Multiple Feature Fusion for Social Media Applications

Bin Cui\(^1\) Anthony K. H. Tung\(^2\)

\(^1\) Department of Computer Science & Key Lab of High Confidence Software Technologies (Ministry of Education), Peking University \{bin.cui,cezhang,zpegt\}@pke.edu.cn

\(^2\) Department of Computer Science
School of Computing
National University of Singapore
atung@comp.nus.edu.sg

ABSTRACT

The emergence of social media as a crucial paradigm has posed new challenges to the research and industry communities, where media are designed to be disseminated through social interaction. Recent literature has noted the generality of multiple features in the social media environment, such as textual, visual and user information. However, most of the studies employ only a relatively simple mechanism to merge the features rather than fully exploit feature correlation for social media applications. In this paper, we propose a novel approach to fusing multiple features and their correlations for similarity evaluation. Specifically, we first build a Feature Interaction Graph (FIG) by taking features as nodes and the correlations between them as edges. Then, we employ a probabilistic model based on Markov Random Field to describe the graph for similarity measure between multimedia objects. Using that, we design an efficient retrieval algorithm for large social media data. Further, we integrate temporal information into the probabilistic model for social media recommendation. We evaluate our approach using a large real-life corpus collected from Flickr, and the experimental results indicate the superiority of our proposed method over state-of-the-art techniques.

Categories and Subject Descriptors
H.2 [Database Management]: Database Applications; H.3 [Information Storage and Indexing]: Information Search and Retrieval

General Terms
Algorithms, Experimentation, Performance

Keywords
Social media, feature fusion, search, recommendation

1. INTRODUCTION

The availability of digital multimedia information continues to grow at an astonishing speed with the development of the Internet. Recent years have witnessed the socio-technological phenomenon of “social media” entering mainstream Web applications. Social media sites (e.g., Flickr, YouTube, and last.fm) are popular distribution media warehouses for users to upload, browse and share images, videos and music. These sites not only host huge amounts of user-contributed multimedia materials of wide diversity, but also serve as platforms where people express and entertain themselves and form user communities of common interests. The multimedia objects appearing in such Web sites are generally associated with textual, visual and user information, which is different from either traditional text information on the Web or multimedia objects in classic media databases.\cite{2, 11, 20, 21, 24, 27, 28} Such new characteristics of multimedia objects in the social media environment bring new research problems to Web applications, such as media retrieval, recommendation, classification, etc.

It is common knowledge that multimedia information management generally faces more problems compared with text information, e.g., the so-called “semantic gap” between the lower level content features and the higher level semantics of multimedia objects. The problems are further exacerbated in the social media environment. First, there are huge volumes of social media data on the Web, and the number of social media keeps increasing every day. For example, at the time of writing, there are more than 2 billion images posted on Flickr, and the number increases by approximately 2 million per day. It is difficult to effectively manage such a large scale database. Second, social media is much noisier than a classic multimedia database since social media are freely generated by users, and Web contents vary widely.

Fortunately, social media systems can provide some useful clues to dealing with the challenges described above. Multimedia objects in the social media environment are associated with textual information, such as title and tags. Moreover, the personal information of creator, uploader, viewer and user group usually reveals the semantics of multimedia objects. Figure 1 shows an example of an image object in a popular social media site Flickr. The image is about an animal “hamster”, which is associated with textual information such as title, comments, description and tag. Flickr also maintains various pieces of user information for this image, e.g., uploader, interested groups and users who labeled it as “favorite”. Note that, we have omitted some texts and detailed user information from the figure due to space and privacy concerns. The various pieces of information together form the multi-modal basis of social media, which provides us with the basis to manage large-scale social media in an effective way.

Yet how to effectively utilize these multiple features of social media remains an open problem. In recent years, one major research trend is to design fusion strategies of multiple features, and different kinds of techniques have been proposed for multimedia retrieval, recommendation and other applications in social systems.
Relying on feature fusion mechanisms, we can accomplish such tasks by exploiting textual, visual, audio, and user information. In the social media literature, there are two widely accepted and yet independent strategies to fusing multiple features:

- **Late fusion**: Most related work on multiple feature fusion of social media applications focuses only on the late fusion of multiple features, i.e., the fusion strategy uses separate result lists obtained from different features, and carries out fusion using these candidate results \([20, 21, 27]\). However, the late fusion strategy often obtains unsatisfactory performance if there exists correlation between original features \([17]\). In social media systems, multiple features are always correlated, and the late fusion procedure cannot effectively capture the interaction among them.

- **Early fusion**: Another line of research work tries to map multiple feature spaces to a unified space, on which traditional similarity evaluation can be conducted. For example, \([3]\) tries to define a kernel on such space based on tensor production, and M-LSA \([22]\) maps original features to a latent space with lower dimensionality based on statistical methods. Although some interactions among features can be coded into such a framework, a number of problems exist. First, such unified feature space is often built according to global statistical information, which incurs extremely high computational cost for a large scale social media database, each with tens of thousands dimensions. Second, since the objects and their associated information in social media are of wide diversity, and Web contents evolve over time, it is nearly infeasible to construct a suitable latent space or find "principal components" in such a dynamic environment. Reducing dimensionality may simply mean losing some meaningful features and correlations.

For the above reasons, both the existing late fusion and early fusion strategies do not effectively exploit multiple features and their correlations for multimedia applications in the social media environment. In this paper, we propose directly integrating multiple features and their correlation for multimedia object similarity evaluation. In that sense, our approach can be considered a variant of "early fusion" methods. Specifically, we first introduce a novel structure Feature Interaction Graph (FIG) to represent a multimedia object. The graph can also be viewed as a two-level tree with a virtual "root node" representing the object itself. All the features are represented as leaf nodes which are connected to the root, and two feature nodes are linked by an edge if there exists correlation between them. To evaluate the similarity between two objects, e.g., \(Q\) and \(O\), we replace the virtual root of FIG with \(O\), and employ a probabilistic model, i.e., Markov Random Field \([13]\), to calculate the joint distribution probability of such a graph that represents compatibility of \(O\) given the features of object \(Q\).

A large value of joint distribution probability means high similarity between two objects. For similarity retrieval in social media databases, we can simply generate FIG for the query object, and evaluate the similarity with all the objects sequentially to find similar objects. To speed up the retrieval process, we represent all data objects in the database as FIGs, and design an inverted list to index cliques which are complete subgraphs in the FIGs (to be explained in Section 3.2). The inverted index structure can provide fast access to related candidate objects for detailed similarity evaluation, and hence accelerate media retrieval.

We also investigate how our mechanism can be deployed to facilitate recommendation tasks in social media systems. Object recommendation can be modeled as a task similar to retrieval in that we examine the similarity between a certain object and a user profile, which is a set of multimedia objects related to a user according to his historical behavior. As user interest changes in the social media environment, we also propose integrating temporal information into the probabilistic model to better model user profile and compute the similarity.

We conduct extensive experiments to evaluate the performance of our proposed method against existing approaches such as \([3, 21, 22]\). Two large real-life datasets crawled from Flickr are used in our experimental study. Each dataset consists of more than 200,000 image objects. The results show the effectiveness of our approaches for multimedia retrieval and recommendation in the social media environment.

The remainder of this paper is organized as follows. In the next section, we review related work. In Section 3, we propose our probabilistic model for similarity evaluation of multimedia objects, and the retrieval algorithm in social media systems. We expand our framework to the application of recommendation in Section 4. Section 5 describes a performance study and presents a detailed analysis of the results. We conclude the paper with a brief discussion on the directions of future work.

## 2. RELATED WORK

Employing multiple features for multimedia applications has attracted more and more interests in recent years. The key problem is how to determine the similarity or relation between two objects based on these multiple features, which can facilitate various applications, such as retrieval, recommendation, classification, clustering, and so on. Typically, the multiple feature fusion techniques can be classified into two categories according to the ways they deal with the multiple features, i.e., late fusion and early fusion.

In \([21]\), Tumbull et al. compared different approaches for multiple feature fusion, e.g., Calibrated Score Averaging, RankBoost \([9]\) and Multi-Kernel SVM \([18]\), to enhance the performance of music discover. Among these methods, CSA and RankBoost emphasize the combination of result lists from different feature types, and hence these methods can be considered as late fusion solutions. Similar work can be founded in \([28]\), which also views this problem as a late fusion problem, and designs an innovative parameter tun-
ing strategy based on regression. In [20], Tollari et al. utilized a late fusion framework that linearly combines the result lists of textual and visual features for Web image retrieval. Similarly, [27] presents a novel merging method for online video recommendation that considers the user interaction with feature fusion. As reported in [17], late fusion based strategies are often less effective as they cannot capture the interaction between features, while normally there exist correlations between original features in social media applications.

There is another line of techniques utilizing early fusion strategy for multimedia applications, which try to integrate the features and their correlations for similarity calculation between objects. In [3], Basilico et al. designed a kernel or similarity function between user-item pairs that allows simultaneous generalization across the user and item dimensions, and presented a novel hybrid recommendation framework by merging different feature kernels as tensor production. By extending LSA or PLSA into different kinds of features, [22, 23] discuss such fusion on folksonomy. [12, 17] present fusion strategy using statistical methods, such as canonical correlation analysis (CCA), for image classification and speaker identification.

Although the aforementioned approaches take the multiple features and their correlations into consideration for similarity measure, these methods may not perform well in social media environment where the objects are with wide variety and have very high-dimensional features, e.g., up to tens of thousands. For example, [3] assumes that all feature dimensions are correlated with each other, and do not carry out any prune process, which renders it less effective in our experiments. For [22, 23], the unified feature space is often built according to the global statistical information, which incurs extremely expensive computational cost for large scale high-dimensional social media. Moreover, since the objects and their associated information in social media are with large diversity, and Web contents are evolving over time, it is nontrivial to construct meaningful latent space with lower dimensionality or find “principal components” in such dynamic environment. Therefore, the reduced dimensionality may lose some meaningful features and correlations of media objects.

3. A PROBABILISTIC MODEL FOR SOCIAL MEDIA RETRIEVAL

In this section, we introduce our approach for multimedia retrieval in social media environment, which can effectively integrate the multiple features and their correlations for similarity evaluation. We first propose a novel structure named Feature Interaction Graph (FIG) to represent multimedia data objects which can code the multiple types of features together with their correlations, then we design a probabilistic model to measure the similarity between multimedia objects. Based on this model, we further design an inverted list based index for efficient multimedia retrieval.

3.1 Problem Formulation

We first formally define our problem in social media environment. All notations defined in this subsection will be used without further explanation in the following sections. Without lose of generality, social media database can be defined as a set of multi-modal multimedia objects, i.e., \( D = \{O_i | i = 1, 2, ..., |D|\} \). For ease of presentation, we take the Flickr which is a social media site for image sharing as an example. However, our solution can be easily extended to facilitate other social media environments, such as video and music. For multimedia object in such social media system, we can extract three typical features to represent the multimedia object.

- **Textual feature**: The text information associated with the multimedia objects, such as tags, titles and comments.
- **Visual content feature**: The visual content information of the image. There are different types of low-level features can be used, such as color, texture and edge. While these features are not effective for media retrieval, we adopt middle level features “visual word” to represent the content, which are generated by clustering image blocks and can perform better than raw features as reported in [25].
- **User feature**: The user or user group related to the image, e.g., uploader, people who set the image as “favorite” or the group sharing the photo.

Given the above features, we can write a given multimodal object \( O \) as \( O = (T, V, U) \), where \( T = \{t_1, t_2, ..., t_{|T|}\} \) \( t_i \) is a textual feature, \( V = \{v_1, v_2, ..., v_{|V|}\} \) \( v_i \) is a visual feature, \( U = \{u_1, u_2, ..., u_{|U|}\} \) \( u_i \) is a user feature.

**Example 1.** For the “hamster” image shown in Figure 1. Textural feature set \( T \) includes “hamster”, “animal”, “eating”, etc; some visual content features are shown in Figure 2; and the user information can be also parsed from the Flickr Web page shown in Figure 1.

Next, we define the problem of multimedia retrieval in social media system as follows.

**Definition 1.** Multimedia Retrieval for Social Media: Given a query object \( O_q \) and an arbitrary multimodal object \( O_i \) in the database, we calculate a score \( s(O_q, O_i) \) to measure the similarity between them, rank all the objects by similarity scores and return the top listed objects as the result of retrieval.

Note that, social media retrieval is generally same as other multimedia retrieval tasks, while the key point is how to calculate the similarity score between two social media objects with rich social information. Our work is to address the problem of effective similarity measure by exploiting the multiple features as well as the interactions between different features in social media environment.

3.2 Feature Interaction Graph

As introduced previously, we can use multiple features to represent a social media object, i.e., \( O = (T, V, U) \), where \( T, V, U \) are the sets of different features respectively. However, there exist various correlations between different features, e.g., two different words are semantically similar or have frequent co-occurrence, a user’s interest is related to a special word, etc. So one open issue here is how we can represent such interaction, and use this information for similarity calculation. In this section, we describe our novel representation, named Feature Interaction Graph (FIG), to integrate features and their correlations into an undirected graph.

To build the FIG of a multimedia object, we first represent each feature as a node. Since all the nodes are isolated, we design a “virtual root” node to represent the object itself and link all the feature nodes to the root. The edges between feature node and root can be explained that the feature node is designed to represent a feature of multimedia object itself, which is a natural relationship. On the other hand, in our probabilistic model for similarity measure based on FIG, the virtual root can be replaced by other multimedia objects to compute the joint-distribution probability (to be discussed in Sec 3.3).

The next step is to draw the edges between feature nodes in FIG which is used to represent the interaction relationship between different features. In our FIG representation, two nodes are linked
Given two nodes $n_1$ and $n_2$, we use a function $Cor(\ldots)$ to measure the correlation between these nodes, and draw an edge between them if $Cor(n_1, n_2)$ is larger than a trained threshold.

- Intra-type correlation represents the relation between features with the same type. The edges between these nodes can be determined by the single-modal characters.

Edges between textual nodes: To determine whether two nodes representing textual features are related, we employ WordNet [8] based approach to determine the word similarity. WordNet is a hierarchical structure in which the relations between words are coded. There are masses of similarity measures based on WordNet, and we adopt WUP [26] to measure the correlation between two given textual nodes $n_1$ and $n_2$, i.e., $Cor(n_1, n_2) = WUP(n_1, n_2)$. Note that, we can utilize any other similarity measure such as term co-occurrence [6], as it is orthogonal to our mechanism.

Edges between visual content nodes: To determine whether two nodes representing visual features are related, we use the visual similarity between visual words corresponding to nodes. In our implementation, each visual word is a 16-D feature vector, and traditional Euclidean distance function can be deployed to measure the similarity between visual words.

Edges between user nodes: To determine whether two nodes representing user features are related, specially in this paper, the similarity between users, we employ the user groups which the user belongs to for such evaluation. In social media, each user may belong to one or more groups according to the user interests. If two users belong to the same group, two users are considered to be correlated.

- Inter-type correlation represents the relation between features with different types. This kind of correlation is very common in social media environment. For example, the word *dog* is often related to a user who is interested in *animal pictures*. For each pair of feature nodes, e.g., $n_1$ and $n_2$, they can be associated with vector $\vec{n}_1$ and $\vec{n}_2$, where each dimension of such vector represents a corresponding multimedia object, and the value on this dimension equals to the frequency of $n_1$ or $n_2$ appearing in this object. Therefore, the strength of correlation between these two feature nodes can be calculated as

$$Cor(n_1, n_2) = \frac{\vec{n}_1 \cdot \vec{n}_2^T}{|\vec{n}_1| \times |\vec{n}_2|}$$

(1)

This equation is the statistical co-occurrence correlation between two variables. If the correlation between two features is higher than a certain threshold, we add an edge between two nodes with different feature types.

Therefore, given arbitrary two nodes $n_1$ and $n_2$ in $FIG$, we can calculate the correlation value between them, and the edge between them can be determined by using the aforementioned correlation calculation methods and the trained correlation threshold. The $FIG$ representation will be used by probabilistic model for similarity measure between multimedia objects. It is clear that the strength of correlation plays an important role if we want to evaluate the similarity of different objects. This problem will be addressed in our probabilistic model for similarity calculation, while the $FIG$ representation aims at showing the possibility of relationship between features.

3.3 Probabilistic Model for Similarity Measure

The Feature Interaction Graph (FIG) can represent the features and also code the relationships between different features of multimedia object, while it does not provide a solution of using these relationships for similarity measure between different objects. In this subsection, we employ a probabilistic model based on FIG to calculate the similarity between objects.

Given a query object $O_q$, we can generate the FIG $G$. In $G$, the “virtual root” node represents the object $O_q$, and all the leaf nodes, say $\langle n_1, n_2, ..., n_m \rangle$, represent the features of the object. For an arbitrary object $O_i$ in media database, if we replace the root node $O_q$ in $G$, we can get a FIG $G'$ with “virtual root” $O_i$. If we regard each node in $G'$ as a random variable, this graph has a joint-distribution probability. Mathematically, $P(G') = P(O_i, n_1, ..., n_m)$, which defines the probability of $G'$ that $O_i$ and $\langle n_1, n_2, ..., n_m \rangle$ appear together. Because $n_i$ is decomposed from query object $O_q$, and each feature in $O_q$ has been represented as node, we can regard $O_q$ as the set of feature nodes $\langle n_1, n_2, ..., n_m \rangle$. Therefore, we can have

$$P(G') = P(O_i, n_1, ..., n_m) = P(O_q).$$

(2)

For similarity retrieval, we can calculate the probability of $P(O_q | O_q)$ as similarity score $s(O_i, O_q)$. Because $P(O_q)$ is a constant for a given query object $O_q$, we have

$$P(G') = P(O_i, O_q) = P(O_q) \times P(O_i | O_q) \propto P(O_i | O_q).$$

(3)
This indicates that the joint-distribution probability of the FIG $G'$ is proportional to $P(O_i|O_q)$, and this factors our problem of similarity measure to calculating the joint-distribution probability of $G'$.

Furthermore, since FIG structure is an undirected graph, the calculation of joint probability can be solved by treating this graph as a Markov Random Field (MRF)[13]. A Markov random field is a graphical model in which a set of random variables have a Markov property described by an undirected graph, and is commonly used in the area of statistical learning to model joint distributions. For example, in [16], Metzler et al. developed a general framework for modeling term dependencies via MRF, and further utilized it for text retrieval.

According to the MRF model, we have

$$P(G') = \frac{1}{Z_\Lambda} \prod_{c \in C(G')} \varphi(c; \Lambda)$$

$$\propto \log \frac{1}{Z_\Lambda} \prod_{c \in C(G')} \varphi(c; \Lambda)$$

$$\propto \sum_{c \in C(G')} \log \varphi(c; \Lambda);$$

where $\Lambda$ is a set of parameters in MRF model, $C(G')$ is the set of cliques in graph $G'$, each $\varphi(c; \Lambda)$ is a potential function over cliques which describes the compatibility of clique $c$ with the graph, and $Z_\Lambda$ normalizes the distribution of probability. $\Lambda$ is a set of parameters in MRF, and we represent each individual parameter in $\Lambda$ as symbol $\lambda$ with different subscripts for ease of presentation in the following.

In our situation, the term “clique” refers to the complete subgraph of FIG, and here, we constrain the definition of clique to the complete subgraph of FIG with the virtual root and at least one feature node $n_i$. Take the FIG shown in Figure 2 as example, we can have different cliques with different number of vertices, e.g., (“hamster”, O), (“animal”, User2, O) and (“hamster”, “animal”, User2, O).

The “potential function” refers to a real-value function of clique. For each clique, it returns the compatibility of the given clique, i.e., a real-value score measuring the probability that of the nodes in the given clique appear together. Therefore, this joint distribution probability of MRF relies on the structure of graph $G'$, the potential function $\varphi(c; \Lambda)$ and the parameter set $\Lambda$. For more details about joint distribution probability of MRF, interested readers can refer to [13, 16].

Furthermore, the potential function must be non-negative and is often with the formulation of

$$\varphi(c; \Lambda) = e^{\phi(c)} = e^{\lambda_c f(c)}$$

where $f(c)$ is a real-value feature function over clique $c$, and $\lambda_c$ is the weight of clique $c$. Substitute this into $P(O_i, O_q)$, we have

$$P(O_i, O_q) \propto \sum_{c \in C(G')} \lambda_c f(c)$$

$$= \sum_{c \in C(G')} \phi(c)$$

Equation 6 means the $P(O_i, O_q)$ can be calculated by the sum of $\phi(c)$ over all the cliques in $G'$. For simplicity, we also name $\phi(c)$ as the potential function of clique $c$. To use this probabilistic model for similarity measure, we first create the FIG graph for each query object and database object pair, e.g., $G'$ for $O_q$ and $O_i$. For each clique in this graph, we calculate the joint distribution probability using defined feature function $f(c)$ and trained parameter $\lambda_c \in \Lambda$. Finally, we calculate the joint distribution probability of $G'$ as the similarity score of the query-object pair, and evaluate all the objects in the database for multimedia retrieval.

How intuitively can this mechanism code the feature correlation between different features for similarity measure? Here we present an intuitive explanation. For example, if we draw an edge between word $t \in T$ and user $u \in U$ to indicate that there exists a relation between $t$ and $u$ in query object $O_q$, the graph must have a clique $c$, such that $t \in c$ and $u \in c$. When calculating the score of $c$, we need to take the influence made by both features, i.e., $t$ and $u$, into effect. $c$ can contribute to the similarity between $O_q$ and $O_i$ if $t$ and $u$ also appear in $O_i$. Therefore, by appropriately defining the potential function $\phi(c; \Lambda)$, we can expect these features can interactively influence the overall similarity score.

### 3.4 Potential Function Design

The definition of potential function is the key factor in the probabilistic model, because it determines whether the probability can effectively measure the similarity between different objects. After building the FIG as discussed in the above section, we need to define the potential function for a given clique $c$. Here the clique $c = \{n_1, ..., n_{|c|-1}, O_i\}$, where $n_i$ refers to the nodes decomposed from query object $O_q$, and $O_i$ represents the object itself.

As we introduced previously, the “potential function” is used to measure the probability of the clique appearing in a multimedia object, i.e., the similarity between the features in the clique to the object. Therefore, we need not only consider the features contained by the clique, but also the correlation between clique features and other features in the object. Here we present our definition enhanced with smoothing factor.

$$\phi(c) = \frac{\lambda_c P(n_1, ..., n_{|c|-1}|O_i)}{Z_{O_i}}$$

$$= \frac{\lambda_c \times \frac{freq(n_1, ..., n_{|c|-1}|O_i)}{|O_i|} + \sum_{n_j \in \{O_i\} - c} Cor(n_i, n_j)}{(1 - \alpha) \sum_{n_j \in \{O_i\} - c} Cor(n_i, n_j)}$$

where $c$ is the given clique $c$, $\lambda_c$ is the weight of this clique will be learned as one parameter of MRF model, $P(n_1, ..., n_{|c|-1}|O_i)$ is to estimate the probability that feature nodes $(n_1, ..., n_{|c|-1})$ in the clique appear together in $O_i$. It is common in social media that the features in the clique may be also similar to some other features in $O_i$, and hence we integrate the smoothing factor into the potential function. Therefore the $P(n_1, ..., n_{|c|-1}|O_i)$ consists of two components. The first part is the $freq(.)$ which refers to the appearance frequency of features $(n_1, ..., n_{|c|-1})$ in $O_i$. And the smoothing part computes the contribution of the rest features in $O_i$, i.e., $(O_i) - c$, by evaluating their correlations with the clique. Note that, the function $Cor(.)$ used to evaluate the correlation between two features may vary from different types of features, e.g., inter-type and intra-type, which has been described in Sec 3.2. The smoothing parameter $\alpha$ can trade-off the effect of two components.

Equation 7 only considers the similarity between the features in the clique and object, but does not take the weight of edge, i.e., the strength of correlation between features, into account. However, the clique itself may have different importance, as the tight connection between nodes in a clique usually yields more semantic information. To capture this information, we employ a correlation matrix like method to calculate the correlation strength $CorS$ be-
between features in the cliques by exploiting the statistical information of media corpus.

\[
CorS(n_1, ..., n_{|d|-1}) = \frac{|D| \sum_{i=1}^{|d|-1} \prod_{j=1}^{|D|} \frac{n_{|d,1|} - \overline{n}_j}{\text{var}(n_j)}}
\]

where \(|D|\) is the cardinality of database, \(n_{ij}\) is the frequency of feature \(j\) in object \(i\), \(\overline{n}_j\) is the average frequency of feature \(j\), and \(\text{var}(.)\) is variance function. Note that, when the number of features is equal to 2, this equation is equivalent to the so-called covariance between features \(n_1\) and \(n_2\), and here we expand it to multiple features. Integrating this information in the definition of \(\phi(c)\) in Equation 7, we can get,

\[
\phi'(c) = CorS(n_1, ..., n_{|d|-1}) \times \phi(c)
\]

Note that, the above definition is equivalent to attaching an extra weight \(CorS(n_1, ..., n_{|d|-1})\) to \(\phi(c)\) we have defined. The larger the \(CorS(n_1, ..., n_{|d|-1})\) is, more important the clique \(c\) is.

On the other hand, as defined in this model, the parameter \(\lambda_c\) can vary for different \(c\). However, the number of cliques can be very large in social media environment because of high-dimensionality of multiple features and their rich correlations. Therefore, training \(\lambda_c\) of MRF for each individual clique is nearly infeasible in reality due to the computation limitation, and \(\lambda_c\) is generally employed with some constraints in the literature. For example, [16] only trains \(\lambda\) of MRF for three patterns of dependencies. In this work, we constrain the parameter only related to the number of elements in a given clique \(c\). In this circumstance, \(\lambda_c\) is trained to code the relative importance between cliques with different number of elements, i.e., \(|c|\), while \(CorS(n_1, ..., n_{|d|-1})\) is deployed to code the importance of different cliques. Both together provide a comprehensive weight of given clique \(c\). Note that, the \(\lambda_c\) can be trained to involve both factors for each individual clique. However, due to the large number of different cliques, our constrain based on \(CorS(n_1, ..., n_{|d|-1})\) and \(\lambda_c\) can significantly decrease the hypothesis space of MRF needed to be optimized.

3.5 Retrieval Algorithm

Given the probabilistic model on FIG, we are able to calculate the similarity score between two objects, which can be used to conduct similarity search in multimedia databases. The overall retrieval process consists of two major stages, i.e., training/preprocessing and retrieval.

In the first stage, we need to generate the FIG representation for all the data objects in the database, train the parameter set \(\Lambda\), i.e., the \(\lambda_c\) related to cliques, and the correlation of features in each clique according to the Equation 8. To generate the FIG, we first construct 6 pair-wise feature correlation tables, i.e., \(T \times T\), \(V \times V\), \(U \times U\), \(T \times V\), \(T \times U\), \(V \times U\), according to the mechanisms introduced in Sec 3.2. Take the textual feature correlation table and the tags in Figure 2 as example, in \(T \times T\), we have \(Cor(\text{“animal”}, \text{“animal”}) = 1, Cor(\text{“hamster”}, \text{“animal”}) = 0.73, Cor(\text{“hamster”}, \text{“vegetable”}) = 0.27\), and so on. By evaluating the correlation of the features of every object with the correlation tables, we can construct FIG for each of them. With the FIGs of multimedia corpus, we train the parameter set \(\Lambda\) for MRF model adopting the training strategy presented in [16].

After the preprocessing and training, the multimedia retrieval task can be conducted by sequentially comparing the query object with the objects in the database. Given a query \(Q_q = \{T_q, V_q, U_q\}\), we first convert it to FIG representation with the help of feature correlation tables, compute the similarity score with each object using MRF model, and return the top ranked list to the user.

Clearly, the sequential comparison method is computationally expensive, and hence we design an inverted list of cliques as index structure to accelerate the retrieval processing. For each clique, we store the correlation strength \(CorS\) of features in the clique and the objects which contain this clique. With this index structure, we find the objects from the database which share some same cliques as the query object, and compute the similarity score to get the rank list. The framework of our social media retrieval approach is presented in Figure 3. In the training and preprocessing stage, we extract multiple features from the objects in the database, generate correlation tables for different types of features, convert the objects to FIG representation, and construct the inverted list index on cliques. For each query object, we first represent it as FIG, access the index structure to compute the similarity with objects in the database, and finally return the top-k results to the user.

The detailed retrieval algorithm is presented in Algorithm 1. Given a query object, we first construct the FIG presentation of the media features FIG_q and extract the cliques from this graph (Lines...
4-5). For each clique \( c_i \), we access the inverted index to find all the objects in the database containing the clique. For each object in the candidate list \( CandidateSet_i \), we compute the score with the potential function Equation 9, and rank the \( CandidateSet_i \) accordingly (Lines 8-11). After all the cliques are processed, we merge and rank the objects from all the \( CandidateSet \). Some efficient approaches for merging different result lists can be applied here, and we adopt Threshold Algorithm [7] in our implementation. The Threshold Algorithm is the most well-known instance due to its simplicity and memory requirements, which is based on an early-termination condition and can evaluate top-k queries without examining all the tuples [4]. Finally, the top-k similar objects in the \( ResultList \) are returned to the user.

**Algorithm 1 Retrieval Algorithm**

1: Input: Query \( O_q \), Database \( D \), Parameter \( \lambda \); Inverted list on cliques.
2: Output: top-k most relevant objects in \( D \).
3: **procedure** SEARCH
4: \( FIG_Q \leftarrow \) graph of \( O_q \);
5: \( CliqueSet \leftarrow \) Set of Cliques in \( FIG_Q \);
6: for \( c_i \in CliqueSet \) do
7: \( CandidateSet_i \leftarrow \) \( InvList(c_i) \);
8: for \( O_j \in CandidateSet_i \) do
9: \( CandidateSet_i[O_j].score \leftarrow \phi(c_i, O_j) \);
10: end for
11: \( CandidateSet_i \leftarrow \) ranked \( CandidateSet_i \) based on scores;
12: end for
13: \( ResultList \leftarrow \) merge and rank the objects in \( CandidateSet_i \) with Threshold Algorithm;
14: Return top-k objects;
15: **end procedure**

4. ENHANCE THE PROBABILISTIC MODEL FOR MEDIA RECOMMENDATION

In this section, we propose to extending the probabilistic model to facilitate the recommendation application in social media environment. Specially, given a user with his history of upload/favorite multimedia data in social media system, we recommend the newly posted multimedia objects that may potentially interest him based on the analysis of his historical behavior. The problem of multimedia recommendation can be defined as follows:

**Definition 2.** Multimedia Object Recommendation for Social Media: Given a user \( u \), we can associate \( u \) with a set of his favorite or uploaded multimedia objects according to his historical behavior, say \( H_u = \{ O_{u1}, O_{u2}, ..., O_{uh} \} \). Given an object \( O_i \) in newly incoming set, we calculate similarity score \( s(H_u, O_i) \) to measure the similarity between them. The most similar objects are recommended to the user.

There exist different kinds of recommendation techniques [1, 5, 10, 14, 15] for various user tasks. According to the survey report on recommender systems, recommendation techniques have a number of possible classifications, such as collaborative, content-based, demographic, etc [5]. Our definition is a variant of approaches so called content/similarity-based technique, i.e., comparing the new multimedia object with the user profile constructed with the user’s historical favorite objects.

For multimedia recommendation, one potential solution is to simply view \( H_u \) as a “big” multimedia object, \( H_u = (\cup T_j, \cup V_j, \cup U_j) \). That is, we union the features of objects related to this user. Thus, the recommendation task can be considered as the same as retrieval operation. However, directly applying the retrieval model may suffer from two problems:

- By regarding \( H_u \) as a “big” object, we may introduce noisy information. As the object set of a certain user may have many different multimedia objects, the “big” object will have much more features than an individual object. The probabilistic model tries to construct the connections of the features in this “big” object, and hence features from different objects may be connected and form unexpected cliques in \( FIG \), which will deteriorate the similarity score computation.

- Another problem is about temporal effect. The user interest generally varies from time to time, and taking \( H_c \) as a “big” object simply ignores this information. For example, a user is interested in “Obama” only during the period of US president election, and later recommendation may not be attractive. Therefore, the temporal information is essential to properly model the user’s up-to-date interest.

Figure 4 shows an example of a user’s historical “favorite” images in our dataset crawled from Flickr, and some sample tags associated with images are also demonstrated. We can clearly see the temporal change of the image topics, which reflects the evolution of user interest and web content. In this example, the images about cosmetic and fashion are the user’s common interest. While during the period of US president election in 2008, he/she is also interested in the images related to Obama. Therefore, the “big” image
users may have a wide variety, and simply combining their features and correlations may degrade the performance.

To solve the above problems, we enhance the probabilistic model of retrieval for recommendation task, which takes such concerns into consideration. First, when regarding $H_u$ as a “big” object, we differentiate the objects in $H_u$, and impose this constraint to construct its $FIG$, i.e., we only connect the feature nodes from each individual object. Thus the $FIG$ can avoid some noisy edges.

For the temporal issue, we solve it by defining a novel function $\phi_{rec}(\cdot)$ for a given clique $c$ by integrating the temporal information. To define this function, we first associate the appearance of clique with a time stamp. Specially, given a clique $c = \{n_1, \ldots, n_{|c|-1}, O_t\}$ in $H_u$, we assign a time stamp $t$ to it, because we always try to match the object with user’s current interest for recommendation task in social media. In our implementation, all time stamps are determined in the basis of month, however different durations can be also applied with minor modification.

The potential function $\phi_{rec}$ for media recommendation can be defined as follow:

$$\phi_{rec}(c_t) = \lambda_{c_t} \times \delta^{-t_t} \times CorS(n_1, \ldots, n_{|c_t|-1}) \times P(n_1, \ldots, n_{|c_t|-1}|O_t)$$  

(10)

where $t_t$ is the time stamp of clique $c_t$ in $H_u$, $CorS(\cdot)$ and $P(\cdot)$ are the functions defined in Equations 8 and 7, and $\delta < 1$ is a decaying parameter to tune the importance of cliques with different time stamps. The decaying parameter means that the smaller time decaying parameter to tune the importance of cliques with different formation. To define this function, we first associate the appearance of clique with a time stamp. Specially, given a clique $c = \{n_1, \ldots, n_{|c|-1}, O_t\}$ in $H_u$, we assign a time stamp $t$ to it, because we always try to match the object with user’s current interest for recommendation task in social media. In our implementation, all time stamps are determined in the basis of month, however different durations can be also applied with minor modification.

The algorithm for recommendation is similar to that of retrieval, and we shall only brief some differences as following. First, we combine the objects related to a certain user as a “big” object $H_u$, while only connecting the features from each individual object with edges to construct the $FIG$ representation. Second, for a set of new multimedia objects, we evaluate the similarity of objects with the user graph $FIG$, and recommend the top objects to the user. In this step, the potential function enhanced with temporal information will be utilized.

5. EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to evaluate the performance of the proposed approach for social media applications and demonstrate the superiority of our method by comparing with other competitive techniques. All our experiments are conducted on a PC running Ubuntu Linux with 2.4 GHz CPU and 3G memory.

5.1 Experimental Setup

5.1.1 State-of-the-art techniques

In this experimental study, we compare our approach, named FIG, with some state-of-the-art feature fusion strategies [22, 20, 3, 23] for social media retrieval and recommendation on real-world data corpus collected from Flickr. The competitors fall in two general categories, i.e., early fusion and late fusion, and our approach can be considered as a variant in early fusion category since our approach tries to directly integrate the multiple features and their correlations for similarity measure.

• Early Fusion: We compare our method with early fusion methods based on statistical analysis on data corpus [22, 23], specially, which uses Latent Semantic Analysis to convert the multiple features to a unified latent space. We name this approach as LSA. We also compare with the approach proposed in [3] which employs Tensor Production as fusion method, and we name this approach as TP.

• Late Fusion: We compare our approach with late fusion based methods which generate the candidate result lists for different features and combine these lists for final results. One of the latest work [21] investigated different late fusion mechanisms, e.g., Calibrated Score Averaging and RankBoost [9], and we select RankBoost based approach to combine information sources in this work, which is named as RB.

5.1.2 Data preparation

Since no benchmark social media data is available for performance evaluation, we collect the real-life dataset from Flickr which is a popular social media site for image sharing. Additionally, because the requirement is different from multimedia retrieval and recommendation applications, i.e., the retrieval task is image oriented and the recommendation is user oriented, we adopt different crawling strategies for data collection, and represent these two datasets as $D_{rec}$ and $D_{ret}$ respectively.

• Dataset $D_{ret}$ for Retrieval: For the dataset used for retrieval, we collect the most “interesting” images from Flickr on daily basis. “interesting” score is defined by Flickr, which demonstrates the popularity of the image. The reason we select the “interesting” images is that such images are generally associated with rich social information, e.g., more tags and interested users. These information can facilitate the evaluation of effect on the feature interaction in social media environment. On the other hand, these images have a wide variety which can guarantee the unbiased evaluation. We crawled totally 236,600 images uploaded from 2008.1 to 2008.6, and for each day, 1,300 images are downloaded due to the constraint of Flickr. Furthermore, we use Flickr API to collect the social information associated to these pictures, including tags and users. The overall information forms the dataset for social media retrieval evaluation.

• Dataset $D_{rec}$ for Recommendation:

The user information is the key factor for recommendation task in social media environment. Different from retrieval evaluation, where the tester can help evaluate whether an image is similar to the query image, we cannot judge whether the image recommended is the favorite of the user. The Flickr system allows user to label his interested image as his “favorite”, and we can utilize this Flickr function for recommendation evaluation, i.e., the image in the “favorite” list is the correct recommendation. Although this evaluation is too strict, as the user may be also interested in other images which are not in his “favorite” list, this is a fair evaluation to compare the different recommendation approaches. Based on the above considerations, we first initialize a user on set on 2008.1.1 and then use Flickr API to download the images they labeled as “favorite” from 2008.1 to 2008.6. We then eliminate the users who have favorite images less than 100 and larger than 1,000 during 2008.1 to 2008.3. 279 users and totally 207,909 image objects are left. We use images

http://www.flickr.com/services/api/
from 2008.1 to 2008.3 to model the users’ interests, and the rest images for recommendation evaluation.

5.1.3 Feature extraction

For performance study, we need to extract different types of features from multimedia objects. In this work, we extract three types of features from the social images in Flickr, i.e., textual feature, visual content feature and user information, which are described as follows:

- **Textual Feature**: We utilize the tags associated to the images in the dataset as the textual feature. A WordNet stemmer is used to do stemming, and a snowball stop word list is used to eliminate stop words [8]. Because tags are free style words in social media environment, the number of tags can be very large. Therefore, we eliminate those tags with frequency less than 5 in the whole corpus, which are generally noise or typo according to the previous study. The final dimensionality of textual feature is about 60,000.

- **Visual Feature**: Visual words are extracted to be used as content features which yield better performance than raw visual features, such as color or texture as reported in [25]. We first divide each image into uniformly distributed equal-size blocks (16*16 pixels in our approach). Then raw visual features are extracted for each block, and converted to 1022 visual words by k-means clustering [25]. For each image, we use a group of visual words contained in the image to represent the visual content information.

- **User Feature**: In this work, we use the users who uploaded image or labeled the image as “favorite” as the user feature which consists of 273340 users totally. The groups that they belong to are used to determine the correlation between the users.

Note that, the social media systems contain more information such as title, comments, user who posted comments, etc, which can be also used for media retrieval and recommendation. However, we believe the aforementioned features are sufficient for performance comparison purpose.

5.1.4 Evaluation metric

For social media retrieval evaluation, due to the absence of golden standard set, we use Precision@N to measure the performance of different algorithms, which is the percentage of top-N answers retrieved that are correct. Specially, given a query object, we find the top-N images with the highest similarity, and ask three evaluators to judge the relevance between query and returned images. 20 randomly selected images are used as query, and the average Precision@N is used to evaluate the retrieval performance. We also report the result on efficiency which is the average response time for each query. In this paper, we omit the performance on storage cost for different approaches due to the space constraint, as the storage costs for different approaches have no significant difference, and storage cost is a less critical issue compared with efficiency and effectiveness given the massive storage available nowadays.

For recommendation task, [10] has given a comprehensive study on the metrics used to evaluate the recommendation performance, and classified recommendation accuracy metrics into three classes: predictive accuracy metrics, classification accuracy metrics, and rank accuracy metrics. However, there is no standardized metric due to the large diversity of user tasks and data features, e.g., restaurants are in different locations while movies have rating information. According to the dataset from Flickr, we believe classification accuracy metric is an appropriate choice for our evaluation which measures the frequency with which a recommender system makes correct or incorrect decisions about whether an item is good. Specifically, we adopt the popular metric Precision@N to measure the percentage of “favorite” images in the top-N recommended images for a certain user.

5.2 Performance on retrieval

In the first set of experiments, we study the performance of different approaches on social media retrieval. The parameter λ, of probabilistic model in our approach is critical issue for the system performance. Different training mechanisms have been discussed [16, 19], in this work, we simply adopt the method proposed in [16] which yields better performance as reported, and omit the detailed discussion.

5.2.1 Effect of different type combinations

As we introduced previously, the social media is different from traditional multimedia in terms that the social media consists of multiple types of features and these features are interacted. Therefore, we first evaluate the effect of different features and their combinations.

![Figure 5: Retrieval Performance with Varied Feature Combinations](image)

Figure 5 shows the performance of individual features and various feature combinations in our FIG model. Among three individual features, i.e., text, visual and user, the visual feature yields worst performance. There exists semantic gap between visual feature and image content, thus direct application of visual feature cannot achieve satisfactory results. Our finding is consistent with other work on multimedia retrieval [27]. The text and user features are better choice than visual feature, and textual feature performs slightly better than user information. The textual feature can effectively represent the semantic information of multimedia objects, and hence can be utilized to measure the similarity between multimedia objects. The user information is also valuable in social media environment, as the user’s interest is generally limited and users in a certain group share the similar interest.

On the other hand, the combination of multiple features generally performs better than individual type of feature, as they can better capture the similarity between multimedia objects. The involvement of visual feature is not so helpful as textual and user features, because visual feature may also input noisy information due to the low precision of visual effect. Overall, the combination of all three features provides the most satisfactory result, the FIG
representation integrates these features and their correlations for social media retrieval, which can effectively evaluate the similarity between objects.

![Query](image1)

**Figure 6: An Example of Query Result by FIG**

Figure 6 illustrates an example of query result by our approach, i.e., the query image and 4 sample results returned by our FIG method, to show the functionality of multiple features and their interactions. All the four returned images have similar semantic content with the query image. Besides the similar visual content, the first three objects have some same tags, users (IDs), or both of them as the query object. This example clearly demonstrates the capability of our approach to exploit various features and their correlations for effective social media retrieval.

### 5.2.2 Retrieval precision

In this section, we present the experimental results to verify the effectiveness of our approach FIG compared other three approaches with different feature fusion strategies, named LSA, TP and RB respectively.

![Figure 7](image2)

**Figure 7: Retrieval Performance with Varied N**

Figure 7 shows the precision with varied N for different approaches. Our proposed FIG method yields better performance than other competitors. The probabilistic model can effectively code the detailed feature interactions into the similarity measure.

The FIG representation represents the features and their correlations with cliques, and also takes the correlation strength of features within the clique into consideration. Therefore, our approach can better model the similarity between social media objects. Overall, the late fusion based method RB performs as good as early fusion based approaches, and even better than the TP variant in our evaluation. The disadvantage of late fusion strategies is not significant as reported in the literature, such as [17], although there exists rich correlation between original features in social media environment. The reason is that the performances based on individual features, such as textual and user information, are not so depressed in social media environment, due to the rich semantic information they contain. Therefore, late fusion from such “good” candidate lists can still provide satisfactory results.

We next to evaluate the precision performance of different approaches in terms of the scalability by varying the data size. Scalability is a critical issue for social media retrieval, as social media database is typically very large. In this experiment, we randomly split the database with different sizes, and Figure 8 shows the precision@10 results of different approaches by varying the data size from 50K to 236K. When the data size increases, the performances of all competitors improve. This can be explained that the large dataset contains more similar images to the query image. Note that, we do not conduct evaluation on “precision-recall”, because the datasets are real-life data crawled from Flickr, and it is infeasible for evaluators to annotate all the relevant images in such large dataset for 20 different queries. However, the figure is sufficient to demonstrate the advantage of our approach on scalability, i.e., the proposed method can effectively find well-matched images from a large database.

![Figure 8](image3)

**Figure 8: Retrieval Performance with Different Data Size**

### 5.2.3 Time efficiency

Figure 9 illustrates the time cost per query in retrieval task with varied scale of datasets from 50K to 236K. All the approaches can return the query results in less than 0.6 second. We can see that when the scale of dataset increases, the time cost increases accordingly. Among these approaches, early fusion based approaches, i.e., TP and LSA, perform better than RB and our proposed FIG method. The early fusion methods generally map the original multiple features into a unified latent space with the reduced dimensionality, which decreases the computational cost for retrieval processing. While the late fusion method has to merge multiple result lists for final answer generation, which makes it less efficient than early fusion methods. Although our approach performs worst among all the competitors, the differences are not significant. In our approach, we try to integrate more valuable information about
feature interactions for similarity measure, and the probabilistic model shows better performance as demonstrated in the former study, although it introduces relatively high time cost. We believe this trade-off is worthwhile in social media application. Note that, the time efficiency can be potentially increased by deploying parallel algorithms and distributed architectures, however such discussion is beyond our work in this paper.

5.3 Performance on recommendation

Having evaluated our approach on social media retrieval, we proceed to examine its performance for media recommendation task. All the competitors employ similarity-based strategy by comparing the new multimedia object with the user profile constructed with the user’s historical favorite objects. Specifically, the popular metric Precision@N is used to measure the percentage of “favorite” images in the N recommended images for a certain user in Flickr.

5.3.1 Effect of delaying parameter

In our approach for social media recommendation, we integrate the temporal information into the probabilistic model, i.e., the time-stamped clique construction and decaying parameter which tunes the importance of images in the user’s favorite list, while three other competitors do not take the temporal issue into consideration. Therefore, we first investigate the effect of the decaying parameter for recommendation task.

Figure 10 shows the Precision@10 of recommendation with varied decaying parameter δ. If δ is equal to 1, it means we do not differentiate the images of a user when we construct the user profile; while the smaller value of δ means we give higher weight to the recent “favorite” images. In our experiment, we set the time window as one month and the value of δ means that for two consecutive time windows, the importance of images in former window is δ times of image importance in the later window.

We can see that when the value of δ decreases, i.e., we give higher weight to recent “favorite”, the proposed FIG approach yields better performance, which is consistent with our expectation. When we decrease the δ from 1 to 0.4, the Precision@10 increases from 39.8% to 42.1%. As we introduced previously, the user interest generally evolves in social media environment, and hence the new image which is similar to user’s up-to-date favorites has higher probability to be his favorite. However the further decrease of δ slightly degrades the performance of FIG, as too small δ may de-validate the effect of early images, which can still contribute to model the overall user interest. Another thing worth noting is the effects of user and textual features are contrary to that in retrieval evaluation as shown in Figure 5, where the textual feature performs better than user information. The reason is that the recommendation task is user oriented, and hence the user information is more crucial for “favorite” image judgement.

5.3.2 Comparison with other approaches

We next compare our approach with other three methods for social media recommendation. Note that, since we utilize similarity-based approach for recommendation task, the retrieval algorithms of these approaches can be used only with minor modification. We present two variants of our approach for this comparison, i.e., FIG and FIG-T which integrates the temporal information.

Figure 11 shows the precision of recommendation with varied N for different methods. The figure clearly shows the advantage of our approaches over the three other methods. The FIG method is about 15% better than other approaches averagely, because our probabilistic model can effectively measure the similarity between multimedia objects, i.e., the recommended object and the user profile in this circumstance. The performance of FIG-T further increases 5% more than that of FIG as the time parameterized enhancement can better model the user interest in the dynamic social media environment.

6. CONCLUSION

In this paper, we have presented the novel probabilistic model for fusing the multiple features of multimedia objects for social media applications. We first modeled a multimedia object as a graph re-
representation FIG which can code multiple features and the correlations among them. Using this model, we introduced a novel probabilistic model based on MRF to effectively measure the similarity between different objects. We further presented a detailed algorithm for social media retrieval and recommendation. Our experimental results demonstrate the advantage of our approach over existing techniques on large real-life datasets downloaded from Flickr.

Several interesting directions for future work exist. Parameter tuning is a critical issue that affects the overall performance of a probabilistic model. Thus, it should be interesting to investigate the means to determine appropriate parameters for dynamic social media data of wide diversity. We also plan to extend the current techniques to integrate more social information and support other social media applications and environments.

7. ACKNOWLEDGMENTS

This research was supported by the National Natural Science foundation of China under Grant No. 60933004 and 60811120098.

8. REFERENCES


