Reconstructing Storyline Graphs for Image Recommendation from Web Community Photos

Gunhee Kim
Disney Research Pittsburgh
gunhee@cs.cmu.edu

Eric P. Xing
Carnegie Mellon University
epxing@cs.cmu.edu

Abstract

In this paper, we investigate an approach for reconstructing storyline graphs from large-scale collections of Internet images, and optionally other side information such as friendship graphs. The storyline graphs can be an effective summary that visualizes various branching narrative structure of events or activities recurring across the input photo sets of a topic class. In order to explore further the usefulness of the storyline graphs, we leverage them to perform the image sequential prediction tasks, from which photo recommendation applications can benefit. We formulate the storyline reconstruction problem as an inference of sparse time-varying directed graphs, and develop an optimization algorithm that successfully addresses a number of key challenges of Web-scale problems, including global optimality, linear complexity, and easy parallelization. With experiments on more than 3.3 millions of images of 24 classes and user studies via Amazon Mechanical Turk, we show that the proposed algorithm improves other candidate methods for both storyline reconstruction and image prediction tasks.

1. Introduction

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

In this paper, as shown in Fig.1, we investigate an approach for *inferring storyline graphs* from a large set of photo streams contributed by multiple users for a topic of interest (*e.g.* *independence+day*), of which a photo stream is a set of images that are taken in sequence by a single photographer within a fixed period of time (*e.g.* one day). A storyline usually refers to a series of events that have *chronological* or *causal* relations, which are commonly represented by a directed graph [11, 17]. Likewise, our goal in this paper is to automatically infer such directed storyline graphs from a large set of photo streams. Conceptually, the vertices in the graph correspond to dominant image clusters across the dataset, and the edges connect the vertices that sequentially recur in many photo streams. Its more rigorous definition will be developed throughout this paper.

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

Our objective differs from the *private* storyline [15], which is a summary of a single user's photo albums only. In this scenario, the face identification is important so that the storyline lays out in the center of herself or her close friends. Although the private storylines are also demanding, we here aim at building *collective* storyline graphs by leveraging all available photo sets. In addition, we also discuss *weakly-personalized* storyline graphs, in which we leverage a friendship graph so that we weight more on the photo streams of a particular user's close friends.

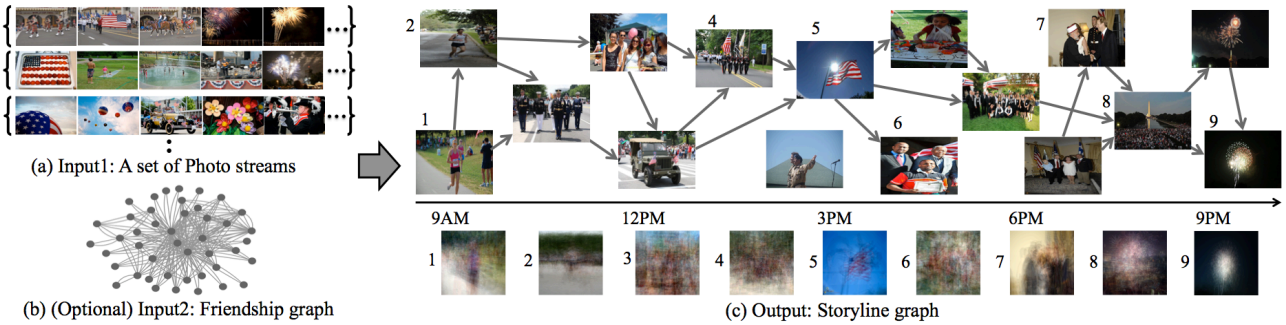


Figure 1. Motivation for reconstructing storyline graphs from large sets of Web photo streams with an *independence+day* example. The input is two-fold: (a) A set of photo streams that are independently taken by multiple users at different time and places, and (b) optionally a friendship graph. (c) The output is the storyline graph as a structural summary. The vertices are the exemplars of image clusters, and the edges connect sequentially recurring nodes across photo streams. We show the average images of nine selected node clusters in the bottom.

In order to show the usefulness of storyline graphs further, we leverage them to perform image sequential prediction tasks, which are directly connected to the *photo recommendation* applications. For example, once we have storylines as pictorial summary of what people usually do during *snowboarding* trips, we can recommend a part of their experiences to a user who is about to start his own *snowboarding* trip. This is analogous to the Amazon's function of *Customers Who Bought This Item Also Bought*.

We formulate the storyline reconstruction as an inference problem of sparse time-varying directed graphs (*e.g.* [22]). We then propose an optimization algorithm that enjoys several appealing properties for large-scale problems such as optimality guarantee, linear complexity, easy parallelization, and asymptotic consistency. For evaluation, we collect more than 3.3 millions of Flickr images of 42 thousands of photo streams for 24 topic classes. In our experiments, we first show that our storylines are more successful structural summary than other baselines, using the annotations obtained from the Amazon Mechanical Turk. We also quantitatively demonstrate that our approach outperforms other candidate methods for image sequential prediction tasks.

Relation to previous work. In the recent research of web mining, much work has been done to extract diverse threads of stories from online text corpora such as news articles and scientific papers [1, 5, 19, 20]. Partly inspired by this line of research, our work fundamentally differs from them that we leverage Web image collections instead of text data. In [25], images are jointly used with texts to generate storylines; however, only primitive image features are used, and more importantly, the algorithm is tested with a cleaned small dataset of 355 images.

In computer vision, the storyline mining has been actively studied for the videos of sports [6] and news [13]. However, videos usually contain a small number of specified actors in fixed scenes with synchronized voices and captions, all of which are not available in Web community photos. Another thread of related research is to explore the collections of landmark photos taken by tourists. In this line of work, the storylines are implicitly implemented in geometric ways such as 3D models of landmarks [21] or tourists' paths [2, 7]. Our work differs in that we aim at building storyline graphs of general topics in which no geometric constraints are available (such as *fly+fishing*). Other notable related work is summarized as follows. In [15], a storyline-based summarization is discussed for small-sized private photo albums. However, it is tested with only small data sets of about 200 images, and it cannot correctly handle multiple users' pictures. In [9], the temporal evolution of subtopics of Web images is visualized on timeline. But its output is a similarity graph between images, and thus no concept of story is implemented. The work of [8] is motivated by the photo storyline reconstruction for outdoor activity classes. However, it is a preliminary research that solely focuses on alignment and segmentation of photo streams; no storyline reconstruction is explored.

Contributions. The main contributions of this paper can be summarized as follows.

(1) To the best of our knowledge, our work is the first attempt so far to address the automatic reconstruction of storyline graphs from large sets of online images, especially for the topics of recreational activities, holidays, and sports events. Our method delivers a novel structural summary, which can not only visualize various events or activities associated with the topic in a form of a branching network, but also potentiate applications such as image recommendation.

(2) We develop an optimization algorithm for inferring sparse time-varying directed storyline graphs from large-scale photo streams along with other side information, while attaining several key Web-scale challenges, including global optimality, linear complexity, and easy parallelization. With experiments on more than 3.3 millions of images of 24 classes and user studies via Amazon Mechanical Turk, we demonstrate that the proposed method is more successful than other candidate methods for storyline reconstruction and image prediction tasks.

2. Problem Formulation

The input of our algorithm is two-fold. The first input is the set of photo streams of a particular topic. It is denoted by $\mathcal{P} = \{P^1, \cdots, P^L\}$, where L is the number of photo streams. Each photo stream $P^l = \{p_1^l, \cdots, p_{L^l}^l\}$ is a set of sequential images taken by a single photographer within a period of time $[0, T]$, which is set to one day. Thus, the resultant storyline graph is defined in the range of $[0, T]$. We assume each image p_i^l is associated with owner ID u^l and timestamp t_i^l, and images in each photo stream are sorted by timestamps. The second optional input is a friendship graph $\mathcal{G}_F = (\mathcal{U}, \mathcal{E}_F)$, which is a weighted symmetric graph. The vertex set is the set of users, and the edge weights indicate the degrees of friendship.

Since the image set is too large and much of images are highly overlapped, it is inefficient to build a storyline graph over individual images. Preferentially, the vertices of storyline graphs correspond to the clusters of images that recur in the input image set. We implement such *image clusters* by using the idea of encoding and decoding of neural coding [16]. Conceptually, the *encoding* represents each image by a small set of codewords. Then the storyline graph is defined over the codewords. The *decoding* can instantiate the graph over the codewords into the graph over images.

Image encoding. In order to capture various visual information of images, we use four different image descriptors, which are denoted by (`SIFT`), (`HOG2x2`), (`Tiny`), and (`Scene`). The (`SIFT`) and the (`HOG2x2`) are three-level spatial pyramid histograms for densely extracted HSV color SIFT and histogram of oriented edge (HOG) features, respectively. The (`Tiny`) denotes the Tiny image feature [24] that is RGB values of a 32×32 resized image. Since all three features are high-dimensional, we use the soft vector quantization for compact representation; for each feature type, we construct $D_j (= 600)$ image clusters by applying K-means to randomly sampled image features, and then each image is assigned to the c nearest image clusters with Gaussian weighting. Finally, the (`Scene`) denotes the score vector of linear one-vs-all SVM classifiers for 397 scene categories of the SUN dataset [26]. The (`Scene`) conveys a meaningful high-level description of images since much of Web images contain scenes. Likewise, we limit the (`Scene`) vector to retain only top-c highest values. We use $c = \{1, 3, 5\}$, and ℓ_1-normalize all four descriptor vectors.

Consequently, each image is assigned to J sets of descriptor vector $\mathbf{x}_j \in \mathbb{R}^{D_j}$ with c nonzeros each. Although we here use $J = 4$ with $[D_j]_{j=1}^4 = [600, 600, 600, 397]$, one can append any number of different image descriptors. We can concatenate J vectors to $\mathbf{x}$, where $|\mathbf{x}| = \sum_{j=1}^J D_j$, which does not affect our graph inference algorithm thanks to the independence assumption discussed in section 3.

Definition of storyline graphs. The storyline graph $\mathcal{G} = (\mathcal{O}, \mathcal{E})$ is defined as follows. Each node in the vertex set $\mathcal{O}$ corresponds to a codeword (*i.e.* $|\mathcal{O}| = D$), and the edge set $\mathcal{E} \subseteq \mathcal{O} \times \mathcal{O}$ includes directed edges between them. We let the storyline graph be *sparse* and *time-varying* [10, 22]. The sparsity is encouraged in order to avoid any unnecessarily complex story branches per node in which any images can follow any images. The time-varying graph means that we allow $\mathcal{E}^t$ to smoothly change over time in $t \in [0, T]$. It is based on that the popular transition between image codewords can vary over time; for example, in the *scuba+diving* class, the *underwater* images may be followed by *bright sky* images around noon but *sunset* images in the evening.

Therefore, the output of our algorithm is a set of storyline graphs $\{\mathbf{A}^t\}$ for $t \in [0, T]$, where $\mathbf{A}^t$ is the adjacency matrix of $\mathcal{E}^t$. Although we can compute $\mathbf{A}^t$ at any point t, in practice, we uniformly split $[0, T]$ into multiple time points (*e.g.* every 30 minutes), at which $\mathbf{A}^t$ is estimated. Sparsity encourages each $\mathbf{A}^t$ to have a small number of nonzero elements, while smoothness boosts the edge structure between consecutive $\mathbf{A}^t$ and $\mathbf{A}^{t+1}$ to changes smoothly.

Decoding: The decoding step retrieves the most suitable images for transitions between the codewords defined by $\mathbf{A}^t$ at time t. We adopt the approach of continuous error-correcting output codes (ECOC) [3], with the histogram intersection as the decoding metric. Any codeword or its combination of $\mathbf{A}^t$ can be represented by $\mathbf{h} \in \mathbb{R}^D$. Thus we can rank images near t by calculating the sum of element-wise minimum: $\sum_{d=1}^D \min(\mathbf{h}_d, \mathbf{x}_d)$, and retrieve the top-ranked image as a representative of $\mathbf{h}$.

3. Estimating Photo Storyline Graphs

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

For statistical tractability and scalability, our algorithm builds on four assumptions about photo streams that are reasonable in practice. Three of them are introduced in the following, and the fourth one is presented later. (`A1`) All photo streams are assumed to be taken independently of one another. (`A2`) We employ the k-th order Markovian assumption between the consecutive images in the photo stream[1]. (`A3`) The graphs are sparse and vary smoothly across time.

As a result of image encoding, each image is associated with a descriptor vector $\mathbf{x} \in \mathbb{R}^D$. Thus, we can denote a photo stream by $P^l = \{(\mathbf{x}_1^l, t_1^l), \ldots, (\mathbf{x}_{L^l}^l, t_{L^l}^l)\}$. We begin

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

our model by deriving the likelihood $f(\mathcal{P})$ of an observed set of photo streams $\mathcal{P}$. Based on the assumption (A1) and (A2), the likelihood $f(\mathcal{P})$ is defined as follows.

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

where $f(\mathbf{x}_i^l,t_i^l|\mathbf{x}_{i-1}^l,t_{i-1}^l)$ is the conditional likelihood of consecutive occurrence from image $\mathbf{x}_{i-1}^l$ at time t_{i-1}^l to $\mathbf{x}_i^l$ at t_i^l in photo stream l whose size is L^l. The forth assumption is imposed on the transition model. (A4) The codewords of $\mathbf{x}_i^l$ are conditionally independent one another given $\mathbf{x}_{i-1}^l$. That is, the transition likelihood factors over individual codewords: $f(\mathbf{x}_i^l,t_i^l|\mathbf{x}_{i-1}^l,t_{i-1}^l) = \prod_{d=1}^{D} f(x_{i,d}^l,t_i^l|\mathbf{x}_{i-1}^l,t_{i-1}^l)$.

As a simple transition model $f(\mathbf{x}_i^l,t_i^l|\mathbf{x}_{i-1}^l,t_{i-1}^l)$, we use a *linear dynamics model*: $\mathbf{x}_i^l = \mathbf{A}_e\mathbf{x}_{i-1}^l + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon}$ is a vector of Gaussian noise with zero mean and variance σ^2 (*i.e.* $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$). In order to encode temporal information between t_{i-1}^l and t_i^l into $\mathbf{A}_e \in \mathbb{R}^{D\times D}$, we use one of the two parametric rate models, the *exponential* and the *Rayleigh* model, which have been widely used to represent temporal dynamics of diffusion networks [18]. With $\Delta_i = t_i^l - t_{i-1}^l$, the (x,y) element a_{xy} of $\mathbf{A}_e$ is defined as

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

where $\alpha_{xy} \geq 0$ is the transmission rate from codeword x to y. Since we are interested in time-varying graphs, the α_{xy} is a function of time t_{i-1}^l. However, for simplicity, we here let α_{xy} stationary, and its dynamics will be discussed in next section. As $\alpha_{xy} \to 0$, the consecutive occurrence from codeword x to y is very unlikely. By letting $\mathbf{A} = \{\alpha_{xy}\exp(-\alpha_{xy})\}_{D\times D}$, we obtain the following transition model:

$$\mathbf{x}_i^l = g_i\mathbf{A}\mathbf{x}_{i-1}^l + \boldsymbol{\epsilon},\ \ g_i = \begin{cases} \exp(\Delta_i) & \text{(Exponential)} \\ \Delta_i\exp(\Delta_i^2/2) & \text{(Rayleigh)} \end{cases} \quad (3)$$

From Eq.(3), we can express the transition likelihood as Gaussian distribution: $f(x_{i,d}^l,t_i^l|\mathbf{x}_{i-1}^l,t_{i-1}^l) = \mathcal{N}(x_{i,d}^l; g_i\mathbf{A}_{d*}\mathbf{x}_{i-1}^l,\sigma^2)$, where $\mathbf{A}_{d*}$ denotes the d-th row of $\mathbf{A}$. Finally, the log-likelihood $\log f(\mathcal{P})$ of Eq.(1) is

$$\log f(\mathcal{P}) = -\sum_{l=1}^{L}\sum_{i=2}^{L^l}\sum_{d=1}^{D} f(x_{i,d}^l) \quad \text{where} \quad (4)$$

$$f(x_{i,d}^l) = \left(\frac{L^l}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(x_{i,d}^l - g_i\mathbf{A}_{d*}\mathbf{x}_{i-1}^l)^2\right).$$

3.1. Optimization

Now we discuss the optimization method to discover nonzero elements of $\mathbf{A}^t$ for any $t \in [0,T]$, by maximizing the log-likelihood of Eq.(4). One difficulty here is that for a fixed t, the learning data (*i.e.* images occurring at t) may be scarce, and thus the estimator may suffer from extremely high variance. To overcome such difficulty, we take advantage of the assumption (A3), which allows to estimate $\mathbf{A}^t$ by re-weighting the observation data near t accordingly. Furthermore, thanks to the assumption (A4), we can separately perform an optimization for each codeword d ($d = 1,\ldots,D$). This approach is known as *neighborhood selection* in graph inference literature [12]. Consequently, we iteratively solve the following optimization problem per dimension D times:

$$\widehat{\mathbf{A}}_{d*}^t = \operatorname{argmin}\sum_{l=1}^{L}\sum_{i=2}^{L^l} w^t(i)(x_{i,d}^l - g_i\mathbf{A}_{d*}^t\mathbf{x}_{i-1}^l)^2 + \lambda\|\mathbf{A}_{d*}^t\| \quad (5)$$

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

$$w^t(i) = \frac{\kappa_h(t-t_i^l)}{\sum_{l=1}^{L}\sum_{i=2}^{L^l}\kappa_h(t-t_i^l)},\ \ \kappa_h(u) = \frac{\exp(-u^2/2h^2)}{\sqrt{2\pi}h} \quad (6)$$

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

In Eq.(5), we include ℓ_1-regularization for a sparse graph structure, where λ is a parameter that controls the sparsity of $\widehat{\mathbf{A}}_{d*}^t$. This approach not only avoids overfitting but also is practical because the branches of storylines at each node are simple enough to be easily understood. Consequently, our graph inference reduces to solving a standard weighted ℓ_1-regularized least square problem, whose global optimum solution can be attained by highly scalable techniques such as the coordinate descent [4]. Therefore, the overall graph inference can be performed in a linear time with respect to all parameters, including the number of images and the dimension of codewords D. Our MATLAB code takes less than five minutes to obtain the set of 40 $\{\mathbf{A}\}$ for 245K images of the *surfing+beach* topic with $D = 1,800$. Note that the scalability of our algorithm, including linear complexity and trivial parallelization per codeword dimension, is of particular importance in our problem using millions of images with possibly many different image descriptors. We present more details of the algorithm in the supplementary, including the pseudocode and its asymptotic statistical consistency, which guarantees that true graph can be discovered as the number of data points increases indefinitely [22].

It is straightforward to extend the above optimization to the k-th order Markovian assumption. Simply, Eq.(3) is extended to an autoregressive model with the k-th order: $\mathbf{x}_i^l = \sum_{q=1}^{k} g_i(q)\mathbf{A}(q)\mathbf{x}_{i-q}^l + \boldsymbol{\epsilon}$, and the square loss function of Eq.(5) is changed accordingly.

Once $\{\mathbf{A}\}$ is discovered, the *parameter learning* updates the associated weights of nonzero entries of each $\mathbf{A}^t$,

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

3.2. Incorporating Meta-Data as Side Information

When side information is available such as a friendship graph, GPS data, and other types of temporal information, we can customize the storyline graphs accordingly. For example, given a particular user u_q, the storyline graph can be recast by weighting more the photo streams of u_q's neighbors in the friendship graph $\mathcal{G}_F$. Another example is a season-specific storyline graph, given that the popular activities or events of outdoor activities (*e.g. fly+fishing*) would change much from summer to winter. We utilize the *product kernel* as a unified framework to incorporate such side information for graph inference. For example, if a particular user u_q and a month s_q is given, the weighting function of Eq.(6) is replaced by

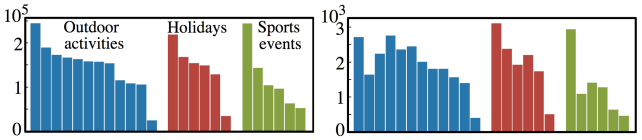
$$w^t(i,u_q,m_q)=\frac{\kappa_h(t-t_i^l)\kappa_s(s_q-s_i^l)\kappa_u(\rho(u_q,u^l))}{\sum_{l=1}^{L}\sum_{i=2}^{L^l}\kappa_h(t-t_i^l)\kappa_s(s_q-s_i^l)\kappa_u(\rho(u_q,u^l))} \quad (7)$$

where $\rho(u_q,u_i^l)$ is the distance between user u_q and u_i^l in the friendship graph. For the user distance, we use the inverse of the score of *random walk with restart* [23]. Consequently, this kernel weighting technique is flexible; we can easily extend the product kernel by including other continuous side information to enforce the smooth variation effect.

3.3. Image Recommendation using Storylines

Leveraging storyline graphs, we perform two sequential image prediction tasks, which are closely connected to photo recommendation applications. (I) Given a short sequence of images taken by a user, we predict K next likely images, which can help the user instantaneously preview the pictures of other users who already had the similar experience. (II) Given two parts of temporally distant images, we estimate the most likely paths between them. This function can be applied to *fill in missing parts* of one's photo stream, by referring to the summary of other users' pictures.

The inference of storyline graphs produces a set of $\{\mathbf{A}^t\}$, which can be regarded as a state transition matrix between codewords at each time point t. Therefore, the *state space model* (SSM) is one of natural but powerful framework to achieve the sequential prediction task [14]. For the task (I), we use the forward algorithm to compute the most probable state vector $\mathbf{x}_{i+k} \in \mathbb{R}^D$ for $k = \{1,\dots,K\}$; the d-th element of $\mathbf{x}_{i+k}$ indicates the probability that an image of codeword d occurs at time $i+k$. For each $\mathbf{x}_{i+k}$, we find out the best correspondent image from the ranking scores computed using the decoding method discussed in section 2. For the task (II), we obtain state vectors by running the forward-backward algorithm with EM, since the observation in the middle of the photo stream are missing. Then, the same decoding method is used to retrieve the most probable images.



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

Figure 2. The Flickr datasets of 24 classes of three categories. The number of images and photo streams are shown in (a) and (b), respectively. The dataset sizes are (3,320,080, 42,744) in total.

4. Experiments

We first evaluate reconstructed storyline graphs via user studies using Amazon Mechanical Turk. Then, we quantitatively compare the performance of our method for the two image prediction tasks with other candidate methods. The MATLAB code is available in our homepage for the better understanding of our algorithm.

4.1. Evaluation Setting

Flickr Dataset. Fig.2 summarizes our Flickr dataset that consists of about 3.3M of images of 42K photo streams for 24 classes, which are classified into three categories: outdoor recreational activities, holidays, and sports events. We use the topic names as search keywords and download all queried photo streams that contain more than 30 images with correct timestamps and user information.

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

Baselines. Since the storyline reconstruction is a novel task, there are few existing methods to be compared. Hence, we select and adapt the following three baselines that are not originally developed for the storyline reconstruction, but are appealing candidate methods to visualize the topic evolution of image collections, and perform the sequential prediction tasks. The first baseline, denoted by (Page), is a Page-Rank based image retrieval. It is one of most successful methods to retrieve a small number of canonical images,

but unable to model any structural information. We also implement the baseline (HMM) using the HMM, which is one of most popular frameworks for modeling tourists' sequential photo sets [2, 7]. The (Clust) is a clustering-based summarization on the timeline [8], in which images on the timeline are grouped into 10 clusters using K-means at every 30 minutes.

4.2. Results on Storyline Summarization

Task. It is inherently difficult to quantitatively evaluate the reconstructed storyline graphs due to the absence of groundtruth. Moreover, evaluation by human subjects is also hopelessly challenging because the storyline graphs are the summary of large image collections with possibly hundreds of vertices. For overcoming such difficulty, we take advantage of crowdsourcing-based evaluation via Amazon Mechanical Turk (AMT). The basic idea is to let each turker to compare between very small parts of the storylines built by our algorithm and baselines, and aggregate such crowd of assessments for the evaluation of the whole.

We first run our algorithm and baselines to generate storylines from the dataset of each class. We then sample 100 most canonical images from the dataset as test instances $\mathcal{I}_Q$. For each test instance $I_q \in \mathcal{I}_Q$, we localize the node v_q that includes I_q in the storyline graph of each algorithm. Then we find the node v_e that is most strongly connected to v_q, and obtain one central image I_e from the node v_e. For evaluation, we show I_q and a pair of images predicted by our algorithm and one of baselines, and ask a turker to choose one of them that is most likely to follow I_q. We design the AMT task as a pairwise preference test instead of a multiple choice test because it could be easier not only for turkers with all ranges of expertise levels but also for us to statistically analyze the responses. We obtain such pairwise comparison for each of $\mathcal{I}_q$ from at least three different turkers for the validity of AMT's annotations. In summary, the idea of our evaluation is to recruit a crowd of annotators to measure the preference of *each important edge*, instead of the whole storyline graphs, which is practically impossible.

Fig.3 shows examples of the predicted images by our algorithm and three baselines. In each set of preference tests, we show the *given* image, and a pair of images predicted our algorithm and one of baselines. In actual user studies, algorithm names are hidden, and image orders are shuffled.

Quantitative Results. Fig.4 shows the results of pairwise AMT preference tests between our method and the three baselines. The number indicates the mean percentage of responses that choose our prediction as a more likely one to come next after each I_q than that of the baseline. That is, the number should be higher than at least 50% to validate the superiority of our algorithm. Although the answer to "*What comes next?*" is rather subjective, and a certain level of noisiness of AMT's annotations are unavoidable, our algorithm significantly dominates the votes; for example, our algorithm (Ours) gains 66.5% of votes over the best baseline (HMM) in the average over 24 classes.

4.3. Results on Sequential Image Prediction

Task. In the second experiments, we evaluate our storyline graphs in the context of photo recommendation, which can be regarded as one foremost practical use of storylines. We perform the two image sequential prediction tasks: (I) predicting next likely images and (II) filling in missing parts of a photo stream. We first randomly select 80% of photo streams of each class as a training set and the others as a test set. Then we reduce each test photo stream into uniformly sampled 50 images, since consecutive images can be very similar in many long photo streams. For the task (I), we randomly divide the test photo stream into two disjoint parts. Then, the goal of each algorithm is, given the first part and next 10 query time points $\mathbf{t}_q = \{t_{q1}, \ldots, t_{q10}\}$, to retrieve 10 images that are likely to appear at $\mathbf{t}_q$ from the training set. The actual images of the test photo stream at $\mathbf{t}_q$ are used as groundtruths. Likewise, for the task (II), we randomly crop out 10 images in the middle of each test photo stream. Then, the algorithms predict the likely images for the missing part given the time points $\mathbf{t}_q$. We also perform experiments for the weakly-personalized prediction; the tests are the same only except that a pair of query user and month (u_q, m_q) of a test photo stream is given. Thus, the algorithms can leverage the month m_q and the friendship graph to figure out the friends of u_q. In summary, we examine more than 10K test instances in total to evaluate the performance of algorithms. The prediction quality is measured using peak signal-to-noise ratio (PSNR) between predicted and groundtruth images. Note that a higher value indicates that the two images are more similar. We describe more details of how to apply our method and baselines in the supplementary.

Quantitative Results. Fig.5 shows the quantitative comparison between our method and three baselines for task (I) and (II) with or without weak-personalization. The leftmost bar set is the average performance of 24 classes, and the PSNR values of individual classes follow. Our algorithm outperforms all the competitors in most topic classes for the both tasks. For example, in the average accuracies of normal prediction, our PSNR performance gains (in dB) over the best baseline (NET) are 0.61 and 0.52 (See the accurate numbers in the caption of Fig.5). Interestingly, the weakly-personalized prediction leads only a slight increase of prediction accuracies. It may be because the photo-taking behaviors between neighbors in the user graph are not always similar one another. The friendship graph is built from users' Flickr group memberships, and thus many query users are likely to be skilled photographers that have their own unique styles.

Figure 3. Examples of the images predicted by our algorithm and three baselines. In each preference test via AMT, the task is to select the best one that is likely to occur next after the *given* image among a pair of images predicted by our algorithm and one of baselines.

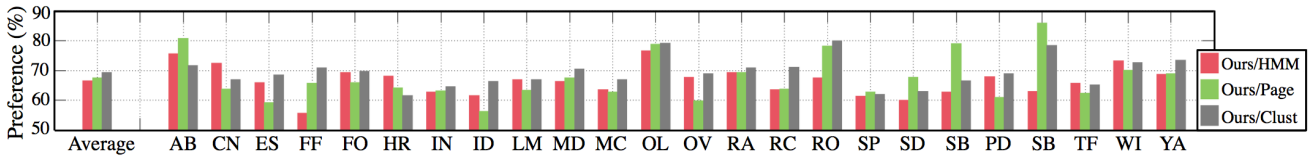


Figure 4. The results of pairwise preference tests via AMT between our method (`Ours`) and three baselines. The numbers indicates the percentage of responses that our prediction is more likely to occur next after I_q than that of the baseline. At least the number should be higher than 50% to validate the superiority of our algorithm. The leftmost bar set shows the average preference of our method (`Ours`) for all 24 classes: [**66.5**, **67.5**, **69.4**] over (`HMM`), (`Page`), and (`Clust`). The acronyms of classes are referred to Fig.2.

Fig.6 shows some selected examples produced by different algorithms for the prediction tasks. In the first row of Fig.6.(a)–(d), we show two sampled *given* images and positions of images to be predicted (*i.e.* $\{I_{q1}, \ldots, I_{q5}\}$). Then, we show the hidden groundtruth images in the second row, and predicted images by our algorithm and three baselines in the following rows. Since training and test sets are disjoint, each algorithm can only retrieve similar (but not identical) images from training data at best. The (`HMM`) retrieves reasonably good but highly redundant images, which are in part due to its inability to represent various branching structures. The (`Clust`) baseline prefers temporally-connected images from the largest clusters on the timeline, and the performance is not as good as ours. The (`Page`) simply retrieves the top-ranked (*i.e.* representative high-quality) images at each query time point. Due to lack of use of the sequential information, there is no connected story between the predicted images. Fig.6.(e)–(g) show downsized versions of our storyline graphs that are used for the prediction tasks of Fig.6.(a),(b),(d), respectively. Although we here show simplified graphs only, it is possible to illustrate the storyline graphs in various ways, some of which will be presented in the supplementary.

5. Conclusion

We proposed an approach for reconstructing storyline graphs from large sets of photo streams available on the Web. With experiments on more than three millions of Flickr images for 24 classes and user studies via AMT, we validated that our scalable algorithm can successfully create storyline graphs as an effective structural summary of large-scale and ever-growing image collections. We also quantitatively showed the excellence of our storyline graphs for the two prediction tasks over other candidate methods.

Acknowledgement: This work is supported in part by NSF IIS-1115313, AFOSR FA9550010247, Google, and Alfred P. Sloan Foundation.

References

[1] A. Ahmed, Q. Ho, J. Eisenstein, E. P. Xing, A. J. Smola, and C. H. Teo. Unified Analysis of Streaming News. In *WWW*, 2011. 2

[2] C.-Y. Chen and K. Grauman. Clues from the Beaten Path: Location Estimation with Bursty Sequences of Tourist Photos. In *CVPR*, 2011. 2, 6

[3] K. Crammer and Y. Singer. On the Learnability and Design of Output Codes for Multiclass Problems. *Machine Learning*, 47:201–233, 2002. 3

[4] W. J. Fu. Penalized Regressions: The Bridge Versus the Lasso. *J. Computational Graphical Statistics*, 7:397–416, 1998. 4

[5] J. Gillenwater, A. Kulesza, and B. Taskar. Discovering Diverse and Salient Threads in Document Collections. In *EMNLP*, 2012. 2

[6] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding Videos, Constructing Plots: Learning a Visually Grounded Storyline Model from Annotated Videos. In *ICCV*, 2009. 2

[7] E. Kalogerakis, O. Vesselova, J. Hays, A. Efros, and A. Hertzmann. Image Sequence Geolocation with Human Travel Priors. In *ICCV*, 2009. 2, 6

[8] G. Kim and E. P. Xing. Jointly Aligning and Segmenting Multiple Web Photo Streams for the Inference of Collective Photo Storylines. In *CVPR*, 2013. 2, 6

[9] G. Kim, E. P. Xing, and A. Torralba. Modeling and Analysis of Dynamic Behaviors of Web Image Collections. In *ECCV*, 2010. 2

[10] M. Kolar, L. Song, A. Ahmed, and E. P. Xing. Estimating Time-Varying Networks. *Ann. Appl. Stat.*, 4(1):94–123, 2010. 3

[11] J. M. Mandler and N. S. Johnson. Remembrance of Things Parsed: Story Structure and Recall. *Cognitive Psychology*, 9(1):111–151, 1977. 1

[12] N. Meinshausen and P. Bühlmann. High-Dimensional Graphs and Variable Selection with the Lasso. *Ann. Statist.*, 34(3):1436–1462, 2006. 4

[13] H. Misra, F. Hopfgartner, A. Goyal, P. Punitha, and J. M. Jose. TV News Story Segmentation Based on Semantic Coherence and Content Similarity. In *MMM*, 2010. 2

[14] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002. 5

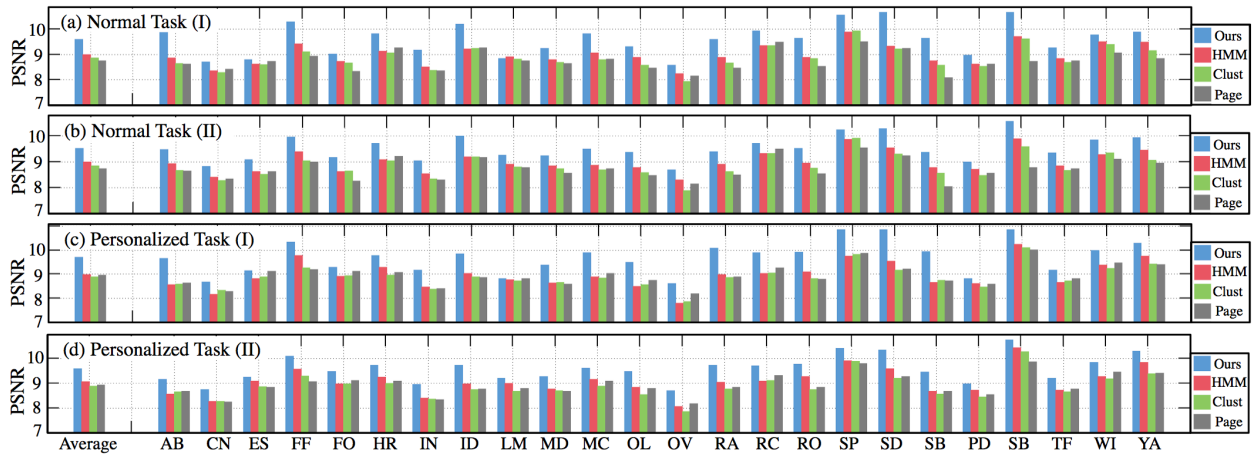


Figure 5. Results of our method and three baselines for the task (I) (*i.e. predicting likely next images*) and the task (II) (*i.e. filling in missing parts*) with or without the *week-personalization*. The average PNSR in the left-most bar set are [ours, (HMM), (Clust), (Page)] = [**9.60**, 8.99, 8.86, 8.75] for (a), [**9.53**, 9.01, 8.85, 8.75] for (b), [**9.70**, 8.97, 8.89, 8.96] for (c), and [**9.57**, 9.05, 8.87, 8.93] for (d).

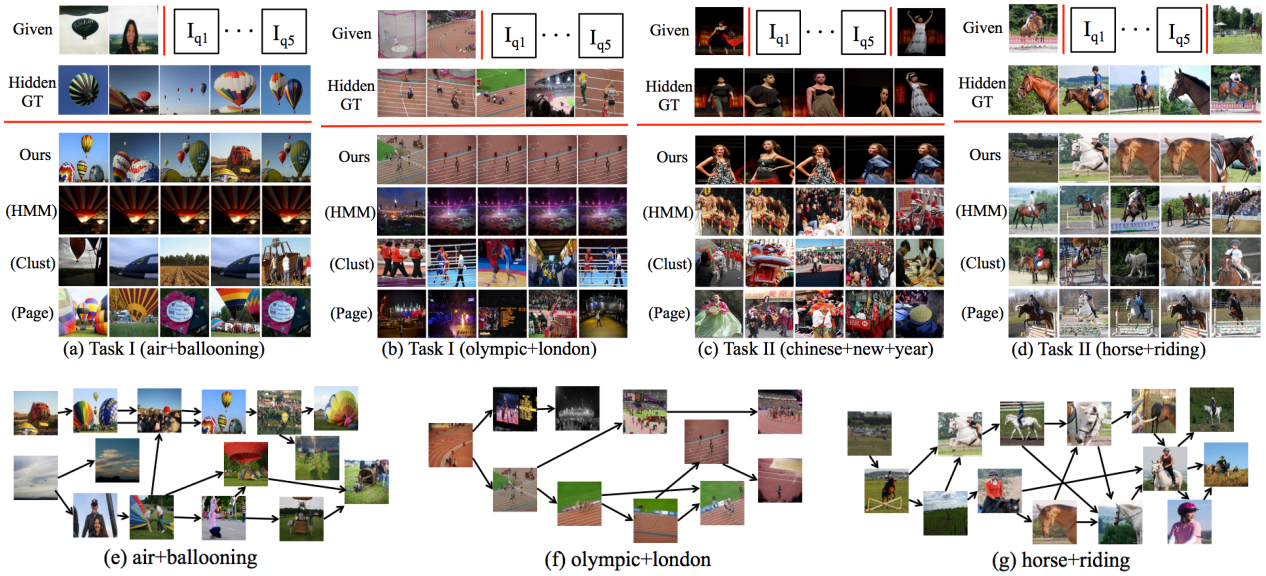


Figure 6. Examples of results for the two prediction tasks: (a)–(b) for task (I), and (c)–(d) for task (II). The goal of each algorithm is to predict likely images for $\{I_{q1}, \ldots, I_{q5}\}$ using its storylines and *given* images in the first row. In each set, we show hidden groundtruth images in the second row and the predicted images by different algorithms in the other rows. We also present downsized versions of our storyline graphs in (e)–(g), which are used for the prediction tasks of (a),(b), and (d), respectively.

[15] P. Obrador, R. de Oliveira, and N. Oliver. Supporting Personal Photo Storytelling for Social Albums. In *MM*, 2010. 1, 2

[16] B. A. Olshausen and D. J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, 37(23):3311–3325, 1997. 3

[17] M. O. Riedl and R. M. Young. From Linear Story Generation to Branching Story Graphs. *IEEE Computer Graphics and Applications*, 26(3):23–31, 2006. 1

[18] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the Temporal Dynamics of Diffusion Networks. In *ICML*, 2011. 4

[19] D. Shahaf and C. Guestrin. Connecting the Dots Between News Articles. In *KDD*, 2010. 2

[20] D. Shahaf, C. Guestrin, and E. Horvitz. Trains of Thought: Generating Information Maps. In *WWW*, 2012. 2

[21] N. Snavely, I. Simon, M. Goesele, R. Szeliski, and S. M. Seitz. Scene Reconstruction and Visualization from Community Photo Collections. *Proceedings of the IEEE*, 98(8):1370–1390, 2010. 2

[22] L. Song, M. Kolar, and E. Xing. Time-Varying Dynamic Bayesian Networks. In *NIPS*, 2009. 2, 3, 4

[23] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood Formation and Anomaly Detection in Bipartite Graphs. In *ICDM*, 2005. 5

[24] A. Torralba, R. Fergus, and W. T. Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE PAMI*, 30:1958–1970, 2008. 3

[25] D. Wang, T. Li, and M. Ogihara. Generating Pictorial Storylines Via Minimum-Weight Connected Dominating Set Approximation in Multi-View Graphs. In *AAAI*, 2012. 2

[26] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In *CVPR*, 2010. 3