Image Ranking with Eye Movements

Kitsuchart Pasupa[*]
School of Electronics & Computer Science
University of Southampton
kp2@ecs.soton.ac.uk

Sandor Szedmak[†]
School of Electronics & Computer Science
University of Southampton
ss03v@ecs.soton.ac.uk

David R. Hardoon[‡]
Data Mining Department
Institute for Infocomm Research (I^2R)
drhardoon@i2r.a-star.edu.sg

Abstract

In order to help users navigate an image search system, one could provide explicit rank information on a set of images. These rankings are learnt so to present a new set of relevant images. Although, requiring explicit information may not be feasible in some cases, we consider the setting where the user provides implicit feedback, eye movements, to assist in such a task. This paper explores the idea of implicitly incorporating eye movement features in an image ranking task. Previous work had demonstrated that combining eye movement and image features improved the retrieval accuracy. Despite promising results the proposed approach is unrealistic as no eye movements are given a-priori for new images. We propose a novel search approach which combines image together with eye movements features in a tensor Ranking Support Vector Machine, and show that by extracting the individual source-specific weight vectors we are able to construct a new image-based semantic space which outperforms in retrieval accuracy.

1 Introduction

Recently, relevance feedback, which is explicitly provided by the user while performing a search query on the quality of the retrieved images, has shown to be able to improve on the performance of Content-Based Image Retrieval systems, as it is able to handle the large variability in semantic interpretation of images across users. Many systems rely on an explicit feedback mechanism, where the user explicitly indicates which images are relevant for their search query and which ones are not. However, providing explicit feedback is also a laborious process as it requires continues user response. Alternatively, it is possible to use implicit feedback (e.g. eye movements, mouse pointer movements) to infer relevance of images. In other words, user responses that are implicitly related to the task performed.

In this study we explore the use of eye movements as a particular source of implicit feedback to assist a user when performing such a task (i.e. image retrieval). Eye movements can be treated as an implicit relevance feedback when the user is not consciously aware of their eye movements being tracked. This work is an extended study from [4] where they demonstrated that ranking of images can be inferred from eye movements using Ranking Support Vector Machine (Ranking SVM). Their experiment shows that the performance of the search can be improved when simple images

[*] www.ecs.soton.ac.uk/∼kp2
[†] www.ecs.soton.ac.uk/∼ss03v
[‡] www.davidroihardoon.com

features namely histograms are fused with the eye movement features. Despite their encouraging results, their proposed approach is largely unrealistic as they combine image and eye features for both training and testing. Whereas in a real scenario no eye movements will be presented a-priori for new images. Therefore, we propose a novel search methodology which combines image features together with implicit feedback from users' eye movements during training, such that we are able to rank new images with only using image features. For this purpose, we propose using tensor kernels in the Ranking SVM framework. Tensors have been recently used in bioinformatics [2, 6]. In this study we use the tensor product to constructed a joined semantics space by combining eye movements and image features.

2 Ranking SVM

We are given a set of samples $\mathcal{S} = \{(\mathbf{x}_i, r_i)\}, i = 1, \ldots, M$, and a set of pairs of indices $\mathcal{P} = \{(i,j)|i < j\}$,$m = |\mathcal{P}|$. Let $\mathbf{x}_i$ denote some feature vector and r_i denote the ranking assigned to $\mathbf{x}_i$. If $r_1 \succ r_2$, it means that $\mathbf{x}_1$ is more relevance than $\mathbf{x}_2$. Consider a linear ranking function,

$$\mathbf{x}_i \succ \mathbf{x}_j \Longleftrightarrow \langle \mathbf{w}, \mathbf{x}_i \rangle - \langle \mathbf{w}, \mathbf{x}_j \rangle > 0,$$

where $\mathbf{w}$ is a weight vector and $\langle \cdot, \cdot \rangle$ denotes dot product between vectors. This can be placed in a binary SVM classification framework where $c_{(i,j)}$ is the new label indicating the quality of rank pair,

$$sgn(\langle \mathbf{w}, \mathbf{x}_i - \mathbf{x}_j \rangle) = \begin{cases} c_{(i,j)} = +1 & \text{if } r_i \succ r_j \\ c_{(i,j)} = -1 & \text{if } r_j \succ r_i \end{cases},$$

which can be solved by the following optimisation problem,

$$\min \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{(i,j)\in\mathcal{P}} \xi_{(i,j)} \tag{1}$$

subject to the following constrains:

$$\begin{aligned} \forall (i,j) \in \mathcal{P} \quad &: \quad c_{(i,j)}(\langle \mathbf{w}, \mathbf{x}_i - \mathbf{x}_j \rangle + b) \geq 1 - \xi_{(i,j)} \\ &: \quad \xi_{(i,j)} \geq 0 \end{aligned}$$

where C is a hyper-parameter which allows trade-off between margin size and training error, and $\xi_{(i,j)}$ is training error. Alternatively, we are represent the Ranking SVM as a vanilla SVM where we re-represent our samples as

$$\phi(\mathbf{x})_{(i,j)} = \mathbf{x}_i - \mathbf{x}_j$$

with label $c_{(i,j)}$. Finally, we quote from [3] the general dual SVM optimisation as

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} - \frac{1}{2} \sum_{(i,j),(k,l)\in\mathcal{P}} \alpha_{(i,j)}\alpha_{(k,l)}c_{(i,j)}c_{(k,l)}\kappa(\phi_{(i,j)}(x_i, x_j), \phi_{(k,l)}(x_k, x_l)) \tag{2}$$

$$\text{subject to } \textstyle\sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)}c_{(i,j)} = 0 \text{ and } \alpha_{(i,j)} \geq 0, \;\; (i,j) \in \mathcal{P},$$

where we again use $c_{(i,j)}$ to represent the label and $\kappa(\phi_{(i,j)}(x_i, x_j), \phi_{(k,l)}(x_k, x_l))$ to be the kernel function between $\phi_{(i,j)}$ and $\phi_{(k,l)}$.

3 Tensor Ranking SVM

Let us consider the learning situation when we have two sources of feature vectors, e.g. in our scenario we are given a feature vector to every image pairs $\phi(x)_{(i,j)} = \mathbf{x}_i - \mathbf{x}_j$ and we have feature vectors $\phi(y)_{(i,j)} = \mathbf{y}_i - \mathbf{y}_j$ derived from the eye movement, $\mathbf{y}_i$, to a subset of the images. The task is to compute a hyperplane to the Ranking SVM such that the normal vector $\mathbf{w}$ of this hyperplane also incorporates the information from the partially given eye movements and not only from the image features. We suppose that both sources of the information are available in the training procedure. To combine the two sources into a common feature vector the tensor product of the feature vectors $\phi(x)_{(i,j)} \circ \phi(y)_{(i,j)}$ of the sources is computed, and let the optimal normal vector of the common

space be denoted by W. Then we assume that the projection of W into the space of image features can express the effect of the eye movements. Thus, in the common space we have the following constraints

$$\forall (i,j) \in \mathcal{P} : c_{(i,j)}(\langle W, \phi(x)_{(i,j)} \circ \phi(y)_{(i,j)} \rangle + b) \geq 1 - \xi_{(i,j)}.$$

For the sake of simplicity, we change the index expression exploiting the fact that Ranking SVM is a conventional SVM with special feature vectors. The index i denotes the constraints in the next formulas.

In the following section we propose to construct a tensor kernel on the ranked image and eye movements features, i.e. following equation (2), to then to train an SVM. Therefore, let $\mathbf{X} \in \mathbb{R}^{n \times m}$ and $\mathbf{Y} \in \mathbb{R}^{\ell \times m}$ be the matrix of sample vectors, $\mathbf{x}$ and $\mathbf{y}$, for the image and eye movements respectively, where n is the number of image features and ℓ is the number of eye movement features and m are the total number of samples. We continue to define K^x, K^y as the kernel matrices for the ranked images and eye movements respectively. In our experiments we use linear kernels, i.e. $K^x = X'X$ and $K^y = Y'Y$. The resulting kernel matrix of the tensor $T = X \circ Y$ can be expressed as pair-wise product (see [5] for details)

$$\bar{K}_{ij} = (T'T)_{ij} = K^x_{ij} K^y_{ij}.$$

We use $\bar{K}$ in conjunction with the vanilla SVM formulation as given in equation (2). Whereas the set up and training are straight forward the underlying problem is that for testing we do not have the eye movements. Therefore we propose to decompose the resulting weight matrix from its corresponding image and eye components such that each can be used independently.

The goal is to decompose the weight matrix W given by a dual representation

$$W = \sum_i^m \alpha_i c_i \phi_x(\mathbf{x}_i) \circ \phi_y(\mathbf{y}_i)$$

without accessing the feature space. Given the paired samples $\mathbf{x}, \mathbf{y}$ the decision function in equation is

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}) &= W \circ \phi_x(\mathbf{x}) \phi_y(\mathbf{y})' \\ &= \sum_{i=1}^m \alpha_i c_i \kappa_x(\mathbf{x}_i, \mathbf{x}) \kappa_y(\mathbf{y}_i, \mathbf{y}). \end{aligned}$$

3.1 Decomposition

We want to decompose the weight matrix into a sum of tensor products of corresponding weight components for the images and eye movements

$$W \approx W^T = \sum_{t=1}^T \mathbf{w}_x^t \mathbf{w}_y^{t\prime},$$

so that $\mathbf{w}_x^t = \sum_{i=1}^m \beta_i^t \phi_x(\mathbf{x}_i)$ and $\mathbf{w}_y^t = \sum_{i=1}^m \gamma_i^t \phi_y(\mathbf{y}_i)$ where $\boldsymbol{\beta}^t, \boldsymbol{\gamma}^t$ are the dual variables of $\mathbf{w}_x^t, \mathbf{w}_y^t$.

We compute

$$WW' = \sum_{i,j}^m \alpha_i \alpha_j c_i c_j \kappa_y(\mathbf{y}_i, \mathbf{y}_j) \phi_x(\mathbf{x}_i) \phi_x(\mathbf{x}_j)' \tag{3}$$

and are able to express $K^y = (\kappa_y(\mathbf{y}_i, \mathbf{y}_j))_{i,j=1}^m = \sum_{k=1}^K \lambda_k \mathbf{u}^k \mathbf{u}^{k\prime} = U \Lambda U'$, where $U = (\mathbf{u}_1, \ldots, \mathbf{u}_K)$ by performing an eigenvalue decomposition of the kernel matrix K^y with entries $K^y_{ij} = \kappa_y(\mathbf{y}_i, \mathbf{y}_j)$. Substituting back into equation (3) gives

$$WW' = \sum_k^K \lambda_k \sum_{i,j}^m \alpha_i \alpha_j c_i c_j \mathbf{u}_i^k \mathbf{u}_j^{k\prime} \phi_x(\mathbf{x}_i) \phi_x(\mathbf{x}_j)'.$$

Letting $\mathbf{h}_k = \sum_{i=1}^{m} \alpha_i c_i \mathbf{u}_i^k \phi_x(\mathbf{x}_i)$ we have $WW' = \sum_k^K \lambda_k \mathbf{h}_k \mathbf{h}_k' = HH'$ where $H = \left(\sqrt{\lambda_1}\mathbf{h}_1, \ldots, \sqrt{\lambda_K}\mathbf{h}_K\right)$. We would like to find the singular value decomposition of $H = V\Upsilon Z'$. Consider for $A = diag(\boldsymbol{\alpha})$ and $C = diag(\mathbf{c})$ we have

$$\begin{aligned}[H'H]_{k\ell} &= \sqrt{\lambda_k \lambda_\ell} \sum_{ij} \alpha_i \alpha_j c_i c_j \mathbf{u}_i^k \mathbf{u}_j^\ell \kappa_x(\mathbf{x}_i, \mathbf{x}_j) \\ &= \left[\left(CAU\Lambda^{\frac{1}{2}} \right)' K^x \left(CAU\Lambda^{\frac{1}{2}} \right) \right]_{k\ell},\end{aligned}$$

which is computable without accessing the feature space. Performing an eigenvalue decomposition on $H'H$ we have

$$H'H = Z\Upsilon V' V \Upsilon Z' = Z\Upsilon^2 Z'$$

with Υ a matrix with υ_t on the diagonal truncated after the j^{th} eigenvalue, which gives the dual representation of $\mathbf{v}_t = \frac{1}{\upsilon_t} H\mathbf{z}_t$ for $t = 1, \ldots, T$, and since $H'H\mathbf{z}_t = \upsilon_t^2 \mathbf{z_t}$ we are able to verify that

$$WW'\mathbf{v}_t = HH'\mathbf{v}_t = \frac{1}{\upsilon_t} HH'H\mathbf{z}_t = \upsilon_t H\mathbf{z}_t = \upsilon_t^2 \mathbf{v}_t.$$

Restricting to the first T singular vectors allows us to express $W \approx W^T = \sum_{t=1}^{T} \mathbf{v}_t \left(W'\mathbf{v}_t\right)'$, which in turn results in

$$\mathbf{w}_x^t = \mathbf{v}_t = \frac{1}{\upsilon_t} H\mathbf{z}_t = \sum_{i=1}^{m} \beta_i^t \phi_x(\mathbf{x}_i),$$

where $\beta_i^t = \frac{1}{\upsilon_t} \alpha_i c_i \sum_{k=1}^{T} \sqrt{\lambda_k} \mathbf{z}_k^t u_i^k$. We can now also express

$$\mathbf{w}_y^t = W'\mathbf{v}_t = \frac{1}{\upsilon_t} W'H\mathbf{z}_t = \sum_{i=1}^{m} \gamma_i^t \phi_y(\mathbf{y}_i),$$

where $\gamma_i^t = \sum_{j=1}^{m} \alpha_i c_i \beta_j^t \kappa_x(\mathbf{x}_i, \mathbf{x}_j)$ are the dual variables of $\mathbf{w}_y^t$. We are therefore now able to decompose W into W_x, W_y without accessing the feature space giving us the desired result.

We are now able to compute, for a given t, the ranking scores in the linear discriminant analysis form $s = \mathbf{w}_x^t \hat{X}$ for new test images $\hat{X}$. These are in turn sorted in order of magnitude (importance). Equally, we can project our data into the new defined semantic space β where we train and test an SVM. i.e. we compute $\tilde{\phi}(\mathbf{x}) = K^x \beta$, for the training samples, and $\tilde{\phi}(\mathbf{x}_t) = K_t^x \beta$ for our test samples. We explore both these approaches in our experiments.

4 Experiments

We evaluate two different scenarios for learning the ranking of image based on image and eye features; 1) Predicting rankings on a page given only other data from a single specific user. 2) A global model using data from other users to predict rankings for a new unseen user. We compute a 256-bin grey scale histogram on the whole image as the feature representation. These features are intentionally kept relatively simple. The 33 eye movement features are computed based only on the eye trajectory and locations of the images in the page. This type of features are general-purpose and are easily applicable to all application scenarios. We compare our proposed tensor Ranking SVM algorithm which combines both information from eye movements and image histogram features to a Ranking SVM using either of these features alone, and to a Ranking SVM using both eye movements and histogram features. We further emphasis that training and testing a model using only eye movements is *not realistic* as there are no eye movements presented a-priori for new images, i.e. one can not test. This comparison provides us with a baseline as to how much it may be possible to improve on the performance using eye movements. In the experiments we use a linear kernel function. Although, it is possible to use a non-linear kernel on the eye movement features as this would not effect the decomposition for the image weights. See [4] for more details on experiment setup and feature extraction.

4.1 Page Generalisation

In the following section we focus on predicting rankings on a page given only other data from a single specific user. We employ a leave-page-out routine where at each iteration a page, from a given user, is withheld for testing and the remaining pages, from the same user, are used for training.

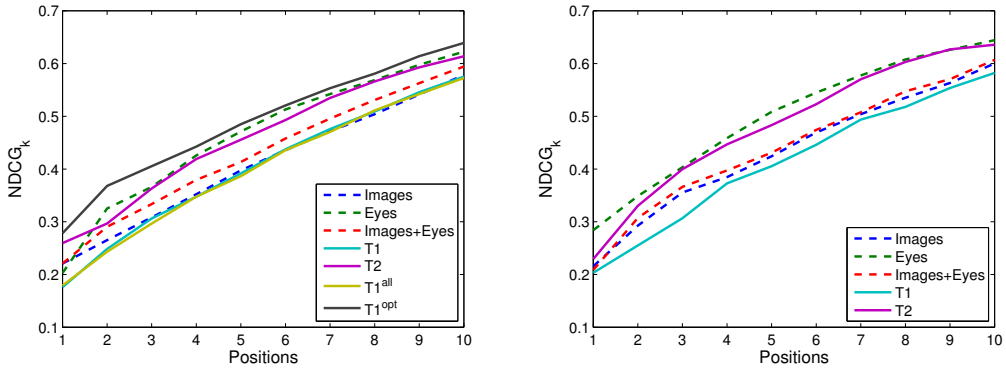
We evaluate the proposed approach with the following four setting: (1) $T1$: using the largest component of tensor decomposition in the form of a linear discriminator. We use the weight vector corresponding to the largest eigenvalue (as we have a t weights). (2) $T2$: we project the image features into the learnt semantic space (i.e. the decomposition on the image source) and train and test within the projected space a secondary Ranking SVM. (3) $T1^{all}$: similar to $T1$ although here we use all t weight vectors and take the mean value across as the final score. (4) $T1^{opt}$: similar to $T1$ although here we use the n-largest components of the decomposition. i.e. we select n weight vectors to use and take the mean value across as the final score.

We use a leave-one-out cross-validation for $T1^{opt}$ to obtain the optimal model for the later case which are selected based on maximum average Normalised Discount Cumulative Gain (NDCG) across 10 positions.

In figure 1(a) we plot the average performance across all users. The figure shows that $T1$ and $T1^{all}$ are slightly worse than using image histogram alone. However, when we carefully select the number of largest components in tensor decomposition, the performance of the classifier is greatly improved and clearly outperforms the Ranking SVM with eye movements. Using classifier $T2$, the performance is improved above the Ranking SVM with image features and it is competitive with Ranking SVM with eye movements features.

4.2 User Generalisation

In the following section we focus on learning a global model using data from other users to predict rankings for a new unseen user. Although, as the experiment is set up such that each user views the same pages as all other users we employ a leave-user-leave-page-out routine. We evaluate the proposed approach with the following two setting: (1) $T1$ (2) $T2$. We plot in figure 1(b) the average NDCG performance on the leave-user-leave-page-out routine, demonstrating that on average we improve on the ranking of new images for new users.



(a) Page generalisation

(b) User generalisation

Figure 1: Sub-figures 1(a) shows average NDCG performance across all users for predicting rankings on a page given only other data from a single specific user. Sub-figures 1(b) shows the average NDCG performance where we are able to observe that $T2$ outperforms the ranking of only using image features. The 'Eyes' and 'Images+Eyes' plots in all the figures demonstrate how the ranking would perform if eye-features were indeed available a-priori for new images.

5 Discussion

Improving search and content based retrieval systems with implicit feedback is an attractive possibility given that a user is not required to explicitly provide information to then improve, and personalise, their search strategy. This, in turn, can render such a system more user-friendly and simple to use (at least from the users' perspective). Although, achieving such a goal is non-trivial as one needs to be able to combine the implicit feedback information into the search system in a manner that does not then require the implicit information for testing. In our study we focus on implicit feedback in the form of eye movements, as these are easily available and can be measured in a non-intrusive manner. Previous studies [1] have shown the feasibility of such systems using eye moments for a textual search task. Demonstrating that it is indeed possible to 'enrich' a textual search with eye features. Although their proposed approach is computationally complex since it requires the construction of a regression function on eye measurements on each word. This was not realistic in our setting. Furthermore, [4] had extend the underlying methodology of using eye movement as implicit feedback to an image retrieval system, combining eye movements with image features to improve the ranking of retrieved images. Although, still, the proposed approach required eye features for the test images which would not be practical in a real system. We propose a novel search strategy for combining eye movements and image features with a tensor product kernel used in a Ranking SVM framework.

Acknowledgments

The authors would like to acknowledge financial support from the European Community's Seventh Framework Programme (FP7/2007–2013) under *grant agreement* n° 216529, Personal Information Navigator Adapting Through Viewing (PinView) project (http://www.pinview.eu) and the IST Programme of the European Community, PASCAL2 Network of Excellence (http://www.pascal-network.org), IST-2007-216886. This publication only reflects the authors' views. The authors would also like to thank Craig J. Saunders for data collection.

References

[1] Antti Ajanki, David R. Hardoon, Samuel Kaski, Kai Puolamäki, and John Shawe-Taylor. Can eyes reveal interest? Implicit queries from gaze patterns. *User Modeling and User-Adapted Interaction*, 19(4):307–339, 2009.

[2] Asa Ben-Hur and William Stafford Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21:i38–i46, 2005.

[3] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[4] Kitsuchart Pasupa, Craig Saunders, Sandor Szedmak, Arto Klami, Samuel Kaski, and Steve Gunn. Learning to rank images from eye movements. In *HCI '09: Proceeding of the IEEE 12th International Conference on Computer Vision (ICCV'09) Workshops on Human-Computer Interaction*, pages 2009–2016, 2009.

[5] Sylvia Pulmannová. Tensor products of Hilbert space effect algebras. *Reports on Mathematical Physics*, 53(2):301–316, 2004.

[6] Jian Qiu and William Stafford Noble. Predicting co-complexed protein pairs from heterogeneous data. *PLoS Computational Biology*, 4(4):e1000054, 2008.