Time-Sensitive Web Image Ranking and Retrieval via Dynamic Multi-Task Regression

Gunhee Kim
Computer Science Department
Carnegie Mellon University
Pittsburgh, 15213 PA
gunhee@cs.cmu.edu

Eric P. Xing
Machine Learning Department
Carnegie Mellon University
Pittsburgh, 15213 PA
epxing@cs.cmu.edu

ABSTRACT

In this paper, we investigate a time-sensitive image retrieval problem, in which given a query keyword, a query time point, and optionally user information, we retrieve the most relevant and temporally suitable images from the database. Inspired by recently emerging interests on query dynamics in information retrieval research, our time-sensitive image retrieval algorithm can infer users' implicit search intent better and provide more engaging and diverse search results according to temporal trends of Web user photos. We model observed image streams as instances of multivariate point processes represented by several different descriptors, and develop a regularized multi-task regression framework that automatically selects and learns stochastic parametric models to solve the relations between image occurrence probabilities and various temporal factors that influence them. Using Flickr datasets of more than seven million images of 30 topics, our experimental results show that the proposed algorithm is more successful in time-sensitive image retrieval than other candidate methods, including ranking SVM, a PageRank-based image ranking, and a generative temporal topic model.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithm, Experimentation

Keywords

Web image retrieval, Point process, Multi-task regression

1. INTRODUCTION

As digital images are gaining popularity as a form of communicating information online, image search and retrieval has become an indispensable feature in our daily Web uses. Most commercial Web image search engines such as Bing, Google, and Yahoo largely rely on the text-based approach [5], in which given a query keyword, relevant pictures are retrieved and ranked by matching textual information of images such as surrounding texts, titles, or captions. Although the text-based image search has been successful as an effective and scalable image retrieval approach, it suffers from ambiguous and noisy results due to the mismatch between images and their surrounding texts. Moreover, it is still limited to correctly exploit visual contents of images and identify implicit or explicit search intent of a user.

In this paper, we study one additional aspect to improve image search quality: *temporal dynamics of image collections*. In other words, given Web image collections associated with keywords of interest, we aim at identifying their characteristic temporal patterns of occurrences on the Web, and leveraging them to improve search relevance at a query time. This problem is closely related to one recent emerging research in information retrieval: *exploring the temporal*

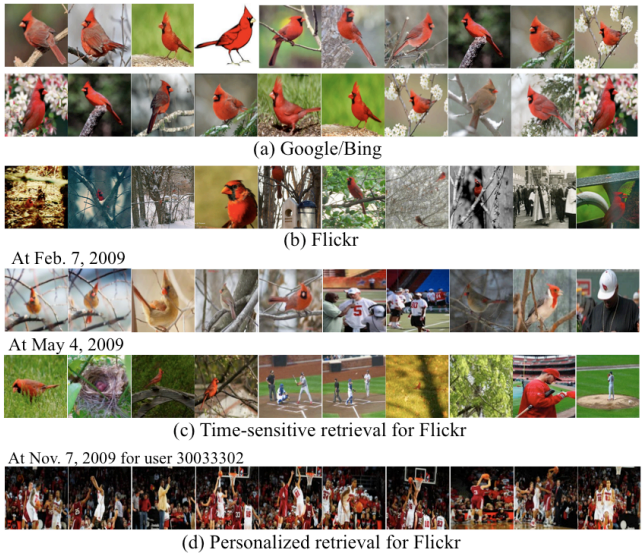


Figure 1: Overview of time-sensitive Web image ranking and retrieval with a query example of the *cardinal*. (a)-(b) Top ten images retrieved by Google/Bing and Flickr search engines at 7/31/2012. (c) The results of our time-sensitive image retrieval for two query time points in winter and summer. (d) The result of our personalized image retrieval for a designated time and user.

WSDM'13, February 4–8, 2013, Rome, Italy.

dynamics of Web queries to improve search relevance [6, 17, 20, 22]. Many queries are time-sensitive; the popularity of a query and its most relevant documents change over time. For example, a statistical analysis of Web query logs in [20] reported that more than 7% of queries have implicitly temporal intents (*e.g.* NBA, Olympics). This new area of research has cast a variety of interesting research questions, for example, identifying search terms that are sensitive to time, and reranking documents according to the query time. However, much of previous work has targeted at the search of text documents such as blogs and news archives by analyzing the query log data; the time-sensitive Web image retrieval has yet received little attention.

With our experiments on more than seven million Flickr images, we have found three good reasons why the discovery of *temporal* patterns in Web image collections is beneficial to existing image retrieval systems. We present them with a query example of the *cardinal* in Fig.1. *First, knowing when search takes place is useful to infer users' implicit intents.* Fig.1.(a) shows the top ten images retrieved by Google and Bing image search engines. Seemingly, they are reasonable because the *cardinal* usually refers to the red bird in America. However, the term *cardinal* is polysemous; it is also the names of popular sports teams (*e.g.* the American football and the baseball team). Therefore, some of *cardinal* queries in summer and winter are likely to be associated with the baseball and football team, respectively, according to the scheduled seasons of the sports.

Second, the timing suitability can be used as a complementary attribute to relevance. Fig.1 illustrates two such cases: One is that, as shown in Fig.1.(a), due to explosion of images shared on the Web, there are redundantly relevant images to popular queries like the *cardinal*. Timing suitability would be a good complementary ranking attribute to improve diversity or break ties between almost equally relevant images. The other case is that, in Fig.1.(b), the actual user images in the photo-sharing site Flickr are extremely diverse, and thus it is still very challenging to rank those images in any meaningful order. As shown in Fig.1.(c), the query time information can help obtain a more focused search output, which may include the images about *a cardinal bird in snowy field* in winter, but the images of *baby cardinals or eggs* in summer.

Third, temporal information is synergetic in personalized image retrieval. If a query word has a broad range of concepts, its dominant usages vary much according to users. Our experiments show that once we can identify a user's preference, image retrieval can be further specific since the term usages of individual users are relatively stationary. For example, as shown in Fig.1.(d), if a user took or searched *cardinal* pictures a lot for a basketball team last winter, he tends to do the same this winter as well.

Problem Statement: As an input, we gather a large-scale pool of unordered Web images along with metadata (*e.g.* timestamps, owners) by querying Q topic keywords from a text-based image search engine. We use raw Flickr images since our time-sensitive image retrieval is more interesting for extremely diverse general users' photos rather than sufficiently cleaned-up Google or Bing images. In this paper, our objective is two-fold: Our first goal is to automatically identify the temporal properties of each topic keyword, because every topic is not necessarily time-sensitive, and has its own characteristic temporal behaviors. The second goal is to leverage the learned temporal models to rank the images in database according to temporal suitability when a topic keyword and a query time are given. In real scenarios, the query time is usually *now* (*i.e.* the time when the search takes place), but we assume that it can be *any time even in future* for generality. We also address the *personalized* time-sensitive image ranking, which is customized image retrieval for a designated user.

Proposed method: The two objectives are accomplished by a unified statistical model: regularized multi-task regression on multivariate point process. We view an observed image stream as an instance of multivariate point process, which is a stochastic process that consists of a series of random events occurring at points in time and space [7]. Then, we automatically test what temporal models or their combinations are the best to describe the image occurrence behaviors, and formulate a regression problem to learn the historical relations between image occurrence probabilities and various temporal factors or covariates that influence them (*e.g.* seasons, dates, and other external events). From the learned models, we can easily compute the ranking scores of images for any given time point. We explore the idea of multi-task learning to incorporate multiple types of image representation for a more accurate ranking. Consequently, our algorithm offers several important advantages for large-scale image retrieval as follows: (i) *Flexibility*: We can easily build a set of parametric models to capture any number of possible temporal behaviors of image collections, and automatically choose the most statistically suitable ones (Section 4.1). We can achieve a globally optimal or approximate solution to the learning of temporal models (Section 4.4). (ii) *Scalability*: The learning is performed offline once, and the online query step is very fast. Both processes run in a linear time with most parameters such as time steps and the number of image descriptors (Section 5.3). (iii) *Retrieval accuracy*: We perform experiments on more than seven millions of Flickr images over a wide range of 30 topic keywords. We demonstrated that our image retrieval algorithm outperforms other candidate methods including Ranking SVM [14], a PageRank-based image retrieval [13, 16] and a generative author-time topic model [23] (Section 6).

1.1 Relations to Previous work

The time-sensitive image retrieval for large-scale Web photo collections remains an under-addressed topic in information retrieval literature. Our work is remotely related to following two lines of research, but is significantly different on the task, utility and methodology.

Image retrieval and reranking: Recently, image reranking has been actively studied to improve text-based image search by leveraging visual or user feedback information [5, 13, 19, 21, 29, 31]. Most image reranking methods have exploited three sources of information, which are human-labeled training data [31], user relevance feedback [5, 29], and pseudo-relevance feedback [21]. Given an image database retrieved by text-based search, the user relevance feedback approach asks a user to select a query image to clarify her search intent. The pseudo-relevance feedback approach assumes the top images retrieved by text-based search as pseudo-positive examples and bottom ranked images as pseudo-negative examples. Once the training data are obtained, almost all existing methods learn ranking models relying on the semantic meaning of a query word and the

feature-wise image similarity, beyond which our approach additionally emphasizes the temporal trends and user history associated with the images. The time-sensitive retrieval can be becoming more important and anticipated, given that the majority of Web photos are now coming from hundreds of millions of general users with different experiences.

Web image dynamics: This line of research aims at modeling how the contents of large-scale Web image collections change over time [12, 15, 16, 24]. In [12] and [24], Flickr images are analyzed to infer other phenomena in user behaviors such as social trends in politics and markets [12] and spatio-temporal events [24]. In [16], the topic evolution is modeled to perform the subtopic outbreak detection and the classification of noisy web images. However, the study of Web content dynamics has not yet contributed much to solve the image ranking and retrieval, which is the main objective of this paper. Also, they did not explore any issues regarding personalization, as we do in this work. The most related work to ours may be [15], whose task is to predict future likely images in Flickr datasets. However, our work overcomes their two critical limitations to be a practical time-sensitive image retrieval method. First, their approach has no mechanism to test which queries are actually time-sensitive and to discover query-specific temporal properties, even though all submitted queries are not necessarily time-sensitive. Therefore, all topics are equally treated, which may cause degrading performance for some query classes. Second, they oversimplified the image representation by assuming that millions of images can be clearly clustered into 500 clusters. This assumption leads their approach to be an incomplete image ranking since it lacks a way of ranking the images in the same cluster.

1.2 Summary of Contributions

Departing from the literature reviewed above, the main contributions of our work can be summarized as follows:

(1) We develop an approach for time and optionally user sensitive image ranking and retrieval. To the best of our knowledge, there have been few attempts so far on such retrieval methods that leverage the temporal aspects of large-scale Web photo collections.

(2) We design our image retrieval algorithm using multi-task regression on multivariate point processes. Consequently, our algorithm can automatically select and learn stochastic temporal models while satisfying a number of key challenges of Web image ranking, including flexibility, scalability, and retrieval accuracies.

2. META-DATA OF IMAGES

We assume that each of input Flickr images is assigned to topic keywords, timestamp, and owner ID. In addition to such meta-data from Flickr, we extract two types of information modalities: image description and user description.

2.1 Image Descriptions

In this paper, we extract four different image descriptors because no single descriptor can completely capture various contents of an image, and thus leveraging multiple descriptors is a widely accepted common practice in recent computer vision research. The four descriptors that are explained below can be classified into two low-level (SIFT and HOG) and two high-level descriptors (Tiny and Scene), all of which are extracted by using publicly available codes[1].

Color SIFT (SIFT): We densely extract HSV color SIFT on a regular grid at steps of 4 pixels. We form 300 visual words by applying K-means to randomly selected SIFT descriptors. The nearest word is assigned to every SIFT, and binned using a three-level spatial pyramid.

HOG2x2 (HOG): We also use the histogram of oriented edge (HOG) feature, inspired by its recent success in object detection research [9]. We extract HOG descriptors on a regular grid at steps of 8 pixels by following the method called HOG2x2 in [30].

Tiny Image: Inspired by [27], we resize an image to a 32×32 tiny color image, and use pixel values as features. This approach not only reduces image dimensionality to be computationally feasible, but also is discriminative enough to convey high-level statistics of an image.

Scene description: Since a large portion of Web images contain scenes, the scene classifier outputs can be a meaningful high-level description of an image. SUN database [30] is an extensive dataset of 397 scene categories. As a scene descriptor, we compute the scores of linear one-vs-all SVM classifiers for 397 scene categories using Hog2x2 features, by following the classification benchmark protocol in [30].

Visual clusters: Since all the above descriptors except (Scene) are high-dimensional (*e.g.* 6,300 of (SIFT)), they are down-sampled further by the soft-assignment idea. For each descriptor type, we construct $L(=300)$ *visual clusters* by applying K-means to randomly sampled image descriptors. Then, an image I is assigned to r-nearest visual clusters for each descriptor type with the weights of an exponential function $\exp(-d^2/2\sigma^2)$, where d is the distance between the descriptor and the visual cluster and σ is a spatial scale. Finally, an image I is described by four ℓ-1 normalized vectors with only r nonzero weights. They are denoted by $\{\mathbf{h}_k(I)\}_{k=1}^4$ with dimensions of $[L_k]_{k=1}^4 = [300\ 300\ 300\ 374]$.

2.2 User Description

Clustering users and measuring similarity between users are important for personalization in collaborative filtering [8]. Its basic assumption is that similar users are likely to share common photo taking and search behaviors. We use the pLSA (Probabilistic latent semantic analysis) clustering as proposed in Google News personalization [8]. We first choose a fixed number of top users who have uploaded images most, and compute an L-dimensional histogram for each user where each bin represents the count of images belonging to the corresponding visual cluster. In pLSA, the distribution of visual cluster v in user u_i's images, $p(v|u_i)$, is given by the following generative model:

$$p(v|u_i) = \sum_{z \in \mathcal{Z}} p(v|z)p(z|u_i). \quad (1)$$

The latent variable $z \in \mathcal{Z}$ represents the cluster of user propensity. Thus, $p(z|u_i)$ is proportional to the fractional membership of user i to cluster z. We use $p(z|u_i)$ as the descriptor of user u_i. The user clustering can be done by grouping users with the same $z^* = \text{argmax}_z\, p(z|u_i)$ or run K-means on the user descriptors $p(z|u_i)$. The user similarity is calculated by histogram intersection on user descriptors.

[1]We use following codes: (SIFT) at http://www.vlfeat.org, (HOG) at http://www.cs.brown.edu/∼pff/latent, (Tiny) and (Scene) at http://people.csail.mit.edu/jxiao/SUN/.

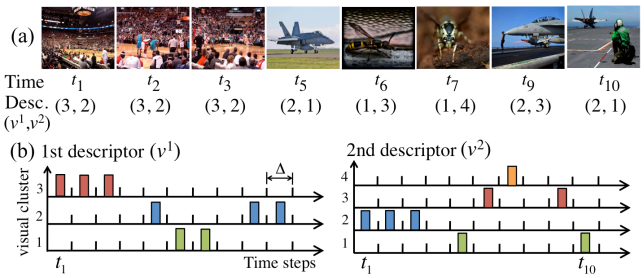


Figure 2: A multivariate point process for a short image stream of the *hornet*. (a) Each image is assigned to a timestamp and visual clusters of two different descriptors ($K = 2$, $L_1 = 3$, $L_2 = 4$). (b) The image stream is modeled by two multivariate discrete-time point processes.

3. MULTIVARIATE POINT PROCESSES

In this section, we discuss the mathematical background of multivariate point process for modeling Web photo streams. Fig.2 shows a toy example for a short image stream of the *hornet*. Suppose that we extract K image descriptors from each image, and for each descriptor, we cluster the images into L_k visual clusters (In this example, $K = 2, L_1 = 3, L_2 = 4$). Intuitively, one can easily construct K multivariate point processes as shown in Fig.2.(b). For simplicity, we first assume that the occurrence of each visual cluster is independently modeled. Hence, the point process of Fig.2 can be regarded as a single multivariate point process with $L = L_1 + L_2 = 7$. In section 4.3, we will consider an extended multi-task framework with considering correlations between different descriptors.

Intensity functions: Since the intensity function can completely define a point process [7], we first introduce its definition. Formally, a multivariate point process can be described by a counting process $\mathbf{N}(t) = (N^1(t), \cdots, N^L(t))^T$ where $N^l(t)$ is the total number of observed images assigned to visual cluster l in the interval $(0, t]$. Then, $N^l(t + \Delta) - N^l(t)$ represents the number of images in a small interval Δ. By letting $\Delta \to 0$, we obtain the *intensity function* at t, which is the infinitesimal expected occurrence rate of visual cluster l at time t [7]:

$$\lambda^l(t) = \lim_{\Delta \to 0} \frac{P[N^l(t+\Delta) - N^l(t) = 1]}{\Delta}, \; l \in \{1, \ldots, L\}. \quad (2)$$

Generalized Linear Model: We assume that the intensity function $\lambda^l(t)$ is represented by the covariates that influence the occurrence of visual cluster l. We define the the parametric form of $\lambda^l(t_i|\theta^l)$ as the exponential of a linear summation of the functions f_j^l of the covariates x_j with a parameter vector $\theta^l = (\theta_1^l, \cdots, \theta_J^l)$:

$$\log \lambda^l(t_i|\theta^l) = \sum_{j=1}^{J} \theta_j^l f_j^l(x_j), \quad l \in \{1, \ldots, L\}. \quad (3)$$

Data likelihood: Suppose that we partition the interval $(0, T]$ by a sufficiently large number M (*i.e.* $\Delta = T/M$) so that in each time bin Δ only one or zero image occurs. Then, we can denote the sequence of images up to T by $N_{1:M}^l = n_1^l \cdots n_M^l$ with $n_i^l \in \{0, 1\}$. It is shown in [28] that the likelihood of such a point process along with λ^l of Eq.(3) is identical to that of the *Poisson regression*. Therefore, the log-likelihood of an observed image sequence is

$$\ell(N_{1:M}^l|\theta^l) = \sum_{i=1}^{M} \Big(n_i \lambda^l(t_i|\theta^l) - \exp(\lambda^l(t_i|\theta^l)) - \log n_i! \Big). \quad (4)$$

L1 regularized likelihood: Although numerous factors or covariates can be plugged in Eq.(3), each visual cluster is likely to depend on only a small subset of them. Hence, it is important to detect a few strong covariates by encouraging a sparse estimator of θ^l for each visual cluster l. This approach is also practical because we usually do not know what factors are important beforehand; we safely include as many candidate factors as possible, and then choose only a few covariates for each visual cluster via MLE learning. Therefore, we introduce *Lasso* penalty [25] into the likelihood of Eq.(4) with a regularization parameter μ controlling sparsity level:

$$\ell_L(N_{1:M}^l|\theta^l) = \ell(N_{1:M}^l|\theta^l) - \mu \sum_{j=1}^{J} |\theta_i^l|. \quad (5)$$

4. TEMPORAL MODELING OF PHOTO STREAMS

Our first objective is to identify the temporal properties of a given image stream. This goal is achieved via the learning of temporal models as follows. We first represent the image stream with a multivariate point process $\{N_{1:M}^l\}_{l=1}^L$ as described in previous section. Then, we define multiple models for $\lambda^l(t_i|\theta^l)$ by enumerating all possible temporal factors that influence the image occurrences (section 4.1). Finally, for each occurrence data $N_{1:M}^l$ of visual cluster l, we select a subset of most statistically plausible models (section 4.2), and learn the parameters θ^{l*} of the models to discover which factors are actually contributing (section 4.4). Note that the whole processes above can be automatically performed.

4.1 Models of Temporal Behaviors

In this section, we enumerate a set of models for the intensity functions, each of which is designed to capture a particular temporal property. Thanks to the flexibility of our framework, one can freely add or remove such models according to the characteristics of image topics unless they contradict the definition of Eq.(3). In this paper, we construct two groups of models: *temporal attributes* and traditional *time series*.

Temporal attributes: Human time perception and photo taking and search behaviors are not only continuous on time but also temporal attribute-driven. For example, *zoo* photos may be more frequently taken in weekend rather than in weekdays, or *ski* images appear more often in January than in June. Therefore, we build a set of intensity function models for temporal attribute-driven covariates as follows.

$$\log \lambda_y^l(t_i|\alpha^l) = \alpha_0 + \sum_{y=Y_s}^{Y_t} \alpha_y I_y(t_i) \quad (6)$$

$$\log \lambda_m^l(t_i|\beta^l) = \beta_0 + \sum_{t=1}^{12} \beta_t g(t_i - t) \quad (7)$$

$$\log \lambda_d^l(t_i|\gamma^l) = \gamma_0 + \sum_{t=1}^{12} I_t(t_i) \sum_{d=1}^{31} \gamma_{m,d} I_d(t_i) \quad (8)$$

$$\log \lambda_w^l(t_i|\zeta^l) = \zeta_0 + \sum_{w \in \{M, \ldots, S\}} \zeta_w I_w(t_i) \quad (9)$$

$$\log \lambda_h^l(t_i|\eta^l) = \eta_0 + \sum_{h \in \mathcal{H}} \eta_h I_h(t_i). \quad (10)$$

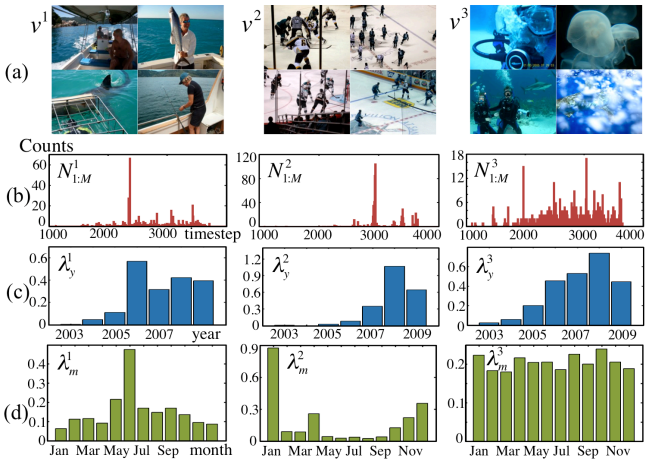


Figure 3: Examples of intensity function models of years and months for three visual clusters (VC) of the *Shark*: v^1(*sea tour*), v^2(*ice hockey*), v^3(*diving in aquarium*). (a) Four images sampled from each VC. (b) Observed image occurrences. (c)-(d) Estimated intensity functions for years and months. λ^1_m and λ^2_m have different image occurrence rates peaked in summer and winter, respectively. λ^3_m is stationary along the timeline.

In equations, λ^l_y, λ^l_m, λ^l_d, λ^l_w, and λ^l_h are the models of intensity functions for years, months, days, weekdays (from Monday to Sunday), and holidays[2], whose lists are denoted by $\mathcal{H}$. The parameter set to be learned comprises $\{\alpha^l, \beta^l, \gamma^l, \zeta^l, \eta^l\}$. $I_y(t_i)$ is an indicator function that is 1 if the year of t_i is y, and 0 otherwise (*e.g.* $I_y(t_i) = 1$ if $y = 2008$ and t_i= 6/3/2008). Similarly, $I_w(t_i)$ and $I_h(t_i)$ are indicators for week and holidays. For month covariates, we use Gaussian weighting $g(t_i - t) \propto \exp(-(t_i - t)^2/\sigma)$, which leads that if an image occurs in May, for example, some contributions are also given on nearby months like April and June, assuming that images smoothly change on the timeline.

Here, our models are mainly built based on calendric temporal attributes, but the models driven by other textual or social factors (*e.g.* news articles) can be supplemented.

An example: Fig.3 is a toy example of the *shark* topic to intuitively show how the intensity function models are used for fitting observed image streams. This example illustrates the intensity function models for years (λ^l_y of Eq.(6)) and months (λ^l_m of Eq.(7)), where the parameter set comprises seven $[\alpha^l_y]^{2009}_{y=2003}$ and twelve $[\beta^l_m]^{12}_{m=1}$. Fig.3.(a) shows four sampled images from three visual clusters, each of which approximately corresponds to *sea tour*, *ice hockey*, *diving in aquarium*. Fig.3.(b) presents their actual occurrence sequences. Fig.3.(c)-(d) show the learned intensity functions λ^l_y and λ^l_m. Most of intensity functions for years roughly increase every year because the number of uploaded photos in Flickr grows yearly. Interestingly, the visual clusters show different monthly behaviors in Fig.3.(d). λ^1_m has a higher intensity value (*i.e.* more frequently occurred) in June, λ^2_m peaks around January, and λ^3_m is stationary all year long. This result is reasonable because sea tours are popular in summer, the ice hockey season takes place during winter, and visiting aquarium is favored regardless of season. The learned intensity functions can be used for a simple time-sensitive image retrieval. For example, if the month of the query time t_q is January, then $\lambda^2_m(t_q) \gg \lambda^3_m(t_q) > \lambda^1_m(t_q)$. Therefore, we can rank the images from v^2 as the highest.

Autoregression: The other group of temporal models is based on autoregression, which is one of most popular models for the analysis of time series. We assume that the occurrence of each visual cluster is affected by its own history in Eq.(11), and the history of other visual clusters in Eq.(12). The first history model is represented by a linear autoregressive process:

$$\log \lambda^l_a(t_i|\phi^l) = \phi_0 + \sum_{p=1}^{P_d} \phi_p \Delta N^l_{i-dp} + \sum_{p=1}^{P_w} \phi_p \Delta N^l_{i-wp} + \sum_{p=1}^{P_m} \phi_p \Delta N^l_{i-mp} \quad (11)$$

where ΔN^l_{i-dp} denotes the occurrence counts of visual cluster l during $[t_i - dp, t_i)$, and d is the time window width. In Eq.(11), we use three different time windows: $d = 1$ day, and w =1 week, and $m = 1$ month. That is, λ^l_a is modeled by three different time-scaled (daily, weekly, and monthly) regressors whose orders are P_d, P_w, and P_m, respectively. The history model can capture the dynamic behavior of a visual cluster such as periodic or bursty occurrences.

The second correlation model represents the influence from the history of other visual clusters. Its mathematical form is almost identical to that of Eq.(11):

$$\log \lambda^l_c(t_i|\psi^l) = \psi_0 + \sum_{c=1, c\neq l}^{L} \left(\sum_{q=1}^{Q_d} \psi^{lc}_q \Delta N^l_{i-dq} + \sum_{q=1}^{Q_w} \psi^{lc}_q \Delta N^l_{i-wq} + \sum_{q=1}^{Q_m} \psi^{lc}_q \Delta N^l_{i-mq} \right) \quad (12)$$

The parameter set consists of $(L-1) \times (Q_d + Q_w + Q_m) + 1$ number of ψ in the full model. For fast computation, instead of using the full pairwise model, we can learn the correlations with respect to some selected most frequent visual clusters. This correlation model is useful when the existence or absence of a particular visual cluster can give a strong clue for others' occurrences.

4.2 Model Selection

In previous section, we introduce rather exhaustive seven temporal models from Eq.(6) to Eq.(12). However, the occurrence of each visual cluster does not necessarily depend on all the above models. For example, the occurrence of the *ice hockey* visual cluster v^2 of Fig.3 can be explained sufficiently well by the month intensity function model λ^l_m while other models may not be required any further. Therefore, we perform a model selection procedure, to choose a subset of temporal models by removing the ones with little or no predictive information.

Algorithm 1 summarizes the overall procedure of our model selection. It is based on the well-known *greedy forward selection* scheme, in which we keep increasing models one by one by adding at each step the one that increases the goodness-of-fit score the most, until any further addition does not increase the score. As the goodness-of-fit test, we use Kolmogorov-Smirnov (KS) test using time-rescaling theorem [2], which is one of most popular approaches for statistical model assessment in point process literature. The KS statistic is a quantitative measure for the agreement between

[2]We use the lists at http://vpcalendar.net/Holiday_Dates/.

Algorithm 1: Model selection for each visual cluster

Input: (a) A set of intensity function models in Eq.(6)–(12): $\Lambda^l = \{\lambda_y^l, \lambda_m^l, \lambda_d^l, \lambda_w^l, \lambda_h^l, \lambda_a^l, \lambda_c^l\}$. (b) $N_{1:M}^l$: Occurrence data of visual cluster l.

Output: The best intensity function model λ^{l*} with learned parameter set θ^{l*}.

1: Let $\theta_i \leftarrow$ param_est $(\lambda_i, N_{1:M}^l)$ be the function that computes the MLE solution of parameters for a given λ_i and $N_{1:M}^l$. This will be discussed in section 4.4.
2: For λ_a^l and λ_c^l, decide the AR orders using AIC measure: $AIC(P) = -2\log \ell(N_{1:M}^l|\theta_t) + 2P$ where P is the total number of parameters.
foreach $\lambda_i \in \Lambda^l$. **do**
  3: Compute $\theta_i \leftarrow$ param_est $(\lambda_i, N_{1:M}^l)$.
  4: Compute KS static d_i by applying time rescaling theorem to the learned $\lambda_i(\theta_i)$ and $N_{1:M}^l$.
5: Sort λ_i in an increasing order. Let o to be this order. Initialize $\lambda^{l*} = \text{argmin}_{\lambda_i \in \Lambda^l} d_i$ and $d^{l*} = \min d_i$.
repeat
  foreach $\lambda_i \in \Lambda^l$ *and* $\lambda_i \notin \lambda^{l*}$ *in the order of* o. **do**
    6: Set $\lambda_t = \lambda_i^{l*} \cdot \lambda_i$. $\theta_t \leftarrow$ param_est $(\lambda_t, N_{1:M}^l)$.
    7: Compute KS static d_t as done in step **4**.
    if $d_t < d^{l*}$ **then** $\lambda^{l*} \leftarrow \lambda_t$, $d^{l*} = d_t$, and $\theta^{l*} = \theta_t$.
until λ^{l*} *is not updated*;

Algorithm 2: Build the correlation set $\mathcal{E}$.

Input: (1) A set of images $\mathcal{I}$, each of which is assigned to the closest visual clusters of K descriptors.

Output: The correlation set $\mathcal{E}$.

1: Initialize $K(K-1)/2$ number of co-occurrence matrices $\mathcal{C}$, where $\mathbf{C}_{ab} \in \mathcal{C}$ is an $(L_a \times L_b)$ zero matrix between descriptor a and b.
foreach $I \in \mathcal{I}$. *Let visual cluster of* I *be* $(v_1, \ldots, v_K)$ **do**
  foreach $a, b \in \{1, \ldots, K\}$ *with* $a \neq b$ **do**
    2: $\mathbf{C}_{ab}(v_a, v_b) \leftarrow \mathbf{C}_{ab}(v_a, v_b) + 1$.
foreach $a, b \in \{1, \ldots, K\}$ *with* $a \neq b$ **do**
  3: $C_{ab} =$ row_normalize(C_{ab}) + column_normalize(C_{ab}).
3: Select top R highest edges (v_a, v_b) from $\mathcal{C}$. The weight of a pair is $r_{ab} \propto \mathbf{C}_{ab}(a,b)/|\mathcal{I}|$. Set $\mathcal{E} \leftarrow (v_a, v_b, r_{ab})$.

a learned intensity function and actual image occurrence data. This value is a distance metric, and thus a smaller value indicates a better model. In step 2 of Algorithm 1, the orders of the autoregressive models, λ_a^l of Eq.(11) and λ_c^l of Eq.(12), are decided by Akaike's information criterion (AIC). We choose the order parameters that lead to the smallest AIC, implying that the approximate distance between the model and the true process generating the data is the smallest. In practice, this step is important because temporal behaviors of visual clusters can operate at different time scales (*i.e.* monthly, weekly, or daily).

4.3 Regularized Multi-Task Regression

Until now, each visual cluster is independently modeled and learned without considering which description it is derived from. In order to fully take advantage of any arbitrary number of image descriptions, we introduce the idea of multi-task learning [3, 18], in which multiple related tasks are jointly learned by analyzing data from all of the tasks at the same time. This framework is powerful when the multiple tasks of interest are *different enough* to be specified by separate models, but are at the same time *similar enough* to be jointly learned.

We treat each descriptor as a *task*. Since each descriptor characterizes an image from a different perspective, it should be separately expressed. However, at the same time, it is likely that the descriptors from the same image share enough correlation that makes simultaneous learning beneficial. For example, suppose that a large portion of images of visual cluster 35 of HOG are also assigned to visual cluster 27 of Scene descriptors. It indicates that these two visual clusters are highly correlated, and thus are likely to share common covariates affecting their occurrences. Algorithm 2 discovers the set of frequently co-occurred visual cluster pairs as $\mathcal{E}$, where $e = (v_a, v_b, r_{ab}) \in \mathcal{E}$ consists of three tuples: a pair of visual clusters v_a and v_b with correlation weight $r_{ab} > 0$. We can model this dependency structure across multiple tasks (*e.g.* the correlations between the visual clusters of different image descriptors) by introducing regularization term $\Omega(\mathbf{\Theta}_E)$ to the log-likelihood:

$$\mathcal{L} = \sum_{l \in \mathcal{E}} \ell(N_{1:M}^l|\theta_k^l) - \Omega(\mathbf{\Theta}_E) \tag{13}$$

$$\Omega(\mathbf{\Theta}_E) = \mu \sum_{l \in \mathcal{E}} \|\theta_k^l\|_1 + \nu \sum_{(a,b) \in \mathcal{E}} r_{ab} \sum_{j=1}^{J} |\theta_j^a - \theta_j^b| \tag{14}$$

The regularization term $\Omega(\mathbf{\Theta}_E)$ consists of two different types of penalties, which are the *Lasso* penalty [25] and *graph-guided fusion* penalty [3]. μ and ν are regularization parameters that control sparsity and fusion levels. The overall effect of *graph-guided fusion* penalty is that each subgraph of visual clusters in $\mathcal{E}$ tends to share common relevant covariates, and the degree of commonality is proportional to the correlation strength r_{ab}.

4.4 Optimization for Parameter Learning

The goal of parameter learning is to obtain the MLE solution θ^{l*} that maximizes the likelihood with respect to an intensity function model λ^l and an observed image sequence $N_{1:M}^l$ for all $l = 1, \ldots, L$. Alternatively, if we explicitly represent the descriptor k as subscript, the set of parameters is denoted by $\mathbf{\Theta}^* = \{\mathbf{\Theta}_1^*, \ldots, \mathbf{\Theta}_K^*\}$ where $\mathbf{\Theta}_k^* = \{\theta_k^{1*}, \cdots, \theta_k^{L_k*}\}$ is the set of learned parameters for all visual clusters of descriptor k. We have introduced three likelihoods with different regularizations, which are optimized differently. First, the likelihood of Eq.(4) with no regularization term reduces to that of Poisson regression, and the globally-optimal solution can be attained by an iteratively reweighted least square algorithm [7]. Second, for the likelihood of Eq.(5) with the Lasso penalty, the globally-optimal MLE solution can be achieved by using the cyclical coordinate descent in [10]. Finally, for the likelihood of Eq.(13) with the graph-guided fusion penalty, we obtain an approximate MLE solution by modifying the *Proximal-gradient method* [3], which is a scalable first-order method (*i.e.* using only gradient) with a fast convergence rate. We extend this method that was developed for linear regressions to the regularized Poisson regressions.

5. TIME-SENSITIVE IMAGE RETRIEVAL

In this section, we discuss our second goal, which is to perform image ranking using the learned temporal models.

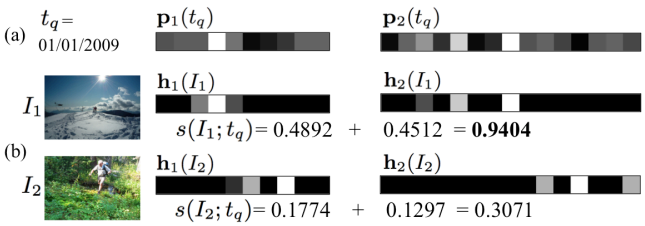


Figure 4: A toy example of computing ranking scores of two *mountain+camping* images I_1 and I_2 for t_q = (01/01/2009) with ($K = 2$, $L_1 = 10$, $L_2 = 15$). (a) Two membership vectors $\mathbf{p}_1(t_q)$ and $\mathbf{p}_2(t_q)$ are computed from the learned intensity functions. (b) Two descriptor vectors $\mathbf{h}_1$ and $\mathbf{h}_2$ are extracted from each image, and the ranking scores $s(I_1;t_q)$ and $s(I_2;t_q)$ are computed by Eq.(15).

5.1 Predictive Ranking

Computing intensity functions: As a result of optimization, we have the learned parameters of all visual clusters of all K descriptors: $\mathbf{\Theta}^* = \{\mathbf{\Theta}_1^*, \ldots, \mathbf{\Theta}_K^*\}$.

In the retrieval step, given a query time t_q, we first obtain $\mathbf{\Lambda}(t_q|\mathbf{\Theta}^*) = \{\mathbf{\Lambda}_1(t_q|\mathbf{\Theta}_1^*), \ldots, \mathbf{\Lambda}_K(t_q|\mathbf{\Theta}_K^*)\}$, which is the set of intensity functions of all visual clusters of all K descriptors for t_q. ($|\mathbf{\Lambda}(t_q|\mathbf{\Theta}^*)| = \sum_{k=1}^K L_k$). Each $\lambda_k^l(t_q|\theta_k^{l*}) \in \mathbf{\Lambda}_k(t_q|\mathbf{\Theta}_k^*)$ is computed by gathering covariate values for t_q, and plugging them along with learned θ_k^{l*} into Eq.(3). Here, let us remind that $\lambda_k^l(t_q|\theta_k^{l*}) \propto P(N_k^l(t_q+\Delta) - N_k^l(t_q)|N_{1:M}^l)$[3]. That is, the intensity function of a visual cluster at t_q is proportional to its occurrence probability at t_q. Therefore, for each $k \in K$, we can define a membership vector: $\mathbf{p}_k(t_q) = \mathbf{\Lambda}_k(t_q|\mathbf{\Theta}_k^*)/\|\mathbf{\Lambda}_k(t_q|\mathbf{\Theta}_k^*)\|_1 (\in \mathbb{R}^{L_k \times 1})$, where each $p^l \in \mathbf{p}_k(t_q)$ is the membership probability that an image occurred at t_q belongs to visual cluster l of descriptor k.

Ranking: The next step is to compute the ranking score of any given image I for t_q. We use the idea of continuos error-correcting output codes (ECOC) [4]. We first extract K image descriptors $\{\mathbf{h}_k(I)\}_{k=1}^K$ by the feature extraction methods in section 2.1. Then, the ranking score of image I at t_q is defined by the histogram intersection[4]

$$s(I;t_q) = \sum_{k=1}^{K} \|\min(\mathbf{h}_k(I), \mathbf{p}_k(t_q))\|_1. \quad (15)$$

Fig.4 illustrates a toy example of computing ranking scores for two images of the *mountain+camping* with $K = 2, L_1 = 10, L_2 = 15$. Fig.4.(a) shows two membership vectors $\mathbf{p}_k(t_q)$ that are computed from the learned intensity functions for t_q=(01/01/2009), and Fig.4.(b) illustrates four descriptor vectors for I_1 and I_2. The $\mathbf{p}_k(t_q)$ are more similar to the descriptors $\mathbf{h}_k(I_1)$ of image I_1 (*snowy mountain*) than $\mathbf{h}_k(I_2)$ of I_2(*tracking in woods*), and thus image I_1 is ranked higher.

The computation of our ranking score is very fast; the histogram intersection requires only element-wise min operations between K vector pairs. It is also easy to organize the descriptor vectors of images in the database by using any data structure such as trees or hashes for fast retrieval.

[3]It can be easily shown by that λ_k^l is an infinitesimal expected occurrence rate at t_q and a series of images is modeled by a sequence of conditionally independent Bernoulli trials during the derivation of the likelihood function of Eq.(4).

[4]In terms of the ECOC terminology, Eq.(15) means that the histogram intersection is chosen as the decoding metric.

5.2 Personalization

The key idea of personalization is, given a query user u_q, to assign more weights to the pictures taken by u_q and similar users to u_q during learning. In a normal setting, one image occurrence is equally counted by one for $N_{1:M}^l$ (See an example in Fig.3.(b)). However, for personalization, the images by u_q and the users in the same user cluster with u_q are weighted by larger values so that model fitting is more biased to their images. We implement the personalization by using the locally weighted learning framework [1], which is a form of lazy learning for a regression to adjust the weighting of data samples according to a query.

In order for personalization to be done offline, we apply this idea at the user cluster level. Suppose that there are Z user clusters as a result of pLSA based user clustering in section 2.2. We then compute $Z \times Z$ pairwise user similarity matrix $\mathbf{U}$ by $\mathbf{U}(x,y) = \sqrt{\exp(-(\mathbf{u}_x - \mathbf{u}_y)^2/\sigma)}$, where $\mathbf{u}_x$ and $\mathbf{u}_y$ are the user descriptors of cluster centers of U_x and U_y, respectively[5]. We learn the personalized model for each user cluster U_z, in which the weights of image occurrences are adjusted by $\mathbf{U}(z,*)$ (*i.e.* the z-th row of $\mathbf{U}$). That is, if the owner of an image I is in user cluster U_x, then I is reweighted by $\mathbf{U}(z,x)$. At the query stage, given a query user u_q, we identify the user cluster to which u_q belongs, and then use the pre-computed learned model of that cluster.

5.3 Computation time

Learning: The learning step performs only once, offline. The learning time without the graph-guided fusion penalty is $O(L|T|J)$ while that with the fusion penalty is $O(L|T|J^2)$ where $|T|$ is the number of time steps (*e.g.* discretized by day), J is the number of covariates, and $L = \sum_{k=1}^K L_k$. For a linear model, our Matlab code takes about less than one hour to learn the model for the 553K of *world+cup* images with $L = 1,050$, $|T| = 2,000$, and $J = 32$.

Querying: At the online querying stage, computing the intensity functions for a given query time t_q (and optionally a user u_q) runs in $O(LJ)$, and calculating the ranking scores of N images takes $O(LN)$. The overall querying step takes less than 0.5 second with $N = 1$K in the same experiment. Querying is fast enough to run online, but it can be also precomputed offline, for example, processing queries for next one year (365 days) can be done within a couple of hours.

6. EXPERIMENTS

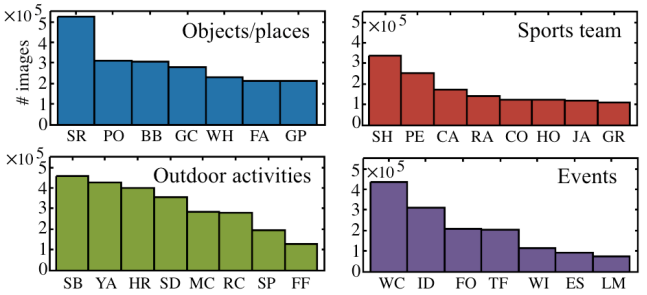
We evaluate the performance of our time-sensitive image retrieval algorithm using Flickr datasets.

6.1 Evaluation Setting

Datasets: Fig.5 summarizes our dataset that consists of more than seven million images of 30 topics from Flickr. We download all images that are retrieved by topic names as search keywords from Flickr without any filtering. The *date_taken* field of each image provided by Flickr is used for the timestamp.

Tasks: We first divide each image set into training and test set by time $T_T = T - (1 \text{ year})$ where T is the end time point of the dataset. That is, for each topic, the test set consists of the images in the last one year of the database, and the training set $\mathcal{I}_B$ comprises the other images, which are used to learn the image occurrence patterns.

[5]We use the Gaussian kernel function for image weighting.



SR(spider), PO(potato), BB(blackberry), GC(grandcanyon), WH(white +house), FA(fine+art), GP(grape), SH(shark), PE(penguin), CA(cardinal), RA(raptor), CO(coyote), HO(hornet), JA(jaguar), GR (grizzly), SB(snowboarding), YA(yacht), HR(horse+riding), SD(scuba+diving), MC(mountain+camping), RC(rock+climbing), SP(safari+park), FF(fly+fishing), WC(world+cup), ID(independence+ day), FO(formula+one), TF(tour+de+france), WI(wimbledon), ES (easter+sunday), LM(london+marathon)

Figure 5: 30 topics of our Flickr dataset. The topic words are classified into four categories. The total numbers of images and users are (7,592,426, 1,434,749).

Our tasks for experiments are similar to those of other image ranking and retrieval papers [5, 19, 29] except that time suitability of retrieved images is the key performance index to be evaluated. Our image retrieval task is performed as follows; a topic name and a query time point $t_q > T_T$ are given. That is, t_q is a *future* time point with respect to training data since T_T is the time threshold that divides the training set from the test set. The images that are actually taken in t_q are the positive test set $\mathcal{I}_P$. The negative test set $\mathcal{I}_N$ is gathered by randomly selecting the same number images outside of [$t_q \pm 3$ months] from the test set. The algorithm is supposed to rank the test images $\mathcal{I}_P \cup \mathcal{I}_N$ from which average precisions are computed. The personalized retrieval is the same except that a query user u_q is specified at the test. u_q is randomly chosen from a set of users who have at least 100 images in both training and test sets. For each topic, we randomly generate 36 t_q test cases (*i.e.* three random choices per month) for normal retrieval, and 20 (t_q, u_q) test pairs on average for personalized retrieval. That is, we examine more than $1,500$ test instances in total to evaluate the performance of our algorithm.

Baselines: The time-sensitive image retrieval is relatively new, and thus there are few existing methods to be compared. Hence, we select and adapt three baselines from popular image ranking methods for quantitative comparison with our algorithm. Below we summarizes the baselines, each of which is denoted by (`RSVM`) [14], (`PageR`) [13, 16], and (`Topic`) [23]. In the personalized retrieval, the locally weighted learning is also applied to all the competitors.

- Ranking SVM(`RSVM`) [14]: We obtain pseudo-relevant and pseudo-irrelevant training data by sampling images from the training set $\mathcal{I}_T$ based on their timestamps. The pseudo-relevant images are randomly sampled from Normal distributions whose mean are the same dates (m/d) of t_q in previous years. The pseudo-irrelevant images are randomly chosen from the images whose timestamps are outside [date(m/d) of $t_q \pm$ 3 months] at every year. Then, we learn the Ranking SVM using the code provided by the authors of [14].
- PageRank-based model(`PageR`) [11, 16]: Given the same training data above, we build a similarity graph between training and test data by using HOG and SIFT features, and compute ranking scores using the *random walk with restart* [26] (*i.e.* a query-specific PageRank).
- Author-Time Topic Model(`Topic`) [23]: We modify the Author-Topic model [23] to jointly model users, months, and visual clusters of images. Using the same training data above, we estimate the subtopic distribution of each month and the subtopic assignments of visual clusters, from which we compute the ranking scores of test images for t_q.

6.2 Quantitative Results

Fig.6 and Fig.7 show the quantitative comparison of normal and personalized image retrieval between our approach and three baselines, respectively. We report the mean average precision at top 40 and 80 ranked images, which are denoted by mAP@40 and mAP@80. In each figure, the leftmost bar set is the average performance of 30 topics, and the results of all 30 topics follow. Our algorithm significantly outperformed all the competitors in most topic classes for both tasks. In the average accuracy of normal retrieval, our mAP@40(80) values are higher by 5.6% (8.0%) points than the best baseline (`Topic`). In the average accuracy of personalized retrieval, our method also outperforms the best baseline (`PageR`) by 4.7% (4.2%) points for mAP@40(80). The personalized retrieval is more accurate to rank the images than the normal one, because knowing the user at query time provides a strong clue to narrow down the search space.

6.3 Qualitative Results

Fig.8 shows some examples of retrieval comparison between our method in the top row and the best baseline in the bottom row. We illustrate top eight ranked images by each method, along with the average images of top 100 images to show the mean statistics of the two output sets. In these examples, our method reports fewer false positives (*i.e.* the images with red boundaries) than the best baselines.

As another qualitative result, Fig.9 shows the predicted power of our algorithm for unseen future images. Fig.9.(a) illustrates retrieval results for the *independence+day* at four t_q from different months. In each set, the top row shows five images that are sampled out of ten highest ranked training images for t_q, and the bottom row presents their best-matched test images. The matched pairs are obtained from one-to-one correspondences by feature-wise distances. If the matched pairs are similar each other, it means that our algorithm can predict unseen future images very well. The *independence+day* is national holidays for many countries with different dates. Hence, according to four different t_q, we can observe various views of the events in different countries. For example, the second t_q (top-right) of Fig.9.(a) is near to the US independence day; the high ranked images show its common storyline: parades, parties with children, and fireworks at night. They are distinctive with the scenes in the Independence day of India (bottom-left) and an African country (bottom-right) of Fig.9.(a).

Fig.9.(b) shows examples of the importance of personalization. The *raptor* in Fig.9.(b) show the variation of the term usages including a basketball team, a fighter aircraft, an eagle, and an ice hockey team, most of which are seemingly irrelevant to its first semantic meaning as a dinosaur. Each user perceives the term *raptor* narrowly for his or her interests, which are relatively stationary and predictable once they are learned.

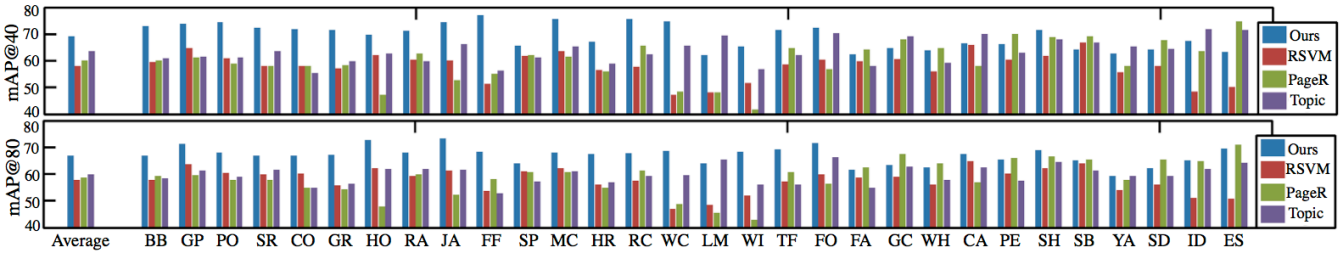


Figure 6: Quantitative comparison of image retrieval between our method and three baselines (`RSVM`, `PageR`, `Topic`) using mAP@40(top) and mAP@80(bottom) metrics. The average performances for mAP@(40,80) in the left-most bar set are ours (69.1%,66.7%), `RSVM` (58.0%,57.6%), `PageR` (60.1%,58.6%), `Topic` (63.5%,59.7%).

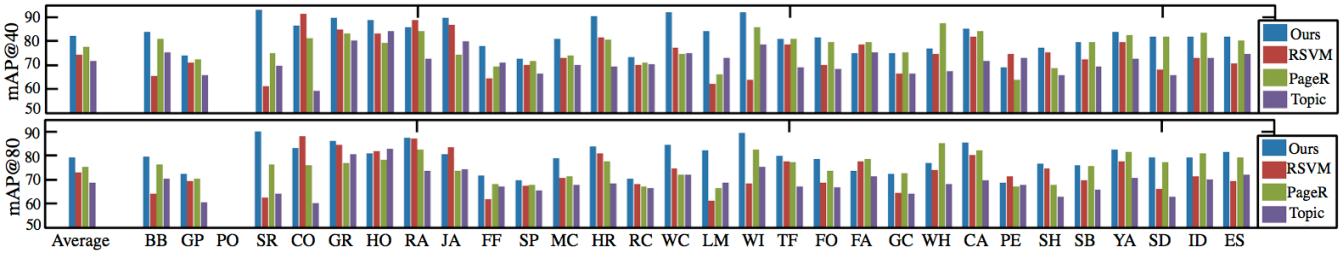


Figure 7: Quantitative comparison of personalized image retrieval between our method and three baselines using mAP@40(top) and mAP@80(bottom). The average performances for mAP@(40,80) in the left-most bar set are ours(82.1%,79.2%), `RSVM`(74.3%,72.8%), `PageR`(77.4%,75.0%), `Topic`(71.4%,68.7%).

Our experimental results conclude that some topics follow periodical patterns that are predictable, and our algorithm can enhance the image retrieval quality according to the temporal trends. Specifically, our method is successful for polysemous topics that show strong annual or periodic trends (*e.g.* sports related topics such as *shark* and *hornet*), and event topics that many people share but experience in different ways (*e.g.* outdoor activities such as *mountain+camping*). Moreover, we observe that the time-sensitive personalization is promising for image retrieval when a query keyword has a board range of concepts, which are differently recognized according to people's thoughts and interests. Although the personalized search has been studied much in text retrieval research, our results reveal that images can convey more subtle information about user preferences that are hardly captured by texts.

7. CONCLUSION

In this paper, we propose an approach for time-sensitive image ranking and retrieval that is based on multi-task regression on multivariate point processes. With experiments on more than seven millions of Flickr images for 30 topic keywords, we show the superiority of the proposed approach over other candidate method. Among future work that could further boost performance, first, we can incorporate other meta data of Flick (*e.g.* comments or favs) as covariates for the temporal models; second, it is worth exploring the joint temporal behaviors of topics along with other data modalities such as associated texts or social networks of users.

Acknowledgement: This work is supported by NSF IIS-1115313 and AFOSR FA9550010247.

8. REFERENCES

[1] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally Weighted Learning. *AI Review*, 11(1):11–73, 1997.

[2] E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank. The Time-Rescaling Theorem and Its Application to Neural Spike Train Data Analysis. *Neural Computation*, 14(2):325–346, 2001.

[3] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing. Smoothing Proximal Gradient Method for General Structured Sparse Learning. In *UAI*, 2011.

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

[5] J. Cui, F. Wen, and X. Tang. Real Time Google and Live Image Search Re-ranking. In *ACM MM*, 2008.

[6] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering General Time-Sensitive Queries. In *CIKM*, 2008.

[7] D. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer, 2003.

[8] A. S. Das, M. Datar, , A. Garg, and S. Rajaram. Google News Personalization: Scalable Online Collaborative Filtering. In *WWW*, 2007.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE PAMI*, 32(9):1627–1645, 2010.

[10] J. Friedman, T. Hastie, and R. Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Statistical Software*, 33:1–22, 2010.

[11] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video Search Reranking Through Random Walk over Document-Level Context Graph. In *ACM MM*, 2007.

[12] X. Jin, A. Gallagher, L. Cao, J. Luo, and J. Han. The Wisdom of Social Multimedia: Using Flickr for Prediction and Forecast. In *ACM MM*, 2010.

[13] Y. Jing and S. Baluja. VisualRank: Applying PageRank to Large-Scale Image Search. *IEEE PAMI*, 30:1877–1890, 2008.

[14] T. Joachims. Optimizing Search Engines Using Clickthrough Data. In *KDD*, 2002.

[15] G. Kim and E. P. Xing. Web Image Prediction Using Multivariate Point Processes . In *KDD*, 2012.

[16] G. Kim, E. P. Xing, and A. Torralba. Modeling and

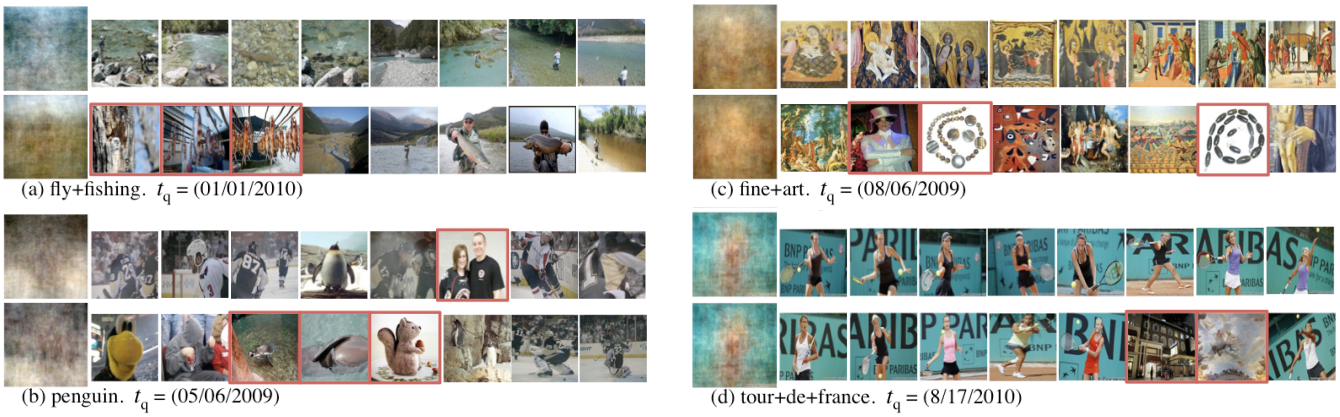


Figure 8: Comparison of eight top-ranked images for normal image retrieval in (a) and (b) and for personalized image retrieval in (c) and (d), by our method in the top row and by the best baseline in bottom row. The pictures with red boundaries are false positives. We also present average image of top 100 retrieved images in the left-most of each row.

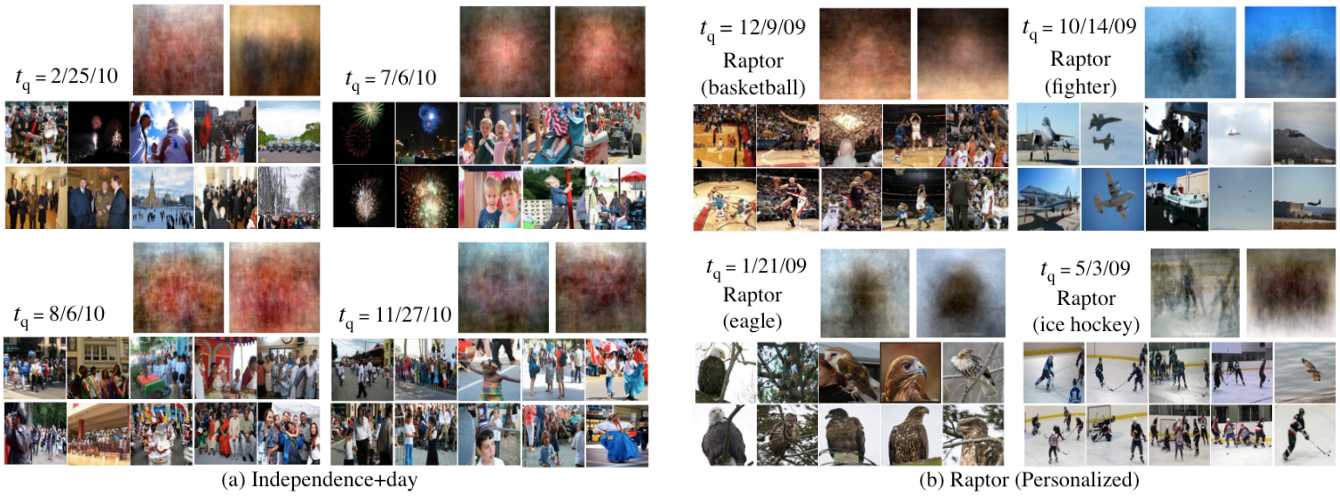


Figure 9: Image retrieval examples. (a) Normal retrieval for the *independence+day* at four t_q in different months. In each set, we show five sampled images form top 10 ranked training images in the top row, and their best-matched test images in the bottom row. We also present the average images of top 100 retrieved training images (left) and their best-matched test images (right). Independence day scenes in different countries are shown according to query time t_q. (b) Personalized retrieval for the *raptor* at four different (t_q, u_q) pairs. Even though the images are associated with the same keyword, their contents extremely vary according to users' interests.

Analysis of Dynamic Behaviors of Web Image Collections. In *ECCV*, 2010.

[17] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais. Understanding Temporal Query Dynamics. In *WSDM*, 2011.

[18] H. Liu, M. Palatucci, and J. Zhang. Blockwise Coordinate Descent Procedures for the Multi-task Lasso, with Applications to Neural Semantic Basis Discovery. In *ICML*, 2009.

[19] W. Liu, Y.-G. Jiang, J. Luo, and S.-F. Chang. Noise Resistant Graph Ranking for ImprovedWeb Image Search. In *CVPR*, 2011.

[20] D. Metzler, R. Jones, F. Peng, and R. Zhang. Understanding Temporal Query Dynamics. In *SIGIR*, 2009.

[21] N. Morioka and J. Wang. Robust Visual Reranking via Sparsity and Ranking Constraints. In *ACM MM*, 2011.

[22] K. Radinsky, K. M. Svore, S. T. Dumais, J. Teevan, A. Bocharov, and E. Horvitz. Modeling and Predicting Behavioral Dynamics on the Web. In *WWW*, 2012.

[23] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The Author-Topic Model for Authors and Documents. In *UAI*, 2004.

[24] V. K. Singh, M. Gao, and R. Jain. Social Pixels: Genesis and Evaluation. In *ACM MM*, 2010.

[25] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *J. Royal. Statist. Soc B.*, 58(1):267–288, 1996.

[26] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast Random Walk with Restart and Its Applications. In *ICDM*, 2006.

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

[28] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects. *J. Neurophysiol*, 93(2):1074–1089, 2005.

[29] X. Wang, K. Liu, and X. Tang. Query-Specific Visual Semantic Spaces for Web Image Re-ranking. In *CVPR*, 2011.

[30] 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.

[31] L. Yang and A. Hanjalic. Supervised Reranking for Web Image Search. In *ACM MM*, 2010.