Event-Based Media Enrichment using an Adaptive Probabilistic Hypergraph Model

Xueliang Liu, Meng Wang, Bao-Cai Yin, Benoit Huet, Xuelong Li, *Fellow, IEEE*

***Abstract*—Nowadays, with the continual development of digital capture technologies and social media services, a vast number of media documents are captured and shared online to help attendees record their experience during events. In this paper, we present a method combining semantic inference and multimodal analysis for automatically finding media content to illustrate events using an adaptive probabilistic hypergraph model. In this model, media items are taken as vertices in the weighted hypergraph and the task of enriching media to illustrate events is formulated as a ranking problem. In our method, each hyperedge is constructed using the K-nearest neighbors of a given media document. We also employ a probabilistic representation, which assigns each vertex to a hyperedge in a probabilistic way, to further exploit the correlation among media data. Furthermore, we optimize the hypergraph weights in a regularization framework, which is solved as a second-order cone problem. The approach is initiated by seed media and then used to rank the media documents using a transductive inference process. The results obtained from validating the approach on an event dataset collected from EventMedia demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Event enrichment, hypergraph, transductive learning.**

I. Introduction

In recent years, we have witnessed a growth in the popularity of social media websites, such as Flickr, YouTube, and Facebook. These social media sites provide an interactive sharing platform where vast amounts of unstructured data are uploaded every minute. How we can benefit from such rich media is still an open and challenging problem.

Events are a natural way of referring to any observable occurrence grouping persons, places, times, and activities that can be described [37]. Events are also observable experiences that are more and more frequently documented by people through different media (e.g., videos and photos). To help users grasp events effectively, various event browsing and searching platforms have been built, which have benefited greatly from social media event content, e.g., eventful.com, upcoming.org, last.fm, and Facebook.com/events, to name but a few. These services sometimes have an explicit connection with media sharing platforms. Often there is overlap in terms of coverage of upcoming events. Moreover, they provide social network features to support users in sharing and deciding upon attending events. However, in these Web services, less attention is paid to improving the end-user experience when searching and browsing content, while the functionality of finding target media content to provide vivid information on given events is still missing.

In fact, automatically associating social media content with known events is a challenging problem owing to the heterogeneous and noisy nature of the data. In recent years, several works have been proposed to investigate searching event related media data. For example, Trad et al. [31] proposed a visual-based method for retrieving events in photo collections, although textual feature, which is very useful in describing media content, has not been investigated. Liu et al. [18] developed a framework to associate social media documents with events in two steps: first, time, location, and textual features are extracted via a query of the media candidates on a given event, and then to improve the performance, a visual filter is created to remove noisy data. Obviously, it is not sufficient to model only the relations among images and events. The social media content could be represented by multimodal features, such as title, description, capture time, location, and so on. Most existing methods explore these multimodal features of media data separately or sequentially. Moreover, owing to the lack of comprehensive analysis on multi-facets of social data, these methods have a marginal effect on modeling the relation between social multimedia and events.

In machine learning, employing hypergraph model is a natural way of formulating this kind of complex relation and fully exploiting the multimodal features among social media documents [11], [42]. Hypergraph is a generalization of the graph model in which an edge can connect a number of vertices. Each social media document, represented by spatial, temporal, visual, and textual features, is a vertex in a hypergraph. The hypergraph is then constructed using the K-nearest neighbor method. In other words, for each vertex, its K-nearest neighbors, measured by the similarity of multiple features, are used to generate a hyperedge.

Although the hypergraph is promising and desirable, there are still many challenges to overcome. First, the traditional hypergraph model assigns a vertex to a hyperedge by binary decision, thereby ignoring the diverse information of the different media content. Second, it is hard to find the optimized "K" when constructing the hypergraph model, which plays a key role in building a robust model. Finally, it should be noted

X. Liu and M. Wang is with School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China. E-mail: eric.mengwang@gmail.com.

B.-C. Yin is with the School of Transportation, Beijing University of Technology, Beijing 100124, China.

B. Huet is with department of Multimedia, EURECOM, sophia-antipolis, 06904, France. E-mail: huet@eurecom.fr.

X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shanxi, P. R. China. Email: xuelong_li@opt.ac.cn.

that the different modalities have different effects on modeling the relation between media data and events. In addition, an investigation of the weights of different modalities could further improve the performance of the proposed solution.

To solve these problems, we propose an adaptive probabilistic hypergraph learning method to find media content relevant to a given event. In the proposed method, for each modality, hyperedges are generated by the nearest neighbor method and represented in a probabilistic way; then, all the hyperedges are aggregated as a unified hypergraph model. In addition, we also optimize the hypergraph weights using a regularization framework, which further exploits the correlation among media data. An overview of our proposed approach is illustrated in Figure 1. The contribution of this paper is twofold:

- We study the event-based media enrichment task as a ranking problem, and solve it using hypergraph modeling.
- To find as much media content as possible for a given event, we propose an adaptive probabilistic hypergraph method to rank the content. In this method, a probabilistic incidence matrix is employed to construct the hypergraph, while an alternative optimization method is used to optimize the ranking scores and hypergraph weights simultaneously.

The remainder of this paper is structured as follows. First, we review some related work in Section II. We then explain our approach for associating media with events in Section III and discuss our results in Section V. Finally, our conclusions are presented in Section VI.

II. Related Work

In this section, we briefly introduce related work on social event illustration/detection and hypergraph modeling.

A. Social Event Analysis

In recent years, research on how to better support the end-user experience when searching and browsing multimedia content has drawn much attention [16] [19]. It is well known that vivid photos attract human attention more than textual descriptions. The authors in [7] aimed to improve users' attention when reading news articles by illustrating the story and proposed a system to realize this. The application provides mechanisms to select the best illustration for each scene automatically and a set of illustrations to improve the story sequence. In [13], an unsupervised approach was proposed to describe stories with automatically collected pictures. In this approach, semantic keywords are extracted from the story and used to search an annotated image database. Then, a novel image-ranking scheme automatically chooses the most appropriate images. A text-to-picture system that synthesizes a picture from natural language text without limitations, is presented in [43]. The system first identified "picturable" textual units through natural language processing, then searched for the most likely image parts based on the text, and finally optimized the picture layout based on both the textual and image parts. Besides the works that illustrate text with photos, some studies have also been carried out to generate video representation from textual content. For example, the system presented in [25] creates a visual representation of a given short text. In this system, the authors used a variety of techniques to query images using a given text string with the novelty being that the final images are selected in a user-assisted process and automatically used to create a storyboard animation. All of these approaches and systems studied ways of demonstrating textual content using multimedia data. In [40], the authors investigated the density-based clustering algorithm and proposed a scalable distance-based clustering technique for Web opinion clustering to discover ongoing topics of interest and understand how topics evolve together with the underlying social interaction between participants. The authors in [3] proposed a system to detect emerging topics from social streams and illustrate the topics with corresponding information in multiple modalities. The authors in [4] followed a very similar approach, exploiting the rich "context" associated with social media content and applied clustering algorithms to identify social events. In [10], the authors studied the problem of browsing and organization of picture collections in a natural way, by events, and proposed a method to classify Flickr images into different event categories. A demonstration of categorizing photos by events/subevents through visual content analysis is presented in [21]. In [9], Twitter messages corresponding to large scale media events were investigated to improve event reasoning, visualization, and analytics, while other research has been carried out to find events directly from Twitter posts [36], [24]. In [36], the authors studied how to employ a wavelet-based techniques to detect events from Twitter streams. A similar method was reported in [6] to detect events from Flickr time series data. In [24], the authors investigated how to filter tweets to detect seismic activity as it occurs. A new scheme was proposed in [41] to discover and track spatiotemporal patterns in noisy sequences, while in [31], a method was introduced to retrieve event-related photos in collections.

Previous work suggests that fusing multimodal features could improve system performance [14], [27], [30], [26]. The scheme presented in this work attempts to enrich a set of images/videos to illustrate social events by matching concert events with photos based on different modalities, such as text (tags), time, and geo-location, to produce an enriched photo set that better illustrates events. A similar work, presented in [9], proposed a strategy for extracting valuable information from the overwhelming amount of social media content on a variety of broadcast news. However, this work focused on filtering noisy information and producing a summary, whereas illustrating events with different media addresses the problem of how to leverage vivid multimodal content to share experiences.

B. Hypergraph Modeling

In machine learning, the graph is a fundamental tool for modeling pairwise relationships among objects and solving many tasks like classification [29], [35], ranking [1], [34], [28], and clustering [14], [23]. For example, in a social network, the relations of different people can be formulated using a graph model, where the vertices and edges represent people and their relations, respectively. However, graph-based models do not handle heterogeneous data well.

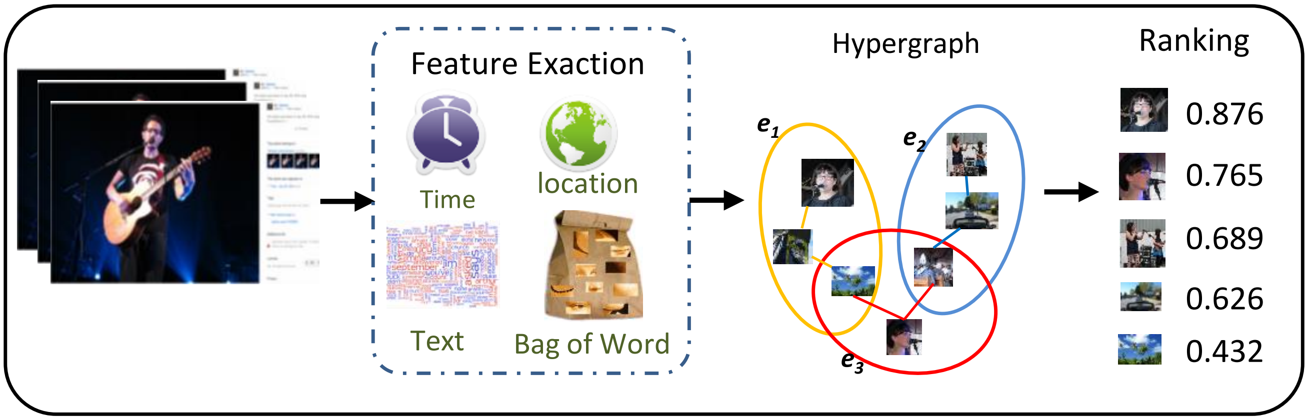

Fig. 1: Overview of the proposed method.

As a natural extension of the graph-based model, the hypergraph has been employed as a useful tool for modeling multimodal and high-order data and analyzing the structure of a system [15], [33].

In [11], a probabilistic hypergraph learning framework for image retrieval is presented. In this approach, images are taken as vertices in a weighted hypergraph and the task of searching for images is formulated as a hypergraph ranking problem. In [38], Wong et al. proposed a hypergraph-based 3D object description method, in which the vertices denote the surface patches of an object in a computer-aided design system and the hyperedges represent connections between pairs of boundary segments. A class-specific hypergraph was proposed in [39] to explore both scale-invariant-feature-transform (SIFT) features and global geometric constraints for object recognition, in which the vertices of the constructed hypergraph represent images belonging to an object category.

These works have demonstrated the effectiveness of the hypergraph model in formulating higher-order relationships. Inspired by the probabilistic hypergraph [11], we propose the adaptive probabilistic hypergraph to model multimodal features among social images and events. In our work, we improve the model in [11] by simultaneously learning the ranking scores of image samples and the weights of hyperedges, so that the relations among social images can be deeply exploited.

III. Hypergraph Learning

A. Problem Formulation

In machine learning, graphs are a fundamental tool for modeling pairwise relationships among objects, where the vertices denote the objects and the relationships between two objects are measured by edges. A learning algorithm can be performed on such a graph to classify unlabeled samples. For example, in a social network, the relations of different people could be formulated as a graph model, with the vertices and edges representing people and their relationships, respectively. However, a graph model does not handle heterogeneous data well. For example, in our problem, an image could be described by multimodal features, such as title, tags, capture time, and location, which are difficult to represent as a single node in a graph model. A natural way of formulating complex relational objects is the hypergraph model, where an edge can connect more than two vertices. For convenience, some important notations used in this paper are listed in Table I.

Mathematically, let $\mathcal{V}$ denote a finite set of objects, and $\mathcal{E}$ be the family of subsets e of $\mathcal{V}$ such that $\bigcup_{e\in\mathcal{E}} = \mathcal{V}$. Then, we call $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ a hypergraph with vertex set $\mathcal{V}$, hyperedge set $\mathcal{E}$, and hyperedge weight vector w. A $|\mathcal{V}| \times |\mathcal{E}|$ incidence matrix $\mathbf{H}$ represents $\mathcal{G}$ with the following elements:

$$H(v,e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Equation (1) defines the traditional hypergraph structure, which assigns vertex v to hyperedge e by means of a binary decision; that is, whether a vertex belongs to an edge. In this model, all vertices on a hyperedge are treated equally, but some information is lost, which may be harmful to hypergraph-based applications.

Similar to [11], in this paper we employ a probabilistic hypergraph model to overcome this limitation. The incidence matrix $\mathbf{H}$ of a probabilistic hypergraph is defined as

$$H(v,e) = \begin{cases} Sim(v,e), & \text{if } v \in e \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Here, $Sim(v,e)$ is defined as the similarity of v and the "centroid" vertex of e, i.e., the vertex that generates hyperedge e. The similarity estimation method is detailed in Section IV.

According to this formulation, vertex v is "softly" assigned to edge e based on the similarity $Sim(i,j)$ between v and e. In this way, not only the local grouping information but also the probability that a vertex belongs to a hyperedge are considered when the graph is constructed, so that the correlation among vertices is more accurately described.

Based on Equation (2), the vertex degree of each vertex $v \in \mathcal{V}$ is defined as

$$\mathbf{d}(v) = \sum_{e\in\mathcal{E}} \mathbf{w}(e)H(v,e), \tag{3}$$

while the edge degree of hyperedge $e \in \mathcal{E}$ is given by

$$\delta(e) = \sum_{v\in\mathcal{V}} H(v,e). \tag{4}$$

TABLE I: Notations and definitions.

Notation	Definition
$\mathcal{X} = (x_1, x_2, \ldots, x_n)$	$\mathcal{X}$ denotes the media dataset, where x_i is the i-th document.
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$	$\mathcal{G}$ denotes a hypergraph, where $\mathcal{V}$, $\mathcal{E}$, and w are the set of vertices, set of edges, and weights of the hyperedges, respectively.
n	Number of vertices, that is, the size of $\mathcal{V}$.
$\mathbf{W}$	The diagonal matrix of w.
$\mathbf{D}_v$	The diagonal matrix of the vertex degree.
$\mathbf{D}_e$	The diagonal matrix of the edge degree.
$\mathbf{H}$	The incidence matrix of the hypergraph.
$\mathbf{y}$	The labels of media samples, in which relevant elements are set to one and irrelevant elements are set to zero.
$\mathbf{L}$	The Laplacian matrix of the hypergraph.
$\mathbf{f}$	The ranking scores obtained by the proposed method.

We use $\mathbf{D}_v$, $\mathbf{D}_e$, and $\mathbf{W}$ to denote, respectively, the diagonal matrices of the vertex and hyperedge degrees, and the weights.

In the constructed hypergraph structure, each image is denoted by a vertex while its K-nearest neighbors are linked via a hyperedge. In this setting, the media enrichment problem can be solved as a ranking problem; that is, ranking the media candidates according to their relevance to an event. Huang [11] proposed an image ranking framework, but the weights of different edges are directly computed from the weights of the incidence matrix and are not well investigated in the work. In fact, hyperedges have different effects and performing a weighting of the hyperedges is useful to exploit the relevance between the media items and event. In this paper, instead of directly computing the weights from the incidence matrix, we integrate the learning of hyperedge weights into the formulation, with the hypergraph model formulated as a regularization framework.

$$\arg\min_{\mathbf{f},\mathbf{w}} \mathbf{F} = \arg\min_{\mathbf{f},\mathbf{w}} \{\mathbf{\Omega}(\mathbf{f}) + \lambda \mathbf{R}_{emp}(\mathbf{f}) + \mu \mathbf{\Phi}(\mathbf{w})\} \tag{5}$$

where $\mathbf{f}$ is the ranking score to be learned. $\mathbf{\Omega}(\mathbf{f})$ is the empirical loss defined as

$$\begin{aligned}\mathbf{\Omega}(\mathbf{f}) =& \frac{1}{2} \sum_{e \in \mathcal{E}} \sum_{u,v \in \mathcal{V}} \frac{\mathbf{w}(e) H(u,e) H(v,e)}{\delta(e)} \\ &\times \left(\frac{\mathbf{f}(u)}{\sqrt{\mathbf{d}(u)}} - \frac{\mathbf{f}(v)}{\sqrt{\mathbf{d}(v)}} \right)^2 .\end{aligned} \tag{6}$$

Letting $\mathbf{\Theta} = \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}}$ and $\mathbf{\Delta} = \mathbf{I} - \mathbf{\Theta}$, the normalized cost function can be rewritten as

$$\mathbf{\Omega}(\mathbf{f}) = \mathbf{f}^T \mathbf{\Delta} \mathbf{f}, \tag{7}$$

where $\mathbf{\Delta}$ is a positive semi-definite matrix, known as the Laplacian of the hypergraph. $\mathbf{R}_{emp}(f)$ is the loss function defined as

$$\mathbf{R}_{emp}(\mathbf{f}) = \|\mathbf{f} - \mathbf{y}\|^2 = \sum_{v \in \mathcal{V}} (\mathbf{f}(v) - \mathbf{y}(v))^2, \tag{8}$$

where $\mathbf{y}$ is the initial label vector, which is computed based on pseudo-relevant samples in the hypergraph learning algorithm. The selection of pseudo-relevant samples is detailed in Section IV.

The last part in model, $\mathbf{\Phi}(\mathbf{w})$, is a regularizer designed to avoid over-fitting. As we already know, in hypergraph models, all the hyperedges are initialized with a weight. Obviously, different hyperedges have different effects on modeling the relations in different media. As shown in previous work [8], an optimized weighting is helpful to improve system performance. In machine learning, a natural approach for optimizing the weights, w, is regularization, which can be formulated as

$$\mathbf{\Phi}(\mathbf{w}) = \|\mathbf{w}\|_q, \tag{9}$$

where $q = 2$ denotes L_2 regularization. L_2 regularization, which is widely used in machine learning algorithms, decreases the model's non-linearity and makes the model robust by penalizing certain parameter configurations. This is the regularization method used in our proposed approach.

With L_2 regularization, the optimization function is formulated as

$$\begin{aligned}\arg\min_{\mathbf{f},\mathbf{w}} \mathbf{F} = \arg\min_{\mathbf{f},\mathbf{w}} &\left\{ \mathbf{f}^T \Delta \mathbf{f} + \lambda \|\mathbf{f} - \mathbf{y}\|^2 + \mu \sum_{i=1}^{Ne} \|\mathbf{w}_i\|_2 \right\} \\ s.t. &\sum_{i=1}^{Ne} \mathbf{w}_i = 1.\end{aligned} \tag{10}$$

The two parameters $\lambda > 0$ and μ in Equation (5) are trade-off parameters that balance empirical loss and regularization term.

B. Solution

The objective function (10) is not jointly convex with respect to $\mathbf{f}$ and $\mathbf{w}$. However, it is convex with respect to $\mathbf{f}$ if $\mathbf{w}$ is fixed, and vice versa. Thus, an alternative optimization method can be used to approximate the optimal parameters.

First, if we fix $\mathbf{w}$, the learning task is to minimize the sum of the two terms [11]:

$$\arg\min_{\mathbf{f},\mathbf{w}} \mathbf{F} = \arg\min_{\mathbf{f},\mathbf{w}} \left\{ \mathbf{f}^T \Delta \mathbf{f} + \lambda \|\mathbf{f} - \mathbf{y}\|^2 \right\}. \tag{11}$$

From

$$\frac{\partial \mathbf{F}}{\partial \mathbf{f}} = 0$$

we have that

$$\begin{aligned} &f(\mathbf{I} - \mathbf{\Theta}) + \lambda(\mathbf{f} - \mathbf{y}) = 0 \\ \Longrightarrow \mathbf{f} = &\frac{\lambda}{1+\lambda} \left(\mathbf{I} - \frac{\mathbf{\Theta}}{1+\lambda} \right)^{-1} \mathbf{y}, \end{aligned} \tag{12}$$

where $\mathbf{\Theta} = \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-1/2}$, as defined previously.

Then, with a fixed $\mathbf{f}$, we can optimize parameter $\mathbf{w}$. Now the optimization becomes

$$\arg\min_{\mathbf{w}} \mathbf{F} = \arg\min_{\mathbf{w}} \left\{ \Omega(\mathbf{f}) + \mu \sum_{i=1}^{Ne} \|\mathbf{w}_i\|_2 \right\}$$
$$s.t. \sum_{i=1}^{Ne} \mathbf{w}_i = 1. \tag{13}$$

Letting

$$\gamma(e) = \frac{1}{2} \sum_{u,v \in \mathcal{V}} \frac{H(u,e)H(v,e)}{\delta(e)} \times \left(\frac{\mathbf{f}(u)}{\sqrt{\mathbf{d}(u)}} - \frac{\mathbf{f}(v)}{\sqrt{\mathbf{d}(v)}} \right)^2, \tag{14}$$

Equation (13) can be rewritten as

$$\arg\min_{\mathbf{w}} \mathbf{F} = \arg\min_{\mathbf{w}} \left\{ \mathbf{w}^T \gamma + \mu \sum_{i=1}^{Ne} \|\mathbf{w}_i\|_2 \right\}$$
$$s.t. \sum_{i=1}^{Ne} \mathbf{w}_i = 1, \tag{15}$$

which could be cast as the following second-order cone problem and solved by the inner-point method [5]:

$$\arg\min_{\mathbf{w},\mathbf{t}} \left\{ \mathbf{w}^T \gamma + \mu \mathbf{t} \right\}$$
$$s.t. \sum_{i=1}^{Ne} \mathbf{w}_i = 1, \|\mathbf{w}_i\|_2 <= \mathbf{t}_i. \tag{16}$$

It is worth noting that the computation of γ requires the value of $\mathbf{d}$, which is related to $\mathbf{w}$. However, the minimization of the cost function in Equation (13) is iteratively solved in the global alternative optimization, and $\mathbf{w}$ converges in the iteration.

The alternative optimization implementation is summarized in Algorithm 1.

Algorithm 1 Proposed adaptive probabilistic hypergraph model method

1: Compute similarity matrix Sim_t based on various features.
2: Construct the probabilistic hypergraph G. For each vertex, find its K-nearest neighbors to build a hyperedge.
3: Compute the hypergraph incidence matrix $\mathbf{H}$, and hypergraph Laplacian $\mathbf{L}$.
4: Optimize $\mathbf{f}$ according to Equation (12).
5: Optimize $\mathbf{w}$ from function (16) by the inner point method.
6: Update hypergraph Laplacian $\mathbf{\Delta}$ according to the new $\mathbf{w}$.
7: Go to step 4 until max loops have been completed.

C. Computational Cost

From the above solution process, we can see that there are two steps in the alternative optimization framework, while most of the computation is spent on optimizing the hypergraph weight $\mathbf{w}$, especially solving the objective function in Equation (16). The function could be optimized by a primal-dual interior-point algorithm with computational complexity $O(N^3)$ [22], where N is the dimension of $\mathbf{w}$. Hence, the computational complexity of our proposal is $(K_l O(N^3))$, where K_l is the number of iterations of the global alternative optimization. In practice, the hypergraph incidence matrix is typically sparse, which implies the problem can be solved much faster when the sparsity is exploited and can scale well with larger datasets.

IV. Event-based Media Enrichment

In this section, we apply the proposed adaptive hypergraph learning to media enrichment problem. First, different features, such as visual, textual, spatial, and temporal features, are extracted from social images. Second, for each kind of feature, a set of hyperedges is generated from each sample and its corresponding neighbors, and the hypergraph is constructed based on these hyperedges. After collecting some pseudo samples to initiate the learning, the ranking score of each sample and the weights of the hyperedges are simultaneously optimized through an alternating optimization. Finally, we obtain the enrichment list according to the ranking scores.

A. Hypergraph Construction

The media content used in this study is a set of images downloaded from Flickr. To model the images using a hypergraph model, we create a hyperedge for each image in the dataset by taking each image as the "centroid" vertex and forming a hyperedge with the center image and its K-nearest neighbors. To formulate the hyperedges of the constructed hypergraph, we consider visual, textual, spatial, and temporal features, which are often used in social event analysis tasks [17], [2].

- **Textual feature** We use the tags and title of each image as the textual source to compute the textual feature. At first, we utilize the Google Translate API to translate non-English words into English. Then, the textual metadata are cleaned by removing stopwords, HTML tags, and some noise terms. Finally, we employ the Boolean weighting scheme to measure the term's frequency of tags [20], and represent each document as a textual vector, while each dimension in the vector corresponds to a separate term. If a term occurs in the document, its value in the vector is one, otherwise it is zero. The dimension of the textual feature is equal to the size of the word dictionary n_w.
- **Visual feature** The content of an image is represented as a bag-of-visual words feature. The generation of visual words comprises three steps: First, we apply a difference-of-Gaussian filter on the grayscale image to detect the salient points. Then, we calculate the SIFT features over the detected salient points, and finally, we employ the K-means clustering algorithm to quantize the SIFT descriptor as a visual feature vector. We employ the K-means clustering algorithm to quantize the SIFT descriptor as a n_v-dimensional visual feature vector.

- **Temporal feature** Time is one of the key components of an event. The temporal source used in this paper is the photo capture time. In our work, we segment the time span of the images in the dataset every n_t days, and each image x_i is represented by an n_t feature vector t, where t_i = 1 indicates that x_i falls within the i-th interval.
- **Location feature** To create the location feature, we first extract the GPS metadata, that is, the latitude and longitude coordinates. Then we use the K-means clustering algorithm to cluster the data into n_l clusters, the GPS data for each image is projected onto the cluster, and each image p_i is represented by an n_l feature vector l, where l_i = 1 indicates that p_i is closed within the i-th clustering center. GPS information is not required in photo metadata and for this reason, the feature vector is filled with zeros if a value is missing.

For each of the above four features, we use Euclidean distance to calculate the distances $Dist_t(i, j)$, and then we compute the similarity matrix between two images as

$$Sim_t(i, j) = exp(-\frac{Dist_t(i, j)}{D_t}), \tag{17}$$

where D_t is the mean value of distances calculated by the t-th feature.

The hyperedges are constructed based on similarity matrix Sim_t: we take each media document as the "centroid" vertex and form a hyperedge with the center image and its K-nearest neighbors. That is, for each vertex, its K-nearest neighbors measured by similarity in the feature space generate a hyperedge. All the hyperedges generated based on different features are aggregated and the final hypergraph is constructed accordingly.

B. Seed Sample Collection

Note that we use a set of pseudo-relevant samples for hypergraph learning. In this section, we introduce the pseudo-relevant sample selection method. As is known, title, time, and location are three key factors identifying an event. The corresponding photo metadata are the textual description, capture time, and place. Since the three factors are independent, we can measure the relevance of a given photo P to event E by

$$R(P, E) = R(P.text, E.title)R(P.time, E.time)R(P.geo, E.geo), \tag{18}$$

where the first item measures the similarity of the photo textual description and event title. As they are short and sparse, the most straightforward way to measure them is

$$R(Text1, Text2) = \frac{|Text1 \cap Text2|}{|Text2|}, \tag{19}$$

where function $|\cdot|$ is the total number of words in a textual vector.

The second item in Equation (18) measures the span between the photo capture time and the event time as

$$R(Time1, Time2) = e^{-\left|\frac{date(Time2-Time1)}{N_t}\right|}, \tag{20}$$

where function $date(\cdot)$ calculates the day of the given time span.

The last item in Equation (18) measures the distance between the photo geotag and the event location. Because of the many photos without geotags, as well as the limited accuracy of GPS data in the Flickr dataset, we only use the city/venue description to measure the location feature. The method is the same as that defined in Equation (19).

All the media documents are ranked by their similarity to a given event and the top N_s samples are selected as seed samples.

V. Experiments

To demonstrate the effectiveness of the proposed approach, we conducted experiments on a dataset collected from EventMedia [32], and used the method proposed in [18] to find media candidates illustrating events. We compared the performance of the proposed adaptive probabilistic hypergraph learning approach with the latest ranking approaches, including K-nearest neighbor-based ranking, graph ranking [1], probabilistic hypergraph ranking, and SVM^{rank} [12].

A. Dataset

To evaluate the proposed approach, we collected an event dataset originating from EventMedia, which was created by Troncy et al. [32] using linking data techniques. There are about 100,000 events in this corpus, illustrated by 1.7M photos. Since we need sufficient examples for training and testing, we randomly selected 60 events with at least 50 relevant photos each. In total, there are 4560 images with machine tags in the dataset.

In the dataset, the explicit relationships between these events and photos hosted on Flickr can be looked up using special machine tags such as lastfm:event=XXX or upcoming:event=XXX. These tags are usually manually generated by photo uploaders. Hence, media items labeled with relevant machine tags can be used as positive samples of events.

Besides the media items with machine tags, we also collected a set of illustrative candidates potentially taken at an event, according to the method proposed in [18]. In this method, the event metadata, such as title and capture place and time are completely extracted from the event dataset, while an online query with geographical, temporal, and textual parameters is performed to collect the social media data potentially taken at the event. In total, for the 60 events, 13510, and 5218 images were collected by querying Flickr based on "title" and "location", respectively, with time constraints. Therefore, our dataset contains 28,288 images in total. Then, for each event, the seed sample collection method was performed to collect pseudo-relevant samples. Some photo samples are illustrated in Figure 2.

B. Experimental configuration

For our event-illustrating task, we compared the proposed adaptive probability hypergraph ranking method with the following methods.

1) Proposed adaptive probabilistic hypergraph ranking. Parameters λ and μ are selected by 5-fold cross validation.

Algorithm 2 Probabilistic hypergraph ranking approach [11], empirically compared with our approach

1: Compute similarity matrix Sim_t based on various features.
2: Construct the probabilistic hypergraph G. For each vertex, find the K-nearest neighbors to build a hyperedge.
3: Compute the hypergraph incidence matrix $\mathbf{H}$, and hypergraph Laplacian $\mathbf{\Delta}$.
4: Compute w by $w(e_i) = \sum_{v_j \in e_i} A(i,j)$
5: Optimize f according to Equation (12).

Algorithm 3 Pairwise graph-based ranking approach, empirically compared with our approach

1: Compute similarity matrix Sim based on various features: $Sim(i,j) = exp(-\frac{1}{4}\sum_{k=1}^{4}\frac{Dist_k(i,j)}{D_k})$
2: Construct the simple graph G_s according to the similarity matrix. For each vertex, connect the K-nearest neighbors.
3: Compute the simple graph affinity matrix A_g: $A_g(i,j) = Sim(i,j)$ if the i-th and j-th vertices are connected, otherwise $A_g(i,j) = 0$.
4: Compute the vertex degree matrix $D = \sum_j A_g(i,j)$
5: Compute the simple graph Laplacian $\Delta_g = I - \Theta_g = I - D^{-1/2}A_g D^{-1/2}$
6: Optimize f according to Equation (12).

The neighborhood size varies in 5, 10, 15, 20 for the hyperedge generation process and the number of iterations for the alternative optimization process is set to 20.

2) Adaptive hypergraph ranking [42]. The normal 0-1 representation is used to indicate whether a vertex belongs to an edge. The parameter settings are the same as those for the adaptive probability hypergraph ranking.
3) Probability hypergraph ranking as detailed in Algorithm 3. This method does not learn the weights of the hyperedges, but instead, computes them directly from the similarity matrix. Parameter λ is selected by five-fold cross validation, while the neighborhood size also varies from five to 20 in steps of five, with the best result reported.
4) K-nearest neighbor-based ranking. In pattern recognition, the K-nearest neighbors algorithm (k-NN) is an instance-based method for classification and regression. It can also be employed to solve the ranking problem, that is, ranking an object using the average Euclidean distance to its K-nearest neighbors in the training set. Parameter K is set to ten experimentally.
5) Pairwise graph-based ranking as detailed in Algorithm 3. A simple graph is constructed according to the similarity matrix of the object, and then the graph Laplacian is computed based on the affinity and vertex degree matrix. Finally, the ranking score is optimized according to Equation (12).
6) Support Vector Machine(SVM) ranking [12]. The basic idea of the ranking SVM is to formalize learning to rank as a binary classification problem on instance pairs, and then solving the problem using SVMs. The original multiple features are exploited to train the SVM ranking model. Since it is hard to label the relevance of all images in our dataset, for simplicity we set the relevance score of images with an event machine tag to one, and those without machine tags to zero.

(A) event: "Indietracks Festival 2012", taken on 6-8 July 2012, Butterley Hill, United Kingdom.

(B) event: "Pukkelpop 2013", taken on 15-17 Aug 2013, Kempische Steenweg Hasselt

Fig. 2: Some samples and illustrative candidates for two events.

Two measures are employed to evaluate the performance of the ranking methods discussed above: (1) precision vs. recall (PR) curve; and (2) mean average precision (MAP). In pattern recognition, the PR curve measures the relation between the fraction of retrieved instances that are relevant (Precision), and the fraction of relevant instances that are retrieved (Recall). The area under the curve is the average precision (AP), which is one of the most popular criteria for evaluating classification and information retrieval tasks.

C. Experimental Results

1) Comparative Results: First, we evaluated the effectiveness of our adaptive probabilistic hypergraph method compared with other state-of-the-art methods. For each event, we built a ranking model with the collected image documents. In this experiment, we randomly selected the top N_s image samples according to the pseudo-sample selection method as the initial relevance for each event, and used all the samples with machine tags for testing purposes. For all the methods evaluated in this paper, we independently repeated the experiments ten times with randomly selected training samples and report the average results in Figure 3. Our method, which not only takes advantage of probabilistic hypergraph ranking, but also optimizes the hyperedge weight by L_2 regularization, achieves the best performance of the four methods (AP: 0.950). From the results, we see that both learning the weights of the hyperedges and representing the incidence matrix in a probabilistic way further exploit the relation among the media documents and achieve better performance. This demonstrates the effectiveness and feasibility of our approach.

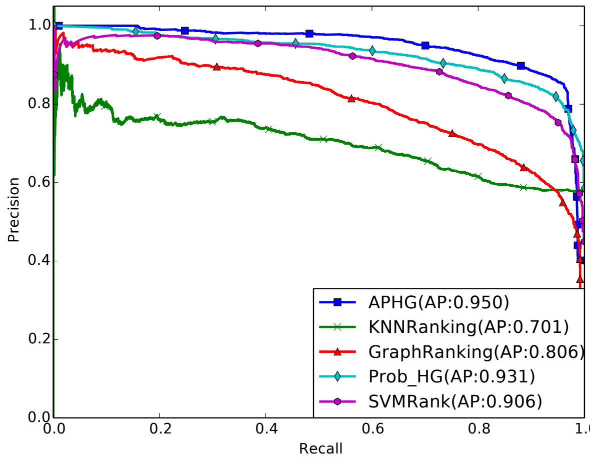

Fig. 3: Performance comparison between our approach and other ranking methods. **APHG**: the proposed adaptive probabilistic hypergraph ranking; **KNNRanking**: K-nearest neighbor-based ranking; **GraphRanking**: simple graph-based ranking; **Prob_HG**: probabilistic hypergraph ranking; **SVM-Rank**: SVM ranking method.

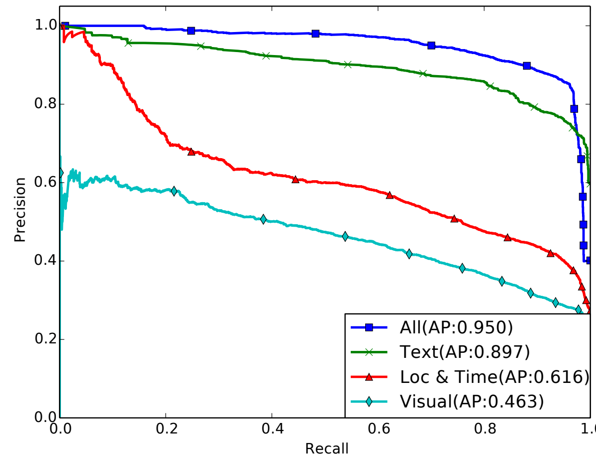

Fig. 4: The top results obtained by adaptive probabilistic hypergraph learning with (a) the integration of all modalities; (b) only textual modality; (c) only time and location modalities; and (d) only visual modality. From the results we can see that the proposed method integrating multiple modalities obtains much more relevant results.

In addition, the statistical importance of the different ranking methods was examined by T-test. For each event, we evaluated the performance of different ranking approaches under the MAP criterion. The p-values of the T-test comparing the proposed method with all the other methods, are shown in Table (II). From the results, we can see that the proposed method yields statistically significant improvement.

TABLE II: The p-values of the significance tests.

Comparison	p-value
APHG vs. GraphRanking	$5.71 \times e^{-12}$
APHG vs. HG_L2	$2.71 \times e^{-9}$
APHG vs. KNNRanking	$5.20 \times e^{-13}$
APHG vs. Prob_HG	$9.66 \times e^{-16}$
APHG vs. SVMRank	$4.41 \times e^{-15}$

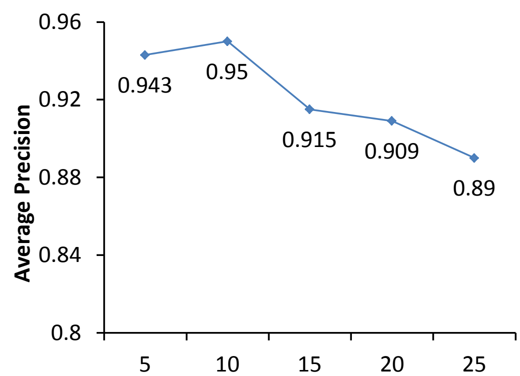

Fig. 5: The AP performance variation using different neighborhood sizes to construct the hypergraph.

D. Results of Exploiting Different Modalities

To further investigate the performance of our proposed method using different multimodal features, we compared the performance of hypergraph learning with different modalities. It should be noted that there are many missing values in the location modality, which is featured as a zero vector in the preprocessing. However, having zero columns in the hypergraph index matrix leads to unpredictable ranking results in the hypergraph. We avoided this problem by concatenating the time and location features when constructing the hypergraph. From the results reported in Figure 4, we conclude the following. (1) The hypergraph model is able to combine different modalities well and achieves the best performance. (2) Among the single features, textual features are the most representative since they directly show the event content. (3) Visual feature is not so robust in representing events, as images from the same types of events have a similar visualization.

E. On Parameter k

In our proposal, the nearest neighbor method is employed to generate a hyperedge. For each vertex, the K-nearest neighbors are found to construct the hyperedge. In our experiment, we also evaluated the impact of neighborhood size on our adaptive probabilistic hypergraph ranking model. Thus, we varied the size of the neighborhood used to generate the hypergraph from five to 20. The results, as reported in Figure 5, show that the best score is achieved with $k = 10$ using the average precision metric.

F. On Parameters λ and μ

In Equation (10), parameters λ and μ modulate the weights of the loss and regularization term, respectively. In other words, the value of λ determines the effect of the closeness of $\mathbf{f}$ and $\mathbf{y}$, while the value of μ determines the role that the

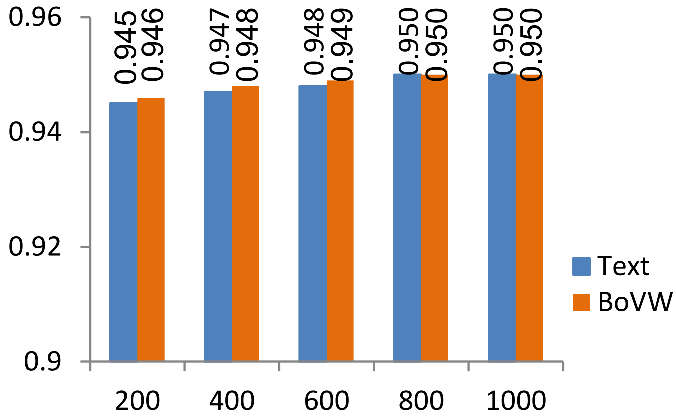

Fig. 6: The AP Results obtained by employing textual and visual dictionaries with different sizes.

weight of the hyperedge plays in the model. For example, if μ = 0, the model degrades to the probability hypergraph model in [11]. Hence, the effects of the two parameters need to be investigated. In our experiments, we set K = 10, and varied the values of λ from 0.005 to 50, and μ from 0.02 to 20. The AP results, reported in Table III, show that the best performance is obtained with $\lambda = 0.5$ and μ = 2. We can also see that the performance of our approach does not severely degrade when the parameters vary in a wide range.

TABLE III: The variation in average precision results when varying λ and μ

	μ=0.02	μ=0.2	μ=2	μ=20
λ=0.005	0.875	0.881	0.905	0.900
λ=0.05	0.930	0.934	0.939	0.936
λ=0.5	0.932	0.943	0.950	0.947
λ=5	0.933	0.941	0.944	0.938
λ=50	0.933	0.940	0.937	0.936

G. *On the Size of Textual and Visual Dictionaries*

As stated in Section IV, we used a clustering method to generate textual and visual dictionaries to generate textual and visual feature vectors. We compared the experimental results with different sizes of the textual and visual word dictionaries, that is, parameters n_w and n_v, respectively. The size of the two dictionaries ranged from 200 to 1000, with the results shown in Figure 6. It can be seen from the results that the algorithm is stable and robust against textual and visual dictionary size variations. When the dictionary size varies from 200 to 1000, the AP scores obtained by using textual and visual features changes from 0.946 to 0.950, and 0.947 to 0.950, respectively.

We also evaluated the sizes of the temporal and spatial feature vectors, n_t and n_l, respectively. The results are shown in Figure 7. We can see that as the sizes of the temporal and spatial features increase, system performance improves. Stable results are achieved with n_l = 80 and n_t = 600.

H. *On Parameter N_s*

In our experiment, we also evaluated the impact of the number of pseudo-relevant samples N_s on our adaptive probabilistic hypergraph ranking model. We varied the number of pseudo-relevant samples from 10 to 60. The results reported in Figure 8 show that system performance improves with an increasing N_s, where the optimal score is achieved when N_s is in the range [20,40]. However, system performance degrades when N_s is greater than 50. This makes sense, since sufficient samples ensure adequate diversity of the event, but employing too many pseudo-relevant samples may involve many noisy samples. Either way, the performance of our approach does not severely degrade as the number of pseudo-relevant images varies.

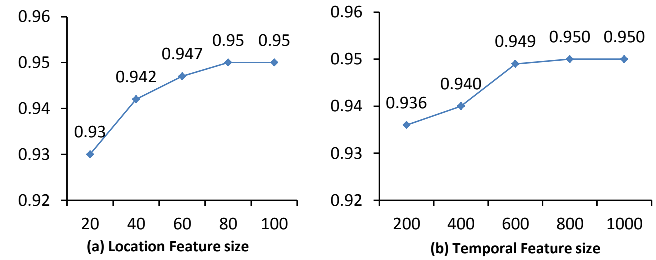

Fig. 7: The AP results obtained with different temporal and spatial feature sizes.

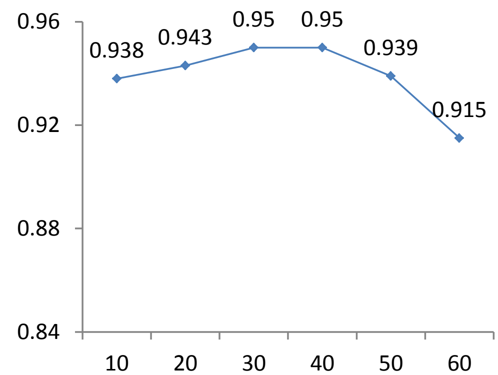

Fig. 8: The AP results obtained by using different numbers of pseudo-relevant samples.

From the above analysis, we conclude that: 1) probabilistic similarity is more suitable for modeling social media documents than the traditional 0-1 incidence matrix; and 2) by optimizing the hypergraph weights, the adaptive probabilistic hypergraph can better exploit the relations of social media data and realize improved performance.

VI. Conclusion

The exponential growth in social media data available online as witnessed over recent years has brought new challenges for managing and organizing media efficiently and effectively. Thanks to its multi-dimensional nature (who, what, when, and where), events are a powerful instrument to organize media. Hence, illustrating events using social media is timely and has started receiving considerable attention from the multimedia research community.

In this paper, we proposed a ranking method for finding photos relevant to a given event. The ranking is performed using a hypergraph model, which is constructed according to a probabilistic strategy. For each vertex, we find its K-nearest neighbors to build a hyperedge based on the similarity of the vertex pairs. In addition, we use L_2 regularization to optimize the hypergraph weights, which is considered a second-cone problem and solved by the inner-point method. For our event enrichment task, we collected an event dataset from EventMedia, and downloaded a photo collection by querying media from the social photo-sharing platform Flickr. Temporal, spatial, textual, and visual features were extracted to train the adaptive probabilistic hypergraph model. We evaluated our method by comparison with three different ranking methods. The results show that the proposed method simultaneously models the multimodal information well and achieves better results than the other approaches.

References

[1] S. Agarwal. Ranking on graph data. In *Proceedings of the International Conference on Machine Learning*, pages 25–32, 2006.

[2] B.-K. Bao, W. Min, K. Lu, and C. Xu. Social event detection with robust high-order co-clustering. In *Proceedings of the international conference on Multimedia Retrieval*, pages 135–142, 2013.

[3] B.-K. Bao, W. Min, J. Sang, and C. Xu. Multimedia news digger on emerging topics from social streams. In *Proceedings of the ACM international conference on Multimedia*, pages 1357–1358, 2012.

[4] H. Becker, M. Naaman, and L. Gravano. Event Identification in Social Media. In *12th International Workshop on the Web and Databases*, pages 31–36, Providence, USA, 2009.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[6] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of International Conference on Information and Knowledge Management*, 2009.

[7] D. Delgado, J. Magalhaes, and N. Correia. Assisted News Reading with Automated Illustrations. In *Proceedings of ACM conference on Multimedia*, pages 1647–1650, 2010.

[8] J. Derrac, I. Triguero, S. Garcia, and F. Herrera. Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(5):1383–1397, Oct 2012.

[9] N. Diakopoulos, M. Naaman, and F. Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 115–122, Oct. 2010.

[10] C. S. Firan, M. Georgescu, W. Nejdl, and R. Paiu. Bringing order to your photos: Event-Driven Classication of Flickr Images Based on Social Knowledge. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, Oct. 2010.

[11] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas. Image retrieval via probabilistic hypergraph ranking. In *Proceedings of the internal conference on Computer Vision and Pattern Recognition*, pages 3376–3383, 2010.

[12] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the international conference on Knowledge discovery and data mining*, pages 133–142, New York,USA, 2002.

[13] D. Joshi, J. Z. Wang, and J. Li. The Story Picturing Engine—a system for automatic text illustration. *ACM Transactions on Multimedia Computing Communications and Applications*, 2(1):68–89, 2006.

[14] P. Kazienko, K. Musial, and T. Kajdanowicz. Multidimensional social network in the social recommender system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 41(4):746–759, July 2011.

[15] H. Lee-Kwang and C.-H. Cho. Hierarchical reduction and partition of hypergraph. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(2):340–344, Apr 1996.

[16] X. Li, L. Mou, and X. Lu. Scene parsing from an map perspective. *IEEE Transactions on Cybernetics*, PP(99):1–1, Nov 2014.

[17] X. Liu and B. Huet. Heterogeneous features and model selection for event-based media classification. In *Proceedings of the ACM international conference on Multimedia Retrieval*, pages 151–158, 2013.

[18] X. Liu, R. Troncy, and B. Huet. Finding Media Illustrating Events. In *Proceedings of the ACM international conference on Multimedia Retrieval*, Trento, Italy, 2011.

[19] X. Lu and X. Li. Multiresolution imaging. *IEEE Transactions on Cybernetics*, 44(1):1–1, Jan 2014.

[20] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. 1 edition, July 2008.

[21] R. Mattivi, J. Uijlings, F. De Natale, and N. Sebe. Categorization of a collection of pictures into structured events. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, page 1, June 2012.

[22] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM J. on Optimization*, 8(2):324–364, Feb. 1998.

[23] P. Qian, F.-L. Chung, S. Wang, and Z. Deng. Fast graph-based relaxed clustering for large data sets using minimal enclosing ball. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3):672–687, 2012.

[24] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *International conference on World Wide Web*, pages 851–860, Apr. 2010.

[25] K. Schwarz, P. Rojtberg, J. Caspar, I. Gurevych, M. Goesele, and H. P. A. Lensch. Text-to-Video: Story Illustration from Online Photo Collections. In *Knowledge Based and Intelligent Information and Engineering Systems*, volume 6279, pages 402–409. 2010.

[26] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost. Facial action recognition combining heterogeneous features via multikernel learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4):993–1005, 2012.

[27] N. Shaker, S. Asteriadis, G. Yannakakis, and K. Karpouzis. Fusing visual and behavioral cues for modeling user experience in games. *IEEE Transactions on Cybernetics*, 43(6):1519–1531, Dec 2013.

[28] J. Song, Y. Yang, X. Li, Z. Huang, and Y. Yang. Robust hashing with local models for approximate similarity search. *IEEE Trans. on Cybern.*, 44(7):1225–1236, July 2014.

[29] J. Tang, X.-S. Hua, M. Wang, Z. Gu, G.-J. Qi, and X. Wu. Correlative linear neighborhood propagation for video annotation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.*, 39(2):409–416, April 2009.

[30] U. Tariq, K.-H. Lin, Z. Li, X. Zhou, Z. Wang, V. Le, T. S. Huang, X. Lv, and T. X. Han. Recognizing emotions from an ensemble of features. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 42(4):1017–1026, 2012.

[31] M. R. Trad, A. Joly, and N. Boujemaa. Large scale visual-based event matching. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, pages 1–7, Apr. 2011.

[32] R. Troncy, B. Malocha, and A. Fialho. Linking Events with Media. In *International Conference on Semantic Systems*, Graz, Austria, 2010.

[33] J. Wang, F. Wang, C. Zhang, H. Shen, and L. Quan. Linear neighborhood propagation and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1600–1615, Sept 2009.

[34] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu. Multimodal graph-based reranking for web image search. *IEEE Trans. Image Processing*, 21(11):4649–4661, Nov 2012.

[35] M. Wang and *et al.* Unified video annotation via multigraph learning. *IEEE Transactions on Circuits and Syst. Video Technol.*, 19(5):733–746, May 2009.

[36] J. Weng and B.-s. Lee. Event Detection in Twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*, pages 401–408. HP Laboratories, 2011.

[37] U. Westermann and R. Jain. Toward a Common Event Model for Multimedia Applications. *IEEE MultiMedia*, 14(1):19–29, 2007.

[38] A. K. C. Wong, S. Lu, and M. Rioux. Recognition and shape synthesis of 3-d objects based on attributed hypergraphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3):279–290, 1989.

[39] S. Xia and E. Hancock. 3d object recognition using hyper-graphs and ranked local invariant features. In *Structural, Syntactic, and Statistical Pattern Recognition*, LNCS, pages 117–126. Springer Berlin Heidelberg, 2008.

[40] C. Yang and T. Dorbin Ng. Analyzing and visualizing web opinion development and social interactions with density-based clustering. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 41(6):1144–1155, Nov 2011.

[41] A. Yazidi, O.-C. Granmo, and B. Oommen. Learning-automaton-based online discovery and tracking of spatiotemporal event patterns. *IEEE Transactions on Cybernetics*, 43(3):1118–1130, June 2013.

[42] J. Yu, D. Tao, and M. Wang. Adaptive hypergraph learning and its application in image classification. *IEEE Transactions on Image Processing*, 21(7):3262–3272, 2012.

[43] X. Zhu, A. B. Goldberg, M. Eldawy, C. R. Dyer, and B. Strock. A Text-to-Picture Synthesis System for Augmenting Communication. In *Proceedings of the international conference on Artificial intelligence*, number 2, pages 1590–1595, 2007.