# Advanced Tools for Video and Multimedia Mining

**Jia-Yu Pan**

CMU-CS-06-126

May 12, 2006

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

**Thesis Committee:**
Christos Faloutsos, Co-chair
Howard Wactlar, Co-chair
Christopher Olston
Shih-Fu Chang, Columbia University

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



This research was sponsored by the National Science Foundation under grant nos. IIS-0205219, IIS-0534625, IIS-9817496, EAI-0121641, EF-0331657, and the Department of the Interior under contract no. NBCHC040037. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

**Keywords:** multimedia data mining, video mining, multi-modal pattern discovery, biomedical data mining, independent component analysis, random walk with restarts, image captioning, time series and text mining

## Abstract

How do we automatically find patterns and mine data in large multimedia databases, to make these databases useful and accessible? We focus on two problems: (1) mining "uni-modal patterns" that summarize the characteristics of a data modality, and (2) mining "cross-modal correlations" among multiple modalities. Uni-modal patterns such as "news videos have static scenes and speech-like sounds", and cross-modal correlations like "the blue region at the upper part of a natural scene image is likely to be the 'sky'", could provide insights on the multimedia content and have many applications.

For uni-modal pattern discovery, we propose the method *AutoSplit*. AutoSplit provides a framework for mining meaningful "independent components" in multimedia data, and can find patterns in a wide variety of data modalities (e.g., video, audio, text, and time sequences). For example, in video clips, AutoSplit finds characteristic visual/auditory patterns, and can classify news and commercial clips with 81% accuracy. In time sequences like stock prices, AutoSplit finds hidden variables like "general growth trend" and "Internet bubble", and can detect outliers (e.g., lackluster stocks). Based on AutoSplit, we design a system, ViVo, for mining biomedical images. ViVo automatically constructs a visual vocabulary which is biologically meaningful and can classify 9 biological conditions with 84% accuracy. Moreover, ViVo supports data mining tasks such as highlighting biologically interesting image regions, for biomedical research.

For cross-modal correlation discovery, we propose *MAGIC*, a graph-based framework for multimedia correlation mining. When applied to news video databases, MAGIC can identify relevant video shots and transcript words for event summarization. On the task of automatic image captioning, MAGIC achieves a relative improvement of 58% in captioning accuracy as compared to recent machine learning techniques.

*Dedicated to my parents*

# Acknowledgments

This thesis cannot be completed without the help from many people. I owe my thanks to all of them.

First and foremost, I thank my advisor Christos Faloutsos. I am deeply grateful for his guidance and support. Among other things, I appreciate his encouragement, patience, and (sometimes) his jokes. Thank you, Christos!

I would also like to thank my co-advisor Howard Wactlar. Howard has been giving me advices and supporting me in many ways. My work on video data mining cannot be done without the help from him and the members of the Informedia group.

My thanks go to members of my thesis committee – Shih-Fu Chang and Christopher Olston. Their comments and advices have provided different perspectives on my work, and have made my thesis more complete and easier to read.

Hyungjeong Yang, Pinar Duygulu and Hiroyuki Kitagawa deserve special thanks. I am fortunate to be able to work with them on many projects. We have been productive – eight papers, including two best student paper awards. What a ride!

I thank all of my other co-authors: Arnab Bhattacharya, Vebjorn Ljosa, Mark R. Verardo, Ambuj K. Singh, Masafumi Hamamoto, David A. Forsyth, Jernej Barbič, Alla Safonova, Jessica K. Hodgins, Nancy S. Pollard, and Srinivasan Seshan. Having the chance to work with them has been a blessing.

I am grateful to know Tai Sing Lee, who was my contact professor when I came to CMU and was also the instructor of the computer vision course which I was a TA. The discussions with him are inspiring and have given me many ideas in my research.

I also owe many thanks to the friends and colleagues at the Informedia group, the CMU Database group, and the CMU Center for Bioimage Informatics. My special thanks go toward Anastassia Ailamaki, Alexander Hauptmann, Michael Christel, and Robert F. Murphy, for their encouragement and inspiration. The DB group buddies have made my days fun and memorable. Especially, I thank Deepay Chakrabarti, Leejay Wu, Spiros Papadimitriou, Jimeng Sun, Hanghang Tong, Jose F. Rodrigues Jr., and André Guilherme Ribeiro Balan.

I cannot survive the Ph.D. study without the help from my wonderful officemates. Especially, I thank Mark Moll, Patrick Riley, and Katrina Ligett. Mark taught me almost everything about doing research at CMU – giving me tips on computing at CMU, solving my programming/LaTeX problems, and so on. Patrick has been a good model of a graduate student, who does good research and manages to have an interesting life outside of school. Katrina has been very supportive at the most stressful time of my graduate study.

Special thanks go to my musician sister Chialin, my bio-statistician friend Joseph Ko, my physicist friend Feng Wu, and all the members of the Pittsburgh Tzu Chi Foundation, for all the good times we had and their help during the tough times.

I thank Sharon Burks, Catherine Copetas, Colleen Everett, Charlotte Yano, and Denny Marous for their administrative support throughout these years. I thank Dale James for providing editing suggestions on this thesis. The thesis is prepared using a LaTeX style file from Håkan L. S. Younes.

Thank you to everyone, both those who are mentioned here and those who are not.

Last, but not least, I thank my parents. This thesis is dedicated to them.

Jia-Yu Pan

May 12, 2006

# Contents

# Chapter 1

# Motivation

How do we automatically find patterns and mine data in large multimedia databases, to make such databases useful and accessible? How do we find uni-modal patterns in different data modalities? How do we find correlations that involve multiple modalities? For example, how do we find patterns such as "*news videos have static scenes and speech-like sounds*," or "*the blue region at the upper part of an image of natural scenes is likely to be the 'sky'*?"

Advances in digital technologies make possible the generation and storage of large amounts of multimedia objects such as images and video clips. Multimedia content contains rich information in various modalities such as image, audio, video frames, time series, etc. Advanced tools that can find characteristic patterns and correlations in multimedia content are required for the effective exploitation of multimedia databases.

In a video database [127] that contains thousands of hours of video clips, users require functionalities such as content-based search or summarization to efficiently obtain the information they need. Finding characteristic patterns in various data modalities of video (image, audio, transcript text, etc.) is essential to support these content-based functionalities. For example, to support efficient retrieval, continuous video clips need to be partitioned into segments of coherent content (e.g., individual news stories); for identifying relevant information, video segments need to be classified (sports or entertainment, for example) according to their content. On the other hand, patterns

('sea', 'sun', 'sky', 'waves') ('cat', 'forest', 'grass', 'tiger') no caption

(a) Captioned image $I_1$ (b) Captioned image $I_2$ (c) Image $I_3$

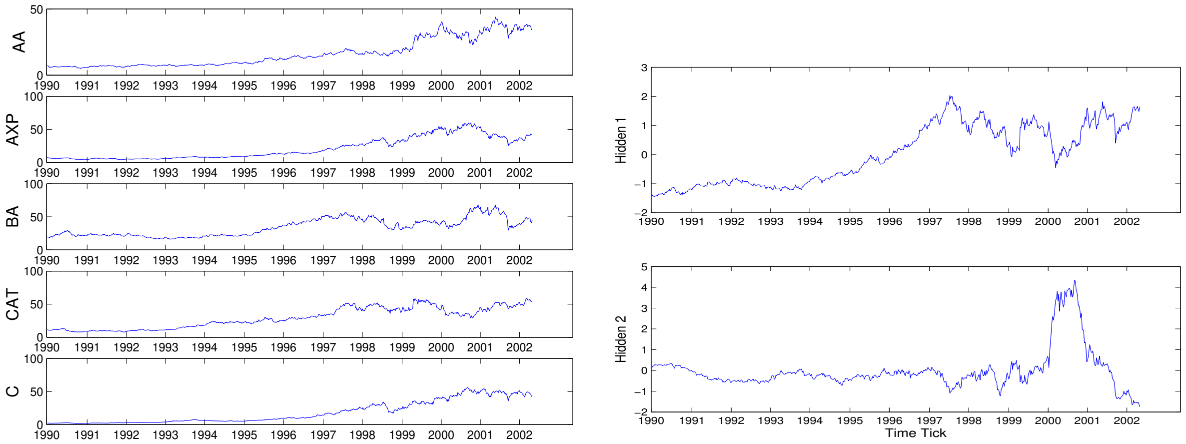
**Figure 1.1:** Captioned images and the cross-modal pattern discovery. Given two captioned images, (a) and (b), each is captioned with terms describing the content. Discovering cross-modal patterns like the correlations between image content and terms might help to automatically caption a new image (c). (Figures look best in color.)

that combine information from multiple data modalities also offer useful clues to understanding the video content. For example, the background noise of traffic in a scene of a two-man conversation suggests that the conversation takes place on the street.

**Thesis Statement:** To make multimedia data accessible and useful, it is required to have advanced data mining tools to find patterns that are both meaningful and can support data mining tasks like segmentation, classification, and summarization.

Interesting patterns exist in multimedia data sets other than video databases. For example, in a collection of captioned images, where each image is annotated with words that describe it, it is desirable to discover the correlations between the image content and the words. Such image-word correlations would be helpful for annotating a new image without human intervention (Figure 1.1). On the other hand, in a set of time sequences, such as the audio in video clips or stock prices, patterns that summarize the data characteristics could provide informative insights (Figure 1.2).

Finding patterns in multimedia data is a challenging problem. Unlike text documents, which are composed of tokens (words) from a vocabulary that bear semantic meanings, and can serve as natural units for content-based analysis, multimedia data like images or audio does not have such




(a) Example sequences (b) Patterns discovered

**Figure 1.2:** Pattern discovery in (co-evolving) stock price sequences. (a) Example sequences of stock prices (X-axis: year, Y-axis: stock price). (b) Patterns discovered from the sequences (top: general trend, bottom: Internet bubble). Company stock symbols: AA: Alcoa, AXP: American Express, BA: Boeing, CAT: Caterpillar, C: CitiGroup.

tokens. For example, "*what is the 'vocabulary' that we can use to describe image content?*" On the other hand, multimedia objects contain information richer than text, and have information from more than one modality: "*how do we find correlation patterns across different modalities, like the audio and frames of a video clip?*

Finding patterns that are meaningful and understandable could benefit many data mining applications like segmentation, classification, and rule discovery. The goal of segmentation, such as image segmentation, is to separate multimedia data (image) into parts (regions), such that each has a homogeneous characteristic [99]. In classification, we want to differentiate objects' different types, such as classification of video clips by their genres (e.g., news or commercial) [98]. Sometimes classification and segmentation are achieved at the same time [3]: if we have the labels of data from classification, we can easily identify the segmentation boundaries which separate data with different labels. Automated systems to discover rules and mine data is crucial to make the most out of the plentiful raw data in today's massive databases [10].

In this thesis, we focus on finding patterns in multimedia data and data mining. Patterns in multimedia data could be *uni-modal* – the characteristics of one particular data modality, or *cross-modal* – the correlations between attributes of different modalities. Our goal is to design data mining tools which extract patterns that not only are meaningful by themselves, but also could improve performances on multimedia applications. The general questions that we are interested in are:

- How do we find uni-modal patterns and cross-modal correlations?
- How do we interpret the patterns discovered and do data mining?

## 1.1 Contributions

The main contributions of this thesis are the studies of the two settings that we proposed for pattern discovery: *uni-modal pattern discovery* and *cross-modal pattern discovery*.

For the first problem, we proposed an algorithm, *AutoSplit*, which can find patterns in a variety of modalities such as video, audio, images, text, and time sequences. AutoSplit uses a method, *independent component analysis (ICA)*, which is better than more popular methods such as *principal component analysis (PCA)* on finding patterns in real-world data sets. In addition, AutoSplit provides a general procedure to do data mining on the results of ICA. As we will show later, AutoSplit finds patterns that are meaningful and consistent with human intuition. The patterns found are also useful in applications such as classification or outlier detection.

For the problem of cross-modal pattern discovery, we proposed a graph-based method, *MAGIC*, for finding correlations across different modalities. MAGIC turns the multimedia correlation discovery problem into a graph problem, by representing multimedia data as a graph. Using the technique "random walk with restarts" on the graph, MAGIC is able to find correlations among all modalities in a multimedia data set. These cross-modal correlations found by MAGIC are helpful in several multimedia applications such as automatic image captioning and news event summarization.

**Thesis Layout** The thesis is organized into two main parts: uni-modal pattern discovery and cross-modal pattern discovery.

In part I – uni-modal pattern discovery, we introduce our proposed method *AutoSplit*, and discuss our experimental results of AutoSplit on finding uni-modal patterns in a variety of data modalities. We start with an overview of related work (Chapter 2). Particularly, our AutoSplit method is an unsupervised method which automatically finds patterns that summarize the characteristics of a given data set. Chapter 3 describes the details of AutoSplit, with a brief review on the concept of independent component analysis, which is a major component of AutoSplit.

In Chapters 4 and 5, we discuss our experimental results of using AutoSplit on various data modalities. In Chapter 4, we apply AutoSplit on video clips. AutoSplit finds spatial-temporal patterns in video frames (*VideoBasis*), and characteristic auditory signals in audio tracks (*AudioBasis*). The patterns are consistent with our daily experiences with broadcast news video. On the task of video genre classification (news versus commercial), representing video clips using these patterns (VideoBasis/AudioBasis) achieves a classification accuracy that is greater than 81%. In Chapter 5, AutoSplit is used to find hidden patterns on co-evolving time series, such as human motion capture data or stock prices. AutoSplit can distinguish the characteristics during different time periods, and can also find the hidden variables of time sequences (e.g., the "general trend" or the "Internet bubble" event in stock price sequences).

In Chapter 6, we propose a system called *ViVo* for mining biomedical images. Using an idea from AutoSplit, the ViVo system can automatically construct a visual vocabulary for describing a set of images. The vocabulary tokens, which we called *ViVos*, are able to classify images of a variety of biological conditions with more than 82% accuracy. Surprisingly, the ViVos are biologically meaningful and have many data mining applications, including identifying interesting regions of an image, or highlighting image regions that distinguish images of two medical conditions.

In part II – cross-modal pattern discovery, we introduce our proposed method *MAGIC* and discuss our experimental results in finding patterns involving multiple data modalities. Cross-modal patterns are useful in a variety of applications, ranging from multi-modal summarization to automatic image captioning. In Chapter 7, we start with the motivation of cross-modal pattern discovery and describe some related work in this area.

In Chapter 8, we present the details of our proposed graph-based framework, MAGIC, for cross-modal pattern discovery. MAGIC provides an intuitive graph-based framework for incorporating information from all kinds of data modalities. MAGIC's graph-based framework also creates opportunities for applying graph algorithms to the multimedia domain. In particular, MAGIC uses the *random walk with restarts (RWR)* technique and finds *correlations* between all modalities in a given data set.

In Chapter 9, we apply MAGIC to do automatic image captioning. By finding robust correlations between text and image, MAGIC achieves a relative improvement of 58% in captioning accuracy as compared to recent machine learning techniques [101]. In Chapter 10, we present another application of MAGIC on news event summarization. In a collection of broadcast news videos, MAGIC can find correlations between news events, video shots, and transcript words. Based on these correlations, we can put together shots (keyframes) and transcript words that are correlated with a news event, and create a multi-modal summary for the event.

# Part I

# Uni-Modal Pattern Discovery

# Chapter 2

# Related Work

Pattern discovery in multimedia data has been an ongoing research effort, and researchers have borrowed ideas from diverse areas, ranging from signal processing, computer vision, and databases, to address this problem.

In this chapter, we discuss previous work on pattern discovery from uni-modal data, i.e., *uni-modal pattern discovery*. Since this is still an on-going research topic, our survey does not attempt to be comprehensive. Instead, we will highlight several general approaches to this problem, and describe where our approach resides in the big picture.

A research topic that is closely related to uni-modal pattern discovery is *feature extraction* in the area of content-based retrieval. Uni-modal pattern discovery has borrowed ideas from works in feature extraction, but with an additional emphasis on the interpretability of the extracted patterns. For data mining and knowledge/rule summarization, we want to find patterns that are novel, meaningful, and understandable.

## 2.1 Uni-Modal Pattern Discovery in Multimedia Data

We categorize previous approaches in two levels, using two criteria: Is a method *data-independent/manual* or *adaptive*? And for an adaptive method, is it a *supervised* or *unsupervised* method? (Figure 2.1).

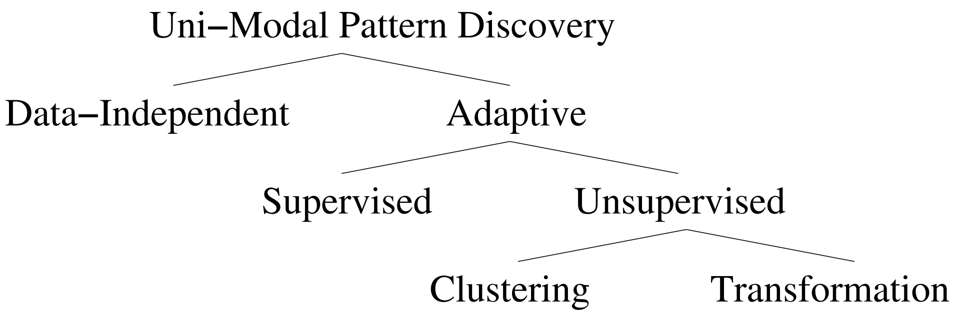



**Figure 2.1:** The categories of approaches for uni-modal pattern discovery. Our proposed method, AutoSplit, is in the category: “Adaptive/Unsupervised/Transformation”.

One major difference between these different approaches is the amount of external information available to the pattern discovery process. For the *data-independent/manual* approach, the domain knowledge about the characteristics of a data set is used to (manually) design methods for extracting patterns. On the other hand, when less knowledge about the domain is available, we need *adaptive* methods that can adapt to a data set and find patterns.

For the adaptive methods, when we have partial information about the patterns we want to discover, we can design *supervised* methods which use this information to guide the pattern discovery. For example, if we have the class labels on the data objects, we can use this information to find patterns that can distinguish different object classes. For situations where no specific information about a data set is available, we would use *unsupervised* methods.

Examples of data-independent/manual methods include techniques like the Fourier transform from general signal processing theory [96], or the manually designed statistics such as the Tamura texture features from studies on the human visual perception [110]. Adaptive-supervised methods for pattern discovery include methods such as the Fisher Discriminant Analysis [116] and classifiers such as decision trees [87]. As for the adaptive-unsupervised methods, popular methods includes clustering [87] and transformation methods such as the principal component analysis [55]. Our proposed method, AutoSplit (Chapter 5), belongs to the category “adaptive-unsupervised”.

**Pattern Discovery and Feature Extraction** A research topic that is closely related to uni-modal pattern discovery is *feature extraction* in the area of content-based retrieval. For many multimedia applications, it is essential to discover features/patterns that capture the content of an image or a video clip [110].

However, most of the work on feature extraction has focused on the performance of the target application, whether it be retrieval or classification. The meanings of extracted features are not a major concern, and the extracted features are often hard to understand or even are not interpretable at all.

On the other hand, for data mining and knowledge/rule summarization, we want to find patterns that are meaningful and understandable. Pattern discovery has borrowed ideas from works in feature extraction, but with an additional emphasis on the interpretability of the extracted patterns. Based on these human-interpretable patterns, we can further design data mining tools and extract rules from the databases.

## 2.2 Data-Independent Pattern Discovery

A pattern discovery method is *data-independent/manual* if the method focuses on extracting the same set of patterns for every data set, and does not adapt to the properties of an individual data set. Usually, the patterns are manually designed based on knowledge on the data domain. In this section, we discuss two categorizes of methods of this approach: methods that extract manually defined features, and those that are based on techniques from general signal processing theory.

**Manually Defined Features** For finding patterns in multimedia data such as images or video clips, a natural direction is to observe how people perceive and describe a multimedia object, and design features that match human perception. One advantage of this approach is that it directly addresses the needs of human users of multimedia systems.

There have been many manually defined features proposed for images and audio signals. These features are usually derived from psychological or human-computer-interface studies in human perception [88]. For images, most of these features are based on the concepts of color, texture and shape [110].

Color histogram [120, 29] is the most popular color feature, with advanced variants such as color moments and color sets [110] that improve the characterizations of the color information. Other popular color features include the color correlogram [45] and the color structure [83], which combine the color information with the layout information to provide better descriptions of the image content.

Texture, such as coarseness, contrast, or directionality, is another visual cue people use to describe images. Texture features such as the Haralick moments and the Tamura texture representation [110] have been proposed to quantify human perception on different texture patterns, based on results from user studies.

Shape information is also used in human perception. In situations where color and texture information are not available (e.g., in a dark room), sometimes we still recognize objects based only on shape information. Many shape features have been proposed in the literature [136, 110], and the most popular ones are features based on the region covered by the shape, such as the area of the convex hull, the eccentricity, the geometric moments, or Zernike moments [136].

Several novel, domain-specific features have been proposed while exploring new problems and applications [88, 93]. For example, in [93], features based on geometry and physical laws are designed to distinguish the subtle differences between photographic images and photorealistic computer graphic images. In [88], knowledge about human perception of color patterns is used to design new features for image retrieval.

For audio applications, typical auditory features are designed to capture concepts such as rhythm, timbre, pitch, and beat. These features are usually derived according to results from human psychological studies, based on statistics collected by signal analysis techniques. For example, the Mel-frequency cepstral coefficients (MFCC) are a specific wrapping of auditory frequency that adjust the spectral resolution to human ear. More details on audio features can be found in [123].

**Methods from Signal Processing** There are methods for uni-modal pattern discovery that are also based on concepts from signal processing. Among them, the most popular ones are the transformation-based methods such as the Fourier transform, the discrete cosine transform, and the wavelet transform. Each of these transformation-based methods uses a specific set of *basis vectors* to represent the data, where each basis vector captures some properties (patterns) of the data. For example, the Fourier basis vectors correspond to global periodic (cosine) patterns, while the wavelet basis vectors capture localized patterns at different scales. These transformations have been shown useful in many applications like retrieval and compression. Recently, Gabor transformation [67] has become popular due to its effectiveness in representing image content. Interestingly, Gabor transformation has been shown to resemble the functionality of simple cells in the human visual cortex [67].

Recent works go beyond the complete set of transformation basis functions and study feature extraction from an *over-complete* set of basis functions (e.g, wavelet packet tree [117, 40]). In an over-complete set, the available basis functions are fixed but redundant, providing more flexibility for feature extraction, and a better chance to capture specific characteristics of a data set.

## 2.3 Adaptive Uni-Modal Pattern Discovery

Adaptive methods for pattern discovery are useful when we do not have much knowledge about a data set. The advantage of these methods is that they can adapt to the characteristics of a data set, and have the potential to find specific patterns that can represent the data set more efficiently. Due to the infinite number of possible patterns, methods for adaptive pattern discovery usually depend on some guiding criteria to select good patterns. The quality of the patterns depends on the guiding criterion, i.e., how good the criterion matches the data characteristics.

We divide the adaptive pattern discovery methods as *supervised* methods and *unsupervised* ones, depending on whether extra information about the data objects is used. A method is *supervised* if it has access to external information on individual data objects (such as the class labels).

In contrast, a *unsupervised* pattern discovery method has no information on individual data objects, and the pattern discovery process is guided by some data-independent principles, such as *the minimization of the representation error* or *the maximization of independence*, etc.

### 2.3.1 Supervised Adaptive Methods

When the goal of pattern discovery is to support applications like classification, it would be beneficial to take into consideration the external information such as class labels when finding patterns. In this subsection, we discuss three approaches on supervised pattern discovery, namely, *rule deduction from classifiers*, *supervised clustering*, and *discriminant analysis methods*.

**Rule Deduction from Classifiers** One way to discover patterns using class labels is to construct a classifier and extract rules from the classifier. The classifier implicitly summarizes the differences between data classes, and the goal here is to explicitly interpret the distinguishing patterns (or rules) found by the classifier.

Extracting rules from a classifier is not always easy. Previous work has proposed methods to extract rules from classifiers such as decision tree classifiers [87] and linear support vector machines [34]. A rule extracted usually contains an *antecedent*, which is a logical expression of data attributes, and a *consequent*, which is the predicted label for objects that satisfy the antecedent. Each extracted rule specifies the relation between data attribute values (specified in the antecedent) and the class label (as in the consequent).

**Supervised Clustering** Clustering is an another way to find adaptive patterns in a data set. Clustering algorithms find the data clusters which summarize the patterns of different object types in the data set, and are useful in data mining.

Usually, a clustering method is *unsupervised* and requires only some "hints" (e.g., similarity function and goodness criterion) to form clusters. However, when additional information is available about whether specific objects should be in the same cluster or not, we would like the clustering method to take advantage of this information. Methods that incorporate these additional information in clustering are called *supervised clustering* methods.

*Supervised clustering* methods take advantage of additional information to obtain better clustering quality. The additional information can be class labels on selected data points [133], or "must-link/cannot-link" *constraints* on the data points [18]. Several strategies have been used to incorporate the additional information into clustering, for example, by modifying distance function [133], by minimizing the violations on the given constraints [18], or by preserving the "information content" from the class labels [43].

**Discriminant Analysis Methods** Given a data set with class labels on the data objects, discriminant analysis methods perform transformations on data attributes to find the best hidden variables (features) to distinguish data classes. By examining the relationship between the discovered features and the original data attributes, we can also discover rules about differences between the data classes.

Statistical methods for discriminant analysis, such as Fisher/linear Discriminant Analysis (LDA) [24, 116], are also commonly used to find features for distinguishing data classes. Traditional LDA assumes each data class forms a single Gaussian data cluster, and computes data-distinguishing features as linear weighted sums of original data attributes. Several extensions of LDA have been proposed, for example, Multiple Discriminant Analysis (MDA) [24] for distinguishing more than 2 classes, Multi-modal oriented discriminant analysis (MODA) [62] for the single-class-multiple-cluster settings, etc.

Using additional information about data classes, supervised methods may be able to discover useful data patterns. However, obtaining such additional information is not easy. For example, preparing the class labels usually requires human involvement, and is labor-intensive and subject to the personal bias of human annotators. When additional information is not available, *unsupervised methods* for adaptive pattern discovery, which we explain next, would be a helpful option.

### 2.3.2 Unsupervised Adaptive Methods

There are two major categories of unsupervised approaches for adaptive pattern discovery: clustering-based methods and transformation-based methods. Traditional unsupervised methods such as K-means [87] and PCA [55] assume that the data has a Gaussian characteristic, which is not always true for real world data sets. Recently, there have been methods proposed for finding non-Gaussian data patterns [12, 48].

Clustering methods find groups of data objects where objects in the same group are similar and objects in different groups are not similar. Each cluster represents a pattern that is present in the data set. Clustering methods have been used to summarize low-level statistics computed from multimedia objects to produce better features for data representation. For example, K-means clustering has been used to cluster similar image regions ("blobs") together for object recognition and image annotation [25, 100]. In [130], Gaussian mixture modeling is used to generate components to describe image content for image retrieval.

A transformation-based method for adaptive pattern discovery tries to find good hidden variables to describe the data objects in a given set. An example of this type of method is the singular value decomposition (SVD). SVD finds hidden variables that give the best low-dimensional representation of the data set, a representation that has the least distortion in terms of the $L_2$ distance. SVD has been widely used for finding hidden variables in multiple settings: for text retrieval [20], under the name of Latent Semantic Indexing (LSI); for face matching in the eigenface project [122]; for pattern analysis under the name of Karhunen-Loeve transform [24] and principal component analysis (PCA) [55]. Recently, methods have been proposed to make the idea of SVD/PCA more robust, by taking into account the outliers in data samples [61].

Techniques used in previous work, such as K-means or SVD, usually have an assumption that the data has a Gaussian characteristic. Unfortunately, the Gaussian assumption does not hold for many real world data sets. To discover non-Gaussian patterns, one recent development in clustering methods is *correlation clustering* [12], where new methods are designed to find non-Gaussian clusters that display (linear) correlations.

In this thesis, we proposed an unsupervised, transformation-based method (AutoSplit, Chapter 5) for uni-modal pattern discovery, based on the concept of independent component analysis (ICA) [48, 46, 47]. ICA is an unsupervised, adaptive method that finds hidden variables that are (statistically) independent from each other. ICA does not assume data distribution to be Gaussian, and is able to extract patterns better than PCA in many cases [99, 37].

# Chapter 3

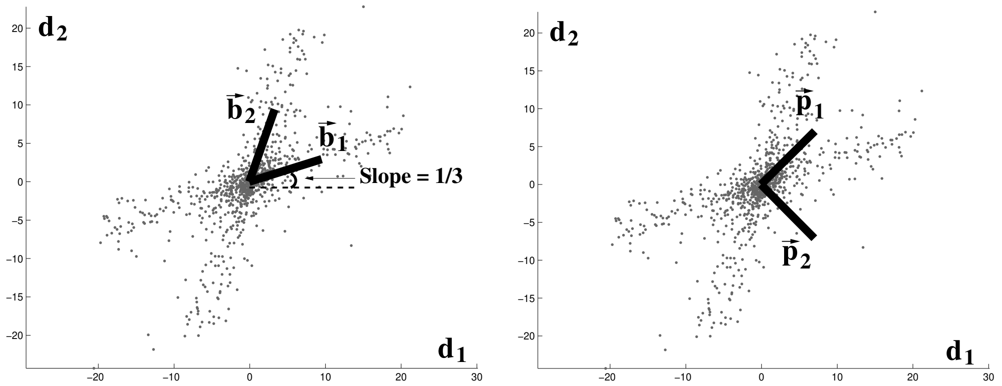
# Proposed Method: AutoSplit

In this chapter, we present our proposed algorithm, AutoSplit, for mining uni-modal patterns in multimedia data sets. AutoSplit uses *independent component analysis (ICA)* as a tool, utilizing its ability to find non-redundant patterns in real-world data sets. AutoSplit provides a step-by-step guide to find and interpret uni-modal patterns, and is applicable to various data modalities (video, audio, text, time sequences, etc.).

In the rest of this chapter, we first introduce the background and definitions of ICA, including a brief discussion on the relationship between ICA and human perceptual processing. Then, we describe in detail our proposed steps for mining multimedia data using AutoSplit: how to compute the patterns, and how to mine (interpret) the meanings of the patterns.

## 3.1 Background: Independent Component Analysis

Independent component analysis (ICA) [46, 47], like *principal component analysis (PCA)*, is a method for finding structures to describe the distribution of a set of data points. Both ICA and PCA represent a multivariate data set by a linear coordinate system (Figure 3.1). Unlike PCA, which gives orthogonal coordinates (or basis vectors) to model the distribution of data points, ICA is more generalized and can give non-orthogonal basis vectors.



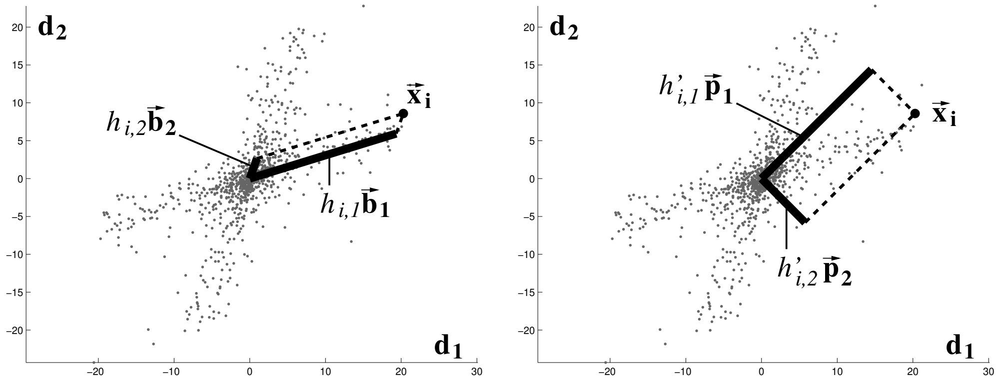

(a) ICA basis vectors (b) PCA basis vectors

**Figure 3.1:** Basis vectors by ICA and PCA on the "X-shape" data set. ICA captures the real components of the data set, while PCA fails. (a) Basis vector $\vec{\mathbf{b}}_1$ has slope 1/3, indicating a pattern in the data set: a 3-to-1 ratio relationship between the two data attributes $\mathbf{d_1}$ (X-axis) and $\mathbf{d_2}$ (Y-axis). The basis vectors ($\vec{\mathbf{b}}_1$, $\vec{\mathbf{b}}_2$, $\vec{\mathbf{p}}_1$, $\vec{\mathbf{p}}_2$) have length 1, but are shown with length 10 for better visualization.

Figure 3.1 demonstrates the difference between ICA and PCA. The PCA basis vectors capture the major variations in data distribution by examining the second-order statistics (covariance) of the data, and give the best coordinate axes for dimensionality reduction (optimizing the $L_2$-norm error caused by the dimensional reduction). However, the overall variation patterns do not always reveal the correct structure of a data set, especially when the data is not distributed according to a Gaussian distribution. For a data set that is non-Gaussian, like the example in Figure 3.1(b) shows, PCA misses the correct distribution patterns. On the other hand, ICA is able to find the non-orthogonal structure of the data set which PCA misses, as we show in Figure 3.1(a). In a near-Gaussian data set, ICA is as good as PCA, finding almost the same patterns as PCA.

### 3.1.1 ICA Definitions: Basis Vectors and Hidden Variables

A data point with $m$ attributes can be considered as a point ($\vec{\mathbf{x}}_i$) in the $m$-dimensional attribute space. ICA assumes a data model where each data point $\vec{\mathbf{x}}_i$ is represented as a linear combination




(a) ICA representation of $\vec{\mathbf{x}}_i$
$\vec{\mathbf{x}}_i$=$h_{i,1}\vec{\mathbf{b}}_1$+$h_{i,2}\vec{\mathbf{b}}_2$

(b) PCA representation of $\vec{\mathbf{x}}_i$
$\vec{\mathbf{x}}_i$=$h'_{i,2}\vec{\mathbf{p}}_1$+$h'_{i,2}\vec{\mathbf{p}}_2$

**Figure 3.2:** Representation of a data point $\vec{\mathbf{x}}_i$ using (a) ICA and (b) PCA. (a) $\vec{\mathbf{b}}_1$, $\vec{\mathbf{b}}_2$: ICA basis vectors; $h_{i,1}$, $h_{i,2}$: ICA hidden variables. (a) $\vec{\mathbf{p}}_1$, $\vec{\mathbf{p}}_2$: PCA basis vectors; $h'_{i,1}$, $h'_{i,2}$: PCA hidden variables. The hidden variables control how the basis vectors are combined to represent a data point. ICA hidden variables are sparse (i.e., most have near-zero values, e.g. $h_{i,2}$) and are more efficient in representing data points. In (a), ICA still give a good representation of $\vec{\mathbf{x}}_i$, even if we ignore the term $h_{i,2}\vec{\mathbf{b}}_2$.

of the $m$ unknown basis vectors $\vec{\mathbf{b}}_k$, $k$=$1,\ldots,m$. In other words,

$$\vec{\mathbf{x}}_i = \sum_{k=1}^{m} h_{i,k}\vec{\mathbf{b}}_k, \tag{3.1}$$

where $h_{i,k}$ is the corresponding coordinate in the direction of vector $\vec{\mathbf{b}}_k$, for $k = 1,\ldots,m$.

The $m$ ICA basis vectors define a new space (a coordinate system) for describing data points. We will call the new space the *ICA basis vector space*, to differentiate it from the original data attribute space. The coordinates $h_{i,k}$ ($k$=$1,\ldots,m$) determine the location of a point $\vec{\mathbf{x}}_i$ in the ICA basis vector space. These coordinates $h_{i,k}$'s are called the *hidden variables* of the data point $\vec{\mathbf{x}}_i$, and are different from the original *observed variables* (data attributes) of $\vec{\mathbf{x}}_i$.

The idea of representing a data point $\vec{\mathbf{x}}_i$ using ICA basis vectors and the corresponding hidden variables are illustrated in Figure 3.2(a). Instead of describing the point $\vec{\mathbf{x}}_i$ using the data attributes (coordinates $\mathbf{d_1}$ and $\mathbf{d_2}$), the ICA basis vectors provides an alternative representation for $\vec{\mathbf{x}}_i$, where $\vec{\mathbf{x}}_i$=$h_{i,1}\vec{\mathbf{b}}_1$+$h_{i,2}\vec{\mathbf{b}}_2$, and $h_{i,1}$ and $h_{i,1}$ are the ICA hidden variables corresponding to the two ICA basis vectors $\vec{\mathbf{b}}_1$ and $\vec{\mathbf{b}}_2$, respectively. We call $[h_{i,1}, h_{i,2}]$ the *ICA hidden variable representation* (or in short, *ICA representation*) of the data point $\vec{\mathbf{x}}_i$.

For a point in $m$-dimensional space, it is known that we need $m$ basis vectors to completely specify the point. Therefore, in a data set with $m$ attributes, we could extract $m$ ICA basis vectors, and at the same time, the $m$ corresponding hidden variables.

Each ICA basis vector captures a pattern in the data set, as we showed in Figure 3.1(a). To understand the meaning of the pattern captured by a basis vector, one way is to examine the slope of the vector. Since the basis vectors are also in the $m$-dimensional data attribute space, each basis vector has $m$ elements. The slope of a basis vector indicates a pattern of "ratio correlation" between the $m$ data attributes. For example, in Figure 3.1(a), the ($m$=2)-dimensional basis vector $\vec{\mathbf{b}}_1$ has slope 1/3, which corresponds to a pattern in the data – a 3-to-1 ratio relationship between the two data attributes $\vec{\mathbf{d}}_1$ and $\vec{\mathbf{d}}_2$. If we combine our knowledge about the physical meanings of the data attributes with the ratio correlations indicated by a basis vector, we could gain insights about the pattern specified by the basis vector.

**Mathematical Details** To represent the idea of ICA using a matrix formulation, given a collection of $n$ data points with $m$ attributes, we can represent the data points as a $n$-by-$m$ data matrix $\mathbf{X}_{[\mathbf{nxm}]}$, where the $i$-th row $\vec{\mathbf{x}}_i$ represents the $i$-th data point. ICA decomposes the data matrix $\mathbf{X}$ into two matrices, $\mathbf{H}_{[\mathbf{nxm}]}$ and $\mathbf{B}_{[\mathbf{mxm}]}$ that satisfy two properties: (a) $\mathbf{X}$=$\mathbf{HB}$ and (b) the columns of $\mathbf{H}$ are as "independent" as possible.

The matrix $\mathbf{B}_{[\mathbf{mxm}]}$ is called the *basis matrix*, whose rows $\vec{\mathbf{b}}_j$ are the basis vectors that describe the distribution structure of the data set $\mathbf{X}$. The matrix $\mathbf{H}_{[\mathbf{nxm}]}$ is called the *hidden matrix*, whose $j$-th column $\vec{\mathbf{h}}_j^T$ is consisted of the values of the $j$-th hidden variable for every data point (i.e., the point's coordinate on the basis vector $\vec{\mathbf{b}}_j$). Equations (3.2) to (3.4) show the relationship between the data points $\vec{\mathbf{x}}_i$, the hidden variables $\vec{\mathbf{h}}_j$, and the basis vectors $\vec{\mathbf{b}}_j$ (for $i = 1,\ldots,n$ and $j = 1,\ldots,m$).

$$\mathbf{X}_{[\mathbf{nxm}]} = \mathbf{H}_{[\mathbf{nxm}]}\mathbf{B}_{[\mathbf{mxm}]}, \tag{3.2}$$

$$\Rightarrow \begin{bmatrix} -\vec{\mathbf{x}}_1- \\ -\vec{\mathbf{x}}_2- \\ \vdots \\ -\vec{\mathbf{x}}_n- \end{bmatrix} = \begin{bmatrix} | & | & \cdots & | \\ \vec{\mathbf{h}}_1^T & \vec{\mathbf{h}}_2^T & \cdots & \vec{\mathbf{h}}_m^T \\ | & | & \cdots & | \end{bmatrix} \begin{bmatrix} -\vec{\mathbf{b}}_1- \\ -\vec{\mathbf{b}}_2- \\ \vdots \\ -\vec{\mathbf{b}}_m- \end{bmatrix} \tag{3.3}$$

$$= \begin{bmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,m} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,m} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ h_{n,1} & h_{n,2} & \ldots & h_{n,m} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} & \ldots & b_{1,m} \\ b_{2,1} & b_{2,2} & \ldots & b_{2,m} \\ \vdots & \vdots & & \vdots \\ b_{m,1} & b_{m,2} & \ldots & b_{m,m} \end{bmatrix}. \tag{3.4}$$

To simplify our presentation, in the following we will use the vector notation $\vec{\mathbf{h}}_j$ to represent both the $j$-th column of the hidden matrix **H**, as well as the symbol of the the hidden variable itself. The values in the column $\vec{\mathbf{h}}_j{}^T$ are considered as values sampled from the $j$-th hidden variable.

Finding the "best" basis vectors from a given set of data points $\vec{\mathbf{x}}_i$ ($i$=$1,\ldots,n$) is not easy. The challenge is that both basis vectors and hidden variables are unknown; only the data points are given. To decompose the matrix **X** into matrices **H** and **B** (**X=HB**, equation 3.2), extra assumptions are needed to guide the decomposition and establish a computable procedure. The assumption that ICA makes is that it will find hidden variables (columns of **H**) that are *as independent as possible*, where "independent" means "statistically independent". Surprisingly, this independence assumption also gives us the basis vectors that correctly capture patterns in a data set.

Mathematically, if two hidden variables $\vec{\mathbf{h}}_1$ and $\vec{\mathbf{h}}_2$ are random variables that are statistically independent, the joint density p($\vec{\mathbf{h}}_1$,$\vec{\mathbf{h}}_2$) is the product of the marginal density functions p($\vec{\mathbf{h}}_1$) and p($\vec{\mathbf{h}}_2$):

$$p(\vec{\mathbf{h}}_1, \vec{\mathbf{h}}_1) = p(\vec{\mathbf{h}}_1)p(\vec{\mathbf{h}}_2).$$

The intuition of ICA is that independent variables do not share redundant information (i.e., mutual information is zero) and minimize the redundancy (thus, maximize the efficiency) when representing the data points. At the same time, it turns out that the basis vectors corresponding to the most efficient hidden variables are the ones that point along the major structures of the data distribution, which are exactly what we want (Figure 3.1(a)).

In practice, the solution to matrices **H** and **B**, where the hidden variables $\vec{\mathbf{h}}_i$'s (columns of **H**) are as independence as possible, is not unique. The solutions are subjected to a scaling factor between **H** and **B** (i.e., $\mathbf{HB}=(\mathbf{H}\alpha)(\frac{1}{\alpha}\mathbf{B})$, for every scaling factor $\alpha \neq 0$). A common strategy is to scale the matrices so that each basis vector in matrix **B** (rows of **B**) has length 1. In our experiments, we always let ICA basis vectors (rows of matrix **B**) have length 1.

### 3.1.2 Basis Vectors as Vocabulary — Sparse Coding

The ICA basis vectors can capture the major patterns of a data set, as shown in Figure 3.1(a), where the two ICA basis vectors lie exactly on the two patterns of the data set. Therefore, the ICA basis vectors could be considered as a good vocabulary to efficiently describe the data set.

For example, using the basis vectors $\vec{\mathbf{b}}_1$ and $\vec{\mathbf{b}}_2$ as "vocabulary", we could describe the point $\vec{\mathbf{x}}_i$ shown in Figure 3.1 by its hidden variable representation, $[h_{i,1}, h_{i,2}]$. In fact, since the point $\vec{\mathbf{x}}_i$ is located on the pattern aligned with $\vec{\mathbf{b}}_1$ (Figure 3.2(a)), we could get a pretty good description of $\vec{\mathbf{x}}_i$ by using only $h_{i,1}$: that is, $\vec{\mathbf{x}}_i \approx h_{i,1}\vec{\mathbf{b}}_i$. In general, the vocabulary made of ICA basis vectors is efficient in that most points in Figure 3.2(a) can be efficiently represented using only one ICA basis vector (either $\vec{\mathbf{b}}_1$ or $\vec{\mathbf{b}}_2$) in the vocabulary.

The set of ICA basis vectors is a good vocabulary that can represent data points efficiently. Each data point is located on one of the ICA basis vectors, and the ICA hidden variable representation will have most values near zero, except the one that corresponds to the basis vector on which the data point lies. The near-zero values can be efficiently coded, or even be discarded without losing much accuracy in describing the location of a point. We call such a hidden variable representation with many near-zero values a *sparse* representation.

On the other hand, in Figure 3.2(b), both PCA hidden variables $h'_{i,1}$ and $h'_{i,2}$ are significant non-zero values; that is, the PCA representation of $\vec{\mathbf{x}}_i$ is not sparse. We would need both $h'_{i,1}$ and $h'_{i,2}$ to represent $\vec{\mathbf{x}}_i$ – ignoring either of them will give a bad representation of $\vec{\mathbf{x}}_i$.

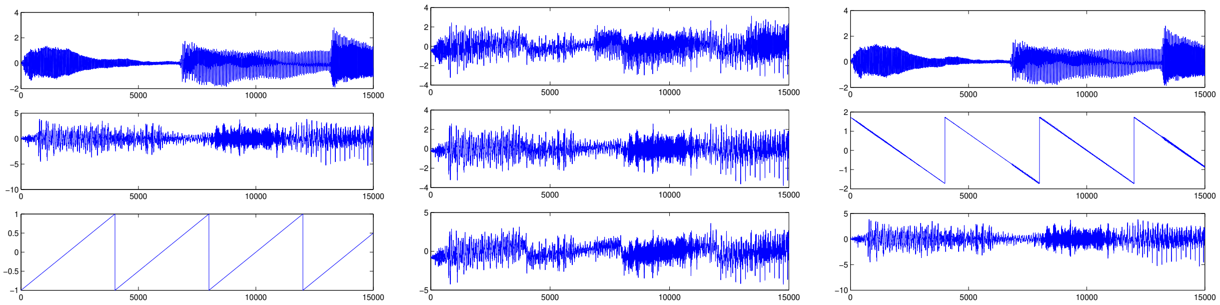
The sparse representation of ICA is efficient for describing a set of data points, and it achieves good compression of the data set. In fact, it also implies that the corresponding ICA basis vectors are descriptive in that they capture data characteristics very well (this is the reason that representing data points using these vectors is so efficient).

The above discussions establish the relation between "having a sparse hidden variable representation" and "having basis vectors that precisely capture the data patterns." In fact, computationally, the analysis of ICA focuses on finding hidden variables that are independent. The independent hidden variables are less redundant, and therefore more efficient and sparse. The excellent descriptive power of the ICA basis vectors is actually a fortunate consequence of finding independent hidden variables. ICA connects the notions of "independence", "efficient encoding", "sparse representation", and "meaningful basis vectors", making it a promising tool for pattern discovery and data mining.

### 3.1.3 Hidden Variables — Source Separation

In the previous discussions, we have seen that ICA basis vectors can capture the structure of the data distribution. The ICA analysis also gives us ICA hidden variables which are mutually independent. In the following discussion, we will show that these independent hidden variables are also useful in pattern discovery. In particular, the independent hidden variables could reveal the unknown sources that contribute to the observed data. We will illustrate the use of ICA hidden variables in pattern discovery using an application called *the "cocktail party" problem.*

In a crowded party with various simultaneous conversations mixed up with each other, how does our brain separate each conversation for further processing? This is known as the "cocktail party" problem. The goal is to separate the signal sources (e.g., conversations), with no knowledge about the property of each source (e.g., conversation content) and how they are mixed (e.g., the room acoustics). Since no particular information about the unknown sources is assumed, the



(a) Sources (b) Observed (c) Recovered

**Figure 3.3:** Discovering hidden variables/sources using ICA. Each signal is plotted as value (Y-axis) versus time tick (X-axis). (a) 3 hidden sources: (top) a classical music, (middle) a human speech, (bottom) a synthetic signal. (b) 3 observed signals: each observed signal has a different mix of the hidden sources. (c) The recovered hidden variables by ICA resemble the true hidden sources (up to a sign-flip).

problem is also called the *blind source separation* problem.

As an illustration of the cocktail party problem, suppose that we have three simultaneous conversations/sources like the three time series in Figure 3.3(a). Suppose that we also have three microphones which record the mixed up conversations, where each recording has a different mix of the sources, as shown in Figure 3.3(b). Given these recordings, could a program recover the three original time series, *without* any other hints? Clearly, our brains can do that in a cocktail party setting and identify individual conversation. We can use the idea of ICA to do it, too. Figure 3.3(c) shows the result of the "un-mixing": the three hidden sources (variables) are well recovered, up to a sign flip.

The idea of using ICA for source separation is based on the assumptions that the original conversations are independent sources, and the recordings of mixed conversations are linear sums of these signals. Given $m$ recordings of $n$ time ticks, we can formulate a $n$-by-$m$ data matrix $\mathbf{X}$, where each column contains the recorded signals at one microphone, i.e., the $(i,j)$-element of $\mathbf{X}$, $x_{i,j}$, is the recorded signal on time tick $i$ from the $j$-th microphone. To discover the hidden, independent conversations from the recordings data $\mathbf{X}$, we can apply ICA, which will give us the $n$-by-$m$ hidden matrix $\mathbf{H}$ (Eq. (3.2)) where each column of $\mathbf{H}$ corresponds to a hidden variable (source). Surprisingly, these hidden variables are exactly the hidden sources (i.e., conversations in the cocktail party) "separated" from the mixed up recordings.

For the example in Figure 3.3, we have $m$=3 microphone recordings at $n$=15,000 time ticks (Figure 3.3(b)). Therefore, the data matrix is $\mathbf{X}_{[\mathbf{15000x3}]}$, and applying ICA gives us two matrices: $\mathbf{H}_{[\mathbf{15000x3}]}$ and $\mathbf{B}_{[\mathbf{3x3}]}$. Plotting the values at each of the 3 columns of matrix $\mathbf{H}$ (time ticks at the X-axis and column value at the Y-axis), we can recover the three original conversations (Figure 3.3(c)), up to a sign flip.

### 3.1.4 Summary

In summary, given a data matrix, ICA analysis can provide two types of patterns: basis vectors (rows of matrix **B**) and hidden variables (columns of matrix **H**) that have have the following properties:

- The *ICA basis vectors* can identify major patterns of a data distribution and provide a good vocabulary to describe the data.
- The data representation based on the ICA basis vectors is called the *ICA hidden variable representation*. The ICA hidden variable representation gives an efficient and *sparse* representation of the data points.
- The *ICA hidden variables* are statistically independent, and can reveal the unobserved trends or hidden sources that influence the observed data points (Section 3.1.3).

Table 3.1 summarizes the symbols used in Part I of this thesis.

## 3.2 ICA and Human Perceptual Processing

The mechanism of how human perceptual systems represents things we see and hear has long been an intriguing topic. Barlow [4] proposed that neurons perform redundancy reduction by making

| Symbol | Description |
|---|---|
| $n$ | The number of data points in a data set. |
| $d$ | The number of attributes per data point. |
| $m$ | The number of hidden variables extracted from the data set ($m < d$). |
| $\mathbf{X_0}$ | The $n$-by-$d$ raw data matrix. |
| $\mathbf{X}$ | The $n$-by-$m$ data matrix (after dimensionality reduction). |
| $\mathbf{H}$ | The $n$-by-$m$ hidden (variable) matrix. |
| $\mathbf{B}$ | The $m$-by-$m$ basis matrix. |
| $\mathbf{B_0}$ | The $m$-by-$d$ (raw) basis matrix. |
| $\vec{\mathbf{x}}_i$ | The $i$-th row of $\mathbf{X}$: the $i$-th data point (a $m$-dim vector). |
| $\vec{\mathbf{b}}_i$ | The $i$-th row of $\mathbf{B}$: the $i$-th basis vector (a $m$-dim vector). |
| $\vec{\mathbf{r}}_i$ | The $i$-th row of $\mathbf{B_0}$: the projection of the $i$-th basis vector in the $d$-dim attribute space. |
| $\vec{\mathbf{h}}_i$ | The $i$-th column of $\mathbf{H}$, containing the values of the $i$-th hidden variable for all $n$ data points in $\mathbf{X}$ (a $n$-dim vector). |
| $b_{i,j}$ | The $j$-th element of the $i$-th basis vector. |
| $h_{i,j}$ | The value of the $j$-th hidden variable on the $i$-th data point (or equivalently, at the $i$-th time tick). |

**Table 3.1:** Summary of symbols used in Part I

up a factorial code for the input, that is, a representation composed of independent elements. This proposal is supported by recent studies on efficient natural image encoding, from the notion of sparse coding [95] (maximizing redundancy reduction), and also from the aspect of independent components [7]. These experimental results show that human perceptual processing is based on *independent features which encode the input signals efficiently*.

For images or auditory signals, the derived independent components generally resemble the receptive fields of neurons measured from mammals' cortex (which is similar to a human's cortex). These independent components resemble Gabor wavelet filters, which are (1) oriented, (2) localized in space (or time), and (3) band-pass in the frequency domain [7, 69]

Analysis has also been extended to the spatial-temporal domain [125] where the independent components of video clips of natural scenes and those of color images [41] are examined. The results are qualitatively similar to those of the static, gray-scale images and are again closely related

**Steps**:
1. Prepare a data matrix **X**.
   **X** is obtained from preprocessing the raw data matrix $\mathbf{X_0}$.
2. Process the data matrix using ICA.
   Result: Basis matrix **B** and hidden matrix **H**.
3. Interpret patterns from the ICA basis vectors and hidden variables, and do data mining.
   3.1 Find the meanings of the patterns.
   3.2 Apply the patterns to applications such as classification.

**Figure 3.4:** General framework of AutoSplit

to the experimental measurements from studies on human perception.

Due to the close relation between independent component analysis (ICA) and human perceptual processing, ICA has been used in applications such as face recognition [6], object modeling [140] and speech recognition [66]. In these applications, methods based on ICA have reported performances that are comparable or better than conventional approaches. However, research is still needed to exploit the full strength of ICA, especially on its application in data mining.

## 3.3 Mining Multimedia Data using AutoSplit

In this section, we introduce our proposed method, AutoSplit [99], for mining uni-modal patterns in multimedia data sets. The general framework of AutoSplit contains three steps (Figure 3.4): preparation of the data matrix, ICA analysis, and pattern mining on ICA basis vectors and hidden variables. In particular, AutoSplit proposes methods to interpret the meaning of patterns that ICA finds, by associating a pattern with the original data attributes.

In the following, we discuss each of these steps, with emphasis on the data matrix preparation (step 1), and the methods for pattern interpretation and mining (step 3).

**Input:** 1. $\mathcal{M}$: A multimedia data set.
2. $m$: The number of patterns and hidden variables to be found.

**Output:** 1. Data matrix: $\mathbf{X}_{[\mathbf{nxm}]}$.
2. Matrix $\mathbf{V}_{[\mathbf{dxd}]}$: principal vectors.
(For mapping from $m$-dimensional space to the original $d$-dimensional data attribute space.

**Steps**:
1. Collect data points from the multimedia data set $\mathcal{M}$.
In total, let there be $n$ data points, each with $d$ attributes (i.e, each point is a $d$-dimensional vector).
2. Construct a $n$-by-$d$ (raw) data matrix $\mathbf{X_0}_{[\mathbf{nxd}]}$, by stacking data point vectors as matrix rows.
2.1 (Optional) Normalize the columns of $\mathbf{X_0}$ to be zero mean and unit variance.
3. Reduce the dimensionality of $\mathbf{X_0}$ from $d$ to $m$, using SVD (Equations 3.5 and 3.6).
3.1 Let $\mathbf{X}_{[\mathbf{nxm}]}$ be the dimension-reduced data matrix.
3.2 The matrix $\mathbf{V}$ from Eq. (3.5) contains principal vectors that will be used
to map from the $m$-dimensional space back to original $d$-dimensional data attribute space.
3.3 If normalization is done at step 2, then $\mathbf{X}$ is equivalent to the result of PCA:
the *PCA basis vectors* are the columns of $\mathbf{V}$,
the *PCA hidden variables* are the columns of $\mathbf{X}$.

**Figure 3.5:** AutoSplit, step 1: Preparing the data matrix

### 3.3.1 Collecting Multimedia Information into a Data Matrix

The first step of AutoSplit is to collect data points – data samples – from the multimedia data set. Depending on the data modality, a data sample could be either the auditory signals in a 1-second long audio segment, or a 12-by-12 block of image pixels from a video frame, or the closing stock prices of 100 companies on a particular day. We represent each data sample as a *raw data vector*, where each raw data vector is a $d$-dim vector of attribute values. A $n$-by-$d$ *raw data matrix* $\mathbf{X_0}$ is formed by stacking $n$ raw data vectors as rows of the matrix. For example, if a data sample is a 12-by-12 block of RGB pixels which contains 12*12*3=432 values of color information, then it will be represented as a 432-dimensional raw data vector (equivalently, a data point in the 432-dimensional space).

The raw data matrix $\mathbf{X_0}_{[\mathbf{nxd}]}$ is then preprocessed by normalizing the data attributes and dimensionality reduction using SVD. Specifically, each data attribute is normalized so that the sample

**Input:** 1. Data matrix: $\mathbf{X}_{[\mathbf{nxm}]}$.
2. Matrix $\mathbf{V}_{\mathbf{m}[\mathbf{dxm}]}$: the first $m$ principal vectors.
**Output:** 1. Matrix $\mathbf{H}_{[\mathbf{nxm}]}$: ICA hidden variables.
2. Matrix $\mathbf{B}_{[\mathbf{mxm}]}$: ICA basis vectors.
3. Matrix $\mathbf{B}_{\mathbf{0}[\mathbf{mxd}]}$: ICA basis vectors in the original data attribute space.
**Steps**:
1. Apply ICA on $\mathbf{X}_{[\mathbf{nxm}]}$ and compute the hidden matrix $\mathbf{H}_{[\mathbf{nxm}]}$ and the basis matrix $\mathbf{B}_{[\mathbf{mxm}]}$.
2. Mapping matrix $\mathbf{B}$ back to original $d$-dim space, obtaining a matrix $\mathbf{B_0}$ (Equation 3.10).
$\mathbf{B_0}=\mathbf{B}\mathbf{V_m}^T$.

**Figure 3.6:** AutoSplit, step 2: Finding patterns using ICA

mean is zero (*zero mean*), and the sample variance is 1 (*unit variance*). Then, the dimensionality of each data point is reduced from $d$ to $m$ ($m \leq d$), where $m$ is the number of uni-modal patterns that we would like to discover.

**Normalizing the Data Attributes** It is common to normalize the columns of the raw data matrix $\mathbf{X_0}$, so that each column has zero mean and unit variance. In the raw data matrix, data attributes may have values from very different ranges. Without normalization, data attributes with extreme values or wide variations may dominate the subsequent analysis and lead to biased results. The normalization step equalizes the data ranges and places equal emphasis on all data attributes. In addition, normalization also helps the numerical computation in subsequent analysis.

However, the decision of whether to have this normalization step or not depends on the data and the application domain. All experimental results reported in the following chapters in Part I of this thesis use this normalization step when preprocessing the data.

**Dimensionality Reduction using SVD** The dimensionality reduction is done by *singular value decomposition* (SVD). There are two reasons for doing this preprocessing: First, the basic ICA model in equation (3.2) requires the dimensionality of a data matrix (the number of columns) be the same as the number of hidden variables, since multimedia data samples usually have high

**Input:** 1. $\mathbf{B}$: The ICA basis vectors.
$\mathbf{B_0}$: The basis vectors in the original $d$-dim attribute space.
2. $\mathbf{H}$: The ICA hidden variables.

**Interpretation**:
(I1) Basis vectors (rows of matrix $\mathbf{B}$) as "vocabulary" for describing the data points.
(I2) Inferring the meaning of a basis vector $\vec{\mathbf{b}}_i$ ($i$-th row of $\mathbf{B}$) (Section 3.3.2):
Examining the elements in the $i$-th row of $\mathbf{B_0}$, e.g., find maximum elements or compute ratios, etc.
Inferring physical meaning by correlating elements with their corresponding data attributes.
(I3) Hidden variables (columns of $\mathbf{H}$) as hidden sources (e.g. trends) of the observed data values.
(I4) Inferring the meaning of a hidden variable $\vec{\mathbf{h}}_i$ ($i$-th column of $\mathbf{H}$) (Section 3.3.2):
Examining the elements of the corresponding basis vector ($\vec{\mathbf{r}}_i$) in matrix $\mathbf{B_0}$.
E.g., find the elements in $\vec{\mathbf{r}}_i$ with maximum magnitude (ignore sign),
the corresponding data attributes are those heavily influenced by the $i$-th hidden variable
(I5) Hidden variable representation (rows of matrix $\mathbf{H}$):
A *sparse* representation of data points, which
could possibly facilitate data mining applications, such as classification.

**Figure 3.7:** AutoSplit, step 3: Interpreting results from ICA

dimensionality, higher than the number of patterns in the data set.

The second reason for applying SVD is that SVD produces dimensionality-reduced data points that have *uncorrelated* attributes (dimensions) [48]. Having uncorrelated attributes facilitates the computation of ICA to discover the *independent* hidden variables of the given data. The intuition is that, although *uncorrelation* does not necessarily imply *independence*, but since *independence* implies *uncorrelation*, having uncorrelated attributes is a stepping stone to obtain independent hidden variables. In fact, for Gaussian data sets, uncorrelation does imply independence.

Mathematically, SVD decomposes the (normalized) matrix $\mathbf{X_{0\,[nxd]}}$ as a product of three matrices:

$$\mathbf{X_0}_{[nxd]} = \mathbf{U}_{[nxd]}\Sigma_{[dxd]}\mathbf{V}^T_{[dxd]}, \tag{3.5}$$

where $\mathbf{U}$ and $\mathbf{V}$ are column-orthonormal matrices, and $\Sigma$ is a diagonal matrix ($\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_m)$), where $\sigma_1 \geq \ldots \geq \sigma_m \geq 0$.

The columns of the matrix $\mathbf{V}$ are called the *principal vectors*, for their relationship with the *principal component analysis (PCA)*. In fact, the combination of data attribute normalization and dimensionality reduction using SVD is equivalent to the processing of PCA. For example, the basis vectors shown in Figure 3.1(b) are principal vectors in $\mathbf{V}$ of the "X-shape" data set.

Since $\sigma_i \approx 0$ for large $i$, we do not lose a lot of information in $\mathbf{X_0}$ if we discard (i.e, set to zero) those near-zero $\sigma$s. As a result, we achieve dimensionality reduction. In other words, let matrix $\mathbf{U_m}$ be the $n$-by-$m$ submatrix of the first $m$ columns of $\mathbf{U}$, and $\Sigma_m$ be the upper-left m-by-m submatrix of $\Sigma$. The dimensionality-reduced data matrix $\mathbf{X_{[nxm]}}$ is defined as the product of $\mathbf{U_m}$ and $\Sigma_m$, i.e.,

$$\mathbf{X}_{[nxm]} = \mathbf{U_m}\Sigma_m. \tag{3.6}$$

The data matrix $\mathbf{X_{[nxm]}}$ is the matrix on which we will apply ICA to find $m$ hidden variables and $m$ basis vectors.

A rule-of-thumb for determining the value of $m$ (the amount of patterns) is to pick a value such that the sum $\sum_{i=1}^{m}\sigma_i^2$ is a significant portion (say, 95%) of the sum of all $\sigma$s: $\sum_{i=1}^{d}\sigma_i^2$. Other threshold percentages like 90% or 99% are also usually used. That is, pick the value of $m$ such that

$$\frac{\sum_{i=1}^{m}\sigma_i^2}{\sum_{j=1}^{d}\sigma_j^2} \geq 95\%. \tag{3.7}$$

**Data Matrix X and the PCA Basis Vectors** In Eq. (3.5), the construction of the data matrix $\mathbf{X}$ is equivalent to projecting the raw data matrix $\mathbf{X_0}$ on to the first $m$ columns of the matrix $\mathbf{V}$, $\mathbf{V_m}$.

$$\mathbf{X}_{[nxm]} = \mathbf{X_0}_{[nxd]}\mathbf{V_m}_{[dxm]}. \tag{3.8}$$

Rewriting Eq. (3.8) using the orthonormal property of $\mathbf{V_m}$ (i.e., $\mathbf{V_m}^{-1}$=$\mathbf{V_m}^T$), we have another view of the data matrix $\mathbf{X}$:

$$\mathbf{X_0}_{[nxd]} = \mathbf{X}_{[nxm]}(\mathbf{V_m}^T)_{[mxd]}, \tag{3.9}$$

where the $m$ principal vectors in matrix $\mathbf{V_m}$ can be considered as the basis vectors computed by PCA (*PCA basis vectors*), and the data matrix $\mathbf{X}$ can be considered as the corresponding *PCA hidden variables*.

We can compare the PCA basis vectors in matrix $\mathbf{V_m}$ with the ICA basis vectors in matrix $\mathbf{B}$, and evaluate the strengths and disadvantages of the two methods. For example, we can compare which set of basis vectors describe the patterns in the data set better, as we did in Figure 3.1. It is worth mentioning that the PCA result (matrices $\mathbf{V_m}$ and $\mathbf{X}$) is computed with no extra cost: they are products of an intermediate step in the ICA analysis.

### 3.3.2 Mining/Interpreting the ICA Result

The last step of mining multimedia data using ICA is to interpret the discovered hidden variables and basis vectors. In the following, we describe our proposed methods for using the ICA basis vectors and hidden variables, and inferring information by inferring their possible physical meanings. Figure 3.7 summarizes our proposed interpretation of ICA basis vectors and hidden variables.

The ICA basis matrix $\mathbf{B}$ and the hidden matrix $\mathbf{H}$ are meaningful and could be used in many applications. For example, we can distinguish two (or more) data sets ($\mathbf{X_1}$, $\mathbf{X_2}$) by comparing the basis matrices ($\mathbf{B_1}$, $\mathbf{B_2}$) computed from each of them. The basis vectors in $\mathbf{B}$ are a good vocabulary for describing a data set, and data sets with different characteristics will have different vocabularies. By comparing the differences in the vocabularies, we can identify how one data set differs from another one. This idea can also be used for classification, where a data point $\vec{\mathbf{x}}_i$s classified to the data set whose vocabulary best describes $\vec{\mathbf{x}}_.$ An application of this is given in chapter 4.

To infer the physical implications of basis vectors and hidden variables, one common way is to relate them to the original data attributes of the raw data matrix $\mathbf{X_0}$. Very often, the data attributes in $\mathbf{X_0}$ are measurements of real-world objects and bear interpretable meanings. Examples of raw data attributes include the loudness of an audio signal, the energy exerted at human knees in a motion capture data set, or the stock price of a company.

**Interpreting the Basis Vectors** To interpret the $m$-dimensional ICA basis vectors of the $\mathbf{B}_{[\mathbf{mxm}]}$ matrix, we first map $\mathbf{B}$ back to the $d$-dimensional ($d \geq m$) space of original data attributes. The mapping is done by multiplying $\mathbf{B}$ with the matrix $(\mathbf{V}^T)_{[mxd]}$, resulting in a $m$-by-$d$ matrix $\mathbf{B_0}$:

$$\mathbf{B_0}_{[mxd]} = \mathbf{B}_{[mxm]}(\mathbf{V_m}^T)_{[mxd]}, \tag{3.10}$$

where $\mathbf{V_m}$ be the $d$-by-$m$ matrix which contains the first $m$ columns of the matrix $\mathbf{V}$ from the SVD of raw data matrix $\mathbf{X_0}$ (Equation 3.5.

Each row vector in matrix $\mathbf{B_0}$ corresponds to a row vector in matrix $\mathbf{B}$: the $i$-th row of $\mathbf{B_0}$ is the mapping of the $i$-th row of $\mathbf{B}$ in the $d$-dim space. Since the row vectors of $\mathbf{B}$ are the ICA basis vectors, in the following, we will call a row vector in $\mathbf{B_0}$ a "*raw ICA basis vector*", using the notation $\vec{\mathbf{r}}_i$ for the $i$-th row of $\mathbf{B_0}$.

Note that each raw ICA basis vector $\vec{\mathbf{r}}_i$ is unit-length, just as the ICA basis vectors $\vec{\mathbf{b}}_i$. This is because the columns of matrix $\mathbf{V}$ are orthonormal (therefore, unit-length), and can be checked by simple algebra.

**Interpreting the Hidden Variables** To infer the potential physical meaning of a basis vector $\vec{\mathbf{b}}_i$ (the $i$-th row of $\mathbf{B}$), we examine the elements on the $i$-th row of $\mathbf{B_0}$. Let the $i$-th row of $\mathbf{B_0}$ be $\vec{r}_i = [r_{i,1}, \ldots, r_{i,d}]$. The relationships among elements $r_{i,j}$, such as ratios, give information about the meaning of basis vector $\vec{\mathbf{b}}_i$. As an example, in chapter 5, we examine the ratio between the energy exerted at right knee versus that at left knee, to determine the patterns we found in human motions.

On the other hand, to interpret the physical meaning of a hidden variable, we consider which data attributes (real world measurements) are "sensitive" and highly "influenced" by the given hidden variable. From Eq. 3.2, we can see that the $i$-th hidden variable ($i$-column of $\mathbf{H}$) corresponds to the $i$-th basis vector (row in $\mathbf{B}$), and therefore corresponds to the $i$-th row $\vec{\mathbf{r}}_i$ in $\mathbf{B_0}$. The magnitude of an element $r_{i,j}$ shows how much the $j$-th data attribute is sensitive to the value of the $i$-th hidden variable $\vec{\mathbf{h}}_i$. Knowing which data attributes are largely influenced by a hidden variable $\vec{\mathbf{h}}_i$, we could infer the physical meaning of $\vec{\mathbf{h}}_i$. As an example, in chapter 5, we identify the potential meanings of two interesting hidden variables in the stock prices of 29 companies, which are consistent to the known historical trend and events in the stock market.

Mathematically, the relationship between hidden variables (matrix $\mathbf{H}_{[\mathbf{nxm}]}$), the (raw) basis matrix in the $d$-dim data space (matrix $\mathbf{B_0}_{[\mathbf{mxd}]}$), and the raw data matrix $\mathbf{X_0}_{[\mathbf{nxd}]}$ is as follows:

$$\begin{aligned}
&(3.11) & \mathbf{X_0}_{[nxd]} &= \mathbf{X}_{[nxm]}(\mathbf{V_m}^T)_{[mxd]} && \text{(Eq. 3.9)}\\
&(3.12) & \mathbf{X}_{[nxm]} &= \mathbf{H}_{[nxm]}\mathbf{B}_{[mxm]} && \text{(Eq. 3.2)}\\
&(3.13) & \mathbf{B_0}_{[mxd]} &= \mathbf{B}_{[mxm]}(\mathbf{V_m}^T)_{[mxd]} && \text{(Eq. 3.10)}\\
&(3.14) & \Rightarrow \mathbf{X_0}_{[nxd]} &= \mathbf{H}_{[nxm]}\mathbf{B}_{[mxm]}(\mathbf{V_m}^T)_{[mxd]} &&\\
&(3.15) & &= \mathbf{H}_{[nxm]}\mathbf{B_0}_{[mxd]}. &&
\end{aligned}$$

**Summary** The interpretation of the ICA result can be summarized as follows:

- (I1) The ICA basis vectors can be used as a *vocabulary* for describing the data set.

- (I2) The physical meaning of a basis vector $\vec{\mathbf{b}}_i$ can be inferred by mapping the vector to the original data attribute space $\vec{\mathbf{r}}_i$, and examining the elements of $\vec{\mathbf{r}}_i$ (e.g., ratios between vector elements).

- (I3) The ICA hidden variables (columns of $\mathbf{H}$) reveal the hidden sources (e.g. trends) of the observed data values. This is also called *blind source separation*.

- (I4) The physical meaning of a hidden variable $\vec{\mathbf{h}}_i$ can be inferred by examining the elements of the corresponding vector $\vec{\mathbf{r}}_i$. The elements of $\vec{\mathbf{r}}_i$ with large magnitudes correspond to data attributes that are "sensitive" to the change of the hidden variable $\vec{\mathbf{h}}_i$.

- (I5) The "sparse" hidden variable representation of data points are useful in many applications, such as classification.

## 3.4 Organization of Case Studies

In the following chapters of Part I, we explore the applications of ICA in a variety of multimedia data sets, and propose extensions for specific tasks. In chapter 4, we derive spatial-temporal patterns in video frames and audio signals. The patterns are meaningful, and allow us to classify video clips into news and commercials with about 81% accuracy. In chapter 5, we show that the hidden variables and basis vectors extracted from time series and multimedia databases are meaningful, and reveal information hidden behind the observed attributes. We also propose a method to extend our analysis to multiple sets of basis vectors (for mixture of data). In chapter 6, we propose a framework to create visual vocabulary for biomedical images. The visual vocabulary we construct has biological meaning, and enable a variety of data mining applications such as classifying normal and disease cases, or highlighting characteristic regions in images of different conditions.

# Chapter 4

# Uni-Modal Patterns of Video Clips

Video clips contain rich information of various modalities such as image (video frame), audio, and text (transcript), and have become a major source for news, sports, and entertainment, etc. However, making the rich information in video clips accessible is not an easy task: *How do we find the meaningful patterns that summarize the content of video clips automatically? Do these patterns help us to organize clips for searching and browsing, or on content-based classification?*

Automatic pattern discovery methods are needed to analyze the information of various modalities in video clips. We would like to find patterns which summarize the content of video clips, as well as those that differentiate different types of clips. Having an automated method to find such patterns would facilitate information extraction and data mining from video clips. Patterns that are meaningful and useful may also be used as features to classify video clips of different types (e.g., news or commercial), or as database indexes to improve retrieval performance.

Many previous approaches find patterns by extracting features which are manually-designed based on domain knowledge. Despite numerous research efforts, there is no consensus on a set of features that are useful in general video applications. The performance of a feature usually varies depending on the video collection. Other useful features, such as the color correlogram features or the MFCC audio features, are based on general signal processing techniques. However, these features may not have intuitive interpretation, and therefore, provide no insight for a human user to understand the characteristics of a video type.

In this chapter, we focus on automatically finding uni-modal patterns in text, images, and audio from video clips. We propose a framework that could extract meaningful and useful patterns from the data of various modalities in video clips. The questions we are interested in are:

- **Generality:** How do we extract patterns from a variety of modalities (e.g., text, image, audio, etc.) in video clips?

- **Interpretation:** How do we interpret the patterns? Are the extracted patterns meaningful and consistent with human intuition about the content of a video collection?

- **Applicability:** How do we use the extracted patterns, say for classifying clips into different genres (e.g., "news" and "commercial")?

We proposed a procedure for extracting uni-modal patterns from video clips. The proposed framework is based on AutoSplit (Figure 3.4), and the main focus is to capture the temporal characteristic in video clips. Our method is general and can extract meaningful patterns from various modalities. In particular, we extracted spatial-temporal patterns from pixels of video frames (*VideoBasis*), auditory patterns of audio in video clips (*AudioBasis*), as well as the patterns in transcripts which are consistent with the *topics* of the news stories in our video collection. The topics found by AutoSplit are better than those extracted by principal component analysis (PCA). Finally, we also proposed a scheme for classifying news videos and commercials, based on the concept of compression. On a two-class, news-versus-commercial classification problem, using the patterns we found from video frames and audio, we could achieve greater than 81% classification accuracy.

The rest of this chapter is organized as follows: In section 4.1, we describe how we extract temporal characteristics using AutoSplit. In sections 4.2 4.3, and 4.4, we discuss the extracted patterns from video frames, audios, and transcripts, respectively. In section 4.5, we present our proposed classification scheme and the results on classifying news videos and commercials.

## 4.1 Capturing Temporal Characteristics via Windowing

One important aspect of video clips is their temporal characteristics: the audio and transcript are temporal in nature, and the sequential video frames contain information about the spatial pixel configuration in each frame, as well as their behavior over time. In this section, we introduce how to adapt our proposed AutoSplit framework in Chapter 3 (Figure 3.4) for mining temporal patterns from the various modalities (transcripts, video frames and audios) in video clips. Our methods successfully extract uni-modal patterns from data of different modalities, and as we show later, these patterns are meaningful, consistent with human intuition about the video content, and are useful for video applications such as classification.

Our proposed AutoSplit framework for uni-modal discovery, outlined in Figure 3.4 (Chapter 3), contains three steps: preparing a data matrix, finding patterns using ICA, and interpreting the patterns. The first step, preparing the data matrix, is domain-dependent for data of different modalities. For a particular modality, the data samples in the data matrix should contain essential information of that modality. Therefore, we may need different ways to build the data matrix for different modalities of data.

*How do we construct a data matrix that captures essential (temporal) information in the video clips?* We propose a *windowing* approach in constructing the data matrix, to discover temporal patterns. For the three modalities in video (text, audio, and video), three types of sliding windows are designed for the different data modalities.

The transcript of a video clip consists of a sequence of terms. For transcripts, we propose to take windows of consecutive terms. Each window contains the same number of terms, and the windows may overlap with one another. We represent the terms in a window as a $d$-dimensional vector of term frequency counts, where $d$ is the size of the vocabulary of the transcripts. If $n$ windows of transcripts are taken from the video collection, then we have a raw data matrix $\mathbf{X_{0[nxd]}}$ by stacking up the $d$-dimensional vectors of the $n$ windows.

Video frames contains not only temporal information between frames consecutive in time, but also spatial information among pixels in the same frame. To incorporate both spatial and temporal characteristics of video frames, we designed a *3-dimensional window* which collects pixels that are adjacent in space and in time. The results are "cubes" of pixels which contain changes among pixels, spatially and temporally, in video clips. A cube of size $s$-by-$s$-by-$s$ contains $d$=$s^3$ pixels. Each pixel is a scalar of gray-scale between 0 and 255. We will consider each cube as a $d$-dimensional vector, by linearizing $s^3$ pixels (scanning the pixels row by row, starting from the first frame of a cube). We can collect a $n$-by-$d$ raw data matrix $\mathbf{X_{0[nxd]}}$ by randomly taking $n$ cubes from the video frames.

If we are interested in the audio part only, AutoSplit is also applicable to extract patterns from audios. The audio in a video clip is a stream of auditory signals measured at some fixed frequency (e.g., 44.1kHz for CD-quality sound). To prepare the data matrix for audio, we define a window as a sequence of $d$ consecutive auditory signal measurements. We treat the $d$ values in a window as a $d$-dimensional vector. Each $d$-dimensional vector becomes a data sample in the data matrix. The raw data matrix $\mathbf{X_{0[nxd]}}$ of audio in a video collection is formed by taking $n$ windows of audio samples.

As we will show in the following subsections, despite the differences between video frames, auditory signals, and transcript terms, the proposed windowing approach is effective in capturing the essential patterns in these different data modalities. To refer to the different types of windows for different data modalities, we refer to the 3-dimensional windows from video frames as "*VideoCubes*", the 1-dimensional windows from audio tracks as "*AudioCubes*", and the 1-dimensional ones from transcripts as "*TextCubes*". Figure 4.1 summarizes the proposed methods for collecting data matrix from transcript, frame, and audio of video clips.

After we have the raw data matrix ready, we can now find patterns (basis vectors and hidden variables), following the steps of AutoSplit: preprocessing the raw data matrix, determining how many hidden variables to extract, and interpreting the found patterns (Chapter 3, Figure 3.4).

In particular, we will focus on finding good *vocabulary* for describing the patterns in video frames, audios, and transcript text. As we explained at items (I1) and (I2) in Figure 3.7, the ICA

**Input:** 1. $\mathcal{M}$: A uni-modal data set from video clips.
2. $s$: Window size parameter.

**Data samples**:

(D1) Video frame (image stream):
**(VideoCube)** Cubes of $s$-by-$s$-by-$s$ pixels adjacent in space and time (Section 4.2).
Data sample: a $d$-D vector of a cube of $d = s^3$ pixels.
Pixels are linearized into a row vector.

(D2) Audio (signal stream):
**(AudioCube)** Segments of $s$ consecutive audio signals (Section 4.3).
Data sample: a $d$-D vector of audio signals ($d = s$).

(D3) Transcript (text stream):
**(TextCube)** Segments of $s$ consecutive word tokens (Section 4.4).
Data sample: a $d$-D vector of word frequency counts; $d$ is the size of the text vocabulary.

**Figure 4.1:** Preparing raw data matrices from video clips.

basis vectors can act as an good vocabulary for describing the data characteristics. We will compare the differences in vocabularies of different video types: how do they differ in audio, or in video frames? We will also use the difference in vocabularies to classify clips of different types.

In the rest of this chapter, we will explain the specific steps in our experiments and describe the patterns we found: Sections 4.4, 4.2, and 4.3 for patterns from VideoCubes, AudioCubes, and TextCubes, respectively. In Section 4.5, we propose a scheme for distinguishing news videos from commercials, using patterns from VideoCubes and AudioCubes.

### 4.1.1 Related Work

Previous work had tried different visual/auditory features derived from video clips for classification. There are studies based on pixel-domain visual information (color histograms) [75], and transform (compressed) domain visual information [35]. Other kinds of meta-data, such as motion vectors [108] and faces [21], have also been used. On the other hand, time-domain and frequency-domain auditory features have also been used on classifying video genres [80, 107]. Recent studies [31] combined visual and auditory features for video classification. However, these features are usually hand-picked, and their applicability relies on the experiences of the researchers, and their understanding of the deploying domain.

Another design decision in feature extraction is to whether extract global features or local ones. With increased complexity, the latter [113] provides class models that are more accurate and with higher representative power [140].

Several approaches have been examined for the representation of genre classes, including statistical (Gaussian) modeling [107] and the hidden Markov model [21]. There are also works that represent class models using general classifiers, such as decision trees or neural networks.

Previous results on video classification used various features and several modeling approaches to classify different sets of genre classes. For example, Roach et. al [107] used audio features on 5 classes: sports, cartoon, news, commercial and music. They achieved 76% accuracy on classification. Truong et. al [121] used visual statics, dynamics, and editing features on 5 classes: sport, cartoon, news, commercials and music videos. They reported an accuracy of 80%. Liu et. al [80] used audio features on 3 classes, commercial, report (news and weather), and sports, and reported a classification accuracy at 93%.

Because of the different sets of genre classes and the different collections of video clips these previous works chose, it is difficult to compare the performance of the different features and approaches they used.

## 4.2 The Spatial-Temporal Patterns in Video Frames

Video clips of different genres (e.g., news and commercial) display different characteristics in video frames. For example, news clips may have more in-studio scenes and slower motion, while commercials tend to have faster motion. How do we find these kinds of patterns from video frames, automatically?

Video frames are successive images in temporal order. To derive the visual patterns of video clips, we focus on finding patterns among pixels in video frames. In particular, we are interested in questions such as: *What are the spatial-temporal patterns in clips of a video genre? How do the patterns differ in different genres of video?*

In this section, we describe our method for extracting visual patterns from video frames. We first introduce "VideoCubes" which incorporate both spatial and temporal information in video frames, followed by discussions of our experiments and the patterns found.

### 4.2.1 VideoCubes and VideoBases

In a sequence of video frames, there are correlations between pixels in the same frame (spatial correlation), and also between pixels on frames that are close in time (temporal correlation). We say that two pixels are *spatially adjacent* if they are adjacent to each other in the same video frame, and two pixels are *temporally adjacent* if they are at the same position of adjacent frames. For example, pixels (2,1) and (2,2) on frame 1 are spatially adjacent, and pixel (2,1) on frame 1 and pixel (2,1) on frame 2 are temporally adjacent. To jointly consider both the spatial and temporal information and find spatial-temporal patterns, we propose to group spatially and temporally adjacent pixels into "cubes" as basic data units. We call these cubes of pixels *VideoCubes*.

**Definition 1** ***(VideoCube)*** *The pixel located at position (x,y) on frame t is denoted as pixel (x,y,t). A* ***s-by-s-by-s cube*** *located at (x,y,t) contains all pixels (i,j,k) where i=x,...,(x+s-1), j=y,...,(y+s-1), and k=t,...,(t+s-1). We call such cubes* ***VideoCubes****.*

We propose to extract ICA basis vectors from VideoCubes. As described in Chapter 3, ICA basis vectors could act as a vocabulary of the data (Figure 3.7 (I1)). VideoCubes contains information about the spatial-temporal behavior of pixels in video frames, and therefore, the ICA basis vectors may give us the general characteristics (vocabulary) about how pixels change in space and in time. We call such a vocabulary the *VideoBasis*, and each individual vector is called a *VideoBasis vector*.

### 4.2.2 Experiment Setup

In this subsection, we describe the video data set and the parameters used in our experiments. The main focus is on the preparation of the data matrix **X** for subsequent ICA computation.

The overall view of the process is summarized as follows: To find the VideoBasis of a specific genre, we first randomly sample VideoCubes from some given clips of this genre. With these VideoCubes, each of size $s$-by-$s$-by-$s$, we can make a raw data matrix $\mathbf{X_{0[nxd]}}$ ($d = s^3$), as explained in Section 4.1 (D1). Following the steps of AutoSplit (Figure 3.4), we can compute the basis matrix **B**, which contains the VideoBasis (ICA basis vector) of this video genre.

**Data Set** The video clips used in our experiments are segments from CNN nightly news reports. We divided the collection into training and testing sets. The training set contains 8 news clips and 6 commercial clips, each about 30 seconds long. The testing set contains 62 news clips and 43 commercial clips, each about 18 seconds long. We used clips in the training set to construct the VideoBases for the two video genres of interest: news and commercial. Clips in the testing set were used in the classification experiment described in Section 4.5.

In this study, our video clips are in the MPEG format, and we considered only the I-frames. There is one I-frame every 0.5 second, and therefore, each clip in the training set (30 seconds long) gives us a stack of 60 I-frames. In our experiments, we randomly sampled VideoCubes of size 12x12x12 ($s = 12$) from the such stacks of I-frames. Therefore, each VideoCube contains $12^3 = 1728$ pixels.

Each video frame is a 352-by-240 RGB pixel image. In our experiments, we used only the information in the red channel of each RGB pixel. Our experience showed that the VideoBasis based on any of the three channels has similar characteristics. In addition, we divide each video frame into 9 regions (3-by-3 regions) and consider only VideoCubes taken from the central region. This refinement allows us to focus on the central activities in a video clip, and reduce noise and spurious patterns.

**Computing VideoBasis** Suppose that we want to compute the VideoBasis of the video genre "news". We prepare the data matrix $\mathbf{X}$ for ICA computation as follows: We sample VideoCubes of size 12x12x12 from the 8 news clips in the training set. In total, 10,000 VideoCubes are sampled, resulting in a $10,000$-by-1728 raw data matrix $\mathbf{X_0}_{[\mathbf{10,000x1728}]}$. We reduce the dimensionality from 1728 to 160, using SVD as outlined in Figure 3.5. The result is the $10,000$-by-160 data matrix $\mathbf{X}$.

To compute the VideoBasis of commercials, the only difference is that we took VideoCube samples from the 6 commercial clips in the training set. We note that the number of video clips in the training set is not an important factor on the quality of the derived VideoBasis. What really matters is the number of VideoCubes used for extracting features.

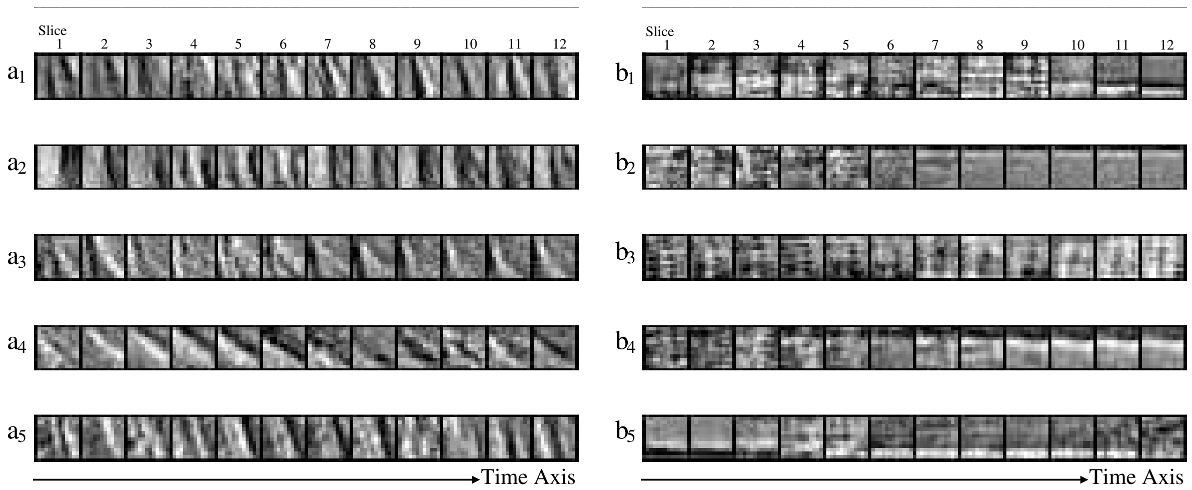
We used the FastICA [48] package from the Helsinki University of Technology for ICA computation. Given the data matrix $\mathbf{X}_{[\mathbf{10,000x160}]}$, we computed the basis matrix $\mathbf{B}_{[\mathbf{160x160}]}$. The hidden matrix $\mathbf{H}_{[\mathbf{10,000x160}]}$ is obtained at the same time, and we will discuss its use in classification in Section 4.5. The rows of the basis matrix $\mathbf{B}$ are the VideoBasis vectors in the 160-D space. We mapped the VideoBasis vectors back to the original 1728-D space, yielding a matrix $\mathbf{B_0}_{[\mathbf{160x1728}]}$, using Equation (3.10), so that we can evaluate the properties of these VideoBasis vectors.

We evaluated the extracted VideoBasis vectors by visually inspecting the spatial-temporal patterns that these vectors depict. The visualization of each VideoBasis vector is done by "de-linearizing" the $s^3$ dimensional vectors into a $s$-by-$s$-by-$s$ cube. In our case, the parameter $s$ is $s = 12$. A $s$-by-$s$-by-$s$ cube can be visualized as a stack of $s$ pixel patches, each of size $s$-by-$s$, as we show in Figure 4.2.

### 4.2.3 Experimental Results

Figure 4.2 shows some VideoBasis vectors of news videos and commercial clips. Each $12^3$-dim VideoBasis vector is viewed as a 12-by-12-by-12 cube, and is visualized by showing its 12 12-by-12 slices. In the figure, the slices of a VideoBasis vector are shown in one row, and are arranged from left to right in their temporal order (the leftmost slice is earliest in time).

In general, VideoBasis vectors of news videos (Figure 4.2 (a)) show clear edge patterns with




(a) VideoBasis (news) (b) VideoBasis (commercial)

**Figure 4.2:** Selected VideoBasis vectors of (a) news clips , and (b) commercial clips. Each $12^3$-dim VideoBasis vector is viewed as a 12-by-12-by-12 cube, and visualized by showing its 12 12-by-12 slices in a row. From left to right, the slices of a VideoBasis vector are arranged in their temporal order. VideoBasis of news shows clear edge patterns with slow movement; that of commercials has no clear pattern, but exhibits significant transitions along the time axis.

slow movement. For example, the vector $\mathbf{a_4}$ shows a white edge with -$45^o$ slope, which moves slowly in time: first from bottom-left to top-right (slice 1 to 4), and then reverse (slice 5 to 8), and then reverse again (slice 9 to 12).

On the other hand, VideoBasis vectors of commercials display no clear pattern (Figure 4.2 (b)), but exhibit significant transitions along the time axis. As an example, the vector $\mathbf{b_4}$ shows a big transition between slices 6 and 7, separating some random patterns in slices 1 to 6, from the pattern of a downward-moving edge shown in slice 7 to 12. The random patterns in slices 1 to 6 of vector $\mathbf{b_4}$ (or slices in other vectors) also indicate fast scene changes that are common in commercials.

The VideoBases of the two video genres, news and commercial, as shown in Figure 4.2, capture patterns that are consistent with a human's intuitions on the major characteristics of these two genres. In a news report, it is more usual to see clear edges, as there are more in-studio scenes or shots of artificial objects like buildings. The object or camera movement in a news report tends to be slow. The edge-like scene and slow movement are exactly what are captured in the VideoBasis of news videos. On the other hand, commercials tends to have many scene changes and motions, partly because they try to attract viewers' attention, and partly due to the shorter durations they usually have. There are also more outdoor or "lively" scenes in commercials, which do not have many regularities such as edges. These two properties, fast scene changes and fewer regularities, are exactly what the VideoBasis of commercials captured.

## 4.3 The Auditory Patterns of Videos

Besides video frames, audio is also an important element of video clips. As we observe in our daily experiences, video clips of different genres have different characteristics in audio. For instance, news clips have mainly human speech in regular volume, while a commercial contains a dynamic mixture of music and human voice. *How do we find these patterns in audio automatically?*

In this section, we describe the extraction of auditory patterns of a video genre using AutoSplit. Like VideoBasis, AutoSplit constructs a vocabulary to characterize the audio in video clips of a genre, based on data samples collected from the audio track in video clips. We call these audio data samples *AudioCubes*, and the auditory patterns *AudioBases*, for their similarity to the VideoCubes and VideoBases in video frames.

### 4.3.1 AudioCubes and AudioBases

As described in Figure 4.1, an AudioCube is a segment of audio signals taken from the audio track of a video clip. Given video clips of a particular video genre, AudioCubes are collected from the audio tracks of these clips, forming a raw data matrix for computing the ICA basis vectors (AudioBases).

We compute the AudioBases of news and commercial videos, based on the same video clips that we used to compute the VideoBases in the previous section (Subsection 4.2.2). This set of clips contains 8 news clips and 6 commercial clips, each about 30 seconds long. The frequency of the audio signals in a clip is 44.1kHz (44100 signals in a second).

In our experiments, an AudioCube corresponds to a 0.5-second audio segment, randomly sampled from a video clip. There are 22050 (=44100*0.5) signals in a 0.5-second segment, and we down-sampled the signals by a factor of 10, keeping only 2205 signals per AudioCube. The 2205 signals in each AudioCube were considered as a 2205-D data vector. These data vectors are put together into a raw data matrix $\mathbf{X_0}$ for computing an AudioBasis.

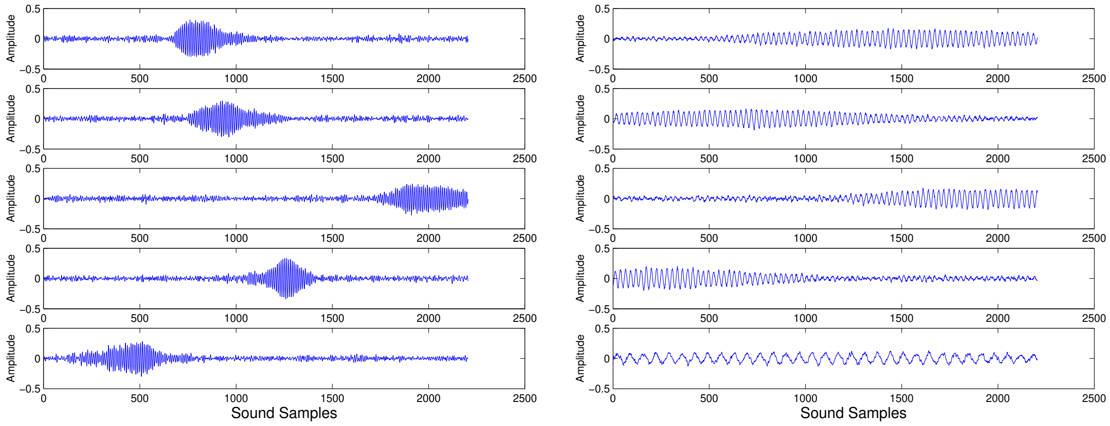
The down-sampling is a tradeoff between the segment length and the data dimensionality: We want segments long enough to capture auditory characteristics. However, without downsampling, the dimensionality of 22,050 is simply too high for subsequent computation.

**Computing AudioBasis** To compute the AudioBasis of a particular video genre, the first step is to prepare a data matrix for the ICA computation. The general procedure was outlined in Figure 3.5: we first prepare the raw data matrix $\mathbf{X_0}$, followed by normalization and dimensionality reduction.

The raw data matrix $\mathbf{X_0}$ contains AudioCubes sampling from video clips of the genre of interest. As an illustration, we detail the steps for the genre "news". For the video genre "news", the AudioCubes are sampled from the 8 news clips in the training set. Each AudioCube is represented as a 2205-D vector. In total, 7,000 AudioCubes are sampled from the 8 clips, resulting in a $7,000$-by-2205 raw data matrix $\mathbf{X_0}_{[\mathbf{7,000x2205}]}$. We reduce the dimensionality from 2205 to 60, using SVD, as outlined in Figure 3.5. The result is the $7,000$-by-60 data matrix $\mathbf{X}$. As for the genre "commercial", the only difference is that the AudioCubes were from the 6 commercial clips in the training set.

As we have mentioned in the section on VideoBasis, the number of video clips in the training set is not an important factor on the quality of the derived AudioBasis. What really matters is the number of AudioCubes used for extracting features.

We use the FastICA [48] package from the Helsinki University of Technology for ICA computation. Given the data matrix $\mathbf{X}_{[\mathbf{7000x60}]}$, we compute the basis matrix $\mathbf{B}_{[\mathbf{60x60}]}$. The rows of the basis matrix $\mathbf{B}$ are the AudioBasis vectors in the 60-D space. (The hidden matrix $\mathbf{H}_{[\mathbf{7000x60}]}$ is




(a) AudioBasis (news) (b) AudioBasis (commercial)

**Figure 4.3:** (AudioBasis) (a) Basis for news resembles waveforms of human speech. (b) Basis for commercials resembles waveforms of natural sounds and animal vocalization (harmonic sounds).

obtained at the same time, but is not discussed in this section.)

To examine the pattern described by the extracted AudioBasis vectors, we first mapped the AudioBasis vectors back to the original 2205-D space, using Equation (3.10), yielding a matrix $\mathbf{B_0}_{[\mathbf{160x2205}]}$. Each row of $\mathbf{B_0}$ corresponds to an AudioBasis vector, and can be considered as a sequence of 2205 auditory signals. Therefore, we can plot the sequence of signals of an AudioBasis vector, and visually inspect and infer the pattern captured in the vector.

### 4.3.2 Experimental Results

Figure 4.3 shows several vectors in the AudioBases of news and commercial clips. The elements of each 2205-D AudioBasis vector are plotted as a sequence of signal values at 2205 time ticks and become a waveform.

The waveforms of AudioBasis vectors of news clips contain amplitude envelopes that are localized in time (Figure 4.3(a)). The waveform pattern is intermediate between waveforms of pure harmonic sounds and pure non-harmonic sounds, and resembles waveforms of human speech (mix of harmonic vowels and non-harmonic consonants). This observation agrees with our experience that the most frequent sound in news stories is human speech.

The AudioBasis vectors of commercials have waveforms that are more harmonic (Figure 4.3(b)), which are similar to those of animal vocalizations and natural sounds [69]. This may be due to the fact that, in commercials, music is more dominant than speech.

In general, these AudioBasis vectors we found coincide with our common knowledge about the sounds that occur in news clips and commercials. It is pleasantly surprising that these AudioBasis vectors are found automatically, with no extra information or supervision from a human, and yet they precisely capture the characteristics of audio in video clips.

## 4.4 Hidden Topics of Video Transcript

A news video usually contains multiple news stories, each with a different topic. *Given the transcripts of news videos, how do we identify the major topics reported in these videos?* In general, *given a stream of text (e.g., transcripts of news videos), how do we identify patterns in the text stream and do segmentation?"*

In this section, we apply AutoSplit for finding patterns in the transcript of broadcast news videos. In particular, AutoSplit computes the ICA basis vectors of the transcripts, where each basis vector can be considered as a hidden news topic in the transcript. Comparing ICA with PCA, each ICA basis vector correspond nicely to one true news topics, while more than one true topics are mixed in an PCA basis vector. The nice correspondence between ICA basis vectors and news topics provides a good topic-based representation for each transcript segment, where the representation is sparse and gives clear indication of the correct topic of a segment.

In the following, we will first describe our data set, and then the details of our method, followed by the experimental results.

| ID | Topic | #Articles |
|---|---|---|
| $TP_1$ | Sgt. Gene McKinney is on trial for alleged sexual misconduct | 60 |
| $TP_2$ | A bomb explodes in a Birmingham, AL abortion clinic | 18 |
| $TP_3$ | The Cattle Industry in Texas sues Oprah Winfrey for defaming beef | 45 |
| $TP_4$ | New impotency drug Viagra is approved for use | 52 |
| $TP_5$ | Diane Zamora is convicted of helping to murder her lover's girlfriend | 22 |
| $TP_6$ | 1998 Winter Olympic games | 20 |
| $TP_7$ | The Pope's historic visit to Cuba in Winter 1998 | 39 |
| $TP_8$ | The economic crisis in Asia | 69 |
| $TP_9$ | Superbowl XXXII | 23 |
| $TP_{10}$ | Tornado in Florida | 38 |

**Table 4.1:** Ten topics in the CNN transcript data set

### 4.4.1 Data Preparation

The data we used is the CNN Headline News articles in the 1998 Topic Detection and Tracking Phase 2 text collection [1]. These news articles are dated between January and June in 1998, and each article was labeled with a specific topic. We remove topics that have few articles, and consider only the articles in the 10 topics which are shown in Table 4.1.

We preprocess the words in the articles by keeping only the text body and applying stop word elimination and stemming. Articles that are shorter (in terms of word counts) than 30 words after preprocessing are dropped. The "#Article" column in Table 4.1 shows the number of articles after this data cleaning process. The remaining 386 news articles are then sorted in chronological order to make a text stream. There are 3,887 distinct words in the text stream, after all the preprocessing.

To employ ICA to find patterns in a text stream, we need to first prepare a data matrix (Figure 3.4). We apply a sliding window approach on the text stream to get the data vectors for the data matrix (Figure 4.1). Specifically, the text stream is split into windows of 30 words; subsequent

[1] http://www.nist.gov/speech/tests/tdt/tdt98/

windows share an overlap of 50%. For each window, a frequency histogram of 3887 terms is generated and used as a data vector. In total, we extract 1,659 windows from the data set, and obtain 1,659 3,887-dimensional vectors. Putting these data vectors together into a matrix form, we could get a 1659-by-3887 raw data matrix $\mathbf{X_0}_{[\mathbf{1659x3887}]}$.

Since we know that there are 10 hidden topics to be discovered from the CNN text stream, we aim at finding *m*=10 basis vectors (topics). To discover these 10 hidden topics, we reduce the dimensionality of the raw data matrix $\mathbf{X_0}$ from 3887 to 10. Following the steps 2 and 3 in Figure 3.5, we first normalize the matrix $\mathbf{X_0}$ and then apply SVD. The result is the data matrix $\mathbf{X}_{[\mathbf{1659x10}]}$. This concludes the data preparation stage.

As we discussed in Chapter 3 (Figure 3.5, step 3.3), the SVD decomposition of the normalized raw data matrix $\mathbf{X_0}$ also gives us the principal vectors of a matrix $\mathbf{V}_{[\mathbf{3887x3887}]}$. We note that the data matrix $\mathbf{X}_{[\mathbf{1659x10}]}$ is equivalent to the PCA hidden variables, and the principal vectors at columns of $\mathbf{V}_{[\mathbf{3887x3887}]}$ are the PCA basis vectors. We will compare the topics these PCA basis vectors represent with those of the ICA basis vectors (compute next).

Given the data matrix $\mathbf{X}_{[\mathbf{1659x10}]}$, ICA will compute 10 ICA basis vectors ($\vec{\mathbf{b}}_1$, ..., $\vec{\mathbf{b}}_{10}$) and 10 ICA hidden variables ($\vec{\mathbf{h}}_1$, ..., $\vec{\mathbf{h}}_{10}$). In other words, ICA gives the basis matrix $\mathbf{B}_{[\mathbf{10x10}]}$=[$\vec{\mathbf{b}}_1{}^T$, ..., $\vec{\mathbf{b}}_{10}{}^T$], and the hidden matrix $\mathbf{H}_{[\mathbf{3887x10}]}$=[$\vec{\mathbf{h}}_1$, ..., $\vec{\mathbf{h}}_{10}$].

We note that the topic labels are *not* used during the entire process. The topic labels are only used later when we interpret and justify the basis vectors and hidden variables.

### 4.4.2 Experimental Results

*What do the ICA basis vectors mean? Do the ICA basis vectors pick up the hidden topics in the text stream? Do the hidden variables convey valuable information about the text stream? Are they better than their PCA counterpart?*

In this subsection, we first show that the 10 ICA basis vectors ($\vec{\mathbf{b}}_i$, $i = 1, \ldots, 10$) successfully discover the 10 hidden true topics (Figure 4.1), better than the 10 PCA basis vectors ($\vec{\mathbf{pc}}_i$, $i = 1, \ldots, 10$). We also show that the ICA hidden variables provide a representation of the text windows

| Basis vector | Top 5 terms (sorted by their absolute values) | | | | |
|---|---|---|---|---|---|
| $\vec{\mathbf{b}}_1$ | mckinnei | sergeant | sexual | major | armi |
| $\vec{\mathbf{b}}_2$ | bomb | rudolph | clinic | atlanta | birmingham |
| $\vec{\mathbf{b}}_3$ | winfrei | beef | texa | oprah | cattl |
| $\vec{\mathbf{b}}_4$ | viagra | drug | impot | pill | doctor |
| $\vec{\mathbf{b}}_5$ | zamora | graham | kill | former | jone |
| $\vec{\mathbf{b}}_6$ | medal | olymp | gold | women | game |
| $\vec{\mathbf{b}}_7$ | pope | cuba | castro | cuban | visit |
| $\vec{\mathbf{b}}_8$ | asia | economi | japan | econom | asian |
| $\vec{\mathbf{b}}_9$ | super | bowl | game | team | re |
| $\vec{\mathbf{b}}_{10}$ | peopl | tornado | florida | re | bomb |

**Table 4.2:** Top 5 terms characterizing each ICA basis vector. Each ICA basis vector correctly corresponds to one true topics. Term suffixes were removed during preprocessing.

and clearly indicate the topic of each text window, while the PCA hidden variables contain more noise.

**ICA Basis Vectors: Hidden Topics in the Transcript** The ICA basis vectors can be interpreted as a vocabulary that summarizes the properties of a data set (Figure 3.7 (I2)). For the transcript data set, each ICA basis vector can be considered as a "topic" in the transcript collection.

To interpret the topic that a basis vector captures, we examine the elements of the vector. We look at the elements with the maximum *magnitude* (i.e., absolute value), and associate the meaning of a basis vector with those elements with maximum magnitude (Figure 3.7 (I2)). In the case of transcript text, each vector element corresponds to a English word. We call the words with large magnitude the *representative words* of a basis vector. Mathematically, term $j$ is a representative term of $\vec{\mathbf{b}}_k$, if the value $|b_{k,j}|$ is among the largest in $\vec{\mathbf{b}}_k$.

Table 4.2 shows the top 5 representative terms. Compared to the true topics in Figure 4.1, we found the 10 ICA basis vectors capture all 10 hidden topics, with topic-describing words as the representative terms. Moreover, each ICA basis vector exactly coincides with one topic in the text stream. There is no noise (terms from other topics) among the representative terms of a topic.

| Basis vector | Top 5 terms (sorted by their absolute values) | | | | |
|---|---|---|---|---|---|
| $\vec{\mathbf{pc}}_1$ | mckinnei | bomb | women | sexual | sergeant |
| $\vec{\mathbf{pc}}_2$ | bomb | mckinnei | rudolph | clinic | atlanta |
| $\vec{\mathbf{pc}}_3$ | winfrei | viagra | texa | beef | oprah |
| $\vec{\mathbf{pc}}_4$ | viagra | winfrei | drug | texa | beef |
| $\vec{\mathbf{pc}}_5$ | zamora | viagra | winfrei | graham | olymp |
| $\vec{\mathbf{pc}}_6$ | zamora | graham | kill | viagra | jone |
| $\vec{\mathbf{pc}}_7$ | pope | cuba | medal | olymp | castro |
| $\vec{\mathbf{pc}}_8$ | asia | economi | japan | econom | asian |
| $\vec{\mathbf{pc}}_9$ | bowl | super | re | peopl | medal |
| $\vec{\mathbf{pc}}_{10}$ | peopl | tornado | super | bowl | florida |

**Table 4.3:** Top 5 terms characterizing each PCA basis vector. PCA mixes up the true topics (e.g., $\vec{\mathbf{pc}}_3$ and $\vec{\mathbf{pc}}_4$). Term suffixes were removed during preprocessing.

For example, the representative terms of the ICA basis vector $\vec{\mathbf{b}}_3$ are `winfrei`, `beef`, `texa`, `oprah`, and `cattl`. The terms clearly indicate that the topic is about an issue involving Oprah Winfrey, the cattle industry in Texas, and beef, and exactly corresponds to the true topic ($TP_3$) in Figure 4.1. (The numbering of the basis vectors and topics have been manually aligned, to make clear the correspondence between them.)

For comparison, Table 4.3 shows the top 5 representative terms of the PCA basis vectors. Unlike the clear 1-to-1 correspondence between ICA basis vectors and the hidden topics, many of the PCA basis vectors obviously mix up multiple real topics. For example, topics $TP_1$ and $TP_2$ are apparently mixed up in the PCA basis vectors $\vec{\mathbf{pc}}_1$ and $\vec{\mathbf{pc}}_2$. Also, components $\vec{\mathbf{pc}}_3$, $\vec{\mathbf{pc}}_4$, $\vec{\mathbf{pc}}_5$ and $\vec{\mathbf{pc}}_6$ are mixtures of topics $TP_3$, $TP_4$ and $TP_5$; and, components $\vec{\mathbf{pc}}_9$ and $\vec{\mathbf{pc}}_{10}$ have information from topics $TP_9$ and $TP_{10}$.

Comparing Tables 4.3 and 4.2, we see that ICA is able to capture the hidden topics in the text stream better than PCA. The failure of PCA is probably due to the situation we show in Figure 3.1(b), where basis vectors found by PCA lie between true patterns (topics), and exhibit mixed characteristics from multiple topics.

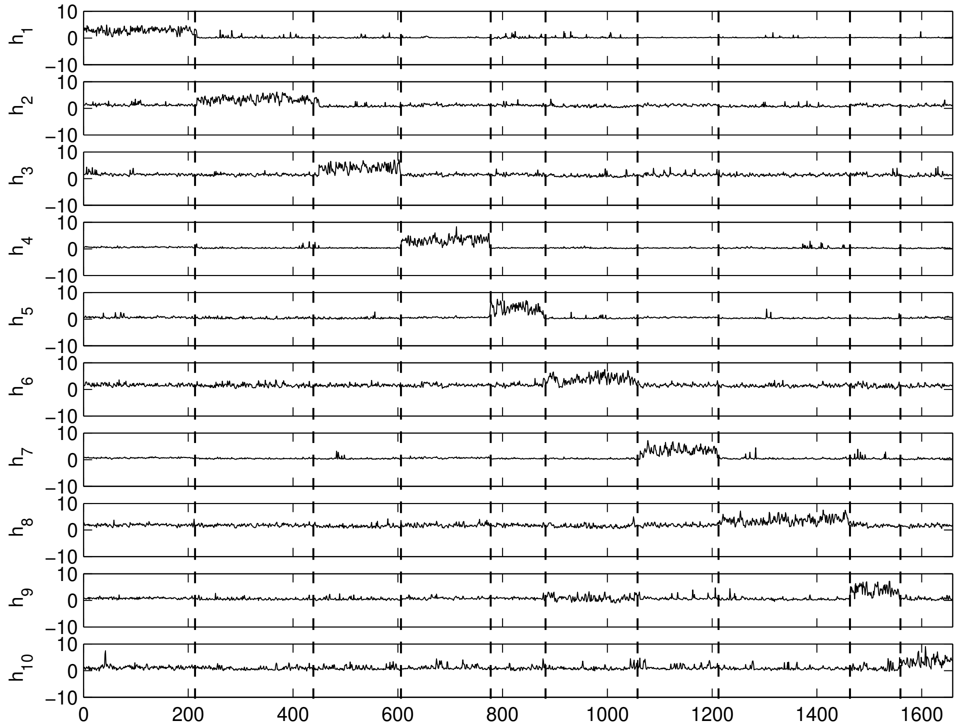



**Figure 4.4:** Representation of the CNN text segments by the ICA hidden variables. Rows: ICA hidden variables. X-axis: Segment ID. The text segments are grouped according to their true topics. From left to right are segments of topics $TP_1$, $TP_2$, and so on. ICA hidden variables is a good indicator of the topic of a segment.

**ICA Hidden Variables: Topic Indicator of Text Segments** If the ICA basis vectors $\mathbf{B}_{[\mathbf{10x10}]}$ correspond to the hidden topics in the text stream, what information do the ICA hidden variables $\mathbf{H}_{[\mathbf{3887x10}]}$ provide? Each row of matrix $\mathbf{H}$ gives the *hidden variable representation* of a text segment (Figure 3.7 (I5)). As we show next, this hidden variable representation is a good indicator of the topic of a text segment, and again, better than what PCA provides. Since the text segments in our experiments are created by windowing (Section 4.1), we will call a text segment a *window* in the rest of this subsection.

Figure 4.4 shows the values of the hidden variables in matrix $\mathbf{H}_{[\mathbf{3887x10}]}$, where the $i$-th waveform in the figure shows the 3887 values of the $i$-th hidden variable $\vec{\mathbf{h}}_i$ ($i = 1, \ldots, 10$). The X-axis is the window ID. Therefore, the hidden variable representation of a window $k$, i.e., $[h_{k,1}, \ldots, h_{k,10}]$

is shown at the column of 10 numbers at location $X = k$.

In Figure 4.4, the windows were sorted according to their true topics. From left to right, the first 60 windows ($X = 1 \ldots 60$) are windows of topic $TP_1$, the next 18 are those of topic $TP_2$, and so on. The vertical broken lines indicate the true boundaries between topics. By doing this window rearrangement, we obtain a better visualization to verify the claim that the ICA hidden variables are a good topic indicator of a window.

As shown in the figure, each window has near-zero values for all hidden variables except one. The exceptional, non-zero hidden variable of a window usually corresponds to the correct topic of the window. Since the windows have been sorted, left to right, from $TP_1$ to $TP_{10}$, the non-zero values in the figure moves from the top ($\vec{\mathbf{h}}_1$) to the bottom ($\vec{\mathbf{h}}_{10}$), forming an overall diagonal pattern in the figure. The sparseness (many near-zero values) of the hidden variable representation is one major characteristic of ICA (Figure 3.7 (I5)), and is related to the ability of ICA basis vectors to capture the exact topic (Table 4.2).

One advantage of this sparse hidden variable representation is that it can be used to identify the topic of a window. That is, the topic of a window is simply the one that corresponds to the non-zero value of the hidden variable. We note that hidden variables $\vec{\mathbf{h}}_6$ and $\vec{\mathbf{h}}_9$ are slightly "confused" in the figure. This is because the two corresponding topics ($TP_6$ about Olympic Games, and $TP_9$ about the Superbowl (Table 4.1)) are both about sports, and share terms such as "game".

In contrast, Figure 4.5 shows the hidden variable representation obtained by PCA. The PCA hidden variables do not provide a topic indicator as good as those of ICA: for a window, there is more than one hidden variable that has non-zero value. This phenomenon is related to the imperfections of PCA basis vectors, where multiple topics are mixed in one PCA basis vector.

In summary, ICA on the text stream gives the following results:

- The ICA basis vectors successfully discover the hidden topics in the text stream, and this is done with no information about the types and properties of the topics. The only information given to ICA is the number of hidden topics ($m$=10) in the text stream.

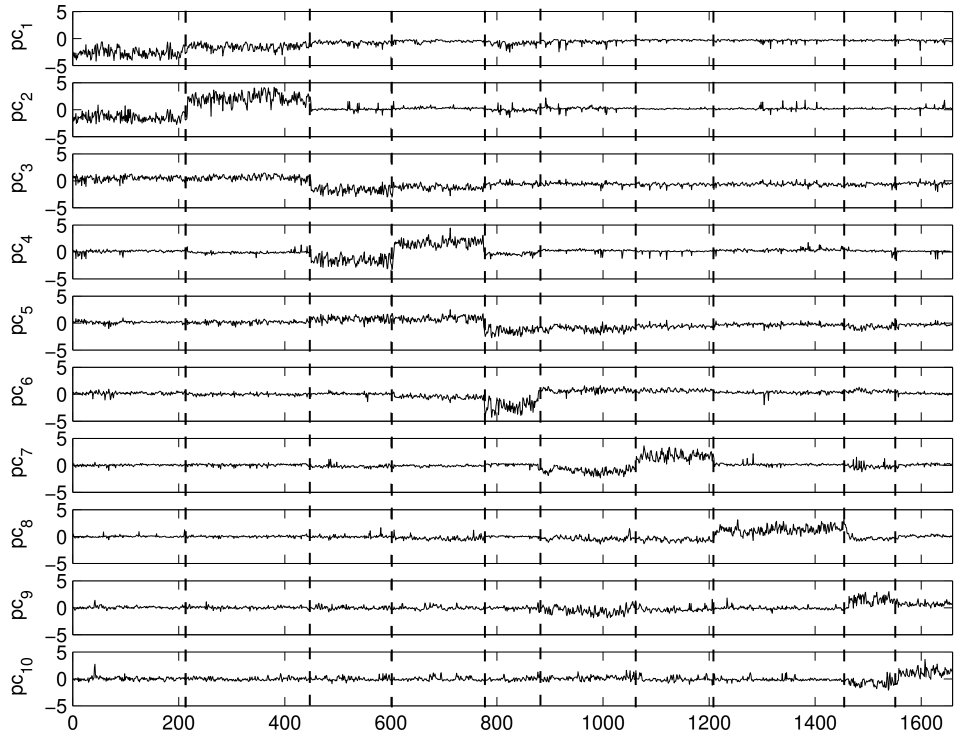



**Figure 4.5:** Representation of the CNN text segments by the PCA hidden variables. Rows: PCA hidden variables. X-axis: Segment ID. The text segments are grouped according to their true topics. From left to right are segments of topics $TP_1$, $TP_2$, and so on. Notice that, for example, $pc_1$ spills over two real topics, $TP_1$ and $TP_2$.

- The ICA hidden variables give a good indicator of the topic of a text segment (window).
- The quality of ICA basis vectors and hidden variables are better than their PCA counterparts.

## 4.5 Classification using VideoCubes and AudioCubes

In sections 4.2 and 4.3, we saw that the two video genres, news and commercial, have different patterns (VideoBasis and AudioBasis) in video frames and in audio. For example, news videos have slower motion in frames and more human speech in audio, while commercials have more scene transitions in frames and music in audio. *How do we exploit the different patterns of various* *video genres for applications like classification?*

In this section, we propose a method to classify news videos and commercials using VideoBasis and AudioBasis. Our classification method is based on the idea of compression: the genre class whose basis compresses a video clip better (smaller representation error) is the class that we will predict for the clip.

### 4.5.1 Proposed Method: Classification by Compression

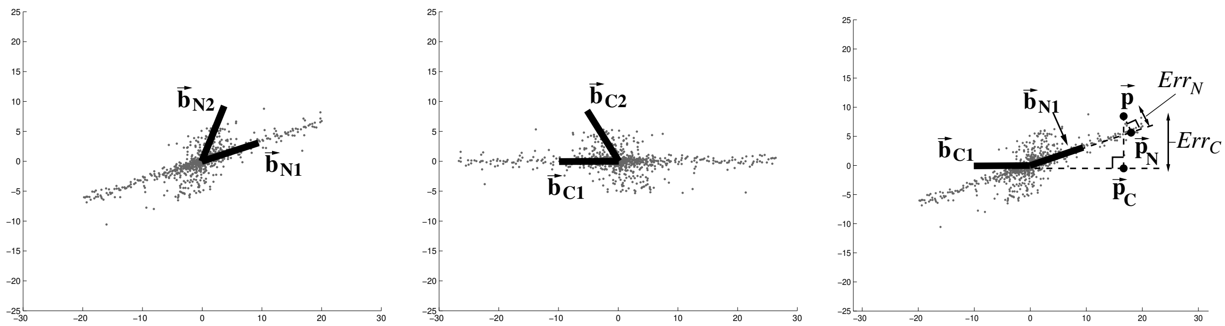
VideoBasis and AudioBasis are both ICA basis vectors in some feature space. In our experiments, VideoBasis contains 160 basis vectors in the $12^3$-dim VideoCube-space (Section 4.2.2), and AudioBasis contains 60 basis vectors in the 2205-dim AudioCube-space (Section 4.3.1). To illustrate our proposed classification method, we will take VideoBasis for example, and explain how classifying video clips is done using VideoBasis vectors.

The 160 VideoBasis vectors define a subspace in the $1,728 (= 12^3)$-dimensional VideoCube-space. VideoBases of different genres define different subspaces in the same VideoCube-space. VideoCubes from a clip of one genre, say "news", will be located in or near the subspace of "news". On the other hand, VideoCubes from commercial clips are located in the subspace of "commercial". By computing the distances of a VideoCube from the genre subspaces and determining in which subspace the VideoCube is, we can predict the most likely genre for the VideoCube.

Figure 4.6 illustrates the idea of our classification approach. Subfigures (a) and (b) show two fictitious data sets representing VideoCubes from news and commercial clips, respectively. In this example, the VideoCube-space is 2-dimensional, and the *complete set* of a VideoBasis will have 2 basis vectors (solid lines in (a) and (b)).

In practice, the dimensionality of VideoCube-space is very high (e.g., $12^3 = 1728$), and we do not keep the complete VideoBasis vectors (e.g., keeping only 160). When the VideoBasis is *under-complete*, there will be *representation error*.

For example, in Figure 4.6(c), we keep only 1 vector from each genre: $\vec{\mathbf{b}}_{N1}$ for "news", $\vec{\mathbf{b}}_{C1}$ for "commercial." Consider a VideoCube $\vec{\mathbf{p}}$, the best representation of $\vec{\mathbf{p}}$ using the VideoBasis vector




(a) News (b) Commercial (c) Classification

**Figure 4.6:** Illustration of the proposed VideoCube classification method. Suppose each VideoCube is a 2-dimensional point: (a) VideoCubes of genre "news", and the VideoBasis vectors: $\vec{\mathbf{b}}_{N1}$, $\vec{\mathbf{b}}_{N2}$. (Most VideoCubes are on vector $\vec{\mathbf{b}}_{N1}$.) (b) VideoCubes of genre "commercial", and the VideoBasis vectors: $\vec{\mathbf{b}}_{C1}$, $\vec{\mathbf{b}}_{C2}$. (Most VideoCubes are on vector $\vec{\mathbf{b}}_{C1}$.) For each genre, we keep only 1 VideoBasis vector: $\vec{\mathbf{b}}_{N1}$ for "news", $\vec{\mathbf{b}}_{C1}$ for "commercial". In (c), the VideoCube $\vec{\mathbf{p}}$ is better represented by $\vec{\mathbf{b}}_{N1}$: the representation error $Err_N < Err_C$). Therefore, $\vec{\mathbf{p}}$ is classified as a VideoCube of "news".

$\vec{\mathbf{b}}_{N1}$ is $\vec{\mathbf{p}}_N$, which is the projection of $\vec{\mathbf{p}}$ onto the subspace of $\vec{\mathbf{b}}_{N1}$. However, the representation $\vec{\mathbf{p}}_N$ is not the same as $\vec{\mathbf{p}}$, and the *representation error* ($Err_N$) is *the distance between* $\vec{\mathbf{p}}$ *and the projection* $\vec{\mathbf{p}}_N$. Similarly, the representation error of the same VideoCube $\vec{\mathbf{p}}$ using the VideoBasis vector of "commercial", $\vec{\mathbf{b}}_{C1}$, is the distance between $\vec{\mathbf{p}}$ and its projection $\vec{\mathbf{p}}_C$ on the subspace of $\vec{\mathbf{b}}_{C1}$, and is denoted as $Err_C$.

In general, we keep more than one VideoBasis vector (e.g., 160) to represent the characteristics of a genre. These VideoBasis vectors are exactly the rows of the raw basis matrix $\mathbf{B_0}$. However, the definition of the representation error of a data point $\vec{\mathbf{p}}$ remains the same, which is *the ($L_2$) distance between* $\vec{\mathbf{p}}$ *and* $\vec{\mathbf{p}}_\perp$*, where* $\vec{\mathbf{p}}_\perp$ *is the projection of* $\vec{\mathbf{p}}$ *in the space spanned by row vectors in* $\mathbf{B_0}$. We will also call the space *the row vector space of* $\mathbf{B_0}$ .

To classify the genre type of a VideoCube $\vec{\mathbf{p}}$, we compute the representation errors of $\vec{\mathbf{p}}$ for each genre, and choose the genre with the smallest representation error as the predicted genre for $\vec{\mathbf{p}}$. Continuing our example in Figure 4.6(c), the representation error of $\vec{\mathbf{p}}$ by the VideoBasis of genre "news", $Err_N$, is smaller than that of genre "commercial", $Err_C$, i.e., $Err_N < Err_C$. Therefore, the VideoCube $\vec{\mathbf{p}}$ is classified as a "news" VideoCube.

**Input:** 1. $\mathbf{B_0}^{(1)}, \ldots, \mathbf{B_0}^{(K)}$: the ICA bases (VideoBases or AudioBases) for the $K$ genres. Each row in the matrices is a (normalized, unit-length) basis vector.
2. $p$: the data point to be classified.

**Output:** 1. $L$: The predicted class label.

**Steps**:

1. For $i = 1, \ldots, K$
   Compute representation error $Err_i$ using Eq. (4.1).
2. $L = argmax_{i=1,\ldots,K} Err_i$.

**Figure 4.7:** Classifying a data sample by ICA basis vectors.

The above description of our classification algorithm uses VideoBasis as an example. However, our classification algorithm is general, and can use ICA basis vectors of any modality (such as the AudioBasis).

Figure 4.7 gives the general classification algorithm, using the ICA basis matrices of the $K$ genres of interest.

**Details on Computing the Representation Error** Let the 1-by-$d$ vector $\vec{\mathbf{p}}$ be the data sample (e.g., a VideoCube) to be classified, and let $\mathbf{B_0}^{(1)}_{[mxd]}, \ldots, \mathbf{B_0}^{(K)}_{[mxd]}$ be the $m$-by-$d$ *raw* ICA basis matrices of the $K$ genres (Eq. (3.10)). In our example of VideoCube classification, the parameter values are: $m = 160$, $d = 12^3 = 1728$, and $K$=2 ("news" and "commercials").

Each row of a *raw* basis matrix $\mathbf{B_0}^{(i)}_{[mxd]}$ ($i$=1,...,$K$) is a $d$-dim basis vector. We use the raw basis vectors because they are in the same $d$-dim space of the data sample $\vec{\mathbf{p}}$. The rows of matrix $\mathbf{B_0}^{(i)}$ defines a *row vector space*, which is a subspace in the $d$-dim space of the data samples. (We do not use the basis matrix $\mathbf{B}^{(i)}_{[mxm]}$, which defines a subspace in the reduced $m$-dim space, different from the $d$-dim data space, where $m < d$.)

The representation error, $Err_i$, for representing a $d$-dimensional data point $\vec{\mathbf{p}}$, using the basis vectors of the $i$-th genre ($\mathbf{B_0}^{(i)}_{[mxd]}$), is defined as *the distance between* $\vec{\mathbf{p}}$ *and* $\vec{\mathbf{p}}_\perp{}^{(i)}$, where $\vec{\mathbf{p}}_\perp{}^{(i)}$ is the projection of $\vec{\mathbf{p}}$ in the row vector space of $\mathbf{B_0}^{(i)}_{[mxd]}$.

To find $\vec{\mathbf{p}}_\perp{}^{(i)}$, we first find an orthogonal basis that spans the same row vector space of $\mathbf{B_0}^{(i)}_{[mxd]}$.

| Symbol | Description |
|---|---|
| $d$ | The dimensionality of the data space. |
| $m$ | The dimensionality of a ICA basis subspace in the $d$-dim data space. |
| $\vec{\mathbf{p}}$ | The 1-by-$d$ vector of a data sample (e.g., VideoCube or AudioCube) |
| $\mathbf{B_0}^{(i)}_{[mxd]}$ | The *raw* ICA basis matrix for the $i$-th genre, where the $m$ rows define a $m$-dim subspace. |
| $\vec{\mathbf{p}}_{\perp}{}^{(i)}$ | A 1-by-$d$ vector; the projection of $\vec{\mathbf{p}}$ in the $m$-dim subspace of $\mathbf{B_0}^{(i)}_{[mxd]}$. |
| $\mathbf{B_0}_{\perp}{}^{(i)}_{[mxd]}$ | The $m$-by-$d$ "orthogonalized" raw ICA basis matrix for the $i$-th genre. |
| $Err_i$ | The representation error of $\vec{\mathbf{p}}$ for the $i$-th genre. |

**Table 4.4:** Summary of symbols for computing the representation error

Notice that the rows of $\mathbf{B_0}^{(i)}_{[mxd]}$ are ICA basis vectors, which are not required to be orthogonal. This "orthogonalization" process can be done by the Gram-Schmidt process, a well-known linear algebra technique. Let $\mathbf{B_0}_{\perp}{}^{(i)}_{[mxd]}$ be the $m$-by-$d$ matrix whose *rows* contain the orthogonal basis vectors from the Gram-Schmidt process.

After we have the orthogonal basis vectors in $\mathbf{B_0}_{\perp}{}^{(i)}_{[mxd]}$, the projection $\vec{\mathbf{p}}_{\perp}{}^{(i)}$ of $\vec{\mathbf{p}}$ onto the space of ICA basis vectors, and the representation error $Err_i$, can be defined as:

$$\vec{\mathbf{p}}^{(i)}_{\perp} = (\mathbf{B_0}_{\perp}{}^{(i)} \vec{\mathbf{p}}^T)^T \, \mathbf{B_0}_{\perp}{}^{(i)}, \tag{4.1}$$

$$Err_i = ||\vec{\mathbf{p}} - \vec{\mathbf{p}}^{(i)}_{\perp}||_2 . \tag{4.2}$$

**Test Data** The video clips used in our experiments are segments from CNN nightly news reports. The training set contains 8 news clips and 6 commercial clips, each about 30-second long. The testing set in our classification experiments contains 62 news clips and 43 commercial clips. Each clip is about 18 seconds long.

In this study, VideoBasis and AudioBasis are used separately for classification. That is, we do two classification tasks: one based on VideoBasis and the other based on AudioBasis.

**Input:** 1. $\mathbf{VB_1^{(k)}}$, ..., $\mathbf{VB_9^{(k)}}$: the VideoBases of the 9 regions of the $k$-th video genre, for $k = 1, \ldots, K$.
In our case, K=2 ("news" and "commercial").
Each row in the matrices is an unit-length basis vector in the $12^3$-dim VideoCube space.
2. $v$: the video clip to be classified.

**Output:** 1. $L$: The predicted class label.

**Steps**:

1. Extract VideoCubes, $c_1, \ldots, c_N$, from video clip $v$.
2. Initialize the vote counters of all classes: `vote_cnt[k]=0`, for all $k = 1, \ldots, K$.
3. For each VideoCube $c_i$ ($i = 1, \ldots, N$)
   3.1 Let $r$ be the region where $c_i$ was extracted.
   3.2 Predict the class label $L_i$ ($1 \le L_i \le K$) of VideoCube $c_i$ using the algorithm in Figure 4.7, according to the VideoBases at region $r$ from the $K$ genres: $\mathbf{VB_r^{(1)}}$, ..., $\mathbf{VB_r^{(K)}}$.
   3.3 Increase the vote count of class $L_i$: `vote_cnt[`$L_i$`]=vote_cnt[`$L_i$`]+1`.
4. $L = argmax_{j=1,\ldots,K}$ `vote_cnt[`$j$`]`.

**Figure 4.8:** Video classification with VideoBases and majority voting.

### 4.5.2 Classification with VideoBasis

In our experiments, in order to handle the complex activities in video scenes, we divided the video frames into 9 regions (arranged in a 3-by-3 configuration), and extracted a VideoBasis from each of the regions. The VideoBasis of each region captures patterns particular to that region, and will be used to classify VideoCubes from the same region in a test video clip. Based on the classification result of all VideoCubes, the class label of the entire clip is determined by majority voting among the VideoCubes.

The VideoBasis of a region was computed based on 10,000 VideoCubes randomly sampled (overlaps were allowed) from that region in the training video clips. In our case, we have 2 genres, "news" and "commercial". The size of a VideoCube is 12-by-12-by-12, and the number of VideoBasis vectors is set at 160. At the end, we have in total 18 sets of VideoBasis vectors: 9 VideoBases for "news" at each region, and the other 9 VideoBases for "commercial".

Mathematically, let the 160-by-1728 matrix $\mathbf{VB_i^{(k)}}$ be the VideoBasis of the $k$-th genre at the $i$-th region, where each row of $\mathbf{VB_i^{(k)}}$ is an unit-length basis vector in the 1728-dimensional VideoCube

| | Predicted genre | | |
|---|---|---|---|
| Truth genre | News | Commercial | Accuracy |
| News | 45 | 17 | 0.73 |
| Commercial | 3 | 40 | 0.93 |

**Table 4.5:** Confusion matrix of Video clip classification using VideoBasis. In total, 62 news and 43 commercial clips are classified, with overall accuracy 81%.

pixel space. Figure 4.8 summarizes our proposed classification algorithm using the normalized VideoBases $\mathbf{VB_i}^{(k)}$ ($i = 1, \ldots, 9$, $k = 1, \ldots, K$). We note that the algorithm is also suitable for other configurations of regions, even though we use the 3-by-3 grid configuration in our experiments.

To classify a test clip, we extracted non-overlapping VideoCubes from each region of the test clip. For a 18-second test clip with 36 I-frames of size 352-by-240 pixels, we extracted 1458 VideoCubes from each region. Each VideoCube was classified as either "news" or "commercial" using the algorithm in Figure 4.7, based on the VideoBases of the same region where the VideoCube is located. The genre of the test clip is determined by majority voting among the VideoCubes. In other words, the genre of a test clip is the genre that is shared the most among the VideoCubes.

Table 4.5 shows the confusion matrix of the news-versus-commercial classification using VideoBasis. The 62 news clips are classified with 73% accuracy, while the 42 commercial clips are classified with 93% accuracy. Commercial clips are classified with higher accuracy than news clips; this is because some news clips contain field coverage that have a lot of background motion and faster transitions, which are similar to the characteristics of commercials. Overall, classification using VideoBasis achieves 81% (=(45+40)/105) accuracy, which is comparable to previous classification results. (For example, using manually-designed visual features, [121] reports 80% classification accuracy on different data sets of 5 video genres.)

| | Predicted genre | | |
|---|---|---|---|
| Truth genre | News | Commercial | Accuracy |
| News | 46 | 16 | 0.74 |
| Commercial | 4 | 39 | 0.91 |

**Table 4.6:** Confusion matrix of Video clip classification using AudioBasis. In total, 62 news and 43 commercial clips are classified, with overall accuracy 81%.

### 4.5.3 Classification with AudioBasis

The proposed classification algorithm in Figure 4.7 also could be used with AudioBases, since AudioBases are also ICA basis vectors (extracted from the audio data in video clips).

The AudioBases of genres "news" and "commercial" are prepared as follow: First, following the steps in Section 4.3, we extracted the AudioBases of genres "news" and "commercial" from the clips in the training set. Similar to the VideoBases in Subsection 4.5.3, the 60 vectors in each AudioBasis were mapped to the 2205-dim AudioCube signal space (Eq. (3.10), and each was normalized to be unit-length. The result, in matrix notation, is the 60-by-2205 matrices $\mathbf{AB}^{(k)}$ of the AudioBasis of the $k$-th genre ($k = 1, 2$), where each row of $\mathbf{AB}^{(k)}$ is an unit-length AudioBasis vector in the 2205-dim AudioCube signal space.

Table 4.6 shows the confusion matrix of the news-versus-commercial classification using AudioBasis. The 62 news clips are classified with 74% accuracy, while the 42 commercial clips are classified with 91% accuracy. Similar to classification using VideoBasis (Table 4.5, commercial clips are classified with higher accuracy than news clips. After examining the video clips, we found that the background non-speech sound in the field reports makes a news clip similar to a commercial in sound, and affects the classification accuracy of news clips. Overall, classification using AudioBasis achieves 81% (=(46+39)/105) accuracy, which is comparable to previous classification results. (For example, on a different data set of 5 video genres, Roach et. al [107] reports $\approx$76% classification accuracy, using the MFCC audio features.)

## 4.6 Summary

To summarize, in this chapter we present the results of applying the AutoSplit framework (Section 4.1) for discovering uni-modal patterns in video clips. We describe a windowing approach to adapt the data in video clips to AutoSplit, and show that AutoSplit is able to find uni-modal patterns from a variety of modalities: the *VideoBasis* from video frames (Section 4.2), *AudioBasis* from audio (Section 4.3), and *hidden topics* from transcript text (Section 4.4). The patterns found are intuitive and consistent with our daily experiences with video clips (news and commercial clips), and are better than those found by the commonly used method, PCA.

The VideoBasis captures the spatial-temporal characteristics of video frames of a video genre. Patterns in the VideoBasis of news clips show clear edge-like shapes in frames, and exhibit slow motion and few scene transitions from frame to frame. On the other hand, patterns in the VideoBasis of commercials display more scene changes and greater motion, but do not have straight edges in the frames. The properties that these VideoBases express generally agree with our experiences with news videos and commercials.

AudioBases of news and commercial clips exhibit the different auditory characteristics of the two genres. The waveforms of the AudioBasis for genre “news” resemble patterns of human speech. On the other hand, waveforms that resemble natural sounds and music (harmonic) are more significant in the AudioBasis for “commercial”. This may be due to the observation that there is more music than speech in commercials, while the sounds in news videos are mostly human monologues and dialogues.

Given the transcript of news videos (as a text stream), AutoSplit is able to find patterns that correspond to the unknown topics in the transcript (text stream). Unlike the more commonly used method PCA, which confuses two true topics, the patterns found by AutoSplit are pure, with one pattern describing exactly one hidden topic.

The patterns found by AutoSplit are useful in many applications. For example, we show that VideoBases and AudioBases are useful in classifying the genre of a video clip, with 81% accuracy in our experiments. Being able to find the hidden topics in a text stream enables a clean (sparse) representation of the content of a text segment, and could be used as an effective indicator of the topic of a segment.

# Chapter 5

# Uni-Modal Patterns of Time Series

Given a collection of co-evolving time sequences, such as the daily closing stock prices of companies in the U.S., or the temperature measurements from sensors deployed on a campus, *how do we discover patterns that, for example, describe the general trend, identify particular events, and detect outliers?*

We would like to have a pattern discovery method that has the following properties:

- It can find meaningful patterns.
- It works in an unsupervised fashion. That is, the method does not require information about the hidden patterns to be found.
- It scales linearly with the data sequence length.

In this chapter, we explore the applications of AutoSplit (Chapter 3, Section 3.3) for finding patterns in co-evolving time sequences. The pattern discovery process of AutoSplit is unsupervised, and no external information about the hidden patterns is needed. Specifically, using AutoSplit, we found that (1) the ICA basis vectors could describe the temporal structure of co-evolving time sequences (e.g., different basic human motions in motion capture data sequences), and (2) the ICA

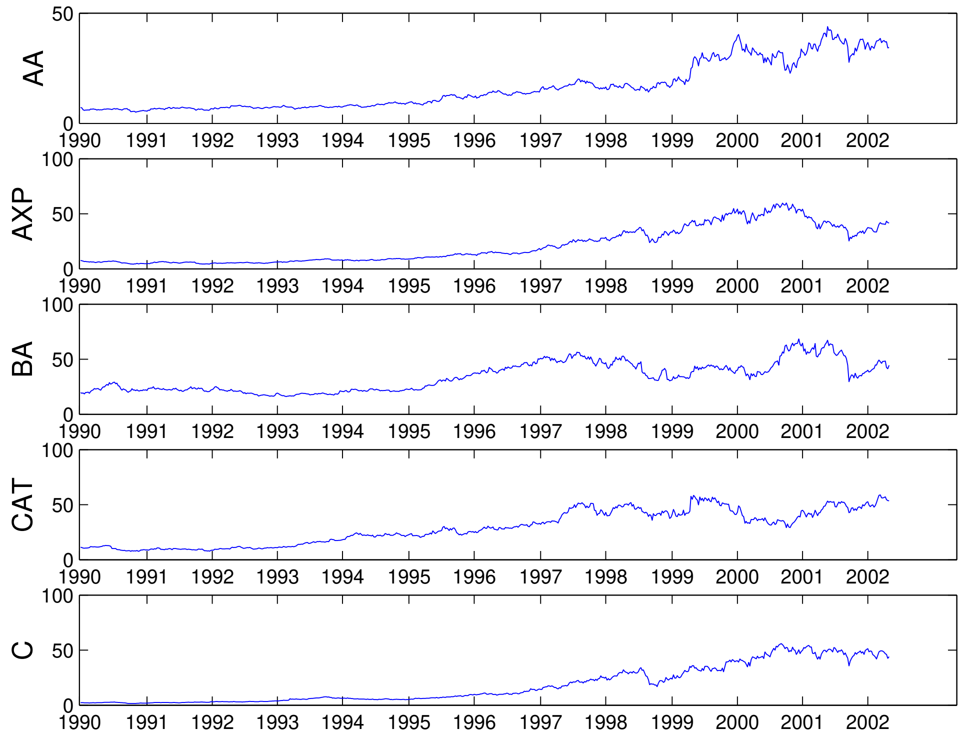



**Figure 5.1:** Example of co-evolving time sequences: weekly closing stock prices. Company names of the stock symbols from top to bottom - AA: Alcoa, AXP: American Express, BA: Boeing, CAT: Caterpillar, C: CitiGroup,

hidden variables are related to the sources that constitute the observed co-evolving sequences (e.g., general trend in stock price sequences).

This chapter is organized as follow: In Section 5.1, we first outline our proposed method of finding patterns in co-evolving time sequences using AutoSplit. Then in Sections 5.2 and 5.3, we discuss our experiments on two sets of co-evolving time sequences: motion capture data and stock price sequences, respectively.

## 5.1 Finding Patterns in Co-evolving Time Sequences

Co-evolving time sequences are sequences of measurements where the measurements are taken at the same time ticks. They are commonly seen in daily life, with examples ranging from the hourly temperatures at cities in a nation, to the weekly closing prices of stocks. Figure 5.1 shows an example of co-evolving time sequences: the weekly closing prices of 5 stocks – Alcoa (AA), American Express (AXP), Boeing (BA), Caterpillar (CAT), and CitiGroup (C) – over a period of 660 weeks. Each sequence contains the closing prices of a stock at 660 weekends from year 1990 to year 2002, and the stock prices at the same time tick are the closing prices of the stock at the same weekend.

Co-evolving time sequences are often correlated. The measurements taken at the same time tick may be affected by the same unknown factors (sources). For example, stock prices of technology/Internet concern companies are all affected by the "Internet bubble" during the years 2000-2001. Moreover, the degree of influence by a source to the stock price differs from company to company, and is also unknown and to be discovered. In addition, measurements taken in one time period may differ from those in another time period. For example, the body measurements while a human is eating are different from those when the person is doing exercise.

*How do we find hidden sources that influence co-evolving time sequences? How do we discover patterns that describe the characteristics of different periods of time sequences?* To address these problems, we propose to use the tool AutoSplit, which has the advantages of finding non-Gaussian patterns (Figure 3.1), as well as on separating unknown sources in observed signals (blind source separation, Figure 3.3). AutoSplit is unsupervised and does not require information about the patterns to be discovered.

Figure 5.2 gives the outline of our proposed algorithm for finding patterns in co-evolving time sequences using AutoSplit. Let $\mathcal{T} = \{T_1, \ldots, T_d\}$ be a set of $d$ co-evolving time sequences, where each time sequence $T_i$ contains measurements $t_{i,j}$ ($j = 1, \ldots, n$) at $n$ time ticks.

The first step of using AutoSplit is to prepare the raw data matrix $\mathbf{X_0}$ and the data matrix $\mathbf{X}$ (Fig-

**Input:** 1. $\mathcal{T}$: A set of co-evolving time sequences
Let $\mathcal{T}$ contains $d$ sequences with measurements at $n$ time ticks.
$T_i$: The $i$-th time sequence ($i = 1, \ldots, d$).
$t_{i,j}$: The measurement at time tick $j$ ($j = 1, \ldots, n$) in sequence $T_i$.
2. $m$: The number of hidden patterns/sources.

**Steps**:
1. Preparing the $n$-by-$d$ raw data matrix $\mathbf{X_0}_{[\mathbf{nxd}]}$.
The $(i,j)$-element $\mathbf{X_0}_{(i,j)} = t_{j,i}$.
(The $i$-th row of matrix $\mathbf{X_0}_{[\mathbf{nxd}]}$ contains the measurements from $d$ sequences at time tick $i$.)
2. (Dimensionality reduction) Obtain the data matrix **X**nm by steps in Figure 3.5.
3. Compute the ICA basis matrix $\mathbf{B}_{[\mathbf{mxm}]}$ and the ICA hidden matrix $\mathbf{H}_{[\mathbf{nxm}]}$, as well as $\mathbf{B_0}_{[\mathbf{mxd}]}$, according to steps in Figure 3.6.
4. Interpretation:
(TS1) **(Time tick characteristics)** The basis vectors at rows of matrix $\mathbf{B_0}$ describe the data characteristics at individual time tick.
(TS2) **(Hidden sources)** Each column of **H** is a hidden source that influences the time sequences in $\mathcal{T}$.
(Note: (TS1) and (TS2) are the (I1) and (I3) interpretations of ICA result in Figure 3.7.)

**Figure 5.2:** Finding patterns in co-evolving time sequences using AutoSplit.

ures 3.4). By considering each time tick as a data point and each time sequence as a data attribute, we can arrange the measurements in the co-evolving sequences $\mathcal{T}$ as a $n$-by-$d$ raw data matrix $\mathbf{X_0}_{[\mathbf{nxd}]}$, where the $i$-th row of matrix $\mathbf{X_0}_{[\mathbf{nxd}]}$ contains the measurements from all $d$ sequences in $\mathcal{T}$ at the $i$-th time tick. In other words, let the $(i,j)$-element of matrix $\mathbf{X_0}$ be $\mathbf{X_0}_{(i,j)}$, then $\mathbf{X_0}_{(i,j)}$=$t_{j,i}$ ($i = 1, \ldots, n$, $j = 1, \ldots, d$).

Given the raw data matrix, $\mathbf{X_0}_{[\mathbf{nxd}]}$, and the number of patterns to extract, $m$, we can obtain the data matrix $\mathbf{X}_{[\mathbf{nxm}]}$, following the preprocessing (dimensionality reduction) procedure in Figure 3.5. Then, following the steps in Figure 3.6, AutoSplit computes the ICA basis matrix $\mathbf{B}_{[\mathbf{mxm}]}$ and the ICA hidden matrix $\mathbf{H}_{[\mathbf{nxm}]}$, as well as the matrix $\mathbf{B_0}_{[\mathbf{mxd}]}$.

AutoSplit interprets the ICA results (Figure 3.7, items (I1) and (I3)), and finds two types of patterns of co-evolving time sequences:

- *(Time tick characteristics)* The basis vectors at rows of matrix $\mathbf{B_0}$ describe the data characteristics at individual time tick.

- *(Hidden sources)* Each column of matrix $\mathbf{H}$ is a hidden source that influences the time sequences in $\mathcal{T}$.

AutoSplit also provides guidelines for examining the physical meanings of these patterns (guidelines at items (I2) and (I4) in Figure 3.7). As we will show in the following sections, AutoSplit is able to find patterns which have meaningful interpretations which are consistent with human intuition/experiences and expert knowledge.

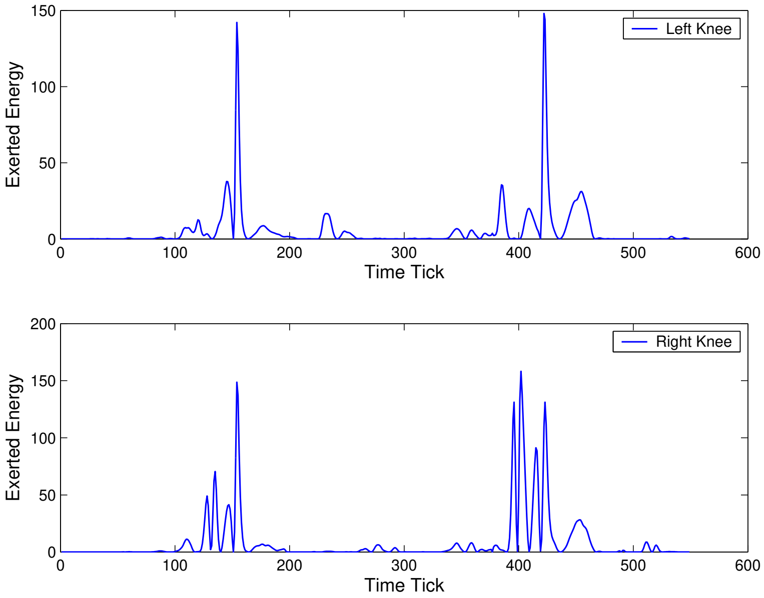
In the rest of this chapter, we will discuss our experimental results on two diverse data sets: the motion capture data (Section 5.2) and the stock price sequences (Section 5.3). Section 5.4 summarizes the chapter.

## 5.2 Experimental Results: Motion Capture Data

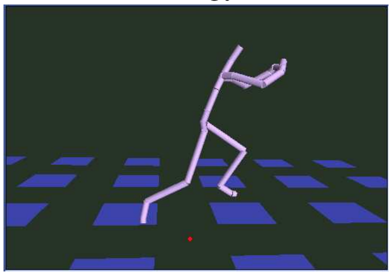
### 5.2.1 The Motion Capture Data: "Broad Jumps"

**Data Description** The "broad jumps" data set from [65] (Figure 5.3) is a *motion capture* data set. In this data set, a human actor was asked to perform two broad jumps during data collection time, and the positions of the two knees are recorded. The recorded position measurements were post-processed, yielding an estimate of the amount of energy exerted at each time tick during the jumps. Figure 5.3(a) shows the energy exerted at each of the two knees at each time tick, and Figures 5.3(b)(c) show synthetic views of the "take-off" and "landing" of one broad jump. In total, measurements are taken at 550 time ticks. The two jumps are performed at time ticks 100 and 380.

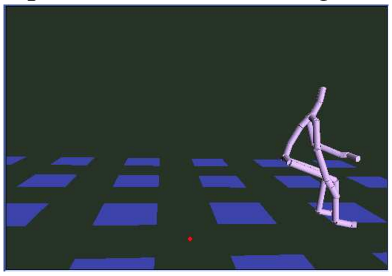
For the "broad jumps" data, we are interested in: (1) discovering the structures of the motion (e.g., taking off, landing), and (2) mining the data for knowledge (rule) discovery. For finding the structure of the broad jumps, we propose to identify different phases ("take-off" or "landing") of a jump by examining the characteristics at each time tick, using AutoSplit's ability on finding "time




(a) Exerted energy versus time tick (top: left knee, bottom: right knee)



(b) Taking off

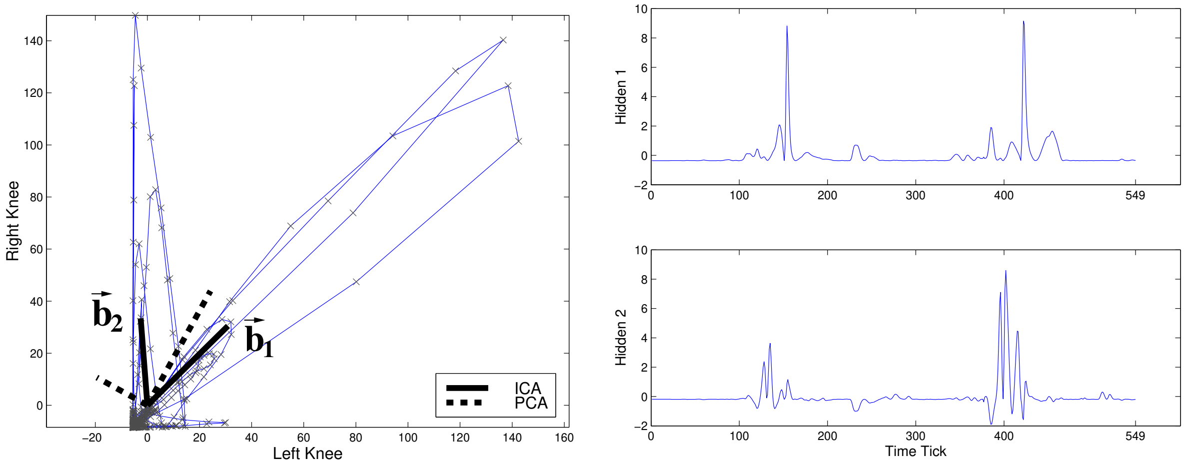


(c) Landing

**Figure 5.3:** (Motion capture data: "Broad jumps") (a) The energy measurements at the two knees while the actor performed two jumps. Two jumps are performed at time ticks 100 and 380. (b)(c): The take-off and landing of a jump.

tick characteristics" (item (TS1) in Figure 5.2). Interpreting the time tick characteristics found by AutoSplit, we could extract information about the broad jumps: For example: *what are the different phases of a broad jump? What peculiarities do we find about the actor who performed the jumps?*

**AutoSplit Analysis** To apply AutoSplit to the broad jump data, we regard the data as a set of two co-evolving time sequences, and conduct experiments following the steps outlined in Figure




(a) (Energy exerted) Right knee vs left knee

(b) Hidden sources $\vec{\mathbf{h}}_1$ and $\vec{\mathbf{h}}_2$

**Figure 5.4:** Pattern discovery in "Broad jumps". (a): The scatter plot of the measurements at each time tick; successive time ticks are connected by a line segment. Dark lines indicate ICA vectors: $\vec{\mathbf{b}}_1$, slope 1:1, corresponds to "landing"; $\vec{\mathbf{b}}_2$, slope 60:(-1), for "take-off". (b) Hidden sources/variables: $\vec{\mathbf{h}}_1$ (top) and $\vec{\mathbf{h}}_2$ (bottom), which correspond to $\vec{\mathbf{b}}_1$ and $\vec{\mathbf{b}}_2$, respectively.

5.2. We formulate the 550 measurements of the left and right knee energy as a $n$-by-$d$ raw data matrix $\mathbf{X}$, with $n$=550 and $d$=2. In other words, the elements at the $i$-th row of $\mathbf{X}$, $[X_{i,1}, X_{i,2}]$, are the energy exerted at the right and left knees, respectively.

For this two-attribute data set, we set the number of patterns to find, $m$, be $m$=2. Actually, two patterns (basis vectors) is also the maximum number of patterns that could be found by the basic ICA method from this 2-dimensional data set. Continuing the steps in Figure 5.2, we compute the basis matrices $\mathbf{B}_{[\mathbf{2x2}]}$ and $\mathbf{B_0}_{[\mathbf{2x2}]}$, as well as the hidden matrix $\mathbf{H}_{[\mathbf{550x2}]}$.

To visualize the patterns in the "broad jumps" data set, we display the scatter plot of the data points in Figure 5.4(a). In the figure, each point corresponds to the measurements at a time tick, and for the point of time tick $i$ ($i$=1,...,550), its location is at $(X_{i,1}, X_{i,2})$, or, informally: (right-knee($i$), left-knee($i$)). Data points at successive time ticks are connected with lines for better visualization, the ICA computation does not use this temporal information. By looking at the figure, we can clearly see that there are two patterns in the motion: one along the $45^o$ degree line, and the other along the Y-axis.

ICA can identify the desired two patterns automatically. The two ICA basis vectors at the rows of matrix $\mathbf{B}_{\mathbf{0}[\mathbf{2x2}]}$ ($\vec{\mathbf{b}}_1$ and $\vec{\mathbf{b}}_2$, solid lines in Figure 5.4(a)) point exactly along the two patterns. While the two truth patterns are not orthogonal, this does not impose difficulty for ICA. Compared to the basis vectors given by PCA (dotted lines in Figure 5.4(a)), PCA misses the truth patterns, due to the orthogonal constraint it has on its basis vectors.

**Interpretation of the ICA Result** We could interpret the information in a basis vector (a row vector of matrix $\mathbf{B_0}$) by examining its components (Item (I2), Figure 3.7). For example, the components of a basis vector give information about how data attributes (energy exerted at the two knees) correlate in the data set. For basis vector $\vec{\mathbf{b}}_1$=[15.51, 14.12], the near 1:1 ratio between the two components indicates a pattern that "same amount of energy is exerted at the two knees". The components of ICA basis vector $\vec{\mathbf{b}}_2$=$[-0.29, 17.65]$ shows an approximate 1:60 ratio on magnitudes of the two components, indicating a pattern that "only the right knee is exerting energy."

To justify the two patterns found by the ICA basis vectors, we investigated the associations between the two patterns and the motions in broad jumps. By playing the synthetic animation and keeping track of the frame-numbers (time ticks), we found that:

- The points along the $\sim 45^o$ degree line ($\vec{\mathbf{b}}_1$) in Figure 5.4(a) are measurements taken during "landing". During landing, the 2 knees exert equal energy (synthesized image at Figure 5.3(c)), which explains the 1:1 slope of $\vec{\mathbf{b}}_1$.

- The points on the near-vertical pattern ($\vec{\mathbf{b}}_2$) are measurements during take-off. During taking off, only the right knee is used (synthesized image at Figure 5.3(b)), which explains the 60:(-1) slope of $\vec{\mathbf{b}}_2$.

Using basis vectors $\vec{\mathbf{b}}_1$ and $\vec{\mathbf{b}}_2$, which correspond to landing and take-off, respectively, we can describe data points very efficiently. Points during the jump take-offs (time ticks 126-137 and 393-415) can be described using only one basis vector ($\vec{\mathbf{b}}_2$) with little error. Similarly, vector $\vec{\mathbf{b}}_1$ can represent the data points during landing (time ticks 150-160 and 420-430) efficiently. This representation of data points using ICA basis vectors is exactly the hidden variables $\vec{\mathbf{h}}_1$ and $\vec{\mathbf{h}}_2$ shown in Figure 5.4(b): during the two take-offs of the broad jump data, $\vec{\mathbf{h}}_1$ is near zero, and $\vec{\mathbf{h}}_2$ has spikes of non-zero values; whereas during the two landings, the situation reverses ($\vec{\mathbf{h}}_1$ is nonzero and $\vec{\mathbf{h}}_2$ is near zero). This kind of representation which has many near-zero values is called a *sparse* representation (Section 3.1.2, and is typical of ICA.

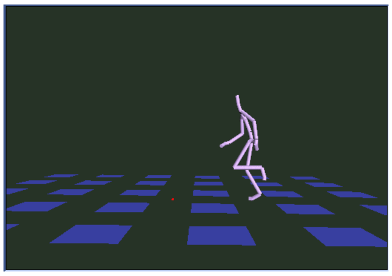
Besides its efficiency, the sparse representation of the time ticks is also useful for a variety of applications. One example of such applications is segmentation, where we want to identify the segment boundaries on which changes of motion occur. In the case of our "broad jumps" data set, let $h_{t,1}$ ($h_{t,2}$) be the values of hidden variable $\vec{\mathbf{h}}_1$ ($\vec{\mathbf{h}}_2$) at time tick $t$. Due to the pronounced sparseness patterns of "take-off" and "landing" that we observed above, we can use a simple metric $f(t) = (h_{t,1} - h_{t,2})$, which gives the difference of the two hidden variables at time $t$, to identify segments of takeoffs, as well as those of landings: for example, $f(t) < 0$ indicates takeoffs, $f(t) > 0$ indicates landing.

The analysis on the patterns of ICA basis vectors also gives us information about the human actor being recorded. Obviously, the actor prefers to use his/her right leg (as opposed to a left-handed human who might prefer the left leg, and a well trained athlete who may use both legs equally). Observations such as this one could help us distinguish human actors (e.g., cluster well-trained from right-handed humans) that are doing the same actions (e.g., broad jumps).

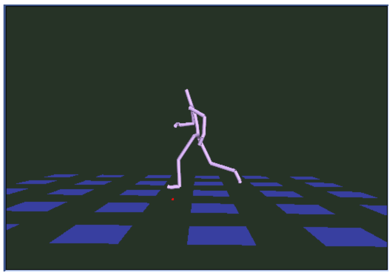
In summary, the ICA basis vectors provide a good "vocabulary" for describing the data characteristics at each time tick (*the time tick characteristics*, item (TS1) in Figure 5.2). The information about the time tick characteristics is useful in various applications, such as segmentation.

### 5.2.2 The Motion Capture Data: "Running"

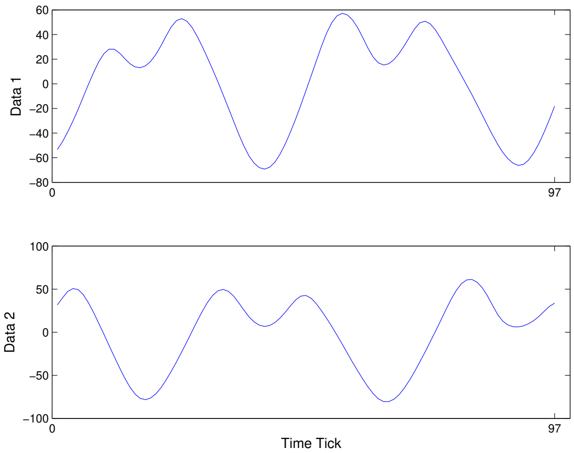
**Data Description** In another experiment, we consider the motion capture data of an actor running a short distance. The measurements collected in this data set, which we called "running", are the angle positions of the left and right knees collected at 97 time ticks. Some sample views of this motion are shown in Figure 5.5(a) and (b), as well as the measurements at each of the two knees



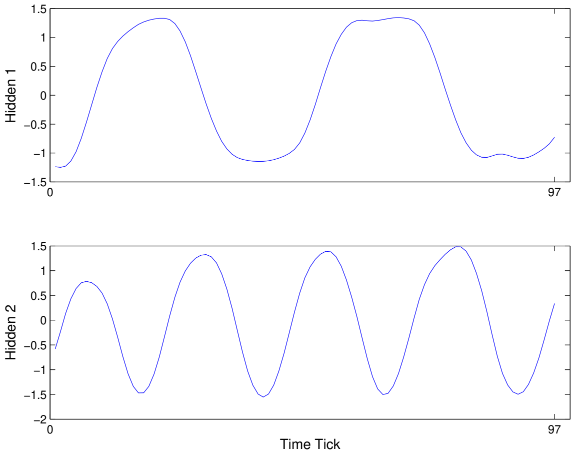
(a) Half-stride



(b) Stride




(c) Measured knee angles




(d) Hidden variables

**Figure 5.5:** The "Running" motion capture data set. The two phases of the motion are (a) "stride" and (b) "half-stride". (c) Measured position of left/right knees (97 time ticks). (d) Hidden variable 1 indicates which leg is leading, hidden variable 2 indicates the phase.

Figure 5.5(c), in the format of time sequences.

For the "running" data set, we are interested in finding the essential elements of the running motion. Watching the animation of the motion, we observed two phases that occur alternatively: "stride", when one leg leads the other (Figure 5.5(a)); and "half-stride", when the relative positions of the two legs switch (Figure 5.5(b)). However, it is difficult to discover such patterns by just looking at the measured values: the two sequences of angle measurements are similar, with one lagging another.

To extract the essential elements of the running motion, we apply ICA and examine the extracted hidden variables. Just as the "cocktail party problem" in Section 3.1.3, where we showed that ICA can recover original signals from noisy mixtures, the goal here is to extract motion elements hidden in the observed measurements, that is, finding the hidden sources of the observed running motion.

**ICA Analysis** The first step of the ICA analysis is creating a raw data matrix $\mathbf{X_0}$. Given the two angle measurements at 97 time ticks, we create a 97-by-2 raw data matrix $\mathbf{X_0}_{[\mathbf{97x2}]}$, where the values at the $i$-th row of $\mathbf{X_0}$ are the two angle measurements at the $i$-th time tick. Following the steps in Figure 5.2, we can compute the hidden matrix $\mathbf{H}_{[\mathbf{97x2}]}$ and the basis matrix $\mathbf{B}_{[\mathbf{2x2}]}$, as well as the matrix $\mathbf{B_0}_{[\mathbf{2x2}]}$.

Figure 5.5(d) shows the time sequences of the two hidden variables. Playing the animation of the motion and cross-referencing with our hidden variables, we find that the first hidden variable indicates which leg is leading (positive when the right leg is leading, and negative for the left leg). The second hidden variable indicates the current phase (positive values for "stride", and negative values for "half-stride").

**Applications** Information from the hidden variables could be useful for motion modeling, synthesis, and prediction. For example, if a motion data sequence has missing values, say, the measurements of the right knee at the last 25% of time ticks are unavailable, and we want to predict these missing measurements based on the first 75% of the data set where data at both knees are available. If we try to do the prediction according to the observed measurements, which are mixtures of unknown sources, we would find difficulties in, for example, the modeling of complex periodic cycles (Figure 5.5(c)).

On the other hand, we can make predictions based on the hidden variables (Figure 5.5(d)), which are sequences with patterns simpler than the original measurements (Figure 5.5(c)). The simpler hidden variables are more likely to be modeled, yielding better predictions. The missing measurements could then be constructed using the predicted values of the hidden variables. By predicting the hidden variables and subsequently the missing measurements, we could simplify the modeling task, and may obtain more accurate predictions.

## 5.3 Experimental Results: Stock Price Sequences

In this section, we show the experimental results of applying ICA to the stock price sequences. Our data set, the **DJIA** data set, contains *weekly* closing prices of 29 companies in the Dow Jones Industrial Average collection. In these co-evolving sequences of stock prices, we are interested in finding patterns such as the hidden sources (variables) that contribute to the ups and downs of the prices. We would like to have patterns that have the following properties:

- **(Meaningful)** The discovered patterns should match human intuition.
- **(Useful)** The discovered patterns should be useful for applications such as outlier detection or clustering.

We propose to use ICA to find patterns that fulfill the two desirable properties above. In particular, we exploit ICA's capability to do blind source separation (Section 3.1.3), to identify the *hidden sources* in the stock price sequences. We will explain how we interpret the possible meaning of the patterns ICA found, and how we make use of the discovered patterns to cluster technical companies from the others, as well as find outliers.

### 5.3.1 The DJIA Stock Price Data Set

The DJIA stock price data set contains the weekly closing prices of 29 companies in the Dow Jones Industrial Average index, starting from the week of January 2, 1990 to that of August 5, 2002. In total, the data set has data at 660 time ticks, and at each time tick, we have the closing prices of the 29 companies. Figure 5.6 plots the stock price sequences of several companies in the DJIA set.

The ups and downs of the stock price may due to a variety of unknown variables, such as the general trend and specific events, including natural phenomena or events in the market. An assumption is that the collective influence of these unknown variables determine the final observed

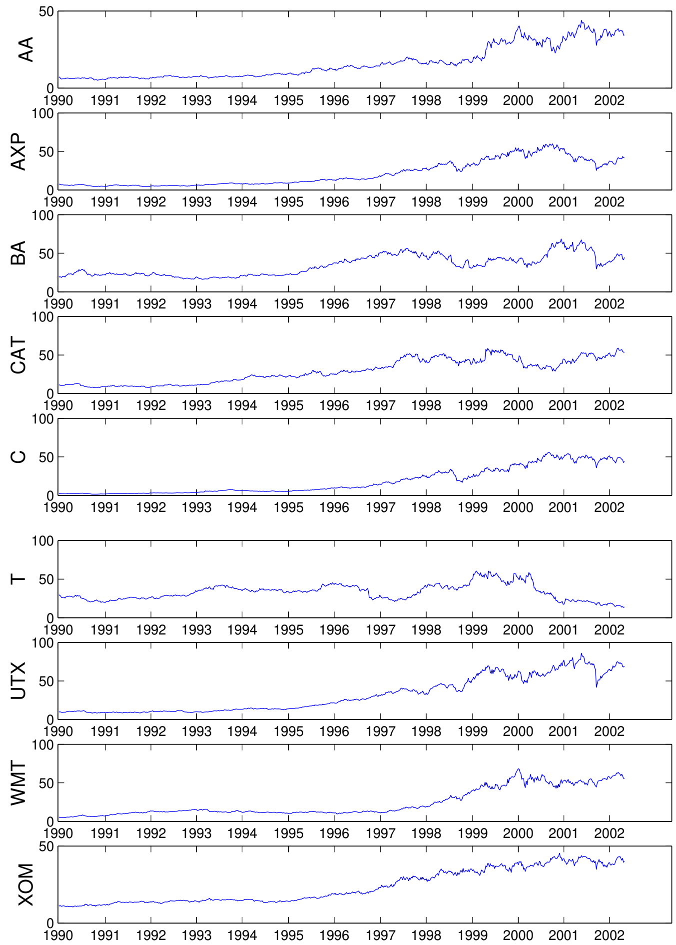



**Figure 5.6:** (DJIA data set) Stock symbols from top to bottom - AA: Alcoa, AXP: American Express, BA: Boeing, CAT: Caterpillar, C: CitiGroup, T: AT&T, UTX: United Technologies, WMT: Wal-Mart, XOM: Exxon Mobil.

stock prices. We are interested in identifying these unknown variables that are hidden behind the observed stock prices.

We consider the problem of finding hidden variables of stock prices as a problem of source separation: the hidden variables are the unknown sources of the observed prices. As discussed in Section 3.1.3, we can use ICA to find these hidden variables.

### 5.3.2 ICA Analysis

To extract the hidden sources using ICA, we first construct a raw data matrix $\mathbf{X_0}_{[\mathbf{660x29}]}$, where the $(i, j)$-element in the matrix $\mathbf{X_0}$ is the stock price of the $j$-th company at the $i$-th time tick. We will follow the steps of our proposed algorithm (in Figure 5.2) to extract patterns from matrix $\mathbf{X_0}$.

Before doing ICA, we preprocess the data matrix $\mathbf{X}_{[\mathbf{660x29}]}$ and make the value of each column (company) zero-mean and unit-variance. This preprocessing step keeps the fluctuation information of the prices, but removes the per-company differences on the absolute stock price.

To extract only the most significant 5 hidden variables, we do dimensionality reduction using singular value decomposition (SVD), reducing the dimensionality from 29 to 5, yielding a 660-by-5 data matrix $\mathbf{X}_{[\mathbf{660x5}]}$. Then, given the data matrix $\mathbf{X}_{[\mathbf{660x5}]}$, ICA computes the hidden matrix $\mathbf{H}_{[\mathbf{660x5}]}$ and the basis matrix $\mathbf{B}_{[\mathbf{5x5}]}$, as well as the matrix $\mathbf{B_0}_{[\mathbf{5x29}]}$, which has the projection of the 5 basis vectors in the original 29-dimensional space (each row is a projected basis vector). This concludes the steps 1-3 outlined in Figure 5.2.

Each column of the hidden matrix $\mathbf{H}_{[\mathbf{660x5}]}$ is a hidden source found by ICA. The 660 values of a hidden variable can be plotted as a time series, showing the behavior of the hidden variable over time. Figure 5.7 plots two hidden variables from the matrix $\mathbf{H}$: hidden variable $\vec{\mathbf{h}}_1$ at the top, and $\vec{\mathbf{h}}_2$ at the bottom. These two hidden variables are the ones that have the strongest "influence" (defined in the following) on the observed data.

### 5.3.3 Patterns in Stock Price Sequences

We would like to interpret the hidden variables of the DJIA stock price sequences. Informally, $\vec{\mathbf{h}}_1$ looks like a upward trend, and $\vec{\mathbf{h}}_2$ seems to correspond to some specific event. However, could we obtain objective evidence to support this interpretation of the two patterns?

In Section 3.3.2, we propose to interpret an ICA hidden variable by examining the components of its corresponding raw basis vector (item (I4) in Figure 3.7). For example, the vector $\vec{\mathbf{b_0}}_i$ at the $i$-th row of the (raw) basis matrix $\mathbf{B_0}$ is the basis vector corresponds to the $i$-th hidden variable $\vec{\mathbf{h}}_i$.

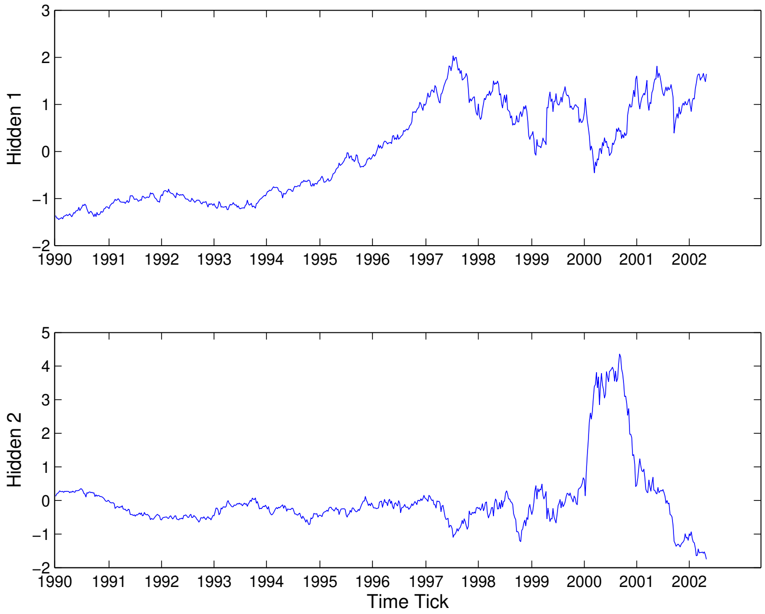



**Figure 5.7:** Hidden variables of the DJIA data set. (Top) Hidden variable $\vec{\mathbf{h}}_1$: probably the general trend of stock prices. (Bottom) Hidden variable $\vec{\mathbf{h}}_2$: probably the "Internet bubble". The values of a hidden variable (a column of the ICA hidden matrix **H**) are plotted as a time sequence (X-axis: time tick, Y-axis: value). These two hidden variables are the ones that have the strongest "influence" on the observed data.

Intuitively, the $j$-th component of a basis vector $\vec{\mathbf{b}}_{\mathbf{0}i}$, $\mathbf{B_0}$(i,j), can be viewed as the "*influence*" of hidden variable $\vec{\mathbf{h}}_i$ to the $j$-th data attribute. In other words, a large $\mathbf{B_0}$(i,j) value indicates that the value of the $j$-th data attribute is more susceptible to the change on the $i$-th hidden variable $\vec{\mathbf{h}}_i$.

In the case of stock price data, the 29 data attributes are the 29 companies. Our proposed method for interpreting the hidden variables of the stock prices becomes: first, identifying companies that are (or are not) influenced by a hidden variable, and then inferring the possible meaning of the hidden variable by examining the common properties of these companies.

We also define the "*overall influence*" to rank the hidden variables by their "significance" to the observed data. The "overall influence", $w_i$, of the $i$-th hidden variable $\vec{\mathbf{h}}_i$ is defined as the sum of the squared influences $\mathbf{B_0}$(i,j), for $j = 1, \ldots, d$. That is,

$$w_i = \sum_{j=1}^{29} \mathbf{B_0}(i,j)^2.$$

| $\mathbf{B_0}$(1,j): Influence of $\vec{\mathbf{h}}_1$ to company $j$ | | | |
|---|---|---|---|
| Highest | | Lowest | |
| CAT | 0.938512 | T | 0.021885 |
| BA | 0.911120 | WMT | 0.624570 |
| MMM | 0.906542 | INTC | 0.638010 |
| KO | 0.903858 | HD | 0.647774 |
| DD | 0.900317 | HWP | 0.658768 |

**Table 5.1:** Top 5 companies that are most influenced and top 5 ones least influenced, by the hidden variable $\vec{\mathbf{h}}_1$. Stock symbols: CAT: Caterpillar, BA: Boeing, MMM: 3M, KO: Kodak, DD: Du Pont, T: AT&T, WMT: Wal-mart, INTC: Intel, HD: Home Depot, HWP: Hewlett-Packard.

In other words, the overall influence of hidden variable $\vec{\mathbf{h}}_i$ is the squared length of its corresponding basis vector $\vec{\mathbf{b}}_i$. Figure 5.7 shows the two hidden variables ($\vec{\mathbf{h}}_1$ and $\vec{\mathbf{h}}_2$) of the DJIA stock price data that have the largest "overall influence".

**Hidden Variable $\vec{\mathbf{h}}_1$: General Trend** Table 5.1 lists the 5 companies that receive the strongest influence from the hidden variable $\vec{\mathbf{h}}_1$, and also the 5 companies receiving the least influence. In other words, the table lists the companies $j$ with the largest and smallest $\mathbf{B_0}$(1,j) values. The hidden variable $\vec{\mathbf{h}}_1$ has strong positive influence on most of the companies, with influence strengths vary from about 0.94 (Caterpillar) to 0.62 (Wal-Mart). The high influence over all companies suggests that $\vec{\mathbf{h}}_1$ represents the general trend of stock price sequences. Comparing the plot of $\vec{\mathbf{h}}_1$ (Figure 5.7 (top)) and the stock price plots in Figure 5.6, we found that $\vec{\mathbf{h}}_1$ indeed shows the general trend of the stock prices of all companies.

The exception is AT&T, which receives small influence, 0.02, from $\vec{\mathbf{h}}_1$, indicating that the company deviates from the general trend. Indeed, in Figure 5.6, it is obvious that AT&T's stock price sequence (6-th plot from the top, stock symbol "T") differs significantly from those of other companies. Thus, we found that AT&T is an outlier whose stock price does not follow the general trend during years 1990-2002.

| $\mathbf{B_0}$(2,j): Influence of $\vec{\mathbf{h}}_2$ to company $j$ | | | |
|---|---|---|---|
| Highest | | Lowest | |
| INTC | 0.641102 | MO | -0.194843 |
| HWP | 0.621159 | IP | -0.089569 |
| GE | 0.509164 | CAT | 0.031678 |
| AXP | 0.504871 | PG | 0.109576 |
| DIS | 0.490529 | DD | 0.133337 |

**Table 5.2:** Top 5 companies that are most influenced and top 5 ones least influenced, by the hidden variable $\vec{\mathbf{h}}_1$. Stock symbols: INTC: Intel, HWP: Hewlett-Packard, GE: General Electric, AXP: American Express, DIS: Disney, MO: Philip Morris, IP: International Paper, CAT: Caterpillar, PG: Procter and Gamble, DD: Du Pont.

**Hidden Variable $\vec{\mathbf{h}}_2$: Internet Bubble** The second hidden variable $\vec{\mathbf{h}}_2$ is shown at the bottom of Figure 5.7). Unlike hidden variable $\vec{\mathbf{h}}_1$, the second hidden variable $\vec{\mathbf{h}}_2$ is mostly silent (i.e., taking on near-zero values), with the exception of a sharp rise and drop in year 2000. This seems to correspond to the event "Internet bubble" which affects technology companies around year 2000. To verify this, we check the influence of $\vec{\mathbf{h}}_2$ on each company, and identify the group of companies that are strongly influenced by this hidden variable.

Table 5.2 lists the 5 companies that receive the strongest influence from the hidden variable $\vec{\mathbf{h}}_2$, and also the 5 companies receiving the least influence. In other words, the table lists the companies $j$ with the largest and smallest $\mathbf{B_0}$(2,j) values. Companies that are greatly influenced by $\vec{\mathbf{h}}_2$ are mainly technology companies, such as Intel (INTC) and Hewlett-Packard (HWP). On the other hand, companies in the goods or manufacturing sector have almost zero response to $\vec{\mathbf{h}}_2$, with examples such as Philip Morris (MO), IP (International Paper), and CAT (Caterpillar). Since the technology companies are more sensitive to $\vec{\mathbf{h}}_2$, we suggest that the hidden variable $\vec{\mathbf{h}}_2$ corresponds to the event "Internet bubble" which largely affected technology companies during years 2000-2001. Interestingly, our analysis also correctly points out that companies related to the media sector, such as General Electric (GE) and Disney (DIS), are also greatly impacted by the "Internet bubble".

To summarize, ICA automatically finds meaningful hidden variables from the DJIA stock price sequences: namely, the general trend and the event "Internet bubble". The meaning of a hidden variable is inferred by analyzing the properties of companies that are (or are not) influenced by the hidden variable, where the "influence" is defined by the elements in the raw basis matrix. By analyzing the influence of hidden variables, we also found outliers and novel rules:

- **(Outlier)** By ranking companies according to the influence they received from the hidden variable "general trend", we could detect outliers, such as AT&T.

- **(Rule Discovery)** Inspecting the influence by the "Internet bubble", we found rules like "besides technology companies, companies involved in media business, such as GE or Disney, were also affected by the Internet bubble."

## 5.4 Summary

In this chapter, we adapted our proposed method AutoSplit (Figure 3.4) to find patterns in co-evolving time sequences. AutoSplit exploits the capabilities of ICA on blind source separation and modeling non-Gaussian patterns to discover meaningful and useful patterns. Specifically, we showed that ICA provides two kinds of patterns in co-evolving sequences, the *time tick characteristics* and the *hidden sources*:

- **(Time Tick Characteristics)** The ICA basis vectors summarize the data characteristics at individual time tick.

- **(Hidden Sources)** The ICA hidden variables can be viewed as hidden sources that affect the fluctuations of the co-evolving sequences.

Moreover, we could interpret the possible meanings of the patterns found, using the guidelines suggested by AutoSplit (Figure 3.7). Our experiments on two data sets, the human motion capture data and the stock price sequence data, showed that the patterns found are *meaningful* – they are consistent with human intuition. For example, we found

- **meaningful time tick characteristics:** the two ICA basis vectors for the "broad jumps" motion correspond to two stages of a jump – "take-off" and "landing" (Section 5.2.1), and
- **meaningful hidden sources:** the ICA hidden variables identify hidden sources such as the "stride" and "half-stride" for the "running" motion capture data (Section 5.2.2, Figure 5.5), or the hidden variables "general trend" and "Internet bubble" for the DJIA stock price sequence data (Section 5.3, Figure 5.7).

The patterns have potential use in applications such as motion segmentation (Section 5.2.2) or rule/outlier discovery. In particular, on the DJIA stock price sequences (Section 5.3), AutoSplit found rules like the following:

- **(Outlier)** AT&T did not follow the general trend during years 1990-2002.
- **(Novel Rule)** Besides technology companies, companies in the media industry were also affected by the "Internet bubble."

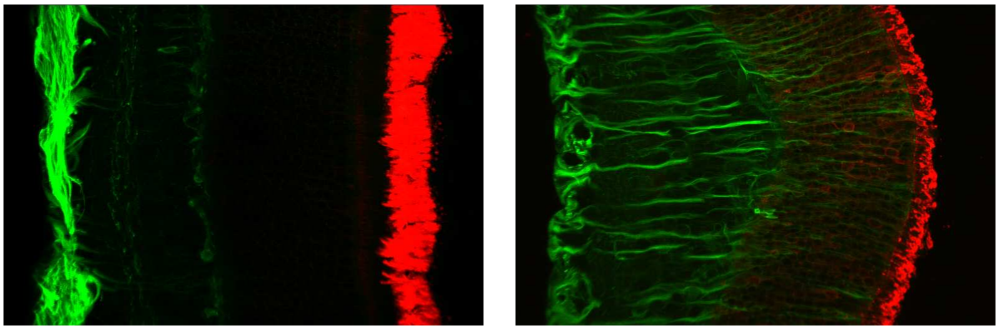
# Chapter 6

# Visual Vocabulary for Biomedical Images

Given a set of medical images of different biological conditions, how do we describe these images? What are the particular properties of an image? What are the differences between two images? How do we extract information from these images?

In this chapter, we describe an automated system, *ViVo*, for mining biomedical images. Using the capability of AutoSplit on constructing a vocabulary for a data set, our system automatically discovers a meaningful *visual vocabulary*, a set of visual "terms" which we called ViVos, for images of retinas in various medical conditions (healthy, diseased, etc.). The ViVos have significant biological meanings, and also enable several data mining tasks such as highlighting the interesting/distinguishing regions in retinal images.

In the following, we will first introduce the problem of mining biomedical images (Section 6.1), followed by a brief overview of the medical background and related work (Section 6.2). Our proposed system, ViVo, is presented in Section 6.3. We evaluate our system in two approaches: quantitatively by classification accuracy (Section 6.4), and qualitatively by consulting with physicians and biologists (Section 6.5). Section 6.6 summarizes our findings in this chapter.



(a) Normal (b) 3d

**Figure 6.1:** Examples of micrographs of (a) a normal retina and (b) a retina after 3 days of detachment. The retinas are labeled with antibodies to rhodopsin (red patterns at the right) and glial fibrillary acidic protein (GFAP, green patterns at the left). Figures are best viewed in color.

## 6.1 Introduction

We focus on the problem of summarizing and discovering patterns in large collections of biomedical images. We would like an automated method for processing the images and constructing a visual vocabulary which is capable of describing the semantics of the image content. Particularly, we are interested in questions such as: *"What are the interesting regions in the image for further detailed investigation?"* and *"What changes occur between images of different medical conditions?"*

As a concrete example, consider the images in Figure 6.1. They depict cross-sections of feline retinas—specifically, showing the distributions of two different proteins—under the experimental conditions "normal" and "3 days of detachment." Even a non-expert human can easily see that each image consists of several vertical layers, despite the fact that the location, texture, and color intensity of the patterns in these layers vary from image to image. A trained biologist can interpret these observations and build hypotheses about the biological processes that cause the differences.

This is exactly the goal of our effort: We want to build a system that will automatically detect and highlight patterns which differentiate image classes. If the problem domain is text documents, we could find differentiate patterns by analyzing the vocabulary words used in documents of different classes. For example, sports articles have a different word usage to that of the academic papers on database technologies. However, for the retinal images that we are interested in, there is no such "visual vocabulary". To approach the goal of detecting and highlighting image patterns, we propose to automatically construct a visual vocabulary, using the ideas from AutoSplit (Chapter 3).

The automatic construction of a visual vocabulary of image patterns is not only important by itself, but also a stepping stone for higher biological goals. Such a system will be of great value to biologists, where it could support valuable functions such as automated classification and various data mining tasks. We illustrate the power of our proposed method on the following three problems:

**Problem 1** *Find a biologically meaningful visual vocabulary for biomedical images.*

**Problem 2** *Identify vocabulary "terms" that distinguish image classes.*

**Problem 3** *Highlight interesting regions in an image.*

Biomedical images bring additional, subtle complications: (1) Some images may not be in the canonical orientation, or there may not be a canonical orientation at all. (The latter is the case for one of our datasets, the Chinese hamster ovary dataset.) (2) Even if we align the images as well as possible, the same areas of the images will not always contain the same kind of tissue because of individual variation. (3) Computer vision techniques such as segmentation require domain-specific tuning to model the intricate texture in the images, and it is not known whether these techniques can spot biologically interesting regions. These are subtle, but important issues that our automatic vocabulary construction system has to tackle.

We would like a system that automatically creates a visual vocabulary and achieves the following goals: (1) *Biological interpretations*: The resulting visual terms should have meaning for a domain expert. (2) *Biological process summarization*: The vocabulary should help describe the underlying biological process. (3) *Generality*: It should work on multiple image sets, either color or gray-scale, from different biological domains.

The major contributions of this chapter are as follows:

- We introduce the idea of "tiles" for visual term generation, and successfully bypass issues such as image orientation and registration.

- We propose a novel approach to group tiles into visual terms, avoiding subtle problems, like non-Gaussianity, that hurt other clustering and dimensionality reduction methods. We call our automatically extracted visual terms "**ViVos**."

This chapter is organized as follows. Section 6.2 describes the related work. In Section 6.3, we introduce our proposed method for visual vocabulary construction, as well as several data mining functions that are enabled by our visual vocabulary. Quantitative evaluation (via classification) of our visual vocabulary is presented in Section 6.4. Experiments illustrating the biological interpretation of ViVos, as well as the proposed data mining functions based on ViVos, appear in Section 6.5. Section 6.6 summarizes our findings in this chapter.

## 6.2 Background and Related Work

Automated analysis tools for biomedical image collections have the potential for changing the way in which biological images are used to answer biological questions, either for high-throughput identification of abnormal samples or for early disease detection [44, 90, 91]. Two specific kinds of biomedical images are studied in this chapter: confocal microscopy images of retina and fluorescence microscopy images of Chinese Hamster Ovary (CHO) cells.

The retina contains neurons that respond to light and transmit electrical signals to the brain via the optic nerve. Multiple antibodies are used to localize the expression of specific proteins in retinal cells and layers. The antibodies are visualized by immunohistochemistry, using a confocal microscope. The images can be used to follow a change in the distribution of a specific protein in different experimental conditions, or visualize specific cells across these conditions. Multiple proteins can be visualized in a single image, with each protein represented by a different color.

It is of biological interest to understand how a protein changes expression and how the morphology of a specific cell type changes across different experimental conditions (e.g., an injury such as retinal detachment) or when different treatments are used (e.g., oxygen administration). The ability to discriminate and classify on the basis of patterns (e.g., the intensity of antibody staining and texture produced by this staining) can help identify differences and similarities of various cellular processes.

The second kind of data in our study are fluorescence microscopy images of subcellular structures of CHO cells. These images show the localization of four proteins and the cell DNA within the cellular compartments. This information may be used to determine the functions of expressed proteins, which remains one of the challenges of modern biology [13].

### 6.2.1 Visual Vocabulary

A textual vocabulary consists of words that have distinct meanings and serve as building blocks of larger semantic constructs like sentences or paragraphs. To create an equivalent visual vocabulary for images, previous work applied transformation on image pixels to derive tokens that can describe image contents effectively [122, 23]. However, an image usually has tens of thousands of pixels. Due to this high dimensionality, a large number of training images is needed by pixel-based methods to obtain a meaningful vocabulary. This has limited the application of these methods to databases of small images.

One way to deal with this dimensionality curse is to extract a small number of features from image pixels. The vocabulary construction algorithm is then applied to the extracted features to discover descriptive tokens. A feature is usually extracted by filtering and summarizing pixel information. In many applications, these tokens have been shown useful in capturing and conveying image properties, under different names such as "blob," "visterm," "visual keywords," and so on. Examples of applications include object detection [104] and retrieval [115], as well as image classification [122, 23, 76] and captioning [25, 53].

Clustering algorithms or transformation-based methods are other techniques employed against the curse of dimensionality. K-means clustering has been applied to image segments [25, 53] and the salient descriptor [115] for vocabulary construction. Examples of transformation-based methods include principal component analysis (PCA) [55, 122, 76] and wavelet transforms [104]. Recently, independent component analysis (ICA) [48] has been used in face recognition [23], yielding facial templates. Like the feature extraction approaches, these methods also have problems with orientation and registration issues, as they rely on global image features.

In this chapter, we present a method that discovers a meaningful vocabulary from biomedical images. The proposed method is based on "tiles" of an image, and successfully avoids issues such as registration and dimensionality curse. We use the standard MPEG-7 features *color structure descriptor* (CSD), *color layout descriptor* (CLD) and *homogeneous texture descriptor* (HTD) [83]. The CSD is an $n$-dimensional color histogram ($n$ is 256, 128, 64, or 32), but it also takes into account the local spatial structure of the color. For each position of a sliding structural element, if a color is present, its corresponding bin is incremented. The CLD is a compact representation of the overall spatial layout of the colors, and uses the discrete cosine transform to extract periodic spatial characteristics in blocks of an image. characterizes region texture using mean energy and energy deviation of the whole image, both in pixel space and in frequency space (Gabor functions along 6 orientations and 5 scales).

Alternatively, there is work on constructing visual vocabulary [88, 82] with a human in the loop, with the goal of constructing a vocabulary that better captures human perception. Human experts are either asked to identify criteria that they used to classify different images [88], or directly give labels to different patterns [82]. The vocabulary is then generated according to the given criteria and labels. These approaches are supervised, with human feedback as input to the construction algorithms. In contrast, our proposed method presented is unsupervised: The image labels are used only after the ViVos are constructed, when we evaluate them using classification.

**Input:** A set of $n$ images $I = \{I_1, \ldots, I_n\}$.
**Output:** Visual vocabulary (ViVos) $\mathcal{V} = \{v_1, \ldots, v_m\}$.
ViVo-vectors of the $n$ images $\{v(I_1), \ldots, v(I_n)\}$.
**Algorithm:**
1. Partition each image $I_i$ into $s_i$ non-overlapping tiles
2. For each tile $j \in \{1, \ldots, s_i\}$ in each image $I_i$, extract $\vec{\mathbf{t}}_{i,j}$
3. Generate visual vocabulary $\mathcal{V} = gen\_vv(\cup_{i=1}^{n} \vec{\mathbf{t}}_{i,j})$
   Also, compute $\mathbf{P}$, the PCA basis for all $\vec{\mathbf{t}}_{i,j}$'s.
4. For each tile $j \in \{1, \ldots, s_i\}$ in each image $I_i$,
   compute the ViVo-vector $v(\vec{\mathbf{t}}_{i,j}) = comp\_vivo(\vec{\mathbf{t}}_{i,j}, \mathcal{V}, \mathbf{P})$
5. For each image $I_i$, compute the ViVo-vector of $I_i$:
   $v(I_i) = summarize(\{\vec{\mathbf{t}}_{i,j} : j = 1, \ldots, s_i\})$

**Figure 6.2:** Algorithm for constructing a visual vocabulary from a set of images.

## 6.3 Visual Vocabulary Construction

In this section, we introduce our proposed method for transforming images into their symbolic representations. The algorithm is given in Figure 6.2, and uses the symbols listed in Table 6.1. The algorithm consists of five steps.

The first step is to partition the images into non-overlapping tiles. The optimal tile size depends on the nature of the images. The tiles must be large enough to capture the characteristic textures of the images. On the other hand, they cannot be too large. For instance, in order to recognize the red layer in Figure 6.1(a), the tile size should not be much larger than the width of the layer. We use a tile size of 64-by-64 pixels, so each retinal image has $8 \times 12$ tiles, and each subcellular protein localization image has $8 \times 6$ or $8 \times 8$ tiles.

In the second step, a feature vector is extracted from each tile, representing its image content. We have conducted experiments using features such as the color structure descriptor (CSD), color layout descriptor (CLD), and homogeneous texture descriptor (HTD). The vector representing a tile using features of, say CSD, is called a *tile-vector* of the CSD. More details are given in Section 6.4.

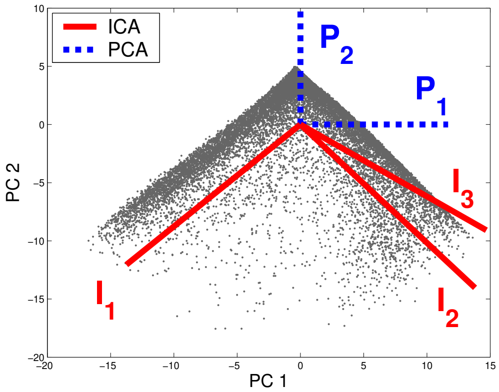
The third step derives a set of symbols from the feature vectors of all the tiles of all the images.

| Symbol | Meaning |
|---|---|
| $\mathcal{V}$ | Set of $m$ ViVos: $\mathcal{V}=\{v_1,\ldots,v_m\}$ |
| $m'$ | Number of ICA basis vectors $m' = m/2$ |
| $\vec{\mathbf{t}}_{i,j}$ | $j$-th tile (or, tile-vector) of image $I_i$ |
| $v(\vec{\mathbf{t}}_{i,j})$ | $m$-dimensional ViVo-vector of tile $\vec{\mathbf{t}}_{i,j}$ |
| $v(I_i)$ | $m$-dimensional ViVo-vector of image $I_i$ |
| $f_k$ | The $k$-th element of $v(\vec{\mathbf{t}}_{i,j})$ |
| $v_k(I_i)$ | The $k$-th element of $v(I_i)$ |
| $c(I)$ | Condition of an image $I$ |
| $\mathcal{S}_{i,k}$ | Set of $\{v_k(I) \mid \forall I, c(I) = c_i\}$ for a condition $c_i$ |
| $\mathcal{T}(v_i)$ | Set of *representative tiles* of ViVo $v_i$ |
| $\mathcal{R}(c_i)$ | Set of *representative ViVos* of condition $c_i$ |

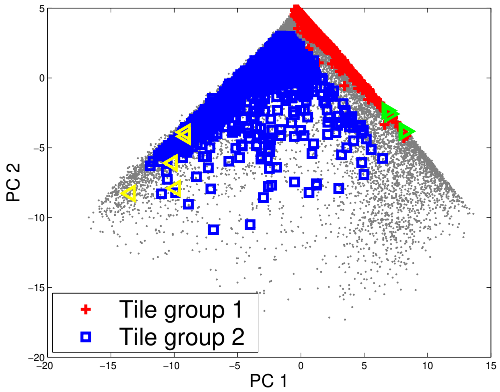
**Table 6.1:** Symbols used in this chapter.

In text processing, there is a similar issue of representing documents by topics. The most popular method for finding text topics is latent semantic indexing (LSI) [20], which is based on analysis that resembles PCA. Given a set of data points, LSI/PCA finds a set of orthogonal (basis) vectors that best describe the data distribution with respect to minimized $L_2$ projection error. Each of these basis vectors is considered a topic in the document set, and can be used to group documents by topics. Our approach is similar: We derive a set of symbols by applying ICA or PCA to the feature vectors. Each basis vector found by ICA or PCA becomes a symbol. We called each such symbol a "***ViVo***", and the set of symbols a *visual vocabulary*.

Figure 6.3(a) shows the distribution of the tile-vectors of the CSD, projected in the space spanned by the two PCA basis vectors with the highest eigenvalues. The data distribution displays several characteristic patterns—"arms"—on which points are located. None of the PCA basis vectors (dashed lines anchored at $\langle 0,0 \rangle$: $\mathbf{P}_1, \mathbf{P}_2$) finds these characteristic arms. On the other hand, if we project the ICA basis vectors onto this space (solid lines: $\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3$), they clearly capture the patterns in our data. It is preferable to use the ICA basis vectors as symbols because they represent more precisely the different aspects of the data. We note that only three ICA basis vectors are shown because the rest of them are approximately orthogonal to the space displayed.




(a) PCA vs ICA




(b) ViVo tile groups

**Figure 6.3:** ViVos and their tile groups. Each point corresponds to a tile. (a) Basis vectors ($\mathbf{P}_1, \mathbf{P}_2, \mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3$) are scaled for visualization. (b) Two tiles groups are shown here. Representative tiles of the two groups are shown in triangles. (Figures look best in color.)

Relating Figure 6.3 to our algorithm in Figure 6.2, each point is a $\vec{\mathbf{t}}_{i,j}$ in step 2 of the algorithm. Function *gen_vv()* in step 3 computes the visual vocabulary which is defined according to the set of the ICA basis vectors. Intuitively, an ICA basis vector defines two ViVos, one along the positive direction of the vector, another along the negative direction.

Formally, let $\mathbf{T_0}$ be a $t$-by-$d$ matrix, where $t$ is the number of tiles from all training images, and $d$ is the number of features extracted from each tile. Each row of $\mathbf{T_0}$ corresponds to a tile-vector $\vec{\mathbf{t}}_{i,j}$, with the overall mean subtracted. Suppose we want to generate $m$ ViVos. We first reduce the dimensionality of $\mathbf{T_0}$ from $d$ to $m' = m/2$, using PCA, yielding a $t$-by-$m'$ matrix $\mathbf{T}$. Next, ICA is applied in order to decompose $\mathbf{T}$ into two matrices $\mathbf{H}_{[\mathbf{txm'}]}$ and $\mathbf{B}_{[\mathbf{m'xm'}]}$ such that $\mathbf{T} = \mathbf{HB}$. The rows of $\mathbf{B}$ are the ICA basis vectors (solid lines in Figure 6.3(a)). Considering the positive and negative directions of each basis vector, the $m'$ ICA basis vectors would define $m = 2m'$ ViVos, which are the outputs of the function *gen_vv()*.

How do we determine the number of ViVos? We follow the rule of thumb [33], and make $m' = m/2$ be the dimensionality which preserves 95 % spread/energy of the distribution.

With the ViVos ready, we can use them to represent an image. We first represent each $d$-dim tile-vector in terms of ViVos by projecting a tile-vector to the $m'$-dim PCA space and then to the

$m'$-dim ICA space. The positive and negative projection coefficients are then considered separately, yielding the $2m'$-dim ViVo-vector of a tile. This is done by *comp_vivo()* in the fourth step of the algorithm in Figure 6.2. The $m = 2m'$ coefficients in the ViVo-vector of a tile also indicate the contributions of each of the $m$ ViVos to the tile.

In the fifth and final step, each image is expressed as a combination of its (reformulated) tiles. We do this by simply adding up the ViVo-vectors of the tiles in an image. This yields a good description of the entire image because ICA produces ViVos that do not "interfere" with each other. That is, ICA makes the columns of **H** (coefficients of the basis vectors, equivalently, contribution of each ViVo to the image content) as independent as possible [48]. Definition 2 summarizes the outputs of our proposed method.

**Definition 2 (ViVo and ViVo-vector)** *A ViVo is defined by either the positive or the negative direction of an ICA basis vector, and represents a characteristic pattern in image tiles. The ViVo-vector of a tile $\vec{\mathbf{t}}_{i,j}$ is a vector $v(\vec{\mathbf{t}}_{i,j}) = f_1, \ldots, f_m]$, where $f_i$ indicates the contributions of the i-th ViVo in describing the tile. The ViVo-vector of an image is defined as the sum of the ViVo-vectors of all its tiles.*

**Representative Tiles of a ViVo** A ViVo corresponds to a direction defined by a basis vector, and is not exactly equal to any of the original tiles. In order to visualize a ViVo, we represent it by a tile that strongly expresses the characteristics of that ViVo.

We first group tiles that are mainly located along the same ViVo direction together as a "tile group". Formally, let the ViVo-vector of a tile $\vec{\mathbf{t}}_{i,j}$ be $v(\vec{\mathbf{t}}_{i,j})$=$[f_1, \ldots, f_m]$. We say that the tile $\vec{\mathbf{t}}_{i,j}$ *belongs to* ViVo $v_k$, if the element with largest magnitude is $f_k$, i.e., $k = \arg\max_{k'} |f_{k'}|$. The *tile group* of a ViVo $v_k$ is the set of tiles that belong to $v_k$. Figure 6.3(b) visualizes the tile groups of two ViVos on the 2-D plane defined by the PCA basis vectors $(\mathbf{P}_1, \mathbf{P}_2)$.

The *representative tiles* of a ViVo $v_k$, $\mathcal{T}(v_k)$, are then selected from its tile group (essentially the tiles at the "tip" of the tile group). The top 5 representative tiles of the two ViVos in Figure 6.3(b) are shown in light triangles. The top representative tile of ViVo $v_k$ has the maximum $|c_k|$ value

| Feature | Dim. | Accuracy | Std. dev. |
|---|---|---|---|
| Original CSD | 512 | 0.838 | 0.044 |
| 14 ViVos from CSD | 14 | 0.832 | 0.042 |
| 12 ViVos from CSD | 12 | 0.826 | 0.038 |
| Original CLD | 24 | 0.346 | 0.049 |
| 24 ViVos from CLD | 24 | 0.634 | 0.023 |
| Original HTD | 124 | 0.758 | 0.048 |
| 12 ViVos from HTD | 12 | 0.782 | 0.019 |

**Table 6.2:** Classification accuracies for combinations of feature and ViVo set size. All ViVo sets reported here are based on ICA.

among all tiles in $v_k$'s tile group. In Section 6.5.1, we show the representative tiles of our ViVos and discuss their biological interpretation.

## 6.4 Quantitative Evaluation: Classification

The experiments in this section evaluate the different combinations of image features and vocabulary sizes for ViVo construction. We want to find the best representation of the images in the symbolic space and ensure that classification accuracies obtained using these symbols are close to the best accuracy that we could obtain with the raw features. However, the overall goal is to use visual vocabulary for data mining tasks (Section 6.5), not solely for classification.

Biologists have chosen experimental conditions which correspond to different stages of the biological process. Thus, a combination that successfully classifies images is also likely to be a good choice for other analyses, such as the qualitative analyses described in Section 6.5, where we investigate the ability of the visual vocabulary to reveal biologically meaningful patterns.

Classification experiments are performed on two datasets: one dataset contains $n$=433 retinal micrographs, and another dataset has $n$=327 fluorescence micrographs showing subcellular localization patterns of proteins in CHO cells. In the following, we refer to the datasets by their cardinality: the 433 dataset and the 327 dataset.

### 6.4.1 Classification of Retinal Images

The 433 dataset contains retinal images from the UCSB BioImage database[1]. The data set contains images of feline retinas detached for either 1 day (label `1d`), 3 days (`3d`), 7 days (`7d`), 28 days (`28d`), or 3 months (`3m`). There are also images of retinas after some treatment, such as reattached for 3 days after 1 hour of detachment (`1h3dr`), reattached for 28 days after 3 days of detachment (`3d28dr`), or exposed to 70 % oxygen for 6 days after 1 day of detachment (`1d6dO2`), and images of control tissues (`n`) [32, 71, 72].

We experiment extensively with different features and vocabulary sizes. Features are extracted separately for the red and green channels and then concatenated. The channels show the staining by two antibodies: anti-rod opsin (red) and anti-GFAP (green). For constructing a visual vocabulary, the number of ViVos should be small, as large vocabularies contain redundant terms and become difficult for domain experts to interpret. We follow the "95 % energy" rule [33] and extract three sets of ViVos, one from each of the three different image features: CSD, CLD, and HTD. Preserving 95 % of the energy results in 14, 24, and 12 ViVos for the features CSD, CLD, and HTD, respectively. We examine these different sets of ViVos and select the set which (a) provides high classification accuracy, and (b) are biological meaningful.

Each set of ViVos gives a different ViVo-vector representation (Definition 2) for the retinal images. For each ViVo-vector representation, we compute the classification accuracy using 5-fold cross-validation using a SVM [15] with a linear kernel. Table 6.2 reports the classification accuracies using the ViVos extracted from different features. We also try other classifiers, such as the SVM with a polynomial kernel or the $k$-NN ($k = 1$, 3, or 5) classifier, which produce results that are not significantly different. ViVos from CSD perform significantly better than ViVos from CLD ($p < 0.0001$) and also significantly better than ViVos from HTD ($p = 0.0492$). Furthermore, manual inspection of HTD ViVos does not reveal better biological interpretations. Therefore, we choose to construct the visual vocabulary from the CSD features.

[1] http://bioimage.ucsb.edu/

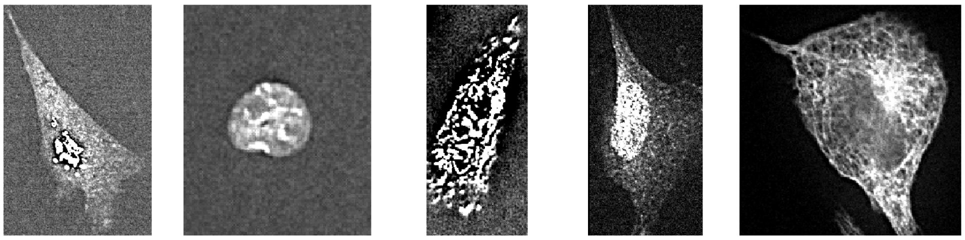
| | Predicted | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Truth | n | 1d | 3d | 7d | 28d | 3m | 3d28dr | 1d6dO2 | 1h3dr |
| n | 13 | 1 | 1 | 1 | - | 1 | - | - | - |
| 1d | 2 | 17 | - | - | 1 | - | - | - | 5 |
| 3d | - | - | 25 | - | 2 | - | - | 1 | - |
| 7d | - | - | 3 | 11 | 2 | - | - | - | - |
| 28d | 1 | - | 3 | - | 31 | - | 2 | 1 | - |
| 3m | - | - | - | - | 1 | 8 | 1 | - | - |
| 3d28dr | - | - | 1 | - | 4 | 1 | 36 | 4 | - |
| 1d6dO2 | - | - | 1 | 3 | 1 | - | - | 17 | - |
| 1h3dr | - | 2 | - | - | - | - | - | - | 10 |

**Table 6.3:** Confusion matrix for the classification experiments on the 433 dataset using an 1-NN classifier. The images are represented by 12 ViVos constructed from the color structure (CSD) features. Overall classification accuracy is 79 %.

We further refine the 14 CSD ViVos by discarding ones that are not significantly expressed in the images. Two of the 14 CSD ViVos are removed because none of the images has high coefficients for them. Manual inspection shows that those two ViVos have no interesting biological interpretation either. As expected, removing these two ViVos (using only 12 ViVos) results in insignificantly ($p = 0.8187$) smaller classification accuracy compared to the 14 CSD ViVos (Table 6.2). The difference from the original CSD features is also insignificant ($p = 0.6567$). We therefore choose to use the 12 CSD ViVos as our visual vocabulary.

To understand the performance of the 12 CSD ViVos on classifying the 9 image classes, we conduct a classification experiment with a 1-NN classifier and show the confusion matrix. In this experiment, the 433 retinal images are split into two groups: 219 images for training and 214 images for testing. A confusion matrix shows the distribution of the class labels predicted by a classifier. The $(i,j)$-element of the confusion matrix shows the counts of images of class $i$ that are classified as class $j$. Therefore, a perfect classifier will have a confusion matrix that has positive values for the elements on the diagonal, i.e., the $(i,i)$-elements, $i$=$1,\ldots,9$, and zeros for other elements.

Table 6.3 shows the confusion matrix of classifying retinal images using the 12 CSD ViVos.



(a) Giantin (b) Hoechst (c) LAMP2 (d) NOP4 (e) Tubulin

**Figure 6.4:** Examples from the dataset of 327 fluorescence micrographs of subcellular protein localization patterns. The images have been enhanced in order to look better in print.

The counts in the confusion matrix are mainly located on the diagonal, showing that the 12 CSD ViVos are good in classifying all 9 classes of images. In this case, the overall classification accuracy is 79 %.

### 6.4.2 Classification of Subcellular Protein Localization Images

In order to assess the generality of our visual vocabulary approach, we also applied our method to classify 327 fluorescence microscopy images of subcellular protein localization patterns [13]. Example micrographs depicting the cell DNA and four protein types are shown in Figure 6.4. We partitioned the data set into training and test sets in the same way as Boland et al. [13].

We note that although these images are very different from the retinal images, the combination of CSD and ICA still classifies 84 % of the images correctly. The 1-NN classifier achieves 100 % accuracy on 3 classes: Giantin, Hoechst, and NOP4. The training images of class LAMP2 in the data set have size 512-by-512, which is different from that of the others, 512-by-382. Due to this discrepancy, class LAMP2 is classified at 83 % accuracy, and around half of Tubulin images are classified as LAMP2. Table 6.4 shows the confusion matrix of the classification result.

To summarize, our classification experiments show that the symbolic ViVo representation captures well the contents of microscopy images of two different kinds. Thus, we are confident that the method is applicable to a wider range of biomedical images.

| | Predicted | | | | |
|---|---|---|---|---|---|
| Truth | Giantin | Hoechst | LAMP2 | NOP4 | Tubulin |
| Giantin | 30 | - | - | - | - |
| Hoechst | - | 30 | - | - | - |
| LAMP2 | - | - | 50 | 9 | 1 |
| NOP4 | - | - | - | 8 | - |
| Tubulin | - | - | 14 | - | 12 |

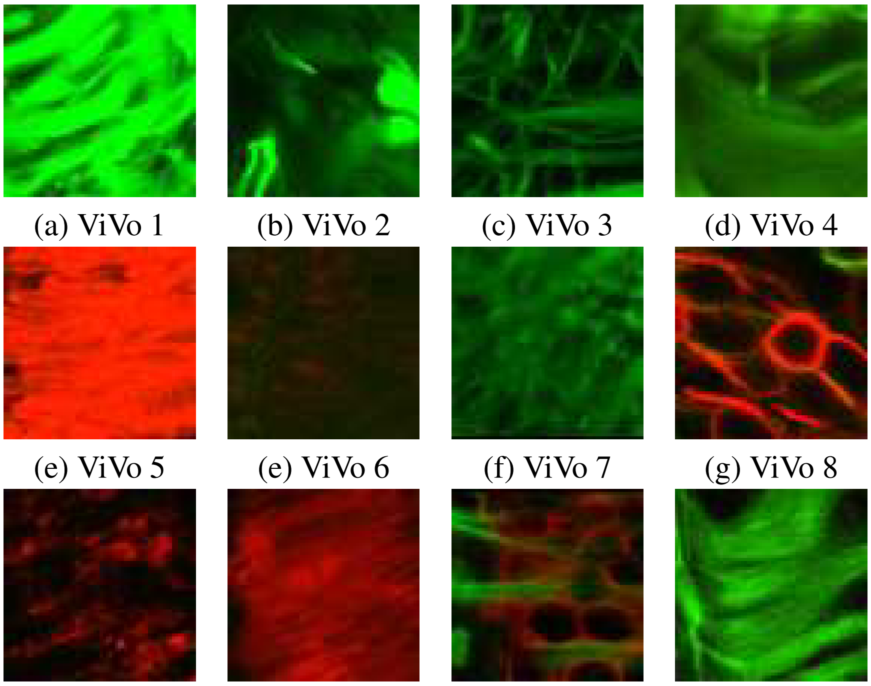
**Table 6.4:** Confusion matrix for the classification experiments on the 327 dataset using an 1-NN classifier. The images are represented by four ViVos constructed from the color structure features. Overall classification accuracy is 84 %.

## 6.5 Qualitative Evaluation: Data Mining Using ViVos

Deriving a visual vocabulary for image content description opens up many exciting data mining applications. In this section, we describe our proposed methods for answering the three problems we introduced in Section 6.1. We first discuss the biological interpretation of the ViVos in Section 6.5.1 and show that the proposed method correctly summarizes a biomedical image automatically (Problem 1). An automated method for spotting differential patterns between classes is introduced in Section 6.5.2 (Problem 2). Several observations on the class-distinguishing patterns are also discussed. Finally, in Section 6.5.3, we describe a method to automatically highlight interesting regions in an image (Problem 3).

### 6.5.1 Biological Interpretation of ViVos

Consulting with domain experts, we find that the 12 ViVos extracted from the CSD features in Section 6.4.1 are all biological meaningful. The biological properties of the 12 ViVos are summarized in Table 6.5. The representative tiles of ViVos 2, 3, 4, 7, and 12 shown in Figure 6.5 demonstrate the hypertrophy of Müller cells. These ViVos correctly discriminate various morphological changes of Müller cells. The green patterns in these representative tiles is due to staining produced by immunohistochemistry with an antibody to GFAP, a protein found in glial cells (in-



(a) ViVo 1 (b) ViVo 2 (c) ViVo 3 (d) ViVo 4
(e) ViVo 5 (e) ViVo 6 (f) ViVo 7 (g) ViVo 8
(i) ViVo 9 (j) ViVo 10 (k) ViVo 11 (l) ViVo 12

**Figure 6.5:** Our visual vocabulary. The vocabulary is automatically constructed from a set of images. Images are best viewed in color.

cluding Müller cells). Our visual vocabulary also captures the normal expression of GFAP in the inner retina, represented by ViVo 1. The Müller cells have been shown to hypertrophy following experimental retinal detachment. Understanding how they hypertrophy and change in morphology is important in understanding how these cells can ultimately form glial scars, which can inhibit the recovery of the nervous system from injury.

Also, our ViVos correctly place tiles into different groups, according to the different anti-rod opsin staining which may occur due to functional consequences following injury. In an uninjured retina, anti-rod opsin (shown in red) stains the outer segments of the rod photoreceptors, which are responsible for converting light into an electrical signal and are vital to vision. ViVos 5 and 10 show a typical staining pattern for an uninjured retina, where healthy outer segments are stained. However, following detachment or other injury to the retina, outer segment degeneration can occur

| ViVo | Description | Condition |
|---|---|---|
| 1 | GFAP in inner retina (Müller cells) | Healthy |
| 2 | Morphological change of GFAP in inner retina (Müller cells) | Detached |
| 3 | Morphological change of GFAP in inner retina (Müller cells) | Detached |
| 4 | Morphological change of GFAP in inner retina (Müller cells) | Detached |
| 5 | Healthy outer segments of rod photoreceptors | Healthy |
| 6 | Rod photoreceptor cell bodies (background labeling) | Background |
| 7 | Morphological change of GFAP in inner retina (Müller cells) | Detached |
| 8 | Redistribution of rod opsin into cell bodies of rod photoreceptors | Detached |
| 9 | Degeneration of outer segments | Detached |
| 10 | Healthy outer segments of rod photoreceptors | Healthy |
| 11 | Co-occurring processes: Muller cell hypertrophy and rod opsin re-distribution | Detached |
| 12 | Morphological change of GFAP in inner retina (Müller cells) | Detached |

**Table 6.5:** The biological properties of the ViVos.

(ViVo 9). Another consequence of retinal detachment can be a re-distribution of rod opsin from the outer segments of these cells to the cell bodies (ViVo 8).

As described above, both the re-distribution of rod opsin and the Müller cell hypertrophy are consequences of retinal detachment. It is of interest to understand how these processes are related. ViVo 11 captures the situation when the two processes co-occur. Being able to sample a large number of images that have these processes spatially overlapping will be important to understanding their relationship. ViVo 6 is rod photoreceptor cell bodies with only background labeling.

### 6.5.2 Finding the Most Discriminative ViVos

We are interested in identifying ViVos that show differences between different retinal experimental conditions, including treatments. Let images $\{I_1,\ldots,I_n\}$ be the training images of condition $c_i$. Suppose that our analysis in Section 6.3 suggests that $m$ ViVos should be used. Following the algorithm outlined in Figure 6.2, we can represent an image $I$ as an $m$-dimensional ViVo-vector $v(I)$. The $k$-th element of a ViVo-vector, $v_k(I)$, gives the expression level of ViVo $v_k$ in the image $I$. Let $\mathcal{S}_{ik}$=$\{v_k(I_1),\ldots,v_k(I_n)\}$ be a set that contains the $k$-th elements of all image ViVo-vectors in

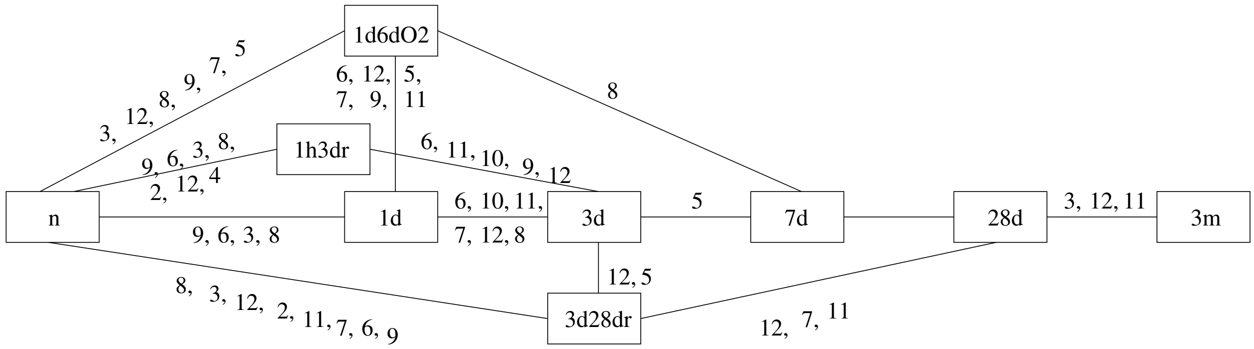



**Figure 6.6:** Pairs of conditions and the corresponding discriminative ViVos. There is an edge in the graph for each pair of conditions that is important from a biological point of view. The numbers on each edge indicate the ViVos that contribute the most to the differences between the conditions connected by that edge. The ViVos are specified in the order of their discriminating power.

condition $c_i$.

To determine if a ViVo $v_k$ is a discriminative ViVo for two conditions $c_i$ and $c_j$, we perform an analysis of variance (ANOVA) test, followed by a multiple comparison [59]. If the 95% confidence intervals of the *true* means of $\mathcal{S}_{ik}$ and $\mathcal{S}_{jk}$ do not intersect, then the means are not significantly different, and we say that ViVo $v_k$ discriminates conditions $c_i$ and $c_j$, i.e., $v_k$ is a discriminative ViVo for $c_i$ and $c_j$. The separation between $\mathcal{S}_{ik}$ and $\mathcal{S}_{jk}$ indicates the "discriminating power" of ViVo $v_k$.

Figure 6.6 shows the conditions as boxes and the discriminative ViVos on edges connecting pairs of conditions that are of biological interest. ViVos 6 and 8 discriminate `n` from `1d` and `1d` from `3d`. The two ViVos represent rod photoreceptor cell bodies with only background labeling and with redistribution of rod opsin, respectively, indicating that the redistribution of rod opsin is an important effect in the short-term detachment. Note also that ViVo 8 distinguishes `1d6dO2` from `7d`. This suggests that there are cellular changes associated with this oxygen treatment, and the ViVo technique can be used for this type of comparison.

The ViVos that represent Müller cell hypertrophy (ViVo 2, 3, 4, 7, and 12) discriminate `n` from all other conditions. We note that ViVo 1, which represents GFAP labeling in the inner retina in both control (`n`) and detached conditions, is present in all conditions, and therefore cannot discriminate any of the pairs in Figure 6.6. In addition, several ViVos discriminate between `3d28dr` and `28d`, and `1h3dr` and `3d`, suggesting cellular effects of the surgical procedure. Interestingly, there are no ViVos that discriminate between `7d` and `28d` detachments, suggesting that the effects of long-term detachment have occurred by 7 days.

Although these observations are generated automatically by an unsupervised tool, they correspond to observations and biological theory of the underlying cellular processes.

### 6.5.3 Highlighting Interesting Regions by ViVos

In this section, we propose a method to find class-relevant ViVos and then use this method to highlight interesting regions in images of a particular class.

In order to determine which condition a ViVo belongs to, we examine its representative tiles and determine the most popular condition among them (majority voting). We define the condition of a tile, $c(\vec{\mathbf{t}}_{i,j})$, to be that of the image from which it was extracted, i.e., $c(\vec{\mathbf{t}}_{i,j}) = c(I_i)$. Intuitively, for a ViVo, the more its representative tiles are present in images of a condition, the more relevant the ViVo is to that condition.

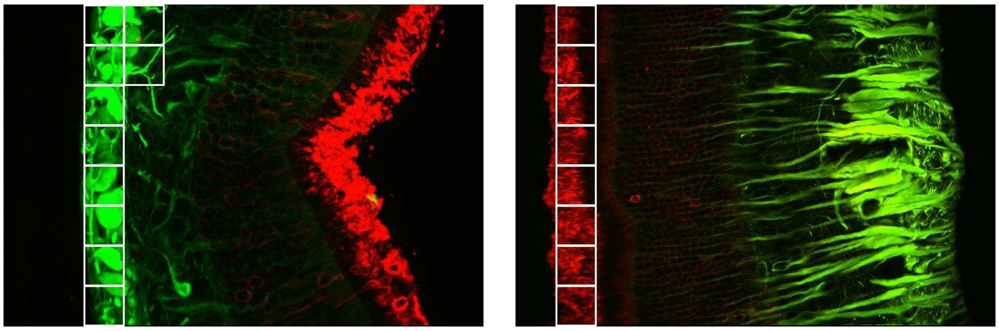
Formally, the set $\mathcal{R}(c_k)$ of representative ViVos of a condition $c_k$ is defined as

$$\mathcal{R}(c_k) = \left\{ v_r : \sum_{t \in \mathcal{T}(v_r)} I(c(t) = c_k) > \sum_{t \in \mathcal{T}(v_r)} I(c(t) = c_q), \forall c_q \neq c_k \right\},$$

where $t$ is a tile, and $I(p)$ is an indicator function that is 1 if the predicate $p$ is true, and 0 otherwise. The representative ViVos of a condition $c_k$ can be used to annotate images of that particular condition in order to highlight the regions with potential biological interpretations.

Figure 6.7(a) shows an annotated image of a retina detached for 28 days. The GFAP labeling in the inner retina is highlighted by ViVo 1 (see Figure 6.5(a)).

Figure 6.7(b) shows an annotated image of a retina detached for 3 days and then reattached for 28 days. The annotation algorithm highlighted the outer segments of the rod photoreceptors with ViVo 10 (see Figure 6.5(j)). As pointed out in Section 6.5.1, ViVo 10 represents healthy



(a) ViVo 1 highlighted (b) ViVo 10 highlighted

**Figure 6.7:** Two examples of images with ViVo-annotations (highlighting) added. (a) GFAP-labeling in the inner retina (28d); (b) rod photoreceptor recovered as a result of reattachment treatment (3d28dr).

outer segments. In the retina depicted in Figure 6.7(b), the outer segments have indeed recovered from the degeneration caused by detachment. This recovery of outer segments has previously been observed [32], and confirms that ViVos can recognize image regions that are consistent with previous biological interpretations.

## 6.6 Summary

Mining biomedical images is an important problem because of the availability of high-throughput imaging, the applicability to medicine and health care, and the ability of images to reveal spatio-temporal information not readily available in other data sources such as genomic sequences, protein structures and microarrays.

We focus on the problem of describing a collection of biomedical images succinctly (Problem 1). Our main contribution is to propose an automatic, domain-independent method to derive meaningful, characteristic tiles (*ViVos*), leading to a *visual vocabulary* (Section 6.3). We apply our technique to a collection of retinal images and validate it by showing that the resulting ViVos correspond to biological concepts (Section 6.5.1).

Using ViVos, we propose two new data mining techniques. The first (Section 6.5.2) mines a large collection of images for patterns that distinguish one class from another (Problem 2). The second technique (Section 6.5.3) automatically highlights important parts of an image that might otherwise go unnoticed in a large image collection (Problem 3).

The conclusions are as follows:

- **Biological Significance:** The terms of our visual vocabulary correspond to concepts biologists use when describing images and biological processes.
- **Quantitative Evaluation:** Our ViVo-tiles are successful in classifying images, with accuracies of 80 % and above. This gives us confidence that the proposed visual vocabulary captures the essential contents of biomedical images.
- **Generality:** We successfully applied our technique to two diverse classes of images: localization of different proteins in the retina, and subcellular localization of proteins in cells. We believe it will be applicable to other biomedical images, such as X-ray images, MRI images, and electron micrographs.
- **Biological Process Summarization:** Data mining techniques can use the visual vocabulary to describe the essential differences between classes. These techniques are unsupervised, and allow biologists to screen large image databases for interesting patterns.

# Part II

# Cross-Modal Pattern Discovery

# Chapter 7

# Motivation and Related Work

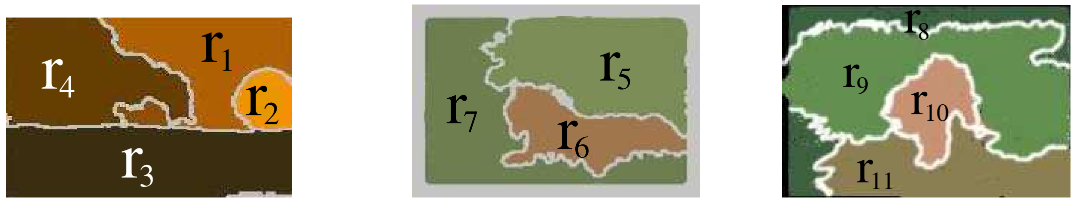
In part I, we discuss methods and results on finding uni-modal patterns from text documents, images, audio signals, and time sequences. For multimedia objects such as video clips that have attributes of more than one modality, the correlations between different modalities also carry plenty of information about the characteristics of these multimedia objects. The correlations and patterns across various media in multimedia objects could be very useful in practical multimedia applications like annotation and summarization.

In this part of the thesis, we will discuss methods for *cross-modal pattern discovery*, or *cross-modal correlation discovery*. We start by motivating and specifying the problem of cross-modal pattern discovery, followed by a literature survey on related work in this area. In chapter 8, we introduce our proposed method, MAGIC, for cross-modal pattern discovery. We apply MAGIC to two different domains: captioned images and video clips. In chapter 9, MAGIC finds correlations between caption words and the corresponding image, and achieves better captioning accuracy, compared to recent machine learning approaches. In chapter 10, MAGIC is applied to news videos, to identify materials of various modalities for news topic summarization and other data mining applications.

('sea', 'sun', 'sky', 'waves') (a) Captioned image $I_1$

('cat', 'forest', 'grass', 'tiger') (b) Captioned image $I_2$

no caption (c) Image $I_3$




(d) (e) (f)

**Figure 7.1:** Three sample images: (a),(b) are captioned with terms describing the content; (c) is an image to be captioned. (d)(e)(f) show the regions of images (a)(b)(c), respectively. Figures look best in color.

## 7.1 Introduction

Advances in digital technologies make possible the generation and storage of large amount of multimedia objects such as images and video clips. Multimedia content contains rich information in various modalities such as images, audios, video frames, time series, etc. However, making rich multimedia content accessible and useful is not easy. Advanced tools that find characteristic patterns and correlations among multimedia content are required for the effective usage of multimedia databases.

We call a data object which has its content presented in more than one modality a *mixed media* object. For example, a video clip is a mixed media object with image frames, audios, and other information such as transcript text. Another example is a captioned image such as a news picture with an associated description, or a personal photograph annotated with a few keywords (Figure 7.1). In this thesis, we use the terms *medium* (plural form *media*) and *modality* interchangeably.

It is common to see correlations among attributes of different modalities on a mixed media object. For instance, a news clip usually contains human speech accompanied with images of static scenes, while a commercial has more dynamic scenes and loud background music [98]. In image archives, caption keywords are chosen such that they describe objects in the images. Similarly, in digital video libraries and in the entertainment industry, motion picture directors edit sound effects to match the scenes in video frames.

Cross-modal correlations provide helpful hints on exploiting information from different modalities for tasks such as segmentation [42] and indexing [16]. Also, establishing associations between low-level features and attributes that have semantic meanings may shed light on multimedia understanding. For example, in a collection of captioned images, discovering the correlations between images and caption words, could be useful for content-based image retrieval, and image annotation and understanding.

The question that we are interested in is: "*Given a collection of mixed media objects, how do we find correlations across data of various modalities?*" For example, given a collection of captioned images (Figure 7.1), we want to find correlations between images and keywords, so that when given a new, uncaptioned image, we can caption the new image with keywords correlated with it.

A desirable solution for cross-modal correlation discovery should be able to include all modalities that have different properties, overcome noise in the data, and detect correlations between any subset of available modalities. Moreover, in terms of computation, we would like a method that is scalable to the database size and does not require human fine-tuning.

In particular, we want a method that can find correlations among all attributes, rather than just between specific attributes. For example, we want to find not just the image-term correlation between an image and caption terms, but also term-term and image-image correlations, using one single framework. This *any-to-any medium correlation* provides a greater picture of how attributes are correlated, e.g., "which word is usually used for images with blue top," "what words have related semantics," and "what objects appear often together in an image."

We proposed a novel, domain-independent framework, *MAGIC*, for cross-modal correlation discovery. MAGIC turns the multimedia problem into a graph problem, by providing an intuitive framework to represent data of various modalities. The proposed graph framework enables the application of graph algorithms to multimedia problems. In particular, MAGIC employs the *random walk with restarts* technique on the graph to discover cross-modal correlations.

In summary, MAGIC has the following advantages:

- It provides a graph-based framework which is domain independent and applicable to mixed media objects which have attributes of various modalities;
- It can spot any-to-any medium correlations;
- It is completely automatic (its few parameters can be automatically preset);
- It can scale up for large collections of objects.

In this study, we evaluate the proposed MAGIC method on two tasks: *automatic image captioning* and *multi-modal summarization*. For automatic image captioning, the correlations between image and text are used to predict caption words for an uncaptioned image. For multi-modal summarization, we applied MAGIC to generate summaries of news events in broadcast news videos, by identifying video shots and transcript words relevant to an event.

**Application 1 (Automatic Image Captioning)** *Given a set $I_{core}$ of color images, each with caption words; and given an uncaptioned image $I_{new}$, find the best q (say, q=5) caption words to assign to it.*

**Application 2 (Multi-Modal Summaries of News Events)** *Given a collection of news video clips, in which each has one or more (set-valued) multimedia attributes (keyframes, transcripts, event symbols, etc.), identify relevant materials of all media which are pertinent to a user-specified news event.*

The proposed method can also be easily extended for various related applications such as captioning images in groups, or retrieving relevant video shots and transcript words.

## 7.2 Related Work

Combining information of multiple data modalities and mining cross-modal correlations provide a better multimedia data representation and have been shown useful in applications such as retrieval, segmentation, classification, and pattern discovery [16]. In this section, we survey previous work on cross-modal correlation mining/modeling. We also discuss previous work on image captioning and news event summarization, which are our application domains on which we evaluate our proposed method for cross-modal correlation discovery.

### 7.2.1 Multimedia Cross-modal Correlation

Combining information about multimedia correlations in applications leverages all available information, and has led to improved performances in segmentation [42], classification [77, 19], retrieval [131, 138, 128], and topic detection [132, 26]. One crucial step of fusing multi-modal correlations into applications is to detect, extract, and model the cross-modal correlations from data.

We categorize previous methods for multimedia correlation modeling into two categories: *model-driven* approaches and *data-driven* approaches. A model-driven method usually assumes a model of the correlations in data, and extract the correlations that are captured by the assumed model. A data-driven method finds data correlations using solely the relationship (e.g., similarity) between data objects.

The model of a model-driven method is usually hand-designed based on the knowledge to the domain, and it is a good way to incorporate prior knowledge into the correlation discovery process. However, the quality of the extracted correlations depends on the correctness of the assumed model. On the other hand, the performance of a data-driven method is less dependent on the available domain knowledge, but the ability to incorporating prior knowledge to guide the discovery of correlations is more limited.

**Model-Driven Methods** Previous model-driven methods have proposed a variety of models to extract information from multi-modal data. These models include linear models [73, 119], graphical models [8, 92, 30, 52], statistical models [132, 42, 19], and meta-classifiers [131, 77]. Most of these works are designed for particular applications on which domain knowledge is available for model design. Leveraging domain knowledge also provides better opportunities to boost performance.

When the data is consistent with the assumed model, a model-driven method can capture the correct cross-modal correlations and achieve best performance. Linear models [73, 119] assume that data variables have linear correlations, and they are computationally friendly. In cases where linear models do not approximate the real correlations well, more complex statistical models can be used: for example, the mixture of Gaussians [19], the maximum-entropy model [42], or the hidden Markov model [132]. Graphical models [8, 92, 30, 52] have attracted much attention for its ability to incorporate domain knowledge into data modeling. A graphical model is usually used to define a joint probability of all variables participated in the analysis, based on some assumed data generating process. The quality of the graphical model depends on the correctness of the assumed generative process, and sometimes the modeling is computationally intractable if the model is too complex.

Classifier-based models are suitable when the application is data classification [131, 77]. Classifiers are useful in capturing the discriminative patterns between different data types. To combine information from multiple data modalities, one common method is to use a meta-classifier. A meta-classifier is essentially a hierarchy of classifiers [131, 77], where the fusion of multi-modal information can be done directly either by feeding a classifier with multi-modal input, or by merging the information with a classifier that takes as input the outputs of multiple uni-modal classifiers.

Domain knowledge is needed to determine the types and parameters of the individual classifiers participated in the meta-classifier (e.g., what kernel to use for a SVM classifier), and how these classifiers are connected in the hierarchy. Outputs of classifiers can also be considered as *mid-level features* or *concepts*, such as “human” or “explosion”, etc. By combining the classifiers with other models (such as graphical models), one can take advantage of both classifiers and graphical models, and model complex relationship between multi-modal concepts [8, 92].

**Data-Driven Methods** Unlike a model-driven method which fits a pre-specified correlation model to the given data set, a data-driven method finds cross-modal correlations by using the similarity relationship between data objects in the set. A natural way to present the similarity relationship between multimedia data objects is using a graph representation, where nodes symbolize objects and edges (with weights) indicate the similarity between objects.

Different graph-based algorithms have been proposed to find correlations from the graph representation of the data set, according to the application domains. For example, "spectral clustering" has been proposed for clustering data from different video sources [137], as well as for grouping relevant data of different modalities [26]. Link analysis techniques have been used for deriving a multi-modal (image and text) similarity function for web image retrieval [128]. For these methods, graph nodes are used represent multimedia objects, and the focus is on finding correlations between data objects. This *object-level graph representation* make deciding the similarity function between objects (for constructing the graph edges) difficult. For complex multimedia objects, designing a good similarity function is even more difficult.

In this thesis, we propose a method to find general cross-modal correlations, between data objects as well as their attributes. By explicitly representing the data attributes as graph nodes, we can find *any-to-any correlations*: correlations between objects and objects, objects and attributes, or attributes and attributes, etc. Moreover, the graph of data objects can be constructed via connections through similar attributes, whose similarity functions are relatively easier to define. We show that the any-to-any correlations could be applied to several applications, including image captioning and video news story summarization, and achieve better performance.

Our proposed framework, MAGIC, does not need a training phase, and has fewer parameters to tune. In fact, as we show later, the results are insensitive to parameter values in our experiments (Section 11.1). MAGIC uses a graph to represent the relations between objects and low-level attribute values, and does not need detailed specifications of concepts or complicated similarity functions.

### 7.2.2 Image Captioning

Although a picture is worth a thousand words, extracting the abundant information from an image is not an easy task. Computational techniques are able to derive low-to-mid level features (e.g., texture and shape) from pixel information; however, the gap still exists between mid-level features and concepts used by human reasoning [111, 139, 138]. One consequence of this semantic gap in image retrieval is that the user's need is not properly matched by the retrieved images, and may be part of the reason that practical image retrieval has yet to become popular.

Automatic image captioning, where the goal is to predict caption words to describe image content, is one research direction to bridge the gap between concepts and low-level features. Previous work on image captioning employs various approaches such as linear models [100, 89], classifiers [84], language models [126, 25, 52], graphical models [5, 11], statistical models [74, 54, 30], and a framework with user involvement [129].

Most previous approaches derive features from image regions (regular grids or blobs [25]), and construct a model between images and words based on a reference captioned image set. Images in the reference set are captioned by human experts; however, there is no information about the associations between individual regions and words. Some approaches attempt to explicitly infer the correlations between regions and words [25], with enhancements that take into consideration interactions between neighboring regions in an image [74]. Alternatively, there are methods which model the collective correlations between regions and words of an image [101, 102].

Comparing the performance of different approaches is not easy. Several benchmark data sets are available; however, not all previous work reports results on the same subset of images. On the other hand, various metrics such as accuracy, term precision and recall, and mean average precision have been used to measure the performance. Since the perception of an image is subjective, some work also reports user evaluation of the captioning result.

In Chapter 9, our proposed method, MAGIC, is applied to automatic image captioning. The correlations between words and images are detected and applied to predict caption words of a previously unseen image. To better evaluate our approach, we conduct experiments on the same data sets and report using the same performance metrics used in other previous work [126, 25, 5, 11].

### 7.2.3 Broadcast News Event Summarization

Video summarization is in great demand to enable users to efficiently access massive video collections [127]. In this study, we apply our proposed method to summarize events in broadcast news videos by detecting cross-modal correlations between video frames, shots and transcript words.

Analyzing *events* in video has attracted much attention recently. The goal of event summarization is to identify and gather pieces of video shots which are related to one event, as opposed to creating a condensed version of a video clip for browsing [118, 94]. Some work focuses on using only the visual information to detect events such as simple motions ("walking" and "waving" [135, 63]) or repeating/periodic scenes [78]. On the other hand, when incorporated with information from multiple media, it is possible to detect events at a higher semantic level [85, 17, 49]. Similar research directions have also been pursued in the textual domain, where the focus is to summarize multiple documents of the same event [36, 9].

To visualize the summary of a video collection, the video collage method [17] arranges frames as a storyboard [134, 124] and shows them alongside the corresponding transcripts or other metadata. A similar presentation is adopted in our experiment to present our result on summarizing broadcast news events (Chapter 10).

The following chapters are organized as follow: In Chapter 8, we present our proposed method, MAGIC, which provides a graph-based framework for correlation discovery. We apply MAGIC to two problems, automatic image captioning (Chapter 9 and news event summarization (Chapter 10). For automatic image captioning, MAGIC finds robust image-text correlations, and achieves a captioning accuracy better than recent machine learning methods. When applied to broadcast news video, MAGIC gives promising results for news event summarization. In Chapter 11, we discuss some system issues and propose a method to further speedup MAGIC by precomputation. Chapter 12 summarizes the MAGIC method and the correlations that MAGIC found.

# Chapter 8

# Proposed Method: MAGIC

The mixed media correlation discovery method we proposed contains two components: a graph-based representation for multimedia objects with set-valued attributes, and a technique based on random walks for finding any-to-any medium correlation. In this section, we explain how to generate the graph representation and how to detect cross-media correlations.

## 8.1 The MAGIC Graph

In relational database management systems, a multimedia object is usually represented as a vector of *m* features/attributes [28]. The attributes must be *atomic* (i.e., taking single values) like "size" or "the amount of red color" of an image. However, for mixed media data sets, the attributes can be *set-valued*, such as the caption of an image (a set of words) or the image regions.

Finding correlations among set-valued attributes is not easy: Elements in a set-valued attribute could be noisy or missing altogether; regions in an image are not perfectly identified (noisy regions); the image caption may be incomplete, leaving out some aspects of the content. Set-valued attributes of an object may have different numbers of elements, and there is no given alignment between set elements. For instance, an image may have unequal numbers of caption words and regions, where a word may describe multiple regions and a region may be described by zero or more than one word.

We assume that the elements of a set-valued attribute are tokens drawn from a *domain*. We propose to gear our method toward set-valued attributes, because they include atomic attributes as a special case; and they also smoothly handle the case of missing values (null set).

**Definition 3 (Domain and Domain Token)** *The* **domain** $D_i$ *of (set-valued) attribute i is a collection of atomic values, which we call* **domain tokens***, which are the values that attribute i can take.*

A domain can consist of categorical values, numerical values, or numerical vectors. For example, for automatic image captioning, we have objects with $m$=2 attributes. The first attribute, "caption", has a set of categorical values (English terms) as its domain ; the second attribute, "regions", is a set of image regions, each of which is represented by a $p$-dimensional vector of $p$ features derived from the region (e.g., color histogram with $p$ colors). As described later in Chapter 9, we extract $p$=30 features from each region. To establish the relation between domain tokens, we assume that we have a similarity function for each domain. Domain tokens are usually simpler than mixed media objects, and therefore it is easier to define similarity functions on domain tokens than on mixed media objects.

**Assumption 1** *For each domain $D_i$ ($i = 1, \ldots, m$), we are given a similarity function $Sim_i(*,*)$ which assigns a score to a pair of domain tokens.*

For example, for the attribute "caption", the similarity function could be 1 if the two tokens are identical, and 0 if they are not.

Perhaps surprisingly, with Definition 3 and Assumption 1, we can encompass all the applications mentioned in Section 7.1. The main idea is to represent all objects and their attributes (domain tokens) as nodes of a *graph*. For multimedia objects with $m$ attributes, we obtain a $(m+1)$-layer graph. There are $m$ types of nodes (one for each attribute), and one more type of nodes for the

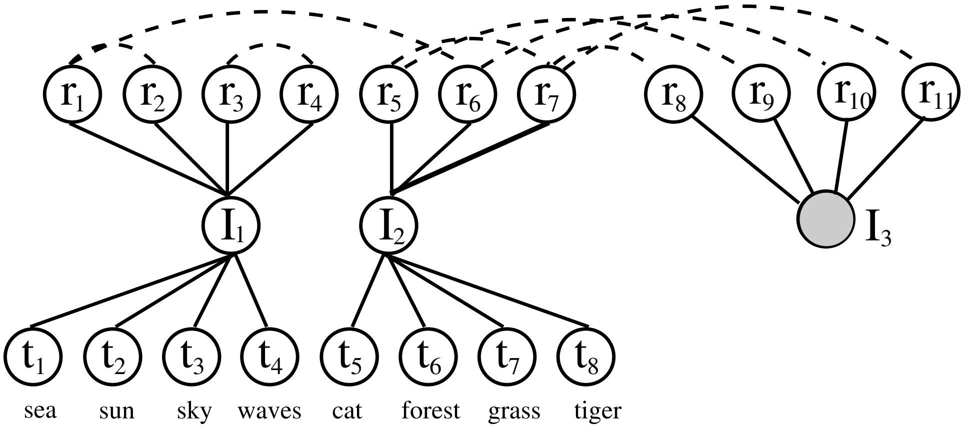



**Figure 8.1:** MAGIC graph ($G_{MAGIC}$) corresponds to the 3 images in Figure 7.1. Solid edges: OAV-links; dash edges: NN-links.

objects. We call this graph a MAGIC graph ($G_{MAGIC}$). We put an edge between every object-node and its corresponding attribute-value nodes. We call these edges *object-attribute-value links* (OAV-links).

Furthermore, we consider that two objects are similar if they have similar attribute values. For example, two images are similar if they contain similar regions. To incorporate such information into the graph, our approach is to add edges to connect pairs of domain tokens (attribute values) that are similar, according to the given similarity function (Assumption 1). We call edges that connect nodes of similar domain tokens *nearest-neighbor links* (NN-links).

We need to decide on a threshold for "closeness" when adding NN-links. We propose to make the threshold adaptive: each domain token is connected to its $k$ nearest neighbors. Computing nearest neighbors can be done efficiently, because we already have the similarity function $Sim_i(*,*)$ for any domain $D_i$ (Assumption 1). In Section 11.1, we discuss the choice of $k$, as well as the sensitivity of our results to $k$.

We illustrate the construction of $G_{MAGIC}$ graph by the following example.

**Example 1** *For the images $\{I_1, I_2, I_3\}$ in Figure 7.1, the MAGIC graph ($G_{MAGIC}$) corresponding to these images is shown in Figure 8.1. The graph has three types of nodes: one for the image objects*

**Input:**
1. $O$: a set of $n$ objects (objects are numbered from 1 to $n$).
2. $\mathcal{D}_1, \ldots, \mathcal{D}_m$: the domains of the $m$ attributes of the objects in $O$.
3. $Sim_1(*,*), \ldots, Sim_m(*,*)$: the similarity functions of domains $\mathcal{D}_1, \ldots, \mathcal{D}_m$, respectively.
4. $k$: the number of neighbors a domain token connects to.

**Output:**
$G_{MAGIC}$: a MAGIC graph with a $(m+1)$-layer structure.

**Steps:**
1. Create $n$ nodes (the object nodes), one for each object. These nodes form the layer 1.
2. For each domain $\mathcal{D}_i$, for $i = 1, \ldots, m$.
   (2.1) Let $n_i$ be the number of tokens in the domain $\mathcal{D}_i$.
   (2.2) Create $n_i$ nodes (the token nodes), one for each domain tokens in $\mathcal{D}_i$. This is the $(i+1)$-th layer.
   (2.3) Construct the OAV-links from the object nodes to these token nodes.
   (2.4) Construct the NN-links between the token nodes.
3. Output the final $(m+1)$-layer graph, with $N$=$n$+$\sum_{i=1}^{m} n_i$ nodes, and the OAV-links and NN-links.

**Figure 8.2:** Algorithm: $G_{MAGIC}$= buildgraph($O$, $\{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$, $\{Sim_1(*,*), \ldots, Sim_m(*,*)\}$, $k$)

*$I_j$'s ($j = 1,2,3$); one for the regions $r_j$'s ($j = 1, \ldots, 11$), and one for the terms $\{t_1, \ldots, t_8\}$=$\{$sea, sun, sky, waves, cat, forest, grass, tiger$\}$. Solid arcs are the object-attribute-value links (OAV-links), and dashed arcs are the nearest-neighbor links (NN-links).*

In Example 1, we consider only $k$=1 nearest neighbor, to avoid cluttering the diagram. Because the nearest neighbor relationship is not symmetric and because we treat the NN-links as un-directional, some nodes are attached to more than one link. For example, node $r_1$ has two NN-links attached: $r_2$'s nearest neighbor is $r_1$, but $r_1$'s nearest neighbor is $r_6$. There is no NN-link between term-nodes, due to the definition of its similarity function: 1, if the two terms are the same; or 0 otherwise. Figure 8.2 shows the algorithm for constructing a MAGIC graph.

We use image captioning only as an illustration: the same framework can be used generally for other problems. To solve the automatic image captioning problem, we also need to develop a method to find good caption words - words that correlate with an image, using the $G_{MAGIC}$ graph. This means that, for example, for image $I_3$ we need to estimate the affinity of each term (nodes $t_1$,

| Symbol | Description |
|---|---|
| | Sizes |
| $n$ | The number of objects in a mixed media data set. |
| $m$ | The number of attributes (domains). |
| $N$ | The number of nodes in $G_{MAGIC}$. |
| $E$ | The number of edges in $G_{MAGIC}$. |
| $k$ | Domain neighborhood size: the number of nearest neighbors that a domain token is connected to. |
| $c$ | The restart probability of RWR (random walk with restarts, RWR). |
| $\mathcal{D}_i$ | The domain of the $i$-th attribute. |
| $Sim_i(*,*)$ | The similarity function of the $i$-th domain. |
| | Image captioning |
| $I_{core}$ | The given captioned image set (the core image set). |
| $I_{test}$ | The set of to-be-captioned (test) images. |
| $I_{new}$ | An image in $I_{test}$. |
| $G_{core}$ | The subgraph of $G_{MAGIC}$ containing all images in $I_{core}$ (Chapter 9). |
| $G_{aug}$ | The augmentation to $G_{core}$ containing information of an image $I_{new}$ (Chapter 9). |
| $\mathcal{GW}$ | The gateway nodes, nodes in $G_{core}$ that adjacent to $G_{aug}$ (Chapter 9). |
| | Random walk with restarts (RWR) |
| $\mathbf{A}$ | The (column-normalized) adjacency matrix. |
| $\vec{\mathbf{v}}_{\mathcal{R}}$ | The restart vector of the set of query objects $\mathcal{R}$, where components correspond to query objects have value $1/\|\mathcal{R}\|$, while others have value 0). |
| $\vec{\mathbf{u}}_{\mathcal{R}}$ | The RWR scores of all nodes with respect to the set of query objects $\mathcal{R}$. |
| $\vec{\mathbf{v}}_q$, $\vec{\mathbf{u}}_q$ | $\vec{\mathbf{v}}_{\mathcal{R}}$ and $\vec{\mathbf{u}}_{\mathcal{R}}$ for the singleton query set $\mathcal{R}=\{q\}$. |
| $\vec{\mathbf{v}}_{\mathcal{GW}}$, $\vec{\mathbf{u}}_{\mathcal{GW}}$ | $\vec{\mathbf{v}}_{\mathcal{R}}$ and $\vec{\mathbf{u}}_{\mathcal{R}}$ for RWR restarting from the gateway nodes $\mathcal{GW}$. |

**Table 8.1:** Summary of symbols used in Part II of the thesis.

..., $t_8$) to node $I_3$. The terms with the highest affinity to image $I_3$ will be predicted as its caption words.

Table 8.1 summarizes the symbols we used in Part II of this thesis.

## 8.2 Correlation Detection with Random Walks on Graphs

Our main contribution is to turn the cross-media correlation discovery problem into a graph problem. The previous section describes the first step of our proposed method: representing set-valued mixed media objects in a graph $G_{MAGIC}$. Given such a graph with mixed media information, *how do we detect the cross-modal correlations in the graph?*

We define that a node $A$ of $G_{MAGIC}$ is correlated to another node $B$ if $A$ has an "affinity" for $B$.

There are many approaches for ranking all nodes in a graph by their "affinity" for a reference node. We can tap the sizable literature of graph algorithms and use off-the-shelf methods for assigning importance to vertices in a graph. These include the electricity-based approaches [97, 22], random walks (PageRank, topic-sensitive PageRank) [14, 39], hubs and authorities [58], elastic springs [81] and so on. Among them, we propose to use *random walk with restarts* (RWR) [39] for estimating the affinity of node $B$ with respect to node $A$. However, the specific choice of method is orthogonal to our framework.

The method "random walk with restarts" (RWR) is chosen because (a) it takes into account the query node (the restart node) when ranking the nodes in the graph (we call this *query-dependent ranking*), and (b) it considers the entire graph structure when ranking the nodes. RWR is closely related to the topic-sensitive PageRank method [39], which is a variant of the PageRank algorithm [14]. The PageRank method ranks the "importance" of a graph node (say, a web page) by estimating how frequently the node will be visited by a random "surfer" on the graph. The ranking of PageRank is with respect to the graph and is not query-dependent, that is, it does not adjust the ranking according to the location of the query node.

Graph-based ranking algorithms such as HITS [58] do query-dependent ranking by limiting the ranking to a subset of nodes that are relevant to the query. That is, a filtering step is performed first to select nodes relevant to the query, and then the ranking is computed only on these relevant nodes. However, the quality of the final result depends on the relevance function for filtering. Another concern is that after filtering, the ranking is based on the subgraph of relevant nodes: it is not using the full information of the entire graph.

When the query contains both relevant and irrelevant nodes, we can choose to use graph-based ranking algorithms based on concepts from electricity networks [97, 22] (or equivalently, networks of elastic springs [81]). The graph is viewed as an electrical network, where each edge represents a unit resistance. If we consider that a +1V voltage is applied to all relevant query nodes and a 0V voltage is applied to all irrelevant query nodes, then the ranking of a node is defined to be the voltage measured at that node.

For the problem of cross-modal correlation discovery, we are given relevant query objects in some modality and we want to find other relevant objects in all modalities. Since information about irrelevant objects is not given, we cannot use the electricity-based methods. We do not choose the HITS algorithm because of lacking a proper relevance function for filtering. In fact, the cross-modal correlation discovery problem is essentially finding this relevance function. Therefore, for cross-modal correlation discovery, we decide to use "random walk with restarts".

The "random walk with restarts" operates as follows: To compute the affinity $u_A(B)$ of node $B$ for node $A$, consider a random walker that starts from node $A$. The random walker chooses randomly among the available edges every time, except that, before he makes a choice, he goes back to node $A$ (restart) with probability $c$. Let $u_A(B)$ denote the steady state probability that our random walker will find himself at node $B$. Then, $u_A(B)$ is what we want, the affinity of $B$ with respect to $A$. We also call $u_A(B)$ the *RWR score* of $B$ with respect to $A$. The algorithm of computing RWR scores of all nodes with respect to a subset of nodes $\mathcal{R}$ is given in Figure 8.3.

**Definition 4 (RWR Score)** *The RWR score, $u_A(B)$, of node B with respect to node A is the steady state probability of node B, when we do the random walk with restarts from A, as defined above.*

Let $\mathbf{A}$ be the adjacency matrix of the given graph $G_{MAGIC}$, where columns of the matrix are normalized such that each sums up to 1. Let $\vec{\mathbf{u}}_q$ be a vector of RWR scores of all $N$ nodes, with respect to a restart node $q$. Let $\vec{\mathbf{v}}_q$ be the "restart vector", which has all $N$ elements zero, except for the entry that corresponds to node $q$, which is set to 1. We can now formalize the definition of RWR scores (Definition 5).

**Definition 5 (RWR Score Computation)** *The $N$-by-1 steady state probability vector $\vec{\mathbf{u}}_q$, which contains the RWR scores of all nodes with respect to node q, satisfies the equation:*

$$\vec{\mathbf{u}}_q = (1-c)\mathbf{A}\vec{\mathbf{u}}_q + c\vec{\mathbf{v}}_q, \tag{8.1}$$

*where c is the restart probability of the RWR from node q.*

**Input:**
1. $G_{MAGIC}$: a MAGIC graph with $N$ nodes (nodes are numbered from 1 to $N$).
2. $\mathcal{R}$: a set of restart nodes. (Let $|\mathcal{R}|$ be the size of $\mathcal{R}$.)
3. $c$: the restart probability.
**Output:**
$\vec{\mathbf{u}}_{\mathcal{R}}$: the RWR scores of all nodes with respect to $\mathcal{R}$
**Steps:**
1. Let $\mathbf{A}$ be the adjacency matrix of $G_{MAGIC}$. Normalize the columns of $\mathbf{A}$ and make each column sum up to 1.
2. $\vec{\mathbf{v}}_{\mathcal{R}}$ is the $N$-by-1 restart vector, whose $i$-th element $\vec{\mathbf{v}}_{\mathcal{R}}(i)$ is $\frac{1}{|\mathcal{R}|}$, if node $i$ is in $\mathcal{R}$; otherwise, $\vec{\mathbf{v}}_{\mathcal{R}}(i)$=0.
3. Initialize $\vec{\mathbf{u}}_{\mathcal{R}}$=$\vec{\mathbf{v}}_{\mathcal{R}}$.
4. while($\vec{\mathbf{u}}_{\mathcal{R}}$ has not converged)
  4.1 Update $\vec{\mathbf{u}}_{\mathcal{R}}$ by $\vec{\mathbf{u}}_{\mathcal{R}}$ = (1-c)$\mathbf{A}\vec{\mathbf{u}}_{\mathcal{R}}$ + c$\vec{\mathbf{v}}_{\mathcal{R}}$
5. Return the converged $\vec{\mathbf{u}}_{\mathcal{R}}$.

**Figure 8.3:** Algorithm: $\vec{u}_{\mathcal{R}}$ = RWR($G_{MAGIC}$, $\mathcal{R}$, $c$)

The computation of RWR scores can be done efficiently by matrix multiplication (Step 4.1 in Figure 8.3), with computational cost scaling linearly with the number of elements in the matrix $\mathbf{A}$, i.e., the number of graph edges determined by the given database. In general, the computation of RWR scores converges after a few ($\sim$ 10) iterations (Step 4 in Figure 8.3). In our experiments, each RWR computation takes less than 5 seconds. Therefore, the computation of RWR scales well with the database size. Fortunately, MAGIC is modular and can continue to improve its performance by including the best module [56, 57] for fast RWR computation.

The RWR scores specify the correlations across different media and could be useful in many multimedia applications. For example, to solve the image captioning problem for image $I_3$ in Figure 7.1, we can compute the RWR scores $\vec{\mathbf{u}}_{I_3}$ of all nodes and report the top few (say, 5) term-nodes as caption words for image $I_3$. Effectively, MAGIC exploits the correlations across images, regions and terms to caption a new image.

The RWR scores also provide a means for detecting any-to-any medium correlation. In our running example of image captioning, an image is captioned with the term nodes of highest RWR

Step 1: Identify the objects $O$ and the $m$ attribute domains $\mathcal{D}_i$, $i = 1, \ldots, m$.
Step 2: Identify the similarity functions $Sim_i(*, *)$ of each domain.
Step 3: Determine $k$: the neighborhood size of the domain tokens. (Default value $k = 3$.)
Step 4: Build the MAGIC graph,
  $G_{MAGIC}$= buildgraph($O$, $\{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$, $\{Sim_1(*, *), \ldots, Sim_m(*, *)\}$, $k$).
Step 5: Given a query node $\mathcal{R}$=$\{q\}$ ($q$ could be an object or a token),
  (Step 5.1) Determine the restart probability $c$. (Default value $c = 0.65$.)
  (Step 5.2) compute the RWR scores:
    $\vec{u}_{\mathcal{R}}$ = RWR($G_{MAGIC}$, $\mathcal{R}$, $c$).
Step 6: Objects and attribute tokens with high RWR scores are correlated with $q$.

**Figure 8.4:** Instructions for detecting correlations using MAGIC. Functions "buildgraph()" and "RWR()" are given in Figures 8.2 and 8.3, respectively.

scores. In addition, since all nodes have their RWR scores, other nodes (say, image nodes) can also be ranked and sorted, for finding images that are most related to image $I_3$. Similarly, we can find the most relevant regions. In short, we can restart from any subset of nodes, say term nodes, and derive term-to-term, term-to-image, or term-to-*any* correlations. We discuss this more in Section 9.3. Figure 8.4 shows the overall procedure of using MAGIC for correlation detection.

# Chapter 9

# Case Study: Automatic Image Captioning

Cross-modal correlations are useful for many multimedia applications. In this section and the next, we present results of applying the proposed MAGIC method to two applications - image captioning [101, 102] and broadcast news events summarization [103]. Intuitively, the cross-modal correlations are used in the way that an image will be captioned with words correlated with the image content, and a video news event will be summarized by correlated multimedia materials (video shots, transcript terms, and other event symbols).

On captioning images, we evaluate the quality of the cross-modal correlations by MAGIC in terms of the captioning accuracy. We show experimental results to address the following questions:

- Quality: Does MAGIC predict the correct caption terms?
- Generality: Beside the image-to-term correlation for captioning, does MAGIC capture any-to-any medium correlations?

Our results show that MAGIC successfully exploits the image-to-term correlation to caption test images. Moreover, MAGIC is flexible and can caption multiple images as a group. We call this operation "*group captioning*" and present some qualitative results.

We also examine MAGIC's performance on spotting other cross-modal correlations. In particular, we show that MAGIC can capture same-media correlations such as the term-term correlations:

E.g., "given a term such as 'sky', find other terms that are likely to correspond to it." Potentially, MAGIC is also capable of spotting other correlations such as the reverse captioning problem: E.g., "given a term such as 'sky', find the regions that are likely to correspond to it." In general, MAGIC can capture any-to-any medium correlations.

## 9.1 Data set and the MAGIC Graph Construction

*Given a collection of captioned images $I_{core}$, how do we select caption words for an uncaptioned image $I_{new}$?* For automatic image captioning, we propose to caption $I_{new}$ using the correlations between caption words and images in $I_{core}$.

In our experiments, we use the same 10 sets of images from Corel that are also used in previous work [25, 5], so that our results can be compared to the previous results. In the following, the 10 captioned image sets are referred to as the "001", "002", ..., "010" sets. Each of the 10 data sets has around 5,200 images, and each image has about 4 caption words. These images are also called the *core images* from which we try to detect the correlations. For evaluation, accompanying each data set, a non-overlapping test set $I_{test}$ of around 1,750 images is used for testing the captioning performance. Each test image has its ground truth caption.

Similar to previous work [25, 5], each image is represented by a set of image regions. Image regions are extracted using a standard segmentation tool [114], and each region is represented as a 30-D feature vector. The regional features include the mean and standard deviation of RGB values, average responses to various texture filters, its position in the entire image layout, and some shape descriptors (e.g., major orientation and the area ratio of bounding region to the real region). The image content is represented as a set-valued attribute "regions". In our experiments, an image has 10 regions on average. Figure 7.1(d,e,f) show some examples of image regions.

The exact region segmentation and feature extraction details are *orthogonal* to our approach - any published segmentation methods and feature extraction functions [28] will suffice. All our MAGIC method needs is a black box that will map each color image into a set of zero or more feature vectors.

We want to stress that there is no given information about which region is associated with which term in the core image set - all we know is that a set of regions co-occurs with a set of terms in an image. That is, no alignment information between individual regions and terms is available.

Therefore, a captioned image becomes an object with two set-valued attributes: "regions" and "terms". Since the regions and terms of an image are correlated, we propose to use MAGIC to detect this correlation and use it to predict the missing caption terms correlated with the uncaptioned test images.

The first step of MAGIC is to construct the MAGIC graph. Following the instructions for graph construction in Section 8.1, the graph for captioned images with attributes "regions" and "terms" will be a 3-layer graph with nodes for images, regions and terms. To form the NN-links, we define the distance function (Assumption 1) between two regions (tokens) as the $L_2$ norm between their feature vectors. Also, we define that two terms are similar if and only if they are identical, i.e., no term is any other's neighbor. As a result, there is no NN-link between term nodes.

For results shown in this section, the number of nearest neighbors between attribute/domain tokens is $k$=3. However, as we will show later in Section 11.1, the captioning accuracy is insensitive to the choice of $k$. In total, each data set has about 50,000 different region tokens and 160 words, resulting in a graph $G_{MAGIC}$ with about 55,500 nodes and 180,000 edges. The graph based on the core image set $I_{core}$ captures the correlations between regions and terms. We call such graph the *"core" graph*.

*How do we caption a new image, using the information in a MAGIC graph?* Similar to the core images, an uncaptioned image $I_{new}$ is also an object with set-valued attributes: "regions" and "caption", where attribute "caption" has null value. To find caption words correlated with image $I_{new}$, we propose to look at regions in the core image set that are similar to the regions of $I_{new}$, and find the words that are correlated with these core image regions. Therefore, our algorithm has two main steps: finding similar regions in the core image set (augmentation) and identifying caption words (RWR). Next, we define "core graph", "augmentation", and "gateway nodes", to facilitate the description of our algorithm.

**Definition 6 (Core Graph, Augmentation, and Gateway Nodes)** *For automatic image captioning, we define the **core** subgraph of $G_{MAGIC}$ be the subgraph that constitutes information in the given captioned images $I_{core}$, and is denoted as $G_{core}$. The graph $G_{MAGIC}$ for captioning a test image $I_{new}$ is an **augmented graph**, which is the core $G_{core}$ augmented with the region-nodes and image-node of $I_{new}$. The augmentation subgraph is denoted as $G_{aug}$, and hence the overall $G_{MAGIC}$=$G_{core} \cup G_{aug}$. The nodes in the core subgraph $G_{core}$ that are adjacent to the augmentation are called the **gateway** nodes, $\mathcal{GW}$.*

As an illustration, Figure 8.1 shows the graph $G_{MAGIC}$ for two core (captioned) images $I_{core}$=$\{I_1, I_2\}$ and one test (to-be-captioned) image $I_{test}$=$\{I_3\}$, with the parameter for NN-links $k$=1. The core subgraph $G_{core}$ contains region nodes $\{r_1, \ldots, r_7\}$, image nodes $\{I_1, I_2\}$, and all the term nodes $\{t_1, \ldots, t_8\}$. The augmentation $G_{aug}$ contains region nodes $\{r_8, \ldots, r_{11}\}$ and the image node $\{I_3\}$ of the test image. The gateway nodes are the region nodes $\mathcal{GW}$=$\{r_5, r_6, r_7\}$ that bridge the $G_{core}$ and $G_{aug}$.

Different test images have different gateway nodes, and therefore have different augmented graphs. However, since we will caption only one test image at a time, the symbols $G_{aug}$ and $\mathcal{GW}$ represent the augmented graph and the set of gateway nodes of the test image in question, respectively.

The first step of our image captioning algorithm, augmentation, can be done by finding the gateway nodes - the collection of the $k$ nearest neighbors of each region of $I_{new}$. In the second step, we propose to use RWR, restarting from the test image-node, to identify the correlated words (term-nodes). A predicted caption of $g$ words for the image $I_{new}$ will correspond to the $g$ term-nodes with highest RWR scores. Figure 9.1 gives the details of our algorithm.

To sum up, for image captioning, the core of the $G_{MAGIC}$ is first constructed based on the given captioned images $I_{core}$. Then, each test image $I_{new}$ is captioned, one by one, in steps summarized in Figure 9.1.

**Input:** 1. The core graph $G_{core}$, an image $I_{new}$ to be captioned, and
2. $g$, the number of caption words we want to predict for $I_{new}$.
**Output:** Predicted caption words for $I_{new}$.
**Steps:**
1. Augment the image node and region nodes of $I_{new}$ to the core graph $G_{core}$.
2. Do RWR from the image node of $I_{new}$ on the augmented graph $G_{MAGIC}$ (Algorithm 8.3).
3. Rank all term nodes by their RWR scores.
4. The $g$ top-ranked terms will be the output - the predicted caption for $I_{new}$.

**Figure 9.1:** Steps to caption an image, using the proposed MAGIC framework.

## 9.2 Captioning Performance

We evaluate the performance of MAGIC on image captioning by two measurements: the *captioning accuracy* and the *precision/recall of single-word query*. The captioning accuracy summarizes the correctness of a proposed method on predicting the human-annotated captions. The precision/recall of a single-word query measures the goodness of using the predicted captions as image indexes, for text-based image retrieval. These two measurements are commonly used to evaluate the captioning performance [25, 5, 30]. In this section, we will introduce the definition of these measurements and compare the performance of our method with other approaches [25, 5, 30].

**Captioning Accuracy** *Captioning accuracy* is defined as the fraction of terms which are correctly predicted. Following the same evaluation procedure as that in previous work [25, 5], for a test image which has $g$ ground-truth caption terms, MAGIC will also predict $g$ terms. If $p$ of the predicted terms are correct, then the captioning accuracy $acc$ on this test image is defined as

$$acc = \frac{p}{g}.$$

The average captioning accuracy $\overline{acc}$ on a set of $T$ test images is defined as

$$\overline{acc} = \frac{1}{T}\sum_{i=1}^{T} acc_i,$$

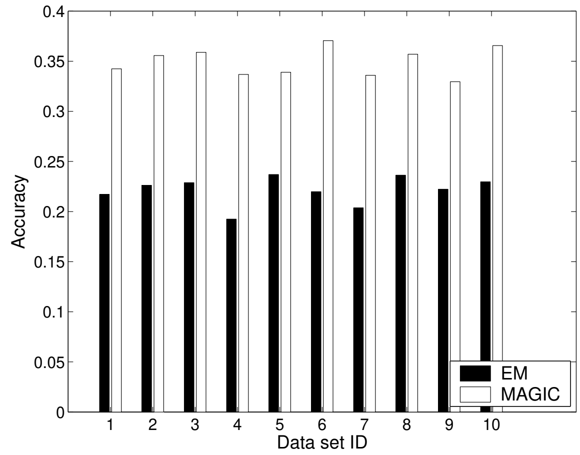



**Figure 9.2:** Comparing MAGIC with the EM method. The parameters for MAGIC are $c = 0.66$ and $k = 3$. The x-axis shows the 10 data sets, and the y-axis is the average captioning accuracy over all test images in a set.

where $acc_i$ is the captioning accuracy on the $i$-th test image.

Figure 9.2 shows the average captioning accuracy on the 10 image sets. We compare our results with those reported in [25]. The method in [25] is one of the most recent and sophisticated: it models the image captioning problem as a statistical translation modeling problem and solves it using expectation-maximization (EM). We refer to their method as the "EM" approach. The x-axis groups the performance numbers of MAGIC (white bars) and EM (black bars) on the 10 data sets. On average, MAGIC achieves captioning accuracy improvement of 12.9 percentage points over the EM approach, which corresponds to a relative improvement of 58%.

We also compare the captioning accuracy with even more recent machine vision methods [5], on the same data sets: the Hierarchical Aspect Models method ("HAM"), and the Latent Dirichlet Allocation model ("LDA"). Figure 9.3 compares MAGIC with LDA and HAM, in terms of the mean and variance of the average captioning accuracy over the 10 data sets. Although both HAM and LDA improve on the EM method, they both lose to our generic MAGIC approach: HAM and LDA achieve 29% and 25% average captioning accuracy, respectively, while MAGIC achieves a better accuracy at 35%. It is also interesting that MAGIC gives significantly lower variance, by

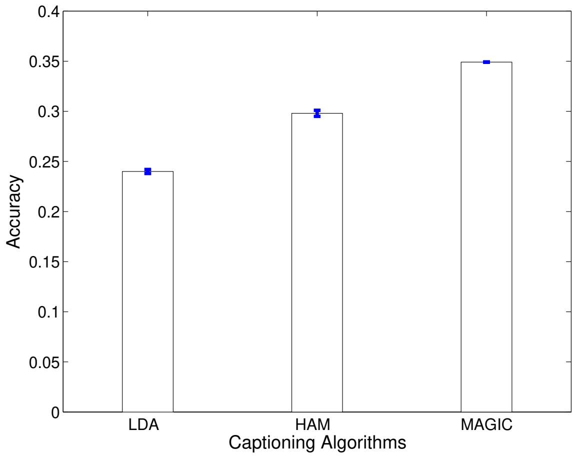



**Figure 9.3:** Comparing MAGIC with LDA and HAM. The mean and variance of the average accuracy over the 10 Corel data sets are shown at the y-axis - LDA: ($\mu$, $\sigma^2$)=(0.24,0.002); HAM: ($\mu$, $\sigma^2$)=(0.298,0.003); MAGIC: ($\mu$, $\sigma^2$)=(0.3503, 0.0002). $\mu$: mean average accuracy. $\sigma^2$: variance of average accuracy. The length of the error bars at the top of each bar is $2\sigma$.

| | (a) | (b) | (c) |
|---|---|---|---|
| Truth | cat, grass,<br>tiger, water | mane, cat,<br>lion, grass | sun, water,<br>tree, sky |
| MAGIC | grass, cat,<br>tiger, water | lion, grass,<br>cat, mane | tree, water,<br>buildings, sky |

**Figure 9.4:** Terms are ordered by their given importance. Figures look best in color.

roughly an order of magnitude: 0.002 versus 0.02 and 0.03. A lower variance indicates that the proposed MAGIC method is more robust to variations among different data sets.

Figure 9.4 shows some examples of the captions given by MAGIC. For the test image $I_3$ in Figure 7.1, MAGIC captions it correctly (Figure 9.4(a)). In Figure 9.4(b), MAGIC surprisingly gets the word “mane” correctly; however, it mixes up “buildings” with “tree” (Figure 9.4(c)).

**Precision/Recall of Single-word Query** The predicted caption words can be used as image indexes to facilitate text-based image retrieval. *Precision* and *recall* are two standard measurements for evaluating image retrieval performance. Given a query and the corresponding retrieval result, *precision* is defined as the fraction of retrieved images that are correct/relevant, and *recall* is defined as the fraction of correct/relevant images that are retrieved.

In our experiments, we consider single-word queries, that is, text queries that contain only one word. Our method MAGIC predicts caption words for each image. For a single-word query $q$, an image is retrieved if MAGIC captions the image with the query word $q$. For evaluation, the human-annotated captions are used as ground truth. In other words, an image is “relevant” to a single-word query $q$, if the image is annotated by human with the word $q$.

Therefore, for a single-word query, the precision is defined as $C/A$, and the recall is defined as $B/A$, where $A$ is the number of images that MAGIC captions with the query word, $B$ is the number of images that are “relevant”, and $C$ is the number of images that MAGIC correctly annotated with the query word. This definition of precision and recall is the same as that in previous work [25, 30].

We compare the performance of MAGIC with the EM method [25] and the CRM (Continuous-space Relevance Model) method [30]. The EM and CRM methods are designed particularly for the image captioning problem, while our MAGIC method is a general cross-modal correlation discovery tool. The image captioning problem is just one of the many applications on which MAGIC can be applied.

We follow the evaluation steps in the previous work: Each of the 260 possible caption words is used as a single-word query, and we report three measurements: (1) the number of queries (words) that have positive recall; (2) the mean recall and precision of the top 49 queries; and, (3) the mean recall and precision of all queries.

Table 9.1 lists the performance of the EM method, the CRM method, and our general MAGIC method. The parameters for MAGIC are: $K$=5 and $c$=0.65. MAGIC performs better than the EM

| | EM | CRM | MAGIC |
|---|---|---|---|
| # words with recall >0 | 49 | 107 | 61 |
| Results on 49 best words, as in [25, 30] | | | |
| Mean Per-word Recall | 0.34 | 0.70 | 0.54 |
| Mean Per-word Precision | 0.20 | 0.59 | 0.56 |
| Results on all 260 words | | | |
| Mean Per-word Recall | 0.04 | 0.19 | 0.07 |
| Mean Per-word Precision | 0.06 | 0.16 | 0.09 |

**Table 9.1:** Comparing MAGIC with the EM and CRM methods.

method, but does not have good recall as the CRM method. For the mean per-word precision of the top 49 queries, MAGIC achieves comparable performance to CRM method (56% to 59%).

Unlike the CRM method, which is designed specifically for the image captioning problem, our proposed method MAGIC can discover cross-modal correlations in general settings. That is, MAGIC not only can do image captioning by finding correlations between two modalities (image and text), and it can be easily applied to find correlations between more than two modalities and find any-to-any correlations (Section 7.1).

## 9.3 Generalization

MAGIC treats information from all media uniformly as nodes in a graph. Since all nodes are basically the same, we can do RWR and restart from any subset of nodes of any medium, to detect any-to-any medium correlations. The flexibility of our graph-based framework also enables novel applications, such as captioning images in groups (*group captioning*). In this subsection, we show results on (a) spotting the term-to-term correlation in image captioning data sets, and (b) group captioning.

| | Top 5 most correlated terms | | | | |
|---|---|---|---|---|---|
| Query term | 1 | 2 | 3 | 4 | 5 |
| branch | birds | night | owl | nest | hawk |
| bridge | water | arch | sky | stone | boats |
| cactus | saguaro | desert | sky | grass | sunset |
| car | tracks | street | buildings | turn | prototype |
| f-16 | plane | jet | sky | runway | water |
| market | people | street | food | closeup | buildings |
| mushrooms | fungus | ground | tree | plants | coral |
| pillars | stone | temple | people | sculpture | ruins |
| reefs | fish | water | ocean | coral | sea |
| textile | pattern | background | texture | designs | close-up |

**Table 9.2:** Correlated terms of sample query terms: correlated terms have related concepts.

**Beyond Image-to-Term Correlation** MAGIC successfully exploits the image-to-term correlation for captioning images. However, the MAGIC graph $G_{MAGIC}$ contains correlations between all media (image, region, and term). To show how well MAGIC works on objects of any medium, we design an experiment to identify correlated captioning terms, using the term-to-term correlation in the graph $G_{MAGIC}$.

We use the same 3-layer MAGIC core graph $G_{core}$ that was constructed in the previous subsection for automatic image captioning. Given a query term $t$, we use RWR to find other terms correlated with it. Specifically, we perform RWR, restarting from the query term(-node). The terms deemed correlated with the query term are term(-node)s that receive high RWR scores.

Table 9.2 shows the top 5 terms with the highest RWR scores for some query terms. In the table, each row shows the query term at the first column, followed by the top 5 correlated terms selected by MAGIC (sorted by their RWR scores). The selected terms make semantic sense, and have meanings related with the query term. For example, the term "branch", when used in image captions, is strongly related to forest- or bird- related concepts. MAGIC shows exactly this, correlating "branch" with terms such as "birds", "owl", and "nest".

A second, subtle observation, is that our method does not seem to be biased by frequent words. In our collection, the terms "water" and "sky" are more frequent than the others (like the terms "the" and "a" in normal English text). Yet, these frequent terms do *not* show up too often in Table 9.2, as a correlated term of a query term. It is surprising, given that we did nothing special when using MAGIC: no tf/idf weighting, no normalization, and no other domain-specific analysis. We just treated these frequent terms as nodes in our MAGIC graph, like any other nodes.

**Group Captioning** The proposed MAGIC method can be easily extended to caption a group of images, considering all of them at once. This flexibility is due to the graph-based framework of MAGIC, which allows augmentation of multiple nodes and doing RWR from any subset of nodes. To the best of our knowledge, MAGIC is the first method that is capable of doing group captioning.

**Application 3 (Group Captioning)** *Given a set $I_{core}$ of color images, each with caption words; and given a (query) group of uncaptioned images $\{I'_1, \ldots, I'_t\}$, find the best g (say, g=5) caption words to assign to the group.*

Possible applications for group captioning include video segment captioning, where a video segment is captioned according to the group of keyframes associated with the segment. Since keyframes in a segment are related, captioning them as a whole can take into account the inter-keyframe correlations, which are missed if each keyframe is captioned separately. Accurate captions for video segments may improve performances on tasks such as video retrieval and classification.

The steps to caption a group of images are similar to those for the single-image captioning outlined in Figure 9.1. A core MAGIC graph is still used to capture the mixed media information of the given collection of images. The differences for group captioning are, instead of augmenting the single-image to the core and restarting from it, now we augment all $t$ images in the query group $\{I'_1, \ldots, I'_t\}$ to the core, and restart randomly from one of the images in the group (i.e., each with probability $1/t$ to be the restart node).

Figure 9.5 shows the result of using MAGIC for captioning a group of three images. MAGIC found reasonable terms for the entire group of images: "sky", "water", "tree", and "sun". Captioning multiple images as a group takes into consideration the correlations between different images

| | (a) | (b) | (c) |
|---|---|---|---|
| Truth | sun, water, tree, sky | sun, clouds, sky, horizon | sun, water |
| MAGIC | tree, people, sky, water | water, tree, people, sky | sky, sun |
| Group | | sky, water, tree, sun | |

**Figure 9.5:** Group captioning: Captioning terms with highest RWR scores are listed first.

in the group, and in this example, this helps reduce the scores of irrelevant terms such as "people". In contrast, when we caption these images individually, MAGIC selects "people" as caption words for images in Figure 9.5(a) and (b), which do not contain people-related objects.

## 9.4 Summary

Given a collection of captioned images, our MAGIC method provides an intuitive, graph-based framework to represent the associations between image content and captioning terms. MAGIC uses the RWR technique (Figure 8.3) and successfully finds correlations between images and terms for image captioning. In particular, our experiments show the following properties of MAGIC:

- MAGIC achieves good captioning accuracy, and outperforms recent machine learning approaches by 58% on captioning accuracy (Figures 9.2 and 9.3).

- MAGIC helps in multiple other data mining tasks, for example, group captioning (Problem 3) and finding any-to-any medium correlations (Figure 9.2).

# Chapter 10

# Case Study: Mining News Videos

As more and more digital video libraries [127] become available, video summarization is in great demand for accessing these video collections efficiently. A video clip is essentially a mixed media object, containing video and audio streams, transcripts, and objects extracted from keyframes such as people's faces or overlaid text. In order to facilitate browsing and utilizing video clips in a large digital library that contains thousands of hours of video clips, it is desirable to have summaries of video clips.

Particularly for broadcast news video, summaries for individual news events could help users gain an overall picture of an event: "*how did it start?*" and "*what were the following developments?*" [85, 17, 49]. Usually, a news event is covered by news stories (or *shots*) scattered in daily broadcasts during a period of time. A summary of a news event, say "winter olympics", may consist of scene shots of various sport events, the names of relevant locations and players, as well as relevant topics such as "the tourism industry of the hosting country". Identifying event-relevant materials from different media in the video and collecting them together as a summary of an event would benefit users and find applications in areas such as education, decision making, and entertainment, etc.

Correlations across media in video provide strong clues for collecting summarization materials. News channels compile news shots by showing footage (images) correlated with the story the


winter olympics nagano ceremony night round competition result headline news superticker medal watch headline news superticker result
(a) "Winter Olympics"

money trie source lawyer house intern monica information counsel starr immunity learned starr
(b) "Lewinsky"

arab league secretarygeneral strike iraq reaction arab brent sadler report iraqi president saddam hussein kind sentiment attack
(c) "Iraq"

**Figure 10.1:** Three logo events, their keyframes (containing logos) and nouns in the transcripts.

anchorperson is reporting (audio/text). There are also correlations among video shots of a continuing event - the same scenes or symbols are repeatedly used in these correlated shots. Information about such correlations would facilitate gathering of information relevant to an event.

MAGIC provides a graph-based framework to find correlations among video shots, transcripts, and news events. In this section, we apply MAGIC to the problem of video summarization, and design experiments to examine two issues: (a) *what is the quality of summaries by MAGIC?* And, (b) *what advantages does MAGIC provide over traditional retrieval techniques?*

## 10.1 Event (Logo) Identification and Problem Formulation

A news video contains information in different media such as shots and transcripts. Besides shots and transcripts, in order to do event summarization, we need to identify the specific event of each shot, which is not a easy task. Particular difficulties include the definition of events and the identification of video shots of the same event.

Daily news reports usually consist of shots about different events, where these shots have various durations and are covered in different orders from day to day. Broadcast news programs usually show a small icon next to the anchorperson as the symbol for the story which the anchorperson is reporting at the time [26]. The same icon is commonly re-used in other shots of the same event, as an aid for viewers to link the present story to the past coverages. These icons are called *logos* and have been shown useful for defining events and linking stories of an event [26]. Figure 10.1 shows three example logos of stories "winter olympics", "Lewinsky", and "Iraq", extracted using an off-the-shelf method, the "iconic matching" method [27, 50].

In this study, the logos are used as identifiers of news events. The events that have corresponding logos are called *logo events*, and these are the events that will be considered in this work. Therefore, the formulation of our problem becomes: Given the shots, transcripts, and (event) logos from video clips, construct a *summary* for a (query) logo event.

## 10.2 Data Set and MAGIC Graph Construction

We conduct our experiments on the TRECVID 2003 data set, which contains 115 news clips (46 giga bytes) from a U.S. news source. We process video clips to extract the shots, transcripts and logos: Each news clip is segmented into individual video shots using an off-the-shelf algorithm [106]. A keyframe is extracted from each shot, and the logo is extracted from the keyframe if there exists one. We call a shot a *logo shot* if its keyframe contains a logo (we call such keyframes *logo frames*). The transcript words mentioned in a shot are also collected, but we use only nouns in our experiments. Since there is a one-to-one correspondence between a shot and its keyframe, in the following discussions, we will use a shot' keyframe to visualize (in figures) the content of the shot, and will use the words "shot" and "frame" interchangeably.

The data set becomes a collection of shots (objects) that each associates with zero or one logo, and a set of transcript words (nouns). In other words, a shot is an object with two attributes: "logo" and "transcript". The "logo" attribute has categorical values; however, many shots have this value missing. The "transcript" attribute is a set-valued attribute.

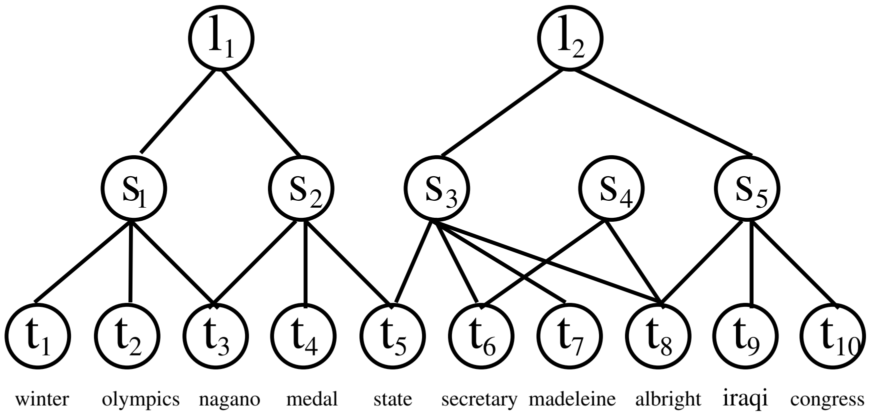



**Figure 10.2:** The MAGIC graph for broadcast news event summarization. There are three types of nodes: logo-nodes ($l_i$), shot-nodes ($s_i$), and term-nodes ($t_i$).

Although there are missing values and set-valued attributes, our proposed MAGIC method has no problem in representing these video shots by using our $G_{MAGIC}$ graph. Following the instructions in Section 8.1, the graph for this video data set contains three types of nodes: shot-nodes ($s_i$) for the shot objects, while logo-nodes ($l_i$) and term-nodes ($t_i$) are for the domain tokens. Figure 10.2 illustrates how the graph looks like on a set of 5 shots, 2 logos and 10 transcript nouns. The full graph of our data set contains 27,809 nodes (17,528 shot-nodes, 80 logo-nodes and 10,201 term-nodes) and 69,710 edges.

We also need to define the similarity functions between domain tokens (Assumption 1). In this study, we consider every logo or transcript noun to be unique. In other words, the similarity function between two "logo" tokens is defined as: 1 if the logos are identical, and 0 otherwise. The similarity function between two transcript nouns is defined similarly. Consequently, on the graph $G_{MAGIC}$, neither a "logo" token, nor a "transcript" token has neighbors - the graph only has OAV-links, but no NN-links (Subsection 8.1).

## 10.3 Cross-Modal Correlations for Event Summarization

Having identified the shots, transcripts, and (event) logos from video clips, we can now construct a summary for a (query) logo event. The idea is to collect shots, transcripts, and logos that are correlated with the query logo event as the event's summary.

We propose to use MAGIC and the graph $G_{MAGIC}$ that contains information of shots, transcripts, and logos (Figure 10.2) to approach this event summarization problem. To find the shots and terms correlated with the query logo event, we perform the random walk with restarts (RWR) from the corresponding query-logo-node. The shots and terms whose corresponding nodes have the highest RWR scores are selected for the summary of the query event.

We examine the contents of shots and terms selected by MAGIC, and evaluate whether they are relevant to the query event. Figure 10.3 shows the top 20 shots selected by MAGIC for the logo event "Iraq". We use the keyframe of a shot to present the content in a video shot. The top 7 shots correspond to logo shots which are directly connected to the query-logo-node. These are the logo shots identified by the logo extraction algorithm when we are building the graph $G_{MAGIC}$. Interestingly, MAGIC finds logo shots that are not detected by the logo extraction algorithm (e.g. the shot ranked 16). Informative shots, such as faces of major players involved in the query event, are selected by MAGIC. For example, Kofi Annan, the UN secretary-general, appears in the shots at rank 9, 11, 20, and William S. Cohen, the U.S. secretary of defense, in shots at rank 13, 15. Other logos pertaining to the query logo "Iraq" are also selected, for example, the "Yeltsin" logo at rank 14, which corresponds to the Russian president Yeltsin's involvement in the affairs with Iraq. The selection of shots for other query logos yields similar observations.

The terms that MAGIC selected for the event "Iraq" are shown in Table 10.1 (first row). The selected terms are relevant, showing the names of key players (e.g. 'Kofi Annan' and 'Albright'), the person's position ('secretary general') and relevant locations ('baghdad', 'sudan'), and activities ('(air) strike', 'weapon talk'), etc. On other events like "Winter Olympics" and "Lewinsky", MAGIC also successfully selected relevant terms that convey the content of the events (Table 10.1,


**Figure 10.3:** Keyframes of the selected shots for the query logo event "Iraq". Frames are sorted (highest RWR score first).

second and third rows).

In general, MAGIC is capable of finding meaningful shots and terms for an event. The selection of shots and terms can be done simultaneously (using RWR), in spite of the different medium types.

## 10.4 Any-to-Any Medium Correlation in Broadcast News

MAGIC finds more correlations other than the specific event-to-shot and event-to-term correlations used for event summarization in the previous section. Its capability of finding any-to-any medium correlations also gives us, for example, the term-to-shot and term-to-term correlations, and provides the opportunity for other multimedia applications. In this section, we show results of using

| Logo event | Summarizing terms |
|---|---|
| "Iraq" | iraq minister annan kofi effort baghdad report president arab strike defense sudan iraqi today weapon secretary talk school window problem there desk peter student system damage apart arnett albright secretarygeneral |
| "Winter Olympics" | winter medal gold state skier headline news result superticker olympics competition nagano ceremony watch night round game team sport weather photo woman that today canada bronze year home storm coverage |
| "Lewinsky" | house lawyer intern ginsburg starr bill whitewater counsel immunity president clinton monica source information money trie learned iraq today state agreement country client weapon force nation inspection courthouse germany support |

**Table 10.1:** Selected terms by MAGIC as summary for a logo event. Terms are sorted (highest score first).

the term-to-shot and term-to-term correlations for another application: *video retrieval with textual summary*.

**Application 4 (Video Retrieval with Textual Summary)** *Given a query of one or more terms, return a list of video shots relevant to the query, and furthermore, give a textual summary of the retrieved shot list.*

Query by text has always been an important mean for users to express their information needs, even when the targets are mixed media objects such as video clips. Having a list of shots relevant to a query is useful, if one is exploring the collection. However, sometimes it is desirable to summarize the retrieved shots, to provide a concise view of the retrieved query result, especially in situations where the user is not able to watch every video shot in the retrieved set. One possible solution is to summarize the retrieval result by a textual summary - a set of terms which are correlated with the retrieved shots.


Rank 1 Rank 2 Rank 3 Rank 4 Rank 5

Rank 6 Rank 7 Rank 8 Rank 9 Rank 10

**Figure 10.4:** Keyframes of the top 10 shots retrieved by MAGIC on the query {`` ``lewinsky'' ``, `` ``clinton'' ``}. Frames are sorted (highest RWR score first).

Traditional document retrieval methods usually give a list of relevant documents, but do not provide a summary of the retrieved result. In contrast, MAGIC provides a convenient framework to implement a video shot retrieval mechanism, and is able to provide a textual summary of the retrieved shots. Intuitively, we could use the cross-modal correlations detected by MAGIC to achieve video retrieval with textual summary: the term-to-shot correlation for shot retrieval, and the term-to-term correlation for textual summary.

We conduct video retrieval experiments on the same TRECVID 2003 video data set that we used for event summarization (Section 10.2). There are two steps to apply MAGIC: (1) graph construction and (2) RWR. Since the same data set is used, the $G_{MAGIC}$ graph for video retrieval is the same as that for the event summarization - a graph with three types of nodes (terms, shots, logos) (Figure 10.2). Given a textual query of one or more terms, we perform RWR from the corresponding query term-nodes and assigns RWR scores to all shots, logos and terms. The RWR scores will be used to select shots and terms as the retrieval result. Example queries used in our experiments are: {`` ``lewinsky'' ``}, {`` ``clinton'' ``}, {`` ``lewinsky'' ``, `` ``clinton'' ``}, {`` ``white'' ``, `` ``house'' ``, `` ``scandal'' ``}, {`` ``annan'' ``}, {`` ``iraq'' ``}, {`` ``annan'' ``, `` ``iraq'' ``}, {`` ``olympics'' ``}, etc.

| Method | Textual summary of the retrieval result |
|---|---|
| MAGIC | clinton lewinsky president monica attorney today house jury starr bill washington relationship story whitewater lawyer daughter jones counsel intern investigation immunity conversation headline minute ginsburg affair question judge mother office |
| OKAPI | clinton(16) lewinsky(11) monica(6) president(6) service(2) minute(2) report(2) blitzer(1) wolf(1) conversation(1) controversy(1) lewis(1) testimony(1) agent(1) nature(1) officer(1) claim(1) exchange(1) immunity(1) intern(1) house(1) affair(1) attorney(1) question(1) relationship(1) whitewater(1) headline(1) time(1) office(1) |
| LSI (using 50 singular vectors) | clinton(20) president(20) mandela(1) friend(1) congress(1) lady(1) visit(1) administration(1) relationship(1) washington(1) |

**Table 10.2:** Summarizing retrieval result of query {``lewinsky'', ``clinton''}. Numbers in the parentheses are frequency counts of terms in the top 30 shots retrieved by OKAPI or LSI. Terms are sorted (highest RWR scores or frequency counts first).

Figure 10.4 shows the shots retrieved by MAGIC for the query {``lewinsky'', ``clinton''}. The shots are ranked in the order of relevance depicted by the RWR scores. Shots containing scenes of key persons related to the query are ranked at the top of the list, for example, prosecutor Kenneth Starr at rank 1 and Monica Lewinsky at rank 4. Relevant logo stories are also found, for example, "Clinton investigation" at rank 3 and "Jordan" (a Democrat lawyer) at rank 5. In the figure, we use the shot keyframe as the surrogate of a shot, for visualization purpose. The textual summary for this query is shown in Table 10.2 (first row). Words in the summary (e.g., 'lawyer', 'whitewater', 'jones', 'affair') give users a good picture about the retrieved set for the query {``lewinsky'', ``clinton''} which are stories about a judicial investigation involving U.S. president Bill Clinton and the White House intern Monica Lewinsky on their relationship.

To evaluate our result, we compare MAGIC to two state-of-the-art document retrieval techniques: OKAPI [109] and the Latent Semantic Indexing (LSI) [20]. To apply document retrieval techniques to our domain, we consider each shot as a document of transcript words. The goal is to compare the quality of the retrieved shots.

Unlike MAGIC, which can do both shot retrieval and textual summary generation at the same time, OKAPI and LSI do only shot retrieval but not query result summarization. In fact, the topic of multi-document summarization is still an important, ongoing research area [36, 86]. Therefore, we design an approach to create a textual summary from the retrieval results by OKAPI or LSI: we collect the frequency histogram of all terms appear in the top *s* retrieved documents, and make the textual summary as the list of these terms, ranked by the frequency. In our experiments, we choose *s*=30, as human users seldom check more than 30 documents in a query result.

Table 10.2 compares the textual summaries by the three methods: MAGIC, OKAPI and LSI, for the query {``lewinsky'', ``clinton''}. We found that MAGIC's result is as good as that of OKAPI, and is more informative than that of LSI. Instead of focusing on details about ``lewinsky'' and ``clinton'', the textual summary by LSI has words about a broader concept of "politics in Washington": 'president', 'congress', 'washington', etc. This is because that LSI was designed to focus on the semantic "topics" of documents, rather than the specifics about a query.

Moreover, OKAPI and LSI are limited to the terms which appear in the top *s*=30 retrieved shots. Relevant terms that are not in the retrieved shots can not be found by OKAPI or LSI. For example, relevant terms "starr" and "jones" are selected by MAGIC, but not by OKAPI or LSI, because these two terms are not mentioned in the top 30 shots retrieved. On the other hand, our MAGIC method does not have this limitation – the graph framework of MAGIC provides an elegant way to select terms effectively.

## 10.5 Summary

Besides finding cross-modal correlations in among captioned images, MAGIC can be easily applied to other multimedia data set. In this chapter, we show the application of MAGIC on finding correlations in a collection of broadcast news videos. We focus on three types of data modalities:

the news logos (events), video shots (keyframe), and transcript terms. We make the following observations from our experiments:

- For a news logo (event), MAGIC can identify video shots and terms that are correlated with the event. The shots (keyframes) and terms correlated to an event can be used to create a meaningful, multi-modal summary of that particular event (Figure 10.3 and Table 10.1).

- MAGIC can find term-to-video correlations for video retrieval (Figure 10.4). Unlike traditional textual retrieval models which consider only the similarity between video clips via transcripts, MAGIC can take into consideration not just the similarity via transcripts, but also the similarity via any modality. Furthermore, the term-to-term correlations found by MAGIC, could be used to form a textual summary of the video retrieval result (Table 10.2).

# Chapter 11

# System Issues

MAGIC provides an intuitive framework for detecting cross-modal correlations. In previous chapters (Chapters 9 and 10), we showed results where MAGIC is successfully applied to automatic image captioning and news event summarization. In terms of computation, the RWR computation of MAGIC is fast in that it scales linearly with graph size (the number of nodes and edges). For example, in our experiments, a straightforward implementation of RWR, using iterative matrix multiplications (Figure 8.3), can caption an image in less than 5 seconds.

In this chapter, we discuss system issues on MAGIC, such as parameter configuration and fast computation. We are interested in issues such as: How does the performance of MAGIC change with the settings of parameters such as restart probability of RWR? How about the effects of graph structure, e.g., different number of NN-links, or different link weights? How do we further speedup the already-efficient RWR computation?

We use the performance on the image captioning problem (Chapter 9) to benchmark the effects of parameter configurations and fast computation techniques. In particular, we show that:

- MAGIC is insensitive to parameter settings, and
- MAGIC is insensitive to small variations/errors on the graph.

We also propose a technique which uses precomputation and approximation to speed up the

RWR computation for image captioning. With precomputation (a one-time cost), captioning a new image becomes a constant time operation. The approximation taken by our technique does not affect the captioning accuracy – the difference on average captioning accuracy over 10 data sets, compared to that of exact computation, is just 0.08 percentage points.

In fact, MAGIC is modular so that we can easily employ the best module to date to speedup MAGIC, for example, faster nearest-neighbor search methods for speeding up the identification of NN-links in the MAGIC graph, or advanced random walk methods for faster RWR computation.

## 11.1 Optimization of Parameters

There are several design decisions to be made when employing MAGIC for correlation detection: *what should be the values for the two parameters: the number of neighbors k of a domain token, and the restart probability c of RWR?* And, *should we assign weights to edges, according to the types of their end points?* In this section, we empirically show that the performance of MAGIC is insensitive to these settings, and provide suggestions on determining reasonable default values.

We use automatic image captioning as the application to measure the effect of these parameters. The experiments in this section are performed on the same 10 captioned image sets ("001", ..., "010") described in Section 9.1, and we measure how the values of these parameters effect the captioning accuracy.

**Number of Neighbors** $k$ The parameter $k$ specifies the number of nearest domain tokens to which a domain token connects via the NN-links (Section 8.1). With these NN-links, objects having little difference in attribute values will be closer to each other in the graph, and therefore, are deemed more correlated by MAGIC. For $k$=0, all domain tokens are considered distinct; for larger $k$, our application is more tolerant to the difference in attribute values.

We examine the effect of various $k$ values on image captioning accuracy. Figure 11.1 shows the captioning accuracy on the data set "006", with the restart probability $c$=0.66. The captioning

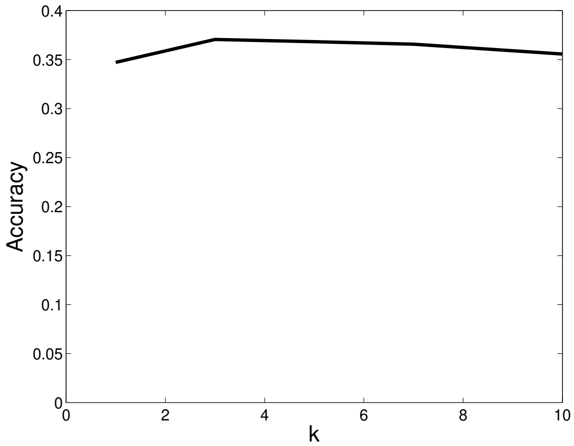



**Figure 11.1:** The plateau in the plot shows that the captioning accuracy is insensitive to value of the number of nearest neighbors $k$. Y-axis: Average accuracy over all images of data set "006". The restart probability is $c$=0.66.

accuracy increases as $k$ increases from $k$=1, and reaches a plateau between $k$=3 and 10. The plateau indicates that MAGIC is insensitive to the value of $k$. Results on other data sets are similar, showing a plateau between $k$=3 and 10.

In hindsight, with only $k$=1, the collection of regions (domain tokens) is barely connected, missing important connections and thus leading to poor performance on detecting correlations. At the other extreme, with a high value of $k$, everybody is directly connected to everybody else, and there is no clear distinction between really close neighbors or just neighbors. For a medium number of neighbors $k$, the NN-links apparently capture the correlations between the close neighbors, and avoid noise from remote neighbors. Small deviations from that value make little difference, which is probably because that the extra neighbors we add (when $k$ increases), or those we retained (when $k$ decreases), are at least as good as the previous ones.

**Restart Probability** $c$ The restart probability $c$ specifies the probability to jump back to the restarting node(s) of the random walk. Higher value of $c$ implies giving higher RWR scores to nodes closer in the neighborhood of the restart node(s).

Figure 11.2 shows the image captioning accuracy of MAGIC with different values of $c$. The

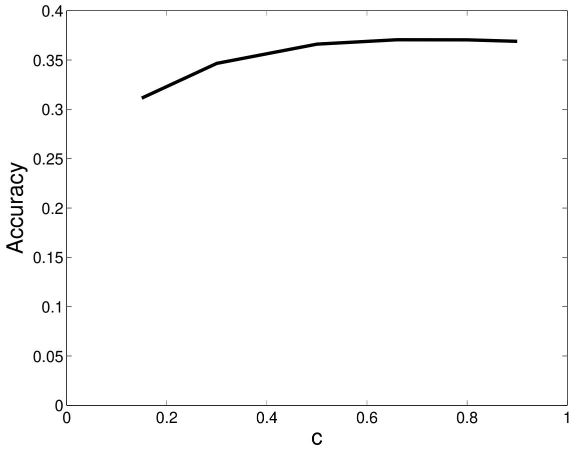



**Figure 11.2:** The plateau in the plot shows that the captioning accuracy is insensitive to value of the restart probability $c$. Y-axis: Average accuracy over all images of data set "006". The number of nearest neighbors per domain token is $k$=3.

data set is "006", with the parameter $k$=3. The accuracy reaches a plateau between $c$=0.5 and 0.9, showing that the proposed MAGIC method is insensitive to the value of $c$. Results on other data sets are similar, showing a plateau between $c$=0.5 and 0.9.

For web graphs, the recommended value for $c$ is typically $c$=0.15 [38]. Surprisingly, our experiments show that this choice does not give good performance. Instead, good quality is achieved for $c$=0.6 $\sim$ 0.9. What is the reason for this discrepancy?

We conjecture that what determines a good value for the restart probability is the diameter of the graph. Ideally, we want our random walker to have a non-trivial chance to reach the outskirts of the whole graph. If the diameter of the graph is $d$, the probability that the random walker (with restarts) will reach a point on the periphery is proportional to $(1-c)^d$.

For the web graph, the diameter is estimated to be $d$=19 [1]. This implies that the probability $p_{periphery}$ for the random walker to reach a node in the periphery of the web graph is roughly

$$p_{periphery} = (1-c)^{19} = (1-0.15)^{19} = 0.045 \ . \tag{11.1}$$

In our image captioning and event summarization experiments, we use graphs that have three layers of nodes (Figures 8.1 and 10.2). The diameter of such graphs is roughly $d$=3. If we demand

| | $w_{region}$ | | |
|---|---|---|---|
| $w_{term}$ | 0.1 | 1 | 10 |
| 0.1 | 0.370332 | 0.371963 | 0.370812 |
| 1 | 0.369900 | 0.370524 | 0.371963 |
| 10 | 0.368969 | 0.369181 | 0.369948 |

**Table 11.1:** Captioning accuracy is insensitive to various weight settings on OAV-links to the two media: region ($w_{region}$) and term ($w_{term}$).

the same $p_{periphery}$ as equation (11.1), then the $c$ value for our 3-layer graph would be

$$(1-0.15)^{19} = (1-c)^3 \tag{11.2}$$

$$\Rightarrow c = 0.65\ , \tag{11.3}$$

which is much closer to our empirical observations. Of course, the problem requires more careful analysis - but we are the first to show that $c$=0.15 is not always optimal for random walk with restarts.

**Link Weights** MAGIC uses a graph to encode the relationship between mixed media objects and their attributes of different media. The OAV-links in the graph connect objects to their domain tokens (Figure 8.1). To give more attention to an attribute domain $D$, we can increase the weights of OAV-links that connect to tokens of domain $D$. *Should we treat all media equally, or should we weight OAV-links according to their associated domains? How should we weight the OAV-links? Could we achieve better performance on weighted graphs?*

We investigate how the change on link weights influences image captioning accuracy. Table 11.1 shows the captioning accuracy on data set "006" when different weights are assigned on the OAV-links to regions (weight $w_{region}$) and those to terms ($w_{term}$). For all cases, the number of nearest neighbors is $k$=3 and the restart probability is $c$=0.66. The case where ($w_{region}$, $w_{term}$)=(1,1) is that of the unweighted graph, and is the result we reported in Chapter 9. As link weights vary from 0.1, 1 to 10, the captioning accuracy is *basically unaffected*. The results on other data sets are similar - captioning accuracy is at the same level on a weighted graph as on the unweighted graph.

This experiment shows that an unweighted graph is appropriate for our image captioning application. We speculate that an appropriate weighting for an application depends on properties such as the number of attribute domains (i.e., the number of layers in the graph), the average size of a set-valued attribute of an object (such as, average number of regions per image), and so on. We plan to investigate more on this issue in our future work.

## 11.2 Speeding up Graph Construction by Fast K-NN Search

The proposed MAGIC method encodes a mixed media data set as a graph, and employs the RWR algorithm to find cross-modal correlations. The construction of the MAGIC graph is intuitive and straightforward, and the RWR computation is light and linear to the data base size. One step which is relatively expensive is the construction of NN-links in a MAGIC graph.

When constructing the NN-links of a MAGIC graph, we need to compute the nearest neighbors for every domain token. For example, in our image captioning experiments (Section 9.1), to form the NN-links among region-nodes in the MAGIC graph, $k$-NN searches are performed 50,000 times (one for each region token) in the 30-dimensional region-feature space.

In MAGIC, the NN-links are proposed to capture the similarity relation among domain tokens. The goal is to associate tokens that are similar, and therefore, it could be suffice to have the NN-links connect to neighbors which are close enough, even if they are not exactly the closest ones. The approximate nearest neighbor search is usually faster, by trading accuracy for speed. The interesting questions are: *How much speedup could we gain by allowing approximate NN-links? How much is the performance reduction by approximation?*

For efficient nearest neighbor search, one common way is to use a spatial index such as R-tree [112], which give exact nearest neighbor in logarithmic time. Fortunately, MAGIC is modular and we can pick the best module to perform each step. In our experiments, we used the approximate nearest neighbor method (ANN) [2], which supports both exact and approximate nearest neighbor search. ANN estimates the distance to a nearest neighbor up to $(1+\varepsilon)$ times the actual distance:

| | ANN | | | Sequential search (SS) |
|---|---|---|---|---|
| | ε=0 | ε=0.2 | ε=0.8 | |
| Elapse time (msec.) | 3.8 | 2.4 | 0.9 | 46 |
| Speedup over SS | 12.1× | 19.2× | 51.1× | 1 |
| Error (in top *k*=10) | - | 0.0015% | 1.67% | - |
| Error (in top *k*=3) | - | - | 0.46% | - |

**Table 11.2:** Computation/approximation trade off in the NN-link construction among image regions. The distance to a neighboring point is approximated to within (1+ε) times the actual distance. ε=0 indicates the exact k-NN computation. Elapse time: average wall clock time for one nearest neighbor search. Speedup: the ratio of elapse time, with respect to the time of sequential search (SS). Error: the percentage of mistakes made by approximation in the *k* nearest neighbors. The symbol "-" means zero error.

$\varepsilon = 0$ means exact search, no approximation; bigger ε values give rougher estimation.

Table 11.2 lists the average wall clock time to compute the top 10 neighbors of a region in the 10 Corel image sets of our image captioning experiments. Compared to sequential search, the speedup of using a spatial method increases from 12.1 to 51.1, from exact search to a rough approximation of $epsilon = 0.8$. For the top *k*=3 nearest neighbors (the setting used in our experiments), the error percentage is at most 0.46% for the roughest approximation, equivalent to making one error in every 217 NN-links. The sequential search method is implemented in C++, and is compiled with the code optimization (`g++ -O3`).

The small differences on NN-links do not change the characteristic of the MAGIC graph significantly, and has limited affect on the performance of image captioning. At ε=0.2, when considering only $k = 3$ nearest neighbors, no error is made on the NN-links in the MAGIC graph, and therefore the captioning accuracy is the same as exact computation. At ε=0.8, the average captioning accuracy decreases by just 1.59 percentage point (for $k = 3$ nearest neighbors), averaged over the 10 Corel image sets (Figure 11.3).

Recently, there are other sub-linear time methods for approximate nearest neighbor search [79]

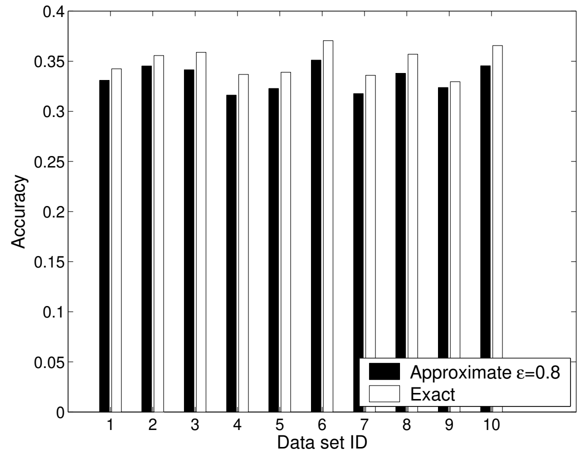



**Figure 11.3:** Using approximate NN-links (*epsilon*=0.8) reduces captioning accuracy by just 1.59% on the average. X-axis: 10 data sets. Y-axis: average captioning accuracy over test images in a set. The parameters for MAGIC are $c = 0.66$ and $k = 3$.

being developed. Thanks to the modularity of MAGIC, we can easily leverage advances in these works to further improve the efficiency and performance of MAGIC.

## 11.3 Precomputation for Fast RWR Computation

As outlined in Figure 8.3, our implementation of RWR is fast already (linear to the database size). Nevertheless, the computation of the RWR scores can be even further accelerated. In this section, we discuss approaches for fast RWR computation and introduce our approach for speeding up image captioning by precomputation.

**RWR on Sparse Graph** Our proposed MAGIC graph $G_{MAGIC}$, by construction, has a structure of layers (Figure 10.2). Edges exist only between certain nodes, making the graph $G_{MAGIC}$ a *sparse* graph. In fact, only a small fraction of all possible edges are in the graph. Figure 11.4 shows the adjacency matrix of the MAGIC graph for broadcast news event summarization in Chapter 10. The adjacency matrix is sparse and shows block structures corresponding to the entities involved: logos, shots, and terms.

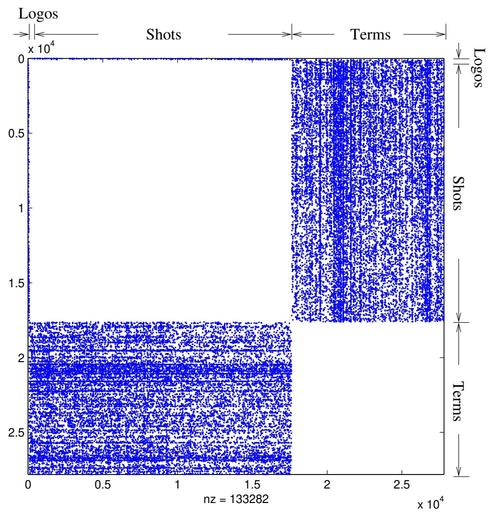



**Figure 11.4:** The adjacency matrix of the MAGIC graph for broadcast news event summarization (Chapter 10). The graph has three types of nodes, for logos, shots, and terms. A dot at position $(i, j)$ indicates that there is an an edge between the $i$-th and $j$-th node. Graph statistics: 27,809 nodes and 133,282 edges. The graph is sparse, only 0.17% of all possible edges are connected.

From Equation 8.1 (Section 8.2), it is easy to show that

$$\vec{\mathbf{u}}_q = c\,(\mathbf{I} - (1-c)\mathbf{A})^{-1}\,\vec{\mathbf{v}}_q \tag{11.4}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix, $\vec{\mathbf{v}}_q$ is the restart vector, and $\vec{\mathbf{u}}_q$ is the vector of RWR scores on all $N$ nodes. To solve for the RWR scores $\vec{\mathbf{u}}_q$, we can take advantage of the sparse structure in the adjacency matrix $\mathbf{A}$, and tap the old and recent literature of fast solutions to sparse linear systems [105], to efficiently solve the matrix inversion in equation (11.4).

Recently, there are also methods for fast random walk computation which exploit the block-structure in a graph [38], or adapting the computation to the convergence behavior [56]. Given that this area is still under intensive research [51, 57], we only point out that our MAGIC approach for correlation discovery is modular, and it can trivially include whichever is the best module for fast RWR computation.

**Speeding up Image Captioning by Precomputation** In image captioning, MAGIC first builds a core graph (Definition 6) according to the given captioned images. This core graph is then used to caption thousands of new images, one at a time, using RWR on the augmented graph (Figure 9.1). As there are many images to be captioned in practice, fast RWR score computation is desirable.

Before introducing the proposed method to speedup image captioning, it is helpful to review how MAGIC captions an image. Let $q$ be the node of the image which we want to caption. To caption image $q$, we first take the core graph $G_{core}$ and augment the image-node and region-nodes of image $q$ to $G_{core}$, via the gateway nodes $\mathcal{GW}$, to get the augmented MAGIC graph $G_{MAGIC}$. Then, we compute the RWR scores $\vec{\mathbf{u}}_q$, with the restart vector $\vec{\mathbf{v}}_q$ - a vector of all zeros, except one 1 for node $q$. (Figure 9.1 gives the detail algorithm of captioning an image.) The vector $\vec{\mathbf{u}}_q$ satisfies equation (8.1), which is reproduced here in equation (11.5) for the reader's convenience.

$$\vec{\mathbf{u}}_q = (1-c)\mathbf{A}\vec{\mathbf{u}}_q + c\vec{\mathbf{v}}_q. \tag{11.5}$$

Captioning different test images will use different $G_{MAGIC}$ graphs (and therefore, different column-normalized adjacency matrices, $\mathbf{A}$), due to the different augmentation subgraphs they have. However, since the specific augmentation for each test image is small (with only about 10 nodes) compared to the core graph $G_{core}$, the augmented graph $G_{MAGIC}$ (matrix $\mathbf{A}$) of each test image is similar to the core graph (the matrix $\mathbf{A_{core}}$).

Whenever we caption a new image, we solve a version of equation (11.5) with a slightly different $\mathbf{A}$. However, since all matrices $\mathbf{A}$ share the same $\mathbf{A_{core}}$, *how could we make use of this observation? Can we precompute some information from* $\mathbf{A_{core}}$*, to speedup the captioning of a new image?*

We propose a method, *PRECOM*, which uses precomputation to speedup the captioning of thousands of images. Before introducing our proposed method (*PRECOM*), we need more definitions. For a test image $q$, suppose that the set of gateway nodes connecting $q$ and graph $G_{core}$ is $\mathcal{GW}$ and has size $z = |\mathcal{GW}|$. Let $\vec{\mathbf{u}}_{\mathcal{GW}}$ denote the vector of RWR scores, when restarting from the

**Input:** 1. The core graph $G_{core}$, an image $I_{new}$ to be captioned, and
2. $g$, the number of caption words we want to predict for $I_{new}$.
**Output:** Predicted caption words for $I_{new}$.
**Steps:**
0. (One-time cost) Precompute the RWR score vector $\vec{\mathbf{u}}_i$ of each blob-node on $G_{core}$.
1. Identify the gateway nodes $\mathcal{GW}$ of $I_{new}$.
2. Compute the PRECOM score vector $\vec{\mathbf{u}}_{\mathcal{GW}}$, with parameter $c$=0.65 (using Lemma 1).
3. Rank all term nodes by their PRECOM scores.
4. The $g$ top-ranked terms will be the output - the predicted caption for $I_{new}$.

**Figure 11.5:** Steps to caption an image, using the proposed PRECOM method.

gateway nodes on the core graph $G_{core}$. That is, $\vec{\mathbf{u}}_{\mathcal{GW}}$ satisfies the following equation:

$$\vec{\mathbf{u}}_{\mathcal{GW}} = (1-c)\mathbf{A_{core}}\vec{\mathbf{u}}_{\mathcal{GW}} + c\vec{\mathbf{v}}_{\mathcal{GW}}, \tag{11.6}$$

where $\vec{\mathbf{v}}_{\mathcal{GW}}$ is the restart vector, whose $i$-th element is $\frac{1}{z}$ if node $i \in \mathcal{GW}$, and is 0, otherwise.

The idea of PRECOM is based on the resemblance between equations (11.5) and (11.6), as well as that between matrices $\mathbf{A}$ and $\mathbf{A_{core}}$. The precomputation is a one-time cost, and all subsequent captioning can be done very efficiently in constant O(1) time. The basic ideas are:

- Approximate $\vec{\mathbf{u}}_q$ by $\vec{\mathbf{u}}_{\mathcal{GW}}$ (the PRECOM scores).
- $\vec{\mathbf{u}}_{\mathcal{GW}}$ can be computed in constant time, using precomputed information (explained next).

In the following, we first outline the algorithm of the PRECOM method, followed by discussions on how the PRECOM scores ($\vec{\mathbf{u}}_{\mathcal{GW}}$) can be computed in constant time using precomputed results (Lemma 1). And then, we empirically show that $\vec{\mathbf{u}}_{\mathcal{GW}}$ is approximately the same as $\vec{\mathbf{u}}_q$ (scores used by MAGIC on captioning).

Figure 11.5 gives the algorithm of PRECOM. PRECOM replaces the RWR computation at steps 1 and 2 of the original algorithm in Figure 9.1, with precomputation and an O(1) time captioning step.

The information that PRECOM precomputes is the RWR score vector $\vec{\mathbf{u}}_i$ of every region-node $i$ in the core graph $G_{core}$. Let $\vec{\mathbf{v}}_i$ be the restart vector with all elements 0, except the $i$-th element, which is 1. The RWR score vector $\vec{\mathbf{u}}_i$, restarting with respect to $\vec{\mathbf{v}}_i$ on the core graph, satisfies

$$\vec{\mathbf{u}}_i = (1-c)\mathbf{A_{core}}\vec{\mathbf{u}}_i + c\vec{\mathbf{v}}_i. \tag{11.7}$$

Also, $\vec{\mathbf{v}}_{\mathcal{GW}}$ and $\vec{\mathbf{v}}_i$ are related as follows:

$$\vec{\mathbf{v}}_{\mathcal{GW}} = \frac{1}{z}\sum_{i\in\mathcal{GW}} \vec{\mathbf{v}}_i, \tag{11.8}$$

where $z = |\mathcal{GW}|$ is the number of the gateway nodes in set $\mathcal{GW}$.

If we precompute the RWR score vector $\vec{\mathbf{u}}_i$ for every node $i$ in the core graph, Lemma 1 says that the $\vec{\mathbf{u}}_{\mathcal{GW}}$ vector of a test image $q$ is a linear sum of the precomputed $\vec{\mathbf{u}}_i$'s, and can be computed in constant time. In the following, we will refer to $\vec{\mathbf{u}}_{\mathcal{GW}}$ as the *PRECOM score vector* for a test image $q$. Figure 11.6 gives the proof of Lemma 1.

**Lemma 1** *Let $z = |\mathcal{GW}|$ be the size of the set of gateway nodes ($\mathcal{GW}$), then*

$$\vec{\mathbf{u}}_{\mathcal{GW}} = \frac{1}{z}\sum_{i\in\mathcal{GW}} \vec{\mathbf{u}}_i. \tag{11.9}$$

**Captioning Accuracy of PRECOM** We would empirically show that the proposed PRECOM predicts similar captioning terms as MAGIC does. Basically, we compare the orderings of the terms according to scores $\vec{\mathbf{u}}_{\mathcal{GW}}$ by PRECOM and $\vec{\mathbf{u}}_q$ by MAGIC, and show that the two methods give similar term orderings.

We assign rank 1 to the term with the highest score, rank 2 to the next, and so on. Each term gets two ranks: one according to the scores $\vec{\mathbf{u}}_{\mathcal{GW}}$ from PRECOM, and one based on $\vec{\mathbf{u}}_q$ (from MAGIC). In Figure 11.7(a), we plot the ranks of terms by PRECOM (Y-axis) against those by MAGIC (X-axis), for an image shown in Figure 9.4(b) (the image with two lions). Specifically, a point at location ($i$,$y_i$) corresponds to the rank $i$-th term dictated by MAGIC, which has rank

**Proof:**

1. We want to show that equation (11.9) is consistent with equation (11.6).
   We do this by showing that equation (11.6) still holds, after substituting equation (11.9) into it.
2. After substitution, the left hand side of (11.6) is
   LHS = $\frac{1}{z}\sum_{i\in\mathcal{GW}}\vec{\mathbf{u}}_i$,
   and the right hand side is
   RHS = $(1-c)\mathbf{A_{core}}\frac{1}{z}\sum_{i\in\mathcal{GW}}\vec{\mathbf{u}}_i + c\vec{\mathbf{v}}_{\mathcal{GW}}$.
3. Substitute equation (11.8) into RHS, we have
   RHS = $(1-c)\mathbf{A_{core}}\frac{1}{z}\sum_{i\in\mathcal{GW}}\vec{\mathbf{u}}_i + c\frac{1}{z}\sum_{i\in\mathcal{GW}}\vec{\mathbf{v}}_i$
   = $\frac{1}{z}\sum_{i\in\mathcal{GW}}((1-c)\mathbf{A_{core}}\vec{\mathbf{u}}_i + c\vec{\mathbf{v}}_i)$
   = $\frac{1}{z}\sum_{i\in\mathcal{GW}}\vec{\mathbf{u}}_i$ (By equation 11.7)
   = LHS. ■

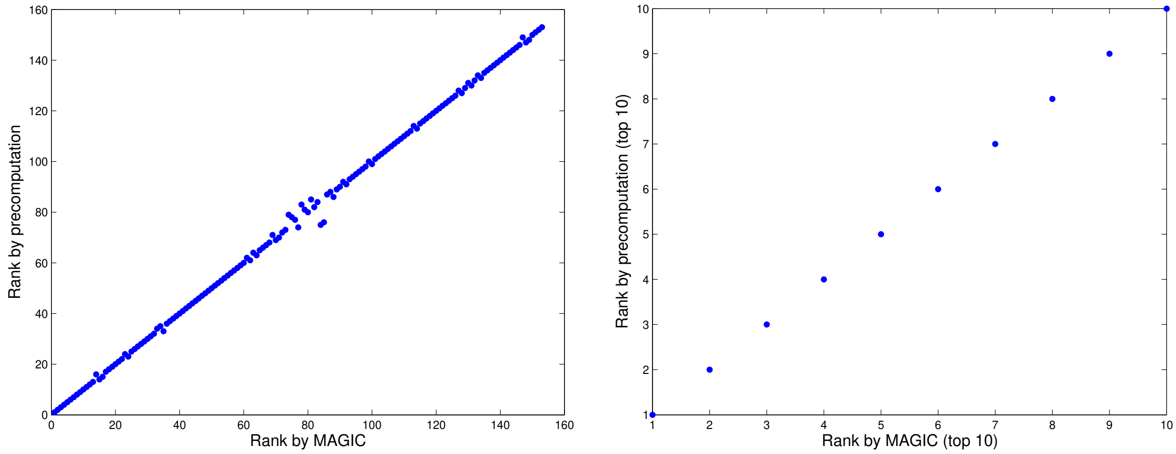
**Figure 11.6:** Proof of Lemma 1

$y_i$ according to PRECOM. A perfect diagonal pattern in the figure means no difference between PRECOM and MAGIC. We can see that the two methods give roughly the same ranking on terms - the data points are along the 45$^o$ diagonal line.

The two methods, PRECOM and MAGIC, agree on the rankings of the terms in general. Especially, they agree on the top-ranked words, which are our predicted caption words. The ranking of the top-ranked words are shown at Figure 11.7(b), which is an expanded view of the bottom-left corner of Figure 11.7(a). In fact, for this image, the captions from the two methods are the same.

To summarize the overall captioning differences between PRECOM and MAGIC, we compute the sum of absolute rank difference in the top 5 terms which, intuitively, indicates the total amount of rank reshuffling between the two methods. On average, the top 5 terms are reshuffled by 2. In other words, a top term dictated by MAGIC remains a top ranked term predicted by PRECOM, with the ranks differing by less than 2 for most of the query images. Therefore, in most cases, the two methods will give the same caption terms to an image.

Figure 11.8 shows the captioning accuracy of MAGIC and PRECOM on our 10 Corel image sets. The average difference on captioning accuracy over the 10 sets is just about 0.08%. Therefore, speeding up image captioning by precomputation does not compromise the captioning accuracy of




(a) Comparing the term rankings (b) Rankings of predicted caption terms

**Figure 11.7:** Comparing the rankings of terms according to the RWR scores on term-nodes from PRECOM and MAGIC. A point corresponds to a term; the point's location is defined by (x,y)=(rank according to MAGIC, rank according to PRECOM). (a): ranks of all terms; (b): ranks of the top 10 terms. The query image is the image of two lions shown at Figure 9.4(b). A $45^o$ diagonal line pattern means PRECOM and MAGIC give similar rankings on terms.

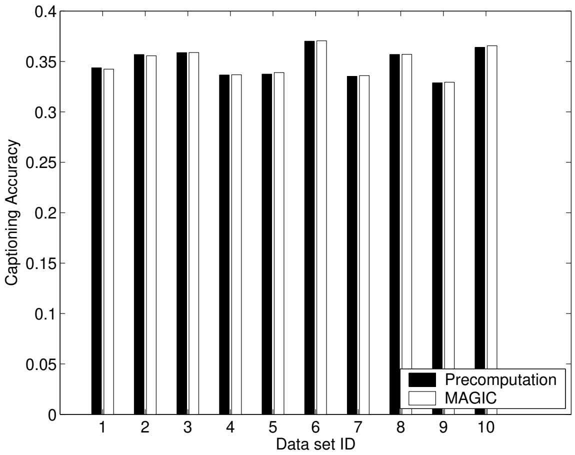



**Figure 11.8:** The proposed method PRECOM achieves the same captioning accuracy as MAGIC.

PRECOM: PRECOM achieves the same captioning accuracy as MAGIC.

| Graph ID | # total nodes (image-nodes, word-nodes, region-nodes) | # total edges |
|---|---|---|
| 1 | 10355 (1000, 113, 9242) | 67210 |
| 2 | 20997 (2000, 129, 18868) | 137024 |
| 3 | 31437 (3000, 138, 28299) | 204904 |
| 4 | 41858 (4000, 148, 37710) | 273294 |
| 5 | 52235 (5000, 152, 47083) | 341226 |

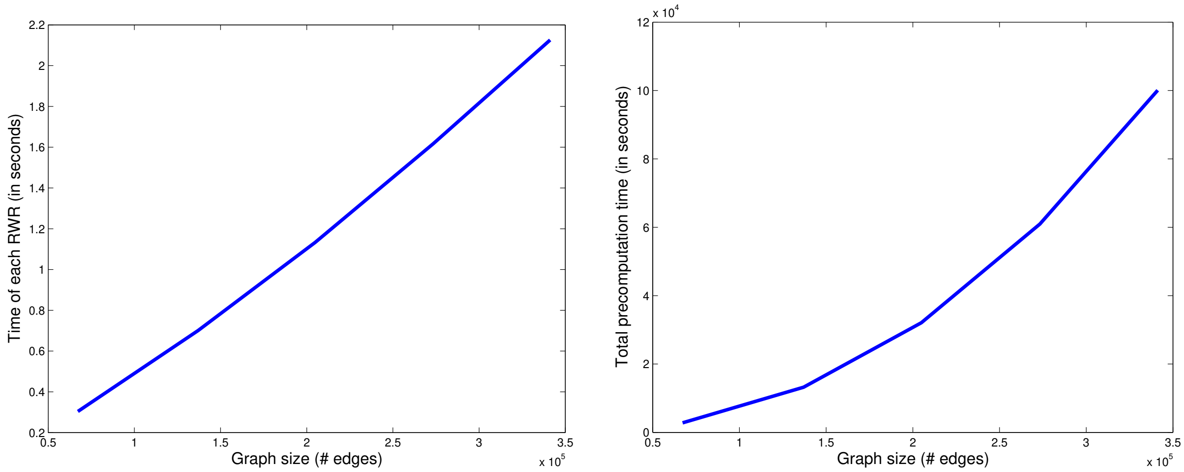
**Table 11.3:** Statistics of the 5 graphs used in the scalability experiment.

**Scalability of PRECOM** PRECOM precomputes the RWR score vector $\vec{\mathbf{u}}_i$ of each blob-node $i$ on the core graph $G_{core}$ (Step 0 in Figure 11.5). To examine the scalability of PRECOM, we measure the cost of this one-time precomputation cost, under various graph sizes. In particular, we measure (a) the average time to compute a RWR score vector $\vec{\mathbf{u}}_i$, and (b) the total one-time precomputation cost, and study how these measures change with the number of edges in the graph.

The graphs we used in this scalability experiment are the MAGIC graph of subsets of images from the "001" captioned image set (Chapter 9, Section 9.1). The count of images varies from 1000, 2000, to 5000, resulting in 5 graphs that have different counts of nodes and edges. An illustration of such a 3-layer graph is shown in Figure 8.1 (Chapter 9, Section 8.1). Figure 11.3 shows the statistics of these graphs. Since the graphs are sparse, we will use the number of edges in the group as the indicator of the *graph size*.

Figure 11.9(a) shows the average wall-clock time of computing a RWR score vector $\vec{\mathbf{u}}_i$, with respect to a region $i$, on graphs of various sizes. The figure shows that the computation of $\vec{\mathbf{u}}_i$ scales linearly with the graph size (the number of edges in the graph). This makes sense since each RWR computation (algorithm at Figure 8.3) requires about the same number of matrix multiplications (due to the large value $c$=0.65), where the matrix is the normalized adjacency matrix of the graph and depends on the graph size.

Figure 11.9(b) shows the total precomputation cost on graphs of various sizes. The precomputation computes the RWR score vector $\vec{\mathbf{u}}_i$ for every region node $i$, and therefore, besides the cost of computing a $\vec{\mathbf{u}}_i$, the total precomputation cost also depends on the number of region nodes in the




(a) Time of each RWR vs. graph size (b) Total precomputation time vs. graph size

**Figure 11.9:** Scalability of the PRECOM method: (a) average time to compute a RWR score vector $\vec{\mathbf{u}}_i$, and (b) the total time of precomputation. X-axis: graph size (the number of graph edges). The graphs used in this experiment are 5 subgraphs of the graph $G_{MAGIC}$based on the "001" captioned image set (Chapter 9).

graph. As we can observe from the numbers in Table 11.3, the total number of regions also grows linearly with the number of graph size. Since both the number of regions and the cost of RWR from a node grow linearly with graph size, the total precomputation time grows quadratically with graph sizes.

# Chapter 12

# Summary

Mixed media objects such as captioned images or video clips have attributes of different media (image, text, or audio). Detecting correlations and patterns across different media is useful for tasks such as imputation of missing media values, and has many applications such as image captioning (predicting missing caption words of an image). In this work, we developed MAGIC, a graph-based method for detecting cross-media correlations in a mixed media data set.

There are two challenges in detecting cross-media correlations, namely, representation of set-valued attributes and the detection of correlations between any medium and all media (the any-to-any medium correlation). MAGIC adopts a graph-based model which provides a natural way to represent set-valued attributes. The proposed model accommodates problems such as missing values or the non-aligned, noisy set elements, with no extra effort. In addition, the graph-based framework of MAGIC can find any-to-any medium correlations, using the technique of random walk with restarts (RWR).

MAGIC provides cross-media correlations which can be applied in many multimedia applications. In this study, we applied MAGIC on two applications: automatic image captioning and news event summarization (Applications 1 and 2).

For image captioning, MAGIC spots the correlations between images and caption words, and applies the correlations to caption new images. On the same benchmark, MAGIC outperforms previous methods on captioning accuracy (up to 58% relative improvement) (Chapter 9). Moreover, MAGIC is able to spot correlations between video shots, logos, and transcript terms, and gives meaningful news event summarization (Chapter 10). Although some video shots do not have logo information, MAGIC deals with these missing values smoothly.

Furthermore, the graph framework of MAGIC is versatile, giving any-to-any medium correlations which enable other applications such as group captioning (Problem 3) and video shot retrieval (Problem 4). In fact, to the best of our knowledge, MAGIC is the first attempt for group captioning which is useful for applications such as video segment captioning.

Technically, MAGIC has the following desirable characteristics:

- It is domain independent: The $Sim_i(*,*)$ similarity functions (Assumption 1) completely isolate our MAGIC method from the specifics of an application domain, and make MAGIC able to detect correlations in all kinds of mixed media data sets.

- It requires no fine-tuning on parameters or link weights: The performance is not sensitive to the two parameters - the number of neighbors $k$ and the restart probability $c$, and it requires no special weighting scheme like tf/idf for link weights (Section 11.1).

- It is fast and scales up well with the database/graph size. We also proposed a constant-time method (PRECOM) for speeding up image captioning, using precomputation (Section 11.3).

- It is modular and can easily tap recent advances in related areas to improve performance (Section 11.2).

We are pleasantly surprised that such a domain-independent method, with no parameters to tune, managed to outperform some of the most recent and most carefully tuned methods for automatic image captioning. Future work could further exploit the promising connection between multimedia databases and graph algorithms, including outlier detection and any other data mining task that requires the discovery of correlations as its first step.

# Part III

# Conclusions

# Chapter 13

# Conclusions

As more and more large multimedia databases of images, video clips, or biomedical data become available, new techniques are needed to make multimedia information accessible and useful. Such techniques should be able to analyze multimedia content and extract understandable and meaningful patterns. The meaningful patterns would be useful in applications such as classification or retrieval, as well as data mining tasks like summarizing into rules the representative and distinguishing characteristics of the data.

Multimedia objects like video clips contain data of different modalities, such as image, audio, and transcript text. In a multimedia database, there are two types of patterns: the *uni-modal patterns* that involve only data of one modality, and the *cross-modal correlations* that associate two or more different modalities. In this thesis, we focus on two problems: uni-modal pattern discovery and cross-modal correlation discovery in multimedia databases.

For uni-modal pattern discovery (Part I), we proposed an algorithm, AutoSplit, which can find patterns in various types of data modality, including video frames, audio, text, time sequences, and biomedical images (Chapter 3). AutoSplit uses independent component analysis (ICA) as a tool, and provides steps for mining and interpreting uni-modal patterns. In particular, AutoSplit can find patterns (*basis vectors*) that capture the characteristics of the data distribution, and the *hidden variables* that compose the observed data (Figure 3.7).

We showed the effectiveness of AutoSplit on a wide variety of settings:

- In a collection of video clips, AutoSplit can find spatial-temporal patterns in video frames, and characteristic patterns in audio, as well as the topics in the transcript text. The patterns found are consistent with our daily experiences on broadcast news video, and are useful in applications such as classification or segmentation (Chapter 4). Using the patterns we found in video frames (VideoBasis) and in audio (AudioBasis), experiments on classifying news and commercials gives 81% accuracy.

- For co-evolving time sequences such as stock prices or the motion capture data, AutoSplit can find the hidden variables that influence the observed sequences. For example, on stock price sequences, AutoSplit found the general trend and the "Internet bubble" phenomenon (Chapter 5).

- Using the idea in AutoSplit, we built a system (ViVo) for mining biomedical image databases (Chapter 6). Our system can automatically construct a visual vocabulary (which we called ViVos) that are biological meaningful and can properly summarize the characteristics of an image. With the ViVos, we could classify bio-medical images from 9 conditions with 83% accuracy. We also proposed several data mining functionalities that are enabled by the ViVos, such as highlighting the representative regions of an image, and identify regions that distinguish two images of different conditions.

Cross-modal correlations combine information from multiple data modalities and are useful in multimedia applications ranging from summarization to semantic captioning. For discovering cross-modal correlations, we proposed a graph-based method which is called MAGIC (Part II).

MAGIC turns the multimedia problem into a graph problem, by representing multimedia data as a graph (Chapter 8). Using the "random walk with restarts" on the graph, MAGIC can find correlations among all modalities. In particular, MAGIC has the following desirable properties:

- It gives an intuitive way to incorporate information from multiple modalities.

- It is domain independent: Attributes of any modality can be incorporated in the MAGIC graph. All MAGIC needs is a similarity function between the domain tokens of a modality (Chapter 8, Assumption 1).
- It can find correlations among all modalities incorporated in the graph, which we called the *any-to-any correlation discovery*.
- It requires no fine-tuning on parameters or link weights: The performance is insensitive to the parameters (Figures 11.1 and 11.2), and we suggest ways to determine the default values for the parameters.
- Its computation scales linearly with respect to the graph size (equivalently, linear to the database size).
- It is modular and can easily tap recent advances in related areas to improve performance (Chapter 11, Section 11.2).

We applied MAGIC to two applications, namely, automatic image captioning (Chapter 9) and event summarization in broadcast news video (Chapter 10).

On automatic image captioning, by finding robust correlations between text and image, MAGIC achieves a relative improvement of 58% in captioning accuracy as compared to recent machine learning techniques (Figure 9.3). Furthermore, the MAGIC framework enables novel data mining applications, such as *group captioning* where multiple images are captioned simultaneously, taking into account the possible correlations between the multiple images in the group (Figure 9.5).

MAGIC has also been applied to create multi-modal summary of a news event (Chapter 10). In a collection of broadcast news video, we used MAGIC to identify video shots and transcript words that are relevant to a news event. A multi-modal summary for a news event can be created by collecting relevant video shots and transcript words. Besides, MAGIC can find the correlations between words and video shots, which could be used as an alternative method for multi-modal retrieval (Section 10.4).

## 13.1 Future Work

The rich content in multimedia data provides challenges and opportunities for data mining. In this thesis, we proposed methods for pattern discovery and data mining in multimedia data: AutoSplit for uni-modal pattern discovery and MAGIC for cross-modal pattern discovery. In the future, we consider three research directions: extending the capability of our tools, applying our proposed tools to solve data mining problems in other domains, and exploring new techniques for multimedia mining.

**Extending the Capability of Existing Tools** There are several interesting extensions of AutoSplit and MAGIC that we could explore. Interesting extensions of AutoSplit include exploring the applications of *overcomplete ICA* [70] and the *mixture of ICA* [99, 68]. The overcomplete ICA computes more basis vectors (patterns), more than the dimensionality of the observed data (the amount of observed attributes). For example, if a motion is composed by three hidden variables, but we are only given two sequences of measurements (say at the left and right knees), with the overcomplete capability, AutoSplit would still be able to discover the three hidden variables, even only two measurement sequences are given. The mixture of ICA can find multiple sets of patterns (basis vectors) in a data set, and will be suitable for finding non-Gaussian clusters in real-world data sets.

MAGIC enables the application of graph algorithms to multimedia problems. Given a MAGIC graph which represents the multi-modal information of a multimedia database, numerous graph algorithms could be applied to extract rich information in multimedia databases. For example, algorithms such as graph partitioning or community identification which find strongly correlated subgraphs, could give more clues about correlations in multimedia databases.

**Applying Existing Tools to Other Domains** In areas such as biomedicine and cyber-security, there is an increasing need to analyze the huge amount of data being generated everyday: for finding patterns and summarization, for abnormality detection, and for knowledge/rule extraction. For example, in biomedicine, the data such as patient records or examination results could be in various modalities, such as text, audio, 2D/3D image, temporal-3D image, microarray, time sequences, etc. In these cases, new algorithms are needed to find patterns in these data. Similarly, in cyber-security, effective algorithms for analyzing data, such as user logs or traffic in networked environments, could help the development of failure/invasion monitoring and detection systems.

As stepping stones toward the solutions, we could first extend the applications of our existing tools to these domain. For example, in the biomedical domain, possible applications include: AutoSplit for finding patterns in microarrays, or detecting outliers in biological time sequences; ViVo for automated analysis and screening of 2D/3D/temporal biomedical images; and MAGIC for associating the multi-modal information in patient records, or the correlation analysis in biological networks.

As for the cyber-security domain, possible applications of our existing tools include abnormality detection in system logs using AutoSplit, or the correlation analysis of traffic between networked systems using MAGIC.

In fact, our proposed tools, AutoSplit and MAGIC, are general and can be used in many other areas, such as WWW, sensor network, motion animation, finance, or software engineering, where time series and graphs are also the major data types.

**Exploring Other Methods for Mining Multimedia Data** To achieve better performance in content-based retrieval, and the goal of understanding multimedia data, recent research has focused on finding mid-/high- level features that are associated with concepts that human used in perception and reasoning.

Our work reported in this thesis shows that by employing the tool ICA, AutoSplit can find meaningful and helpful patterns in multimedia data. Technically, ICA gives a meaningful decomposition of a data matrix into two matrices, the hidden matrix and the basis matrix, which contain the information about patterns in the data.

Recently, there have been development of other matrix decomposition techniques, for example, the semidiscrete matrix decomposition [60] or the non-negative matrix factorization [64]. One direction of future work could be exploring the properties of these methods, and study their ability on mining multimedia data.

Graph-based methods often represent a graph as a matrix (e.g., the adjacency matrix). The relations between matrix-based methods and graph-based methods are worth further exploration, for bridging the tools in the two domains and creating novel applications.

Similar to ICA, clustering is also an unsupervised approach to find patterns in data. As we have discussed in Chapter 2, most clustering algorithms implicitly assume data to be Gaussian. Recent developments in clustering algorithms have placed more attention on methods that find correlated clusters [12] that are not Gaussian. Since real-world data is rarely Gaussian and always contains noise, developing a correlation clustering algorithm which is robust to noise could be an interesting future work, and could have valuable applications in multimedia mining.

# Bibliography


[1] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Diameter of the world wide web. *Nature*, 401:130–131, 1999.

[2] S. Arya, David M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891–923, 1998.

[3] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface (GI 2004)*, 2004.

[4] Horace B. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989.

[5] Kobus Barnard, Pinar Duygulu, David A. Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[6] Marian Stewart Bartlett, H. Martin Lades, and Terrence J. Sejnowski. Independent component representations for face recognition. In *Proceedings of the SPIE Conference on Human Vision and Electronic Imaging*, volume III, January 1998.

[7] Anthony J. Bell and Terrence J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.

[8] Ana B. Benitez and Shih-Fu Chang. Multimedia knowledge integration, summarization and evaluation. In *Proceedings of the 2002 International Workshop on Multimedia Data Mining in conjuction with the International Conference on Knowledge Discovery and Data Mining (MDM/KDD-2002), Edmonton, Alberta, Canada, July 23-26*, 2002.

[9] Adam L. Berger and Vibhu O. Mittal. OCELOT: A system for summarizing web pages. In *Proceedings of SIGIR*, 2000.

[10] Arnab Bhattacharya, Vebjorn Ljosa, Jia-Yu Pan, Mark R. Verardo, Hyungjeong Yang, Christos Faloutsos, and Ambuj K. Singh. ViVo: Visual vocabulary construction for mining biomedical images. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM)*, 2005.

[11] David M. Blei and Michael I. Jordan. Modeling annotated data. In *Proceedings of the 26th ACM SIGIR Conference*, July 28-August 1, 2003, Toronto, Canada.

[12] Christian Böhm, Karin Kailing, Peer Kröger, and Arthur Zimek. Computing clusters of correlation connected objects. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, pages 455–466, 2004.

[13] Michael V. Boland, Mia K. Markey, and Robert F. Murphy. Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images. *Cytometry*, 3(33):366–375, 1998.

[14] Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.

[15] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[16] Shih-Fu Chang, R. Manmatha, and Tat-Seng Chua. Combining text and audio-visual features in video indexing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005), Philadelphia, PA*, March 2005.

[17] Michael G. Christel, Alexander G. Hauptmann, Howard D. Wactlar, and Tobun D. Ng. Collages as dynamic summaries for news video. In *Proceedings of the Tenth ACM International Conference on Multimedia*, pages 561–569, December 2002.

[18] Ian Davidson and S. S. Ravi. Clustering under constraints: Feasibility issues and the *k*-means algorithm. In *Proceeding of the 5th SIAM Data Mining Conference*, 2005.

[19] Arjen P. de Vries, Thijs Westerveld, and Tzveta Ianeva. Combining multiple representations on the TRECVID search task. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, volume 3, pages 1052–1055, May 2004.

[20] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–497, 1990.

[21] Nevenka Dimitrova, Lalitha Agnihotri, and Gang Wei. Video classification based on hmm using text and faces. In *Proceedings of the ACM Conference on Multimedia*, 2000.

[22] Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*, volume 22. The Mathematical Association of America, 1984.

[23] Bruce A. Draper, Kyungim Baek, Marian Stewart Bartlett, and J. Ross Beveridge. Recognizing faces with PCA and ICA. *Computer Vision and Image Understanding*, 91:115–137, 2003.

[24] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. New York: Wiley, 2nd edition, 2000.

[25] Pinar Duygulu, Kobus Barnard, Nando de Freitas, and David A. Forsyth. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In *Proceedings of the Seventh European Conference on Computer Vision (ECCV)*, volume 4, pages 97–112, 2002.

[26] Pinar Duygulu, Jia-Yu Pan, and David A. Forsyth. Towards auto-documentary: Tracking the evolution of news stories. In *Proceedings of the ACM Multimedia Conference*, 2004.

[27] Jaety Edwards, Ryan White, and David A. Forsyth. Words and pictures in the news. In *HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data*, May 2003.

[28] Christos Faloutsos. *Searching Multimedia Databases by Content*. Kluwer Academic Publishers Group, The Netherlands, August 1996.

[29] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and Will Equitz. Efficient and effective querying by image content. *Journal of intelligent information systems*, 3(3-4):231–262, July 1994.

[30] S. L. Feng, R. Manmatha, and Victor Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of the International Conference on Pattern Recognition (CVPR 2004)*, volume 2, pages 1002–1009, June 2004.

[31] Stephan Fischer, Rainer Lienhart, and Wolfgang Effelsberg. Automatic recognition of film genres. In *Proceedings of the 3rd ACM International Multimedia Conference and Exhibition*, 1995.

[32] Steven K. Fisher, Geoffrey P. Lewis, Kenneth A. Linberg, and Mark R. Verardo. Cellular remodeling in mammalian retina: Results from studies of experimental retinal detachment. *Progress in Retinal and Eye Research*, 24:395–431, 2005.

[33] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.

[34] Glenn Fung, Sathyakama Sandilya, and R. Bharat Rao. Rule extraction from linear support vector machines. In *Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–40, 2005.

[35] Andreas Girgensohn and Jonathan Foote. Video classification using transform coefficients. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3045–3048, 1999.

[36] Jade Goldstein, Vibhu O. Mittal, Jaime Carbonell, and Jamie Callan. Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the Ninth International Conference on Information Knowledge Management (CIKM-00)*, November 2000.

[37] Masafumi Hamamoto, Hiroyuki Kitagawa, Jia-Yu Pan, and Christos Faloutsos. A comparative study of feature vector-based topic detection schemes for text streams. In *In Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration (WIRI)*, pages 125–130, April 2005.

[38] Taher Haveliwala, Sepandar Kamvar, and Glen Jeh. An analytical comparison of approaches to personalizing pagerank. Technical Report 2003-35, Stanford University, June 2003.

[39] Taher H. Haveliwala. Topic-sensitive pagerank. In *WWW2002*, May 7-11 2002.

[40] Pablo Hennings, Jason Thornton, Jelena Kovačević, and B.V.K.V. Kumar. Wavelet packet correlation methods in biometrics. *Applied Optics, Special Issue on Biometric Recognition Systems*, 44(5):637–646, February 2005.

[41] Patrik O. Hoyer and Aapo Hyvärinen. Independent component analysis applied to feature extraction from colour and stereo images. *Network: Computation in Neural Systems*, 11(3):191–210, 2000.

[42] Winston Hsu, Lyndon Kennedy, Chih-Wei Huang, Shih-Fu Chang, Ching-Yung Lin, and Giridharan Iyengar. News video story segmentation using fusion of multi-level multi-modal

features in TRECVID 2003. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), Montreal, Canada*, May 2004.

[43] Winston H. Hsu and Shih-Fu Chang. Visual cue cluster construction via information bottleneck principle and kernel density estimation. In *Proceedings of the International Conference on Content-Based Image and Video Retrieval (CIVR)*, November 2005.

[44] Yanhua Hu and Robert F. Murphy. Automated interpretation of subcellular patterns from immunofluorescence microscopy. *Journal of Immunological Methods*, 290:93–105, 2004.

[45] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 762–768, 1997.

[46] Aapo Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[47] Aapo Hyvärinen. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[48] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. John Wiley and Sons, 2001.

[49] Ichiro Ide, Hiroshi Mo, and Norio Katayama. Threading news video topics. In *Proceedings of the Fifth ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 239–246, 2003.

[50] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *Proceedings of SIGGRAPH 95*, pages 277–286, August 1995.

[51] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th World Wide Web Conference*, 2003.

[52] Jiwoon Jeon, Victor Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *26th Annual International ACM SIGIR Conference*, July 28-August 1, 2003, Toronto, Canada.

[53] Jiwoon Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *Proceedings of CIVR 2004*, pages 24–32, 2004.

[54] Rong Jin, Joyce Y. Chai, and Luo Si. Effective automatic image annotation via a coherent language model and active learning. In *Proceedings of the 12th ACM International Conference on Multimedia, New York, NY, USA*, pages 892 – 899, October 2004.

[55] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002.

[56] Sepandar D. Kamvar, Taher H. Haveliwala, and Gene H. Golub. Adaptive methods for the computation of pagerank. In *Proceedings of the International Conference on the Numerical Solution of Markov Chains (NSMC)*, September 2003.

[57] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Extrapolation methods for accelerating pagerank computation. In *Proceedings of the 12th World Wide Web Conference*, 2003.

[58] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[59] Alan J. Klockars and Gilbert Sax. *Multiple Comparisons*. Sage Publications, Inc., 1986.

[60] Tamara G. Kolda and Dianne P. O'Leary. A semidiscrete matrix decomposition for latent semantic indexing informmation retrieval. *ACM Transactions on Information Systems*, 16(4):322–346, 1998.

[61] Fernando De la Torre and Michael J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54:183–209, August-October 2003.

[62] Fernando De la Torre and Takeo Kanade. Multimodal oriented discriminant analysis. In *Proceeding of the 22nd International Conference on Machine Learning (ICML)*, pages 177–184, August 2005.

[63] Ivan Laptev and Tony Lindeberg. Space-time interest points. In *Proceedings of ICCV*, pages I:432–439, 2003.

[64] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October.

[65] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. In *SIGGRAPH 2002*, July 2002.

[66] Jong-Hwan Lee, Ho-Young Jung, Te-Won Lee, and Soo-Young Lee. Speech feature extraction using independent component analysis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, June 2000.

[67] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transection of Pattern Analysis and Machine Intelligence*, 18(10):959–971, October 1996.

[68] Te-Won Lee and Michael S. Lewicki. The generalized gaussian mixture model using ica. In *International Workship on Independent Component Analysis*, pages 239–244, June 2000.

[69] Michael S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363, April 2002.

[70] Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[71] Geoffrey P. Lewis, Charanjit S. Sethi, Kenneth A. Linberg, David G. Charteris, and Steven K. Fisher. Experimental retinal detachment: A new perspective. *Mol. Neurobiol.*, 28(2):159–175, October 2003.

[72] Geoffrey P. Lewis, Kevin C. Talaga, Kenneth A. Linberg, Robert L. Avery, and Steven K. Fisher. The efficacy of delayed oxygen therapy in the treatment of experimental retinal detachment. *Am. J. Ophthalmol.*, 137(6):1085–1095, June 2004.

[73] Dongge Li, Nevenka Dimitrova, Mingkun Li, and Ishwar K. Sethi. Multimedia content processing through cross-modal association. In *Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA*, pages 604–611, 2003.

[74] Jia Li and James Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):14, 2003.

[75] Rainer Lienhart, Christoph Kuhmunch, and Wolfgang Effelsberg. On the detection and recognition of television commercials. In *Proceedings of the International Conference on Multimedia Computing and Systems*, pages 509–516, 1996.

[76] Joo-Hwee Lim. Categorizing visual contents by matching visual “keywords”. In *Proceedings of VISUAL ’99*, pages 367–374, 1999.

[77] Wei-Hao Lin and Alexander Hauptmann. News video classification using svm-based multimodal classifiers and combination strategies. In *Proceedings of the 10th ACM International Conference on Multimedia, Juan Les Pins, France*, October 2002.

[78] Fang Liu and Rosalind W. Picard. Finding periodicity in space and time. In *Proceedings of the International Conference on Computer Vision*, 1998.

[79] Ting Liu, Andrew Moore, Alexander Gray, and Ke Yang. An investigation of practical approximate nearest neighbor algorithms. In *Proceedings of the NIPS conference (NIPS 2004)*, December 2004.

[80] Zhu Liu, Jincheng Huang, and Yao Wang. Classification of TV programs based on audio information using hidden markov model. In *Proceedings of the IEEE Second Workshop on Multimedia Signal Processing (MMSP'98)*, pages 27–31, December 1998.

[81] László Lovasz. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:353–398, 1996.

[82] Wei-Ying Ma and B. S. Manjunath. A texture thesaurus for browsing large aerial photographs. *Journal of the American Society for Information Science*, 49(7):633–648, 1998.

[83] B.S. Manjunath, Philippe Salembier, and Thomas Sikora. *Introduction to MPEG-7*. Wiley, 2002.

[84] Oded Maron and Aparna Lakshmi Ratan. Multiple-instance learning for natural scene classification. In *The Fifteenth International Conference on Machine Learning*, 1998.

[85] Mark Maybury and Andrew Merlino. Multimedia summaries of broadcast news. In *Proceedings of the International Conference on Intelligent Information Systems*, December 1997.

[86] Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassloglou, Regina Barzilay, and Eleazar Eskin. Towards multidocument summarization by reformulation: progress and prospects. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence (AAAI/IAAI)*, pages 453–460, 1999.

[87] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[88] Aleksandra Mojsilović, Jelena Kovačević, Jianying Hu, Robert J. Safranek, and S. Kicha Ganapathy. Matching and retrieval based on the vocabulary and grammar of color patterns. *IEEE Transactions on Image Processing*, 9(1):38–54, 2000.

[89] Yasuhide Mori, Hironobu Takahashi, and Ryuichi Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[90] Robert F. Murphy. Automated interpretation of protein subcellular location patterns: Implications for early cancer detection and assessment. *Annals N.Y. Acad. Sci.*, 1020:124–131, 2004.

[91] Robert F. Murphy, Meel Velliste, and Gregory Porreca. Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images. *Journal of VLSI Signal Processing*, 35:311–321, 2003.

[92] Milind R. Naphade, Igor Kozintsev, and Thomas Huang. Probabilistic semantic video indexing. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, 2001.

[93] Tian-Tsong Ng, Shih-Fu Chang, Jessie Hsu, and Lexing Xieand Mao-Pei Tsui. Physics-motivated features for distinguishing photographic images and computer graphics. In *Proceedings of the 2005 ACM Multimedia Conference*, November 2005.

[94] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. Automatic video summarization by graph modeling. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*, 2003.

[95] Bruno A. Olshausen and David J. Field. Wavelet-like receptive fields emerge from a network that learns sparse codes for natural images. *Nature*, 381:607–609, 1996.

[96] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall, 2nd edition, 1998.

[97] Christopher R. Palmer and Christos Faloutsos. Electricity based external similarity of categorical attributes. In *PAKDD 2003*, May 2003.

[98] Jia-Yu Pan and Christos Faloutsos. VideoCube: a novel tool for video mining and classification. In *Proceedings of the Fifth International Conference on Asian Digital Libraries (ICADL 2002)*, 2002.

[99] Jia-Yu Pan, Hiroyuki Kitagawa, Christos Faloutsos, and Masafumi Hamamoto. AutoSplit: Fast and scalable discovery of hidden variables in stream and multimedia databases. In *Proceedings of the The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2004)*, 2004.

[100] Jia-Yu Pan, Hyung-Jeong Yang, Pinar Duygulu, and Christos Faloutsos. Automatic image captioning. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME 2004)*, 2004.

[101] Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, and Pinar Duygulu. Automatic multimedia cross-modal correlation discovery. In *Proceedings of the 10th ACM SIGKDD Conference*, 2004.

[102] Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, and Pinar Duygulu. GCap: Graph-based automatic image captioning. In *Proceedings of the 4th International Workshop on Multimedia Data and Document Engineering (MDDE 04), in conjunction with Computer Vision Pattern Recognition Conference (CVPR 04)*, 2004.

[103] Jia-Yu Pan, Hyungjeong Yang, and Christos Faloutsos. MMSS: Multi-modal story-oriented video summarization. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM 04)*, 2004.

[104] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV'98)*, volume 2, pages 555–562, January 4-7 1998.

[105] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipe in C.* Cambridge University Press, 1992.

[106] Yanjun Qi, Alex Hauptmann, and Ting Liu. Supervised classification of video shot segmentation. In *Proceedings of IEEE Conference on Multimedia & Expo (ICME'03)*, July 2003.

[107] Matthew J. Roach and John S. Mason. Video genre classification using audio. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EuroSpeech)*, 2001.

[108] Matthew J. Roach, John S. Mason, and Mark Pawlewski. Video genre classification using dynamics. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2001.

[109] S.E. Robertson and S. Walker. Okapi/Keenbow at trec-8. In *Proceedings of The Eighth Text REtrieval Conference (TREC 8)*, pages 151–162, 1998.

[110] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62, 1999.

[111] Nicu Sebe, Michael S. Lew, X. Zhou, T.S. Huang, and E. Bakker. The state of the art in image and video retrieval. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR'03)*, pages 1–8, July 2003.

[112] Timos K. Sellis, Nick Roussopoulos, and Christos Faloutsos. The R+-tree: A dynamic index for multi-dimensional objects. In *Proceedings of the 12th International Conference on VLDB*, pages 507–518, September 1987.

[113] Kim Shearer, Chitra Dorai, and Svetha Venkatesh. Local color analysis for scene break detection applied to tv commercials recognition. In *Proceedings of the 3rd International*

*Conference on Visual Information and Information Systems (VISUAL'99)*, pages 237–244, June 1999.

[114] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[115] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the 11th International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.

[116] John R. Smith and Shih-Fu Chang. Transform features for texture classification and discrimination in large image databases. In *Proceedings of the IEEE International Conference on Image Processing (ICIP-94)*, volume 3, pages 407–411, November 1994.

[117] John R. Smith and Shih-Fu Chang. Joint adaptive space and frequency basis selection. In *Proceedings of the 1997 International Conference on Image Processing (ICIP'97)*, volume 3, pages 702–705, 1997.

[118] Michael A. Smith and Takeo Kanade. Video skimming and characterization through the combination of image and language understanding techniques. In *Proceedings of CVPR 1997*, pages 775–781, 1997.

[119] Rohini K. Srihari, Aibing Rao, Benjamin Han, Srikanth Munirathnam, and Xiaoyun Wu. A model for multimodal information retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo, 2000 (ICME 2000)*, volume 2, pages 701–704, July 2000.

[120] Michael J. Swain and Dama H. Ballard. Indexing via color histograms. In *Proceedings of the Third International Conference on Computer Vision*, December 1990.

[121] Ba Tu Truong, Svetha Venkatesh, and Chitra Dorai. Automatic genre identification for content-based video categorization. In *Proceedings of the International Conference Pattern Recognition*, volume 4, pages 230–233, 2000.

[122] Matthew A. Turk and Alex P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–96, 1991.

[123] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, pages 293–302, 2002.

[124] Shingo Uchihashi, Jonathan Foote, Andreas Girgensohn, and John Boreczky. Video Manga: Generating semantically meaningful video summaries. In *Proceedings of ACM MM 1999*, 1999.

[125] J. H. van Hateren and D. L. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proc. Royal Society Lond. B*, 265:2315–2320, 1998.

[126] Paola Virga and Pinar Duygulu. Systematic evaluation of machine translation methods for image and video annotation. In *Proceedings of The Fourth International Conference on Image and Video Retrieval (CIVR 2005), Singapore*, July 2005.

[127] Howard D. Wactlar, Michael G. Christel, Yihong Gong, and Alexander G. Hauptmann. Lessons learned from the creation and deployment of a terabyte digital video library. *IEEE Computer*, 32(2):66–73, February 1999.

[128] Xin-Jing Wang, Wei-Ying Ma, Gui-Rong Xue, and Xing Li. Multi-model similarity propagation and its application for web image retrieval. In *Proceedings of the 12th ACM International Conference on Multimedia, New York, NY, USA*, pages 944–951, October 2004.

[129] Liu Wenyin, Susan Dumais, Yanfeng Sun, HongJiang Zhang, Mary Czerwinski, and Brent Field. Semi-automatic image annotation. In *INTERACT2001, 8th IFIP TC.13 Conference on Human-Computer Interaction*, Tokyo, Japan July 9-13, 2001.

[130] Thijs Westerveld and Arjen P. de Vries. Multimedia retrieval using multiple examples. In *Proceedings of the Third International Conference on Image and Video Retrieval (CIVR 2004)*, volume 3115, pages 344–352, July 2004.

[131] Yi Wu, Edward Y. Chang, Kevin Chen-Chuan Chang, and John R. Smith. Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th ACM International Conference on Multimedia, New York, NY, USA*, pages 572–579, October 2004.

[132] Lexing Xie, Lyndon Kennedy, Shih-Fu Chang, Ajay Divakaran, Huifang Sun, and Ching-Yung Lin. Layered dynamic mixture model for pattern discovery in asynchronous multimodal streams. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2005), Philadelphia*, March 2005.

[133] Eric P. Xing, Andrew Y. Ng, Michael .I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Proceeding of the Advances in Neural Information Processing Systems 16 (NIPS)*, pages 521–528, 2002.

[134] Boon-Lock Yeo and Minerva M. Yeung. Retrieving and visualizing video. *Communications of the ACM*, 40(12):43–52, December 1997.

[135] Lihi Zelnik-Manor and Michal Irani. Event-based analysis of video. In *Proceedings of CVPR*, pages II:123–130, 2001.

[136] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, January 2004.

[137] Dong-Qing Zhang, Ching-Yung Lin, Shih-Fu Chang, and John R. Smith. Semantic video clustering across sources using bipartite spectral clustering. In *Proceeding of IEEE Conference of Multimedia and Expo, 2004 (ICME 2004)*, June 2004.

[138] Zhongfei Zhang, Ruofei Zhang, and Jun Ohya. Exploiting the cognitive synergy between different media modalities in multimodal information retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo, 2004 (ICME 2004)*, volume 3, pages 2227–2230, June 2004.

[139] Rong Zhao and William I. Grosky. Bridging the semantic gap in image retrieval. In T. K. Shih, editor, *Distributed Multimedia Databases: Techniques and Applications*, pages 14–36. Idea Group Publishing, Hershey, Pennsylvania, 2001.

[140] Xiang Sean Zhou, Baback Moghaddam, and Thomas S. Huang. ICA-based probabilistic local appearance models. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, October 2001.