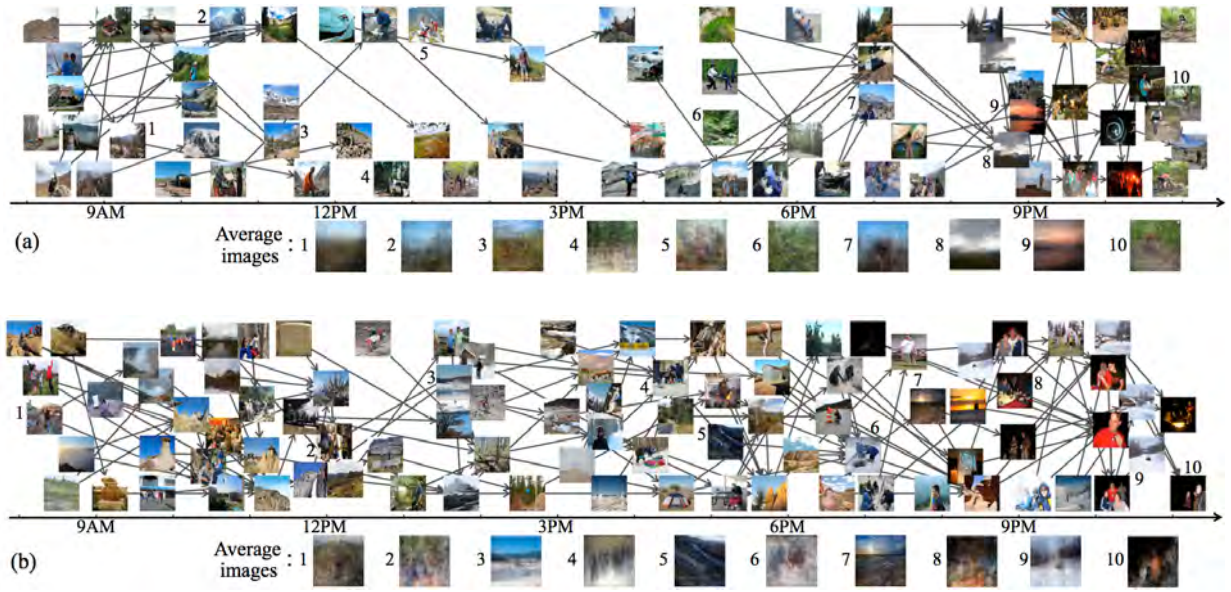


Figure 9.7: Examples of storyline graphs for the *mountain+camping* class with two different months: *September* in the top and *February* in the bottom.

and scenes at the *basic*-level (*e.g.* birds, boats, fish, mountains, and rivers), while they differ at the *subordinate*-level (*e.g.* different fish species at different places by different people with unique styles and preferences).

Fig.9.6 shows the two storyline graphs for the *olympic+london* class with two different months: $m_q =$ *August* in the top and $m_q =$ *May* in the bottom. Since the London Olympic were held from July to August in 2012, the graph for $m_q =$ *August* depicts a variety of sports events that actually occurred during the Olympic. On the other hand, most images of the graph for $m_q =$ *May* were taken after the Olympic event, but they still show the contents that are strongly associated with the Olympic such as sports, stadiums, and other related activities.

Fig.9.7 shows two examples of storyline graphs for the *mountain+camping* class with two different seasons: $m_q = September$ (Summer) in the top and $m_q = February$ (winter) in the bottom. The *mountain+camping* is one of typical examples in the category of outdoor recreational activities, which show dramatic variations of popular activities or scenes according to seasons. However, for more accurate storyline reconstruction, it would be encouraging to include spatial information like GPS data since the scenic views and weather vary much according to the places where the pictures are taken. For example, snow can be observed in some places even in summer, and the season is reversed in the northern and the southern hemisphere.

Fig.9.8 shows two typical examples of coherent and stationary topics: the *safari+park* class in the top and the *rafting* class in the bottom. The storylines are roughly similar no matter when they are taken by whom. In the *safari+park* storylines, almost all pictures are close-ups of various animals. In the *rafting* storylines, most of images depict similar activities on boats in the river, because the rafting is a rather standardized recreation.

Fig.9.9 shows the variation of popular transitions from the same codeword according to time and season. At four different time points from spring to winter, we illustrate the pictures of the

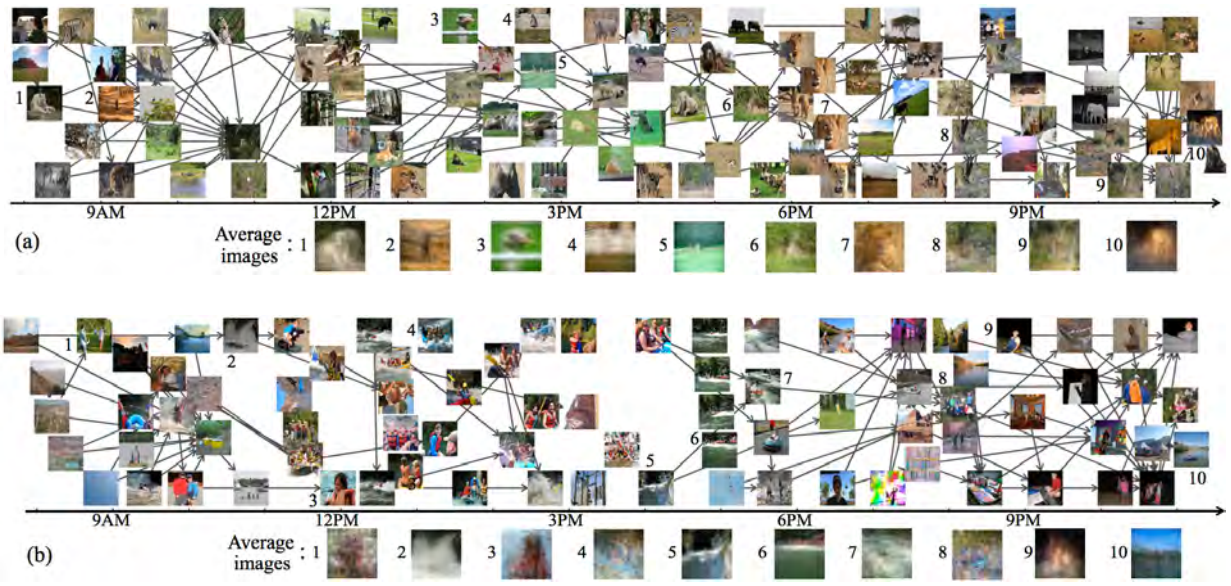


Figure 9.8: Examples of storyline graphs for the *safari+park* in the top and the *rafting* in the bottom.

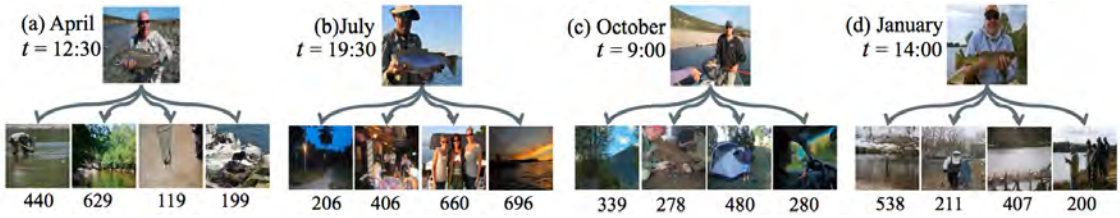


Figure 9.9: Variation of popular transition from the codeword 205 of the *fly+fishing* at four different time points from (a) spring to (d) winter. We find out the four most likely next codewords per time point, and sample one image from each numbered codeword.

four most likely next codewords from the word 205 of the *fly+fishing* (*i.e.* a man with fish). The popular transitions change dramatically; for example, in winter (Fig.9.9.(d)), the next likely codewords are the ones with snowy background, which differ much from those of the other seasons. At late evening in Fig.9.9.(b), the same codeword is likely to be followed by the sunset or dinner codewords. These examples justify that the time-varying nature of storyline graphs is important in practice, and our algorithm can correctly capture such temporal variations.

9.5 Summary

We propose an approach for reconstructing storyline graphs from large-scale community photos available on the Web. Our empirical results validate that the storyline graph provides an effective structural summary of large image collections, which otherwise are hardly understandable for users. We also qualitatively show the usefulness of storyline graphs for the two prediction tasks. To conclude this chapter, we summarize the main contributions of this work as follows.

- To the best of our knowledge, our work is the first attempt so far to address the reconstruction

of storyline graphs from large-scale community photo collections, especially for the topics of recreational activities, holidays, and sports events. Our method delivers a novel structural summary, which can not only visualize various events or activities associated with the topic in a form of branching networks, but also potentiate Web services such as image prediction and recommendation.

- We develop an inference algorithm for sparse time-varying directed graphs from photo streams with optionally other side information. Our approach achieves several key challenges of Web-scale storyline reconstruction, including global optimality, asymptotic consistency, linear complexity, and easy parallelization. With experiments on more than 3.3 millions of images of 24 classes, we show that the proposed algorithm is more successful for the two structural prediction tasks over other candidate methods.

Chapter 10

Visualizing Brand Associations from Web Photos

10.1 Introduction

Brand equity describes a set of values or assets linked to a brand [Aaker, 1996; Keller, 1993]. It is one of core concepts in marketing since it is a key source of bearing the competitive advantage of a company over its competitors, boosting efficiency and effectiveness of marketing programs, and attaining the price premium due to increased customer satisfaction and loyalty, to name a few. A central component of brand equity is *brand associations*, which are the set of associations that consumers perceive with the brand [Keller, 1993]. For example, the brand associations of *Nike* may include *Tiger Woods*, *shoes*, and *basketball*. Its significance lies in that it is a *customer-driven* brand equity; that is, the brand associations are directly connected to customers' *top-of-mind* attitudes or feelings toward the brand, which provoke the reasons to preferentially purchase the products or services of the brand. For instance, if a customer strongly associates *Nike* with *golf shirts*, he may tend to first consider *Nike* products over other competitors' ones when he needs one.

Traditionally, measuring brand associations is a challenging task because it is required to be built from direct consumer responses to carefully designed questionnaires [Chen, 2001; Danes et al., 2010; Schnittka et al., 2012; Till et al., 2011; Keller, 1993]. Surveys over human subjects are usually time-consuming and prone to suffer from sampling bias and common methods bias. To circumvent these issues, with the recent emergence of online social media, it has become popular to indirectly leverage consumer-generated data on online communities such as Weblogs, boards, and Wiki. Beneficially, resources on such social media are obtainable inexpensively and almost instantaneously from a large crowd of potential customers. One typical example of such practice is the *Brand Association Map* developed by *Nielsen Online* [Akiva et al., 2008; Online, 2010], in which important concepts and themes correlated with a given brand name are automatically extracted from billions of online conversations.

In this chapter, for the study of brand associations, we propose to go beyond textual media, and take advantage of large-scale online *photo* collections, which have not been explored so far. Admittedly, pictures can be inferior to mine subjective sentiments than texts (*e.g.* *Nike is too expensive*). However, pictures can be a complementary information modality to show customers' experiences regarding brands within a natural context. With widespread availability of digital cameras and smartphones, people can freely take pictures on any memorable moments, which include experiencing or purchasing products they like. In addition, many online tools enable people to easily share, comment, or bookmark the images of products that they wish to buy.

Figure 10.1: Motivation for two visualization tasks toward brand association study from Web community photos with two competing brands of *Nike* and *Adidas*. (a) Task1: we perform exemplar detection and clustering to reconstruct brand association maps (BAM). A more strongly associated cluster with the brand appears closer to the center of the map. A higher correlated pair of clusters has a smaller angular distance. We show top 20 exemplars (*i.e.* cluster centers) in the map. On the right, for some selected exemplars, we show the average images of 40 nearest neighbors in their corresponding clusters. (b) Task2: we segment the most likely regions of brand in the images.

As an initial technical step toward the study of photo-based brand associations, we develop an approach to jointly achieving the following two levels of visualization tasks regarding brand associations. (See the examples in Fig.10.1).

(1) *Visualizing core pictorial concepts associated with brands*: It has been a key problem in brand association research to concisely visualize important concepts associated with brands in a form of networks or maps [Akiva et al., 2008; Danes et al., 2010; Schnittka et al., 2012; Till et al., 2011]. Therefore, our first task is, as shown in Fig.10.1.(a), to visualize core visual concepts of brands by summarizing online photos that are tagged and organized by general users. This goal involves three sub-problems: identifying a small number of image clusters and exemplars (*i.e.* cluster centers), discovering the similarity relations between clusters, and projecting them into a low-dimensional space.

(2) *Localizing the regions of brand in images*: Our second task is the *sub-image level* visualization of brand associations, while the first task addresses the *image-level* one. We aim to localize the regions that are most associated with the brand in each image in an unsupervised way (*i.e.* without any pre-defined models), as shown in Fig.10.1.(b). In our algorithm, we perform pixel-level image segmentation to delineate the regions of brand. Even though bounding boxes may be better as the final output to the general users, they can be trivially derived from segmentation results, by defining the minimum rectangle that encloses the segment while ignoring tiny unconnected dots.

We choose the above two tasks as the most fundamental building blocks for the study of photo-based brand associations for following reasons. The first task can provide a structural summary of large-scale and ever-growing online image data of brands, which otherwise are too overwhelming for human to grasp any underlying big picture. The second task can not only suppress background clutters, but also help reveal typical interactions between users and products in natural

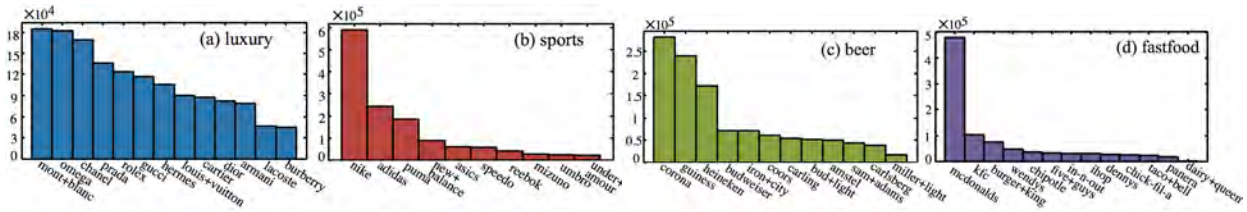


Figure 10.2: The dataset of 48 brands crawled from five photo sharing sites of Table 10.1. The brands are classified into four categories: (a) *luxury*, (b) *sports*, (c) *beer*, and (d) *fastfood*. The total number of images is 4,720,724.

social scenes, which can lead a wide variety of potential benefits, ranging from content-based image retrieval to online multimedia advertisement.

Besides the individual usefulness of these two visualization tasks, it is important to note that jointly solving these two tasks are *mutually rewarding*. The exemplar detection/clustering can group similar images, which can promote the brand localization since we can leverage the recurring foreground signals. In the reverse direction, localizing brand regions can enhance the similarity measurement between images, which subsequently contributes to better exemplar detection/clustering.

For evaluation, we collect about five millions of images of 48 brands of four categories (*i.e.* *sports*, *luxury*, *beer*, and *fastfood*) from five popular photo sharing sites, including FLICKR, PHOTOBUCKET, DEVIANTART, TWITPIC, and PINTEREST. In our experiments, we present the picture-driven brand association maps for some selected brands. We also demonstrate that our approach outperforms other candidate methods on both exemplar detection/clustering and brand localization tasks. Finally, we also compare between the results of our picture-based brand associations and actual sales data of the brands.

In almost all previous research for brand associations, the surveys on customers are the main approach to collect source data. Among many ways to conduct the survey, the *free association* procedure has been one of the simplest but often most powerful ways to profile brand associations [Chen, 2001; Danes et al., 2010; Till et al., 2011]. In this technique, subjects are asked to freely answer their feelings and thoughts about a given brand name without any editing or censoring [Nelson et al., 2004]. (*e.g.* What comes to mind when you think of *Nike*?) Our research is also based on this *free association* idea, because we view the Web photos tagged with a brand name by anonymous users as their candid pictorial impressions to the brand. Therefore, from a viewpoint of brand association research, the contribution of our work is to introduce a novel source of data for the analysis. In this line of research, the brand association map of Nielsen Online [Online, 2010; Akiva et al., 2008] is closely related to our work because both approaches explore online data of general users. However, the uniqueness of our research lies in exploring online image data, which convey complementary views on the associations that can be missed by textual data. In addition, we localize the most brand-related regions in all images, which is another important novel feature of our work.

Web sites	Characteristics
FLICKR/ PHOTOBUCKET	Two largest and most popular photo sharing sites in terms of volumes of photos.
PINTEREST	Image collections bookmarked by users
DEVIANTART	Various forms of artwork created by users.
TWITPIC	Photos shared via Twitter.

Table 10.1: Five Web sites for crawling photos.

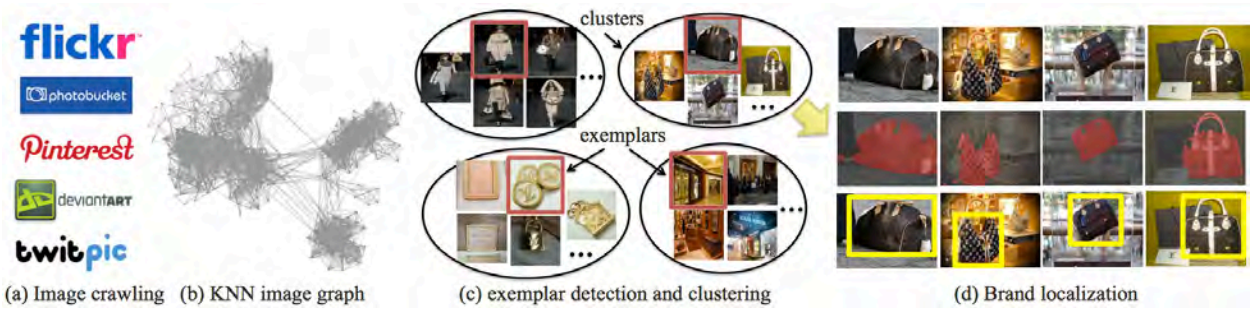


Figure 10.3: The overview of the proposed approach with an example of the *Louis+Vuitton*. (a) As an input, we crawl the photos of the brand from the five photo sharing sites. (b) Next, we build a K-nearest neighbor (KNN) similarity graph between images. (c) We perform the graph-based exemplar detection/clustering. (d) Finally, we cosegment the images in the same cluster in order to discover the regions of a brand in each image. As a closed-loop solution, we can return to the KNN graph construction with the new segmentation-based image similarity metric.

10.2 Problem Formulation

10.2.1 Image Data Crawling

Since we are interested in consumer-driven views on the brands, we use the online photos that are contributed and organized by general Web users. As source data, we crawl images from the five popular photo sharing sites in Table 10.1. The characteristics of the pictures on the five sites are different from one another as shown in Table 10.1. We exclude the GOOGLE IMAGE SEARCH because much of the pictures are originated from online shopping malls or news agencies.

We query the brand names via the built-in search engines of the above sites to search for the pictures tagged with brand names. We download all retrieved images without any filtering. We also crawl meta-data of the pictures (*e.g.* timestamps, titles, user names, texts), if available.

Fig.10.2 summarizes our dataset of 4,783,345 images for 48 brands, which can be classified into four categories: *luxury*, *sports*, *beer*, and *fastfood*. The number of images per brand varies much according to the popularity of the brand.

10.2.2 Overview of Algorithm

Fig.10.3 presents an overview of our approach. The input of our algorithm is a set of photos for a brand of interest. Let $\mathcal{I} = \{I_1, \cdots, I_N\}$ be the set of input images, where N is the number

of images. As shown in Fig.10.3.(b), our first step is to build a K-nearest neighbor (KNN) graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ in which each image I is connected with its K most similar images in $\mathcal{I}$. We will present our image descriptors in section 10.3.1, similarity measures in section 10.3.2, and KNN graph construction in section 10.3.3.

The next step is to perform exemplar detection and clustering on the KNN graph $\mathcal{G}$, which will be discussed in section 10.3.4. Its goal is to discover a small set of representative images called exemplars $\mathcal{A}(\subset \mathcal{I})$, and to partition $\mathcal{I}$ so that each image is associated with its closest exemplar, as shown in Fig.10.3.(c). Therefore, the clusters are the groups of contextually and visually similar images, and the exemplars are the most prototypical images of the clusters.

The clustering helps discover the coherent groups of images from extremely diverse Web images, which is subsequently beneficial to detect the regions of a brand in the images (see examples in Fig.10.3.(d)). In our setting, the brand localization is formulated as the problem of *cosegmentation* [Batra et al., 2011; Rother et al., 2006; Kim and Xing, 2012; Kim et al., 2011], which has been actively studied in image segmentation research. Its goal is to simultaneously segment out recurring objects or foregrounds across the multiple images. Obviously, the images in the same cluster are likely to share the same themes of the brand (*e.g.* bags in Fig.10.3.(d)), which can be discovered by the cosegmentation approach. We summarize the procedure of cosegmentation in section 10.3.5.

In our closed-loop approach, the segmentation can enhance the exemplar detection/clustering by promoting a more accurate image similarity measure, which will be justified in section 10.3.2 with an intuitive example. Hence, after finishing the cosegmentation step, we can return to the KNN graph construction and repeat the whole algorithm again with the new segmentation-based image similarity metric.

The brand association map like Fig.10.1 can be constructed from the exemplar detection/clustering output. The algorithm will be presented in section 10.4.

10.3 Exemplar Detection/Clustering and Brand Localization

10.3.1 Image Description

For image description, we use one of common practices in recent computer vision research: the dense feature extraction with vector quantization. We densely extract two most popular features from each image: HSV color SIFT and histogram of oriented edge (HOG) feature on a regular grid at steps of 4 and 8 pixels, respectively. Then, we form 300 visual words for each feature type by applying K-means to randomly selected features. Finally, the nearest word is assigned to every node of the grid. We use publicly available codes[1] for the whole process of feature extraction.

10.3.2 Image Similarity Measure

One prerequisite to accurate clustering is an appropriate similarity measure between images, denoted by $\sigma : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}$. We assert that even imperfect segmentation helps enhance the measure-

[1] The SIFT and HOG feature extraction codes are available at http://www.vlfeat.org, and at http://www.cs.brown.edu/∼pff/latent, respectively.

Figure 10.4: The benefit of segmentation for image similarity measurement. (a) For an unsegmented image pair, the spatial pyramid histograms are constructed on the whole images, which may not correctly reflect the location and scale variations. (b) After segmentation, the image similarity is computed as the mean of the best assigned segment similarities.

ment of image similarity, which can justify our closed-loop approach. Fig.10.4 shows a typical example, in which the two images are similar in that both include persons with glasses of *Guinness* beer. For an unsegmented image pair, the image similarity is calculated from two-level spatial pyramid histograms on the whole images [Lazebnik et al., 2006], which are not robust against location, scale, and pose variation as shown in Fig.10.4.(a). On the other hand, as shown in Fig.10.4.(b), this issue can be largely alleviated even with an imperfect segmentation. Given the two sets of segments of the images, we find the best matches between them by solving the linear assignment problem. Then, we compute the mean of similarities between corresponding segments, which is used as the image similarity metric. For the segment similarity, we use the histogram intersection kernel on the spatial pyramids of the segments.

10.3.3 Constructing K-Nearest Neighbor Graphs

Given the image descriptors and similarity measures, the construction of a KNN graph is straightforward. However, if we naively compare all pairwise similarity by brute-force, it takes $\mathcal{O}(N^2)$, which can be prohibitively slow for a large $\mathcal{I}$. Fortunately, a large number of algorithms have been proposed to construct exact or approximate KNN graphs without suffering from the quadratic complexity (*e.g.* [Dong et al., 2011; Wang et al., 2012b]). In this work, we exploit the idea of multiple random divide-and-conquer [Wang et al., 2012b], which allows to create an approximate KNN graph of high accuracy in $\mathcal{O}(N \log N)$ time. The method is simple: the dataset is randomly and recursively partitioned into subsets, and build an exact neighborhood graph over each subset. This random divide-and-conquer process repeats for several times, and then the aggregation of all neighborhood graphs of subsets can create a more accurate approximate KNN graph with a high probability. The details of procedures, theoretic analyses, and several heuristics to further enhance accuracy can be found in [Wang et al., 2012b]. In our application, meta-data of images are also exploited for recursive random division. We repeat partitioning the image set into subsets according to each type of meta-data (*e.g.* image sources, owners, titles, or taken times, if available). For example, in one partition, the subsets includes the images that are taken at similar time; in another

Algorithm 14: Exemplar detection and clustering.

Input: (1) Image graph $\mathbf{G}$. (2) Number of exemplars L.
Output: (1) Exemplar set $\mathcal{A}$ and cluster set $\mathcal{C}$.

1: Append a constant vector $\mathbf{z} \in \mathbb{R}^{(N+1)\times 1}$ to the end column of $\mathbf{G}$ and $\mathbf{z}^T$ to the end row of $\mathbf{G}$. ($N = |\mathbf{G}|$).
2: $\mathcal{A} = \texttt{SubmDiv}(\mathbf{G}, M)$.
3: $\{\mathcal{C}_l\}_{l=1}^{L} = \texttt{ClustSrc}(\mathbf{G}, \mathcal{A})$.

/* Select M number of central and diverse exemplars $\mathcal{A}$.
Function $[\mathcal{A}] = \texttt{SubmDiv}(\mathbf{G}, M)$
  1: $\mathcal{A} \leftarrow \emptyset$. $\mathbf{u} = \mathbf{0} \in \mathbb{R}^{N\times 1}$.
  while $|\mathcal{A}| \leq L$ **do**
    2: **for** $i = 1 : N$ **do** $\mathbf{u}(i) = \texttt{TempSrc}(\mathbf{G}, \{\mathcal{A} \cup i\})$.
    3: $\mathcal{A} \leftarrow \mathcal{A} \cup \text{argmax}_i\, \mathbf{u}$. Set $\mathbf{u} = \mathbf{0}$.

/* Get marginal gain u from the $\mathbf{G}$ and the node set $\mathcal{P}$.
Function $[u] = \texttt{TempSrc}(\mathbf{G}, \mathcal{P})$
  1: Solve $\mathbf{u} = \mathbf{L}\mathbf{u}$ where $\mathbf{L}$ is the Laplacian of $\mathbf{G}$ under constraints of $\mathbf{u}(\mathcal{P}) = 1$ and $\mathbf{u}(N+1) = 0$.
  2: Compute the marginal gain $u = |\mathbf{u}|_1$.

/* Get cluster set $\mathcal{C}$ from the graph $\mathbf{G}$ and exemplars $\mathcal{A}$.
Function $\mathcal{C} = \texttt{ClustSrc}(\mathbf{G}, \mathcal{A})$
  1: Let $L = |\mathcal{A}|$ and $L = |\mathbf{G}|$. $\mathcal{V}$ is vertext set of $\mathbf{G}$.
  2: Compute the matrix $\mathbf{X} \in \mathbb{R}^{(L-L)\times L}$ by solving $\mathbf{L}_u\mathbf{X} = -\mathbf{B}^T\mathbf{I}_s$ where if we let $\mathcal{X} = \mathcal{V}\backslash\mathcal{A}$, $\mathbf{L}_u = \mathbf{L}(\mathcal{X}, \mathcal{X})$, $\mathbf{B} = \mathbf{L}(\mathcal{A}, \mathcal{X})$, and $\mathbf{I}_s$ is an $L \times L$ identity matrix.
  3: Each vertex $v \in \mathcal{V}$ is clustered $c_v = \text{argmax}_k\, \mathbf{X}(j, k)$.

partition, the subset comprises the images that are owned by the same user, and so on. The basic assumption is that if images are taken at similar time or by the same user, they are likely to share similar visual contents. In our experiments, this meta-data based heuristics is efficient and effective for the KNN graph construction.

10.3.4 Exemplar detection and clustering

Given a KNN graph $\mathcal{G}$, our next step is to perform exemplar detection. As a base algorithm, we use the diversity ranking algorithm of chapter 6 [Kim et al., 2011], which can choose L number of exemplars that are not only most central but also distinctive one another, by solving submodular optimization on the similarity graph $\mathcal{G}$. Since the L exemplars are discovered in a decreasing order of ranking scores, one can set L to an arbitrary large number. We here do not discuss the details of the algorithm, which can be found in chapter 6. Instead, we denote the exemplar detection procedure by $\mathcal{A} = \texttt{SubmDiv}(\mathbf{G}, L)$ where $\mathcal{A}$ is the set of exemplars and $\mathbf{G} \in \mathbb{R}^{N\times N}$ is the adjacency matrix of graph $\mathcal{G}$. The pseudocode is summarized in the step **1**–**2** of Algorithm 14.

Next, the clustering is performed using the random walk model [Grady, 2006]; each image i is associated with the exemplar that a random walker starting at i is most likely to reach first.

Algorithm 15: Brand localization via cosegmentation.

Input: (1) Cluster set $\mathcal{C} = \{\mathcal{C}_l\}_{l=1}^{L}$. (2) Image graph $\mathbf{G}$.
Output: (1) Set of segmented images $\mathcal{F}$ for each $i \in \mathcal{I}$.

foreach $\mathcal{C}_l \in \mathcal{C}$ **do**
- **1**: Find central image $c = \texttt{SubmDiv}(\mathbf{G}_l, 1)$ where $\mathbf{G}_l = \mathbf{G}(\mathcal{C}_l)$ is the subgraph of $\mathcal{C}_l$.
- **2**: Apply the unsupervised MFC algorithm in chapter 7 to $\{c \cup \mathcal{N}_c\}$ where $\mathcal{N}_c$ is the neighbor of c in the graph $\mathbf{G}_l$. As a result, we obtain segmented images $\mathcal{F}_{c \cup \mathcal{N}_c}$.
- **3**: Let $\mathcal{U}_l \leftarrow \mathcal{C}_l \backslash \{c \cup \mathcal{N}_c\}$. $\mathcal{F} \leftarrow \mathcal{F}_{c \cup \mathcal{N}_c}$.
- **while** $\mathcal{U}_l \neq \emptyset$ **do**
  - **4**: Sample an image i from $\{\mathcal{U}_l \cap \mathcal{N}_{\mathcal{F}}\}$.
  - **5**: Get foreground model $\{v_i\} = \texttt{FM}\,(\{\mathcal{N}_i \cap \mathcal{F}\})$.
  - **6**: Segment the image $\mathcal{F}_i = \texttt{RA}\,(i, \{v_i\})$.
  - **7**: $\mathcal{U}_l \leftarrow \mathcal{U}_l \backslash i$. $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_i$.

/* $\{v_i\} = \texttt{FM}\,(\mathcal{F}_i)$ is the function to learn foreground model $\{v_i\}$ of MFC from the segmented images $\mathcal{F}_i$. /* $\mathcal{F}_i = \texttt{RA}\,(i, \{v_i\})$ is the function to run region assign- ment of MFC on image i using $\{v_i\}$.

Then, we cluster the images that share the same exemplar as the most probable destination. This procedure is implemented as a function `ClustSrc` of Algorithm 14.

10.3.5 Brand Localization via Cosegmentation

As the output of clustering, we obtain the groups of coherent images out of extremely diverse Web photos. The brand localization is achieved by applying the cosegmentation algorithm to each of $\mathcal{C} = \{\mathcal{C}_l\}_{l=1}^{L}$ separately. Such separate cosegmentation scheme is more beneficial not only for parallel computation but also for performance. Especially, for performance, it prevents cosegmenting the images of no commonality, which contradicts the basic assumption of cosegmentation algorithms. For examples, given the *Prada* brand, jointly segmenting bag and fashion model images could be worsen than individually segmenting each image.

The goal of cosegmentation is to partition each image into foreground (*i.e.* the regions recurring across the images like *bags* in Fig.10.3.(d)) and background (*i.e.* the other regions). We select the MFC method in chapter 7 [Kim and Xing, 2012] as our base cosegmentation algorithm, since it is scalable and has been successfully tested with Flickr user images. The MFC algorithm consists of two procedures, which are *foreground modeling* and *region assignment*. The foreground modeling step learns the appearance models for foreground and background, which are accomplished by using any region classifiers or their combinations. We use the Gaussian mixture model (GMM) on the RGB color space. The foreground models can compute the values of any given regions with respect to the foregrounds and background, based on which the region assignment allocates the regions of an image via a combinatorial-auction style optimization to maximize the overall allocation values. More details of the algorithm can be referred to chapter 7.

For each cluster $\mathcal{C}_l$, we perform the cosegmentation by iteratively applying the foreground modeling and region assignment steps under the guidance of the subgraph $\mathcal{G}(\mathcal{C}_l)$ whose vertex set is $\mathcal{C}_l$. Its basic idea is that the neighboring images in $\mathcal{G}(\mathcal{C}_l)$ are visually similar, and thus they are

likely to share enough commonality to be segmented together. Therefore, we iteratively segment each image i by using the learned foreground models from its neighbors in the graph. Then, the segmented image i is subsequently used to learn the foreground models for its neighbors' segmentation. That is, we iteratively run foreground modeling and region assignment by following the edges of $\mathcal{G}(\mathcal{C}_l)$. The overall algorithm is summarized in Algorithm 15. For initialization, as shown in step **1–2** of Algorithm 15, we run the unsupervised version of the MFC algorithm to the exemplar of C_l and its neighbors, from which the above iterative cosegmentation starts.

10.4 Embedding Brand Association Maps

We visualize the clusters (or exemplars) in a circular layout in order to concisely represent both short-range and long-range interactions between them. We place the visual clusters by using two different metrics, the *radial distance* and *angular distance*, inspired by the Nielsen's method [Akiva et al., 2008]:

1. The *radial distance* of a cluster reflects how strongly it associates with the brand. A larger cluster appears closer to the center of the map.
2. The *angular distance* between a cluster pair shows their closeness. The smaller the angular distance between the two is, the higher the correlation is.

Since Nielsen's mapping algorithm is unknown and no photo-based brand association mapping has been developed yet, we design a new embedding algorithm that satisfies the above requirements. Our objective is to calculate $(\mathbf{r}, \boldsymbol{\theta}) \in \mathbb{R}^{L\times 2}$, which are the polar coordinates of all clusters of $\mathcal{C}$. Algorithm 16 summarizes the whole mapping procedure.

Radial distances of clusters: According to the requirement 1, a larger cluster has a smaller radial distance (*i.e.* closer to the center). In order to estimate the cluster sizes, we first compute the stationary distribution $\boldsymbol{\pi} \in \mathbb{R}^{N\times 1}$ of the graph $\mathcal{G}$, where $\boldsymbol{\pi}(i)$ indicates a random walker's visiting probability of node i. We assume that the size of cluster $\mathcal{C}_a$ is proportional to the sum of stationary distribution of the nodes in $\mathcal{C}_a$, which means the portion of time that a random walker traversing the graph stays in the cluster $\mathcal{C}_a$. That is, in a larger cluster, a random walker stays longer.

Given the transition matrix $\mathbf{P}$ obtained by normalizing the rows of $\mathbf{G}$, the stationary probability vector $\boldsymbol{\pi}$ can be computed by solving $\boldsymbol{\pi} = \mathbf{P}^T\boldsymbol{\pi}$ with $\|\boldsymbol{\pi}\|_1 = 1$. However, it is well known from the success of PageRank that a regularized stationary distribution is more robust and can incorporate a prior knowledge; it can be obtained by solving

$$\boldsymbol{\pi} = \widetilde{\mathbf{P}}^T\boldsymbol{\pi} \quad \text{where } \widetilde{\mathbf{P}} = \lambda\mathbf{P} + (1-\lambda)\mathbf{1}\mathbf{v}^T \tag{10.1}$$

where $\mathbf{v} \in \mathbb{R}^{N\times 1}$ is the teleporting probability such that $v_i \geq 0$, $\|\mathbf{v}\|_1 = 1$. It can supply a prior ranking to each node; without it, one can let $\mathbf{v} = [1/N, \cdots, 1/N]^T$ be uniform. $\mathbf{1}$ is an all-one vector, and λ is a regularization parameter to weight the random walker's behavior between edge following and random transporting. We set $\lambda = 0.9$ in all experiments.

Once we have $\boldsymbol{\pi}$, then we compute the stationary probability π_a of each cluster $\mathcal{C}_a$ by summing over the values of vertices in the cluster: $\pi_a = \sum_{i\in\mathcal{C}_a} \boldsymbol{\pi}(i)$. Let r_{max} and r_{min} be max and min

Algorithm 16: Computing polar coordinates of clusters.

Input: (a) Cluster set $\mathcal{C} = \{\mathcal{C}_l\}_{l=1}^{L}$. (b) Image graph $\mathbf{G}$. (c) Image sizes to be drawn $\mathbf{t} \in \mathbb{R}^{L\times 1}$.
Output: Polar coordinates $(\mathbf{r}, \boldsymbol{\theta}) \in \mathbb{R}^{L\times 2}$ of $\mathcal{C}$.

/ Radial coordinates. */*
1: Compute transition matrix $\mathbf{P}$ by row-normalizing $\mathbf{G}$.
2: Solve Eq.(10.1) to get stationary distribution $\boldsymbol{\pi} \in \mathbb{R}^{N\times 1}$.
3: **foreach** $\mathcal{C}_a \in \mathcal{C}$ **do** compute $\pi_a = \sum_{i\in\mathcal{C}_a} \boldsymbol{\pi}(i)$.
4: Let $\pi_{min} = \min_{a\in\mathcal{C}} \boldsymbol{\pi}_a$ and $\pi_{max} = \max_{a\in\mathcal{C}} \boldsymbol{\pi}_a$.
5: **foreach** $\mathcal{C}_a \in \mathcal{C}$ **do** obtain $\mathbf{r}(a)$ by solving Eq.(10.2).
/ Angular coordinates. */*
6: Obtain the cluster similarity $\mathbf{S} \in \mathbb{R}^{L\times L}$ from Eq.(10.4).
7: Initialize $\boldsymbol{\theta}$ by polar dendrogram of hierarchical clustering on $\mathbf{S}$, $J = 0$, $J_{old} =$ a large number.
while $|J - J_{old}| > \epsilon$ **do**
  8: Calculate $\frac{\partial}{\partial \boldsymbol{\theta}} J \in \mathbb{R}^{L\times 1}$. For each $a \in \mathcal{C}$, $\frac{\partial}{\partial \boldsymbol{\theta}_a} J = \sum_{b\in\mathcal{C}} \big(\mathbf{S}(a,b) - \gamma|\theta_a - \theta_b|^{\gamma-1}\big)G$ where $G = -2(1-\cos(\theta_a - \theta_b))^{-1/2}(-\sin\theta_a\cos\theta_b + \cos\theta_a \sin\theta_b)$.
  9: $\boldsymbol{\theta}_{new} = \boldsymbol{\theta} + \mu \frac{\partial}{\partial \boldsymbol{\theta}} J$.
  10: $J_{new} = \sum_a \sum_b \mathbf{S}(a,b)|\theta_a - \theta_b| - \sum\sum |\theta_a - \theta_b|^{\gamma}$.
  11: Update $J_{old} = J, J = J_{new}, \boldsymbol{\theta} = \boldsymbol{\theta}_{new}$.
/ Force-directed refinement. */*
12: Obtain Cartesian coordinates $\mathbf{x} \in \mathbb{R}^{L\times 2}$ from $(\mathbf{r}, \boldsymbol{\theta})$ and a pariwse distance matrix $\mathbf{D}$. Store the original $\mathbf{x}_0$.
while $\mathbf{x}$ *is updated* **do**
  13: Set the displacement vector $\mathbf{d} = \mathbf{0}$. Set attractive and repulsive forces: $f_a(x) = x^2/k$ and $f_r(x) = k^2/x$.
  foreach *pair* (a,b) *if* $\mathbf{D}(a,b) < \gamma(\mathbf{t}(a) + \mathbf{t}(b))$ **do**
    14: $\mathbf{d}(b) += f_r(|\mathbf{x}(b) - \mathbf{x}(a)|)$.
  15: **foreach** $a \in \mathcal{C}$ **do** $\mathbf{d}(a) -= f_a(|\mathbf{x}(a) - \mathbf{x}_0(a)|)$.
  16: **foreach** $a \in \mathcal{C}$ **do** $\mathbf{x}(a) += \mathbf{d}(a)$.
17: Obtain the final $(\mathbf{r}, \boldsymbol{\theta})$ from $\mathbf{x}$.

radius of the circular layout, and π_{max} and π_{min} be max and min cluster stationary probability, respectively. Finally, the radial coordinate $\mathbf{r}(c)$ of cluster $\mathcal{C}_a$ is

$$\mathbf{r}(a) = \frac{r_{max} - r_{min}}{\pi_{max} - \pi_{min}}(\pi_{max} - \pi_a) + r_{min}. \tag{10.2}$$

Angular coordinates of clusters: In order to obtain the angular coordinates $\boldsymbol{\theta}$ of clusters $\mathcal{C}$, we first compute all pairwise similarities $\mathbf{S} \in \mathbb{R}^{L\times L}$ between the clusters, and then apply the modified spherical *Laplacian Eigenmap* technique [Belkin and Niyogi, 2003; Carter et al., 2009] to project the clusters on a circular manifold.

We use the *random walk with restart* (RWR) algorithm [Sun et al., 2005] to define the cluster similarity on a graph. The similarity values of all nodes $\mathbf{s}_a$ with respect to cluster $\mathcal{C}_a$ is defined as

$$\mathbf{s}_a = \lambda \mathbf{P}\mathbf{s}_a + (1-\lambda)\mathbf{v}_a^T \text{with } \mathbf{v}_a(i) = \begin{cases} 1/|\mathcal{C}_a| & \text{if } i \in \mathcal{C}_a \\ 0 & \text{otherwise} \end{cases} \tag{10.3}$$

The score $\mathbf{s}_a(i)$ means the probability that a random walker stays at node i when the walker follows the edge of graph with probability λ and return to uniformly random nodes of cluster $\mathcal{C}_a$ with $1-\lambda$. It is straightforward to compute the similarity score from $\mathcal{C}_a$ to $\mathcal{C}_b$, denoted by $\mathbf{S}(a,b)$, as follows:

$$\mathbf{S}(a,b) = \sum_{i \in \mathcal{C}_b} \mathbf{s}_a(i)/S_a \quad \text{where } S_a = 1 - \sum_{i \in \mathcal{C}_a} \mathbf{s}_a(i). \tag{10.4}$$

The next step is to project the clusters on a unit circle from the pairwise cluster similarity matrix $\mathbf{S}$. Our circular embedding is based on the *Spherical Laplacian Information Maps* (SLIM) [Carter et al., 2009], which extends the Laplacian eigenmap (LEM) optimization [Belkin and Niyogi, 2003] with an additional constraint of embedding data on the surface of a sphere.

Conceptually, if a pair of clusters is similar to each other, then their angular difference in embedding should be small. Hence, the LEM is formulated as finding $\boldsymbol{\theta}$ to minimize

$$\boldsymbol{\theta} = \operatorname{argmin} \sum_a \sum_b \mathbf{S}(a,b)|\theta_a - \theta_b| - \Omega(\boldsymbol{\theta}). \tag{10.5}$$

As a consequence of the LEM objective (*i.e.* the first term of Eq.(10.5)), nearby points in the graph are as close together as possible in the angular representation. However, the optimization of the LEM objective attains a trivial solution to collapse all data to the same point. Therefore, the regularization term $\Omega(\boldsymbol{\theta})$ is included in order to spread the embedded clusters on a circle:

$$\Omega(\boldsymbol{\theta}) = \sum_a \sum_b |\theta_a - \theta_b|^\gamma \tag{10.6}$$

where γ is a power-weighting constant (*e.g.* $\gamma = 0.5$ in our experiments). $\Omega(\boldsymbol{\theta})$ leads the optimization to prefer large angular distances between all pairs of clusters on a circle.

Since the optimization problem in Eq.(10.5) has no closed-form solution, we employ a gradient descent procedure, as summarized in step **7–11** of Algorithm 16. By nature, the final embedding highly depends on the initialization, for which we first perform hierarchical clustering on the $\mathbf{S}$, and then use its polar dendrogram. This initialization enables similar nodes to have small geodesic distances.

Layout refinement: Once we obtain the coordinates of clusters $(\mathbf{r}, \boldsymbol{\theta})$, we slightly change them so that the final visualization is more aesthetic. One modification is to separate any pair of exemplars that are too much overlapped. To this end, we use Fruchterman and Reingold's method, one of popular force-directed drawing algorithms. The positions of exemplars are updated to reach equilibrium states by the attractive and repulsive forces. The attractive forces encourage the updated positions to be as similar to the original $(\mathbf{r}, \boldsymbol{\theta})$ as possible, while the repulsive forces take part severely overlapped exemplars. This refinement step is summarized in step **12–17** of Algorithm 16.

Figure 10.5: Examples of brand association maps for six brands of the *luxury* category.

10.5 Experiments

In our experiments, we first present the brand association maps for several competing brands in section 10.5.1. Then, we quantitatively evaluate the proposed approach from two technical perspectives: exemplar detection/clustering in section 10.5.2, and brand localization via image cosegmentation in section 10.5.3. Since the main goal here is to achieve the two technical visualization tasks for brand associations, we focus on the validating the algorithmic performance over other candidate methods instead of user study. Finally, we examine the correlation between our findings from community photos and the sales data of brands in section 10.5.4.

10.5.1 Results on Brand Association Maps

We present six competing brands of the *luxury* and *sports* categories in Fig.10.5 and Fig.10.6, respectively. For each brand, we first find the 25 largest clusters, from which we manually select 20 ones. Such manual selection is due to remove some noisy or highly redundant ones. We have made several interesting observations as follows. First of all, our algorithm successfully discover brands' characteristic visual themes that are distinctive one another. For example, we can see several watch clusters in the *Rolex*, and the iconic check patterns of the *Burberry*. Second, much of highly ranked clusters attribute to some specific scenes or circumstances where photo-taking is much more preferable. For example, we detect a lot of *fashion show* clusters in almost all brands.

Figure 10.6: Examples of brand association maps for six brands of the *sports* category.

In the *Rolex*, the clusters of some events that are sponsored by the *Rolex* (*e.g.* horse-riding and auto-racing) are as dominant as those of its products (*e.g.* watches). Such topics are more favorable to be recorded as pictures rather than texts. In the *Louis+Vuitton*, there are a lot of *wedding* related clusters, which makes sense because the wedding is not only an event where the products of luxury brands are purchased the most, but also a memorable moment where the photos are taken much.

Although our photo-based brand association map is novel and promising, there are several issues to be explored further. First, we may need to correctly handle highly redundant or noisy clusters, which are mainly caused by the imperfection of image processing and clustering. Second, we also need to deal with polysemous brand names; for example, the *Mont+Blanc* is also the name of the mountain, and the *Corona* indicates the astronomical phenomenon as well. This confusion may hinder the correct brand analysis. If we supplement additional keywords during image crawling to filter them out, the retrieved images can decrease severely.

10.5.2 Results on Clustering

Task: We evaluate the performance of our algorithm for the exemplar detection/clustering task, by comparing with several candidate methods. For quantitative evaluation, we first choose 20 brands (*i.e.* five brands per category), and generate 100 sets of groundtruth per brand as follows. We randomly sample three images (i, j, k) from the image set of a brand, and manually label which of j and k is more similar to image i. We denote $j \succ k|i$ if j is more similar to i than k. Although the

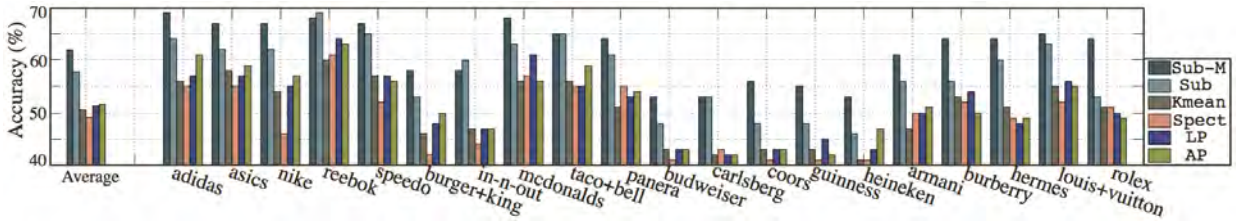


Figure 10.7: Clustering accuracies of two variants of our approach (`Sub-*`) and four baselines for the 20 selected brands. The average accuracies over the 20 brands, shown in the leftmost bar set, are (`Sub-M`): 62.0%, (`Sub`): 57.8%, (`Kmean`): 50.5%, (`Spect`): 49.2%, (`LP`): 51.4%, and (`AP`): 51.7%.

labeled sets are relatively few compared to the dataset size, in practice this sampling-based annotation is commonly adopted in standard large-scale benchmark datasets such as ImageNet [Deng et al., 2009] and LabelMe [Russell et al., 2008].

After applying each algorithm, suppose that C_i, C_j, and C_k denote the clusters that include image i, j, and k, respectively. Then, we compute the similarity between clusters $\sigma(C_j, C_i)$ and $\sigma(C_k, C_i)$ by using the RWR algorithm in section 10.4. Finally, we compute the accuracy of the algorithm using the Wilcoxon–Mann–Whitney statistics:

$$ACC := \frac{\sum_{(i,j,k)} \mathbb{I}(j \succ k | i \ \wedge \ \sigma(C_j, C_i) > \sigma(C_k, C_i))}{\sum_{(i,j,k)} \mathbb{I}(j \succ k | i)} \tag{10.7}$$

where $\mathbb{I}$ is an indicator function. The accuracy increases only if the algorithm can partition the image set into coherent clusters, and the similarities between clusters coincide well with human's judgment on the image similarity.

Baselines: We compare our algorithm with four baselines. The (`KMean`) and the (`Spect`) are the two popular clustering methods, K-means and spectral clustering, respectively. The (`LP`) is a label propagation algorithm for community detection [Raghavan et al., 2007], and the (`AP`) is the *affinity propagation* [Frey and Dueck, 2007], which is a message-passing based clustering algorithm. Our algorithm is tested in two different ways, according to whether image segmentation is in a loop or not. The (`Sub`) does not exploit the image cosegmentation output, whereas the (`Sub-M`) is our fully geared approach. That is, this comparison can justify the usefulness of our alternating approach between clustering and cosegmentation. We set $L = 300$, and use the same image features in section 10.3.1 for all the algorithms.

Quantitative results: Fig.10.7 reports the results of our algorithm and four baselines across 20 brand classes. The leftmost bar set is the average accuracies of 20 classes. In most brand classes, the accuracies of our method (`Sub-M`) are better than those of all the baselines. The average accuracy of our (`Sub-M`) is 62.0%, which is much higher than 51.7% of the best baseline (`AP`). In addition, the average accuracies of the (`Sub-M`) are notably better than (`Sub`), which implicates the segmentation for brand localization can improve the clustering performance as expected.

10.5.3 Results on Brand Localization

Task: The brand localization task is evaluated as follows. As groundtruths, we perform pixel-wise manual annotation for 50 randomly sampled images per brand, for the same 20 brands in the

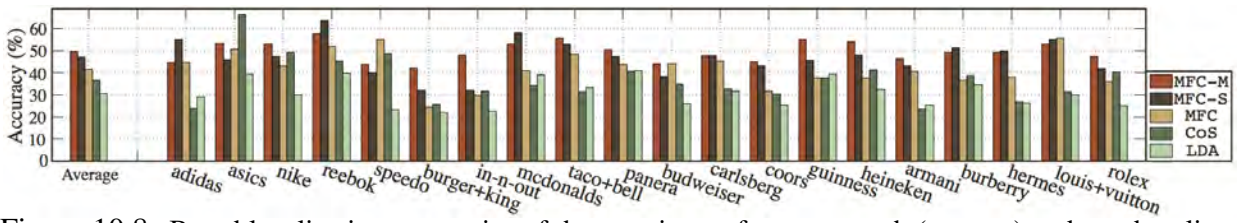


Figure 10.8: Brand localization accuracies of three variants of our approach (MFC−⋆) and two baselines. The average accuracies of the leftmost bar set are (MFC−M): 49.5%, (MFC−S): 46.8%, (MFC): 41.7%, (COS): 36.7%, and (LDA): 30.6%.

previous section. We do not label too obvious images depicting products on white background, since they cannot correctly measure the performances of algorithms. The accuracy is measured by the intersection-over-union metric $(GT_i \cap R_i)/(GT_i \cup R_i)$, where GT_i is the groundtruth of image i and R_i is the regions detected by the algorithm. It is a standard metric in object localization and segmentation literature. We compute the average of the metric from all annotated images for each brand.

Baselines: We select three baselines that can discover the regions of multiple objects from a large-scale image set in an unsupervised manner (*i.e.* without any labeled seed images). The (LDA) [Russell et al., 2006] is an LDA-based unsupervised localization method, and the (COS) is a state-of-art submodular optimization based cosegmentation algorithm in chapter 6. For (COS), we first partition the images into multiple groups, and separately apply the cosegmentation algorithm to each group, as proposed in chapter 6. Our algorithm is tested with three different versions, according to whether exemplar detection/clustering is in a loop or not. The (MFC) runs our cosegmentation without involving our clustering output (but using a random partitioning instead), in order to show the importance of the clustering step when segmenting highly diverse Web images. The (MFC−S) is a single run of our proposed exemplar detection/clustering and cosegmentation, and (MFC−M) iterates this process more than twice. In almost all cases, it converges in two iterations. Hence, this comparison can quantify the accuracy improvement by the iterative algorithm. We run all baselines and our methods in an unsupervised way for a fair comparison. Since it is hard to know the best K beforehand (*e.g.* multiple foregrounds may exist in an image), we repeat each method by changing K from one to five, and report the best results.

Quantitative results: Fig.10.8 shows that our method outperforms other candidate localization methods in almost all classes. Especially, our average accuracy is 49.5%, which is notably higher than 36.7% of the best baseline (COS), In addition, the average accuracies of the (MFC−M) are higher than those of (MFC−S) and (MFC), which demonstrates that the clustering and cosegmentation are mutually-rewarding.

Qualitative analysis: Fig.10.9 shows 12 sets of brand localization examples. The images of each set belong to the same cluster, and thus are cosegmented. We show our pixel-level segmentation output, from which bounding boxed regions can be trivially obtained as well.

We observe that the subjects of pictures and their appearances severely vary even though they are associated with the same brands. However, if we have sufficiently large photo sets, we can leverage the overlapping contents across the datasets. Consequently, our approach is able to quickly cluster a large-scale image set and segment common regions in an unsupervised and

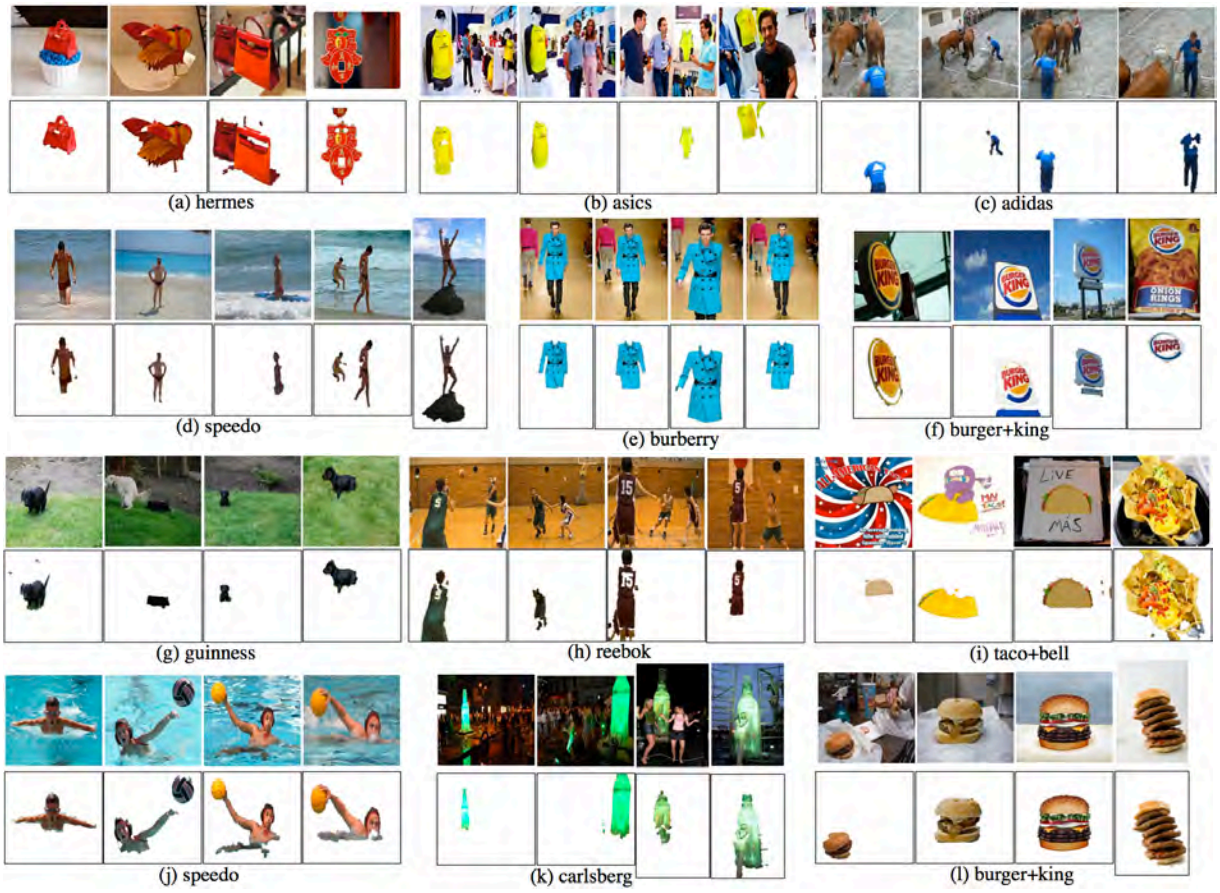


Figure 10.9: 12 groups of brand localization examples. We sample four or five images per group that belong to the same cluster, and thus are jointly segmented. We show input images (top) and their segmentation output (bottom).

bottom-up way, which can be an useful function for various Web applications, including detecting regions of brand for online multimedia advertisement.

Fig.10.10 illustrates three groups of typical failure cases. As we already discussed in chapter 8, our approach has room for improvement by integrating with the learned region classifiers as foreground models, which can provide high-level knowledge about the objects of interest. For example, we can alleviate the issue that a foreground of several distinctive regions is split into multiple parts (*e.g.* McDonalds' mascot in Fig.10.10.(b), and person in Fig.10.10.(c)). In Fig.10.10.(a), some visually-similar background regions are merged with the foreground (*i.e.* cars), which might be relieved by introducing the context model of the scenes [Malisiewicz and Efros, 2009].

10.5.4 Correlations between Image data and Sales Data

Since our work is the first attempt on exploring online photo collections for the study of brand associations, we additionally investigate some correlations between the image data and sales data of the brands. We conduct two different comparisons. First, we observe how the photo popularity

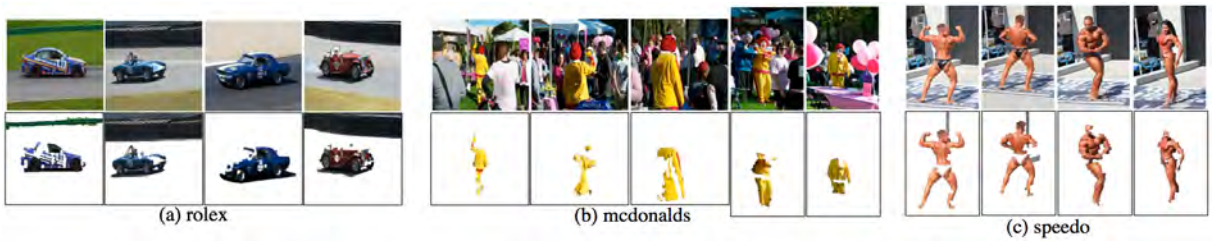


Figure 10.10: 3 groups of examples for typical failure cases. We sample four or five images per group that belong to the same cluster, and thus are jointly segmented. We show input images (top) and their segmentation output (bottom).

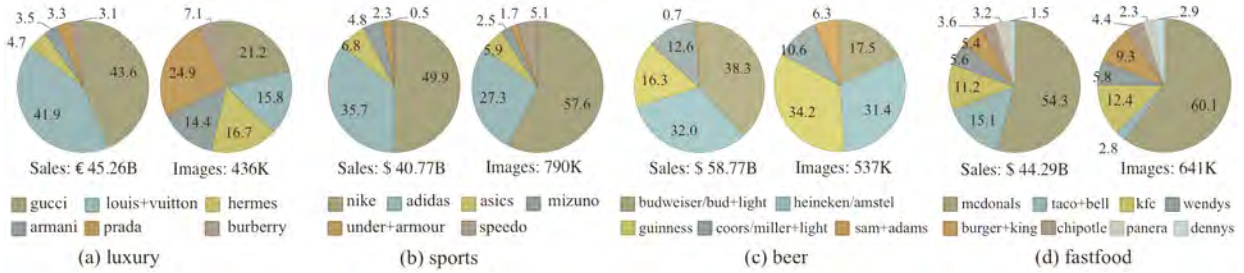


Figure 10.11: Comparison between the market shares (left) and the portions of photo volumes (right) for the brands of four categories: (a) *luxury*, (b) *sports*, (c) *beer*, and (d) *fastfood*. The numbers indicate percentage values.

is correlated with the market share of brands. For example, the average annual revenue of the *Nike* is higher than that of the *Adidas* by about 40% from 2006 to 2011. We examine whether the *Nike* is also dominant over the *Adidas* in the volumes of Web photos. Second, we study in-depth correlation between the product groups of each brand. For example, the annual reports of the *Louis+Vuitton* classify their business into several product groups such as leather goods, perfume, jewelry, and wine. We compare between the proportions of product groups in image data and sales data of the brand.

We obtain the sales data from the annual reports that are publicly available on the companies' webpages. We ignore some brands that are held by private companies (*e.g.* *Chanel*, *New+Balance*), because it is often hard to obtain accurate financial information. In this analysis, we use images and sales data from 2006 to 2011.

Correlation between photo popularity and market share: Fig.10.11 summarizes the proportions of photo volumes and market shares for the brands per category, which can be computed from the dataset sizes of Fig.10.2 and revenue data of annual reports. As shown in Fig.10.11, the ranking of the brands in the two data modalities are roughly similar, but the percentage values do not necessarily agree each other because the preferred scenes or situations of photo taking are different from those of product purchase. For example, the *Guinness* has a larger percentage value in the photo volumes than in the sales data thanks to its positioning as premium beer. An example of its opposite may be the *Taco+Bell*, which occupies a small portion of photo volumes. It may be because the *Taco+Bell* is a cheap fastfood brand, which may not attract people to take pictures for the brand.

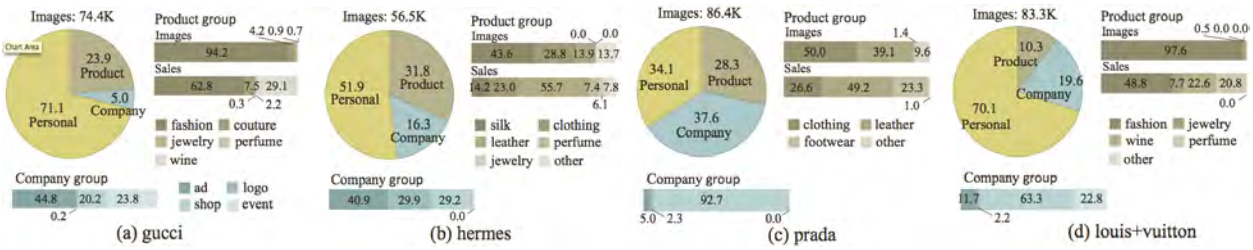


Figure 10.12: Results of the product group analysis for four luxury brands. Each pie chart shows the proportions of three groups in the image volume: *product*, *company*, and *personal*. In the bottom, the images of the *company* group are further classified into one of *advertisement*, *logo*, *shop*, and *event*. In the right, bar charts show the proportions of the images (top) and the actual revenues (bottom) for the *product* group. The classification of product groups is based on the brand's annual reports. The numbers indicate percentage values.

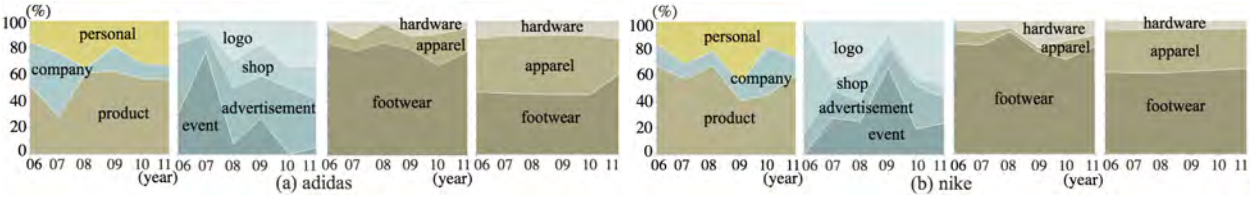


Figure 10.13: Results of the product group analysis per year for two competing sports brands, *Adidas* and *Nike*. The figure illustrates the same information to Fig.10.12 only except showing the temporal variations from 2006 to 2011.

Correlation between product groups: While the first analysis compares between different brands in the same category, now we turn to the comparison between product groups in each brand. The main challenge here is that it is extremely difficult for both human and computers to correctly classify millions of images into the predefined product groups. For human, the dataset sizes are too large to manually classify them. For computers, there is no classification algorithm that is applicable to noisy Web images with high accuracies. Thus, we take advantage of our exemplar detection/clustering results. We manually classify each exemplar into one of predefined groups, and all the images in the same cluster are labeled as the same.

Fig.10.12 shows the results of product group analysis for four luxury brands: *Gucci*, *Hermes*, *Prada*, and *Louis+Vuitton*. We first label exemplar images by one of three groups: *product*, *company*, and *personal*. The *product* group comprises the photos whose main contents are the products of the brand. The *company* group includes the images that are relevant to the brand but do not associate with any particular products. It consists of four subgroups: *advertisement*, *logo*, *shop*, and *event*. The final one is the *personal* group for the private pictures that are not explicitly associated with brands. In Fig.10.12, each pie chart shows the proportion of photos for three *product*, *company*, and *personal* groups. In the bottom, we represent the sub-classification results of the *company* group. In the right, bar charts show the proportions of the images (top) and the actual revenues (bottom) for the *product* group. The classification of product groups is based on the brand's annual reports.

We summarize several observations as follows. First, in most brands, the *personal* group is the first or second largest among the three groups, which may result from that people usually take pictures on personal matters (*e.g.* their dogs, cars, or portraits) and much of them are likely to be poorly labeled as brand names. Second, the *company* group is also very popular; for examples, people enjoy taking pictures on the *Louis+Vuitton*'s stores, and bookmarking their advertisements. Moreover, the events hosted by brands are also popularly taken such as fashion shows, music concerts, and sports activities. Third, in the *product* group, one or two leading product types per brand take the majority of photo volumes while some product segments like *wines*, *perfume*, and *jewelry* rarely appear.

Fig.10.13 shows the results of the same analysis for two competing sports brands, *Adidas* and *Nike*. Only difference is that we conduct the analysis separately per year from 2006 to 2011, in order to observe temporal variation of popularity of photo groups. The fourth graph in each set shows the sales data of three product types, *footwear*, *apparel*, and *hardware*, which are very consistent all years of the range. On the other hand, image data show a lot of fluctuation, which may be caused by other external sports events like World cups, NBA finals, or Olympics. The study on the correlation between popularity of brand images and external events could be another intriguing future project.

10.6 Summary

In this chapter, we develop important building blocks toward the study of photo-based brand associations. The main contributions of this chapter are summarized as follows.

- We study the problem of visualizing brand associations in both image and sub-image levels by leveraging large-scale online pictures. To the best of our knowledge, our work is the first attempt so far on such photo-based brand association analysis. Our work can provide another novel and complementary way to visualize general public's impressions or thoughts on the brands.
- We develop an algorithm to jointly achieve exemplar detection/clustering and brand localization tasks in a mutually-rewarding way. In addition, we design a novel embedding algorithm to visualize the top exemplars/clusters in a circular layout.
- With experiments on about five million images of 48 brands, we have found that the proposed algorithms can comprehensively but succinctly visualize key concepts of large-scale brand image collections. We also quantitatively demonstrate that our approach outperforms other candidate methods on both visualization tasks.

Part IV

Conclusion

Chapter 11

Discussion

This dissertation presents a considerable step towards reconstructing collective storylines from huge image collections shared online, and leveraging them to explore novel applications at the intersection of computer vision and multimedia data mining. We believe that this direction of research becomes more important and anticipating in prevailing Internet era. In this chapter, we summarize the contributions and key observations of our work again, and discuss future research directions that go beyond our current achievement.

11.1 Key Observations and Contributions

As the concluding remarks of this thesis, we recapitulate the key observations and contributions.

Understanding Temporal Trends of Web Image Collections

- Our work is one of very few early studies in computer vision literature to model temporal topic evolution of Web image collections. With experiments on 9 millions of images of 47 topics from Flickr, we demonstrate that the dynamic models help solve better three existing or novel computer vision problems. First, we perform subtopic outbreak detection to point out when the topical contents of images rapidly change. Second, we present that the images can be a complementary source of information beyond tag texts for discovering topical evolution. Finally, we empirically show that training using temporal context can improve object classification performance for extremely diverse Web images.
- We then take advantage of the temporal models of image collections to improve the performance of image ranking and retrieval. On the technical aspect, we design novel scalable algorithm using multi-task regression on multivariate point processes, on which we build the temporal models to rank images based on temporal suitability. We also extend this framework into collective and personalized Web image prediction, which can estimate likely pictures at any future time point.

Discovering Overlapping Contents of Image Collections

- We develop a scalable method for discovering regions of interest (ROI) in the form of bounding boxes from large-scale Web image collections (*e.g.* up to 200K Flickr images). The

unsupervised ROI detection is achieved by alternating optimization based on link analysis techniques, in which we iteratively solve two sub-problems: (1) finding exemplars of objects in the dataset and (2) localizing object instances in each image. Through experiments, we show that our scalable approach achieves compelling performance for variable Flick images without any human annotation or initial seed images.

- We propose a diffusion-based optimization framework that is applicable to a wide range of computer vision problems. We prove that the *temperature* of a linear anisotropic diffusion system, which corresponds to many important objectives in computer vision tasks, is a submodular function. We show that our optimization leads to an effective solution to diversity ranking, single-image segmentation, and cosegmentation. Finally, we present a distributed cosegmentation *CoSand*, which has some unique benefits including compelling performance over previous methods, superior scalability, ability to automatically decide the number of foregrounds K, and robustness against a wrong choice of K.
- We then develop a less restrictive and more practical cosegmentation algorithm in order to be applicable to general users' photo streams, in which a finite number of foregrounds (*i.e.* subjects of interest) irregularly occur in each of input images. Our approach alternates between solving two subtasks: foreground modeling and region assignment. In particular, our approach is flexible enough to integrate any advanced region classifiers for foreground modeling, and our region assignment employs a combinatorial auction framework that enjoys several intuitively good properties such as optimality guarantee and linear complexity.

Reconstruction and Applications of Photo Storylines

- As a first technical step to achieve the goal of inferring collective photo storylines, we propose a method to jointly perform *alignment* of multiple photo streams and *cosegmentation* of aligned images. The alignment is a core task to build a big picture of storylines from a large number of fragmented photo streams of individual users. The cosegmentation can facilitate image understanding such as pixel-level classification in the images by segmenting the aligned images together. We close a loop between solving the two tasks so that solving one task enhances the performance of the other in a mutually-rewarding way. To this end, we design scalable message-passing based optimization framework to jointly achieve both tasks for the whole input image set at once. We show the superior performance and scalability of our approach with experiments on the new Flickr dataset of 15 outdoor recreational activities.
- We then investigate the problem of reconstructing storyline graphs from large-scale photo collections, and optionally other side information such as friendship graphs. We formulate the storyline reconstruction problem as an inference of sparse time-varying directed graphs, and develop an optimization algorithm that achieves a number of key challenges of Web-scale applications, including global optimality, linear complexity, and easy parallelization. We qualitatively show that the storyline graphs can visualize various events or activities recurring across the input photo sets, which otherwise are too overwhelming for users to grasp any underlying big picture. In addition, we quantitatively validate that the storyline

graphs help solve better two image sequence prediction tasks, which are predicting next likely images and filling in missing parts of photo streams.

- Finally, we discover brand associations in both image and sub-image levels by leveraging large-scale online photo collections. Brand associations, one of central concepts in marketing, describe customers' top-of-mind attitudes or feelings to a brand. While brand associations are traditionally measured by analyzing the text data from consumers' responses to the survey or their online conversation logs, our work is the first attempt so far on picture-based brand association study. We first detect core visual concepts associated with brands and visualize them in a circular layout. We then identify the regions of brand in each image, which can potentiate several interesting applications such as content-based image retrieval and online multimedia advertisement. With experiments on about five million images of 48 brands, we have found that the proposed algorithms provide complementary way to visualize general public's impressions or thoughts on the brands. Moreover, We quantitatively show that our approach outperforms other candidate methods on both visualization tasks.

11.2 Future Directions

In this dissertation, we have proposed a set of algorithms that are required to reconstruct collective storylines from Web image collections. In spite of their significant achievement for a wide range of novel and challenging problems, we believe much remains to be done along this line of research. Here, we propose several ideas for future projects to explore further the extension of our approach.

Time-Sensitive image retrieval for real web search

In Chapter 4, we developed the regression-based algorithms for time-sensitive image retrieval and Web image prediction. Since it is the first work to use the temporal dimension for Web image search and prediction, we mainly focused on demonstrating the feasibility of our algorithms. Although the proposed temporal modeling is shown to be interesting and convincing, there are still several points of improvement for the actual implementation of Web image search. In the following, we enumerate several ideas that can push our algorithms to be more realistic.

First of all, we may need to perform more in-depth statistical analysis for the temporal aspects of Web image corpus. In Chapter 4, we tested less than 50 topic keyboards, which are still limited in the coverage of topics. We can carry out the experiments on much more topic keywords (*e.g.* at least hundreds), in order to systemically analyze what types of queries are time-sensitive in actual Web image search. Moreover, it is very common in real-world image retrieval that multiple query terms are correlated one another or a single image can be associated with multiple keywords. Therefore, the correlation studies are encouraged; our framework can be easily extended for that purpose, because it builds on one of widely studied statistical models: the multi-task regression on multivariate point processes.

One intriguing work in recent information retrieval is *recency ranking*, which refers to ranking documents by relevance that takes freshness into account [Dong et al., 2010]. This is relevant to our work because the query time is usually *now* for general users, and the freshness can be an

important temporal attribute for buzz keywords or breaking news queries. Therefore, introducing the recency factor into our framework can be another interesting extension.

For the technical improvement, we may exploit the idea of the *reduced rank regression* [Chen and Huang, 2012], which can be easily integrated with our current formulation. It is one of popular latent space based regression methods that can naturally take advantage of correlations among multiple descriptors, and provide very fast online learning especially for personalized models. As an example, the reduced rank regressions have been successfully used in the recommendation of News articles [Agarwal et al., 2010], in order to significantly reduce computational complexity in the online learning phase.

Finally, we may boost the performance of Web image prediction by jointly learn the temporal models along with other meta-data that were not used yet. They include GPS information, associated text data (*e.g.* titles, comments, favs in Flickr), social networks of users, and user click data, to name a few.

Understanding the economic behaviors of Web users using computer vision techniques

Recently, the Web has become a new medium where a variety of economic phenomena interplay, such as Web advertising, Internet auctions, markets and exchanges, and social and crowdsourcing commerce. In Chapter 10, we solved the problem to visualize the brand associations from Web photo collections, as one of very few early attempts that analyze the economic behaviors of Web users by leveraging online images. We believe that there are many interesting research problems to be explored along this line of work, given that online commerce and advertising are emerging fields in data mining but images have drawn less attention compared to text data yet.

The first possible extension of our work in Chapter 10 may be image based *brand profiling*, in which we identify a list of customer-facing qualities of brands in competitive marketplaces. The images of brand that are taken by general users can describe natural scenes of interactions between users and products of brand, including how customers generally use the products in real lives, and what benefits users enjoy with the products, and so on. The reconstructed brand profiles from images can be used in a wide range of applications toward online brand advertising such as improving audience identification and ad selection.

One limitation of our brand association method is lack of competitiveness analysis. We simply visualized the key associations of several competing brands, but did not perform any in-depth comparative studies: for example, which of *Nike* and *Adidas* is more popular for running shoes? What are the features of bags of *Louis Vuitton* that most affect its competitiveness over other luxury brands? Such comparative research of brands has not been explored much especially in the image domain, but some recent work in competitor mining can be remotely relevant reference to provide some intuitions [Wan et al., 2011; Lappas et al., 2012].

Finally, another interesting area of work related to ours is *contextual image advertising* [Mei et al., 2012], whose goal is, given an image, to generate keywords or categories that describe the image best and find out the most relevant advertisements, as the sponsored search does with text queries (*e.g.* Google AdWords and Bing Ads). Since this area is still largely under-addressed, our work can be extended to this end, especially for multimedia brand advertising.

Story reconstruction and Visualization

We believe that the storyline reconstruction, the main theme of this thesis, can be extended into many different directions.

First of all, we can improve the photo-based storyline reconstruction of Chapter 8–9 using other information modalities available on the Web. Especially, using both pictures and videos can be synergetic for several reasons. First, today's photo taking devices also include the functions of camcorders, so users can seamlessly record their memorable moments via both pictures and videos. Second, more importantly, the pictures and videos have different characteristics as media, which can be complementary each other for the purpose of storyline reconstructions. The strength of images over videos lies in that people usually pay more attention to take pictures so that they can capture the objects and events from canonical viewpoints in a maximally informative way. On the other hand, when general users record their events with videos, they include much noisy information with only a smaller faction of frames where interesting events really happen. Therefore, the images can clean up noisy parts of videos, and efficiently summarize the videos by selecting only most important frames. In the reverse direction, videos can convey sequential information between frames, which are not available in images. Thus, the videos can be used to glue a set of fragmented images into coherent threads of storylines.

Another most demanding future directions for the storyline reconstruction research would be to develop its real applications under more specific and practical scenarios. We believe that one promising and interesting example is the storylines for the *theme parks*. The popular storylines in the theme parks vary much according to visitor types and visiting time. For example, families with kids may move slowly stroll through the park and prefer to visit character attractions that children like, which will be quite different from those that groups of young people do. Since most of theme parks are too large for visitors to explore in a single day, the reconstructed storylines can be useful to recommend temporally and spatially personalized visiting paths according to types of the visitors. From the technical perspective, theme parks are heavily recorded by millions of visitors and pre-installed surveillance videos, which can provide sufficient amount of image and video data for storyline reconstruction. In addition, theme parks are open spaces with no geometric constraints available to organize all the pictures, which is another important technical challenge to be addressed.

11.3 Conclusion

In this dissertation, we first identify several important characteristics of today's image acquisition, processing, and sharing, which are attributed by recent technical progresses in photo-taking devices, ubiquitous high-speed Internet connection, and social networking. From these new challenges, we derive the thesis statement, which we would like to restate here as follows.

> *Given large-scale online image collections and associated meta-data, we aim to create the collective storylines by jointly inferring the temporal trends and the overlapping contents of image collections. We also explore novel computer vision and data mining applications taking advantage of the reconstructed photo storylines.*

We categorize the required technologies into three research directions, which are (i) understanding of temporal trends of image collections, (ii) discovery of overlapping contents across image collections, and (iii) reconstruction and applications of collective photo storylines. All developed algorithms are aligned to accomplish the proposed research goal. We hope that this thesis can inspire others to pursue more interesting and practical projects at the intersection of computer vision and Web data mining.

Bibliography

David A. Aaker. Measuring Brand Equity Across Products and Markets. *Cal. Manag. Rev.*, 38(3): 102–120, 1996.

Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Fast Online Learning through Offline Initialization for Time-sensitive Recommendation. In *KDD*, 2010.

Amr Ahmed, Qirong Ho, Jacob Eisenstein, Eric P. Xing, Alexander J. Smola, and Choon Hui Teo. Unified Analysis of Streaming News. In *WWW*, 2011.

Narendra Ahuja and Sinisa Todorovic. Learning the Taxonomy and Models of Categories Present in Arbitrary Images. In *ICCV*, 2007.

Navot Akiva, Eliyahu Greitzer, Yakir Krichman, and Jonathan Schler. Mining and Visualizing Online Web Content Using BAM: Brand Association Map. In *ICWSM*, 2008.

Giuseppe Amodeo, Roi Blanco, and Ulf Brefeld. Hybrid Models for Future Event Prediction. In *CIKM*, 2011.

M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, 2002.

Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally Weighted Learning. *AI Review*, 11(1):11–73, 1997.

Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. iCoseg: Interactive Co-segmentation with Intelligent Scribble Guidance. In *CVPR*, 2010.

Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. Interactively Co-segmentating Topically Related Images with Intelligent Scribble Guidance. *IJCV*, 93:273–292, 2011.

Suzanna Becker. Implicit Learning in 3D Object Recognition: The Importance of Temporal Context. *Neural Computation*, 11(2):347–374, 1999.

Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *neural computation*, 15(6):1373–1396, 2003.

Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *ECCV*, 2010.

Paul J. Besl and Neil D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

David M. Blei and John D. Lafferty. Dynamic Topic Models. In *ICML*, 2006.

Oren Boiman, Eli Shechtman, and Michal Irani. In Defense of Nearest-Neighbor Based Image Classification. In *CVPR*, 2008.

Ilaria Bordino, Stefano Battiston, Guido Caldarelli, Matthieu Cristelli, Antti Ukkonen, and Ingmar Weber. Web Search Queries Can Predict Stock Market Volumes. *PloS one*, 7(7), 2012.

Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image Classification using Random Forests and Ferns. In *ICCV*, 2007.

Matthew Boutell, Jiebo Luo, and Christopher Brown. A Generalized Temporal Context Model for Classifying Image Collections. *Multimedia Systems*, 11(1):82–92, 2005.

Yuri Boykov and Marie-Pierre Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *ICCV*, 2001.

Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *WWW*, 1998.

Emery N. Brown, Riccardo Barbieri, Valerie Ventura, Robert E. Kass, and Loren M. Frank. The Time-Rescaling Theorem and Its Application to Neural Spike Train Data Analysis. *Neural Computation*, 14(2):325–346, 2001.

Andres Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods. *IJCV*, 61(3):211–231, 2005.

Liangliang Cao, Jiebo Luo, Henry Kautz, and Thomas S. Huang. Annotating Collections of Photos Using Hierarchical Event and Scene Models. In *CVPR*, 2008.

Keven M. Carter, Raviv Raich, and Alfred O. Hero. Spherical laplacian information maps (slim) for dimensionality reduction. In *SSP*, 2009.

Arthur Cheng-Hsui Chen. Using Free Association to Examine the Relationship between the Characteristics of Brand Associations and Brand Equity. *J. Product Brand Management*, 10(7): 439–451, 2001.

Chao-Yeh Chen and Kristen Grauman. Clues from the Beaten Path: Location Estimation with Bursty Sequences of Tourist Photos. In *CVPR*, 2011.

Lisha Chen and Jiahua Z. Huang. Sparse Reduced-Rank Regression for Simultaneous Dimension Reduction and Variable Selection. *J. American Statistical Association*, 107(500):1533–1545, 2012.

Xi Chen, Qihang Lin, Seyoung Kim, Jaime G. Carbonell, and Eric P. Xing. Smoothing Proximal Gradient Method for General Structured Sparse Learning. In *UAI*, 2011.

Hyunyoung Choi and Hal Varian. Predicting the Present with Google Trends. *Econ. Record*, 88: 2–9, 2012.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE trans. Information Theory*, 14:462–467, 1968.

Ondrej Chum and Andrew Zisserman. An Exemplar Model for Learning Object Classes. In *CVPR*, 2007.

Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *ICCV*, 2007.

Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei. Towards Scalable Dataset Construction: An Active Learning Approach. In *ECCV*, 2008.

Dorin Comaniciu and Peter Meer. Mean Shift: A Robutst Approach Toward Feature Space Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active Appearance Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *CVPR*, 2005.

Koby Crammer and Yoram Singer. On the Learnability and Design of Output Codes for Multiclass Problems. *Machine Learning*, 47:201–233, 2002.

Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. The MIT Press, 2005.

David Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-Continuous Optimization for Large-Scale Structure from Motion. In *CVPR*, 2011.

Jingyu Cui, Fang Wen, and Xiaoou Tang. Real Time Google and Live Image Search Re-ranking. In *ACM MM*, 2008.

Wisam Dakka, Luis Gravano, and Panagiotis G. Ipeirotis. Answering General Time-Sensitive Queries. In *CIKM*, 2008.

D.J. Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer, 2003.

Jeffrey E. Danes, Jeffrey S. Hess, John W. Story, and Jonathan L. York. Brand Image Associations for Large Virtual Groups. *Qualitative Market Research*, 13(3):309–323, 2010.

Abhinandan S. Das, Mayur Datar, , Ashutosh Garg, and Shyam Rajaram. Google News Personalization: Scalable Online Collaborative Filtering. In *WWW*, 2007.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. Towards Recency Ranking in Web Search. In *WSDM*, 2010.

Wei Dong, Moses Charikar, and Kai Li. Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures. In *WWW*, 2011.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html, 2010.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Belief Propagation for Early Vision. *IJCV*, 70(1):41–54, 2006.

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IJCV*, 32:1627–1645, 2010.

Rob Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. Learning Object Categories from Google's Image Search. In *ICCV*, 2005.

Brendan J. Frey and Delbert Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315:972–976, 2007.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Statistical Software*, 33:1–22, 2010.

Wenjiang J. Fu. Penalized Regressions: The Bridge Versus the Lasso. *J. Computational Graphical Statistics*, 7:397–416, 1998.

Ke Gao, Shouxun Lin, Yongdong Zhang, Sheng Tang, and Dongming Zhang. Logo Detection Based on Spatial-Spectral Saliency and Partial Spatial Context. In *ICME*, 2009.

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Discovering Diverse and Salient Threads in Document Collections. In *EMNLP-CoNLL*, 2012.

Sharad Goel, Jake M. Hofman, Sebastien Lahaie, David M. Pennock, and Duncan J. Watts. Predicting Consumer Behavior with Web Search. *PNAS*, 2010.

Joseph Gonzalez, Yucheng Low, and Carlos Guestrin. Residual Splash for Optimally Parallelizing Belief Propagation. In *AISTATS*, 2009.

Leo Grady. Random Walks for Image Segmentation. *IEEE PAMI*, 28:1768–1783, 2006.

Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding Videos, Constructing Plots: Learning a Visually Grounded Storyline Model from Annotated Videos. In *ICCV*, 2009.

Maurice Halbwachs. *On Collective Memory*. Univ of Chicago Press, 1992.

Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Dorit S. Hochbaum and Vikas Singh. An Efficient Algorithm for Co-segmentation. In *ICCV*, 2009.

Winston H. Hsu, Lyndon S. Kennedy, and Shih-Fu Chang. Video Search Reranking Through Random Walk over Document-Level Context Graph. In *ACM MM*, 2007.

Michael Isard and Andrew Blake. CONDENSATION – Conditional Density Propagation for Visual Tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.

Xin Jin, Andrew Gallagher, Liangliang Cao, Jiebo Luo, and Jiawei Han. The Wisdom of Social Multimedia: Using Flickr for Prediction and Forecast. In *ACM MM*, 2010.

Yushi Jing and Shumeet Baluja. VisualRank, PageRank for Google Image Search. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(11):1–31, 2008.

Thorsten Joachims. Optimizing Search Engines Using Clickthrough Data. In *KDD*, 2002.

Armand Joulin, Francis Bach, and Jean Ponce. Discriminative Clustering for Image co-segmentation. In *CVPR*, 2010.

Armand Joulin, Francis Bach, and Jean Ponce. Multi-Class Cosegmentation. In *CVPR*, 2012.

Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei A. Efros, and Aaron Hertzmann. Image Sequence Geolocation with Human Travel Priors. In *ICCV*, 2009.

Hongwen Kang, Martial Hebert, Alexei A. Efros, and Takeo Kanade. Connecting Missing Links: Object Discovery from Sparse Observations Using 5 Million Product Images. In *ECCV*, 2012.

Kevin Lane Keller. Conceptualizing, Measuring, and Managing Customer-Based Brand Equity. *J. Marketing*, 57(1):1–22, 1993.

Gunhee Kim and Antonio Torralba. Unsupervised Detection of Regions of Interest using Iterative Link Analysis. In *NIPS*, 2009.

Gunhee Kim and Eric P. Xing. On Multiple Foreground Cosegmentation. In *CVPR*, 2012.

Gunhee Kim and Eric P. Xing. Jointly Aligning and Segmenting Multiple Web Photo Streams for the Inference of Collective Photo Storylines. In *CVPR*, 2013a.

Gunhee Kim and Eric P. Xing. Time-Sensitive Web Image Ranking and Retrieval via Dynamic Multi-Task Regression. In *WSDM*, 2013b.

Gunhee Kim and Eric P. Xing. Reconstructing Collective Storyline Graphs from Web Community Photo Collections. In *CVPR*, 2014a. (*Submitted*).

Gunhee Kim and Eric P. Xing. Visualizing Brand Associations from Web Community Photos. In *WSDM*, 2014b. (*Submitted*).

Gunhee Kim, Christos Faloutsos, and Martial Hebert. Unsupervised Modeling of Object Categories Using Link Analysis Techniques. In *CVPR*, 2008a.

Gunhee Kim, Christos Faloutsos, and Martial Hebert. Unsupervised Modeling and Recognition of Object Categories with Combination of Visual Contents and Geometric Similarity Links. In *ACM MIR*, 2008b.

Gunhee Kim, Eric P. Xing, and Antonio Torralba. Modeling and Analysis of Dynamic Behaviors of Web Image Collections. In *ECCV*, 2010.

Gunhee Kim, Eric P. Xing, Li Fei-Fei, and Takeo Kanade. Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion. In *ICCV*, 2011.

Gunhee Kim, Li Fei-Fei, and Eric P. Xing. Web Image Prediction Using Multivariate Point Processes. In *KDD*, 2012.

Jim Kleban, Xing Xie, and Wei-Ying Ma. Spatial Pyramid Mining for Logo Detection in Natural Scenes. In *ICME*, 2008.

Mladen Kolar and Eric P. Xing. Sparsistent estimation of time-varying markov random fields, 2013. arXiv:0907.2337.

Mladen Kolar, Le Song, Amr Ahmed, and Eric P. Xing. Estimating Time-Varying Networks. *Ann. Appl. Stat.*, 4(1):94–123, 2010.

Andreas Krause and Carlos Guestrin. Beyond Convexity: Submodularity in Machine Learning. In *ICML Tutorials*, 2008.

Anagha Kulkarni, Jaime Teevan, Krysta M. Svore, and Susan T. Dumais. Understanding Temporal Query Dynamics. In *WSDM*, 2011.

Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *PAMI*, 31:2129–2142, 2009.

Theodoros Lappas, George Valkanas, and Dimitrios Gunopulos. Efficient and Domain-Invariant Competitor Mining. In *KDD*, 2012.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006.

Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image Segmentation with A Bounding Box Prior. In *ICCV*, 2009.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective Outbreak Detection in Networks. In *ACM KDD*, 2007.

Alex Levinshtein, Adrian Stere, Kiriakos Kutulakos, David Fleet, Sven Dickinson, and Kaleem Siddiqi. TurboPixels: Fast Superpixels Using Geometric Flows. *IEEE PAMI*, 31(12):2290–2297, 2009.

Li-Jia Li, Gang Wang, and Li Fei-Fei. OPTIMOL: Automatic Object Picture collecTion via Incremental MOdel Learning. In *CVPR*, 2007.

Yunpeng Li, David J. Crandall, and Daniel P. Huttenlocher. Landmark Classification in Large-scale Image Collections. In *ICCV*, 2009.

Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. SIFT Flow: Dense Correspondence across Different Scenes. In *ECCV*, 2008.

Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment. In *CVPR*, 2009a.

Han Liu, Mark Palatucci, and Jian Zhang. Blockwise Coordinate Descent Procedures for the Multi-task Lasso, with Applications to Neural Semantic Basis Discovery. In *ICML*, 2009b.

Tie Liu, Jian Sun, Nan-Ning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to Detect A Salient Object. In *CVPR*, 2007.

Wei Liu, Yu-Gang Jiang, Jiebo Luo, and Shih-Fu Chang. Noise Resistant Graph Ranking for ImprovedWeb Image Search. In *CVPR*, 2011.

David MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2002.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online Dictionary Learning for Sparse Coding. In *ICML*, 2009.

Tomasz Malisiewicz and Alexei A. Efros. Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships. In *NIPS*, 2009.

Jean M. Mandler and Nancy S. Johnson. Remembrance of Things Parsed: Story Structure and Recall. *Cognitive Psychology*, 9(1):111–151, 1977.

Luca Marchesotti, Claudio Cifarelli, and Gabriela Csurka. A Framework for Visual Saliency Detection with Applications to Image Thumbnailing. In *ICCV*, 2009.

Tao Mei, Lusong Li, Xian-Sheng Hua, and Shipeng Li. ImageSense: Towards Contextual Image Advertising. *ACM T. Multimedia Computing, Communications and Applications*, 8(1):89–115, 2012.

Nicolai Meinshausen and Peter Bühlmann. High-Dimensional Graphs and Variable Selection with the Lasso. *Ann. Statist.*, 34(3):1436–1462, 2006.

Donald Metzler, Rosie Jones, Fuchun Peng, and Ruiqiang Zhang. Understanding Temporal Query Dynamics. In *SIGIR*, 2009.

Hemant Misra, Frank Hopfgartner, Anuj Goyal, P. Punitha, and Joemon M. Jose. TV News Story Segmentation Based on Semantic Coherence and Content Similarity. In *MMM*, 2010.

Nobuyuki Morioka and Jingdong Wang. Robust Visual Reranking via Sparsity and Ranking Constraints. In *ACM MM*, 2011.

Lopamudra Mukherjee, Vikas Singh, and Jiming Peng. Scale Invariant Cosegmentation for Image Groups. In *CVPR*, 2011.

Douglas L. Nelson, Cathy L. McEvoy, and Thomas A. Schreiber. The University of South Florida Free Association, Rhyme, and Word Fragment Norms. *Behavior Research Methods, Instruments, Computers*, 36(3):402–407, 2004.

Douglas L. Nelson, Bunvor M. Dyrdal, and Leilani B. Goodmon. What is Preexisting strength? Predicting Free Association Probabilities, Similarity Ratings, and Cued Recall Probabilities. *Psychonomic Bulletin Review*, 12(4):711–719, 2005.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions. *Math. Prog.*, 14:265–294, 1978.

Pere Obrador, Rodrigo de Oliveira, and Nuria Oliver. Supporting Personal Photo Storytelling for Social Albums. In *MM*, 2010.

Bruno A. Olshausen and David J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

Nielsen Online. Brand Association Map, 2010. URL http://www.nielsen-online.com/downloads/us/BAM_US.pdf.

Lucas Paletta, Manfred Prantl, and Axel Pinz. Learning Temporal Context in Active Object Recognition Using Bayesian Analysis. In *ICPR*, 2000.

Karthir Prabhakar, Sangmin Oh, Ping Wang, Gregory Abowd, and James M. Rehg. Temporal Causality and the Analysis of Interactions in Video. In *CVPR*, 2010.

Till Quack, Bastian Leibe, and Luc Van Gool. World-scale Mining of Objects and Events from Community Photo Collections. In *CIVR*, 2008.

Ariadna Quattoni and Antonio Torralba. Recognizing Indoor Scenes. In *CVPR*, 2009.

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. Predicting the News of Tomorrow Using Patterns in Web Search Queries. In *PI*, 2008.

Kira Radinsky, Krysta M. Svore, Susan T. Dumais, Jaime Teevan, Alex Bocharov, and Eric Horvitz. Modeling and Predicting Behavioral Dynamics on the Web. In *WWW*, 2012.

Usha Nandini Raghavan, Reka Albert, and Soundar Kumara. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. *Phys Rev E*, 76(036106), 2007.

Toni M. Rath and R. Manmatha. Word Image Matching Using Dynamic Time Warping. In *CVPR*, 2003.

Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. In *ACM SIGIR*, 2007.

Mark O. Riedl and R. Michael Young. From Linear Story Generation to Branching Story Graphs. *IEEE Computer Graphics and Applications*, 26(3):23–31, 2006.

Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the Temporal Dynamics of Diffusion Networks. In *ICML*, 2011.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The Author-Topic Model for Authors and Documents. In *UAI*, 2004.

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut – Interactive Foreground Extraction using Iterated Graph Cuts. In *SIGGRAPH*, 2004.

Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of Image Pairs by Histogram Matching Incorporating a Global Constraint into MRFs. In *CVPR*, 2006.

Olga Russakovsky and Andrew Y. Ng. A Steiner Tree Approach to Efficient Object Detection. In *CVPR*, 2010.

Brian C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A Database and Web-based Tool for Image Annotation. *IJCV*, 77:157–173, 2008.

Bryan C. Russell and Antonio Torralba. Building a Database of 3D Scenes from User Annotations. In *CVPR*, 2009.

Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, 2006.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009.

Tuomas Sandholm and Subhash Suri. BOB: Improved Winner Determination in Combinatorial Auctions and Generalizations. *Artificial Intelligence*, 145:33–58, 2003.

Subhajit Sanyal and S. H. Srinivasan. LogoSeeker: A System for Detecting and Matching Logos in Natural Images. In *ACM MM*, 2007.

Grant Schindler and Frank Dellaert. Probabilistic Temporal Inference on Reconstructed 3D Scenes. In *CVPR*, 2010.

Oliver Schnittka, Henrik Sattler, and Sebastian Zenker. Advanced Brand Concept Maps: A New Approach for Evaluating the Favorability of Brand Association Networks. *I. J. Research in Marketing*, 2012.

Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Harvesting Image Databases from the Web. In *ICCV*, 2007.

Dafna Shahaf and Carlos Guestrin. Connecting the Dots Between News Articles. In *KDD*, 2010.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. Trains of Thought: Generating Information Maps. In *WWW*, 2012.

Alexander Shekhovtsov, Ivan Kovtun, and Vaclav Hlavac. Efficient MRF Deformation Model for Non-Rigid Image Matching. In *CVPR*, 2007.

Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Vivek K. Singh, Mingyan Gao, and Ramesh Jain. Social Pixels: Genesis and Evaluation. In *ACM MM*, 2010.

Pawan Sinha, Benjamin Balas, Yuri Ostrovsky, and Richard Russell. Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About. *Proceedings of the IEEE*, 94(11):1948–1962, 2006.

Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman. Discovering Objects and Their Location in Images. In *ICCV*, 2005.

Noah Snavely, Ian Simon, Michael Goesele, Richard Szeliski, and Steven M. Seitz. Scene Reconstruction and Visualization from Community Photo Collections. *Proc. IEEE*, 98(8):1370–1390, 2010.

Le Song, Mladen Kolar, and Eric Xing. Time-Varying Dynamic Bayesian Networks. In *NIPS*, 2009.

Jimeng Sun, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. Neighborhood Formation and Anomaly Detection in Bipartite Graphs. In *ICDM*, 2005.

Rob Tibshirani. Regression Shrinkage and Selection via the Lasso. *J. Royal. Statist. Soc B.*, 58(1): 267–288, 1996.

Brian D. Till, Daniel Baack, and Brian Waterman. Strategic Brand Association Maps: Developing Brand Insight. *J. Product Brand Management*, 20(2):92–100, 2011.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast Random Walk with Restart and Its Applications. In *ICDM*, 2006.

Antonio Torralba, Rob Fergus, and Wiliam T Freeman. 80 Million Tiny Images: A Large Dataset for Non-parametric Object and Scene Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

Tom Trabasso and Paul Van Den Broek. Causal Thinking and the Representation of Narrative Events. *J. Memory and Language*, 24:612–630, 1985.

Wilson Truccolo, Uri T. Eden, Matthew R. Fellows, John P. Donoghue, and Emery N. Brown. A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects. *J. Neurophysiol*, 93(2):1074–1089, 2005.

David Tschumperle and Rachid Deriche. Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications. *IEEE PAMI*, 27:506–517, 2005.

Endel Tulving. *Episodic and semantic memory*. E. Tulving and W. Donaldson (Eds.), Organization of Memory. New York: Academic Press., 1972.

Shimon Ullman. *High-Level Vision: Object Recognition and Visual Cognition*. MIT Press, 2000.

Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation Revisited: Modes and Optimization. In *ECCV*, 2010.

Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object Cosegmentation. In *CVPR*, 2011.

David Vickrey, Aaron Bronzan, William Choi, Aman Kumar, Jason Turner-Maier, Arthur Wang, and Daphne Koller. Online Word Games for Semantic Data Collection. In *EMNLP*, 2008.

Sudheendra Vijayanarasimhan and Kristen Grauman. Efficient Region Search for Object Detection. In *CVPR*, 2011.

Ulrike von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Guy Wallis and Heinrich H. Bulthöff. Effects of Temporal Association on Recognition Memory. *PNAS*, 98(8):4800–4804, 2001.

Qian Wan, Raymond Chi-Wing Wong, and Yu Peng. Finding Top-k Profitable Products. In *ICDE*, 2011.

Dingding Wang, Tao Li, and Mitsunori Ogihara. Generating Pictorial Storylines Via Minimum-Weight Connected Dominating Set Approximation in Multi-View Graphs. In *AAAI*, 2012a.

Jing Wang, Jingdong Wang, Gang Zeng, Zhuowen Tu, Rui Gan, and Shipeng Li. Scalable k-NN Graph Construction for Visual Descriptors. In *CVPR*, 2012b.

Xiaogang Wang, Ke Liu, and Xiaoou Tang. Query-Specific Visual Semantic Spaces for Web Image Re-ranking. In *CVPR*, 2011.

Xuerui Wang and Andrew McCallum. Topics Over Time: a Non-Markov Continuous-Time Model of Topical Trends. In *KDD*, 2006.

Joachim Weickert. *Anisotropic Diffusion in Image Processing*. ECMI Series, Teubner-Verlag, 1998.

John Winn and N Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *ICCV*, 2005.

John Winn, Antonio Criminisi, and Thomas Minka. Object Categorization by Learned Universal Visual Dictionary. In *ICCV*, 2005.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In *CVPR*, 2010.

Kota Yamaguchi, M. Hadi Kiapour, Luis E. Ortiz, and Tamara L. Berg. Parsing Clothing in Fashion Photographs. In *CVPR*, 2012.

Jianchao Yang, Jiebo Luo, Jie Yu, and Thomas Huang. Photo Stream Alignment for Collaborative Photo Collection and Sharing in Social Media. In *WSM*, 2011.

Linjun Yang and Alan Hanjalic. Supervised Reranking for Web Image Search. In *ACM MM*, 2010.

Juyong Zhang, Jianmin Zheng, and Jianfei Cai. A Diffusion Approach to Seeded Image Segmentation. In *CVPR*, 2010.

Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the World: building a web-scale landmark recognition engine. In *CVPR*, 2010.

Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on Data Manifolds. In *NIPS*, 2004.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML*, 2003.

Xiaojin Zhu, Andrew B.Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving Diversity in Ranking using Absorbing Random Walks. In *HLT-NAACL*, 2007.