# Learning to Rank Images from Eye movements

Kitsuchart Pasupa[1] kp2@ecs.soton.ac.uk
Craig J. Saunders[1] cjs@ecs.soton.ac.uk
Sandor Szedmak[1] ss03v@ecs.soton.ac.uk
Arto Klami[2] arto.klami@tkk.fi
Samuel Kaski[2] samuel.kaski@tkk.fi
Steve R. Gunn[1] srg@ecs.soton.ac.uk

School of Electronics & Computer Science[1]
University of Southampton
SO17 1BJ, United Kingdom

Department of Information and Computer Science[2]
Helsinki University of Technology
P.O. Box 5400, 02015 TKK, Finland

## Abstract

*Combining multiple information sources can improve the accuracy of search in information retrieval. This paper presents a new image search strategy which combines image features together with implicit feedback from users' eye movements, using them to rank images. In order to better deal with larger data sets, we present a perceptron formulation of the Ranking Support Vector Machine algorithm. We present initial results on inferring the rank of images presented in a page based on simple image features and implicit feedback of users. The results show that the perceptron algorithm improves the results, and that fusing eye movements and image histograms gives better rankings to images than either of these features alone.*

## 1. Introduction

Searching for images from a large collection (for example on the web, or for a designer seeking a professional photo for a brochure) is a difficult task for automated algorithms, and many current techniques rely on items which have been manually *tagged* with descriptors. This situation is not ideal, as both formulating the initial query, and navigating the large number of hits returned is a difficult process. In order to present relevant images to the user, 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. One can then use a machine learning algorithm to try and present a new set of images to the user which are more relevant – thus helping them navigate the large number of hits. An example of such systems is PicSOM [9].

In this work we try to use a particular source of implicit feedback, eye movements, to assist a user when performing such a task. There is a large body of work on eye movements (see *e.g.* [12]), however most of the human-computer interface (HCI) works treated eye movement as an input or explicit feedback mechanism *e.g.* [16]. Eye movements however can also be treated as an implicit feedback when the user is not consciously trying to influence the interface by where they focus their attention. Eye movements as implicit feedback has recently been considered in the text retrieval setting [11, 4, 1]. To the best of our knowledge however, at the time of writing, only [10, 8] used eye movements for image retrieval. They only infer a binary judgement of relevance whereas in our experiments, we make the task more complex and realistic for search-based tasks by asking the user to rank a set of images on a screen in order of relevance to a specific topic while the eye movements are recorded. This is to demonstrate that ranking of images can be inferred from eye movements.

In this work we use eye movements and simple image features in conjunction with state of the art machine learning techniques in order to tackle the image search application. The selected algorithm is a variant of the Support Vector Machine (SVM), the "Ranking SVM" [7], which was developed to automatically improve the retrieval quality of a search engine using click-through data. In this paper we adapt the Ranking SVM into a perceptron-style algorithm in order to suit the setting of online learning, as well as improving its computation performance.

The paper is organized as follows. Section 2 outlines the Ranking SVM algorithm and introduces our proposed perceptron algorithm. Section 3 explains our ranking experimental framework, and section 4 presents how we extract features from eye trajectories and images in a database. Then the results of applying the proposed method to the ranking problem are given in section 5.

2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops
978-1-4244-4441-0/09/$25.00 ©2009 IEEE

## 2. Methodologies

### 2.1. Ranking SVM

Let $\boldsymbol{x}_i^{(n)}$ denote the $m$-dimensional feature vector which describes the match between image $i$ and page $n$. In this paper, subscripts and superscripts indicate the index of images and pages respectively. The exact nature of these features are explained in detail in section 4. A ranking assigned to $\boldsymbol{x}_i^{(n)}$ is denoted by $r_i^{(n)}$; the set of ranks measuring the relevance of images in a page is assumed to be human-annotated. If $r_1 \succ r_2$, it means that $\boldsymbol{x}_1$ is more relevance than $\boldsymbol{x}_2$. Hence, we have a training set of $\{(\boldsymbol{x}_i^{(n)}, r_i^{(n)})\}$ where $n = 1, \ldots, k$ indexes each page and $i = 1, \ldots, p^{(n)}$ indexes each image in a page.

The Ranking SVM was proposed by [7] and is adapted from ordinal regression [5]. It is a pair-wise approach where the solution is a binary classification problem. Consider a linear ranking function,

$$\boldsymbol{x}_i^{(n)} \succ \boldsymbol{x}_j^{(n)} \Longleftrightarrow \langle \boldsymbol{w}, \boldsymbol{x}_i^{(n)} \rangle - \langle \boldsymbol{w}, \boldsymbol{x}_j^{(n)} \rangle > 0, \qquad (1)$$

where $\boldsymbol{w}$ is a weight vector and $\langle \cdot, \cdot \rangle$ denotes dot product between vectors. This can be placed in a binary SVM classification framework,

$$\langle \boldsymbol{w}, \boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)} \rangle = \left\{ \begin{array}{ll} +1 & \text{if } r_i^{(n)} \succ r_j^{(n)} \\ -1 & \text{if } r_j^{(n)} \succ r_i^{(n)} \end{array} \right. , \qquad (2)$$

which can be solved by the following optimization problem,

$$\min \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i,j,k} \xi_{i,j}^{(k)} \qquad (3)$$

subject to the following constrains:

$$\begin{array}{rcl} \forall (i,j) \in \boldsymbol{r}^{(1)} & : & \langle \boldsymbol{w}, \boldsymbol{x}_i^{(1)} - \boldsymbol{x}_j^{(1)} \rangle \geq 1 - \xi_{i,j}^{(1)} \\ \forall (i,j) \in \boldsymbol{r}^{(n)} & : & \langle \boldsymbol{w}, \boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)} \rangle \geq 1 - \xi_{i,j}^{(n)} \\ \forall (i,j,k) & : & \xi_{i,j}^{(k)} \geq 0 \end{array}$$

where $\boldsymbol{r}^{(n)} = [r_1^{(n)}, r_2^{(n)}, \ldots, r_{p^{(n)}}^{(n)}]$, $C$ is a hyper-parameter which allows trade-off between margin size and training error, and $\xi_{i,j}^{(k)}$ is training error.

### 2.2. Perceptron variant

A problem arises when the number of samples is large as it requires high computational cost, thus we propose and implement a perceptron style algorithm for Ranking SVM in order to facilitate on-line learning in the image retrieval task. Consider the error term in the optimization problem 3,

$$\langle \boldsymbol{w}, (\boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)}) \rangle \geq 1 - \xi_{i,j}^{(n)}. \qquad (4)$$

In order to ensure convergence, we introduce a control term for the margin, $f_\lambda = \lambda |r_i^{(n)} - r_j^{(n)}|$, into the loss. This also has the effect of allowing the algorithm to learn a degree of separation between different ranks, rather than simply aiming to optimize the order as in the Ranking SVM algorithm. This gives the following optimization problem,

$$\min \sum_{i,j,n} h(f_\lambda - \boldsymbol{w}^\mathsf{T} (\boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)})). \qquad (5)$$

The function $h(z)$ denotes the hinge loss,

$$h(z) = \left\{ \begin{array}{ll} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{array} \right. . \qquad (6)$$

The above optimization problem has subgradient with respect to $\boldsymbol{w}$,

$$\partial h(f_\lambda - \langle \boldsymbol{w}, (\boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)}) \rangle)|_{\boldsymbol{w}} = \qquad (7)$$
$$\left\{ \begin{array}{ll} -(\boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)}) & \text{if } f_\lambda - \langle \boldsymbol{w}, (\boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)}) \rangle > 0 \\ 0 & \text{otherwise} \end{array} \right.$$

The learning rate can be defined by step size $s$, then we can obtain the Ranking SVM perceptron-like as shown in Algorithm 1. Convergence is declared when the relative change in the norm of the coefficient vectors $\boldsymbol{w}$ is less than some threshold, $\gamma \ll 1$. Here, $\lambda$ is equal to 1. The algorithm will stop when either the convergence is declared or the iteration reaches $N_{It}$.

```
Input: Sample set of $\{(\boldsymbol{x}_i^{(n)}, r_i^{(n)})\}$, step size $s$, and $\lambda$
Output: $\boldsymbol{w} \in \Re^m$
Initialization: $\boldsymbol{w}_t = \mathbf{0}, t = 1$;
while $t \leq N_{It}$ or $\frac{\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|}{\|\boldsymbol{w}_{t-1}\|} \geq \gamma$ do
    for $n = 1, 2, \ldots, k$ do
        read output: $\boldsymbol{r}^{(n)}$;
        read input: $\boldsymbol{x}^{(n)}$;
        sort
        $\{(\boldsymbol{x}_1^{(n)}, r_1^{(n)}), (\boldsymbol{x}_2^{(n)}, r_2^{(n)})\}, \ldots, (\boldsymbol{x}_{p^{(n)}}^{(n)}, r_{p^{(n)}}^{(n)})\}$ in
        order of rank from most to least relevance;
        for $i = 1, \ldots, p^{(n)} - 1$ do
            for $j = i + 1, \ldots, p^{(n)}$ do
                if $r_i \succ r_j$ then
                    if $\langle \boldsymbol{w}, (\boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)}) \rangle \leq \lambda |r_i^{(n)} - r_j^{(n)}|$
                    then
                        $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + s(\boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)})$;
                        $t = t + 1$
                    end
                end
            end
        end
    end
end
```

**Algorithm 1**: Perceptron Ranking Algorithm

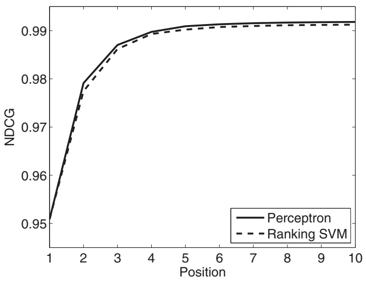

Figure 1. Synthetic Dataset: Comparison of NDCG at each position of Ranking SVM and the proposed perceptron-like algorithm.

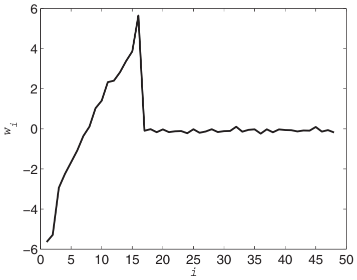

Figure 2. The coefficient $\boldsymbol{w}$ computed by the perceptron algorithm. Features 1–16 are a histogram feature vector computed from "red", features 17–32 are "green", and features 33–48 are "blue".

# 3. Experimental Setup

We first evaluate the Ranking SVM and perceptron algorithm on a synthetic data set. Then we compare both methods on our eye-tracking dataset in an image-search scenario. Our tasks involve several ranks, rather than binary judgements, thus we use the normalized discount cumulative gain (NDCG) [6] as a performance metric. NDCG is designed for tasks which have more than two levels of relevance judgement, and is defined as,

$$\mathrm{NDCG}_k(r,n) = \frac{1}{N_n}\sum_{i=1}^{k} D(r_i)\varphi(g_{ni}) \qquad (8)$$

with $D(r) = \frac{1}{\log_2(1+r)}$ and $\varphi(g) = 2^g - 1$, where $n$ is a page number, $r$ is rank position, $k$ is a truncation level (position), $N$ is a normalizing constant which makes the perfect ranking (based on $g_{ni}$) equal to one, and $g_{ni}$ is the categorical grade; *e.g.*, grade is equal to 5 for the $1^{\mathrm{st}}$ rank and 0 for the $6^{\mathrm{th}}$.

## 3.1. Synthetic dataset

In order to test the performance of the proposed algorithm, we create a synthetic data by randomly selecting 5000 images from the Pascal Visual Objects Challenge 2007 database [2]. The images are divided into 500 pages which give 10 images per page. Each image is given a rank in order of "redness". A Feature vector of an image is represented by 16x3 bins RGB histogram. A leave-one-page-out procedure is used to test the performance of the algorithms, where one page is left out for testing and the training set is the remainder of the pages. The models are selected based on $\mathrm{NDCG}_{10}$. Figure 1 shows NDCG of each position for both methods and the proposed algorithm is slightly better than Ranking SVM.

Figure 2 shows the value of $\boldsymbol{w}$ learned by perceptron algorithm. We can see that the algorithm only weights the histogram feature vectors computed on red while small values or zeros are put on green and blue as we expected.

## 3.2. Ranking images

In this experiment we use a more realistic search scenario. Users are shown 10 images on a page in a five by two grid and they are asked to rank the top five images in order of relevance to the topic of "transport". It should be noted that this concept is deliberately slightly ambiguous given the context of images that were displayed. Each page contains 1–3 clearly relevant images (*e.g.* a freight train, cargo ship or airliner), 2–3 either borderline or marginally relevant images (*e.g.* bicycle or baby carrier), and the rest are non-relevant images (*e.g.* images of people sitting at a dining room table, or a picture of a cat). The experiment has 30 pages, each showing 10 images from the Pascal Visual Objects Challenge 2007 database. The interface consisted of selecting radio buttons (labeled $1^{\mathrm{st}}$ to $5^{\mathrm{th}}$ under each image) then clicking on next to retrieve the next page. This represents data for a ranking task where explicit ranks are given to compliment any implicit information contained in the eye movements. An example of each page is shown in figure 3.

The experiment was performed by six different users, with their eye movements recorded by a Tobii X120 eye tracker which was connected to a PC using a 19-inch monitor (resolution of 1280x1024). The eye tracker has approximately 0.5 degrees of accuracy with a sample rate of 120 Hz and uses infrared leds to detect pupil centers and corneal reflection.

Any pages that contain less than five images with gaze points (for example due to the subject moving and the eye-tracker temporarily losing track of the subject's eyes) were discarded. Hence, only 29 and 20 pages are valid for user 4 and 5, respectively.

# 4. Feature extraction

In these experiments we use standard image histograms and also features obtained from the eye-tracking. The task is then to predict relevant images based on individual image

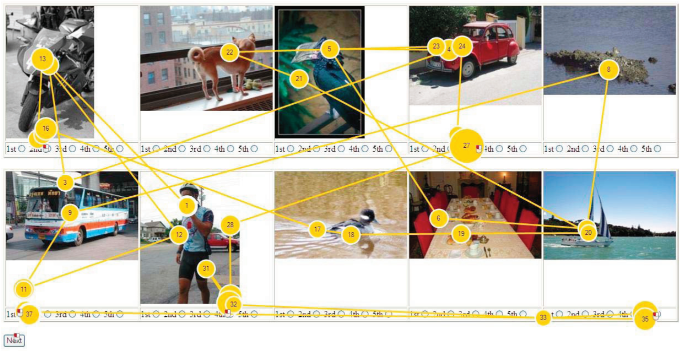

Figure 3. An example of a set of images and the interfaces with overlaid eye movement measurements. The circles mark fixations.

or eye-track features only, or simple combinations including a basic linear sum and using histograms from sub-parts of an image in which the user focussed. First let us discuss the features obtained from the output of the eye-tracking device.

### 4.1. Eye movements

We first consider only features computed for each full image. All features are computed based on only the eye trajectory and locations of the images in the page. This kind of features are general-purpose and easily applicable in all application scenarios. The features are divided into two categories; the first uses directly the raw measurements obtained from the eye-tracker, whereas the second category is based on fixations estimated from the raw data. A fixation means a period in which a user maintains their gaze around a given point. These are important as most visual processing happens during fixations, due to blur and saccadic suppression during the rapid saccades between fixations (see, *e.g.* [3]). Often visual attention features are hence based solely on fixations and relations between them [12]. However, raw measurement data might be able to overcome possible problems caused by imperfect fixation detection.

Table 1 shows the list of candidate features considered. Most of the features are motivated by features considered earlier for text retrieval studies [13]. The features cover the three main types of information typically considered in reading studies: fixations, regressions (fixations to previously seen images), and refixations (multiple fixations within the same image). However, the actual forms of the features have been tailored towards being more suitable for images, trying to include measures for things that are not relevant for texts, such as how big a portion of the image was covered. The features are intentionally kept relatively simple, with the intent that they are more likely to generalize over different users. Fixations were detected using the standard ClearView fixation filter provided with the Tobii eye-tracking software, with settings “radius 30 pixels, minimum duration 100 ms”. These are also the settings recommended for media with mixed content [15].

Some of the features are not invariant of the location of the image on the screen. For example, the typical pattern of moving from left to right means that the horizontal coordinate of the first fixation for the left-most image of each row typically differs from the corresponding measure on the other images. Features that were observed to be position-dependent were normalized by removing the mean of all observations sharing the same position, and are marked in Table 1. Finally, each feature was normalized to have unit variance and zero mean.

### 4.2. Histogram Image Features

As a baseline for simple image features we used an 8-bin grayscale histogram as image-only features. However, we also produced histograms on sub-parts of an image which corresponded to areas on which the user fixated – thus enabling an eye-driven combination of features. Each image is divided into five segments: four quadrants and a central region as shown in figure 4. The feature vector is therefore a combination of five 8-bin grey scale histograms. Any segment which has no gaze information from the user is set to zero, thus incorporating both image and eye movement features.

## 5. Results and Discussion

We evaluate three different scenarios for learning rankings: (i) a global model using data from all users, (ii) using data from other users to predict rankings for a new user, and (iii) predicting rankings on a page given only other data from a single specific user.

We compare the algorithms using different feature sets:

| Number | Name | Description |
|---|---|---|
| Raw data features | | |
| 1 | numMeasurements | total number of measurements |
| 2 | numOutsideFix | total number of measurements outside fixations |
| 3 | ratioInsideOutside | percentage of measurements inside/outside fixations |
| 4 | xSpread | difference between largest and smallest x-coordinate |
| 5 | ySpread | difference between largest and smallest y-coordinate |
| 6 | elongation | ySpread/xSpread |
| 7 | speed | average distance between two consecutive measurements |
| 8 | coverage | number of subimages covered by measurements[1] |
| 9 | normCoverage | coverage normalized by numMeasurements |
| 10* | landX | x-coordinate of the first measurement |
| 11* | landY | y-coordinate of the first measurement |
| 12* | exitX | x-coordinate of the last measurement |
| 13* | exitY | y-coordinate of the last measurement |
| 14 | pupil | maximal pupil diameter during viewing |
| 15* | nJumps1 | number of breaks longer than 60 ms[2] |
| 16* | nJumps2 | number of breaks longer than 600 ms[2] |
| Fixation features | | |
| 17 | numFix | total number of fixations |
| 18 | meanFixLen | mean length of fixations |
| 19 | totalFixLen | total length of fixations |
| 20 | fixPrct | percentage of time spent in fixations |
| 21* | nJumpsFix | number of re-visits to the image |
| 22 | maxAngle | maximal angle between two consecutive saccades[3] |
| 23* | landXFix | x-coordinate of the first fixation |
| 24* | landYFix | y-coordinate of the first fixation |
| 25* | exitXFix | x-coordinate of the last fixation |
| 26* | exitYFix | y-coordinate of the last fixation |
| 27 | xSpreadFix | difference between largest and smallest x-coordinate |
| 28 | ySpreadFix | difference between largest and smallest y-coordinate |
| 29 | elongationFix | ySpreadFix/xSpreadFix |
| 30 | firstFixLen | length of the first fixation |
| 31 | firstFixNum | number of fixations during the first visit |
| 32 | distPrev | distance to the fixation before the first |
| 33 | durPrev | duration of the fixation before the first |

[1] The image was divided into a regular grid of 4x4 subimages.

[2] A sequence of measurements outside the image occurring between two consecutive measurements within the image.

[3] A transition from one fixation to another.

Table 1. List of features considered in the study. First 16 features are computed from the raw data, whereas the rest are based on pre-detected fixations. Note that features 2 and 3 use both types of data since they are based on raw measurements not belonging to fixations. All features are computed separately for each image. Features marked with * were normalized for each image location; see text for details.

information from eye movements only (EYE), image-only histogram features (HIST), histogram features based on the 5-regions as described above (HIST5), a simple linear combination of eye movements and histogram features (EYE+HIST) and finally whole-page eye movement features combined with histogram features based on the five regions (EYE+HIST5).

We found that although the topic was left deliberately vague, the amount of agreement in the rankings (including non-relevant images which are treated as tie ranks) between users was large in each page ($p < 0.01$). The statistical significance of the level of agreement is tested using the Kendall Coefficient of Concordance ($W$) [14] which is used to measure the degree of agreement between the rankings assigned to objects.

In order to test the model, we used a leave-one-out cross validation approach. Leave-one-out cross validation is applied to obtain the optimal model: $C$ for Ranking SVM,

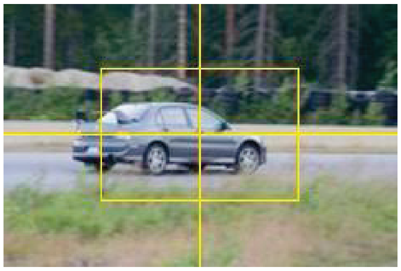

Figure 4. Each image is divided into five segments.

and $s$ for the proposed algorithm. The models are selected based on maximum NDCG$_{10}$.

## 5.1. Global Model – All Users

In this scenario, we train the model given data from all users. It aims to test how useful the gaze data is in the ranking task across all the users. The model is trained using all pages of all users whilst leaving one page out for testing purposes. The perceptron ranking algorithm is compared with Rank-SVM and results are shown in figure 5. The perceptron clearly outperforms Rank-SVM for all features sets. We can see that the proposed perceptron algorithm with all the feature sets are able to achieve higher performance over a random baseline as shown in figure 6. It is clear that using information from eye movements alone is better than using only image histograms $(p < 0.01)$. The significance level is tested using the sign test [14]. However, the results from linearly combining the eye movements and histogram-based features does represent an improvement $(p < 0.01)$. Simply breaking up the image histogram into the five segments and only using those areas which the user looked at (HIST5) always increases performance against whole-image histograms $(p < 0.01)$ and is also better than linearly combining the eye movements and histogram-based features $(p < 0.01)$. However, using EYE+HIST5 gave the best performance among all sets of features $(p < 0.01)$. Indicating that eye-driven features are potentially very useful in such applications.

## 5.2. Global Model – New User

Leave-one-out cross validation is also used in this scenario, however in this case all data for a specific user is left out for each testing phase; thus representing the case when a new user is being encountered. The results are shown in figure 7 and figure 8. Using information from eye movements is better than using information based purely on image histogram in five users $(p < 0.01)$. Other results follow the same pattern as in the previous experiment, with the exception of the combination of EYE with HIST5 (only a significance of $p < 0.1$ is obtained over HIST5). In most cases performance between these features were similar, but for certain users (such as User 1) – the presence of eye movement data greatly enhances the result. This is possibly due to this user not fitting the global model in this case, and therefore the eye movements become a strong discriminative factor.

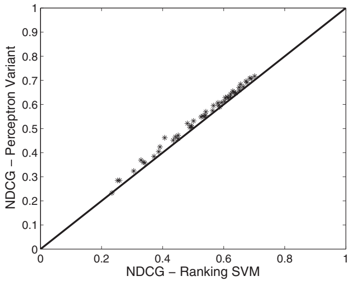

Figure 5. A comparison of NDCG at all positions. The proposed perceptron algorithm is clearly better than Ranking SVM as all the points fall above the diagonal line.

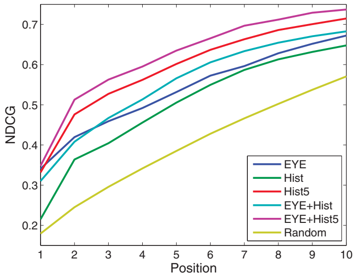

Figure 6. Global model - all users, the average NDCG at each position across all users using five different sets of features.

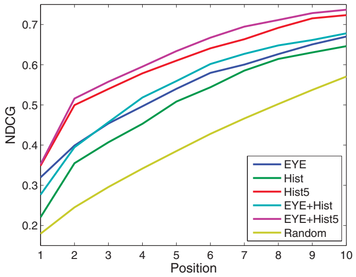

Figure 7. Global model - new user, the average NDCG at each position across all users using five different sets of features.

We further compare the global model for new user together with the global model for all users on EYE+HIST5 features set. The results are shown in figure 10. The global model for all users is slightly better than the global model for new user $(p = 0.0941)$.

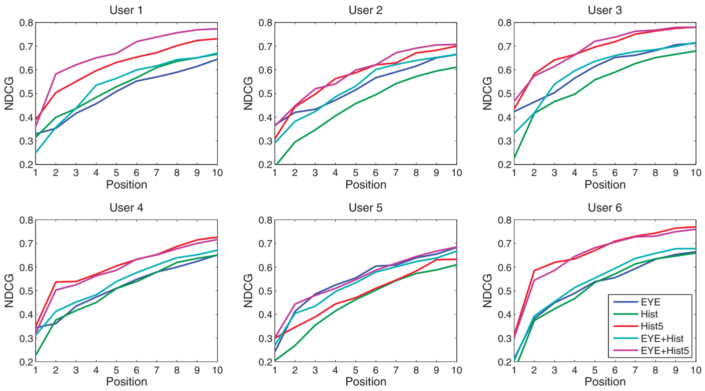

Figure 8. Global model - new user, it shows results of NDCG at each position for individual user using five different sets of features.

## 5.3. User-specific Model

In this scenario, each user has a separate model, and for each user a leave-one-page-out cross validation procedure is used for parameter settings and evaluation of the results. The results are shown in figure 9 and figure 11. It should be noted that we have a limited number of training samples as we only collected 30 pages from each user in this model. From the results one can observe that in general using information from eye movements is often better than classifying purely based on image histograms. Although this is not always the case, the histogram approach may be slightly misleading in that transport images often contain a large portion of sky (as they are often taken outside). Again, the results in the user-specific model are very much the same as the other models. However, in this model combining EYE with HIST5 is once again better than HIST5 at the significance level of $p < 0.01$.

Finally, the three different models are compared together using EYE+HIST5 features set as shown in 10. The user-specific model is clearly worse than both global models. This is most likely caused by having considerably smaller amounts of training data; the user-specific model only has 29 pages for training (if there is no page to be discarded) whereas global model has roughly 138–168 pages. Particularly for user 6, the user-specific model achieves higher performance than the user's global model though the model was trained with a small training set. This shows that user adaptation is very useful.

## 6. Conclusions

In this paper we have adapted and improved Ranking SVM through a perceptron-style algorithm for online learning of rankings. We have demonstrated that it performs as well as or better than conventional Ranking SVM on both synthetic and real-world data. We provide some initial ex-

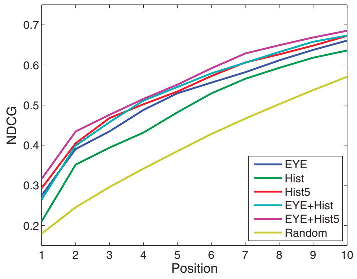

Figure 9. User-specific model, the average NDCG at each position across all users using five different sets of features.

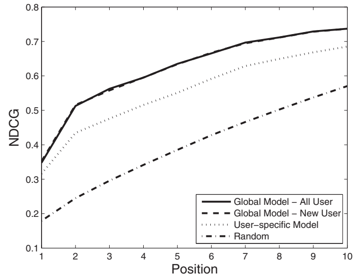

Figure 10. A comparison of NDCG at each position on three different models using Eye+Hist5 features set.

periments based on a simple linear combination of a standard image metric (namely histograms) and features gained from the eye movements, in a novel image-search setting. The experiment shows that the performance of the search can be improved when we fuse simple images features and implicit feedback together. This shows that metric information based on eye movements can be useful, and suggests that there is a large amount of potential in exploiting this information in image retrieval, HCI and many other settings.

Experience with this task showed that it actually took

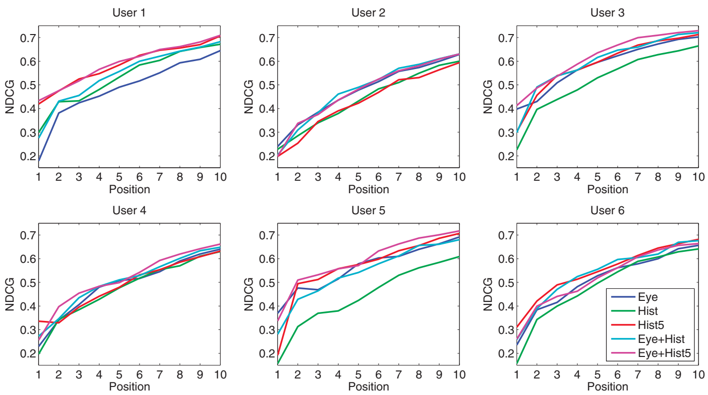

Figure 11. User-specific model, it shows results of NDCG at each position for individual user using five different sets of features.

quite a lot of cognitive processing on the part of the participant. It is unclear how the user interface affected the process for this task, as the temptation is often to click as the images are seen. However, most users ranking the images internally before clicking on the radio buttons. In some cases mistakes were made and the user had to return and rerank or add missing ranks, so post-processing of this data will need to be done with care.

## 7. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under *grant agreement* n° 216529, Personal Information Navigator Adapting Through Viewing, PinView and TKK MIDE programme, project UI-ART.

## References

[1] G. Buscher, A. Dengel, and L. van Elst. Eye movements as implicit relevance feedback. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2991–2996, New York, NY, USA, 2008. ACM.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[3] R. Hammoud. *Passive Eye Monitoring: Algorithms, Applications and Experiments*. Springer-Verlag, 2008.

[4] D. Hardoon, J. Shawe-Taylor, A. Ajanki, K. Puolamäki, and S. Kaski. Information retrieval by inferring implicit queries from eye movements. In *AISTATS '07: Proceeding of International Conference on Artificial Intelligence and Statistics*, 2007.

[5] R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.

[6] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, New York, NY, USA, 2000. ACM.

[7] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM Press.

[8] A. Klami, C. Saunders, T. E. de Campos, and S. Kaski. Can relevance of images be inferred from eye movements? In *MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pages 134–140, New York, NY, USA, 2008. ACM.

[9] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. Picsom–content-based image retrieval with self-organizing maps. *Pattern Recognition Letter*, 21(13-14):1199–1207, 2000.

[10] O. Oyekoya and F. Stentiford. Perceptual image retrieval using eye movements. *International Journal of Computer Mathematics*, 84(9):1379–1391, 2007.

[11] K. Puolamäki, J. Salojärvi, E. Savia, J. Simola, and S. Kaski. Combining eye movements and collaborative filtering for proactive information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 146–153, New York, NY, USA, 2005. ACM.

[12] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372–422, November 1998.

[13] J. Salojärvi, K. Puolamäki, J. Simola, L. Kovanen, I. Kojo, and S. Kaski. Inferring relevance from eye movements: Feature extraction. Technical Report A82, Computer and Information Science, Helsinki University of Technology, 2005.

[14] S. Siegel and N. J. Castellian. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill, Singapore, 1988.

[15] Tobii Technology, Ltd. Tobii Studio Help. http://studiohelp.tobii.com/StudioHelp_1.2/.

[16] D. J. Ward and D. J. C. MacKay. Fast hands-free writing by gaze direction. *Nature*, 418(6900):838, 2002.