2014-04-27

Integration of Content and Context Modalities for Multimedia Big Data Retrieval

Qiusha Zhu
University of Miami, qiusha.zhu@gmail.com

UNIVERSITY OF MIAMI

INTEGRATION OF CONTENT AND CONTEXT MODALTIES
FOR MULTIMEDIA BIG DATA RETRIEVAL

By

QIUSHA ZHU

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

May 2014

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

INTEGRATION OF CONTENT AND CONTEXT MODALITIES
FOR MULTIMEDIA BIG DATA RETRIEVAL

Qiusha Zhu

Approved:

Mei-Ling Shyu, Ph.D.
Professor of Electrical and Computer
Engineering

Xiaodong Cai, Ph.D.
Associate Professor of Electrical
and Computer Engineering

Saman Aliari Zonouz, Ph.D.
Assistant Professor of Electrical and
Computer Engineering

Nigel John, Ph.D.
Lecturer of Electrical and
Computer Engineering

Shu-Ching Chen, Ph.D.
Professor of School of Computing and
Information Sciences
Florida International University

M. Brian Blake, Ph.D.
Dean of the Graduate School

ZHU, QIUSHA   (Ph.D., Electrical and

<u>Integration of Content and Context Modalties</u>   Computer Engineering)
<u>for Multimedia Big Data Retrieval</u>   (May 2014)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Mei-Ling Shyu.
No. of pages in text. (160)

With the proliferation of digital photo-capture devices and the development of web technologies, the era of big data has arrived, which poses challenges to process and retrieve vast amounts of data with heterogeneous and diverse dimensionality. In the field of multimedia information retrieval, traditional keyword-based approaches perform well on text data, but it can hardly adapt to image and video due to the fact that a large proportion of this data nowadays is unorganized. This means the textual descriptions of images or videos, also known as metadata, could be unavailable, incomplete or even incorrect. Therefore, Content-Based Multimedia Information Retrieval (CBMIR) has emerged, which retrieves relevant images or videos by analyzing their visual content. Various data mining techniques such as feature selection, classification, clustering and filtering, have been utilized in CBMIR to solve issues involving data imbalance, data quality and size, limited ground truth, user subjectivity, etc. However, as an intrinsic problem of CBMIR, the semantic gap between low-level visual features and high-level semantics is still difficult to conquer. Now, with the rapid popularization of social media repositories, which allows users to upload images and videos, and assign tags to describe them, it has brought new directions as well as new challenges to the area of multimedia information retrieval. As suggested by the name, multimedia is a combination of different

content forms that include text, audio, images, videos, etc. A series of research studies have been conducted to take advantage of one modality to compensate the other for various tasks.

A framework proposed in this dissertation focuses on integrating visual information and text information, which are referred to as the content and the context modalities respectively, for multimedia big data retrieval. The framework contains two components, namely MCA-based feature selection and sparse linear integration. First, a feature selection method based on Multiple Correspondence Analysis (MCA) is proposed to select features having high correlations with a given class since these features can provide more discriminative information when predicting class labels. This is especially useful for the context modality since the tags assigned to the images or videos by users are known to be very noisy. Selecting discriminative tags can not only remove noise but also reduce feature dimensions. Considering MCA is a technique used to analyze nominal features, a discretization method based on MCA is developed accordingly to handle numeric features. Then the sparse linear integration component takes the selected features from both modalities as the inputs and builds a model that learns a pairwise instance similarity matrix. An optimization problem is formulated to minimize the differences between the similarity matrix generated from the context modality and the differences between the similarity matrix generated from the content modality. Coordinate descent and soft-thresholding can be applied to solve the problem. Compared to the existing approaches, the proposed framework is able to handle noisy and high dimensional features in each of the modalities. Feature correlations are taken into account and no local decision or handcrafted structure is required. The methods presented in this framework

can be carried out in parallel, thus parallel and distributed programming framework, such as MapReduce, can be adopted to improve the computing capacity and scale to very large data sets. In the experiment, multiple public benchmark data sets, including collections of images and videos, are used to evaluate each of the components. Comparison with some existing popular approaches verifies the effectiveness of the proposed methods for the task of semantic concept retrieval.

Two applications using the proposed methods for content-based recommender systems are presented. The first one uses the sparse linear integration model to find similar items by considering the information from both images and their metadata. Experiment and subjective evaluation are conducted on a self-collected bag data set for online shopping recommendations. The second one employs a topic model to the features extracted from videos and their metadata to determine topics in an unified manner. This application recommends movies with similar distributions in textual topics and visual topics to the users. Benchmark MovieLens1M data set is used for evaluation. Several research directions are identified to improve the framework for various practical challenges.

To my family

Acknowledgments

First and foremost, I would like to express my sincerest appreciation to my advisor and chairman of the committee Dr. Mei-Ling Shyu for her support and guidance during my entire Ph.D. program. My appreciation also goes to Dr. Shu-Ching Chen of the School of Computing and Information Sciences at Florida International University (FIU) for his advice and encouragement through the years. Their professional and committed working attitude touches me and will continue to influence me. My deepest thanks go to Dr. Xiaodong Cai, Dr. Nigel John and Dr. Saman Aliari Zonouz of the Department of Electrical and Computer Engineering at the University of Miami for accepting the appointment to the dissertation committee and their consistent help and suggestions. My thanks also go to Prof. Lask for his long-term teaching guidance and Dr. Miroslav Kubat for his teaching advice as well.

Second, I would like to express my appreciation to the department for providing me financial support through my Ph.D. program. It offers me teaching assistantships to multiple courses, which helps me develop my teaching techniques and communication skills. I would also like to extend my appreciation to the research opportunities offered by TCL Research America and Senzari Inc, where I had the opportunity to apply my knowledge to solve real-world problems in areas of movie recommendations and music recommendations. In return my experience and skills in various aspects have also grown during the time working with Dr. Haohong Wang, Mr. Demian Bellumio and many other colleagues.

Third, I'm very grateful for the long-term help and genuine support from the members in the Data mining, Database and Multimedia Research Group at UM, Dr. Dianting Liu, Dr. Tao Meng, Dr. Lin Lin, Dr. Chao Chen, Dr. Zifang Huang and Mr. Yilin Yan, as well as the members from the Distributed Multimedia Information Systems Laboratory at FIU, Mr. Fausto Fleites, Ms. Yimin Yang and Ms. Hsin-Yu Ha.

Finally, my beloved family and friends have always believed in me and have never stopped offering their patient love, continuous encouragement and unconditional support. I wouldn't have been able to conquer so many difficulties in my life without them. To them, I owe my eternal gratitude.

<div align="right">QIUSHA ZHU</div>

University of Miami

May 2014

Contents

List of Figures

List of Tables

xiii

Chapter 1

Introduction

Living in a world where digital photo-capture devices has become ubiquitous, and more and more people share their lives on social networking websites, such as YouTube, Flickr, and Facebook. These media repositories allow users to upload images and videos, and edit their metadata, such as titles, descriptions and tags. This new trend has brought a shift in the research of multimedia information retrieval from traditional text-based retrieval to content-based retrieval, and now to a paradigm that needs to integrate both. It also imposes demands on the scalability of infrastructure, as well as algorithms to handle big data regarding their storage, processing and retrieval.

Traditional text-based approaches can be traced back to 1970s, which usually relied on manual annotation to perform retrieval. The construction of an index (or a thesaurus) was mostly carried out by specialists, who manually assigned a limited number of keywords describing the image and video content. Shortly, the processing speed failed to meet the requirements of fast and automatic searches of multimedia content since a manual analysis of multimedia data can be very expensive or simply not feasible when the time is limited or when the amount of data is enormous. In order to organize the vast amount of increasing online multimedia data, learning techniques focused on content

analysis have gained popularity over traditional text-based analysis [3][4]. Content-based approaches were introduced in the early 1990s to classify and retrieve images and videos on the basis of low-level and mid-level visual features. These features are attributes that describe an instance or item. based on color, texture and shape information [5][6]. Significant improvements were made in content-based retrieval in recent years in areas such as semantic concept detection [7], automatic image annotation [8] and motion detection [9][10].

Compared to information retrieval, recommender systems take one step further by actively recommending interesting items to users. Currently, the content in most recommender systems [11] is still limited to the metadata associated with these items. It represents items as feature vectors, and user interests or preferences are discovered by analyzing these textual features [12][13]. For video recommendation, Netflix and Hulu use movie genres, sub-genres, or a combination to describe and organize user interests. Jinni constructs movie genome by expert knowledge and online reviews to describe each movie. One obvious shortcoming of text-based recommendation is that using textual features to describe items requires accurate text information. However, most online videos are unorganized, which means their metadata could be incomplete, non-existent, or even incorrect. For these videos, either a lot of effort needs to be spent in manually annotating them or automatic tagging methods have to be applied [14][15]; otherwise, these systems would produce poor results. Meanwhile, a user's interests in a video are multifaceted. A user might be interested in the plot, the characters in the video, or the visual appearance of some scenes. In these cases, a deep analysis in the multimedia content is necessary in order to make recommendations based on what users are really interested in, which sometimes is hard to be captured by the metadata. Given

these inevitable problems of using metadata alone, visual content could provide extra information to help better capture user interests.

1.1 Motivations and Challenges

Although significant improvements have been achieved by using low-level visual features, the semantic gap challenge still remains [16]. It refers to the difference between high-level semantic concepts (e.g., sky, buildings, dogs, etc.) and extracted low-level visual features (e.g., color, shape, texture, etc.). It is produced by "the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation" [17]. Various advanced features have been detected in order to visually capture the middle-level to high-level semantics contained in an image or a video. The consequence of adding more features is the high-dimensional feature space, which could cause overfitting of a model due to the "curse of dimensionality".

In light of the advantages and disadvantages of both content-based and text-based approaches, studies in recent years have started to investigate how to utilize both approaches to enhance each other [18][19]. The fundamental property that differentiates these two approaches is the way in which the information is presented, also known as information modality. For content-based approaches, the information is presented by images or videos themselves, which is referred to as the content modality; while for text-based approaches, the information contained in images or videos is presented by texts in the form of metadata, such as titles, descriptions, tags, and surrounding texts of the images or videos. Thus it is referred to as the context modality. On one hand, visual features extracted from the content modality suffer from the semantic gap problem as mentioned before, but they make the automation of organizing multimedia content pos-

Figure 1.1: Social Web image with noisy tags

sible and greatly save human efforts of manual annotation. On the other hand, textual features extracted from the context modality can usually express the semantics contained in an image or a video and bridge the semantic gap that exists in the content modality. However, this metadata is contributed by users, which is known to be imprecise, subjective and uncontrolled. It is too noisy to be used directly as keywords to describe the content. Figure 1.1 shows some sample images from Flickr together with the user assigned tags. As can be seen, useful tags (tags describing the image content correctly) are embedded in noisy ones.

Motivated by the complementary information contained in the content and context modalities, research has been conducted to investigate how to use one modality to help

the other one. For example, tags associated with images are utilized to boost the performance of content-based retrieval [20][21]. In return, visual content is also used to refine noisy tags [22][23][24]. A more general research topic is how to integrate information from different modalities, which is a process also known as multimodal fusion in a broad sense. For multimedia information retrieval, the two core issues of information fusion are (i) levels of fusion and (ii) methods of fusion. Fusion can be performed at the feature level (known as early fusion), or at the decision/modal level (known as late fusion), or in between. In each level, various fusion methods are developed which aim to take advantage of each individual modality, as well as the underlining correlation among them. There are many practical problems in multimodal fusion. For example, the granularity of different modalities may be inconsistent. In video semantic concept detection [25][26], the visual features are extracted from shots in a video, while the textual features extracted from metadata describe a video as a whole without the detailed information about each shot. Thus the feature representation of these two modalities are at different granularities, which needs to be taken into consideration when designing the fusion methods.

Besides the challenges that arise in multimodal integration, processing multimedia data itself is a non-trivial task due to the large volume. A 4 minute video with a frame rate of 30p contains more than 7000 frames. If extracting visual features from images, the dimensions of the feature space are usually between 100 to 1000. Imagine how many instances a collection of videos would contain and how many features they would generate for later process. With the advance of digital technology and social web media, the amount of multimedia data would keep increasing. This data is useless without efficient and scalable algorithms to extract knowledge from them. Arising in the con-

text of big data, algorithms that support parallel and distributed computing are in high demand. The open source project Apache Solr [27] is a highly reliable and scalable text search platform that provides distributed indexing and load-balanced querying. Apache Mahout [28] is a scalable machine learning library that provides some popular recommendations, classifications, and clustering algorithms, such as collaborative filtering, K-Means, and Naive Bayes. However, few applications support a large volume of multimedia data directly, as well as other challenges brought by big data. This dissertation presents a scalable framework for multimedia big data retrieval. The term multimedia big data is used to emphasize the challenges of multimedia data in the big data era.

1.2   Proposed Solutions

In this dissertation, the framework proposed for multimedia big data retrieval integrates the semantic information embedded in the metadata and the visual information contained in images or videos. As shown in Figure 1.2, features are extracted from content and context modalities to generate one feature representation for each modality. A matrix with instances as its rows and features as its columns expresses information contained in a feature representation. Sometimes the transpose of a matrix is used whose rows are features and columns are instances, depending on which way is more convenient for expression. Two components are contained in this framework. The MCA-based feature selection component selects discriminative features from the original feature representation. This is especially useful for the context modality, since the metadata is usually very noisy and the text features can easily reach to a very high dimension. The generated feature representations from the feature selection component are the input to the sparse linear integration component, which integrates the feature representations from different modalities and learns pairwise instance similarities.
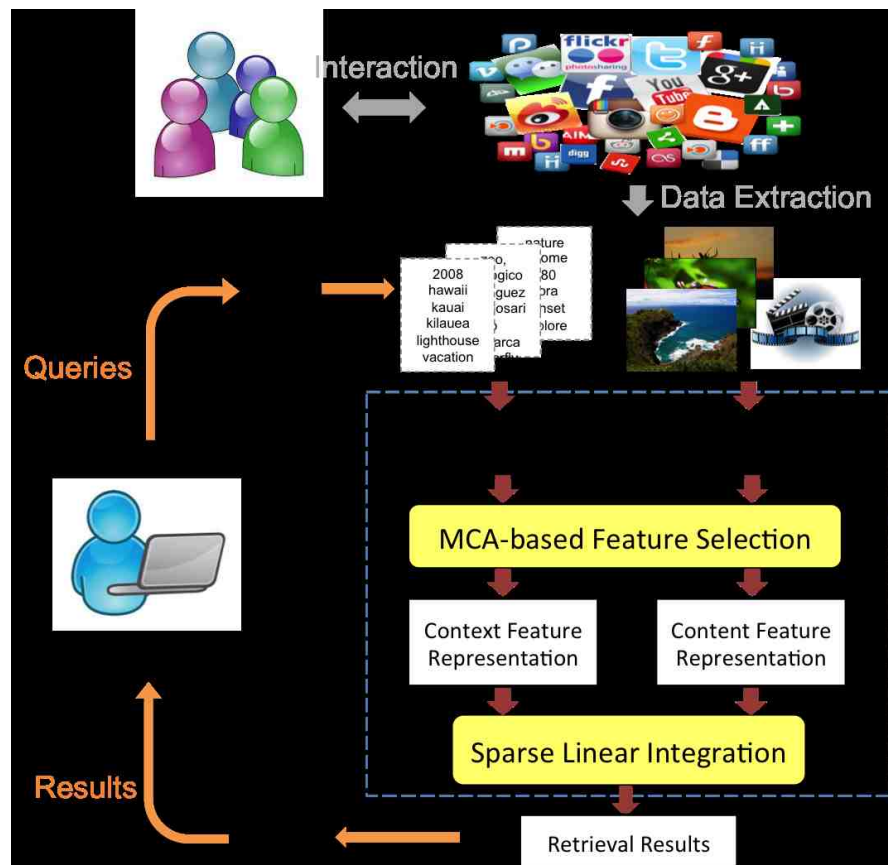
Figure 1.2: Proposed solutions

1.2.1 MCA-based Feature Selection Component

For the context modality, as mentioned before, the metadata is usually noisy, which requires proper methods to remove the noisy words that are not related to the actual content of the image or video. On the content side, various low-level and mid-level visual features are extracted to try to capture the high-level semantic meaning of images and videos, which results in a very high feature dimension and could cause the "curse of dimensionality" issue. Therefore, an effective feature selection method is needed on both modalities to select discriminative features, which could not only improve model performance in the later stage but also reduce the computation complexity. Multiple Correspondence Analysis (MCA) captures the correlation between each feature and the class labels. The metrics developed select features that have a high correlation with a given class label. This approach does not alter the original feature space, and thus the meanings of the features are kept, which can offer the interpretability of the model. This is a very important property compared to dimension reduction approaches such as Principle Component Analysis (PCA). Considering MCA is originally applied to nominal features, a discretization method is developed accordingly to facilitate MCA to handle numeric features so that the MCA-based feature selection method can be applied to both nominal and numeric features. As an application of the MCA technique, the correlation results generated from the MCA-based feature selection can be reused for classification.

1.2.2 Sparse Linear Integration Component

To integrate multiple modalities, early fusion approaches concatenate feature representations of different modalities and generate a combined feature representation. Thus

the integration is performed at the feature-level. To provide more scalability, instead of combining feature representations, late fusion approaches generate a local decision for each modality first, and then treat each local decision as a feature to build a super model for the final decision. However, they discard the underlining correlation of the feature representations from different modalities and treat each modality as an independent signal. The proposed sparse linear integration model is an intermediate fusion approach, which does not generate a combined feature representation, nor does it require local decisions first. An optimization problem is formulated to learn a pairwise instance similarity matrix which considers the information from multiple modalities. The pairwise instance similarities can be directly used for unsupervised applications. A classifier based on the reconstruction error can also be embedded in the model for supervised applications. Associations between modalities are considered to generalize the model to be capable of handling granularity differences of modalities. The granularity difference refers to the instances from different modalities that are not in the same unit. For example, the instance in one modality may be a video and in the other modality is a video frame in the video; or in one modality it is the text in a web page containing several images while in the other modality is an image in a web page.

1.3 Contributions

Several contributions are made in this dissertation on the topic of integrating content and context modalities for multimedia big data retrieval. They are summarized as follows:

1. A feature selection method utilizes Multiple Correspondence Analysis (MCA) to capture the correlation between each feature and the class. A ranking list is generated to order the features according to their correlations with a given class.

A feature having a higher rank indicates a stronger correlation with the class and thus this feature is expected to provide more information about predicting the class label.

2. Due to the fact that MCA can only be applied to nominal attributes, a discretization method is developed that finds a partition scheme on a numeric feature, which ensures a local maximum correlation calculated by MCA between this discretized feature and a given class. A local maximum is preferred over the global maximum in order to seek a balance between classification accuracy and discretization efficiency.

3. A framework of the sparse linear integration component integrates multiple modalities at the intermediate level. An optimization problem is formulated to learn a pairwise instance coefficient matrix from these feature representations. Coordinate descent and soft-thresholding are applied to solve the problem. The learned model can be directly used for unsupervised applications, which usually involve finding the similarity between two instances. Classification can also be embedded in the integration process to extend the sparse linear integration model for supervised learning. The model is also generalized to handle the situation when the instances from different modalities are not in the same granularity.

4. There are two applications of integrating content and context modalities for content-based recommendation. Visual information is introduced to describe the item content in addition to metadata as used by most approaches. The sparse linear integration method is directly applied to find similar items for recommendation purposes. For video recommendation, a topic model represents both videos and

video metadata as distributions over topics, which are viewed as features. Then either early fusion or late fusion can be applied to generate the recommendation results.

1.4   Scope and Limitations

The framework has the following assumptions and limitations.

1. A fundamental assumption is that the metadata which constructs the context modality contains the semantic information reflected by the multimedia content they are associated with. Based on this assumption, the high-level semantic information from the context modality can be utilized to help bridge the semantic gap. If there is too much noise in the metadata, which causes the semantic information in the context modality to be irrelevant to that in the content modality, then the integration of these two modalities would make no sense.

2. The sparse linear integration component requires the feature representation from each modality to be in the same scale. That is, their feature dimensions should be similar; otherwise, one feature representation would overshadow the other and cause the learned model to lean toward the one with a higher dimension, which means the one with higher dimensions would contribute more to the pairwise instance coefficients. Therefore, certain techniques such as feature selection need to be applied to reduce feature dimensions in order to make the different feature representations be in the same scale. In addition, currently a full pairwise instance coefficient matrix is learned. If the number of training instances is very large then the computation complexity of this full matrix would be high even though the sparsity constraint is imposed. Considering those instances with small coeffi-

cients would contribute less to the pairwise similarity matrix, these instances can be ignored. Hence, only instances with big coefficients are used in each iterations to learn this similarity matrix.

3. Offline training mode is assumed in the framework, which means the training instances are known beforehand. Therefore, all the training instances are used to train a model at once instead of training incrementally. This limitation is due to the fact that in the sparse linear integration component, all the original feature representations of the training instances are used to learn the pairwise instance similarity matrix. For the MCA-based feature selection and discretization, MCA is also performed on all the training instances.

4. Some parameters in the framework are based on an iterative search on the training data to find the optimum values. This empirical approach may be inevitable, but in some cases an advanced parameter estimation approach, such as maximum likelihood and maximum a posteriori, can be investigated.

1.5   Evaluation Metrics

Mean average precision (MAP) is one of the most widely used metric in information retrieval. It is the mean of the average precision scores for each query, while average precision (AP) is computed as a function of recall, as shown in Equation (1.1). Q is the total number of queries, and TP@n is the number of true positive at cut-off n. P@i is the precision at cut-off i in the ranking list and $\Delta(i)$ is an indicator function equaling 1 if the item at rank i is a relevant one, zero otherwise. n can be set to such as 5, 10 and 100 depending on the circumstances. If all the retrieval results are considered, then AP@all and MAP@all can be used.

$$MAP@n \quad = \frac{\sum_{q=1}^{q=Q} AP@n}{Q}$$

(1.1)

$$where \quad AP@n \quad = \frac{\sum_{i=1}^{i=n} P@i \times \Delta(i)}{TP@n}$$

Besides AP@n and MAP, precision, recall, and F1-score (F1) which is the harmonic mean of precision and recall, are popular evaluation metrics for classification. Their measurement are calculated according to Equation (1.2).

$$\begin{aligned} precision \quad &= \frac{TP}{TP+FP} \\ recall \quad &= \frac{TP}{TP+FN} \\ F1 \quad &= 2 \times \frac{precision \times recall}{precision + recall} \end{aligned}$$

(1.2)

Another widely adopted metric of information retrieval is the area under the ROC curve (AUC), which is a more general ranking measure. It is equal to the probability that a model will rank a randomly chosen positive item higher than a randomly chosen negative one. The best possible value of AUC is 1, and any non-random ranking that makes sense would have an AUC value greater than 0.5.

1.6 Notation

Some frequently used notations are defined as follows. The meaning of these variables are consistent through out this dissertation. There are also some other variables defined in each chapter.

F: a feature/attribute

K_F: the number of intervals/values of a nominal feature F

C: the class

K_C: number of classes in C

N: the number of instances

M: the number of features

All the vectors are denoted using bold lower-case letters, and matrices are denoted using bold upper-case letters. The dimension of a vector or matrix is denoted using upper-case letters, and the lower-case letters are used to represent an entry in a vector or matrix as well as its index. A superscript T denotes the transpose of a matrix or vector.

1.7 Outline of the Dissertation

This dissertation is organized as follows. Chapter 2 reviews the techniques related to feature representation, feature selection, discretization, information fusion and recommendation. The advantages and limitations of peer work are analyzed and discussed. Chapter 3 presents the MCA-based feature selection component and the related discretization method which extends the proposed feature selection method from nominal data to numeric data. Chapter 4 presents the sparse linear integration component that integrates the feature representation from different modalities. Two applications of content-based recommendation are introduced in Chapter 5, which improve the recommendation accuracy by incorporating visual information into textual information using the proposed methods. Finally, conclusions are drawn in Chapter 6, which also identifies several future directions that can be explored to improvement the framework.

Chapter 2

Literature Review

Based on the framework of integrating content and context modalities for multimedia information retrieval, this chapter provides a thorough review of related work and techniques in the area of feature representation for both content and context modalities, feature selection, discretization, and information fusion. As an important application of the framework, recommendation can also benefit from integrating multiple information sources for accurate recommendation. General research directions of recommendation are introduced in this chapter as well as the popular algorithms used in recommender systems.

2.1 Feature Representation

To represent an instance of various modalities, such as images, videos, audio or documents, typically features are extracted from each modality and an instance is represented as a vector in the feature space. This dissertation is focused on visual representations of content modality and textual representations of context modality. The term instance and item are used exchangeably to denote an image or a video, depending on the context. It is assumed that the content modality and the context modality are avail-

able, which refer to the actual images or videos and the associated textual descriptions, also known as metadata, respectively.

For visual representation, features that characterize the information of color, edge, texture, shape, etc. about images or videos are usually detected. These features can be categorized into global features and local features [29]. Color histogram, color momentum, edge detection histogram, wavelet texture, and CEDD [30] are typically used as global features. Each image corresponds to a feature vector. The global representation is compact, but is also sensitive to occlusion and clutter, especially when the focus is on the objects in the images. In these cases, segmentation has to be applied that gets interested objects before extracting global features. In contrast, local features capture localized image regions, also known as patches, by computing descriptors around interest points. Thus each image is represented by a set of descriptors of different sizes and can not be directly used in standard classification task [31]. Techniques such as clustering need to extract features from these descriptors so that each image can be represented by a feature vector as well. The advantage of local features is that it doesn't need to do segmentation and the features are robust to occlusion and clutter. Scale-Invariant Feature Transform (SIFT) [32] [33] and its variants SURF [34], CSIFT [35] belong to this category. Features such as Local Binary Patterns (LBP) [36] and Histograms of Oriented Gradients (HOG) [37] can also be used as local features. For general data analysis, several features are extracted and are concatenated into a feature vector of higher dimensions. Since the ranges of different features are varied, normalization is commonly applied to ensure they are in the same range.

The "Bag of Words" (BOW) model is a very popular model used in text mining for information retrieval (IR). It considers descriptive data such as titles and abstracts

as collections of words regardless of grammar and word order. Therefore, the tags associated to an image or a video can be considered as a "bag of words" and fit into the BOW model. An image or a video is represented by a feature vector with each of its distinct tag or term as a feature. Descriptions of images or videos can be treated in a similar manner by considering each word as a term using unigram. Of course, typical preprocessing of text such as stop words removal and stemming are usually needed before forming the feature vectors. However, given the huge number of terms, only those representative and discriminative terms should be kept as features. Tf-idf (term frequency-inverse document frequency) [38] is widely used in IR and text mining to weight terms, and many variants have been developed based on it. In conventional tf-idf, the importance of a term increases proportionally to the number of times it appears in a document but is offset by the frequency of the term in the whole collection. For each term, a tf-idf value can be calculated and terms with large tf-idf values are kept as features while less significant ones are excluded. Considering the same tag is usually assigned to the same image or a video only once, the tf value of the tag for this instance is 1, and tf-idf essentially becomes idf in this case. Wang et al. [39] argue that idf is less reasonable to be used in text categorization than it is in IR task. By taking the category/class information into account, it introduces Inverse Category Frequency (ICF) as a supervised term weighting scheme. The weight of a term can directly be used as the feature value of the instance that contains this term, and for instances that don't contain this term, the corresponding values is 0. The binary representation is also commonly used for textual features, where 1 indicates the presence of a term and 0 indicates absence. Textual representations are usually sparse since the feature space of terms is often large and the metadata of an instance is short.

2.2 Feature Selection

With the increasing number of high-dimensional data sets ranging from several hundred to hundred thousand features, the process of selecting a good feature subset has become more and more important. Such a process can remove irrelevant, redundant, or noisy features to improve model performance and make models more cost-effective. Depending on how it is combined with the construction of the classification model, supervised feature selection can be further divided into three categories: wrapper methods, embedded methods, and filter methods. Wrappers choose feature subsets with high prediction performance estimated by a specified learning algorithm which acts as a black box, and thus wrappers are often criticized for their massive amounts of unnecessary computation. Similar to wrappers, embedded methods incorporate feature selection into the process of training for a given learning algorithm, and thus they have the advantage of interacting with the classification model while being less computationally intensive than wrappers. These two categories usually yield better classification results than the filter methods, since they are tailored to a specific classifier, but the improvements of the performance are not always significant because of the "curse of dimensionality" and the fact that the specific tuned classifiers may overfit the data. In contrast, filter methods are independent of the classifiers and can be scaled for high-dimensional data sets while remaining computationally efficient. In addition, filtering can be used as a pre-processing step to reduce space dimensionality and overcome the overfitting problem. Therefore, filter methods only need to be executed once, and then different classifiers can be evaluated based on the generated feature subsets.

Filter methods can be further divided into two main sub-categories. The first one is univariate methods, which consider each feature with the class separately and ignore the

interdependence between the features. Representative methods in this category include information gain and chi-square measure, both of which are widely used to measure the dependence of two random variables. Information gain evaluates the importance of features by calculating their information gain with the class, but this method is biased to features with more values. A new feature selection method was proposed which selected features according to a combined criterion of information gain and novelty of information [40]. This criterion strives to reduce the redundancy between features while maintaining information gain in selecting appropriate features. In contrast, chi-square measure calculates the χ^2 statistics between each feature and the class, and a large value indicates a strong correlation between them. Although this method does not adhere strictly to the statistics theory because the probability of errors increases when a statistical test is used multiple times, it is applicable as long as it only ranks features with respect to their usefulness [41]. Jiang et al. [42] used the bag-of-visual-words (BoW) features to represent keypoints in images for semantic concept detection. As one of the representation choices of BoW, feature selection applied the chi-square measure to calculate the χ^2 statistics between a specific visual word and a binary label of an image class, and eliminated those virtual words with χ^2 statistics below a threshold. Extensive experiments on the TRECVID data indicated that BoW features with appropriate representation choices could produce highly competitive results.

The second sub-category is the multivariate methods, which take features' interdependence into account. However, they are slower and less-scalable compared to the univariate methods. Correlation-based feature selection (CFS) is one of the most popular methods. It searches among the features according to the degree of redundancy between them in order to find a subset of features that are highly correlated with the

class, yet uncorrelated with each other [43]. Experiments on natural data sets showed that CFS typically eliminated over half of the features, and the classification accuracy using the reduced feature set was usually equal to or better than the accuracy using the complete feature set. The disadvantage is that CFS degrades the performance of classifiers in cases where some eliminated features are highly predictive of very small areas of the instance space. These kind of cases could be frequently encountered when dealing with imbalanced data. Relief is another commonly used method, which chooses the features that can be most distinguishable between classes. It evaluates the worth of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different classes. However, relief lacks a mechanism to deal with the outlier instances, and it also has worse performance than the univariate filter methods in most cases [44]. Sun [45] proposed an iterative relief (I-Relief) method by exploring the framework of the Expectation-Maximization algorithm. Large-scale experiments conducted on nine UCI data sets and six microarray data sets demonstrated that I-Relief performed better than relief without introducing a large increase in computational complexity.

According to the form of the outputs, the four aforementioned feature selection methods can also be categorized into ranker and non-ranker methods. A non-ranker method provides a subset of features automatically without giving an order of the selected features such as CFS. On the other hand, a ranker method provides a ranked list by scoring the features based on a certain metric, to which information gain, chi-square measure, and relief belong. Then different stopping criteria can be applied in order to get a subset from it. Most commonly used criteria include forward selection, backward elimination, bi-directional search, setting a threshold, and genetic search.

As an important application, feature selection can be used to identify representative and discriminative terms from metadata. A supervised term weighting scheme, Inverse Category Frequency (ICF), proposed by Wang et al. [39], can also be considered as a feature selection method for terms. The assumption of ICF is that the terms appearing in fewer categories/concepts have a larger discriminative ability. Therefore, for the i-th term, an ICF score is calculated according to equation (2.1).

$$ICF_i = \log_2 \frac{K_C}{CF_i}, \tag{2.1}$$

where K_C is the total number of concepts, and CF_i is the number of concepts that the i-th term occurs. For example, if the term occurs in 10 concepts, then CF_i equals to 10.

Besides the aforementioned methods, there are several popular dimension reduction methods, which are not feature selection techniques in a strict sense. Principal Component Analysis (PCA) is a widely used method, which chooses enough eigenvectors to account for some percentage of the variance in the original data—default 0.95 (95%) and transforms the attributes from the original space to the principal component space. Dimensions are reduced by eliminating some of the worst eigenvectors. As the roles of the users become more and more important, another dimension, users, is added into the original 2-dimension problem, which only involves images or items and tags. Symeonidis et al. [46] proposed a unified framework to represent the three types of entities in a social tagging system, which are users, items or images, and tags as a 3-order tensor. The latent semantic analysis and dimensionality reduction were performed using the Higher Order Singular Value Decomposition (HOSVD) technique to reveal the semantic association between users, items and tags. If only two dimensions are considered, HOSVD is essentially simplified to Singular Value Decomposition (SVD). Li et al. [47] presented a more general algorithm based on HOSVD for indexing and retrieval

of higher order tensor data obtained from a multi-camera system. Experiments showed that their approach could be used to handle a query structure consisting of an arbitrary number of objects, cameras, and modalities. A Pairwise Interaction Tag Factorization (PITF) method models the pairwise interaction between users, items and tags [48] . PITF has a linear runtime for both learning and prediction, thus making it more feasible for midsized and large data sets.

2.3   Discretization

A survey of discretization [49] categorized the discretization methods into χ-square based, entropy based, wrapper based, etc., but didn't cover the category that used the CAIR criterion. This probably was because this CAIR criterion was not popular back then. In this review, we focus on traditional entropy-based discretization methods, the recent proposed CAIR-based methods and also wrapped-based ones.

When developing a discretization algorithm, the following two main questions need to be answered:

1. What's the criterion to select a cut-point?

2. What's the criterion to stop cutting?

Given the notation defined in Section 1.7, the quanta matrix of a feature F is shown in Table 2.1, where Ω is a set of N instances and N_{ij} denotes the number of instances in feature interval j and belong to class C_i. Seven supervised discretization methods are discussed in this section.

Regarding to the first question, information entropy is adopted for many discretization algorithms, such as maximum entropy, which discretizes the numeric attributes using the criterion of minimum information loss. IEM [50] is a widely used one due

Table 2.1: quanta matrix of feature F with K_C classes and K_F intervals

Class	Interval $\{[d_0,d_1],(d_1,d_2),\cdots,(d_{K_F-1},d_{K_F}]\}$	Sum of class
C_1	$N_{11}\cdots N_{1j}\cdots N_{1K_F}$	N_{1+}
$\vdots$	$\vdots\cdots\vdots\cdots\vdots$	$\vdots$
C_i	$N_{i1}\cdots N_{ij}\cdots N_{iK_F}$	N_{i+}
$\vdots$	$\vdots\cdots\vdots\cdots\vdots$	$\vdots$
C_{K_C}	$N_{K_C1}\cdots N_{K_Cj}\cdots N_{K_CK_F}$	N_{K_C+}
Sum of intervals	$N_{+1}\cdots N_{+j}\cdots N_{+K_F}$	N

to its efficiency and good performance in the classification stage. IEM selects the first cut-point that minimizes the entropy function over all possible candidate cut-points and recursively applies this strategy to both induced intervals. For a numeric feature F, t is a candidate cut-point that splits Ω into subsets Ω_1 and Ω_2. $Ent(\Omega)$ defined in Equation (2.2) is the class entropy of Ω, where $p_{i+} = \frac{N_{i+}}{N}$ is the proportion of data instances in Ω and belong to C_i. $Ent(\Omega;t)$ is the class entropy induced by cut-point t to Ω, as shown in Equation (2.3), and N_{+1} and N_{+2} are the number of instances in Ω_1 and Ω_2 respectively. The difference of $Ent(\Omega)$ and $Ent(\Omega;t)$ given by Equation (2.4) is the information gain of partitioning Ω by t.

$$Ent(\Omega) = -\sum_{i=1}^{K} P_{i+}log(P_{i+}); \tag{2.2}$$

$$Ent(\Omega;t) = \frac{|N_{+1}|}{|N|}Ent(\Omega_1) + \frac{|N_{+2}|}{|N|}Ent(\Omega_2); \tag{2.3}$$

$$Gain(\Omega;t) = Ent(\Omega;t) - Ent(\Omega). \tag{2.4}$$

For IEM, the Minimum Description Length (MDL) principle is employed to determine whether to accept a selected candidate cut-point or not, in other words, when to stop cutting. Thus, the recursion can stop if the cut-point does not satisfy a certain

pre-defined condition, which is given in Equation (2.5). K_{C_1} and K_{C_2} are the numbers of classes in Ω_1 and Ω_2, respectively.

$$Gain(\Omega;t) > \frac{\log_2(N-1)}{N} + \frac{\Delta(t)}{N}, where \tag{2.5}$$

$$\Delta(t) = \log_2(3^{K_C} - 2) - [K_C Ent(\Omega) - K_{C_1} Ent(\Omega_1) - K_{C_2} Ent(\Omega_2)].$$

Compared to IEM, the discretization algorithm used the same strategy to select the best cut point but a different criterion modified based on MDL to decide when to stop the recursion [51]. Thus, it is considered as a variant of IEM (called IEMV). IEMV considers discretization as a transmission problem by minimizing the length (the number of bits) of the message. A selected cut-point is accepted if the number of bits needed to encode the total number of instances and the probability distribution of the classes before partition (Prior MDL') are larger than the bits needed after partition (Post MDL'). For a feature F, the prior MDL' ($Prior\_MDL'$) can be approximated with $Ent(\Omega)$ times N plus the number of bits needed to encode the decoder, and the post MDL' ($Post\_MDL'$) is the bits needed to encode the classes of data instances in all M intervals, which can be calculated by Equation (2.6). Therefore, a cut-point is accepted if $Post\_MDL'$ is smaller than $Prior\_MDL'$, which means the feature is compressed during the partitioning. Empirical studies have shown that this criterion is always negative for irrelevant features, which means irrelevant features are non-compressive. For informative features, the compression capability increases with the increase of the number of intervals of features.

However, for IEM and IEMV, if the features are noisy or don't provide much information to predict the classes, then numeric values of these features will be discretized

into the same interval. It means that none of candidate cut-point satisfies the threshold (Equation (2.5 for IEM and Equation (2.6 for IEMV), and these features discretized by IEM and IEMV are useless in classification, even though the original numeric features may still have some relationship with the classes.

$$
\begin{aligned}
Prior\_MDL' &= N \times Ent(\Omega) + \log_2 \binom{N + K_C - 1}{K_C - 1}; \\
Post\_MDL' &= N \times Ent(\Omega; T) + \sum_j^M \log_2 \binom{N._j + K_C - 1}{K_C - 1} + \log_2 M.
\end{aligned} \tag{2.6}
$$

In addition to entropy maximization, another widely used discretization criterion is Class-Attribute Interdependence Redundancy (CAIR), which measures the interdependence between classes and each discretized feature, though it may be overfitting. Based to the quanta matrix shown in Table 2.1, *CAIR* value is calculated according to Equation (2.7).

$$
CAIR = \sum_{i=1}^{K_C} \sum_{j=1}^{M} p_{ij} \log_2 \frac{p_{ij}}{p_{i+}p_{+j}} \Big/ \sum_{i=1}^{K_C} \sum_{j=1}^{M} p_{ij} \log_2 \frac{1}{p_{ij}}, where \tag{2.7}
$$

$$
p_{ij} = \frac{N_{ij}}{N}, p_{i+} = \frac{N_{i+}}{N}, and\, p_{+j} = \frac{N_{+j}}{N}.
$$

CAIM [52] is a representative algorithm that maximizes *CAIR* value and generates possibly the smallest number of intervals for a given numeric feature. In Equation (2.8), max_j is the maximum value among all values of N_{ij} that fall into the j-th interval. The larger the value of CAIM, the higher the interdependence between class labels and discrete intervals. Instead of using the recursive strategy, CAIM selects the first cut-point from all candidates and then selects the next one from the rest of the candidate cut-

points. It keeps the one with the highest CAIM value, and stops until the CAIM value of the next selected cut-point is smaller than the current highest one. Experiments showed that compared to other discretization algorithms including IEM, CAIM generated a better discretization scheme on average as a pre-processing step for classification.

$$CAIM = \frac{\sum_{j=1}^{M} \frac{max_j^2}{N_{+j}}}{N}. \qquad (2.8)$$

However, as pointed out in [53], CAIM gives a high factor to the number of generated intervals, which is usually very close to the number of classes. Also, CAIM only considers the majority class (determined by max_j) and ignores the rest. A discretization algorithm that followed the same strategy to select cut-points but uses "contingency coefficient" to measure the strength of dependence between the variables was proposed by [53]. It calculated a CACC value for each candidate cut-point according to Equation (2.9).

$$CACC = \sqrt{\frac{y}{y+N}}, where \qquad (2.9)$$

$$y = \frac{N}{\log(K_F)} \left[\left(\sum_{i=1}^{K_C} \sum_{j=1}^{K_F} \frac{N_{ij}^2}{N_{i+}N_{+j}} \right) - 1 \right].$$

$\log(K_F)$ was used to reduce the influence of the number of intervals. Experiments on both real and artificial data sets indicated that CACC can generate a higher *CAIR* value compared to CAIM and improve classification accuracy like decision trees.

To handle data uncertainty, a method [54] added an offset to CAIM and defined it as UCAIM shown in Equation (2.10). The purpose of adding that offset was to make the CAIM value more sensitive to the change of values in the quanta matrix. A larger offset means that within interval j, the probability of an instance belongs to the majority

class is higher than the other classes, so the interdependence between interval j and the majority class is also higher.

$$UCAIM = \frac{\sum_{j=1}^{K_F} \frac{max_j^2 \times offset_j}{N_{+j}}}{N}, where \tag{2.10}$$

$$offset_j = \frac{\sum_{i=1}^{K_C} (max_j - N_{ij})}{K_C - 1}.$$

Similar to the wrapped-based approaches in feature selection that take the feedback from an induction algorithm, the discretization methods can also embed a classifier in the process and use the measurement of the classifier to select the cut-points or as the stopping criteria. An error function is used to evaluate the candidate cut-points, which aims to find the best discretization scheme that minimizes the total number of errors (i.e. false positive and false negative instances) [55]. Compared to the error-based approaches, the cost-based approaches [56] took into account the cost of making errors instead of just minimizing the total sum of errors. The specification of the cost function is dependent on the costs assigned to the different error types, and thus it can handle the imbalanced data better than error-based ones.

In addition to the error or cost, accuracy is also used as a classification measurement. Adaptive Quantizer [57] splits an interval into two partitions either by an equal width or equal frequency. It continues splitting in this binary recursive manner until the splitting cannot further improve the accuracy. Depending on whether an equal width (EW) or equal frequency (EF) is used, Adaptive Quantizer derives two discretization methods, denoted as AQEW and AQEF (both are accuracy-based approaches). They attempt to overcome the drawbacks of EW and EF, namely (i) generating the unbalanced intervals, and (ii) wrapping a classification algorithm in their discretization process when deciding

whether to continue splitting or not. Although EW and EF are unsupervised, AQEW and AQEF are considered as supervised discretization approaches because they take into account the class label information.

2.4   Information Fusion

In the field of multimedia retrieval [31] [58], information from multiple modalities have been utilized to complete each other and have shown promising results in tasks such as semantic concept detection, speech recognition, and multi-sensor fusion [59][60][61]. Current methods in information fusion typically fall into one of the four branches:

1. Early fusion typically concatenates features from different modalities and results in a single feature representation to be used as input to a learner. This approach is simple and generic but is subject to the "curse of dimensionality" since the concatenated features can easily reach to very high dimensions.

2. Late fusion applies a separate learner to each modality and fuses their decisions through a combiner. Compared to early fusion, late fusion offers scalability and freedom to choose suitable learning methods for each modality. However, it cannot utilize the feature-level correlations from different modalities and is required to make local decisions first.

3. Hybrid fusion involves both early fusion and late fusion by applying early fusion on some modalities and late fusion on the rest of the modalities. Then these decisions are combined in a late fusion manner. Although it offers the flexibility of choosing the proper fusion approach on a subset of modalities, its structure is often application dependent, which requires domain knowledge.

4. Intermediate fusion is an emerging branch, which does not alter the input feature representation nor require local decisions. It integrates multiple modalities by inferencing a joint model for decision, thus this approach often has superior prediction accuracy [62].

A comparison between early fusion and late fusion was done by Snoek et al. [63], and experiments on broadcast videos for video semantic concept detection showed that late fusion tends to slightly outperform early fusion for most concepts, but for those concepts where early fusion performed better, the gain was more significant.

Many studies have attempted to integrate content and context modality for image retrieval. Nagel et al. [64] presents the participation of the Fraunhofer IDMT in the ImageCLEF 2011 Photo Annotation Task. The text-based features were extracted by computing tf-idf values of each tag and visual features were RGB-SIFT descriptors using the codebook approach. In early fusion manner, both visual and text-based features were considered simultaneously to train the SVM classifier, while in late fusion, two SVMs were trained for each modality and then the results were combined using the geometric mean. The Mean Average Precision (MAP) of 99 concepts showed that the late fusion approach outperformed the early fusion by a very small margin, about 1.5%. An advanced framework proposed in Caicedo et al. [2] connects two data modalities using matrix factorization to project these two matrices into a latent space. Therefore, each representation can be backprojected to the space of the other representation through the common latent space. Then the two backprojected representations are concatenated as well with a weight parameter. Experiments on Corel 5K and MIRFLICKR data sets showed the effectiveness of this framework by comparing with Joint Factorization [65] and their previous work using Non-negative Matrix Factorization (NMF) [66].

Lienhart et al. [67] used multi-layer probabilistic Latent Semantic Analysis (pLSA) [68] and proposed a model with two leaf-pLSAs from two data modalities; one is image tags and the other one is image visual features. Then, a single top-level pLSA is merged from the two leaf-pLSAs and the result on Flickr images outperformed the unimodel (use visual features or tag features only) by approximately 19%. However, Clinchant et al. [69] argued that state-of-the-art models were insufficient to handle the asymmetric complementarities that existed between texts and images. They proposed a semantic combination strategy, which introduces a semantic filter using the textual scores to filter the visual scores and then combines the filtered visual scores with the textual score in a late fusion manner. Experimental results showed this late semantic fusion was more effective in terms of MAP than the direct late fusion.

Recently Multiple Kernel Learning (MKL) [70] has been introduced to the domain of heterogeneous feature fusion. It is regarded as intermediate fusion as compared to early fusion and late fusion since kernels are combined as a way to integrate multiple representations. Yu et al. [71] applied MKL to biomedical data fusion. ℓ_2-norm was adopted to get non-sparse optimal kernel coefficients, which was believed to have more advantages over the sparse solution resulted from ℓ_1-norm in real biomedical applications. Yeh et al. [72] proposed a novel multiple kernel learning (MKL) algorithm with a group lasso regularizer for heterogeneous feature fusion and variable selection It offers a robust way of fitting data extracted from different feature domains by assigning a group of base kernels for each feature representation in an MKL framework. A mixed ℓ_1-norm and ℓ_2-norm constraint enforces the sparsity at the group/feature level and automatically learns a compact feature representation for recognition purposes. Zitnik et al. [62] compared matrix factorization with the state-of-the-art MKL in handling

heterogeneous data fusion. A penalized matrix tri-factorization revealed data hidden associations, which simultaneously factorized data matrices. Good accuracy and time response were reported about this new data fusion algorithm.

2.5 Recommendation

Research on recommendation [73] is generally proceeded along three dimensions: content-based recommendation, which focuses on analyzing the content of items; collaborative filtering, which utilizes user profiles, such as ratings or clicks, to recommend items for like minded users; and hybrid recommendation, which incorporates both approaches. Due to the superior performance in Netflix competition, many state-of-the-art recommendation models adopt the latent factor model (LFM) [74][75][76]. These approaches belong to the collaborative filtering category, which involves analyzing user profiles, typically in the form of the user-item matrix. However, in many situations, user profiles are not available or very sparse, especially for online videos as a large proportion of users browse videos anonymously. As a result, dealing with the cold-start problem is inevitable. The cold-start problem describes the scenarios in recommender systems when user profiles are not available, which commonly arises at the beginning of the recommender systems. Thus, for new items (i.e., items without any user behavior data), collaborative filtering based methods would fail. Some recently proposed frameworks bring the content of the items into consideration. For example, there are works that extend LFM to incorporate item features [77][78], and there are also works considers item features, user features, and global features [79] . However, these approaches can only handle the cold-start problem to some extent since they rely on factorizing the user-item matrix or using it to optimize the models. If all the items are new items, which is very common in real applications, especially when launching a new recommender system,

these improved approaches would still fail. Only content-based recommendation can be applied at this stage before enough user profiles can be gathered.

A few studies have attempted to bring visual content analysis into the scope of content-based recommendations. Mei et al. [80] presented a contextual video recommender system, called VideoReach, which fuses three models based on textual, visual, and aural information, respectively. Video relevance scores from different models are calculated using different distance functions. The weights for shot features are adjusted based on user click-through behaviors on a video. For example, fast forwarding may indicate that the user is not interested in this shot so the weight of this shot should be decreased. The weights of different kinds of features in a single modality and the weights among three modalities are adjusted using relevance feedback by classifying the recommended videos into positive and negative samples based on the lengths that they are watched. If a video is only watched by a small proportion, then this video is considered a negative sample and the weight of the model that recommends this video should be decreased. Attention Fusion Function is applied, which first filters out most of the videos with low textual relevance since textual information is usually more reliable than visual and aural information. Then the relevance scores from these three modalities are combined using the linear weighted appraoch. Online evaluation is performed with 20 subjects using about 6000 videos from MSN Video[1]. A similar framework was presented by Luo et al. [81], where audio, textual, and visual information are first synchronized to detect the predefined topics in news videos. The recommendation strategy recommends the top 5 ranked videos for a given topic as well as the videos of related topics in the topic network. The ranking strategy considers time factor, visiting times,

---

[1]http://video.msn.com/video.aspx?mkt=en-us&tab=soapbox/

and qualities. The evaluation showed that the results of topic detection using the combined information sources were better than the results using a single source, but no concrete experiment was conducted to evaluate the recommendation strategy.

Chapter 3

MCA-based Feature Selection Component

This chapter presents the MCA-based feature selection component, which relies on Multiple Correspondence Analysis (MCA) as introduced in Section 3.1. Two methods are proposed in this component, which are MCA-based feature selection and MCA-based discretization, as illustrated in Section 3.2.1 and Section 3.2.2, respectively. MCA technique can also go beyond feature selection and extend to classification. Section 3.2.3 presents the application of MCA in classification. Experiments are conducted to evaluate each of the proposed method, and their effectiveness are shown in the experimental results in Section 3.3.

3.1   Multiple Correspondence Analysis

Standard Correspondence Analysis (CA) is a descriptive/exploratory technique designed to analyze simple two-way contingency tables containing some measure of correspondence between the rows and columns. Multiple Correspondence Analysis (MCA) can be considered as an extension of the standard CA to more than two variables [82].

The procedure of MCA is divided into the following steps. First, an indicator matrix (i.e., a two-way frequency cross tabulation table) with instances as rows and intervals

of variables as columns is constructed. Assume there are M variables, denoted as a set of vectors $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_m, ..., \boldsymbol{v}_M\}$, each has K_m, $m \in [1,M]$ intervals respectively. Let K denote the total number of intervals of all the variables, that is $K = \sum_{m=1}^{m=M} K_m$ and N denote the total number of data instances, so the size of the indicator matrix (denoted by $\boldsymbol{Z}$) is $N \times K$.

Next, instead of analyzing the indicator matrix $\boldsymbol{Z}$ as in correspondence analysis (CA), the inner product of the indicator matrix $\boldsymbol{Z}^T \boldsymbol{Z}$, also called the Burt Matrix $\boldsymbol{B}$ (with the size of $(K \times K)$, is analyzed in MCA. Dividing $\boldsymbol{B}$ by the total number of data instances results in the probability matrix $\boldsymbol{P}$ with each element denoted as p_{ij}, where i and j are from 1 to K.

Then, centering is performed on $\boldsymbol{P}$, which calculates the differences between the observed and expected relative frequencies. If $\boldsymbol{r}$ and $\boldsymbol{c}$ are the row and column mass vectors of $\boldsymbol{P}$, then each element of $\boldsymbol{r}$ and $\boldsymbol{c}$ is defined as $r_i = \sum_j p_{ij}$ and $c_j = \sum_i p_{ij}$. Thus, the centering can be expressed as $(p_{ij} - r_i c_j)$, Normalization involves dividing these differences by $\sqrt{r_i c_j}$, which leads to a matrix of standardized residuals $\boldsymbol{S}$, with each element $s_{ij} = (a_{ij} - r_i c_j)/\sqrt{r_i c_j}$. Equation (3.1) gives the matrix expression of centering and normalization.

$$S = D_r^{-1/2}(P - rc^T)D_c^{-1/2}, where \tag{3.1}$$

$\boldsymbol{D}_r$ and $\boldsymbol{D}_c$ are diagonal matrices with these masses on the respective diagonals.

Finally, Singular Value Decomposition (SVD) is performed on $\boldsymbol{S}$. Let Σ be the diagonal matrix with singular values, $\Lambda = \Sigma^2$ be the diagonal matrix of the eigenvalues (also called principal inertia), the columns of $\boldsymbol{U}$ be the left singular vectors, the rows of $\boldsymbol{V}^T$ be the right singular vectors, and $\boldsymbol{Q}$ be the projection of $\boldsymbol{S}$ on $\boldsymbol{V}^T$ (with the size of

$K \times K$), we have $\boldsymbol{S} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T$. Now, the summation of each principal inertia is the total inertia, representing the amount that quantifies the total variance of $\boldsymbol{S}$. The first K_1 row vectors be the intervals of $\boldsymbol{v}_1$, the next K_2 row vectors be the intervals of $\boldsymbol{v}_2$, and so on. We can calculate the angles between the row vectors of each intervals of each variable. The MCA algorithm 1 is summarized in the following pseudo code.

Algorithm 1 MCA Algorithm

Input:
A set of variables $\{\boldsymbol{v}_m\}, m \in [1, M]$
Output:
The projection of the variables on the new space

Construct an indicator matrix $\boldsymbol{Z}$
Compute Burt Matrix $\boldsymbol{B} = \boldsymbol{Z}^T \boldsymbol{Z}$
Compute Probability Matrix $\boldsymbol{P} = \boldsymbol{B}/N$
$\boldsymbol{S} = \boldsymbol{D}_r^{-1/2} (\boldsymbol{P} - \boldsymbol{rc}^T) \boldsymbol{D}_c^{-1/2}$
$SVD(\boldsymbol{S}) = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T$
Project $\boldsymbol{S}$ on the subspace spanned by $\boldsymbol{V}^T$ as $\boldsymbol{Q}$

The aim of MCA is to represent the maximum possible variance in a map of a few dimensions. Usually, the first two dimensions could capture over 95% of the total variance. However, in some cases, one dimension is enough; while in other cases, more than two dimensions are preferred. Depending on the characteristics of the data set, a different number of the dimensions could be explored to meet the pre-defined variance requirement, which is generally fixed to 95% or 99% [83]. So instead of using the fixed first two dimensions [26], the number of dimensions is automatically decided that can well capture the variance. MCA provides a graphical representation of these variables by visualizing them as points in a low dimensional map (called the symmetric map). As shown in Figure 3.1, there are totally 9 points in a two dimensional symmetric map, representing the intervals from three variables v_1, v_2 and v_3, and each has 3, 4 and 2 intervals, respectively. The angle between the two points with respect

to the origin in the symmetric map measures the correlation between two intervals. A smaller angle indicates a stronger correlation. Therefore, take v_3 for example, v_1^3 has a higher correlation with v_3^1 compared to the rest of the points, the same to the correlation between v_1^2 and v_3^2 compared to the rest.
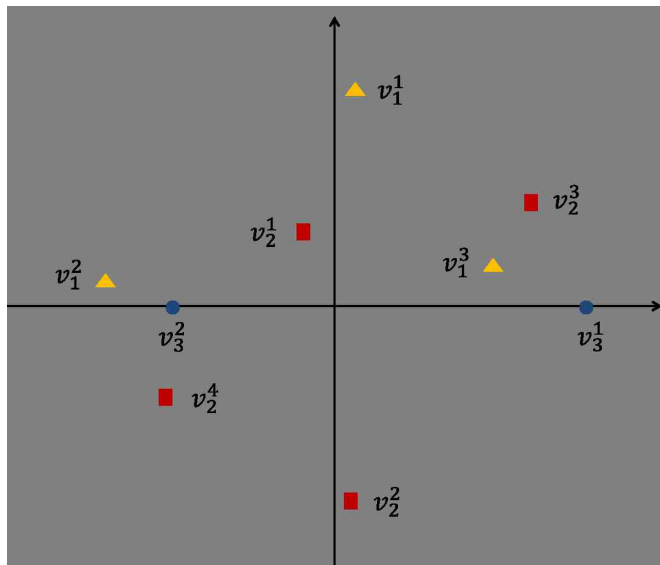
Figure 3.1: The symmetric map of the first two dimensions

The time complexity of SVD is $O(M^3)$, thus the SVD in MCA has a complexity of $O(K^3)$ where K is the dimension of the Burt matrix $\boldsymbol{B}$. If each time calculating the correlation between one feature and the class, then calculating M features would require $O(M(K_m + K_C)^3)$. Compared to PCA having $O(M^3)$, this shows effectiveness with increasing M, especially when handling features with less nominal values. Take the textual features for example, if using the binary representation, $K_m = 2$ where $m \in [1, M]$.

3.2 The Proposed Framework

A framework of the proposed MCA-based feature selection component is shown in Figure 3.2. Being a vital processing step, feature selection can reduce the cost of storage, decrease redundancy, and improve the performance of the model in these aspects. An effective subset of features should not contain (i) noisy features that decrease the retrieval accuracy, or (ii) irrelevant features that increase the computation time. Instead, it should contain those that have high predictive information and could better capture the semantic meaning of the query sample. Thus, a good feature selection can intrinsically help multimedia retrieval overcome these challenges. Therefore, feature selection is performed on both modalities.
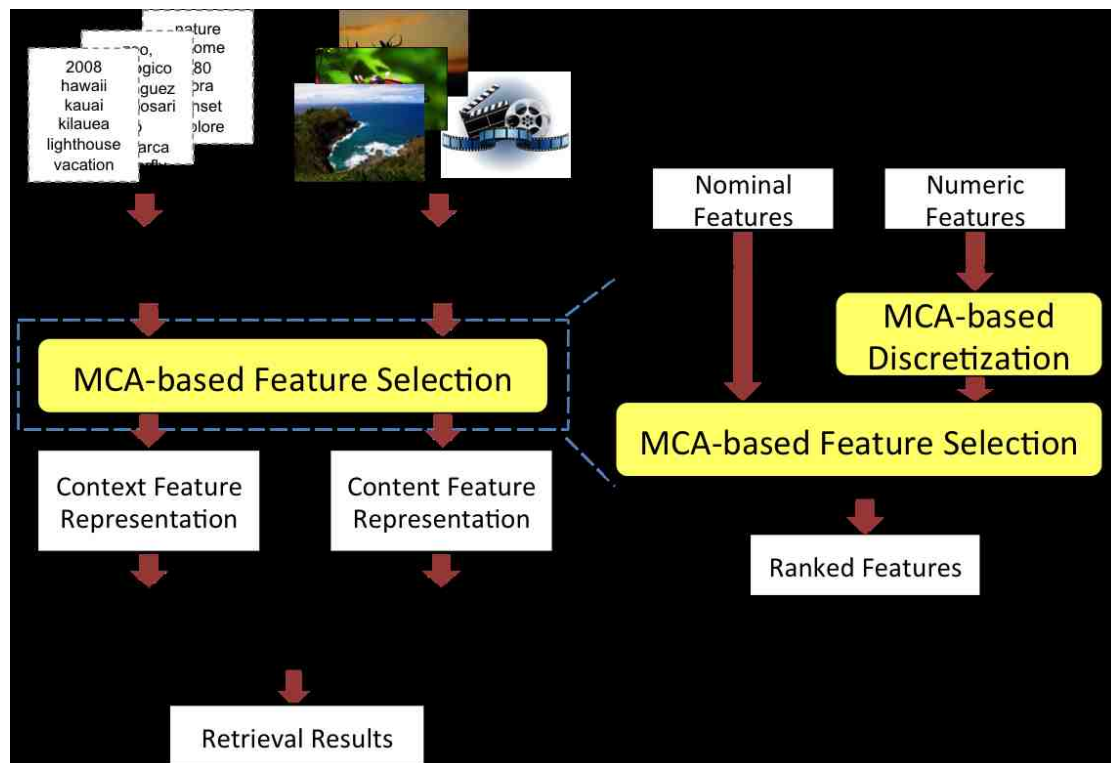
Figure 3.2: The framework of the MCA-based Feature Selection Component

3.2.1 The MCA-based Feature Selection Method

MCA is proved to be effective for capturing the correlations among nominal features by previous studies [84][85]. Features with a strong correlation relationship with the concept are kept, and the rest of the features are regarded as insignificant or noisy ones to be removed. Compared to the methods based on projection (e.g., principal component analysis) and compression (e.g., information theory) [86], the advantages of MCA-based approach are shown in two folds. First, the semantics of the concept can be captured in a more intuitive way. The retained features, especially textual features, can be easily interpreted according to their meanings. Second, textual features are usually represented by a value 0 or 1 to indicate their presence in an instance, and such a representation is rather sparse. MCA-based approach preserves this sparse structure of the instance-tag relationship, which is very efficient when training models.

Therefore , the feature selection component adopts the MCA-based feature selection method, which applies MCA to nominal feature values and classes. If features extracted are numeric, discretization is needed before applying MCA, so each feature would be discretized into multiple intervals, also called feature-value pairs. As shown in Figure 3.3, a nominal feature F with three feature-value pairs corresponds to three points in the map, namely F_1, F_2, and F_3, respectively. Considering a binary class, it is represented by two points lying in the x-axis, where C_1 denotes the positive class and C_2 denotes the negative class. Take F_1 as an example, the angle between F_1 and C_1 is a_{11}, and the distance between them is d_{11}. The meaning of a_{11} and d_{11} can be interpreted as follows:

- Correlation: This is the cosine value of the angle between a feature-value pair and a class in the symmetric map. It represents the percentage of the variance that the

feature-value pair point is explained by the class point. A larger cosine value that is equal to a smaller angle indicates a higher quality of representation.

- Reliability: The χ^2 distance between a feature-value pair and a class can be well represented by the Euclidean distance between them in the symmetric map. χ^2 distance could be used to measure the dependence between a feature-value pair point and a class point. Here, a derived value from χ^2 distance called the p-value is used because it is a standard measure of the reliability of a relation, and a smaller p-value indicates a higher level of reliability.
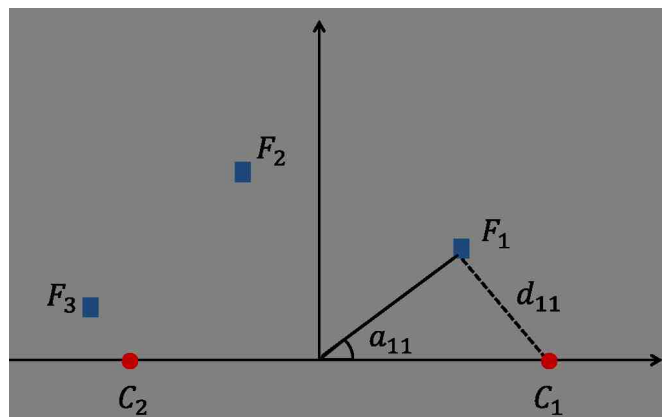
Figure 3.3: The symmetric map of the first two dimensions

For each feature, the angles and p-values between each feature-value pair of this feature to the positive and negative classes are calculated, corresponding to correlation and reliability, respectively. If the angle of a feature-value pair with the positive class is less than 90 degrees, it indicates this feature-value pair is more closely related to the positive class than to the negative class, or vice versa. For p-value, since a smaller p-value indicates a higher reliability, (1 - p-value) can be used as the probability of a correlation being true, except for the situation of Jeffrey-Lindley paradox [87]. This

paradox describes a situation when the p-value is very close to zero but the probability of the correlation being true is very close to zero as well. Such scenario could happen when the prior distribution is the sum of a sharp peak at H_0 with probability p and a broad distribution with the rest of the probability $1 - p$. In our experiments, it occurs when the count of the cross-table constructed by the feature-value pairs and the classes is less than 1% of the count of the corresponding class, which also makes sense since a rare occurrence can be considered as a "fluke".

After getting the correlation and reliability information of each feature-value pair with the class in the MCA calculation stage (represented by the angle values and p-values correspondingly), the equations which take the cosine value of an angle and p-value as two parameters are defined (as presented in Equations (3.2) and (3.3)) in the feature evaluation stage. Since these two parameters may play different roles in different data sets and both of them lie between $[0, 1]$, different weights can be assigned to these two parameters in order to sum them together as an integrated feature scoring metric. Considering different nominal features contain a different number of feature-value pairs, to avoid being biased to features with more categories like Information Gain does, the final score of a feature should be the summation of the weighted parameters divided by the number of feature-value pairs. For the feature F with K_F feature-value pairs, the angles and p-values for the kth feature-value pair are a_{1k} and p_{1k} for the positive class, and a_{2k} and p_{2k} for the negative class, respectively. Then the score of the F can be calculated through Equation (3.2) and (3.3).

$$\frac{\sum_{k=1}^{K_F} w_1 \cos a_{1k} + w_2 \max((1 - p_{1k}), p_{2k})}{K_F}. \tag{3.2}$$

$$\frac{\sum_{k=1}^{K_F} w_1 \cos a_{2k} + w_2 \max((1 - p_{2k}), p_{1k})}{K_F}. \tag{3.3}$$

If a feature-value pair is closer to the positive class, which means a_{1k} is less than 90 degrees, then Equation (3.2) is applied, where $\max((1 - p_{1k}), p_{2k})$ would allow us to take the p-value with both classes into account. This is because that $(1 - p_{1k})$ is the probability of the correlation between this feature-value pair and the positive class being true, and p_{2k} is the probability of its correlation with the negative class being false. Larger values of these two probabilities both indicate a higher level of reliability. On the other hand, if a_{1k} is larger than 90 degrees, which means the feature-value pair is closer to the negative class, then $\max((1 - p_{2k}), p_{1k})$ will be used instead, that is Equation (3.3). w_1 and w_2 are the weights assigned to these two parameters. Finally, after getting a score for each feature, a ranked list would be generated according to these scores, and then different stopping criteria can be adopted to generate a subset of features. Noticing that MCA is applied to each feature and the class label independently, so the score of each feature can be calculated in parallel first and then aggregate to generate the ranked list. In addition, the Burt Matrix counts the occurrence of the instances having certain feature value instead of using a matrix of the size equal to the occurrence as the indicator matrix in CA. This virtually allow infinite number of instances. Therefore, the model performance can be greatly improved using parallel computing and it is scalable to large data sets both feature wise and instance wise..

The MCA-based feature selection method can be used to remove noisy terms/ tags if treating each tag as a feature. Ideally, the remaining tags could predict the class label of the target concepts better. If using the binary representation for tag features, then MCA can be directly applied to these nominal features to produce the correlation of each feature-value pair of a tag feature with a concept class. According to Equation (3.2) and (3.3), tags with weak correlations with the concept classes can be removed.

3.2.2 The MCA-based Discretization Method

The extracted visual features are usually numeric features, which cannot directly be fed to the MCA-based feature selection component. This motivates us to look into feature discretization and explore MCA in solving this problem. Fig. 3.4 shows a numeric feature F with all values sorted to form a set of $D_F + 1$ distinct values $d_0, d_1, ..., d_{D_F}$, where d_0 and d_{D_F} are the minimum and maximum values of the feature, respectively. Candidate cut-points are the midpoints of all adjacent pairs in the set. F_1^q and F_2^q are two generated intervals given a candidate cut-point t^q, depicted in Fig. 3.4. Fig. 3.5 shows the symmetric map of two intervals F_1^q and F_2^q and three classes C_1, C_2 and C_3. a_{ik}^q is the angle between F_k^q and C_i, where $k = 1, 2$ and $i = 1, 2, 3$. For example, a_{11}^q represents the angle between F_1^q and C_1. Since the two intervals are negatively correlated with each other, the angle between them is 180 degrees, which means $cos(a_{i1}^q) = -cos(a_{i2}^q)$. Fig. 3.5 also shows the angles (when $i = 1$) between the two intervals and one class C_1, i.e., a_{11}^q and a_{12}^q. It can also be observed that if one interval is correlated with one class, the other interval is negatively correlated with this class to the same degree.
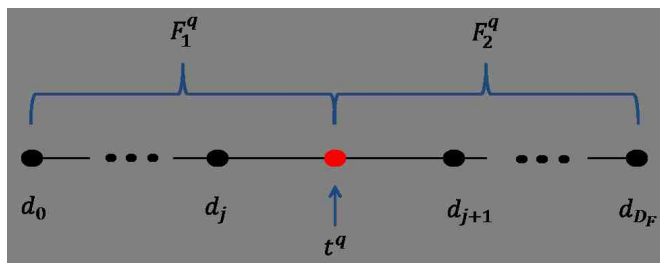
Figure 3.4: Candidate cut-points

If a_{ik}^q is much smaller than 90 degrees, it indicates that there is a higher correlation between F_k^q and C_i. The correlation between F_k^q and C_i can then be indicated by $|cos(a_{ik}^1)|$ or $|cos(a_{ik}^2)|$, but is measured by $(cos(a_{ik}^1))^2$ or $(cos(a_{ik}^2))^2$, which is also

known as the squared coefficient of correlation or quality. This motivates us to use the squared coefficient of correlation calculated from MCA to measure the quality of intervals generated by a candidate cut-point. A discretization scheme should contain the cut-points that maximize the correlation between the feature intervals and the classes, so the discretized feature could mostly indicate the information of the class labels when they are used for classification.
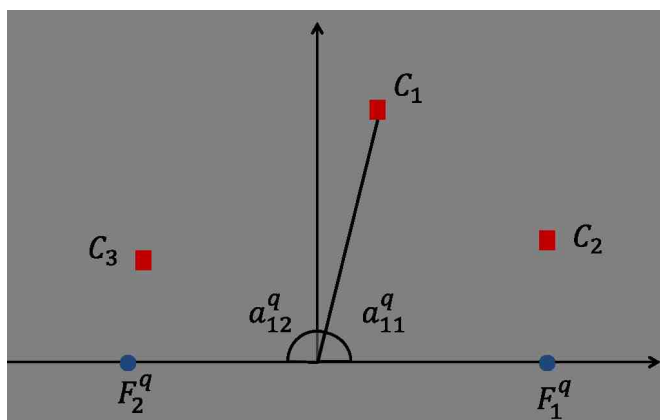
Figure 3.5: The symmetric map of the first two dimensions

According to Fig. 3.4 and Fig. 3.5, for a candidate cut-point t^q, since the left and the right intervals of t^q have the same squared cosine value of the angle with a class, the correlation between one interval and all classes can be used to measure the quality of the intervals generated by that candidate cut-point. For example, a^q_{11} can be used to indicate the correlation between interval F^q_1 and class C_1, and $(cos(a^q_{11}))^2$ is the value associated with t^q and C_1. Equation (3.4) measures the correlation between the generated intervals and all classes, called the *CM* (correlation maximization) value, and thus represents the "discretization quality" of t^q. The one with the largest *CM* value is selected as the first cut-point T_1. The same strategy can be carried out separately in the left and right intervals in a binary recursive way. However, when partitioning an interval

into sub-intervals, the number of data instances in the interval belonging to certain classes may be 0, which would result in failing the SVD calculation for Algorithm 1. Therefore, such classes need to be removed before applying Algorithm 1 to get the correlation information. Thus, in Equation (3.4), K_C is the number of classes in the current interval. In other words, K_C needs to be updated in each recursion according to the class information of the data instances in the current interval.

$$CM^q = \frac{1}{K_C} \sum_{i=1}^{i=K_C} (cos(a_{i1}^q))^2. \tag{3.4}$$

After selecting the cut-points, the question now is when to stop the splitting recursion. The idea is to consider the performance of the classifiers and terminate the recursion if the output measurement of the classifier is lower than the measurement obtained in the previous step, assuming a higher measurement value means a better classification result. This process is fully automatic, adaptive, and no parameter or threshold value needs to be tuned. Take the accuracy measurement as an example. If the accuracy of a classifier after accepting the cut-point is lower than the accuracy obtained earlier (i.e., without this cut-point in the previous step), then this cut-point is rejected and the recursion stops. Other possible measurement could be the weighted F1-score or the weighted area under the ROC curve, which is the sum of all values, and each is weighted according to the number of data instances with that particular class label. We used the accuracy as the classification measurement. Please note that different measurements can be adopted to tailor our discretization algorithm to a specific measure of a classifier. In addition, unlike the wrapper-based methods [55], training a classifier is only involved at the moment when deciding whether a selected cut-point should be accepted. Therefore, the efficiency is not affected much.

Suppose the cut-point T_1 that gives the maximum *CM* value has been selected from the candidate cut-points, generating the left interval F_1 and the right interval F_2. Within each interval, the same selecting procedure applies. Take F_1 as an example. The "discretization quality" of each candidate cut-point within F_1 is calculated and the candidate cut-point resulting in the highest *CM* value is selected as a cut-point of the interval F_1, denoted as T_2. Then a classifier is trained using the discretized F, with the discretization scheme (denoted as *DS*) having two cut-points T_1 and T_2. To avoid overfitting, cross-validation using the training data is adopted while training the classifiers. If the output measurement of the classifier is lower than that of a classifier trained using the discretized F with *DS* having cut-point T_1, then T_2 is rejected since the classification result decreases after T_2 is added into *DS*. Otherwise, T_2 is accepted and the partition continues in each further generated sub-intervals. The same criterion is adopted to decide whether to further partition the right interval F_2. Then the discretization scheme of F_1 denoted by *LDS* and the discretization scheme of F_2 denoted by *RDS* are combined to form the final *DS*. Algorithm 2 presents the pseudo code of the MCA-based Discretization algorithm.

3.2.3 The MCA-based Classification Framework

MCA technique can go beyond feature selection and extend to classification. This section present an MCA-based discriminative learning framework for classification, which includes the aforementioned MCA-based feature selection and MCA-based discretization, as well as an MCA-based classifier. This framework is an application of the MCA-based feature selection component, and is not part of the whole framework.

In classification, the goal is to identify the positive class or the target concept. The angle between a feature-value pair and the positive class has been analyzed in the feature

---

Algorithm 2 MCA-based Discretization

---

for each feature F **do**
    Set $pre\_Measure = 0$ {$pre\_Measure$ is the value of the measurement in the previous step}
    Initialize an empty discretization scheme DS
    Sort the distinct values of F in ascending order
    Calculate the midpoints of each adjacent pair as candidate cut-points
    Set $max\_CM = 0$ {max_CM is the maximum CM value in the current step}
    $DS = FN(F, pre\_Measure)$
end for

FUNCTION: $FN(F, pre\_Measure)$
Calculate the number of classes K in the current interval
for each candidate cut-point t^q **do**
    Perform MCA to get a_{i1}^q of each C_i
    Set $CM^q = \frac{1}{K_C} \sum_{i=1}^{K_C} (cos(a_{i1}^q))^2$
    if $CM^q > max\_CM$ **then**
        Set $max\_CM = CM^q$
    end if
end for
Set $Measure = classifier(F, DS)$ {$Measure$ is the output measurement of a classifier}
if $Measure > pre\_Measure$ **then**
    Set $T = t^q$
    Add T into DS
    $pre\_Measure = Measure$
    $LDS = FN(F_1, pre\_Measure)$ {LDS is discretization scheme of a left interval}
    $RDS = FN(F_2, pre\_Measure)$ {RDS is discretization scheme of a right interval}
    Combine LDS and RDS as DS
else
    return DS
end if

---

selection module. As mentioned before, the cosine value of the angle represents the percentage of the variance that is explained by the positive class. A larger cosine value which is equal to a smaller angle indicates a strong correlation between this feature-value pair and the positive class. Therefore, the cosine value of the angle between the feature-value pair and the positive class can act as the weight for that feature-value pair regarding to its discriminant capability. A transaction weight (TW) of the i-th instance can then be calculated by summing the weights of the feature-value pair along all the features and normalizing it, as shown in Equation 3.5, where W_i^j is the weight of the j-th feature of the i-th instance which is the cosine value of the angle between the corresponding feature-value pair and the positive class. The normalized TW of an instance can be regarded as the prediction score of an instance to be positive, and a classifier can be developed accordingly named as MCA-based classifier, which is introduced in [25][88].

$$TW_i = \frac{1}{M} \sum_{j=1}^{M} W_i^j \qquad (3.5)$$

An example of training data shown in Table 3.1 can then be transformed into Table 3.2 based on the weight of the feature value pair. A positive data instance is expected to have a larger transaction weight compared to a negative data instance since a feature-value pair with a larger weight indicates a stronger correlation with the positive class compared with a smaller weight. Therefore, these transaction weights can be treated as the prediction scores.

Figure 3.6 presents an example framework that adopts MCA-based feature selection, MCA-based discretization and MCA-based classification. First, visual features are extracted from raw videos. In order to evaluate the framework, three-fold cross val-

Table 3.1: An example of discretized training data set

	F^1	F^2	...	F^M
1	F^1_3	F^2_1	...	F^M_2
2	F^1_1	F^2_1	...	F^M_1
...	...	...	...	...
N	F^1_1	F^2_5	...	F^M_3

Table 3.2: Transaction weight of training data set

	W^1	W^2	...	W^M	TW
1	-0.71	0.57	...	-0.23	0.18
2	0.88	0.57	...	0.06	0.36
...	...	...	...	...	...
N	0.88	-0.12	...	0.86	0.47

idation is adopted to split the data into training data set and testing data set. Hence, the whole data set of each concept is randomly split into three sets with an approximately equal number of instances and equal positive to negative ratio. Next, MCA-based discretization is applied to the training set to discretize numeric features into nominal ones, and the same partitions are applied on the testing set. Then, MCA-based feature selection is performed on the training set. The correlation and reliability information generated from MCA are utilized to select two discriminative sets of features, one set for the positive class and the other one for the negative class. For the testing set, the same two sets of features obtained from the training set are selected. The component of MCA-based dual-model classification is enclosed in the dashed rectangular boxes. It contains two MCA-based classifiers, a positive model and negative model, which are trained by the two sets of features from the training set respectively using the aforementioned transaction weight. A strategy is introduced to fuse these two models into a more

powerful classifier to predict the class labels of the testing data instances. The detailed explanation about this framework can be found in [25].
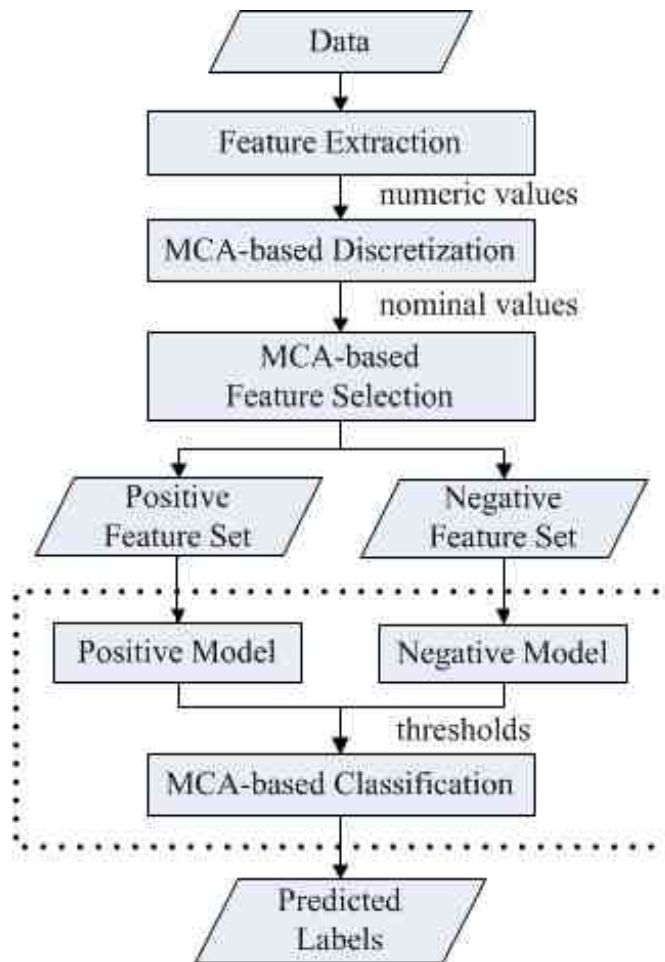
Figure 3.6: The symmetric map of the first two dimensions

3.3 Experimental Results

Experiments are conducted in three parts, namely feature selection, discretization, and classification. Noisy tag removal, as an important application of MCA-based feature selection, is also evaluated. Most of the comparison methods are reviewed in Chapter 2, a few targeting at a specific problem are discussed here.

Table 3.3: Concepts to be evaluated

No.	concept name	PN ratio
1	chair	0.074
2	infant	0.013
3	traffic-intersection	0.014
4	airplain-flying	0.027
5	person-playing-soccer	0.007
6	people-dancing	0.017
7	boat-ship	0.058
8	singing	0.074

3.3.1 Evaluation of the MCA-based Feature Selection

To evaluate the MCA-based feature selection method, experiments are first conducted by comparing it to four popular feature selection algorithms: information gain (IG), chi-square measure (CHI), correlation-based feature selection (CFS), and relief (REF). In order to find a good metric for feature selection that can improve classification accuracy, reduce computational complexity, and enhance semantic interpretability, the evaluation is conducted on the benchmark data from TRECVID 2009 video semantic concepts [89] with totally 48 features, and each instance is a frame or shot from a video. Eight highly imbalanced concepts are chosen to show the effectiveness of different feature selection methods on improving classification accuracy since the performance of the classifiers decreases enormously with an imbalanced data set. The concept numbers, names, and the positive to negative (PN) ratio are shown in Table 3.3.

Three-fold cross validation is first applied to the whole data set of each concept, which randomly splits the data into three sets with an approximately equal number of data instances and an equal PN ratio. Then each fold uses two of three sets as the training data set and the remaining one as the test data set. The final result is the average

of these three folds. To ensure fair comparison, the discretization method applied here is the minimum description length (MDL) [50] [51] provided by WEKA.

Next, all the five feature selection algorithms are performed on the discretized training data set, which also reduce the effect of discretization on our comparison. Then different ranked lists would be generated based on different algorithms, except CFS which automatically produces the preferred subset. For different concepts, the weights in Equations (3.2) and (3.3) are different. Considering both cosine angle value and p-value lie between $[0, 1]$, according to our experiment data, five trials of different ratios between w_1 and w_2, which are 0:1, 1:1, 1:2, 2:1, 1:0, are considered in the experiments to ensure the computational complexity is acceptable.

After applying these five feature selection methods, for ranker methods IG, CHI, REF and the proposed MCA-based algorithm, the generated training data set and the corresponding test data set are data with the sorted features, while for non-ranker CFS, the generated data is a subset of the data with pruned features. Then these five sets of data, one for each feature selection method, are run under five classifiers, namely Decision Tree (DT), Rule based JRip (JRip), Native Bayes (NB), Adaptive Boosting (Ada), and k-Nearest Neighbor classifier where k is 3 (k-NN). The stopping criterion used for the ranker methods is backward elimination which prunes the sorted features one by one backward after each time of classification. Each time, the precision, recall and F1-score of each classifier based on a particular subset of the features can be obtained. To conduct a complete search, 48 features require each classifier to repeat 47 times of classification on the sequentially decreased feature subspace produced by each ranker method. Based on the classification results, different feature subsets could be chosen for different comparison focuses. For example, for each feature selection method, the

subset that results in the highest F1-score can be chosen as the best subset, or the chosen subset could be the one with a minimum number of features but still produces relatively high classification results.

We compare the classification performance of these five classifiers when they are trained and tested using the subset generated by each feature selection method. Since CFS gives out the subset directly, in order to compare with it and not bias to any ranker method, for each concept, the same size of subspace as in CFS is chosen to evaluate each method. In Table 3.4 the average performance measures of five classifiers are shown for each concept. From these three tables, it can be observed that our proposed method outperforms the other four methods in precision, recall, and F1-score measures when the same size of feature subspace is used. On average (avg), it achieves 7% increase of the F1-score measure, and the standard deviation (std) across three folds is comparable to REF which is the best in these four methods. It can also be seen that the performance of IG and CHI are quite similar, and CFS is comparable to them given the size of the subspace is chosen based on it, while REF performs the worst. The number of features reduced as well as the ability to capture the semantics in the videos are reported in [90].

Noisy Tag Removal

Since the metadata from the context modality are usually very noisy, we applied MCA-based feature selection method to remove noisy tags (named as MCA-based tag removal or MCA-TR) and achieved fairly good results comparing to several tag removal methods reported in the literature. Its capability on removing noisy tags is evaluated using a light version (NUS-WIDE-LITE) and a full version (NUS-WIDE- 270K) of the NUS-WIDE data set [91] [58]. In NUS-WIDE-LITE, there are in total $55,615$ images with associated tags crawled from the Flickr website. The images from NUS-WIDE-LITE

Table 3.4: Average F1-score of 5 feature selection methods

No.	MCA-based	IG	CHI	REF	CFS
1	0.40	0.35	0.35	0.33	0.36
2	0.23	0.16	0.15	0.12	0.14
3	0.37	0.31	0.31	0.17	0.30
4	0.17	0.09	0.10	0.05	0.07
5	0.58	0.45	0.45	0.43	0.45
6	0.32	0.24	0.22	0.11	0.25
7	0.33	0.30	0.28	0.29	0.27
8	0.30	0.25	0.24	0.24	0.26
avg	0.34	0.27	0.26	0.21	0.26
std	0.03	0.08	0.07	0.04	0.11

has already been divided by the data set provider into training and test sets in advance, where $27,807$ images are used as the training set and the test set is composed of the rest $27,808$ images. NUS-WIDE-LITE also provides the ground truth of 81 concepts as well as $1,000$ frequent tags. NUS-WIDE-270K is similar to NUS-WIDE-LITE but contains $269,648$ images with tags and the provider splits this data set into a training set ($161,789$ images) and a test set ($107,859$ images).

Comparison methods include a state-of-art method proposed by [22] (LR_ES_CC_TC), a baseline method that adopts ICF [39] introduced in Chapter 2 and a singular value decomposition (SVD) method. We tuned the parameters of LR_ES_CC_TC to fairly compare it with our MCA-based feature selection algorithm. For the baseline method ICF, based on our empirical study, its best performance could be achieved by keeping those tags with ICF scores larger than 0.7 (i.e., those tags occurring in fewer than 50 concepts). This 0.7 is calculated using equation (2.1) with K=81 and cf_t=50. In the comparative SVD method, we keep all non-zero eigenvalues based on an empirical study on the training sets and then the transformed training and testing data are used

to evaluate the retrieval performance. For MCA-TR, we search the optimal threshold for MCA correlation on each training set from 0 to 0.1 with a step size 0.02 on NUS-WIDE-LITE and from 0 to 0.25 with a step size of 0.05 on NUS-WIDE-270K.

The MAP values of all the four methods on the NUS-WIDE-LITE and the NUS-WIDE-270K data sets are shown in Table 3.5 and Table 3.6, respectively. It shows that MCA-TR algorithm overall outperforms the other two methods from 0.63% to 8.7% on NUS-WIDE-LITE, and from 0.57% to 7.85% on NUS-WIDE-270K. Please note that although the result from LR_ES_CC_TC is close to ours, a lot of efforts were spent on tuning its parameters to get the best result. However, our tag removal algorithm does not need to tune any parameters since the MCA threshold corresponding to the best MAP on the training set is automatically selected, and it can dynamically remove trivial and irrelevant tags according to the target concept. Besides, although the SVD-based method is slightly worse than our proposed method, it takes much longer time to train the learning model on the transformed training and testing data given by the SVD-based method since the transformed data have lost the sparsity characteristics in the original image-tag matrix.

The precision-recall curves of all the four methods are shown in Fig. 3.7(a) (for NUS-WIDE-LITE) and Fig. 3.7(c) (for NUS-WIDE-270K). Fig. 3.7(b) and Fig. 3.7(d) show the precision-recall curves of the concept "grass" in NUS-WIDE-LITE and the concept "water" in NUS-WIDE-270K as examples, which show the performance of MCA-TR method is better than the other comparative approaches.

We also observed in our experiment that the threshold for our MCA-TR algorithm is 0 for a number of concepts, which indicates that it is unnecessary to remove tags from the original tag set for these concepts. For NUS-WIDE-LITE, there are 23 concepts

Table 3.5: MAP of the 81 concepts on NUS-WIDE-LITE with one-split

	ICF	LR_ES_CC_TC	SVD	MCA-TR
MAP	0.2430	0.2971	0.3237	**0.3300**
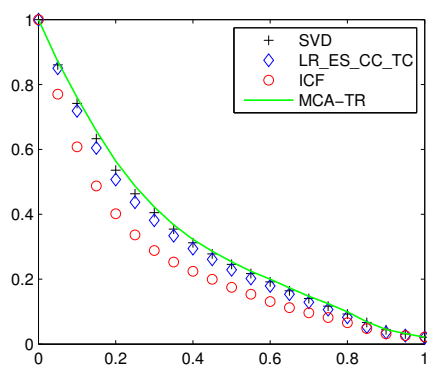
(a) Precision-Recall Curve on NUS-WIDE-LITE
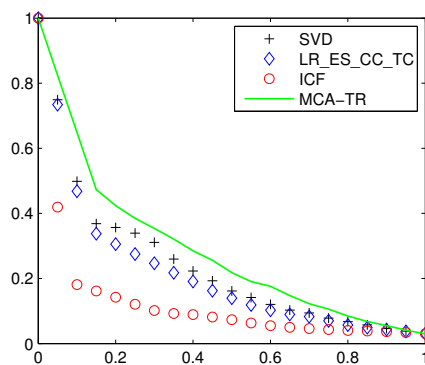
(b) Precision-Recall Curve of the Concept "grass" from NUS-WIDE-LITE
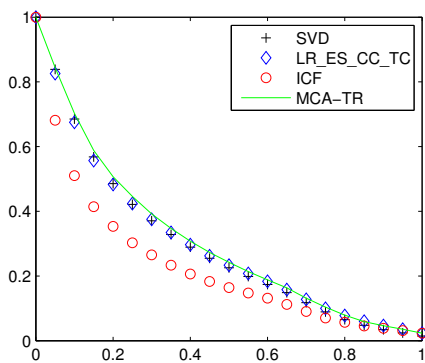
(c) Precision-Recall Curve on NUS-WIDE-270K
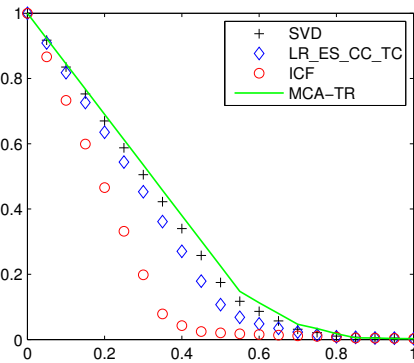
(d) Precision-Recall Curve of the Concept "water" from NUS-WIDE-270K

Figure 3.7: Precision-Recall Curves

Table 3.6: MAP of the 81 concepts on NUS-WIDE-270K with one-split

	ICF	LR_ES_CC_TC	SVD	MCA-TR
MAP	0.2021	0.2680	0.2749	**0.2806**

Table 3.7: MAP of 23 concepts before and after MCA-TR on NUS-WIDE-LITE

	Before	After
MAP	24.5%	27.1%
Average # of tags	999	246

whose MCA correlation thresholds are greater than 0. Therefore, we further show the results of the 23 concepts whose tags actually have been filtered by MCA-TR. Fig. 3.8 shows the ratios of the retained tags to the total tags after MCA-TR is applied. As can been seen from Fig. 3.8, the concepts such as "castle", "coral", and "moon" only retain less than 15% of the tags, which means more than 85% of the tags are removed for these concepts. That is very promising in terms of reducing computational cost and saving storage space. Table 3.7 further summarizes the average MAP improvement and the average number of tags retained before and after using MCA-TR. The results show that MCA-TR can averagely increase the MAP values at about 2.6% for those selected 23 concepts; while the size of retained tag set can have about 75% reduction.

For NUS-WIDE-270K, there are totally 46 concepts whose tags are actually filtered by MCA-TR. Therefore, we further show the results of these 46 concepts before and after MCA-TR is used. The retain ratios of these 46 selected tags before and after MCA-TR is applied are shown in Fig. 3.9. Table 3.8 further summarizes the average MAP improvement and the average number of tags retained before and after using MCA-TR. Although the MAP of these 46 concepts only have a little bit improvement after
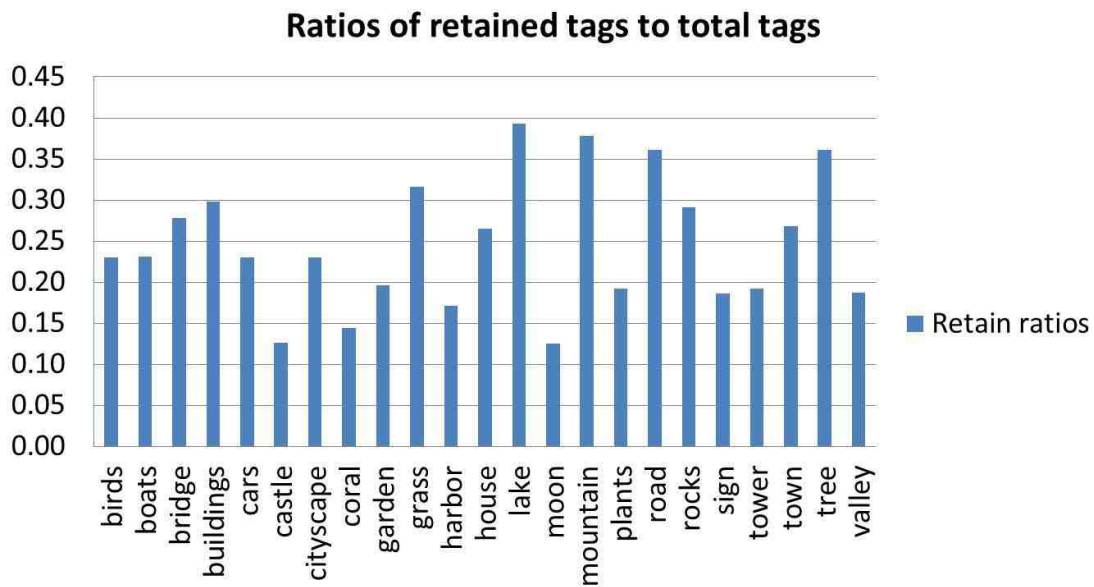
Figure 3.8: Retained tag ratios of selected 23 concepts on NUS-WIDE-LITE

Table 3.8: MAP of 46 concepts before and after MCA-TR on NUS-WIDE-270K

	Before	After
MAP	23.07%	23.42%
Average # of tags	999	319

MCA-TR, the dimensionality of the tags has reduced by more than 2/3. Considering the size of NUS-WIDE-270K, such a reduction in the dimensionality of the tags will significantly decrease the computational cost related to model training as well as the demand for the storage space.

To further reveal the effectiveness of MCA-TR against other comparative approaches, 3-fold cross-validation is conducted on NUS-WIDE-LITE. The experimental results are shown in Table 3.3.1. As can be seen from the results, MCA-TR can still outperform the other methods in terms of MAP. For those concepts that have been refined by MCA-TR,

Ratios of retained tags to total tags
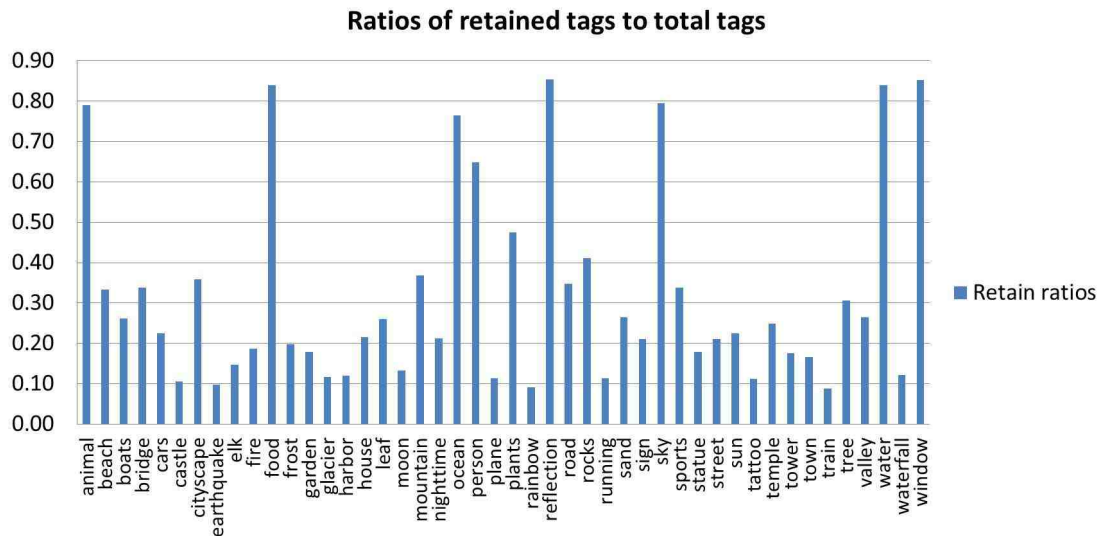
Figure 3.9: Retained tag ratios of selected 46 concepts on NUS-WIDE-270K

Table 3.9: MAP of the 81 concepts on NUS-WIDE-LITE with 3-fold cross-validation

	ICF	LR_ES_CC_TC	SVD	MCA-TR
MAP	0.2330	0.3202	0.3312	**0.3367**

the average retained tags before and after applying MCA-TR are shown in Table 3.10, which indicates that MCA-TR achieves about 1.5% MAP gain, and at the same time removes around 75% tags. Furthermore, in order to demonstrate how significant our proposed MCA-based feature selection is, 5 times 3-fold cross-validation experiments are conducted. The results after the student t-test is applied to the MCA-TR and SVD methods are shown in Table 3.11. As can be seen from this table, MCA-TR can provide significantly better results than SVD since the one-tail p-value is close to 0 and the confidence level is almost 1.

Table 3.10: MAP of those refined concepts before and after MCA-TR on NUS-WIDE-LITE with 3-fold cross-validation

	Before	After
MAP	27.47%	29.05%
Average # of tags	999	238

Table 3.11: Significance test on NUS-WIDE-LITE with 5 times 3-fold cross-validation

	SVD	MCA-TR	one-tail p-value	confidence level
MAP	$0.3332(\pm 0.11\%)$	$0.3391(\pm 0.13\%)$	0.001	99.90%

3.3.2 Evaluation of the MCA-based Discretization

To evaluate the MCA-based discretization method, experiments are conducted using the Benchmark UCI data sets as shown in Table 3.12, with varied numbers of data instances, features, and classes. Since most data sets contain more than 2 classes, only the number of instances in the major class (the class with most number of instances) and the number in the minor class (the class with least number of instances) are listed in Table 3.12. 3-fold cross-validation is applied to each data set with an approximately equal number of data instances in each class. Each fold uses two of three subsets as the training data set and the remaining one as the test data set. Next, discretization is applied to the training data set, and the same discretization scheme obtained from the training data set is used to discretize the test data set. Then different classifiers are learned by the discretized training data set using the corresponding classifier to terminate the splitting recursion, and evaluated by the discretized test data set.

The performance of the MCA-based discretization algorithm is evaluated against seven state-of-the-art supervised discretization algorithms described in Chapter 2, namely

Table 3.12: UCI data sets

No.	name	instances	features	classes	major class	minor class
1	diabetes	768	8	2	500	268
2	glass	214	9	6	76	9
3	kdd_synthetic_control	600	60	6	100	100
4	letter	20000	16	26	813	734
5	liver-disorders	345	6	2	200	145
6	mfeat-factors	2000	216	10	200	200
7	mfeat-fourier	2000	76	10	200	200
8	mfeat-karhunen	2000	64	10	200	200
9	mfeat-morphological	2000	6	10	200	200
10	mfeat-zernike	2000	47	10	200	200
11	optdigits	5620	64	10	572	554
12	page-blocks	5473	10	5	4913	28
13	pendigits	10992	16	10	1144	1055
14	segment	2310	19	7	330	330
15	waveform	5000	40	3	1692	1653
16	harberman	306	3	2	225	81
17	hvwon	606	100	2	305	301
18	ionosphere	351	34	2	225	126
19	wdbc	569	30	2	357	212

IEM, IEMV, CAIM, CACC, UCAIM, AQEW and AQEF. Considering that a single classifier may bias to a certain discretization algorithm, six well-known classifiers in WEKA [92] are used to compare the classification results. They are Adaptive Boosting (Ada), Decision Tree (DT), Rule based JRip (JRip), k-Nearest Neighbor (k-NN) where k=3, Native Bayes (NB), and Support Vector Machine (Sequential Minimal Optimization) (SMO), all available in WEKA. Three evaluation metrics are used in the experiments: accuracy, F1-score, and area under curve (AUC). The F1-scores are weighted to evaluate multi-class classification. The weighted value is the sum of all values, where each is weighted according to the number of data instances in that particular class.

Table 3.13 summaries the performance of the six classifiers using different discretization methods on average of the 19 data sets, in terms of the accuracy, weighted

F1-score, and weighted AUC values. As can be clearly seen from the results, MCA outperforms IEM and IEMV for all the six classifiers in terms of both accuracy and F1-score values, while achieves the same or slightly better AUC values as compared to IEM and IEMV. Very similar performance in the accuracy, F1-score, and AUC results are obtained by the IEM and IEMV pair and the AQEW and AQEF pair. The three discretization methods based on the CAIR criterion are constantly inferior to MCA, and generally inferior to the entropy-based methods and the accuracy-based methods in terms of the accuracy, F1-score, and AUC values, but with a relatively large margin in the accuracy and F1-score values and a small margin in the AUC values. This observation is true to all six classifiers on most of the data sets. In addition, when the number of classes increases (such as data set No. 4, for example), both accuracy and weighted F1-score values drop significantly for CAIM, CACC, and UCAIM for all the six classifiers, especially for CAIM and UCAIM; while IEM, IEMV, AQEW and AQEF still achieve stable performance, and MCA produces very good results. Therefore, we can infer that CAIR is probably not a good discretization criterion for multi-class classification, and the criteria based on information entropy and correlation are likely to produce a discretization scheme with better classification results. On the other hand, CAIR-based discretization methods can achieve the same level or even better F1-scores than those of the entropy-based methods if the features are very noisy or don't contain much information about the classes [93][94].

Finally, the computational complexity of each algorithm is also examined. Due to the implementation issue, it is not fair by purely looking at the running time. Instead, the time complexity can be analyzed. All those six algorithms need to sort the distinct values in a feature. Suppose there are D candidate cut-points, it takes $O(Dlog_2(D))$ for

Table 3.13: Average accuracy, F1-score, and AUC values of the classifiers

Ada	IEM	IEMV	CAIM	CACC	UCAIM	AQEW	AQEF	MCA
Accuracy	0.80	0.79	0.73	0.76	0.74	0.78	0.77	**0.81**
F1-score	0.78	0.77	0.72	0.75	0.73	0.77	0.76	**0.80**
AUC	**0.88**	**0.88**	0.85	0.86	0.85	**0.88**	**0.88**	**0.88**
DT	IEM	IEMV	CAIM	CACC	UCAIM	AQEW	AQEF	MCA
Accuracy	0.76	0.75	0.71	0.74	0.72	0.73	0.72	**0.78**
F1-score	0.73	0.73	0.70	0.73	0.71	0.71	0.71	**0.76**
AUC	0.83	0.83	0.81	**0.84**	0.82	0.82	0.82	**0.84**
JRip	IEM	IEMV	CAIM	CACC	UCAIM	AQEW	AQEF	MCA
Accuracy	0.76	0.75	0.68	0.71	0.69	0.71	0.70	**0.77**
F1-score	0.73	0.73	0.67	0.70	0.68	0.70	0.70	**0.76**
AUC	0.83	0.83	0.79	0.82	0.80	0.81	0.80	**0.84**
k-NN	IEM	IEMV	CAIM	CACC	UCAIM	AQEW	AQEF	MCA
Accuracy	0.81	0.81	0.73	0.76	0.74	0.78	0.78	**0.83**
F1-score	0.79	0.79	0.72	0.74	0.72	0.77	0.78	**0.81**
AUC	**0.88**	**0.88**	0.85	0.87	0.85	0.87	**0.88**	**0.88**
NB	IEM	IEMV	CAIM	CACC	UCAIM	AQEW	AQEF	MCA
Accuracy	0.79	0.79	0.72	0.74	0.73	0.78	0.78	**0.80**
F1-score	0.77	0.77	0.71	0.73	0.71	0.77	0.77	**0.79**
AUC	**0.89**	**0.89**	0.86	0.87	0.85	**0.89**	**0.89**	**0.89**
SMO	IEM	IEMV	CAIM	CACC	UCAIM	AQEW	AQEF	MCA
Accuracy	0.82	0.82	0.73	0.76	0.74	0.81	0.81	**0.84**
F1-score	0.80	0.80	0.72	0.75	0.73	0.80	**0.81**	**0.81**
AUC	**0.86**	**0.86**	0.83	0.85	0.84	**0.86**	**0.86**	**0.86**

sorting. IEM, IEMV, AQEW, AQEF and MCA use a binary recursive way to partition the intervals, so the time complexity is $O(Dlog_2(D))$ in the worst case. While CAIM, CACC, and UCAIM check all the rest of the candidate cut-points at each round, so their time complexity is quadratic $O(D^2)$ to the number of data instances in the worst case. Furthermore, the above results also demonstrate that checking every candidate cut-point does not necessarily ensure a better discretization scheme for classification.

3.3.3   Evaluation of the MCA-based Classification Framework

To evaluate the MCA-based classification framework, experiments are conducted using 10 concepts from TRECVID 2009. Each concept data set has more than 12000 instances and 48 low-level visual features. These ten concepts range from slightly imbalanced (e.g., hand) to highly imbalanced (e.g., traffic-intersection), with positive to negative (PN) ratio shown in Table 3.14. Five classifiers, namely Decision Tree (DT), Rule based JRip (JRip), Native Bayes (NB), Adaptive Boosting (Adaboost) which uses DT as the basic classifier, and k-Nearest Neighbor (KNN) where k is set to 3, are used as comparisons, and the input of these five classifiers are the features selected by four feature selection methods: information gain (IG), chi-square measure (CHI), correlation-based feature selection (CFS), and relief filter (REF). So there are 20 combinations, each with one feature selection method combined with one classifier. Precision (pre), recall (rec), and F1-score (F1) which is the harmonic mean of precision and recall, are adopted as the evaluation metrics for classification.

The following five tables, from Tables 3.15 to Table 3.19, show the precision, recall and F1-score of five classifiers using four different feature selection methods. Since all the selection metrics only generate the ranked list of the features except CFS, which directly provides the selected features, forward selection is used to find the feature subset

Table 3.14: Concepts to be evaluated

No.	concept name	PN ratio
1	chair	0.07
2	traffic-intersection	0.01
3	person-playing-musical-instrument	0.04
4	person-playing-soccer	0.01
5	hand	0.25
6	people-dancing	0.02
7	night-time	0.06
8	boat-ship	0.06
9	female-human-face	0.10
10	singing	0.07

that gives the best performance of a classifier, evaluated by F1-score. For each feature selection method, the best performance produced by a feature subset is recorded in the tables. Table 3.20 shows the performance of the framework (denoted as MCA) as depicted in Figure 3.6. The same forward selection is used in the two models to select the feature sets. As can be seen from Figure 3.10 which shows the F1-scores of MCA and the best F1-scores from the five tables (cells indicated in dark) of each classifier for all 10 concepts, MCA performs the best in all 10 concepts regarding to F1-scores, which is the most important metric taking both precision and recall into account, followed by NB, k-NN and Adaboost. DT performs the worst. MCA outperforms the second best result by an average of 4% in F1-scores, and outperforms the worst one by an average of 15%. In terms of classification performance on the features selected by the four feature selection methods, CFS usually has the worst results except when using NB, 4 out of 10 concepts achieve the best F1-scores. The performance of IG and CHI are quite similar, and REF produces a slightly better result in DT compared to IG and CHI.

Table 3.15: Classification results of Adaboost

No.	CFS			IG			CHI			REF		
	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1
1	0.46	0.31	0.37	0.58	0.31	0.4	0.61	0.31	0.41	0.55	0.33	0.41
2	0.93	0.17	0.28	0.67	0.25	0.36	0.79	0.25	0.38	0.58	0.26	0.36
3	0.72	0.57	0.63	0.76	0.58	0.66	0.77	0.58	0.66	0.76	0.58	0.66
4	0.63	0.44	0.52	0.71	0.54	0.61	0.67	0.5	0.57	0.69	0.49	0.57
5	0.36	0.18	0.24	0.33	0.27	0.3	0.33	0.24	0.3	0.34	0.28	0.31
6	0.48	0.18	0.27	0.55	0.26	0.35	0.54	0.26	0.35	0.5	0.29	0.36
7	0.38	0.22	0.28	0.44	0.23	0.3	0.4	0.23	0.29	0.36	0.23	0.28
8	0.38	0.19	0.25	0.55	0.2	0.29	0.55	0.21	0.3	0.5	0.2	0.29
9	0.52	0.21	0.29	0.49	0.34	0.41	0.49	0.34	0.4	0.49	0.33	0.4
10	0.38	0.2	0.26	0.42	0.24	0.3	0.46	0.23	0.3	0.47	0.22	0.3

Table 3.16: Classification results of DT

No.	CFS			IG			CHI			REF		
	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1
1	0.7	0.23	0.34	0.69	0.23	0.34	0.68	0.24	0.35	0.69	0.25	0.37
2	0.93	0.17	0.28	0.93	0.22	0.36	0.93	0.22	0.36	0.86	0.24	0.37
3	0.76	0.51	0.61	0.79	0.51	0.62	0.79	0.51	0.62	0.8	0.51	0.62
4	0.65	0.2	0.3	0.78	0.26	0.39	0.77	0.25	0.36	0.75	0.3	0.4
5	0.5	0.09	0.15	0.45	0.15	0.22	0.46	0.15	0.22	0.44	0.16	0.23
6	0.6	0.11	0.19	0.6	0.17	0.27	0.58	0.18	0.27	0.68	0.19	0.29
7	0.54	0.15	0.23	0.44	0.19	0.26	0.45	0.19	0.26	0.46	0.19	0.27
8	0.57	0.15	0.23	0.52	0.2	0.29	0.52	0.21	0.3	0.55	0.2	0.29
9	0.5	0.18	0.27	0.56	0.28	0.37	0.56	0.28	0.37	0.55	0.26	0.36
10	0.64	0.15	0.25	0.59	0.15	0.24	0.56	0.16	0.24	0.54	0.15	0.24

Table 3.17: Classification results of JRip

No.	CFS			IG			CHI			REF		
	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1
1	0.69	0.26	0.37	0.64	0.29	0.4	0.66	0.29	0.4	0.63	0.29	0.4
2	0.92	0.18	0.3	0.89	0.22	0.35	0.82	0.22	0.35	0.79	0.23	0.36
3	0.73	0.57	0.64	0.76	0.58	0.65	0.78	0.54	0.64	0.74	0.6	0.66
4	0.62	0.35	0.44	0.55	0.56	0.55	0.53	0.56	0.55	0.62	0.55	0.58
5	0.47	0.1	0.16	0.43	0.14	0.21	0.47	0.15	0.23	0.46	0.13	0.2
6	0.52	0.15	0.24	0.55	0.21	0.3	0.5	0.22	0.3	0.48	0.23	0.31
7	0.51	0.21	0.3	0.45	0.2	0.33	0.48	0.25	0.33	0.44	0.27	0.33
8	0.59	0.2	0.3	0.55	0.24	0.33	0.53	0.24	0.33	0.59	0.23	0.32
9	0.5	0.23	0.32	0.53	0.37	0.43	0.49	0.38	0.43	0.53	0.38	0.44
10	0.51	0.13	0.21	0.54	0.18	0.27	0.52	0.2	0.29	0.51	0.2	0.28

Table 3.18: Classification results of KNN

No.	CFS			IG			CHI			REF		
	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1
1	0.71	0.27	0.39	0.68	0.31	0.43	0.69	0.31	0.43	0.69	0.3	0.42
2	0.95	0.2	0.33	0.91	0.23	0.36	0.91	0.23	0.36	0.83	0.23	0.35
3	0.8	0.53	0.64	0.85	0.58	0.69	0.85	0.56	0.68	0.85	0.57	0.68
4	0.85	0.39	0.52	0.88	0.49	0.62	0.89	0.5	0.64	0.87	0.47	0.61
5	0.43	0.11	0.18	0.42	0.16	0.24	0.42	0.16	0.24	0.43	0.16	0.24
6	0.61	0.14	0.23	0.65	0.16	0.26	0.7	0.16	0.27	0.66	0.15	0.25
7	0.46	0.16	0.23	0.47	0.19	0.27	0.46	0.19	0.27	0.47	0.19	0.27
8	0.59	0.16	0.25	0.62	0.18	0.28	0.59	0.18	0.27	0.6	0.18	0.28
9	0.52	0.23	0.32	0.52	0.31	0.39	0.52	0.31	0.39	0.56	0.3	0.39
10	0.6	0.16	0.25	0.63	0.2	0.3	0.61	0.19	0.29	0.61	0.2	0.3

Table 3.19: Classification results of NB

No.	CFS			IG			CHI			REF		
	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1
1	0.5	0.32	0.39	0.48	0.33	0.39	0.47	0.34	0.39	0.46	0.35	0.4
2	0.43	0.25	0.31	0.45	0.19	0.26	0.52	0.2	0.28	0.27	0.23	0.24
3	0.6	0.6	0.59	0.59	0.62	0.6	0.61	0.6	0.6	0.59	0.64	0.61
4	0.33	0.74	0.46	0.3	0.81	0.44	0.3	0.8	0.44	0.31	0.8	0.44
5	0.34	0.29	0.31	0.33	0.37	0.35	0.33	0.37	0.35	0.33	0.37	0.35
6	0.3	0.35	0.32	0.31	0.31	0.31	0.31	0.31	0.31	0.14	0.53	0.22
7	0.27	0.6	0.37	0.31	0.43	0.36	0.32	0.42	0.36	0.27	0.52	0.36
8	0.3	0.4	0.34	0.37	0.35	0.36	0.39	0.35	0.37	0.39	0.34	0.37
9	0.48	0.33	0.39	0.45	0.48	0.46	0.45	0.48	0.46	0.34	0.63	0.45
10	0.43	0.3	0.35	0.38	0.34	0.36	0.38	0.35	0.36	0.38	0.33	0.35

Table 3.20: Classification results of the proposed framework

No.	MCA		
	pre	rec	F1
1	0.58	0.41	0.48
2	0.62	0.35	0.45
3	0.77	0.69	0.73
4	0.82	0.63	0.71
5	0.58	0.27	0.37
6	0.41	0.34	0.37
7	0.49	0.35	0.41
8	0.55	0.29	0.38
9	0.40	0.68	0.50
10	0.53	0.36	0.43

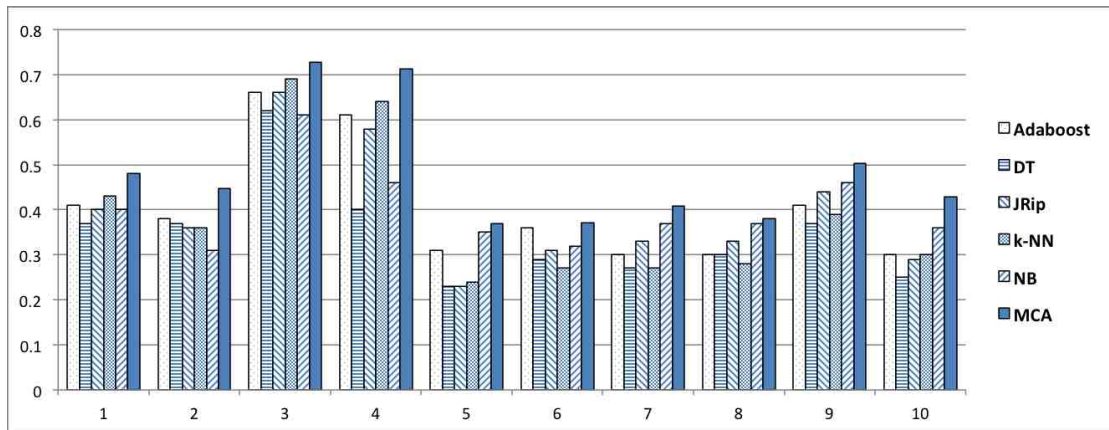Figure 3.10: Comparison of best F1-scores among six classifiers

3.4   Conclusions

The chapter presents the MCA-based feature selection component. Methods are proposed to utilize MCA technique to capture the correlation between a feature and the class labels. This calculation is done for each feature independently, so the proposed methods can be parallelized with each processor to handle the calculation of a set of features. Therefore, the component is scalable as the feature selection process can be decoupled by feature, and results of each feature can be aggregated to determine which features to filter.

The MCA-based feature selection utilizes the correlation and reliability information between a feature/attribute and the class captured by MCA. An effective scoring metric for feature selection is proposed which takes both angle values and p-values into account is developed to score features according to their prediction power of class labels. The MCA-based discretization method utilizes MCA to measure the correlation between feature intervals of a feature attribute and the classes. The candidate cut-point that maximizes the correlation between feature intervals and classes is selected as a cut-

point. This strategy is carried out in each interval recursively to further partition the feature. An MCA-based discriminative learning framework for video semantic classification is proposed to address the challenges such as semantic gap, imbalanced data, and high-dimensional feature space in multimedia semantic analysis. The correlation information is reutilized to build two models based on the transaction weights, and a strategy is introduced to fuse these two models into a more powerful classifier. Evaluation of each of the proposed methods are conducted by comparing them with representative work in the same area. Different data sets are used, including general UCI machine learning data sets and specific multimedia data sets. Experimental results demonstrate the effectiveness of each of the proposed methods.

Chapter 4

Sparse Linear Integration Component

This chapter presents the Sparse Linear Integration (SLI) component. Matrix factorization, as a closely related technique, is introduced in Section 4.1. Section 4.2 gives an overview of the component. The details of the model is illustrated in Section 4.2.1. Section 4.2.2 presents how to adopt the SLI model for supervised learning, and Section 4.2.3 generalizes the SLI model to the situation when the instances from different modalities are not in the same granularity. Experiments are conducted to evaluate the SLI model as well as its generalized SLI, with results shown in Section 4.3.

4.1 Matrix Factorization

Matrix factorization (MF) or matrix decomposition is a factorization of a matrix into a product of two or more matrices. It has been proved to be able to can discover latent factors underlining two kinds of entities. With its recent success in application of recommender systems [76], especially predicting ratings in collaborative filtering, MF has gain attention in the research community. Various algorithms and improvements are developed related to MF, such as non-negative matrix factorization (NMF) [95][96], which imposes a constraint that all the elements in the original matrix as well as the fac-

71

torized matrices are non-negative. Tensor factorization generalizes MF and can handle more than two entities [97][98]. This section introduces the basic MF technique as the Sparse Linear Integration (SLI) model is a special case of MF, which will be described in details in Section 4.2.

Given a matrix $\boldsymbol{R} \in \mathbb{R}^{M \times N}$, the goal is to find two matrices $\boldsymbol{P} \in \mathbb{R}^{M \times K}$ $\boldsymbol{Q} \in \mathbb{R}^{N \times K}$, where K is the number of the latent factors, such that their product approximate $\boldsymbol{R}$, as shown in Equation (4.1). The corresponding matrix illustration is depicted in Figure 4.1. Each row of $\boldsymbol{P}$ represents the associations between the corresponding row of $\boldsymbol{R}$ and the latent factors. Similarly, each column of $\boldsymbol{Q}$ represents the associations between the corresponding column of $\boldsymbol{R}$ and the latent factors. The bigger the value is, the stronger the association between them. By choosing a proper K, the underlyining associations between the rows and the columns of $\boldsymbol{R}$ can be well captured. K is usually smaller than M and N, thus this process is also referred as low-rank approximation. Equation (4.2) shows each value of $\boldsymbol{R}$ can be approximated by $\boldsymbol{P}$ and $\boldsymbol{Q}$.

$$\boldsymbol{R} \approx \boldsymbol{P} \times \boldsymbol{Q}^T \tag{4.1}$$
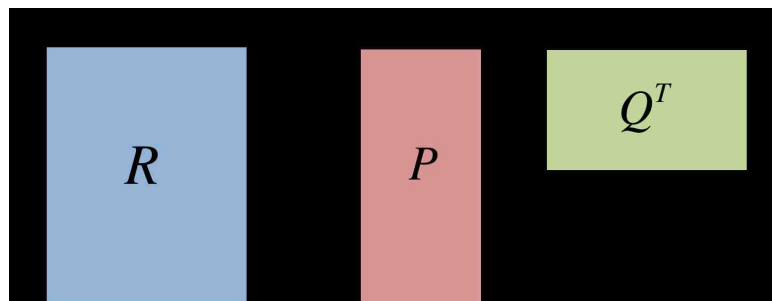
Figure 4.1: The matrix illustration of matrix factorization

$$\hat{r}_{ij} = p_i^T \times q_j \tag{4.2}$$

To accurately approximate $\boldsymbol{P}$ and $\boldsymbol{Q}$, the squared error between $\boldsymbol{R}$ and $\hat{\boldsymbol{R}}$ is used as the cost function to minimize, as shown in Equation (4.12). ℓ_F-norm is the Frobenius norm of matrix. The regularized term $\lambda\,(\|\boldsymbol{P}\|_F^2 + \|\boldsymbol{Q}\|_F^2)$ is often added to the cost function to limit the range of the values in $\boldsymbol{P}$ and $\boldsymbol{Q}$, which can prevent the model from overfitting. To solve this optimization problem, stochastic gradient descent [99] and alternating least squares [100] are two commonly adopted approaches.

$$\min_{\boldsymbol{P},\boldsymbol{Q}} \|\boldsymbol{R} - \boldsymbol{P}\boldsymbol{Q}^T\|_F^2 + \lambda\,(\|\boldsymbol{P}\|_F^2 + \|\boldsymbol{Q}\|_F^2) \tag{4.3}$$

4.2   The Proposed Framework

The framework of the sparse linear integration component is shown in Figure 4.2. Depending on whether it is a supervised learning process or not, the pairwise instance similarity could be one matrix for all the instances or one matrix for the instances belonging to the same class. For unsupervised learning, similarity between two instances can be directly used or can be utilized for later processing such as clustering. For supervised learning, a pairwise instance similarity matrix is learned per class using the positive instances in this class, then the reconstruction module is used to test how well a new instance can be represented using the training instances and the learned similarity between this new instance and each of the training instances. This proposed framework can also handle situations when instances from different modalities can not match. Take video concept detection for example, the concept can be detected at the shot level using visual features. However, the metadata is the description at the video level. Therefore, an instance is a shot in the content modality while an instance is a video in the context modality, which results in the granularity inconsistence issue. An association module is used to construct the associations between instances of different modalities, which can

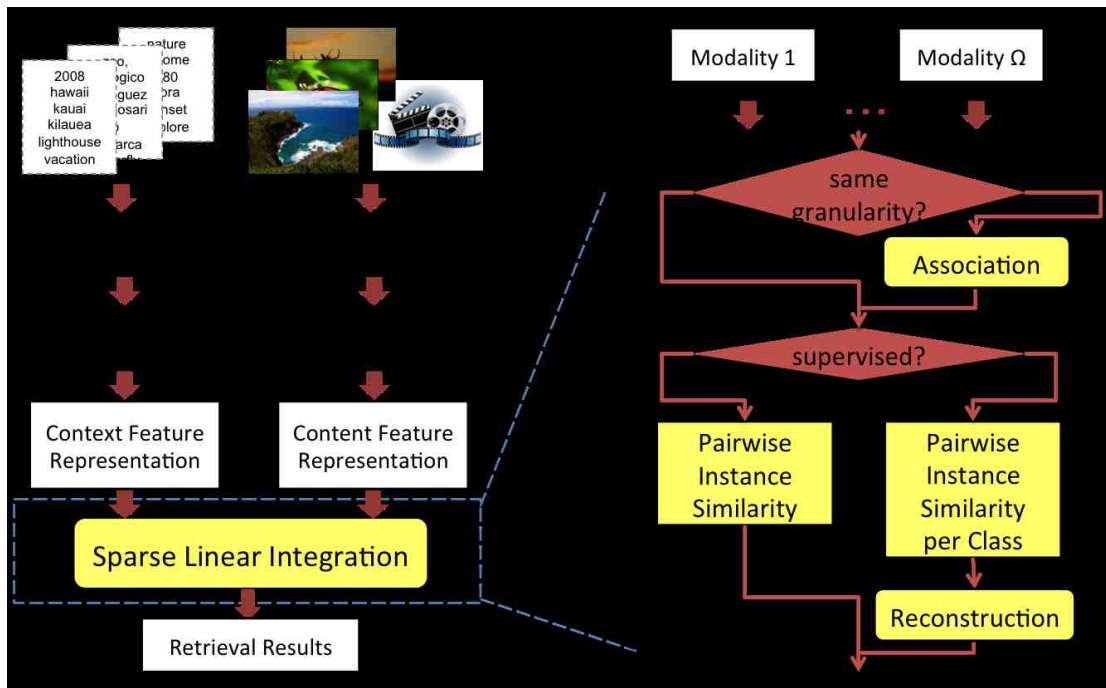generalize the pairwise instance similarity learning module and solve the granularity inconsistence issue.

Figure 4.2: The framework of sparse linear integration

4.2.1 Sparse Linear Integration

A sparse linear method was first introduced in [101] for top-n recommendation, which generated recommendation results by aggregating from user purchase or rating profiles. Later the authors extended the method to incorporate item content information [102], but the basic model was the same. Experiments on various data sets demonstrated high quality recommendations, and the sparsity of the coefficient matrix allowed to generate recommendations very fast. Inspired by this method, the sparse linear integration component combines multiple modalities at the intermediate level. A pairwise instance similarity matrix is learned, which can be viewed as the coefficient matrix in matrix

factorization. However, instead of factorizing the original feature representation into the basis matrix and the coefficient matrix of a low dimension in the latent space, the original feature representation is used as the basis matrix. Therefore, our method can be considered as a special case of matrix factorization. The advantage of this method is that no need to tune the dimension of the latent space as typically required in matrix factorization, which also prevents the potential information loss due to the low-rank approximation. In addition, we incorporates classification into the multimodal integration process, which tends to achieve a higher accuracy compared to methods adopting the early fusion and late fusion approaches that separate integration and classification.

To express and formulate the model in a clear way, the feature representation is denoted by a matrix with each feature or attribute as a row and each instance or item as a column, which is the transpose of the instance-feature matrix. Thus feature representation of the content and context modalities can be described by two feature-instance matrix $\boldsymbol{X}^t \in \mathbb{R}^{M^t \times N}$ and $\boldsymbol{X}^v \in \mathbb{R}^{M^v \times N}$, respectively. N is the total number of instances, M^t is the number of features of the context modality and M^v is the number of features of the content modality. Based on the assumption that there are instance-level consistence between two modalities, in other words, two instances would have a high correlation if they have similar textual representations as well as similar visual representations, and their correlation would be impaired if they are only similar in textual space or visual space or neither, the correlation coefficient between two instances can be learned by integrating the information from $\boldsymbol{X}^t$ and $\boldsymbol{X}^v$. SLI achieves this by updating the feature representation using a linear combination of the original feature representation weighted by these pairwise instance correlation coefficients. In order to get the updated feature representation, the goal is to learn a pairwise instance coefficient matrix $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ in

the updating process, as illustrated in Figure 4.3. Take the context representation $\boldsymbol{X}^t$ for example, Equation (4.4) expresses that the value of the i-th feature of the j-th instance is updated as a linear combination of the i-th feature of all the instances $((\boldsymbol{x}_i^t)^T \in \mathbb{R}^N)$ and the coefficients between the j-th instances and the other instances $(\boldsymbol{s}_j \in \mathbb{R}^N)$. If the j-th instance has a high correlation with the h-th instance, then the i-th feature of the h-th instance would contribute more to the updated value of the i-th feature of the j-th instance, and vice versa. Correspondingly, the update of the j-th column of $\boldsymbol{X}$ is as expressed in Equation (4.5). The same update applies to $\boldsymbol{x}_j^v$.
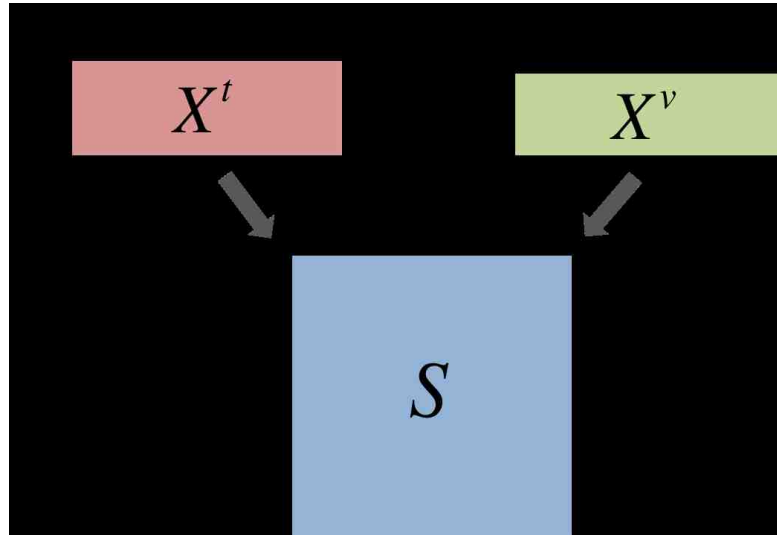
Figure 4.3: The matrix illustration of SLI

$$\boldsymbol{x}_{ij}^t \leftarrow (\boldsymbol{x}_i^t)^T \boldsymbol{s}_j \tag{4.4}$$

$$\boldsymbol{x}_j^t \leftarrow \boldsymbol{X}^t \boldsymbol{s}_j \tag{4.5}$$

Hence, the problem can be formulated into an optimization problem presented in Equation (4.6). The terms $\|\boldsymbol{x}_j^t - \boldsymbol{X}^t \boldsymbol{s}_j\|_F^2$ and $\|\boldsymbol{x}_j^v - \boldsymbol{X}^v \boldsymbol{s}_j\|_F^2$ measure how well the up-

date fits $\boldsymbol{x}_j^t$ and $\boldsymbol{x}_j^v$. α_j^t and α_j^v are their associated weights. The term $\|\boldsymbol{s}_j\|_2^2$ and $\|\boldsymbol{s}_j\|_1$
are the ℓ_2-norm and ℓ_1-norm regularization terms, respectively, and β and γ are their
regularization parameters. A larger regularization parameter imposes a severe regular-
ization. ℓ_1-norm is introduced to get a sparse solution of $\boldsymbol{S}$ [103], which can make the
updating process of Equation (4.5) very fast, especially when dealing with big data. It
also has effect on noise removal, which has been extensively used in image process-
ing [104][105][106]. ℓ_2-norm can restrict parameter range and prevent model from
overfitting. The two regularization terms together lead the optimization problem to an
elastic net [107], which balances between the lasso using ℓ_1-norm and ridge regression
using ℓ_2-norm. The constraint $diag(\boldsymbol{S}) = 0$ is applied to avoid trivial solutions [102],
that is the optimal $\boldsymbol{S}$ is an identical matrix such that an instance is always best related
to itself and not related to any other instance. The constraint $(\alpha_j^t)^2 + (\alpha_j^v)^2 = 1$ is to
balance the weight between the two modalities.

$$\min_{\boldsymbol{s}_j,\alpha_j^t,\alpha_j^v} \quad \frac{(\alpha_j^t)^2}{2}\|\boldsymbol{x}_j^t - \boldsymbol{X}^t\boldsymbol{s}_j\|_2^2 + \frac{(\alpha_j^v)^2}{2}\|\boldsymbol{x}_j^v - \boldsymbol{X}^v\boldsymbol{s}_j\|_2^2$$
$$+ \frac{\beta}{2}\|\boldsymbol{s}_j\|_2^2 + \gamma\|\boldsymbol{s}_j\|_1$$

(4.6)

$$s.t. \quad s_{jj} = 0$$
$$(\alpha_j^t)^2 + (\alpha_j^v)^2 = 1$$

Using Lagrange multiplier, solving Equation (4.6) is equivalent to solve Equation
(4.7). For simplicity, $\boldsymbol{x}_j = [(\alpha_j^t\boldsymbol{x}_j^t)^T, (\alpha_j^v\boldsymbol{x}_j^v)^T]^T$ is used in the following derivation when-
ever applicable, where $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ and $M = M^t + M^v$. Let J denote the cost function in
Equation (4.7), which is depended on $\boldsymbol{s}_j$, α_j^t and α_j^v. All the terms in J are differentiable
except $\|\boldsymbol{s}_j\|_1$. A global minimum of J can be found using coordinate descent [108]. The

partial derivative of J with respect to the i-th entry of $\boldsymbol{s}_j$ is derived as Equation (4.8), and the update form s_{ij} is shown in Equation (4.9), where Υ is the soft-thresholding operator. Similarly, taking the partial derivative of J with respect to α_j^t and α_j^v can get the update form of these two variables, as shown in Equation (4.10). Repeat the update of each of the variables for a certain number of cycles, together with the partial derivative of J with respect to λ or equivalently the constraint $(\alpha_j^t)^2 + (\alpha_j^v)^2 = 1$, J is converged and the optimal values of $\boldsymbol{s}_j$, α_j^t and α_j^v are reached. Aggregating $\boldsymbol{s}_j$ and the corresponding α_j^t and α_j^v, we can get the final solution of $\boldsymbol{S}$, and $\boldsymbol{\alpha}^t$ and $\boldsymbol{\alpha}^v$. Note that $\boldsymbol{\alpha}^t$ and $\boldsymbol{\alpha}^v$ are vectors in $\mathbb{R}^N$ since α_j^t and α_j^v are associated with a column of $\boldsymbol{S}$. The regularization parameters β and γ are typically tuned using grid search and do not vary much for different $\boldsymbol{s}_j$. Apparently, the SLI model can be parallelized using multiple processors with each one handling a couple of columns of $\boldsymbol{S}$.

$$
\begin{aligned}
\min_{\boldsymbol{s}_j, \alpha_j^t, \alpha_j^v} \quad & \tfrac{1}{2}\|\boldsymbol{x}_j - \boldsymbol{X}\boldsymbol{s}_j\|_2^2 + \tfrac{\beta}{2}\|\boldsymbol{s}_j\|_2^2 + \gamma\|\boldsymbol{s}_j\|_1 \\
& + \lambda\left((\alpha_j^t)^2 + (\alpha_j^v)^2 - 1\right)
\end{aligned}
\tag{4.7}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial s_{ij}} \\
= & -\sum_{h=1}^{h=M} x_{hi}\Big(x_{hj} - \sum_{g=1}^{g=N} x_{hg}s_{gj}\Big) + \beta s_{ij} + \gamma \\
= & -\sum_{h=1}^{h=M} x_{hi}\Big(x_{hj} - \sum_{g\neq i} x_{hg}s_{gj} - x_{hi}s_{ij}\Big) + \beta s_{ij} + \gamma \\
= & -\sum_{h=1}^{h=M} x_{hi}\Big(x_{hj} - \sum_{g\neq i} x_{hg}s_{gj}\Big) + \sum_{h=1}^{h=M} x_{hi}^2 s_{ij} + \beta s_{ij} + \gamma \\
= & -\sum_{h=1}^{h=M} x_{hi}\Big(x_{hj} - \sum_{g\neq i} x_{hg}s_{gj}\Big) + \Big(\sum_{h=1}^{h=M} x_{hi}^2 + \beta\Big)s_{ij} + \gamma
\end{aligned}
\tag{4.8}
$$

$$s_{ij} \quad \leftarrow \frac{\Upsilon(\sum_{h=1}^{h=M} x_{hi}(x_{hj} - \sum_{g \neq 4i} x_{hg}s_{gj}), \gamma)}{\sum_{h=1}^{h=M} x_{hi}^2 + \beta}, where$$

$$\Upsilon(z, \gamma) \quad = \begin{cases} z - \gamma & \text{if } z > 0 \text{ and } |z| > \gamma \\ z + \gamma & \text{if } z < 0 \text{ and } |z| > \gamma \\ 0 & \text{if } |z| \leq \gamma \end{cases} \qquad (4.9)$$

$$\begin{aligned} \alpha_j^t &\leftarrow \frac{-2\lambda}{\|x_j^t - X^t s_j\|_2^2} \\ \alpha_j^v &\leftarrow \frac{-2\lambda}{\|x_j^v - X^v s_j\|_2^2} \end{aligned} \qquad (4.10)$$

To summarize, the SLI model consists two steps, which capture the essence of equation (4.6) or equation (4.7). The first step is concatenating X^t and X^v by rows with weight α^t and α^v. Compared to the simple feature concatenation adopted by most of the early fusion approaches, a weight is added to each feature representation of an instance. This can prevent the values of X^t and X^v from overshadowing each other due to their different ranges. Another factor that could also cause this overshadowing issue is the feature dimensions in X^t and X^v. If they are in different scales, the values in S would also lean towards the one with a higher feature dimension. The second step is using the weighted concatenation of X^t and X^v, denoted as X to get S by solving equation (4.7). The above illustration is based on two modalities. Equation (4.6) can be extended to Equation (4.11) for multiple modalities, where x_j^ω is the feature-instance matrix of the ω-th modality and P is the total number of modalities. Therefore, other modalities in multimedia data such as audio can be included as well.

$$\min_{s_j, \alpha_j^\omega} \sum_{\omega=1}^{\omega=\Omega} \frac{(\alpha^\omega)^2}{2} \|x_j^\omega - X^\omega s_j\|_F^2 + \frac{\beta}{2} \|s_j\|_F^2 + \gamma \|s_j\|_1$$

$$\text{(4.11)}$$

$$s.t. \quad s_{jj} = 0$$
$$\sum_{\omega=1}^{\omega=\Omega} (\alpha_j^\omega)^2 = 1$$

4.2.2 The SLI Model for Supervised Learning

So far the SLI model is an unsupervised process, which learns a pairwise instance similarity matrix S. It can be directly used to find similar instances for a query instance. This model can also be adopted for supervised learning. In the training phase, for each class C, X only include the positive instances of this class. If the total number of instances is still N, and the number of positive instances of class C is N_C, then $X \in \mathbb{R}^{M \times N_C}$ and $S \in \mathbb{R}^{N_C \times N_C}$. Follow equations (4.6)(4.7)(4.9)(4.10), we can get the pairwise positive instance similarity matrix S and model parameters α^t, α^v, β and γ.

Sparse representation [109][110] is defined as the representation that account for most or all information of a signal with a linear combination of a small number of elementary signals called atoms. Often, the atoms are chosen from a so called overcomplete dictionary, whose number of atoms exceeds the dimension of the signal space, so that any signal can be represented by more than one combination of different atoms. Borrow the concept of the reconstruction error [111] from the sparse representation, an instance can be reconstructed from other instances weighted by the coefficient between them, which is captured by $\|x_j - X s_j\|_2^2$. Comparing to the sparse representation which introduces ℓ_1-norm to get a sparse solution of S, the proposed model adds ℓ_F-norm on top of it to further prevent the model from overfitting. Therefore, a classifier similar to the sparse representation-based classifier can be built using the reconstruction error gen-

erated from the instances of different classes. The probability of an instance belonging to a certain class is inversely proportional to the reconstruction error of this class.

In the test phase, a test instance y is treated as a new instance to the existing training set. Its context and content feature representations are denoted as $\mathbf{y}^t$ and $\mathbf{y}^v$, shown in Figure 4.4. The goal is to calculate $\tilde{\mathbf{s}} \in \mathbb{R}^{N_C+1}$, with each value indicating how similar each of the instances is to this test instance. Considering the constraint $\tilde{s}_{N_C+1} = 0$, equation (4.7) can be applied here to calculate $\tilde{\mathbf{s}} \in \mathbb{R}^{N_C}$ with the weighted concatenation $\mathbf{X} \in \mathbb{R}^{M \times N_C}$ and $\mathbf{y}$ replacing $\mathbf{x}_j$. $\mathbf{y}$ is also the weighted concatenation of $\mathbf{y}^t$ and $\mathbf{y}^v$, with the corresponding weights denoted as α_y^t and α_y^v. The coefficients in $\tilde{\mathbf{s}'} \in \mathbb{R}^{N_C}$ indicate how similar this test instance is to each of the training instances which can be discarded in the test phase. This information is only useful if we want to add y to the training set later on. After getting $\tilde{\mathbf{s}}$, the test error $err_C(y)$ for each class C is measured according to Equation (4.12). The probability of y belonging to c, denoted as $prob_C(y)$, can be generated from $err_C(y)$ using various mapping functions such as Gaussian kernel as shown in Equation (4.13).

$$err_C(y) = \|\mathbf{y} - \mathbf{X}\tilde{\mathbf{s}}\|_2^2 \tag{4.12}$$

$$prob_C(y) = \exp(-\frac{\|\mathbf{y} - \mathbf{X}\tilde{\mathbf{s}}\|_2^2}{2\sigma^2}) \tag{4.13}$$

4.2.3 The Generalized Model of SLI

Given a video collection with videos and their metadata, feature representations are probably generated at different granularities since metadata is descriptions about videos while visual features are extracted from shots within videos.

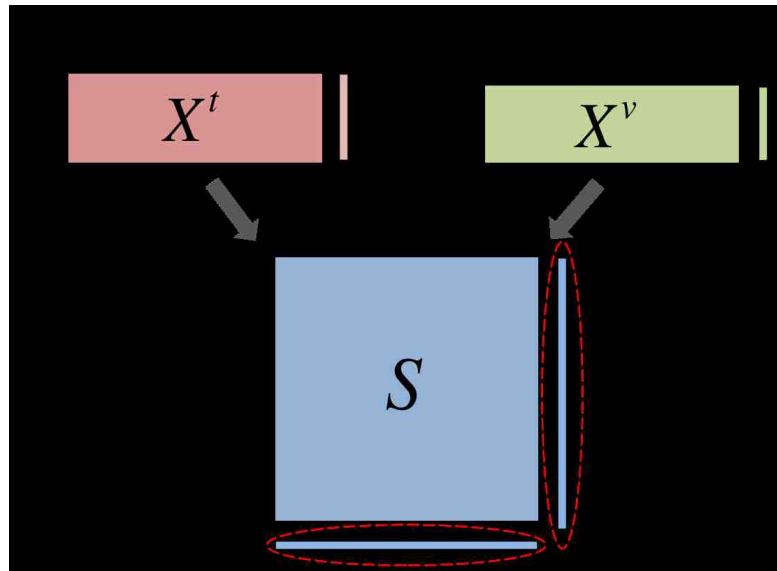For context modality, metadata isa first tokenized into a set of words or terms. Then

Figure 4.4: The matrix illustration of the update of SLI

standard procedures such as stop words removal and stemming are applied. Due to the noisy metadata of the Internet videos, such as typos and editing errors, English word validation using Wordnet [112] is applied to the metadata to only keep the valid English words. Textual features are extracted after this preprocessing by eliminating words with very low or very high frequencies. Binary representation is used to indicate the presence or absence of a feature. Thus a video is represented by a binary feature vector. The advantages of binary representation over common tf-idf used in IR lie in two folds: first, the metadata are typically short where a particular word only appear once, so tf-idf in this situation is essentially idf. After eliminating very rare words, the frequency of each word in the corpus does not provide much useful information any more. Second, due to the sparsity in textual features, a binary representation is more efficient in terms of storage and often the model training time as well. For the content modality, videos are first segmented into shots using shot boundary detection technique [113][114]. A key frame is extracted from each shot to represent the visual content of this shot. Then

several visual features are extracted from these key frames and are concatenated into a large feature vector. Therefore, each key frame or shot is represented by a feature vector.

To generalize the model to handle the aforementioned granularity inconsistence issue, an association matrix is defined to capture the association between instances of different modalities. Figure 4.5 shows an example of this association matrix. An entry in A is set to 1 is a shot is belong to a video, and 0 otherwise. For example, shots s_1 and s_2 belong to video v_1, and shots s_3, s_4 and s_5 belong to video v_2 in Figure 4.5. In this case, a shot can only belong to one video, but a video can contain multiple shots, thus the relationship between a video and a shot is one-to-many. In other applications, the relationship between instances of different modalities might be many-to-one or many-to-many. The values in A could also be decimal instead of binary to indicate the strength of associations.

Figure 4.5: An example of association matrix between videos and their shots

Depending on the targeted granularity, the modalities whose instances are in the same granularity do not need to alter their feature representations. However, the modalities whose instances are in different granularities need to transform their feature representations so that each column of the feature-instance matrix bears the same meaning.

For video concept detection, the context feature representation can be transformed by multiplying $\boldsymbol{A}$, as shown in Equation (4.14). The result is essentially replicating the columns of $\boldsymbol{X}^t$ according to the association indicated in $\boldsymbol{A}$. In other words, all the shots in the same video would have the same context representation, which is a valid assumption given no prior knowledge about shot-level context information. Then $\boldsymbol{X}^v$ and $\boldsymbol{X}^t \times \boldsymbol{A}$ would have the same granularity, which means their numbers of instances are the same. Equation (4.6) is now generalized to Equation (4.15), where $\boldsymbol{a}_j$ is a column of $\boldsymbol{A}$, and $j \in [1, N^v]$, where N^v is the number of instances in the content modality. Since $\boldsymbol{A}$ is known, the same solution based on coordinate descent and soft-thresholding can be applied to solve the problem.

$$\boldsymbol{X}^t \rightarrow \boldsymbol{X}^t \times \boldsymbol{A} \tag{4.14}$$

$$
\begin{aligned}
\min_{\boldsymbol{s}_j, \alpha_j^t, \alpha_j^v} \quad & \frac{(\alpha_j^t)^2}{2} \|\boldsymbol{X}^t \boldsymbol{a}_j - \boldsymbol{X}^t \boldsymbol{A} \boldsymbol{s}_j\|_2^2 + \frac{(\alpha_j^v)^2}{2} \|\boldsymbol{x}_j^v - \boldsymbol{X}^v \boldsymbol{s}_j\|_2^2 \\
& + \frac{\beta}{2} \|\boldsymbol{s}_j\|_2^2 + \gamma \|\boldsymbol{s}_j\|_1 \\
s.t. \quad & s_{jj} = 0 \\
& (\alpha_j^t)^2 + (\alpha_j^v)^2 = 1
\end{aligned}
\tag{4.15}
$$

4.3 Experimental Results

To evaluate the effectiveness of the sparse linear integration component, the retrieval results using this model are compared with several representative early fusion and late fusion approaches as well as the reported results in peer work.

4.3.1 Evaluation of the SLI Model for Supervised Learning

For image semantic concept retrieval, two benchmark data sets are widely used. One is MIRFLICKR-25000 collection [115] and the other one is NUS-WIDE-LITE [91]. Therefore, the experiments are conducted on these two data sets, and results are presented accordingly.

MIRFLICKR-25000 collection contains 25000 images and their associated tags from the Flickr website. 38 concepts are manually annotated for research purposes. Their concept IDs and names are listed in Table 4.1. It includes two types of labels: potential labels (24 concepts out of 38) and relevant labels(14 concepts out of 38). Potential labels of a concept are given to images as long as the concept is visible or applicable to some extent, while relevant labels are given to images only if the annotator found the image really relevant to his/her interpretation regarding to a certain concept. For completeness, all 38 concepts are used in the experiment. A standard way to split the training and test data sets are defined by this collection, 15000 out of 25000 are the training data and the rest 1000 are the test data. Their positive to negative (PN) ratios of the 38 concepts are depicted in Figure 4.6. The concept name ends with "_r1" denotes the concept having relevant labels. As can be seen, the data in MIRFLICKR-25000 ranges from highly imbalanced ones to relatively balanced ones.

To build the content modality, 4 types of features are extracted from the 25000 images, which are color moment in the YCbCr space [116], Local Binary Patterns (LBP) [36], histogram of oriented gradients (HOG) [37], and haar wavelets [117]. The total number of visual feature dimensions is 551. For each image, tags are assigned by Flclikr users, which probably contain typos, non-English words, unrelated tags, etc. Standard procedures such as stop word removal and stemming are applied to these tags.

Table 4.1: Names of the 38 concepts from MIRFLICKR-25000

1	animals	11	dog_r1	21	night	31	sea_r1
2	baby	12	female	22	night_r1	32	sky
3	baby_r1	13	female_r1	23	people	33	structures
4	bird	14	flower	24	people_r1	34	sunset
5	bird_r1	15	flower_r1	25	plant_life	35	transport
6	car	16	food	26	portrait	36	tree
7	car_r1	17	indoor	27	portrait_r1	37	tree_r1
8	clouds	18	lake	28	river	38	water
9	clouds_r1	19	male	29	river_r1		
10	dog	20	male_r1	30	sea		

English word validation is also used to validate each word by checking whether it exists in Wordnet [112], which is a large lexical database of English. Textual features are extracted from 10055 unique terms after this preprocessing. To maintain the textual features in the same scale as visual features, top 500 terms with the highest χ^2 values are selected. The binary representation is used to indicate the presence or absence of a feature.

The logic regression classifier is used to evaluate the performance of different feature representations. To show that content and context modalities can often provide complementary information, the results of AP@10 using content and context feature representations alone are displayed in Figure 4.7. The results of AP@10 are shown because this complementary characteristic is more notable in high ranked instances. As can also be observed from this figure, the context modality performs much better on some concepts than the content modality, such as "baby", "bird", "car", 'and 'dog". However, on concept "river_r1", the context modality completely fails. Therefore, our motivation of integrating content and context modalities can be proved on this data set.
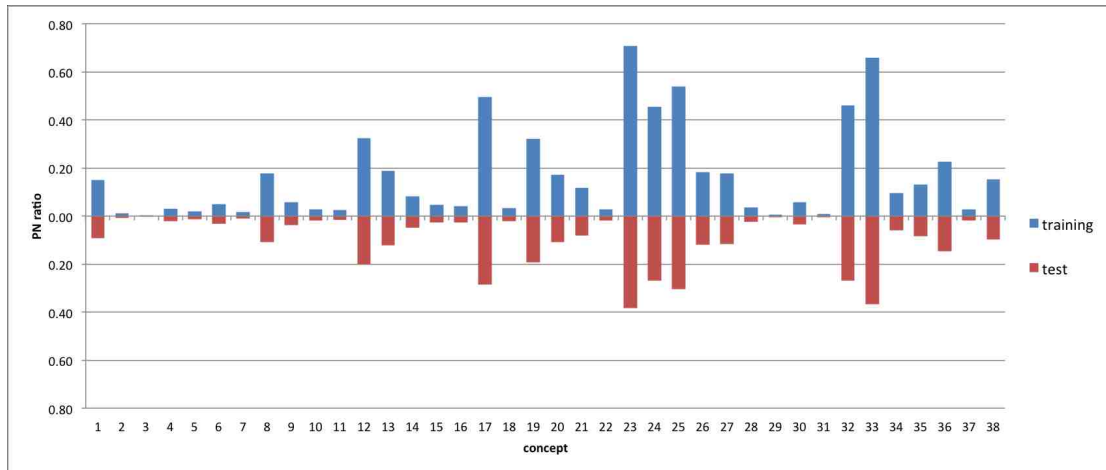
Figure 4.6: The PN ratios of the 38 concepts from MIRFLICKR-25000
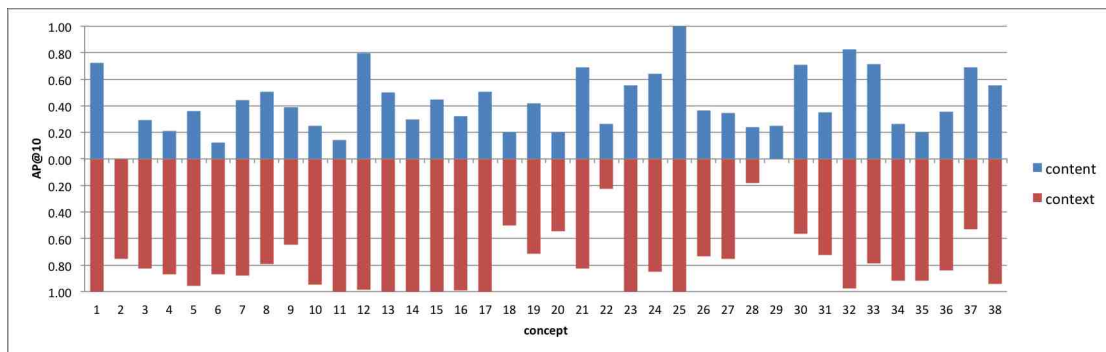
Figure 4.7: AP@10 of content and context modalities on the 38 concepts

The parameters that need to be tuned in the SLI model are β and λ as shown in Equation (4.6), which are the parameters of the ℓ_2-norm and the ℓ_1-norm. The grid search approach is applied that finds $\beta = 1.0$ and $\lambda = 0.01$, which can produce stable and relatively good results on the training data set after three-fold cross-validation. Figure 4.8 shows the MAP of the 38 concepts. EF_LR represents the early fusion approach, which directly concatenates the content and context feature representations, and adopts logistic regression as the classifier. LF_LR represents the late fusion approach, which first adopts logistic regression for local decisions, and then applies it again to fuse local

Table 4.2: Comparison results of MAP on MIRFLICKR-25000

	MAP@10	MAP@20	MAP@50	MAP@100	MAP@all
content	0.557	0.647	0.736	0.782	0.474
context	0.791	0.753	0.721	0.695	0.398
EF_LR	0.757	0.758	0.752	0.735	0.457
LF_LR	0.816	0.803	0.777	0.784	0.476
SLI	0.892	0.856	0.836	0.815	0.497

decisions from different modalities. SLI denotes the proposed sparse linear integration method. As can be seen from the figure, the results of "content" have relatively low precision values in the top retrieved images, but the precision values increase as more images are included till top 100. The results of "context" show an opposite behavior, which achieves high precision values in the top retrieved images but the results decrease when more images are included. These values are also shown in Table 4.2 in a more clear way. In this table, it can be seen that EF_LR achieves stable results than methods using each of the modality. However, on MAP@10 "context" performs slightly better than EF_LR, and on MAP@100 and MAP@all, "content" gives a much better performance. Look more closely at Figure 4.8, we can see that EF_LR actually produces an "averaged" results between "content" and "context". LF_LR outperforms those methods using each of the modalities. From Figure 4.8, we can see LF_LR also achieves better results compared to EF_LR, and does not suffer from the "averaged" problem. On the other hand, SLI achieves the best performance on all metrics, and the relevant improvements on MAP@10, MAP@20, MAP@50, MAP@100 and MAP@all compared to EF_LR are 17.8%, 13.0%, 11.2%, 10.8%, and 8.9%, respectively. The corresponding improvements compared to "LF_LR" are 9.3%, 6.1%, 7.6%, 3.9%, and 4.5%.

A similar work [2] is discussed in Chapter 2, which uses matrix factorization (MF)
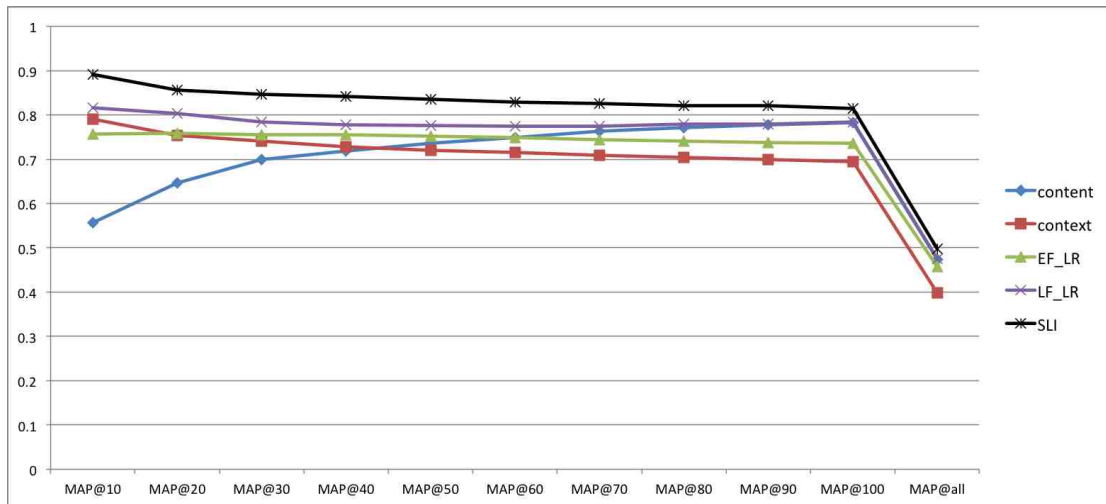
Figure 4.8: Comparison results of MAP on MIRFLICKR-25000

to integrate the content and context feature representations. Compared to the matrix factorization technique, the SLI model does not need to decide the latent factor, which could cause information loss due to the low-rank approximation. In addition, the sparsity also keeps a low computation complexity. Experiments conducted in [2] used the same training and test data sets, but a different set of features was extracted. For the content modality, a dictionary of 2000 visual features is used while for the context modality, 1391 keywords are used by keeping those keywords that appear more than 20 times, and idf weights are used instead of the binary values. However, given a much larger feature space, their reported results as shown in Table 4.3 are quite low, though no information about the classification method they used. MF-visual denotes the backprojected content feature representation, MF-textual denotes the backprojected context feature representation, and MF-latent is the weighted concatenation of the two backprojected representations. The performance of two other methods in [2] are also reported, The visual search method uses the original visual representation, and the semantic embedding method finds a semantic transformation from textual features to visual features.

Table 4.3: MAP of the 38 concepts reported in work [2]

	P@10	MAP@all
MF-visual	0.426	0.315
MF-textual	0.374	0.287
MF-latent	0.440	0.288
visual search	0.526	0.309
semantic embedding	0.258	0.227

The metrics adopted in their experiments are MAP@all, and P@10 which is simply the precision value of the first 10 results. Thus, we also calculate the P@10 values of "content", "context", EF_LR, LF_LR ,and SLI, which are are 0.392, 0.695 , 0.724, 0.807 and 0.835. The P@10 values of "content" are smaller than those of "MF-visual" and "visual search", which are all generated from the content modality only. This is probably due to the common visual features we used, but our visual feature dimension is only about a quarter of theirs. The MAP@all value of "content" outperforms all their methods. On the context side, the 500 terms we extracted are much effective compared to the 1391 terms used in their experiments. Using the context modality alone in our framework outperforms their methods in terms of both P@10 and MAP@all.

The same evaluation is also done on NUS-WIDE-LITE, which contains $55,615$ images with associated tags from the Flickr website. This data set has also been divided by the data set provider into training and test data sets in advance, where $27,807$ images are used as the training data set and the test data set is composed of the remaining $27,808$ images. Some low-level features are provided which include color histogram, wavelet texture, and etc. The low-level features used here in this experiment are 64-dimensional color histogram in LAB color space and 128-dimensional wavelet texture, which are basic features that are commonly extracted to analyze the content of images.

81 concepts provided by this data set are listed in Table 4.4. It also provides $1,000$ frequent tags which are used as the context modality, but they contain much less noisy tags compared to MIRFLICKR-25000. The PN ratios of the concepts are shown in Figure 4.9, It can be seen from this figure that most of the concepts are very imbalanced in that the number of positive images (images containing a target concept) divided by the number of negative images (images without a target concept) is smaller than 0.05. This is very challenging in the area of multimedia semantic information retrieval.
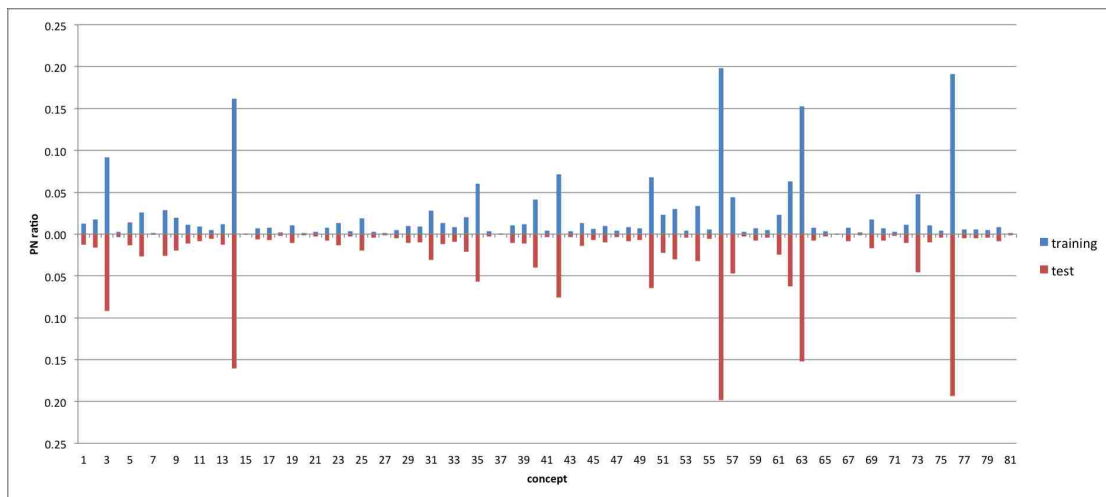
Figure 4.9: The PN ratio of all 81 concepts from NUS-WIDE-LITE

The results of AP@10 using logistic regression based on the content and context feature representations are displayed in Figure 4.10. As can be seen, for this data set, the context modality generates much better performance, which is due to the fact that the tags provided by this data set are already been cleaned and thus the quality is high. On the other hand, the performance generated from the content feature representation is considerably inferior compared to that of the context feature representation. However, given the high quality of the context modality, there still exist concepts that visual fea-

Table 4.4: Names of the 81 concepts from NUS-WIDE-LITE

1	airport	28	frost	55	sign
2	animal	29	garden	56	sky
3	beach	30	glacier	57	snow
4	bear	31	grass	58	soccer
5	birds	32	harbor	59	sports
6	boats	33	horses	60	statue
7	book	34	house	61	street
8	bridge	35	lake	62	sun
9	buildings	36	leaf	63	sunset
10	cars	37	map	64	surf
11	castle	38	military	65	swimmers
12	cat	39	moon	66	tattoo
13	cityscape	40	mountain	67	temple
14	clouds	41	nighttime	68	tiger
15	computer	42	ocean	69	tower
16	coral	43	person	70	town
17	cow	44	plane	71	toy
18	dancing	45	plants	72	train
19	dog	46	police	73	tree
20	earthquake	47	protest	74	valley
21	elk	48	railroad	75	vehicle
22	fire	49	rainbow	76	water
23	fish	50	reflection	77	waterfall
24	flags	51	road	78	wedding
25	flowers	52	rocks	79	whales
26	food	53	running	80	window
27	fox	54	sand	81	zebra

tures are very useful, such as concept No.41 "nighttime". This finding is also intuitive since color-based visual features are expected to play an important role in discriminating this concept. From this data set, the same conclusion is drawn that the retrieval performance can be greatly enhanced if the two modalities are properly integrated.
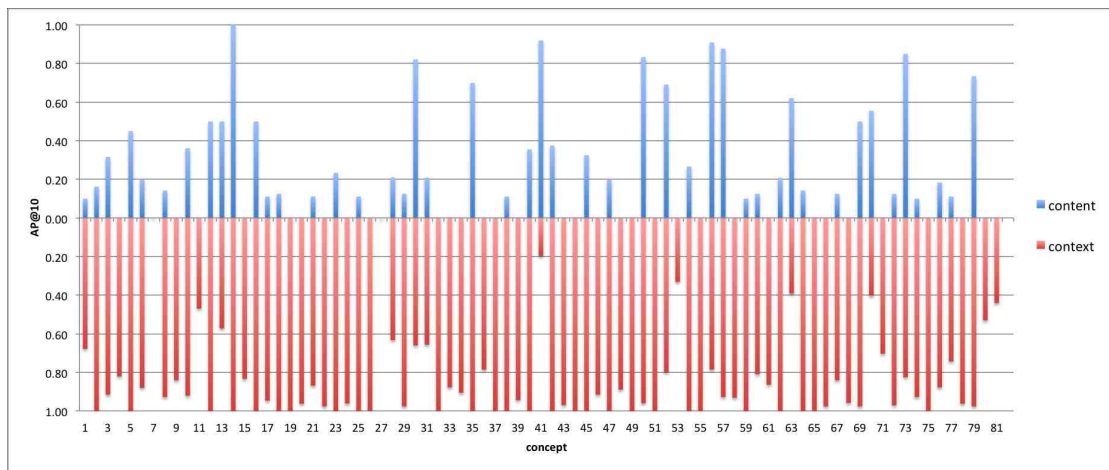
Figure 4.10: AP@10 of content and context modalities on the 81 concepts

Similar to Figure 4.8, Figure 4.11 shows the comparison results of the MAP values. Thus, EF_LR represents the early fusion in combination with logistic regression, and LF_LR represents the late fusion in combination with logistic regression. The fact that the MAP values of the EF_LR are generally smaller than those of "context" is attribute to the low performance of "content". Thus, EF_LR is subject to the modality having the worst performance, though from Table 4.5 we can see that EF_LR is able to achieve slightly better results than "context" at MAP@100 and MAP@all. The performance of LF_LR is much robust compare to that of EF_LR, which constantly generates better MAP values than those using each of the modality alone. SLI still achieves the best results on this data set, and the relative improvements on MAP@10, MAP@20,

Table 4.5: Comparison results of MAP on NUS-WIDE-LITE

	MAP@10	MAP@20	MAP@50	MAP@100	MAP@all
content	0.311	0.307	0.269	0.223	0.067
context	0.835	0.805	0.724	0.662	0.294
EF_LR	0.773	0.749	0.719	0.672	0.298
LF_LR	0.886	0.842	0.769	0.722	0.316
SLI	0.926	0.902	0.810	0.748	0.332

MAP@50, MAP@100 and MAP@all comparing to EF_LR are 19.8%, 20.4%, 12.7%, 9.7% and 11.5%. The corresponding improvements compared to LF_LR are 4.5%, 7.1%, 5.3%, 3.6% and 5.1%, which show a stable improvement of around 4%.
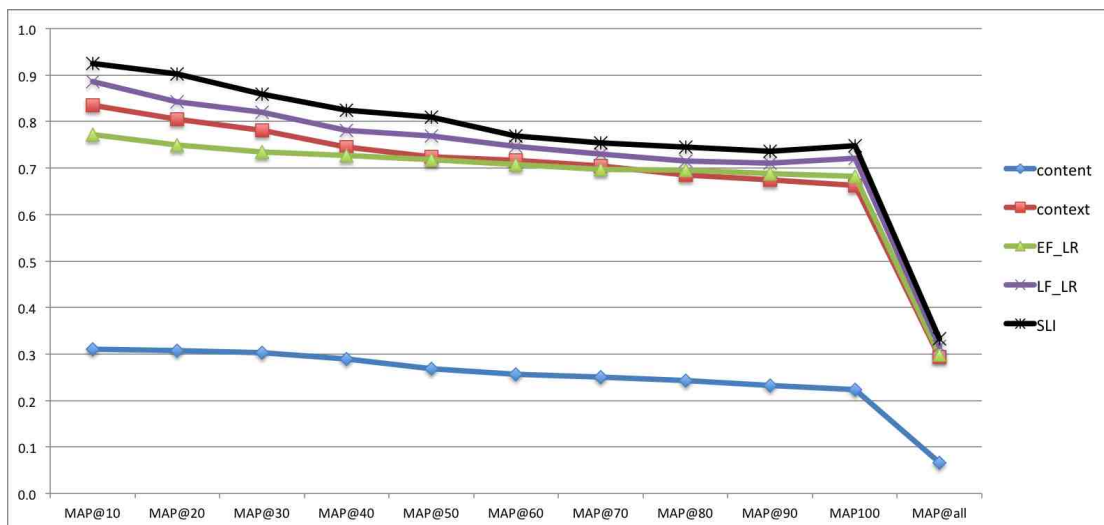
Figure 4.11: Comparison results of MAP on NUS-WIDE-LITE

SLI is evaluated against several popular fusion approaches, including methods using "minimum" (min), "maximum" (max), "median", and "average" rules. Here, majority voting is not included since it requires hard decision on class labels. The super kernel fusion (SKF) method [118] is also compared, as well as one of our previously work [58],

which considers adjustment, reliability, and correlation of the intervals to the target concept (denoted as ARC). The local decisions of these methods are abstained using SVM [119]. The experiment setup is based on the experiment conducted in [58], which treated the two visual features as two modalities, and the textual features as the third modality. So the general form of SLI is used, where the number of modalities $\Omega = 3$. The comparative MAP@all values on the NUS-WIDE-LITE data set are shown in Figure 4.12. It can be observed from this figure that SLI outperforms the comparative approaches by more than 2%-20%. Median fusion gives the worst performance and is outperformed by SLI with a large margin of 23%. ARC produces the second best result but is still 1.0% lower than SLI. The min fusion method shows fairly good results and is better than the average and max fusion methods.
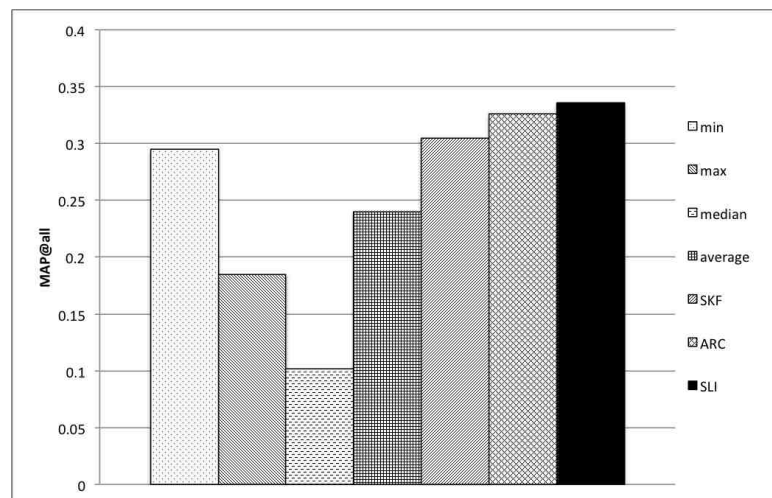
Figure 4.12: Comparison results of MAP on NUS-WIDE-LITE

4.3.2 Evaluation of the Generalized SLI Model

The benchmark data set from TRECVID 2011 Semantic Indexing(SIN) [120] is used to evaluate the generalized sparse linear integration model, which aims to identify the correct semantic concept contained in a video shot. 50 concepts are evaluated by TRECVID 2011 SIN task, their concept IDs and names are shown in Table 4.6. There are 9322 videos in the training data set and 8182 videos in the test data set. Each video is divided into different numbers of shots with shot boundary information provided by TRECVID. For each shot, one key frame is extracted to represent the visual content of the shot. So totally there are 262911 shot-level instances in the training data set and 137327 shot-level instances in the test data set. The number of video-level instances in the training and test data set is equal to the number of videos in these sets. 5 types of visual features are extracted from each key frame, including color histogram in the HSV space, color moment in the YCbCr space [116], Color and Edge Directivity Descriptor (CEDD) [30], Histogram of Oriented Gradients (HOG) [37], and haar wavelets [117]. Histogram equalization is applied to adjust the contrast of frames before extracting the features. The total dimension of visual features is 707. To build textual features, "title", "description", and "subject" are extracted from video metadata. As mentioned before, preprocessing including stop word removal, English word validation, and stemming is performed. Totally, 11083 unique terms are extracted. The MCA-based feature selection method is applied to this binary textual features for each concept and the most discriminative 500 terms are kept.

Figures 4.13 and 4.14 show the ratios of the positive instances to the negative instances (PN ratio) of the 50 concepts in the video-level and shot-level, respectively. Shot-level ground truth is provided. For video-level ground truth, we adopt the same

strategy as multiple instance learning, where a video is considered as positive if at least one shot in it is positive, and a video is considered as negative if all the shots in it are negative. As can been seen from these two figures, the PN ratios of video-level instances are much higher than the PN ratios of the shot-level instances, though the imbalance issue exists in both figures.
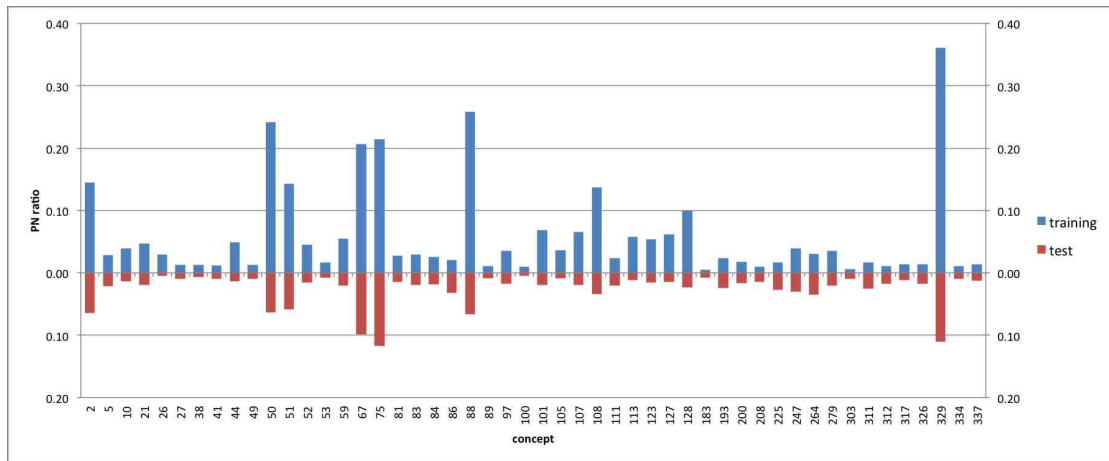
Figure 4.13: The video-level PN ratios of the 50 evaluated concepts
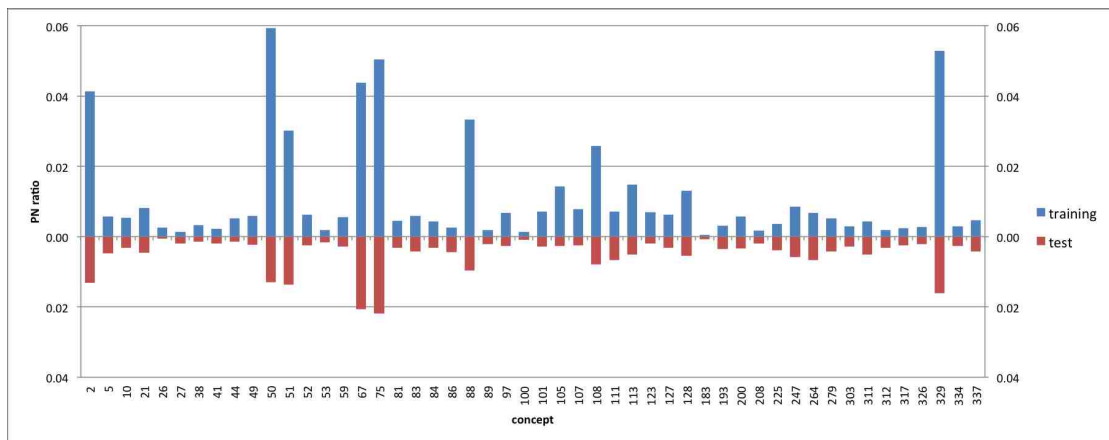
Figure 4.14: The shot-level PN ratios of the 50 evaluated concepts

The comparison methods are similar as before. The first one using the content modality alone and is denoted as "content". The second one uses the context modality

Table 4.6: Names of the 50 concepts

2	Adult	101	Scene_Text
5	Anchorperson	105	Singing
10	Beach	107	Sitting_Down
21	Car	108	Sky
26	Charts	111	Sports
27	Cheering	113	Streets
38	Dancing	123	Two_People
41	Demonstration_Or_Protest	127	Walking
44	Doorway	128	Walking_Running
49	Explosion_Fire	183	Door_Opening
50	Face	193	Event
51	Female_Person	200	Female_Human_Face
52	Female-Human-Face-Closeup	208	Flags
53	Flowers	225	Head_And_Shoulder
59	Hand	247	Male_Human_Face
67	Indoor	264	News
75	Male_Person	279	Quadruped
81	Mountain	303	Skating
83	News_Studio	311	Speaking
84	Nighttime	312	Speaking_To_Camera
86	Old_People	317	Studio_With_Anchorperson
88	Overlaid_Text	326	Table
89	People_Marching	329	Text
97	Reporters	334	Traffic
100	Running	337	Urban_Scenes

alone and is denoted as "context". Please note the scores from this one are video-level retrieval results, no shot-level result is available for this method. EF_LR is the method that uses early fusion to concatenate the replicated video-level features with the shot-level features and logistic regression as the classifier. LF_LR is the method that uses logistic regression to fuse the scores generated from these two modalities, which adopts the same classifier to make the local decisions. Again, instead of replicating video-level features, video-level scores are replicated in order to match the shot-level score.

The performance of both "content" and "context" methods on all 50 concepts are displayed in Figure 4.15. The complementary characteristic of the content and context modalities can also be observed on this data set. For many concepts, "context" outperforms "content" by a large margin, such as "Adult", "Car", "Hand", "Male_Person", "Overlaid_Text", "Sitting_Down", and "Male_Human_Face". There are also concepts that "content" performs relatively good while "context" almost completely fails. These concepts are "Cheering", "Explosion_Fire", "Flowers", "Mountain", "running", "Walking", "Walking_Running", "Head_And_Shoulder" and "News". Therefore, by integrating the scores from these two modalities, we are expecting to see the improvement on shot-level retrieval. Table 4.7 shows the performance of these four methods on these 50 concepts using MAP@10, MAP@100, MAP@1000, MAP@2000, and MAP@all. Since "context" retrieves at the video level, its results are not included in this table. As can be seen, LF_LR achieves higher MAP@100, MAP@1000, and MAP@2000 values compared to those in both "content" and EF_LR, but the improvement of MAP@all is small. Noticeably, its result of MAP@10 is much smaller than "content". This is because too much noise is brought in when duplicating video-level scores. As EF_LR, SLI also replicates video-level features. However, instead of directly concatenating them

Table 4.7: MAP of the 50 evaluated concepts

	MAP@10	MAP@100	MAP@1000	MAP@2000	MAP@all
content	0.507	0.380	0.264	0.230	0.117
EF_LR	0.313	0.305	0.215	0.182	0.106
LF_LR	0.356	0.465	0.314	0.266	0.138
SLI	**0.708**	**0.526**	**0.351**	**0.292**	**0.145**

with the shot-level features and feeding into a classifier, SLI weights two modalities for each instance, which produces a more meaningful feature representation for classification. If the metadata of a video is noisy or unreliable, its corresponding weight would be low, which makes the textual features have less effect in the later process, and vice versa. SLI outperforms the second best one in MAP@10, MAP@100, MAP@1000, MAP@2000, and MAP@all with relative improvements of 39.4%, 13.1%, 2.9%, 9.8% and 5.1%. The improvement on MAP@10 is considerably big, which is important in practical since users care more about the accuracy of the top retrieved results.
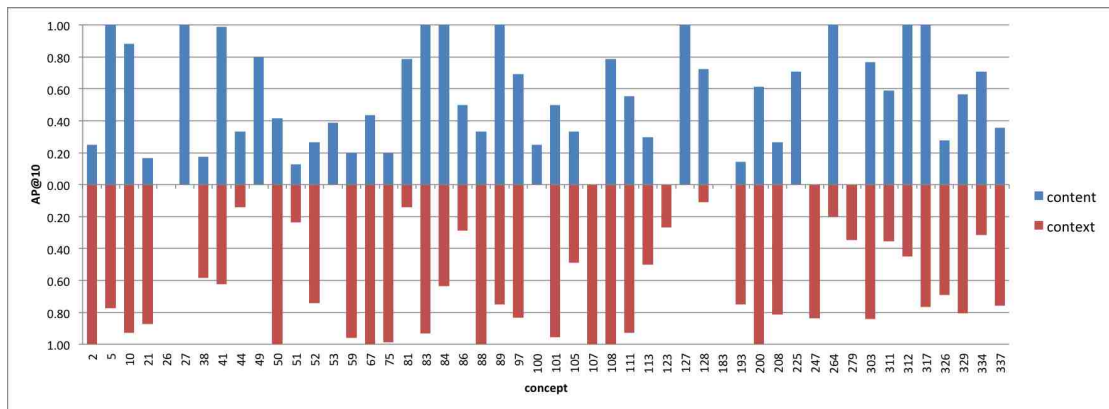
Figure 4.15: AP@10 of content and context modalities on the 50 evaluated concepts

In all three of these experiments, we evaluate SLI against early fusion and late fusion using logistic regression. Compare to early fusion, a weight is added to each

modality when concatenating their feature representations. Compare to late fusion, feature correlation between different modalities are considered instead of being treated independently. Compare to logistic regression, ℓ_1-norm is included in addition to ℓ_2-norm. When tuning the parameters β and γ, which searches β from 0.001 to 10.0 and γ from 0.001 to 0.1, we notice the performance is greatly affected by γ while not affected too much by β within their search ranges. A larger γ introduces more sparsity to the solution, which can remove noise to some extent and improve model performance. However, after certain point, the performance starts to drop quickly and eventually drop to 0. It's because if γ is too large, which means a high sparsity, then an instance is represented by a very limited number of instances, or even no instance at all. Therefore, γ needs to be tuned more carefully in order to achieve a better performance.

4.4 Conclusions

This chapter presents the overall framework of the sparse linear integration component. The model is formulated into an optimization problem, and efficient solutions are provided to solve the problem. As mentioned before, the learning process of the model can be decoupled by instance, so the calculation of the similarity coefficients of each instance can be done independently. Therefore, this component can be parallelized by using a processor to handle a set of instances and aggregate the results to form the complete learned model. The instance pairwise similarity can be directly used for unsupervised applications. A classifier based on the reconstruction error is designed to customize the model for supervised learning. Two benchmark image data sets are used to evaluate the model on integrating visual features and tags for semantic concept retrieval. Comparison results from popular approaches and the state-of-the-art methods demonstrate the effectiveness of the model.

To cope with situations when the instances from different modalities are not in the same granularity, a generalized SLI model is provided by utilizing instance associations between different modalities. A benchmark video data set is also used to evaluation the generalized model. Results once again confirm that not only integration of content and context modalities improve the retrieval results, but also the sparse linear integration model outperforms the comparison methods.

Chapter 5

Applications in Recommender Systems

In this chapter two applications in Recommender Systems are presented that also integrate content and context modalities to improve recommendation accuracy. The first one is described in Section 5.1, which utilizes the sparse linear integration as illustrated in Chapter 3 to perform integration. The second one is described in Section 5.2, which relies on a topic model to represent both features from both content and context modality in an unified framework.

5.1 Multimodal Sparse Linear Integration for Content-based Recommendation

Sparse LInear Methods (SLIM) for top-n recommendation are first introduced in [101], which generates recommendation results by aggregating user purchase or rating profiles. A sparse aggregation coefficient matrix is learned by solving a ℓ_1-norm and ℓ_2-norm regularized optimization problem. The final recommendation is the linear combination of the original user profiles weighted by the learned sparse aggregation coefficients. Later the authors extended SLIM to incorporate item content information [102], but the basic model is the same. The extended method is called SSLIM, which is short

for SLIM with item Side information. Experiments on various data sets demonstrate high quality recommendations, and the sparsity of the coefficient matrix allows SLIM to generate recommendations very fast.

In this section, the sparse linear integration model [121] is applied to facilitate content-based item recommendation. Compared to SLIM and SSLIM, our method is more generic and can be used in information retrieval task. The focus in this section is to utilize multiple modalities of item content to handle recommendation scenarios when user profiles are not available. Therefore, rather than using the learned coefficients to linearly combine user profiles as in SLIM or user profiles with item side information as in SSLIM, we directly use the learned similarity matrix of items to generate the recommendations. Because each entry in the similarity matrix is the similarity between two items. The model learns sparse similarity between items based on features extracted from multiple modalities. We name the method Multimodal Sparse Linear Integration Method (MSLIM), which is the generic form of the sparse linear integration model as presented in Section 4.2.1. A comparison with rule-based late fusion approach is conducted to evaluate MSLIM.

MSLIM learns similarity between items in an unsupervised manner by integrating multiple modalities. A framework adopted MSLIM is proposed accordingly which learns item pairwise similarities based on textual information from item description and visual information from images, and then applies k-nearest neighbor (k-NN) for item recommendation. The rest of section is organized as follows: The detailed problem formalization and solution are presented in Section 5.1.1 followed by the experimental results in Section5.1.2. Conclusions are drawn in Section 5.1.3.

5.1.1 The Framework of MSLIM

Based on the work in [101] [102], an effective and generic method MSLIM is proposed to integrate multimodal information for content-based top-n recommendation. It aims to learn a sparse similarity matrix from multiple modalities in an unsupervised manner. The problem is formulated into a regularized optimization problem in the least-squares sense and a framework of integrating textual and image visual information for item recommendation is proposed accordingly.

Problem Formalization

Assuming there are Ω modalities, and each modality is represented by a feature-item matrix $\boldsymbol{X}^\omega$ and $\omega \in [1, \Omega]$, where each row is a feature or an attribute and each column is an item. If there are totally N items and M^ω features/ attributes for the ω-th modality, then the dimension of $\boldsymbol{X}^\omega$ is $A^\omega * N$. Let A denote the dimension of matrix including all the modalities which is equal to $\sum_{\omega=1}^{\omega=\Omega} A^\omega$. An entry in $\boldsymbol{X}^\omega$ is denoted as x_{ij}^ω which could be a nominal or a numeric value for the i-th feature of the j-th item from the p-th modality. The j-th column of $\boldsymbol{X}^\omega$ is denoted as $\boldsymbol{s}_j^\omega$ while the i-th row is denoted as $(\boldsymbol{s}_i^\omega)^T$.

Similar to Chapter 4, the problem is formulated into an optimization problem as presented in Equation (5.1). Another constraint $\boldsymbol{S} \geq 0$ is added to ensure the learned pairwise coefficients are non-negative. The non-negative property allows the result to be easily inspected and interpreted. As discussed in Chapter 4, Equation (5.1) can be decoupled by columns since each column of $\boldsymbol{S}$ is independent from each other, and co-ordinate descent [108] and soft-thresholding are applied to solve the optimization problem. The aggregation coefficients of items calculated by integrating multiple modalities

are represented by the $N \times N$ matrix $\boldsymbol{S}$. For each item, its neighbors are defined as items having large coefficients with this item, and thus k-NN can be adopted as the recommendation algorithm. In other words, content-based recommendation is achieved by obtaining similar items from multiple modalities.

$$\min_{\boldsymbol{S}} \quad \sum_{\omega=1}^{\omega=\Omega} \frac{\alpha^{\omega}}{2} \|\boldsymbol{X}^{\omega} - \boldsymbol{X}^{\omega}\boldsymbol{S}\|_F^2$$
$$+ \frac{\beta}{2}\|\boldsymbol{S}\|_F^2 + \gamma\|\boldsymbol{S}\|_1$$

$$s.t. \quad \boldsymbol{S} \geq 0, \tag{5.1}$$

$$diag(\boldsymbol{S}) = 0$$

The proposed MSLIM can incorporate user profiles if available, which is equivalent to the method in [102] if considering all the content information of items as side information. If using $\boldsymbol{U}$ to denote users' implicit or explicit feedbacks, then there is an addition term in Equation 5.1, which is $\frac{\mu}{2}\|\boldsymbol{U} - \boldsymbol{U}\boldsymbol{S}\|_F^2$. The same solution applies.

A System Framework

A framework utilizing MSLIM for item recommendation is presented. One modality is the textual information of items, and the other modality is the visual information of items. In this framework, they are item descriptions and images respectively. In Equation (5.1), for clear distinction and consistence, let $\boldsymbol{X}^t$ denote the textual feature-item matrix, and let $\boldsymbol{X}^v$ denote the visual feature-item matrix.

Visual features of images includes color, texture, shape etc, such as HSV color histogram, histogram of oriented gradients (HOG) and popular scale-invariant feature transform (SIFT) [32]. The aim of this section is not to compare various features, but to

validate the effectiveness of MSLIM that can improve content-based recommendation by integrating features from difference modalities. Hence, we only extract one visual feature which is CEDD [30] feature due to its good balance between accuracy and complexity. It is a compact composite descriptor that incorporates both color and texture information in one histogram. The CEDD extraction system is composed of two units, texture unit and color unit, and three fuzzy systems. First, the image is separated into a preset number of blocks (usually 1600 for compromising between the image detail and the computational complexity). Then each of the blocks passes through all the units as follows. In the texture unit, each image block is classified into one of the 6 texture categories by applying with 5 digital filters. In the color unit, the image block is converted into the HSV color space and fed into two fuzzy systems with a set of rules, obtaining a 24-bins histogram (with each bin representing one color). Finally, the overall histogram contains $6 \times 24 = 144$ regions.

For textual features, we extract keywords/terms from descriptions of items, and use binary value to represent the presence of a feature. Take feature "leather" for example, 1 means "leather" exists in the item's descriptions while 0 means the opposite. For descriptions in English, standard procedures such as stop word removal and stemming are usually applied to preprocess the terms. WordNet [112] can also be used to validate English words due to ubiquitous typos, especially in user-contributed social media data such as image tags from Flickr. The descriptions we collected for our bag data set are in Chinese, and more details are given in Section 5.1.2. There are totally 509 binary features extracted, which cover materials, brands, colors, styles, structure and etc. Normalization is performed on extracted features to convert their scales and to ensure that they are suitable for general data analysis. For visual features, min-max normalization

is adopted to scale the feature values between 0 to 1. For textual features, we do not apply any normalization since the extracted features are binary.

Features extracted from each modality are fed into the sparse linear integration module and generate the aggregation coefficients of items. Then k-nearest neighbor (k-NN) is used to find the neighbors of each item based on the aggregation coefficients. Recommendation results are generated by sorting the similarity scores of neighbors in descending order. A framework summarized these procedures is presented in Figure 5.1. Textual features and visual features are extracted from descriptions and images of items respectively. If there are other modalities, such as descriptions from other websites or images from a different view point, then features can be extracted accordingly and fed into the sparse linear integration module.
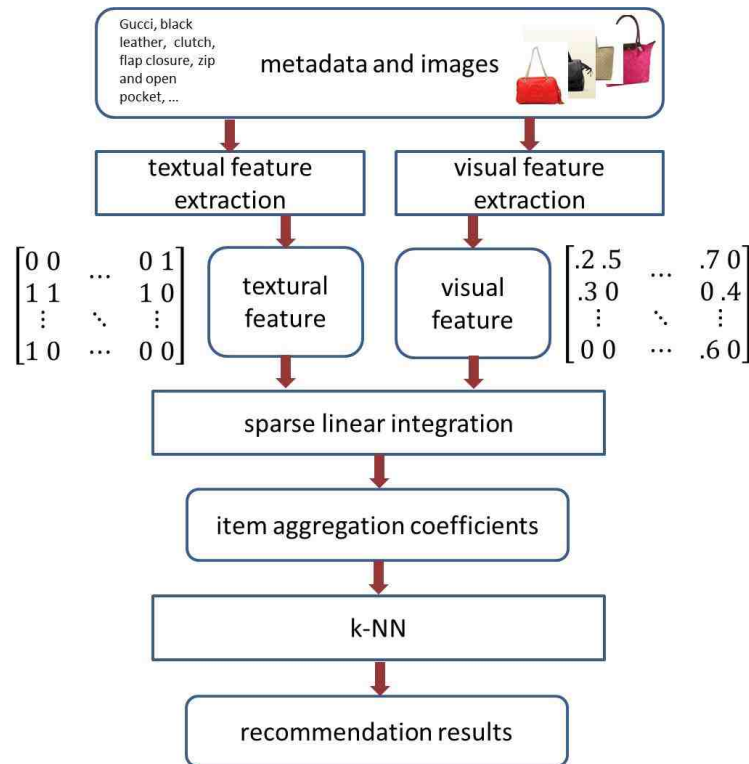
Figure 5.1: The MSLIM framework for content-based multimedia recommendation

5.1.2 Experimental Results

To evaluate MSLIM for content-based recommendation, images and descriptions of handbags are collected to perform handbag recommendation, and user rating data are collected as ground truth for judgement. We first investigate how parameters affect the recommendation results, and then use the best parameter settings to conduct further comparison. The comparison includes methods using a single modality, as well as the method that linearly combines the results from each single modality. To ensure fair comparison, k-NN is used as the recommendation algorithm for all methods.

Data Collection

The data set of handbags is collected based on the bags appeared in a fashion show video which is created by gluing parts from several videos. Specifically, the first sequence (0-27 sec) is the Gucci fall-winter 2010 women's wear show; the second sequence (28-58 sec) is from some advertising videos downloaded from the Gucci web site; the rest of the video (59-end) is the Prada fall-winter 2012 women's wear show. Google image search based on keywords such as "prada fall winter 2012 women's wear show handbag" is used to find the exact bags appeared in the video. Next Google image search using the images from the keyword search results is utilized to find visually similar bags which form the recommendation data set. There are totally 440 bags, and both the images and their descriptions are collected as two modalities. Since the descriptions of bags are relatively structured, that is they are described in similar way such as materials, brands, color, styles etc., thus textual features are extracted based on this structure.

To collect the ground truth, we design a web-based interface for users to provide ratings. Each bag is used as a target item which means the bag is the one the user is

interested in or wants to purchase. 20 other bags are recommended to the user for him or her to rate from 0 to 5. 0 means the user is not interested in the recommended bag while 5 means the recommended bag is very similar to the target bag. There are actually two parts in this user judgement process. The first part is using visual information alone and presenting the images of the top 20 recommended bags to the users, as showed in Figure 5.2. The second part is adding textual information and both the important descriptions and the images of the top 20 recommended bags are presented to the users, as showed in Figure 5.3. In both web interfaces, the target bag is the first one in each row which is highlighted in yellow box. Only the first top 10 recommended bags can be seen from the figures due to the size of the window, but there are actually 20 bags in each row. The reason we design it in two parts is to avoid bias when judging using visual or textual information alone. 11 users participate this judging task, and ratings from both parts are collected. For each target bag, its recommended bags with an average rating equal to or above 3.0 is considered as a relevant recommendation which indicates the interest from users.
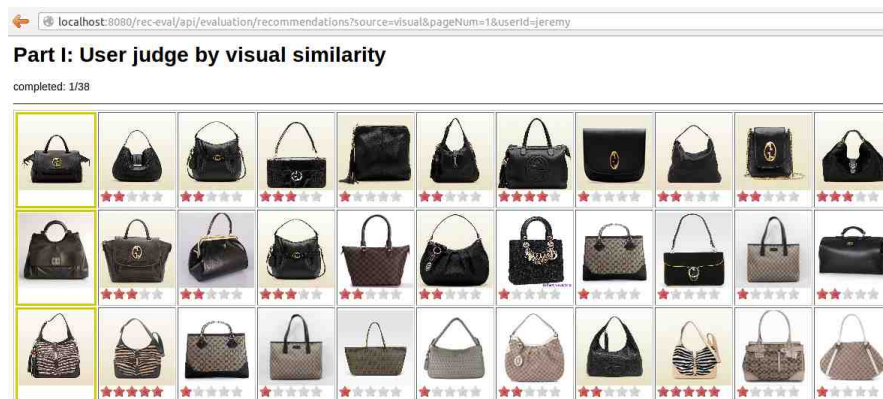
Figure 5.2: The interface of collecting ratings for bags using visual information

Figure 5.3: The interface of collecting ratings for bags using visual and textual information

Evaluation Metrics

For evaluation metrics, precision is the most commonly used metric in the top-n recommendation. It measures the percentage of correctly predicted items, denoted as prec@n, and can be calculated based on Equation (5.2). TP is the number of relevant items or true positive (TP) in the n recommended videos, and n is usually set to 5 or 10.

$$prec@n = \frac{TP}{n} \tag{5.2}$$

Besides AUC and MAP as mentioned in the experiments of the previous chapters, normalized discounted cumulative gain (NDCG) is used in this section. NDCG measures the gain of an item based on its position in the result list according to Equa-

tion (5.3). rel_t is the graded relevance of an item at position t, and IDCG is the ideal DCG which is the maximum possible value of DCG.

$$NDCG \quad = \frac{DCG}{IDCG}$$

(5.3)

$$where \quad DCG \quad = rel_1 + \sum_{t=1}^{t=n} \frac{rel_t}{\log_2(t)}$$

Compared Methods

The proposed MSLIM is first compared with methods using the same k-NN recommendation algorithm but a single modality in order to prove that the integration of multiple modalities helps the final recommendation. The method using textual information only is denoted as textual method (TM) while visual method (VM) denotes the method using visual information only. However, to further evaluate the improvement, a rule-based late fusion method is adopted as a comparison method. It uses Equation (5.8) to linearly weight and combine the recommendation scores from each modality, where w^p is the weight of modality S^p and $f(\cdot)$ is a recommendation algorithm which takes feature-item matrix as input and outputs the recommendation scores. As mentioned before, $P = 2$ since there are two modalities in our data set. This method is denoted as LWM, which stands for linear weighted method.

$$\sum_{\omega=1}^{\omega=\Omega} \quad w^\omega \times f(\boldsymbol{X}^\omega)$$

(5.4)

$$where \quad \sum_{\omega=1}^{\omega=\Omega} w^\omega = 1$$

Experimental 1: Parameter Tuning

The parameters involved in MSLIM are α^t, α^v, β and γ as shown in Equation (5.1) where $\omega = 2$. There are no parameter in TM and VM, and for LWM, the parameters are the weight of textual modality w^t and the weights of visual modality w^v.

Let's start from LWM first. We decrease the value of w^t from 1 to 0 with step of 0.1, and the value of w^v is increased from 0 to 1 with step of 0.1 accordingly. Figure 5.4 presents the performance of LWM using the aforementioned 7 metrics with the weight of visual modality w^v increasing from 0 to 1. As can be seen, AUC reaches the highest when $w^v = 0.2$, and its value at 0.1 is relatively high. While for the rest metrics, their values slightly increase or stay the same when w^v increase from 0 to 0.1, and drop dramatically when w^v continues increasing from 0.1 to 0.2. Therefore, we choose w^2 to be 0.1 by considering the performance on all the metrics. The value of w^t is set to 0.9 to ensure their summation is equal to 1. These parameters indicate that the information from textual modality is more reliable or accurate since w^t is much larger than w^v.

For MSLIM, we first tune the parameter of α^t and α^v with fixed β and γ since α^t and α^v play a local role in integrating textual and visual modality. Therefore, we set $\beta = 1.0$ and $\gamma = 0.01$ empirically first, and fix α^t to 1.0 and only vary α^v. Figure 5.5 shows the performance of MSLIM with α^v set to $\{0.1, 0.5, 1.0, 2.0, 5.0\}$ and the other parameters are set to the fixed values. As shown in the figure, on average, $\alpha^v = 0.5$ gives the best overall performance when considering all the metrics.

The next step is fixing α^t and α^v to the optimal value we find which are 1.0 and 0.5 respectively, and then using grid search to find the optimal value of β and γ. The search range for β is from 0.001 to 10.0 with points at $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$, and the search points for γ are at $\{0.001, 0.005, 0.01, 0.05, 0.1\}$. Figure 5.6 and Fig-
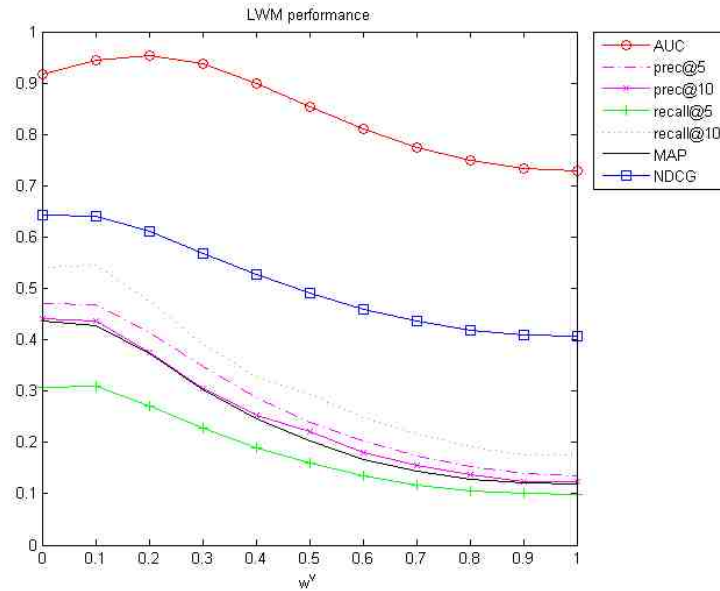
Figure 5.4: The performance of LWM varied by w^v

ure 5.7 display the performance using AUC and prec@5 from different views with β and γ as two variables. The performances varied by β and γ depict similar pattern on the rest metrics. As can be seen from both figures, there is a big decrease when γ keeps increasing after 0.01, and the performances drop to 0 after γ reaches beyond 0.05. It's because the ℓ_1-norm parameter γ controls the sparsity of the coefficient matrix. If γ is too large which means high sparsity, then there is no item can be recommended since the coefficients with the target item are all 0. For β, the performances increase from a relatively low value to the maximum when β increases from 0.001 to 1.0, and stays almost stable when β keeps increasing from 1.0 to 10. This indicates a small ℓ_2-norm regularization improves model performances but after a certain point, in this case when $\beta = 1.0$, it doesn't affect the performances anymore. From both figures, we can see the maximum performance forms a flat area bounded by $\gamma \in (0, 0.01]$ and $\beta \in [1.0, 10]$. Hence, we fix γ to its upper bound 0.01, and β to its lower bound 1.0 as the empiri-
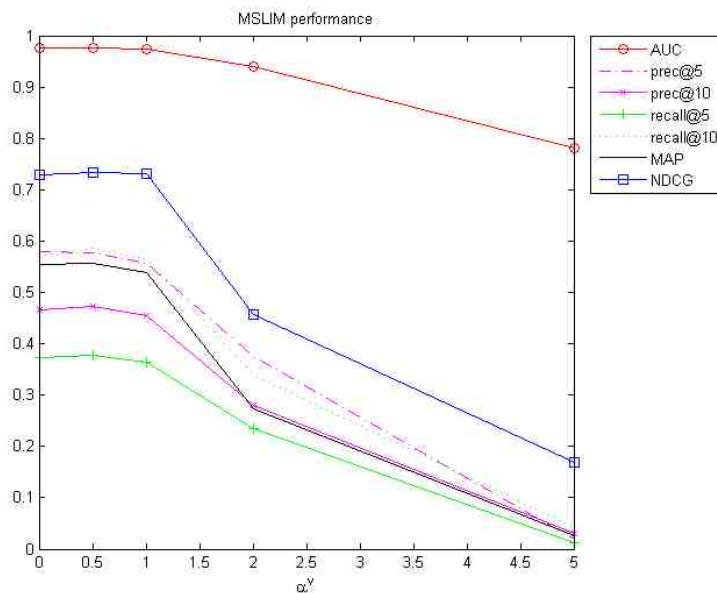
Figure 5.5: The performance of MLIMS varied by α^v

cal values we decide when tuning α^t and α^v. In fact, any value of γ and β within the aforementioned boundaries could assure the maximum model performances.

Experimental 2: Comparison Results

The comparison results of MSLIM against TM, VM, and LWM are shown in Figure 5.8. VM using visual information results inferior performance compared to the other three methods. One reason is that we only use one visual feature which is CEDD. It achieves relatively good performance compared to other visual features, but one single visual feature is very limited. If introducing more types of visual features, the performance of VM would be better. The other reason is that the semantic gap between low-level visual features and high-level semantic concepts. Take the brand of a bag for example, it's not easy to capture the pattern of a brand using visual features, but from the textual point of view, the exact words of a brand are probably already contained in the item descriptions.
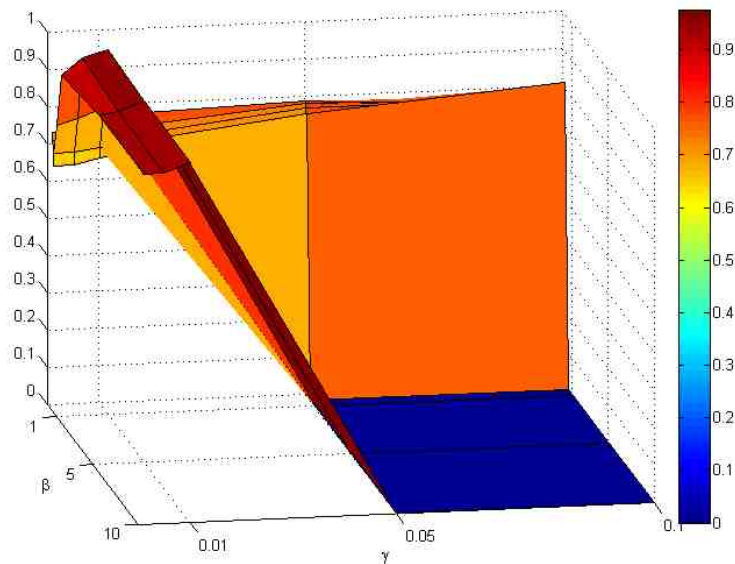
Figure 5.6: AUC of MSLIM varied by β and γ

MSLIM achieves the best results on all the metrics, followed by LWM and then TM. Its absolute improvements compared to TM, VM, LWM are summarized in Table 5.1. The last row shows the average increase over all the seven metrics. MSLIM outperforms VM by a large margin, which is 0.3556, so does TM and LWM, but with a slightly smaller margin. The results from TM and LWM are close, and are outperformed by MSLIM by about 0.072 on average.

5.1.3 Conclusions

In this section, a multimodal sparse linear integration method MSLIM is proposed for content-based item recommendation. It formulates the integration problem into a regularized optimization problem in the least-squares sense. Coordinate descent and soft thresholding are applied to solve the problem and parallel computing can be used to speed up the process. Aggregation coefficients of items are learned in an unsupervised
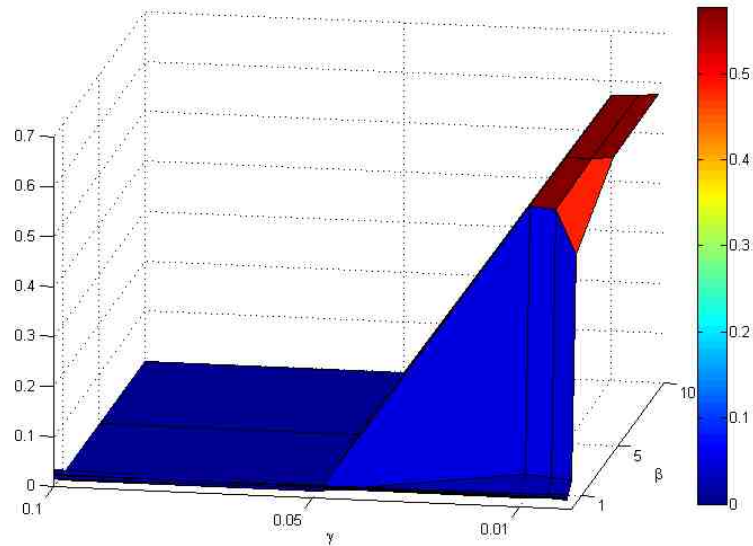
Figure 5.7: prec@5 of MSLIM varied by β and γ

manner during this process, based on which k-nearest neighbor (k-NN) is used to calculate the neighbors and generate the top-n recommendation results. Evaluation compares MSLIM with other three methods and proves its effectiveness on a handbag data set.

One limitation of MSLIM is that the number of features from different modalities should be in a similar scale, otherwise the aggregation coefficients learned would lean toward the modality with more features and thus contain more information from it. To solve this issue, one option is to apply feature selection technique to reduce feature dimensions and make sure the features from different modalities are in the same scale. Another limitation of our current work is that we learn the full $M \times M$ item coefficient matrix, which is tedious for top-n recommendation. Instead, we can only take the necessary neighbors into consideration.
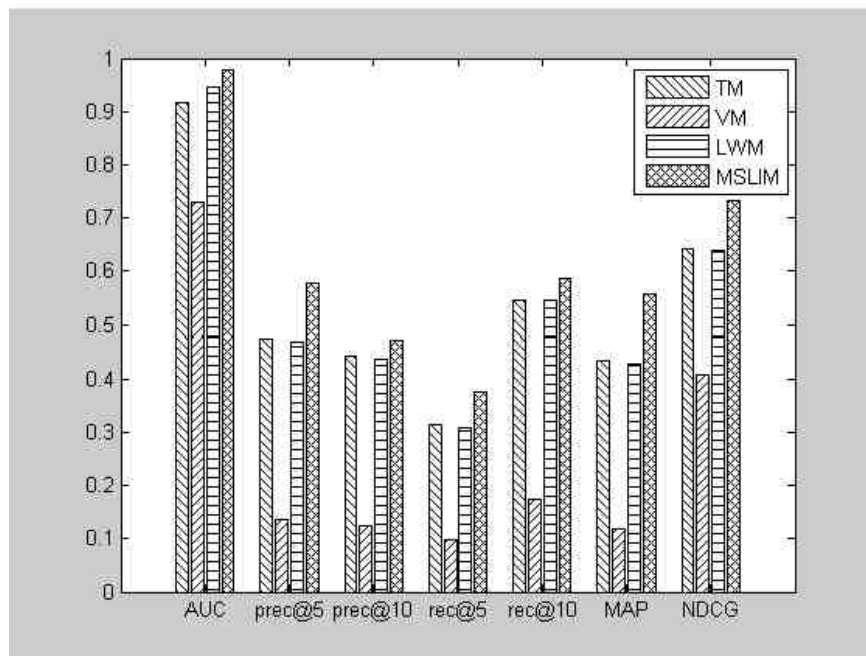
Figure 5.8: The comparison performance

5.2 Video Recommendation Using a Topic Model

Compared to the two content-based video recommendation methods [80][81] discussed in Chapter 2, we limit our content analysis to visual content and metadata. The audio information is not considered in our framework for two reasons. First, a large proportion of online videos have been edited, and background sound is added to the original videos but it seldom reflects the video content itself. Second, in VideoTopic, the "Bag of words" (BoW) model is used to analyze both textual and visual information in an integrated manner. However, the BoW model is not applicable to audio information which has strong temporal characteristics. Other models such as hidden Markov model (HMM) should be considered instead if audio information is consistent with the video content. An advantage of our framework is that a topic model is used to represent

Table 5.1: Improvements by MSLIM compared to TM, VM and LWM

Metric	TM	VM	LWM
AUC	0.0600	0.2475	0.0316
prec@5	0.0998	0.4403	0.1076
prec@10	0.0288	0.3494	0.0371
rec@5	0.0630	0.2777	0.0669
rec@10	0.0404	0.4112	0.0401
MAP	0.1229	0.4374	0.1283
NDCG	0.0911	0.3260	0.0932
avg	0.0723	0.3556	0.0721

the video content as well as user interests, which naturally links them and enables the representation of user interests using the watched videos.

In this section, a content-based video recommendation framework called VideoTopic [122] is proposed, which is particularly useful in the cold-start scenarios. First, both visual and textual features of videos are extracted. Using a topic model, each video is then represented as a mixture of a set of topics, and each topic is a mixture distribution of textual and visual words extracted from a video collection. User interests are then estimated based on users' previously watched videos, and can also be represented as a topic distribution over topics. A list of personalized videos is generated by finding videos with topic distributions as close as the topic distribution of user interests. The assumption is that recommending those videos most similar to user interests can maximize the accuracy. Hence, the contributions of this study lie in two folds:

- A novel content-based recommendation framework, VideoTopic, is proposed which uses a topic model to represent both visual and textual content of videos, and links user interests and video content by estimating user interests using the topic distributions of user watched videos.

- A new approach is proposed that maps the problem of recommending personalized videos to an optimization problem, which maximizes the recommendation accuracy by minimizing the topic distribution differences between user interests and the recommended videos.

This section is organized as follows: Section 5.2.1 presents the framework of VideoTopic and each of its components, followed by the experimental results in Section 5.2.2. Conclusions are drawn in Section 5.2.3.

5.2.1 The Framework of VideoTopic

The recommendation framework first represents the video content using a topic model from a user interest point of view, and then captures the interests from a user's behavior history. A personalized recommendation list is then generated to fit the user's interests. The recommendation framework is not limited to recommending videos. It can be applied to general items, even if only visual or textual information is available. The whole process is performed by two key components in the framework which are video representation and recommendation generation.

Video Topic Model

The "Bag of words" (BoW) model is a very popular model used in information retrieval (IR). It is first introduced in document classification which models a document as a collection or a bag of words regardless of grammar and word order. Thus a document can be represented by a sparse histogram over the vocabulary. The assumption that each word is independent might be over simplified, but the model is very effective and robust. Experiments conducted in [123] found that the representations that are more sophisticated than BoW do not yield significantly better effectiveness. If treating images

as documents and image features/patches as words, an image can be represented by a bag of visual words which is a sparse histogram over a vocabulary of image patches. As a result, a combined vocabulary $V = (w_1, ..., w_V)$ can be generated from a video collection, which contains both textual words from metadata and visual words from raw video frames (or images).

To generate the codebook, visual and textual feature extractions need to be performed on the video collection first. In visual feature extraction, each video is usually divided into shots by shot boundary detection methods, and then visual features are detected from the keyframes of the shots, and each keyframe is represented by several local patches. Scale-invariant feature transform (SIFT) [32] is one of the most famous descriptors that can handle intensity, rotation, scale and affine variation to some extent. SIFT converts each patch to a vector of 128 dimensions which is also called a keypoint. Each keyframe is now represented by a bag of $128-$dimensional keypoints, where each keypoint is considered to be independent. Then clustering is performed on the vectors from all the keyframes to group visually similar patches into the same group. The centers of the clusters are defined as visual words and the number of clusters is the size of the visual vocabulary. After clustering, a cluster membership is assigned to each keypoint of a keyframe and a keyframe can be represented by a histogram of visual words. By adding the histograms of the keyframes from the same video, a video can be represented by a histogram of visual words.

Compared to the visual codebook generation process, the generation of textual codebook is fairly straightforward. The typical filter steps such as stop word removal, synonym expansion, and stemming are necessary as pre-processing. The final forms of terms are considered as the words in the textual codebook, and the metadata associated

with a video can also be mapped to a histogram of textual words. By combining visual and textual codebooks, a video can be represented by a histogram of a single combined codebook with size V. Please note from now on, we do not distinguish visual words from words unless explicitly used.

The work in [124][125][126] consider scenes as the elementary units since they depict self-contained high-level concepts and are mostly autonomous in their meaning. Generally speaking, an image usually contains several different scenes, analog to multiple topics of a document. Hence, it is natural to apply topic models [127] in text mining to tackle the multiple scene problem in images. Given a large collection of unstructured documents, as a type of statistical model, a topic model can uncover the underlying semantic structure of the corpus and automatically discover the latent topics in it. Each topic is a cluster of words that frequently occur together and each document exhibits these topics with different proportions. In topic models, Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) [128] are two representative models that have been widely used. Compared to LDA, pLSA is not scalable since it fixes the topic mixture probability for documents once the model is estimated, and needs to be re-estimated when new documents arrive. In contrast, LDA represents a document as random mixtures over latent topics, denoted as $\mathbf{Z} = (z_1, ..., z_k, ..., z_K)$, where K is the total number of topics, and each topic z_k is characterized by a distribution over words. LDA assumes that words are exchangeable within each document and documents are exchangeable within the corpus. Considering this strong assumption of LDA, several topic models such as correlated topic model (CTM) and dynamic topic model (DTM) are developed based on LDA to loose these constraints or extend it to adapt to certain situations. As the basic form of these variants, LDA is adopted in this section to il-

lustrate the idea, but other methods based on topic models also apply. Borrowing the idea of topic models, an image can also be considered as a mixture of latent scenes, and a scene is a mixture of visual words. In [129], LDA has shown very promising results in categorizing 13 natural scenes. If using a topic in general to stand for both scenes of keyframes and topics of metadata, LDA can model each video as a mixture of topics while each topic is a mixture of words in the combined vocabulary. The topic distribution of a video and the word distribution of a topic can therefore be estimated.

Problem Formalization

Suppose we define K independent topics for a video collection $\mathbf{D} = (d_1, ..., d_i, ..., d_N)$ of size N, and each video d_i in $\mathbf{D}$ is independent from each other, the goal is to calculate the topic distribution of a video, denoted as the probability of the topic set $\mathbf{Z}$ given d_i, $P(\mathbf{Z}|d_i)$. Figure 5.9 is the plate notation of smoothed LDA, where the boxes are "plates" representing replicates. Shaded ones are observed variables and unshaded ones are unobserved or the so called latent variables. A video d_i that contains M_i words is represented by a vector $\mathbf{W_i}$ of dimension M_i, which is the only observed variable. Notations in the figure are explained as follows:

- $\boldsymbol{\theta}$ is a $N * K$ matrix where each row $\boldsymbol{\theta_i}$

  is the Dirichlet distribution of video d_i over total K topics.

- $\boldsymbol{\phi}$ is a $K * V$ matrix where each row $\boldsymbol{\phi_k}$

  is the Dirichlet distribution of topic z_k over total V words.

- $\boldsymbol{\alpha}$ is the K-dimensional parameter of $\boldsymbol{\theta_i}$,

  which means the prior weights of a video over K topics.

- $\boldsymbol{\beta}$ is the *V*-dimensional parameter of $\boldsymbol{\phi}_k$

  which means the prior weights of a topic over *V* words.

- $\boldsymbol{Z_i}$ is a M_i vector for d_i where each element z_{ij}

  (denotes the topic for word w_{ij}) is

  a multinomial distribution with parameter $\boldsymbol{\theta}_i$.

- $\boldsymbol{W_i}$ is a M_i vector for d_i where each element w_{ij}

  (denotes a word in *V*) is

  a multinomial distribution with parameter $\boldsymbol{\phi}_{Z_{ij}}$.
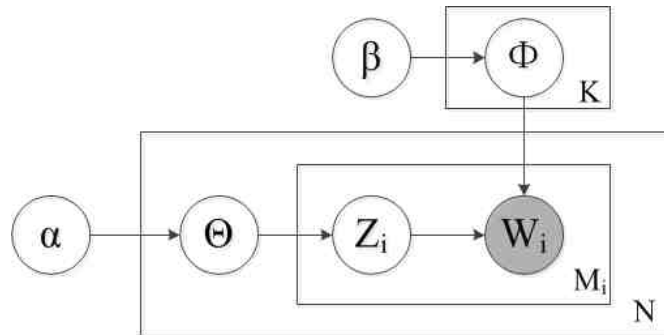
Figure 5.9: Plate notation for smoothed LDA

To learn the topic distribution of a video is to estimate $P(\boldsymbol{Z}|d_i)$, which satisfies Equation (5.5). It constrains the summation of the probability distribution on all the topics to be 1. Estimating $P(\boldsymbol{Z}|d_i)$ is equivalent to estimate $\boldsymbol{\theta}$ since $P(\boldsymbol{Z}|d_i)$ is a row vector of $\boldsymbol{\theta}$. This can typically be solved by Gibbs sampling [130] or variational Bayes approximation [128]. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are prior weights which are predefined random or empirical values. As mentioned before, we use keyframes to represent the visual content of a video, so the topic distribution calculated is frame based, and the average topic distributions of

the keyframes extracted from a video is used to represent the topic distribution of the video. Figure 5.10 shows an example of the topic distribution of a video on the first 10 topics when the total number of topics is 50.

$$\sum_{k=1}^{k=K} P(z_k|d_i) = 1 \tag{5.5}$$
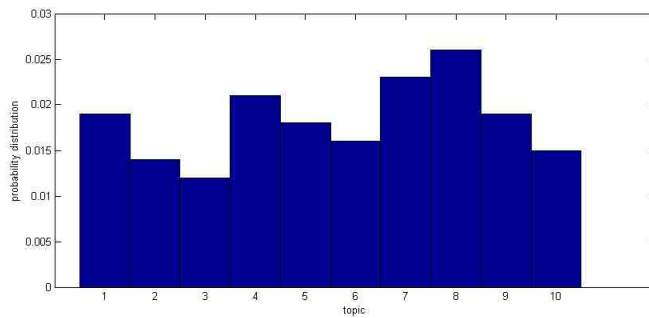
Figure 5.10: An example of the topic distribution of a video

If $\boldsymbol{H} = (d_1,...,d_i,...,d_G)$ denotes the video set containing G videos that have been watched by a user, the user's interests can be computed by the average topic distributions of the videos in $\boldsymbol{H}$. Equation (5.6) shows how to estimate a user's interests based on his or her video history $\boldsymbol{H}$.

Given the reasonable assumption that a user would like to watch videos having contents consistent with his or her interests, the problem of recommending videos can be formalized into an optimization problem, which finds videos having topic distribution as close as the topic distribution of the user's interests, as expressed by Equation (5.7), where d_r denotes the recommended video. ℓ_1-norm or Manhattan distance is adopted to measure the difference between two distributions. For top-n recommendation, the top n ranked videos generated by Equation (5.7) are recommended to the user, and the time complexity of generating the recommendations is O(N*K).

$$P(\mathbf{Z}|\mathbf{H}) = \frac{1}{G} \sum_{i=1}^{i=G} P(\mathbf{Z}|d_i) \tag{5.6}$$

$$\arg\min_{d_r} \quad \|P(\mathbf{Z}|d_r) - P(\mathbf{Z}|\mathbf{H})\|_1$$

$$\tag{5.7}$$

$$= \quad \sum_{k=1}^{k=K} |P(z_k|d_r) - P(z_k|\mathbf{H})|$$

A Practical Framework

Figure 5.11 presents a practical framework for VideoTopic, with the two key components highlighted in bold lines. The video representation can be further divided into three sub-modules: visual feature extraction, textual feature extraction, and topic model. All these tasks can be done offline to compute the topic distribution of each video using LDA, that is $P(\mathbf{Z}|d_i)$. If a new video is added to the collection, with vocabulary and topic fixed, its topic distribution can be calculated using Gibbs sampling without re-estimating the model parameters. Based on the estimated topic distributions of videos, the recommendation generation component can calculate the topic distribution of a user's interests $P(\mathbf{Z}|\mathbf{H})$ according to Equation (5.6) in the user interests estimation sub-module. This simple module allows online updating. That is, for a new user, the interests are learned on the fly as he or she watches the videos; while for an old user, the current interests can be calculated based on the old interests and the current watched videos. After knowing the user's interests, the topic distribution distance calculation sub-module can generate a personalized recommendation list by solving Equation (5.7).

5.2.2 Experimental Results

In the experiments, the performance evaluation of the proposed VideoTopic framework is conducted by first validating the usefulness of the topic representation of videos, and
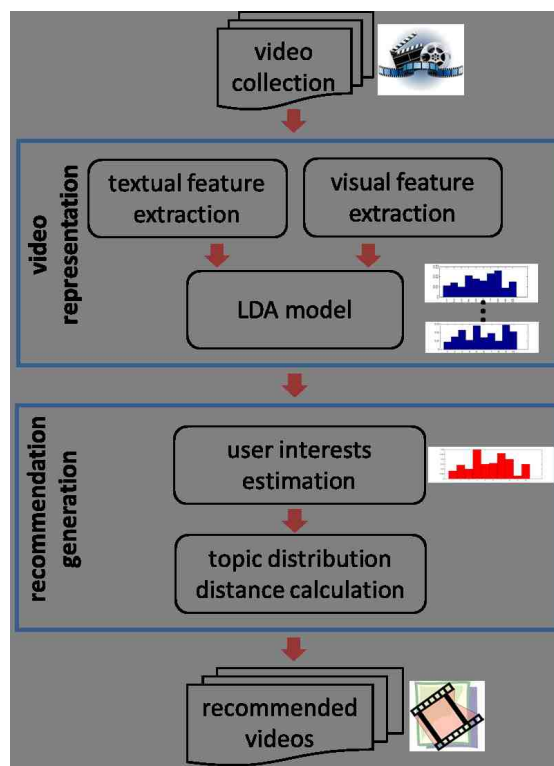
Figure 5.11: A practical framework of VideoTopic

then comparing with other approaches which are able to perform content-based video recommendation. The same evaluation metrics used to evaluate MSLIM are adopted to evaluate VideoTopic.

Data Collection

The MovieLens 1M data set[1] is chosen to evaluate our proposed framework because it is widely used and publicly available. It contains $1,000,209$ anonymous ratings of approximately $3,900$ movies made by $6,040$ MovieLens users. The textual information we used is movie genre information included in the data set. In addition, we crawl movie metadata from the Internet Movie Database (IMDB) using My Movie API[2]. The crawled information includes plot, actors, directors, and writers, but only actors and directors are used considering the quality and the importance of these features, as reported in [78]. For the visual information, we extract visual features from movie trails since trails can usually well represent the important visual content of a movie. We use movie titles, years and "trailer" as keywords to retrieve movie trailers using YouTube API[3], and download the top ranked most relevant videos from YouTube[4]. The trails are typically within the length of 1 minute to 4 minutes. 3 keyframes are extracted from each video. Given the fact that some movies are old and not popular, the retrieved movie trailers are not all correct. We have to manually check the downloaded videos by checking their titles in order to filter out some false positive ones. So the total number of movies we used in the experiments is 3475, and user ratings are removed accordingly. Similar to the experiment done in [78], we randomly split the movies into 5 fold of roughly equal size, and assign ratings to each fold accordingly to perform 5-fold cross-

---

[1]http://www.grouplens.org
[2]http://imdbapi.org/
[3]https://developers.google.com/youtube/2.0/developers_guide_protocol
[4]http://www.youtube.com

validation. Therefore, the items in the test set are new items which do not have any behaviors in the training set.

Results and Discussion

To evaluate the performance of VideoTopic, we conduct experiments in two parts. The first part is to verify the usefulness of the topic representation on new items. For general purposes, a popular model k-nearest neighbor (kNN) is used as the recommendation algorithm to replace the recommendation generation component in Figure 5.11. The input of kNN is the topic distribution of each video, which can be viewed as features. Cosine similarity is adopted to calculate item similarities using the topic features. The kNN model fed with pure visual features extracted from videos is denoted as V-kNN, and T-kNN is the kNN model fed with pure textual features. Rule-based late fusion is employed to combine visual similarity and textual similarity, more specifically, the linear weighted sum approach is used, and the method is denoted as Fusion-kNN. As shown in Equation (5.8), s^t is the similarity score from T-kNN, s^v is the score from V-kNN, and w^t and w^v are their corresponding weights. Since k-NN does not need a training phase, w^ω is empirically determined. We set the number of topics to be 25 for both T-kNN and V-kNN, and vary w^v which is the weight or proportion of the similarity score from V-kNN. The performance of Fusion-kNN on these metrics exhibit a similar pattern, and thus we only show the result of AUC in Figure 5.12. It achieves the highest AUC at $w^v = 0.1$ which means $w^t = 0.9$. This indicates the textual information of this data set is more reliable than the visual information since the weight is biased to T-kNN.

$$\sum_{\omega=1}^{\omega=2} \quad w^{\omega} \times s^{\omega}$$

(5.8)

$$where \quad \sum_{\omega=1}^{\omega=2} w^{\omega} = 1$$
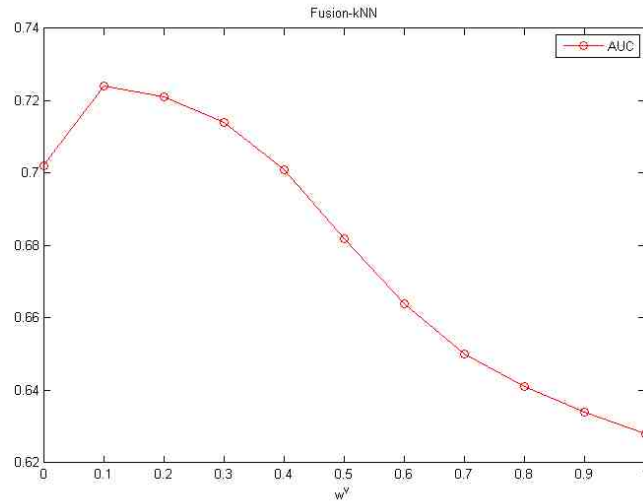
Figure 5.12: Fusion-kNN varied by w^v

After deciding the best values for w^t and w^v in Fusion-kNN, we compare the performance of T-kNN, V-kNN, Fusion-kNN as well as the randomly generated results which correspond to the collaborative filtering based methods when dealing with new items. Their performance results are presented in Figure 5.13. The results of V-kNN on all four metrics are much better than the random generated results, which proves that visual features can provide some useful information. However, T-kNN still outperforms V-kNN by a large margin. This confirms the conclusion we draw from Figure 5.12, which is also consistent with the observation found in [80]. When combining these two information sources, the performance of Fusion-kNN using the empirically tuned parameters is better than that of the V-kNN and T-kNN. We also see that the number of topics affects the model performance. Comparing with V-kNN, the performance of

T-kNN stays relatively stable as the number of topics increases. However, the results from V-kNN drop quickly. The reason is that the visual features we extracted from videos have low quality which brings more noise when the number of topics increases. On average, the best performance of T-kNN and V-kNN is achieved when the numbers of topics are 50 and 20, respectively. For Fusion-kNN, the optimal number of topics is 50, and an equal number of topics is given to T-kNN and V-kNN, which is 25, to prevent one model from overshadowing the other.
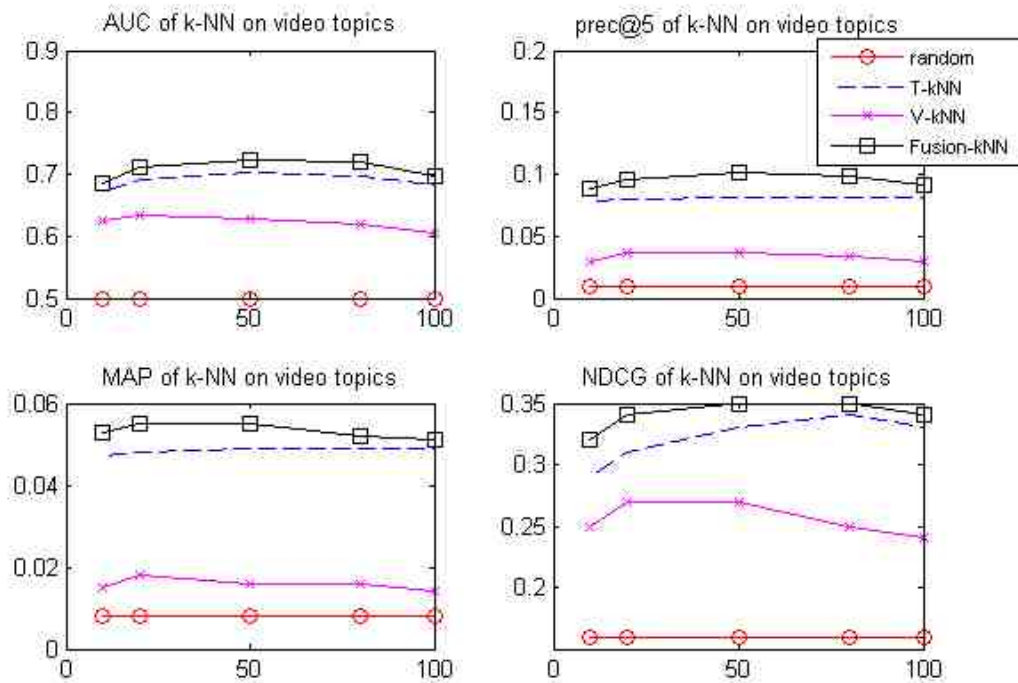
Figure 5.13: NDCG of k-NN on video topics

The second part is to compare the whole framework with the two methods discussed in Chapter 2. The component of recommendation generation is also evaluated by comparing to Fusion-kNN which uses the same topic representation of videos as the input. Fusion-kNN is set as a baseline method, and as mentioned before, the number of topics

is set to 50. The same number of topics is used for VideoTopic. VideoReach introduced in [80] is chosen as a comparison method, where only visual and textual features are fed into the model. According to the properties required by the Attention Fusion Function, VideoReach first filters out videos with low textual similarity to assure all videos are more or less relevant with the query video, and then it only calculates the visual similarity within these videos. We use the grid search to find this filtering threshold, and the reported results are generated by the best threshold when the textual similarity equals 0.6. However, no relevance feedback in this data set is available to adjust the weights of the features in a single modality. As a result, the weight of each feature is set to 1 and the weight of each modality is set to the parameters we tuned for Fusion-kNN. The method used in VideoReach is essentially filtering plus Fusion-kNN, denoted as Filter-Fusion-kNN. Another comparison method is the work presented in [131], which recommends the top 5 ranked videos of the same topic followed by the videos of related topics in the topic network. The topics in [131] are predefined and the construction of the topic network is designed for news videos. Hence, it cannot be applied to the topics here since they are not expected to have any connection. In addition, each video is represented as a distribution over topics rather than being classified into a single topic. Therefore, we ignore the part of recommending videos of related topics and only recommend the top ranked videos of the majority topic as identified from the watched videos. This method is called OneTopic.

Table 5.2 shows the comparison results of the four methods. VideoTopic performs the best in prec@5, AUC, and NDCG, but it is slightly lower than Filter-Fusion-kNN in MAP. On average, Filter-Fusion-kNN achieves the second best results followed by Fusion-kNN. OneTopic gives the worst performance, which is because it only consid-

Table 5.2: Comparison results of VideoTopic

	prec@5	AUC	MAP	NDCG
Fusion-kNN	0.10	0.69	0.062	0.39
Filter-Fusion-kNN	0.11	0.70	**0.072**	0.41
OneTopic	0.08	0.65	0.058	0.39
VideoTopic	**0.14**	**0.75**	0.071	**0.45**

ers the most important topic and discards the information from the rest topics. The relatively high precision of Filter-Fusion-kNN is due to the effect of filtering using textual similarity which removes some noisy irrelevant videos. The fact that VideoTopic outperforms Fusion-kNN validates the effectiveness of the recommendation generation component.

5.2.3 Conclusions

This section presents a recommendation framework that focuses on using a topic model to represent the textual and visual content of the videos in an integrated manner. Topics are used to link video content and user interests which are estimated from the topics of users' previous watched videos. Based on each user's interests, recommending a personalized list of videos is formulated into an optimization problem which maps the problem of maximizing the recommendation accuracy to minimize the topic difference between user interests and the recommended videos. The evaluation on MovieLens 1M data set confirms that for new items, visual information does help and VideoTopic outperforms the other three comparison methods. From the results, we can see that the number of topics plays a role in the final performance, which is a limitation for most of the methods based on the topic models. In particular, V-kNN is subject to the influence of the number of topics because of the relatively low quality of the visual features.

Filter-Fusion-kNN also suggests pre-filtering videos with low textual relevance (i.e., removing noisy videos) can improve the results. Therefore, in our future work, we need to consider the topics used in Equation (5.6) and Equation (5.7), and how to automatically identify the important topics. Another limitation of our current work is that the topic distributions of videos are based on the average of the topic distributions of keyframes. The temporal information needs to be take into the consideration of the framework.

Chapter 6

Conclusions and Future Work

This chapter summarizes the framework of integrating content and context modalities for multimedia big data retrieval. Based on the limitations discussed in previous chapters, several directions of future work are discussed that can improve the framework.

6.1 Conclusions

On the consequences of the exponential growth of multimedia data, the demand for effective and automatic information retrieval has increased tremendously. Besides the intrinsic challenges of multimedia information retrieval such as the semantic gap, the big data era also raises the difficulty in processing large amounts of multimedia data from heterogeneous and distributed data sources. Motivated by the facts that, on one side, the context information of images and videos such as tags, titles, descriptions and surrounding text can help bridge the semantic gap between the low-level visual features and the high-level semantic concepts, and on the other side, the context information contains a lot of noise since it is generated by users, this dissertation takes advantage of both modalities and builds a scalable framework that can handle multimedia big data. A framework containing two components, namely MCA-based feature selection and

sparse linear integration, integrates the content and context modalities at the intermediate level for multimedia big data retrieval.

Considering that the numbers of visual features and textual features being extracted are usually large, it is necessary to use a feature selection component, which can not only improve data quality by removing noisy features, but also reduce computation time by removing irrelevant features. A supervised feature selection method is developed, which utilizes MCA to calculate the correlation between a feature and a class, and those features having high correlations with the class are selected. This method can effectively reduce the feature dimensions without altering the feature space. The advantage of keeping semantic meaning of features is that it can be used to remove noisy tags in the context modality. Since MCA is originally designed for nominal data, a discretization method is developed accordingly to extend MCA to numeric data. The MCA-based discretization method utilizes MCA to measure the correlation between feature intervals of a feature and the classes. The candidate cut-point that maximizes the correlation between feature intervals and classes is selected as a cut-point. This strategy is carried out in each interval recursively to further partition the feature. The correlation between a feature and the class label can be further used in classification. An MCA-based discriminative learning framework for video semantic classification is presented to address the challenges such as semantic gap, imbalanced data, and high-dimensional feature space in automatic multimedia semantic analysis. The correlation information from the MCA-based feature selection is reutilized to build two models based on the transaction weights, and a strategy is proposed to fuse these two models into a more powerful classifier. Evaluations of each of the methods in this component are conducted by comparing them with some representative work in the same area. Dif-

ferent data sets are used including general UCI machine learning data sets and specific multimedia data sets. Experimental results demonstrate the effectiveness of each of the proposed methods.

To integrate content and context modalities, a sparse linear integration model learns a pairwise instance similarity matrix by formulating it into an optimization problem. Sparsity constraint is imposed to enable fast update, which can also limit noise propagation. Coordinate descent and soft-thresholding are applied to solve the problem. The learned model is a pairwise instance similarity matrix which can be directly used for unsupervised applications. A classifier based on the reconstruction error can also be embedded in this model for supervised learning. To cope with situations when the instances from different modalities are not in the same level/granularity, a generalized sparse linear integration model is provided by utilizing instance associations between different modalities and the same solution based on coordinate descent and soft-thresholding can be applied. Two benchmark image datasets are used evaluate the sparse linear integration method and a benchmark video collection is used to evaluate the generalized model. Results from the content and context modalities alone verify the motivation of this dissertation, and the comparisons with similar work in this field show promising results of the proposed component.

As a step forward compared to the traditional information retrieval, the recommender system has gained increasing attention in recent years which involves users as the third dimension in addition to items/instances and item content. One application that applies the sparse linear integration model to integrate item metadata and the associated image or content-based item recommendation is presented. Subjective evaluations are conducted on a self-collected handbag data set to compare the method based

on the sparse linear integration model with three other content-based recommendation methods. The other application used a topic model, called VideoTopic, to represent both content and context modalities in an unified manner for content-based video recommendation is also given. Topics are used to link video content and user interests which are estimated from the topics of users' previous watched videos. Based on each user's interests, recommending a personalized list of videos is formulated into an optimization problem which maps the problem of maximizing the recommendation accuracy to minimize the topic difference between user interests and the recommended videos. The evaluation on MovieLens 1M data set confirms that for new items, visual information does help and VideoTopic outperforms the other three comparison methods.

The learning process of the MCA-based feature selection can be decoupled by features, and using the Burt matrix in MCA instead of the indicator matrix virtually allows an unlimited number of instances. The sparse linear integration method can be decoupled by instances, but the calculation of the similarity coefficients of each instance still needs the information from other instances, so the number of instance is still limited to the memory size. However, a cascading approach can be adopted to first calculate the similarity coefficients in each split of the instances, and further calculate similarity coefficients using only instances with big coefficients in each split. Meanwhile, the number of features can be limited to a certain range after MCA-based feature selection. Therefore, the whole framework is scalable and can handle big data by parallel and distributed computing such as multithreading or more effectively by adopting the MapReduce framework.

6.2 Future Work

Based on the limitation of the current solutions, especially its capability of handling big data, several future research directions are identified and discussed in the following sections.

6.2.1 Parallelizing the Framework Using Hadoop

The two components, MCA-based feature selection and sparse linear integration support parallel computing as the model can be decoupled by features or by instances. The straightforward way is to run the model in parallel using multithreading. However, there are some potential problems with multithreading. For example, the single machine is still limited to its processing capacity. If using multiple machines, then the issues such as coordination and reliability will arise [1]. Though it is feasible to parallelize the processing, it is messy in practice and therefore using the MapReduce framework can be a great help. MapReduce framework breaks the processing into two phases: the map phase and the reduce phase. Each phase has key-value pairs as input and output. The open source project Hadoop [132] and the distributed file system HDFS are widely used as the implementation of the MapReduce framework. Hadoop runs the job by dividing it into two types of tasks: map tasks and reduce tasks. The input data are first split into fixed-size subsets expressed by the key-value pairs. Then each split is processed by a map task which generates a list of intermediate key-value pairs. These intermediate key-value pairs are sorted by the keys and then input to the reduce task. If there are more than one reduce task, a shuffle of the sorted intermediate key-value pairs is performed as each reduce task is fed by many map tasks. The data flow of MapReduce is depicted in Figure 6.1.
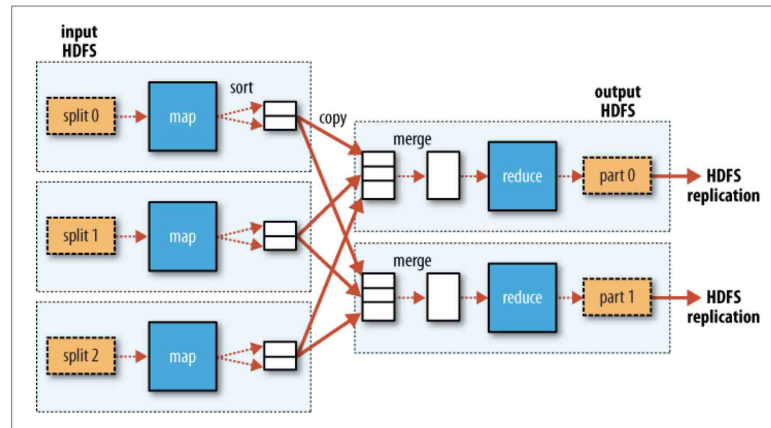
Figure 6.1: MapReduce data flow [1]

The MCA technique that calculates the correlation between each feature and the class label has been parallelized using Hadoop [133]. We can adopt it for MCA-based feature selection. Figure 6.2 illustrates the MapReduce data flow of the sparse linear integration model. The input data are the feature representations of all the instances from multiple modalities, which are $\boldsymbol{X}^{\omega}$ ($\omega \in [1, \Omega]$) in Equation (4.11). The data is split by instances and key-value pairs $< K_1, V_1 >$ are formed, where K_1 can be the instance ID, and V_1 is the feature representation of this instance. In the map task, Equation (4.11) or Equation (4.7) is calculated to get $\boldsymbol{s}_j$ which is the same instance identified by K_1. Since the process of sparse linear integration can be carried out entirely in parallel, there is no need for a reduce task. The output of each map task is denoted as $< K_2, V_2 >$, where V_2 is $\boldsymbol{s}_j$, is the final output. If the number of instances is too large, then we need to split the instances and perform the aforementioned map task on each split. The output V_2 would be the instances with big similarity coefficients. Then a reduce task of the same process as the map task is performed on these selected instances from each split to get the final output $\boldsymbol{s}_j$. Unsupervised or supervised applications can use the calculated $\boldsymbol{s}_j$ of each instance to conduct further processing such as clustering and classification.
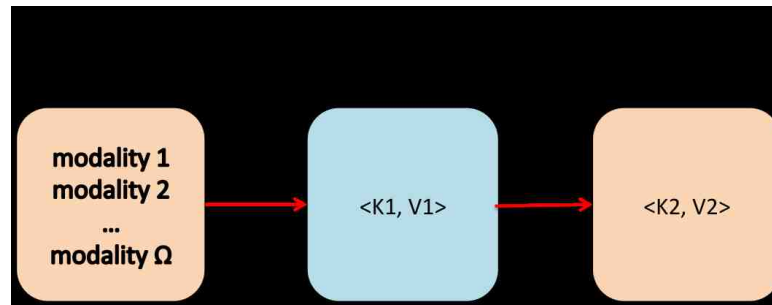
Figure 6.2: MapReduce logical data flow of the sparse linear integration model

6.2.2 Clustering Instances to Reduce Computational Complexity

Besides adopting Hadoop to boost the computation power, instances can be selected to reduce computational complexity from a model point of view. Currently all the instances are used for unsupervised learning, and for supervised learning, all the positive instances of each class are used to build a model for this class. Considering the number of instances are getting larger and larger, using representative instances can not only dramatically reduce computational complexity, but also potentially build a better model.

The technique used for this purpose is clustering. For both supervised and unsupervised applications, the input instances of SLI are first clustered into groups using various cluster models, such as k-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [134], and Gaussian Mixture Models (GMM) [135], so similar instances are grouped together. Figure 6.3 shows an example of 5 clusters, the red dots denote the centroid of clusters. Then SLI is applied on the clusters instead of applying on the instances. To be more specific, the centroid of a cluster is used to represent all the instances belonging to this cluster, which is as an input instance to SLI. Therefore, the computational complexity is greatly reduced. The output of SLI now are the similarities between clusters, which indicate the distances between clusters. As

shown in Figure 6.3, the line between the centroids of cluster 1 and cluster 2 denotes the distances between these two clusters, a dashed line denotes the distances between an instance and the centroid of the cluster it belonged to. The distance of an instance to its centroid can be treated as an offset to the distance between two clusters. Therefore, the similarity between two instances can be further calculated using the distance between clusters and their offsets within clusters.
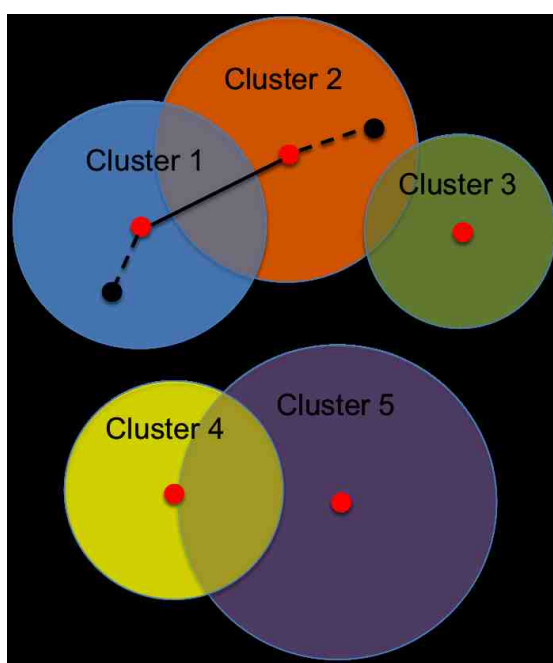
Figure 6.3: An example of instance clusters

6.2.3 Incorporating Unsupervised Feature Selection

As discussed in Chapter 1, the feature dimensions of the two modalities need to be in the same scale for the sparse linear integration component. Otherwise the learn model would lean toward the one with higher dimensions. In the experiment, we control the extracted features from these two modalities. To be specific, we fix the number of visual features, and use MCA-based feature selection to remove noisy features from

the context modality and make the number of textual features close to the number of visual features. This is a practical approach, which uses a proven-effective feature selection method to reduce feature dimensions. However, the sparse linear integration method can be used for unsupervised application directly, while the MCA-based feature selection and many others discussed in this thesis are supervised approaches. In order to make the sparse linear integration component more generic, an unsupervised feature selection method is desired as well.

Dimension reduction, such as PCA [136] and several others as discussed in Chapter 2, can fulfill this purpose since they are also unsupervised approaches. However the features they produce often are not readily interpretable, which is a disadvantage of these methods, especially for the context modality with terms as its features. Transformation to the principle component space would loss track the meaning of the features. Therefore, unsupervised feature selection is preferred in the context modality in order to keep the interpretability of features. Unsupervised feature selection methods are mostly based on clustering [137][138]. In general, similarity is measured between features and redundancy can be removed accordingly. As an essential clustering technique, topic models can be used to reduce dimensions while keeping the semantics. It automatically discover the latent topics in the corpus constructed by metadata. Each topic is a cluster of terms/words that frequently occur together, which can be viewed as features, and each instance exhibits these features with different proportions. However, most popular topic models, such as LSA and LDA, are effective to long documents. The metadata of images and videos is usually very short in length, which make them not suitable for common topic models. One approach is to resort to topic models that are designed for short documents [139][140]. Another possible approach to is to utilize tag corre-

lations [141] to fill the sparse instance-feature matrix so that for example an instance containing 10 tags would now contain 100 tags with different association based on tag correlation. This essentially increases the document length so that common topic models can be applied. Experiments need to be conducted to compare this approach with topic models designed for short documents.

6.2.4 Improving Data Quality in the Context Modality

Feature selection can remove some of the noise in the context modality, such as redundant terms and irrelevant terms, but its ability towards improving data quality in the context modality is still limited. For example, how can one handle missing tags? How can one handle synonyms, hypernyms and hyponyms? One direction is to utilize the visual information that metadata associated with [142], as introduced in Chapter 1. Another direction is to resort to external knowledgebase, such as WordNet and Wikipedia, to handle noisy metadata. A recent publication [143] jointly analyzes three distinct visual knowledge resources which are Flickr, ImageNet[1]/WordNet, and ConceptNet [144], to infer incomplete tag relationships.

Since the information from the content modality is used for integration with the context modality, a preferred direction is to use external textual or visual knowledge to enhance the data quality of the context modality. An initial attempt will be focused on textual recourse WordNet and ConceptNet. WordNet is a large lexical database of English. Words having similar meanings (synonyms) are grouped into sets of cognitive synonyms (synsets), and synsets are further connect by the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation). ConceptNet is a semantic network that contains everyday basic knowledge. It is built from nodes representing

---

[1]http://image-net.org/

concepts, in the form of words or short phrases of natural language, and with relationships labeled between them. These relationships are richer than the ontological relationship provided by WordNet. Figure 6.4 and figure 6.5 show an examples of WordNet[2] and ConceptNet[3], respectively. As can been seen, words or concepts are connected by some predefined relationships, which are not considered in our current context modality. Given these extra knowledge, we can utilize it to enhance the context modality, such as noise removal and tag completion.

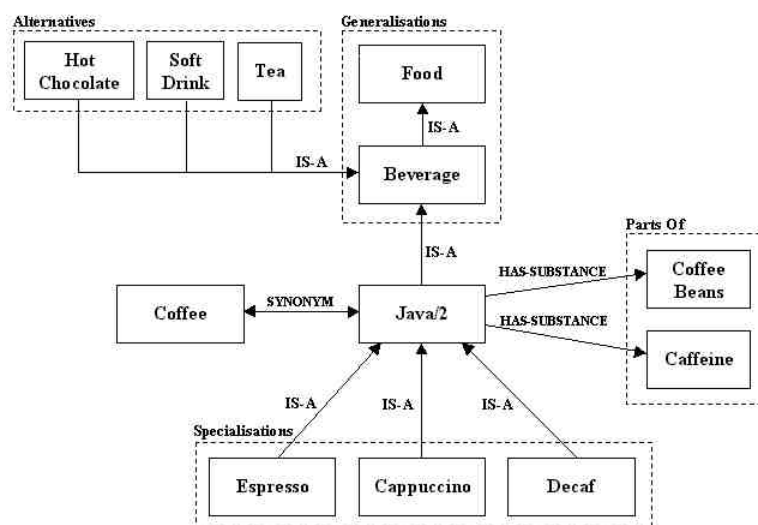Figure 6.4: An example of WordNet

6.2.5 Evaluating the Sparse Linear Integration Component for Queries based on Single Modality

In our previous discussion, we assume for a query instance, both its content and context modalities are available. In fact, this assumption is applied to all the instances, either for supervised or unsupervised applications. However, in real situations, it's possible

---
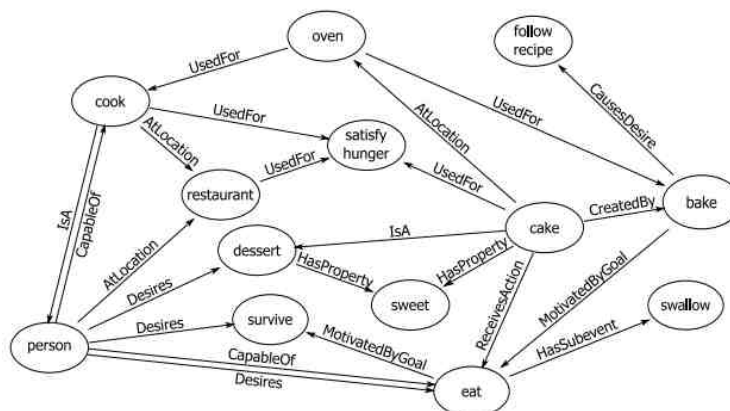
Figure 6.5: An example of ConceptNet

that for an instance, only one of the modalities is available. The popular approach is to preprocess the data and fill in the missing values. K-nearest neighbor are often used to fill the missing values [78]. In this case, it first finds the neighbors of the instance based on its existing modality, and then uses the other modality of the neighbors to approximate its missing modality. After this preprocess, both modalities are available, and common fusion methods can be applied to get the integrated results.

In SLI, we can treat the unavailable modality of an instance as missing values. That is, set the values of the features corresponding to this unavailable modality to be 0, and let these values be updated in the iterations. As discussed by Ning et al. [101][102], the missing values can be approximated by the linear combination of the existing feature representation of its similar instances, which is originally used to predict ratings in recommender systems. Therefore, the missing values in X are approximated by XS. Based on the theoretic analysis, we can see that SLI can handle the missing modality problem. However, its capability needs to be examined. Evaluation can be conducted directly on SLI without any preprocess and on SLI after the missing modality has been treated by k-NN. If they achieve a similar performance, then this indicates that SLI can

handle missing modalities inherently. If possible, some other methods capable of filling missing values should be used as preprocessing to replace k-NN. Fusion methods that can directly handle missing modalities can be further included in the evaluation. For example, this problem has been considered in the work of Caicedo et al. [2], which is one of the comparison method we used to evaluate SLI. In this method, back projection of the factorized matrix is used to approximate the missing modality. It would be interesting to compare the performance drop of SLI with this method due to the missing modality.

Bibliography

[1] T. White, *Hadoop: The Definitive Guide*, 3rd ed. O'Reilly Media, Inc., 2012.

[2] J. C. Caicedo and F. A. González, "Multimodal fusion for image retrieval using matrix factorization," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, 2012, pp. 56:1–56:8.

[3] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[4] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, February 2006.

[5] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 644–655, 1998.

[6] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 81–93, 1999.

[7] T. Meng and M.-L. Shyu, "Concept-concept association information integration and multi-model collaboration for multimedia semantic concept detection," *Information Systems Frontiers*, vol. 1, no. 1, pp. 1–13, April 2013.

[8] T. Meng, M.-L. Shyu, and L. Lin, "Multimodal information integration and fusion for histology image classification," *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 2, no. 2, pp. 54–70, April 2011.

[9] D. Liu and M.-L. Shyu, "Semantic motion concept retrieval in non-static background utilizing spatial-temporal visual information." *International Journal of Semantic Computing*, vol. 7, no. 1, pp. 43–68, 2013.

[10] D. Liu, M.-L. Shyu, and G. Zhao, "Spatial-temporal motion information integration for action detection and recognition in non-static background," in *2013 IEEE 14th International Conference on Information Reuse and Integration (IRI)*, August 2013, pp. 626–633.

[11] X. Cheng, C. Dale, and J. Liu, "Statistics and Social Network of YouTube Videos," in *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, jun 2008, pp. 229–238.

[12] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, "The youtube video recommendation system," in *Proceedings of the 4th ACM conference on Recommender systems*, 2010, pp. 293–296.

[13] X. Zhao, H. Luan, J. Cai, J. Yuan, X. Chen, and Z. Li, "Personalized video recommendation based on viewing history with the study on youtube," in *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*, 2012, pp. 161–165.

[14] M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G. J. F. Jones, "Automatic tagging and geotagging in video collections and communities," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, 2011, pp. 1–8.

[15] S. Siersdorfer, J. San Pedro, and M. Sanderson, "Automatic video tagging using content redundancy," in *Proceedings of the 32nd international ACM SIGIR conference on Research and Development in Information Retrieval*, 2009, pp. 395–402.

[16] R. Zhao and W. I. Grosky, "Narrowing the semantic gap - improved text-based web document retrieval using visual features," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 189–200, 2002.

[17] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, December 2000.

[18] Y. Jin, L. Khan, L. Wang, and M. Awad, "Image annotations by combining multiple evidence & wordnet," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 2005, pp. 706–715.

[19] N. Sawant, J. Li, and J. Z. Wang, "Automatic image semantic interpretation using social action and tagging data," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 213–246, 2011.

[20] H. Ma, J. Zhu, M.-T. Lyu, and I. King, "Bridging the semantic gap between image contents and tags," *IEEE Transactions on Multimedia*, vol. 12, no. 5, pp. 462–473, 2010.

[21] J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua, "Inferring semantic concepts from community-contributed images and noisy tags," in *Proceedings of the ACM International Conference on Multimedia*, 2009, pp. 223–232.

[22] G. Zhu, S. Yan, and Y. Ma, "Image tag refinement towards low-rank, content-tag prior and error sparsity," in *ACM International Conference on Multimedia (MM'10)*, 2010, pp. 461–470.

[23] X. Li, C. G. M. Snoek, and M. Worring, "Learning social tag relevance by neighbor voting," *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1310–1322, 2009.

[24] S. Lee, W. De Neve, and Y. M. Ro, "Tag refinement in an image folksonomy using visual similarity and tag co-occurrence statistics," *Image Communication*, vol. 25, no. 10, pp. 761–773, 2010.

[25] Q. Zhu, M.-L. Shyu, and S.-C. Chen, "Discriminative learning assisted video semantic concept classification," in *Multimedia Security and Steganography*, F. Y. Shih, Ed. CRC Press, 2012.

[26] L. Lin, C. Chen, M.-L. Shyu, and S.-C. Chen, "Weighted subspace filtering and ranking algorithms for video concept retrieval," *IEEE Multimedia*, no. 3, pp. 32–43, July-September 2011.

[27] T. A. S. Foundation. (2014) Apache solr. [Online]. Available: https://lucene.apache.org/solr/

[28] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*. Manning Publications Co., 2011.

[29] D. A. Lisin, M. A. Mattar, M. B. Blaschko, M. C. Benfield, and E. G. Learned-miller, "Combining local and global image features for object class recognition," in *Proceedings of the IEEE CVPR Workshop on Learning in Computer Vision and Pattern Recognition*, 2005, pp. 47–55.

[30] S. A. Chatzichristofis and Y. S. Boutalis, "Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval," in *Proceedings of the 6th international conference on Computer vision systems*, 2008, pp. 312–322.

[31] D. Liu, M.-L. Shyu, Q. Zhu, and S.-C. Chen, "Moving object detection under object occlusion situations in video sequences," in *Proceedings of the 2011 IEEE International Symposium on Multimedia*, 2011, pp. 271–278.

[32] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision*, 1999, pp. 1150–1157.

[33] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[34] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[35] A. Abdel-Hakim and A. Farag, "Csift: A sift descriptor with color invariant characteristics," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1978–1983.

[36] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, June 2005, pp. 886–893.

[38] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," in *Document retrieval systems*, P. Willett, Ed. Taylor Graham Publishing, 1988, pp. 132–142.

[39] D. Wang, H. Zhang, W. Wu, and M. Lin, "Inverse category frequency based supervised term weighting scheme for text categorization," *CoRR*, vol. abs/1012.2609, 2010.

[40] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Information Processing and Management*, vol. 42, no. 1, pp. 155–165, 2006.

[41] D. C. Manning, P. Raghavan, and H. Schutze, "Text classification and naive bayes," in *Introduction to Information Retrieval*. Cambridge University Press, 2008, pp. 253–287.

[42] Y. G. Jiang, J. Yang, C. W. Ngo, and A. G. Hauptmann, "Representations of keypoint-based semantic concept detection: a comprehensive study," *IEEE Transactions on Multimedia*, vol. 12, no. 1, pp. 42–53, 2010.

[43] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 359–366.

[44] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognition*, vol. 42, no. 3, pp. 409–424, 2009.

[45] Y. Sun, "Iterative relief for feature weighting: Algorithms, theories, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1035–1051, 2007.

[46] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Tag recommendations based on tensor dimensionality reduction," in *Proceedings of the 2008 ACM conference on Recommender systems*, 2008, pp. 43–50.

[47] Q. Li, X. Shi, and D. Schonfeld, "A general framework for robust hosvd-based indexing and retrieval with high-order tensor data," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 873–876.

[48] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proceedings of the 3rd ACM international conference on Web search and data mining*, 2010, pp. 81–90.

[49] S. Kotsiantis and D. Kanellopoulos, "Discretization techniques: A recent survey," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 47–58, 2006.

[50] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1993, pp. 1022–1027.

[51] I. Kononenko, "On biases in estimating multi-valued attributes," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995, pp. 1034–1040.

[52] L. A. Kurgan and K. J. Cios, "Caim discretization algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 145–153, 2004.

[53] C.-J. Tsai, C.-I. Lee, and W.-P. Yang, "A discretization algorithm based on class-attribute contingency coefficient," *Information Sciences*, vol. 178, no. 3, pp. 714–731, 2008.

[54] J. Ge, Y. Xia, and Y. Tu, "A discretization algorithm for uncertain data," in *Proceedings of the 21st international conference on Database and expert systems applications: Part II*, 2010, pp. 485–499.

[55] W. Maass, "Efficient agnostic pac-learning with simple hypothesis," in *Proceedings of the 7th annual conference on Computational learning theory*, 1994, pp. 67–75.

[56] D. Janssens, T. Brijs, K. Vanhoof, and G. Wets, "Evaluating the performance of cost-based discretization versus entropy-and error-based discretization," *Computers and Operations Research*, vol. 33, no. 11, pp. 3107–3123, 2006.

[57] C.-C. Chan, C. Batur, and A. Srinivasan, "Determination of quantization intervals in rule based model for dynamic systems," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 1991, pp. 1719–1723.

[58] C. Chen, Q. Zhu, L. Lin, and M.-L. Shyu, "Web media semantic concept retrieval via tag removal and model fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 61:1–61:22, October 2013.

[59] P. Atrey, M. Hossain, A. El Saddik, and M. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia Systems*, vol. 16, no. 6, pp. 345–379, 2010.

[60] T. Meng and M.-L. Shyu, "Model-driven collaboration and information integration for enhancing video semantic concept detection," in *Proceedings of the 2012 IEEE 13th International Conference on Information Reuse and Integration (IRI)*, August 2012, pp. 144–151.

[61] D. Liu and M.-L. Shyu, "Effective moving object detection and retrieval via integrating spatial-temporal multimedia information," in *Multimedia (ISM), 2012 IEEE International Symposium on*, December 2012, pp. 364–371.

[62] M. Žitnik and B. Zupan, "Data Fusion by Matrix Factorization," *ArXiv e-prints*, july 2013.

[63] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 399–402.

[64] K. Nagel, S. Nowak, U. Kühhirt, and K. Wolter, "The fraunhofer idmt at image-clef 2011 photo annotation task," in *CLEF (Notebook Papers/Labs/Workshop)*, 2011.

[65] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative Matrix Factorization in Multimodality Data for Segmentation and Label Prediction," in *16th Computer Vision Winter Workshop*, 2011.

[66] J. C. Caicedo, J. BenAbdallah, F. A. González, and O. Nasraoui, "Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization," *Neurocomputing*, vol. 76, no. 1, pp. 50–60, January 2012.

[67] R. Lienhart, S. Romberg, and E. Hörster, "Multilayer plsa for multimodal image retrieval," in *Proceeding of the ACM International Conference on Image and Video Retrieval*, 2009, pp. 1–8.

[68] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50–57.

[69] S. Clinchant, J. Ah-Pine, and G. Csurka, "Semantic combination of textual and visual information in multimedia retrieval," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, 2011, pp. 44:1–44:8.

[70] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, july 2011.

[71] S. Yu, T. Falck, A. Daemen, L.-C. Tranchevent, J. Suykens, B. De Moor, and Y. Moreau, "L2-norm multiple kernel learning and its application to biomedical data fusion," *BMC Bioinformatics*, vol. 11, no. 1, pp. 1–24, 2010.

[72] Y.-R. Yeh, T.-C. Lin, Y.-Y. Chung, and Y.-C. Wang, "A novel multiple kernel learning framework for heterogeneous feature fusion and variable selection," *Multimedia, IEEE Transactions on*, vol. 14, no. 3, pp. 563–574, 2012.

[73] G. Shani and A. Gunawardana, "Evaluating recommendation systems recommender systems handbook," in *Recommender Systems Handbook*. Springer US, 2011, pp. 257–297.

[74] S. Rendle and L. Schmidt-Thieme, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," in *Proceedings of the 2008 ACM conference on Recommender systems*, 2008, pp. 251–258.

[75] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 880–887.

[76] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, August 2009.

[77] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 19–28.

[78] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, pp. 176–185.

[79] T. Chen, Z. Zheng, Q. Lu, W. Zhang, and Y. Yu, "Feature-based matrix factorization," 2011. [Online]. Available: http://arxiv.org/abs/1109.2271

[80] T. Mei, B. Yang, X.-S. Hua, and S. Li, "Contextual video recommendation by multimodal relevance and user feedback," *ACM Transaction on Information System*, vol. 29, no. 2, pp. 1–24, April 2011.

[81] H. Luo, J. Fan, and D. A. Keim, "Personalized news video recommendation," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 1001–1002.

[82] M. J. Greenacre and J. Blasius, *Multiple Correspondence Analysis and Related Methods*. Chapman and Hall/CRC, 2006.

[83] T. Cserhati, *Multivariate Methods in Chromatography*. John Wiley & Sons, Ltd, 2008.

[84] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, "Effective feature space reduction with imbalanced data for semantic concept detection," in *SUTC '08: Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2008, pp. 262–269.

[85] L. Lin and M.-L. Shyu, "Video high-level semantic retrieval using associations and correlations," *International Journal of Semantic Computing*, vol. 3, no. 4, pp. 421–444, March 2009.

[86] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[87] D. Lindley, "A statistical paradox," *Biometrika*, vol. 44 (1-2), pp. 187–192, 1957.

[88] Q. Zhu, L. Lin, M.-L. Shyu, and D. Liu, "Utilizing context information to enhance content-based image classification," *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 2, no. 3, pp. 34–51, 2011.

[89] A. F. Smeaton, P. Over, and W. Kraaij, "High-level feature detection from video in TRECVid: a 5-year retrospective of achievements," in *Multimedia Content Analysis, Theory and Applications*, A. Divakaran, Ed. Berlin: Springer Verlag, 2009, pp. 151–174.

[90] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, "Feature selection using correlation and reliability based scoring metric for video semantic detection," in *Proceedings of the 2010 IEEE Fourth International Conference on Semantic Computing*, 2010, pp. 462–469.

[91] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "Nus-wide: A real-world web image database from national university of singapore," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2009, pp. 1–9.

[92] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann, June 2005.

[93] Q. Zhu, L. Lin, and M.-L. Shyu, "Correlation maximisation-based discretisation for supervised classification," *International Journal of Business Intelligence and Data Mining*, vol. 7, no. 1/2, pp. 40–59, August 2012.

[94] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, "Effective supervised discretization for classification based on correlation maximization," in *Proceedings of the 2011 IEEE International Conference on Information Reuse and Integration*, 2011, pp. 390–395.

[95] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *In NIPS*. MIT Press, 2000, pp. 556–562.

[96] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, December 2004.

[97] A. Cichocki, A. H. Phan, and R. Zdunek, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, 2009.

[98] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, 2010, pp. 79–86.

[99] M. Zinkevich, M. Weimer, A. Smola, and L. Li, "Parallelized stochastic gradient descent," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 2595–2603.

[100] D. Zachariah, M. Sundin, M. Jansson, and S. Chatterjee, "Alternating least-squares for low-rank matrix reconstruction," *Signal Processing Letters, IEEE*, vol. 19, no. 4, pp. 231–234, April 2012.

[101] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *2011 IEEE 11th International Conference on Data Mining (ICDM)*, 2011, pp. 497–506.

[102] ——, "Sparse linear methods with side information for top-n recommendations," in *Proceedings of the 6th ACM conference on Recommender systems*, 2012, pp. 155–162.

[103] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 58, pp. 267–288, 1996.

[104] T. Karkkainen, K. Kunisch, and K. Majava, "Denoising of smooth images using l1-fitting," *Computing*, vol. 74, no. 4, pp. 353–376, June 2005.

[105] C. A. Micchelli, L. Shen, Y. Xu, and X. Zeng, "Proximity algorithms for the l1/tv image denoising model," *Advances in Computational Mathematics*, vol. 38, no. 2, pp. 401–426, February 2013.

[106] Y. Traonmilin, S. Ladjal, and A. Almansa, "Outlier removal power of the l1-norm super-resolution," in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, Eds., 2013, vol. 7893, pp. 198–209.

[107] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[108] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.

[109] M. Fadili, J.-L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," *The Computer Journal*, vol. 52, no. 1, pp. 64–79, 2009.

[110] Y. Prat, M. Fromer, N. Linial, and M. Linial, "Recovering key biological constituents through sparse representation of gene expression," *Bioinformatics*, vol. 27, no. 5, pp. 655–661, 2011.

[111] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, February 2009.

[112] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[113] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 2, pp. 168–186, 2007.

[114] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," 2010.

[115] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, 2008, pp. 39–43.

[116] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the hsv color space for image retrieval," in *International Conference on Image Processing (ICIP)*, 2002, pp. 589–592.

[117] D. Verma and V. Maru, "An efficient approach for color image retrieval using haar wavelet," in *Proceedings of International Conference on Methods and Models in Computer Science*, 2009, pp. 1–5.

[118] Y. Wu, E. Y. Chang, K. C. Chang, and J. R. Smith, "Optimal multimodal fusion for multimedia data analysis," in *ACM International Conference on Multimedia (MM'04)*, 2004, pp. 572–579.

[119] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[120] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quenot, "Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[121] Q. Zhu, Z. Li, H. Wang, Y. Yang, and M.-L. Shyu, "Multimodal sparse linear integration for content-based item recommendation," in *Proceedings of the 2013 IEEE International Symposium on Multimedia*, 2013, pp. 187–194.

[122] Q. Zhu, M.-L. Shyu, and H. Wang, "Videotopic: Content-based video recommendation using a topic model," in *Proceedings of the 2013 IEEE International Symposium on Multimedia*, 2013, pp. 219–222.

[123] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proceedings of the 7th international conference on Information and Knowledge Management*, 1998, pp. 148–155.

[124] S. Benini, L. Canini, and R. Leonardi, "A connotative space for supporting movie affective recommendation," *IEEE Transaction on Multimedia*, vol. 13, no. 6, pp. 1356–1370, December 2011.

[125] H. L. Wang and L.-F. Cheong, "Affective understanding in film," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 6, pp. 689–704, June 2006.

[126] A. Hanjalic, R. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, jun 1999.

[127] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, April 2012.

[128] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, March 2003.

[129] F.-F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 524–531.

[130] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling, "Fast collapsed gibbs sampling for latent dirichlet allocation," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2008, pp. 569–577.

[131] J. Liu, P. Dolan, and E. R. Pedersen, "Personalized news recommendation based on click behavior," in *Proceedings of the 15th international conference on Intelligent user interfaces*, 2010, pp. 31–40.

[132] T. A. S. Foundation. (2014) Apache hadoop. [Online]. Available: http://hadoop.apache.org/

[133] F. C. Fleites, H.-Y. Ha, Y. Yang, and S.-C. Chen, "Large-scale correlation- based semantic classification using mapreduce," in *Cloud Computing and Digital Media*, K.-C. Li, Q. Li, and T. K. . Shih, Eds.   CRC Press, 2014, pp. 169–190.

[134] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[135] D. A. Reynolds, "Gaussian mixture models." in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds.   Springer US, 2009, pp. 659–663.

[136] C. Boutsidis, M. W. Mahoney, and P. Drineas, "Unsupervised feature selection for principal components analysis," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 61–69.

[137] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, March 2002.

[138] J. G. Dy, C. E. Brodley, and S. Wrobel, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, vol. 5, pp. 845–889, 2004.

[139] A. Karandikar, "Clustering short status messages: A topic model based approach," Master's thesis, July 2010.

[140] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22Nd International Conference on World Wide Web*, 2013, pp. 1445–1456.

[141] X. Yan, J. Guo, S. Liu, X. Cheng, and Y. Wang, "Learning topics in short texts by non-negative matrix factorization on term correlation matrix," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, 2013, pp. 749–757.

[142] L. Wu, R. Jin, and A. K. Jain, "Tag completion for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 716–727, March 2013.

[143] L. Xie and X. He, "Picture tags and world knowledge," in *Proceedings of the 21th ACM international conference on Multimedia*, 2013, pp. 967–976.

[144] C. Havasi, "Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge," in *the 22nd Conference on Artificial Intelligence*, 2007.