UNIVERSITY OF Southampton
School of Electronics and Computer Science

PinView

Deliverable D5.1

Ranking algorithms for implicit feedback

Contract number: **FP7–216529** PinView

Personal Information Navigator Adapting Through Viewing

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.

European Commission

SEVENTH FRAMEWORK PROGRAMME

Identification sheet

Project ref. no.	**FP7–216529**
Project acronym	PinView
Status and version	Final, Revision: 1.10
Contractual date of delivery	31.12.2009
Actual date of delivery	29.12.2009
Deliverable number	D5.1
Deliverable title	Ranking algorithms for implicit feedback
Nature	Report
Dissemination level	PU – Public
WP contributing to the deliverable	WP5 Model adaptation and user personalisation
Task contributing to the deliverable	Task 5.1 Ranking algorithms for implicit feedback
WP responsible	School of Electronics & Computer Science, University of Southampton
Task responsible	School of Electronics & Computer Science, University of Southampton
Editor	Kitsuchart Pasupa, <kp2@ecs.soton.ac.uk>
Editor address	School of Electronics & Computer Science, University of Southampton, Southampton SO17 1BJ, United Kingdom
Authors in alphabetical order	Peter Auer, Steve R. Gunn, David R. Hardoon, Samuel Kaski, Arto Klami, Alex Leung, Kitsuchart Pasupa, Craig J. Saunders, and Sandor Szedmak
EC Project Officer	Pierre-Paul Sondag
Keywords	implicit relevance feedback, eye movements, ranking, tensor, support vector machine, image retrieval
Abstract	This report presents novel algorithms to use eye movements as an implicit relevance feedback in order to improve the performance of the searches. The algorithms are evaluated on "Transport Rank Five" Dataset which were previously collected in Task 8.3. We demonstrated that simple linear combination or tensor product of eye movement and image features can improve the retrieval accuracy.

List of annexes

none

Contents

1 Overview **4**

2 Introduction **5**

3 Learning to Rank Images from Eye Movements **6**
3.1 Methodologies 6
3.1.1 Ranking SVM 6
3.1.2 Perceptron Variant 7
3.2 Experimental Setup 8
3.2.1 Synthetic Dataset 8
3.2.2 Ranking Images 9
3.3 Feature Extraction 10
3.3.1 Eye Movements 11
3.3.2 Histogram Image Features 11
3.4 Results and Discussion 11
3.4.1 Global Model – All Users 13
3.4.2 Global Model – New User 13
3.4.3 User-specific Model 14

4 Image Ranking with Implicit Feedback from Eye Movements **17**
4.1 Ranking SVM as a Baseline SVM 18
4.2 Tensor Ranking SVM 18
4.2.1 Decomposition 19
4.3 Experiments 20
4.3.1 Page Generalisation 20
4.3.2 User Generalisation 21
4.4 Discussion 23

5 Tensor Kernelised LinRel **23**

6 Conclusions **25**

1 Overview

This deliverable constitutes the output of Task 5.1 *Ranking algorithms for implicit feedback* of the *Personal Information Navigator Adapting Through Viewing*, PinView, project, funded by the European Community's Seventh Framework Programme under Grant Agreement n° 216529.

This task aims to develop algorithms which use implicit relevance feedback from the user (i.e. eye movements) to improve the retrieval accuracy. We introduce a new search strategy which uses eye movements information and image features to rank images and show that eye movements information is useful in a search task. We introduce a ranking algorithm the so-called "tensor Ranking Support Vector Machine" which fuses image features with eye movements information. We show that the joint learnt semantic space of eye and image features can be efficiently decomposed into its independent sources allowing us to further test or train only using images. Furthermore, we will extend kernelised LinRel – an on-line learning algorithm developed in Work Package 4 to learn from multiple sources (i.e. eye movements, images).

The results of this task will be integrated in Task 8.5 for deliverable D8.5.2: (i) a new image ranking algorithm, tensor Ranking Support Vector Machine which combines image features with eye movements features. (ii) an extension of kernelised LinRel – the current state of Task 8.5 for deliverable D8.5.1 to learn from multiple sources.

The involvement of TKK in this task has consisted of the eye movements feature extraction that is used in all the experiments in this deliverable. UCL's involvement in this task was considering the development of tensor Ranking Support Vector Machine. The involvement of MUL was conducting the extension of kernelised LinRel to learn from multiple sources.

2 Introduction

This task aims to develop new image search strategies and ranking algorithms which use implicit relevance feedback from the user (i.e. eye movements) and explicit relevance feedback from the user (i.e. user clicks) to improve the retrieval accuracy. This report consists of three main sections.

Section 3 introduces 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, a perceptron formulation of the Ranking Support Vector Machine (Ranking SVM) algorithm is developed. 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. This section was published in Proceedings of 2009 IEEE 12th International Conference on Computer Vision (ICCV'2009) Workshop on Human-Computer Interaction (HCI'2009) [18].

In section 4, we explore the idea of implicitly incorporating eye movement features in an image ranking task where only images are available during testing. The first section had demonstrated that combining eye movement and image features improved on the retrieval accuracy when compared to using each of the sources independently. Despite these encouraging results the proposed approach is unrealistic as no eye movements will be presented a-priori for new images (i.e. only after the ranked images are presented would one be able to measure a user's eye movements on them). Hence, in this section, we propose a novel search methodology which combines image features together with implicit feedback from users' eye movements in a tensor Ranking SVM and show that it is possible to extract the individual source-specific weight vectors. Furthermore, we demonstrate that the decomposed image weight vector is able to construct a new image-based semantic space that outperforms using solely the image features. This section was published in Proceedings of the Neural Information Processing Systems (NIPS'2009) Workshop on Advances in Ranking [19] and will be appeared in Proceedings of Eye Tracking Research & Applications (ETRA'2010) [9].

In section 5, we aim to accommodate the kernelised LinRel developed in Work Package 4 to learn from multiple sources (i.e. image features, eye movements features) using tensor kernel ideas in section 4. Preliminary results are shown in Deliverable 4.2 with image and eye movements information used in the feature vector for LinRel. The results show that the performance of on-line learning using LinRel is improved when eye movements information is added into the feature vector. Again, this is not realistic because there is no eye movements information for unseen images in database. Hence, tensor kernel will be used as a principled way to combine image features with eye movements information in the feature vector for kernelised LinRel.

3 Learning to Rank Images from Eye Movements

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 [15].

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. [23]), however most of the human-computer interface (HCI) works treated eye movement as an input or explicit feedback mechanism e.g. [27]. 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 [21, 10, 5]. To the best of our knowledge however, at the time of writing, only [17, 14] 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" [13], which was developed to automatically improve the retrieval quality of a search engine using click-through data. In this section we adapt the Ranking SVM into a perceptron-style algorithm in order to suit the setting of on-line learning, as well as improving its computation performance.

The section is organised as follows. Subsection 3.1 outlines the Ranking SVM algorithm and introduces our proposed perceptron algorithm. Subsection 3.2 explains our ranking experimental framework, and Subsection 3.3 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 Subsection 3.4.

3.1 Methodologies

3.1.1 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 section, subscripts and superscripts indicate the index of images and pages respectively. The exact nature of these features are explained in detail in section 3.3. 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 [13] and is adapted from ordinal regression [11]. It is a pair-wise approach where the solution is a binary classification problem. Consider a linear ranking function,

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

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,

$$\langle \boldsymbol{w}, \boldsymbol{x}_i^{(n)} - \boldsymbol{x}_j^{(n)} \rangle = \begin{cases} +1 & \text{if } r_i^{(n)} \succ r_j^{(n)} \\ -1 & \text{if } r_j^{(n)} \succ r_i^{(n)} \end{cases}, \tag{2}$$

which can be solved by the following optimisation problem,

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

subject to the following constrains:

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

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.

3.1.2 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 optimisation problem (3),

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

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 optimise the order as in the Ranking SVM algorithm. This gives the following optimisation problem,

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

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

$$h(z) = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

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

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

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, λ 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 λ
Output: $\boldsymbol{w} \in \Re^m$
Initialisation: $\boldsymbol{w}_t = \boldsymbol{0}, t = 1$;
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

3.2 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 normalised discount cumulative gain (NDCG) [12] as a performance metric. NDCG is designed for tasks which have more than two levels of relevance judgement, and is defined as,

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

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 normalising 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^{st} rank and 0 for the 6^{th}.

3.2.1 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 [7]. 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 NDCG_{10}. Figure 1 shows NDCG of each position for both methods and the proposed algorithm is slightly better than Ranking SVM.

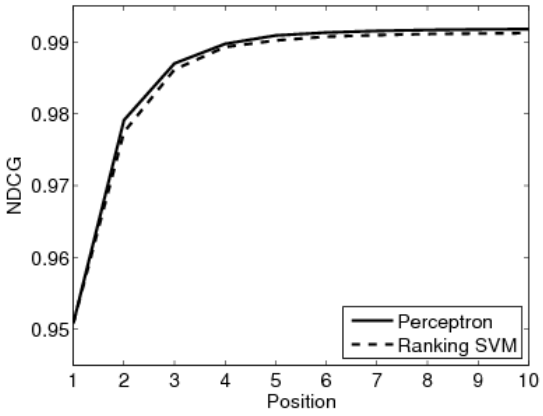


Figure 1: Synthetic Dataset: Comparison of NDCG at each position of Ranking SVM and the proposed perceptron-like algorithm.

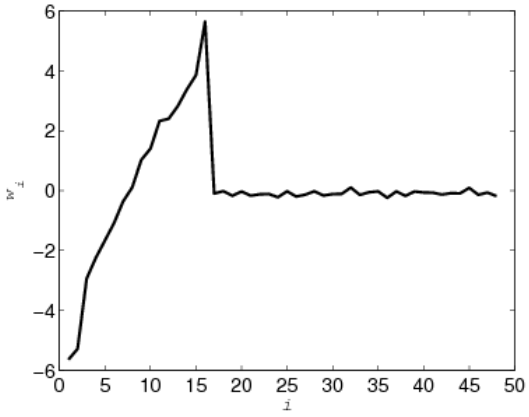


Figure 2: The coefficient $\boldsymbol{w}$ computed by the perceptron algorithm. Features 1–16 are a histogram feature vector computed from "red", features 17–32 are "green", and features 33–48 are "blue".

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.

3.2.2 Ranking Images

The experiment is previously described in deliverable of Task 8.3 [25]. 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 (labelled 1^{st} to 5^{th} under each image) then clicking on next to retrieve the next page. This represents data for a ranking

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

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 centres 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.

Table 1: The data collected per user. *Pages with less than five images with gaze points were removed. Therefore users 4 and 5 only have 29 and 20 pages viewed respectively.

User #	Pages Viewed
1	30
2	30
3	30
4*	29
5*	20
6	30

3.3 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 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.

3.3.1 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. [8]). Often visual attention features are hence based solely on fixations and relations between them [23]. However, raw measurement data might be able to overcome possible problems caused by imperfect fixation detection.

Table 2 shows the list of candidate features considered. Most of the features are motivated by features considered earlier for text retrieval studies [24]. 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 generalise 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[1].

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 normalised by removing the mean of all observations sharing the same position, and are marked in Table 2. Finally, each feature was normalised to have unit variance and zero mean.

3.3.2 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.

3.4 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: 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 fea-

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

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 normalised 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 2: 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 normalised for each image location; see text for details.

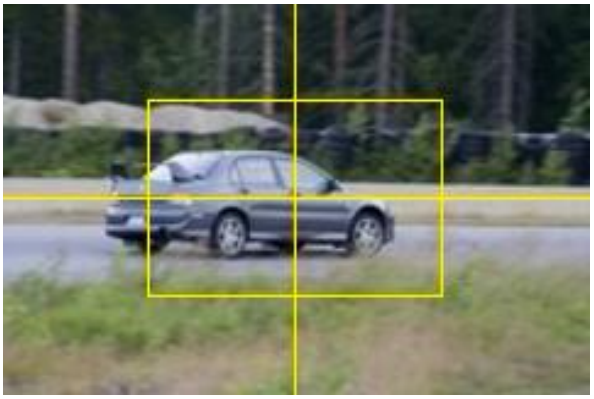


Figure 4: Each image is divided into five segments.

tures (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) [26] 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, and s for the proposed algorithm. The models are selected based on maximum NDCG_{10}.

3.4.1 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 [26]. 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.

3.4.2 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

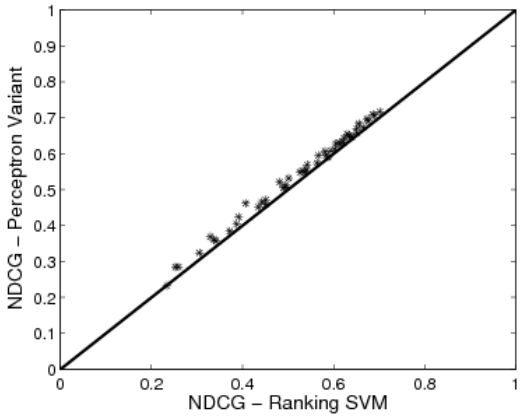


Figure 5: A comparison of NDCG at all positions. The proposed perceptron algorithm is clearly better than Ranking SVM as all the points fall above the diagonal line.

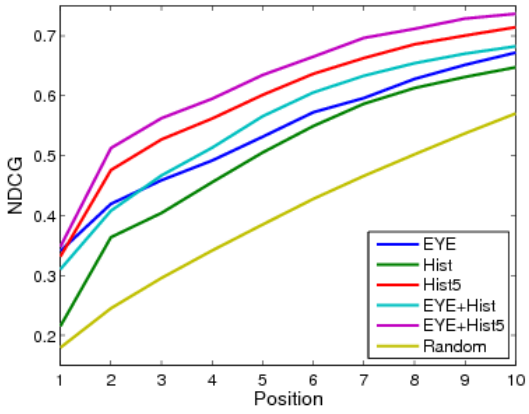


Figure 6: Global model - all users, the average NDCG at each position across all users using five different sets of features.

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.

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 11. The global model for all users is slightly better than the global model for new user ($p = 0.0941$).

3.4.3 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 10. 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

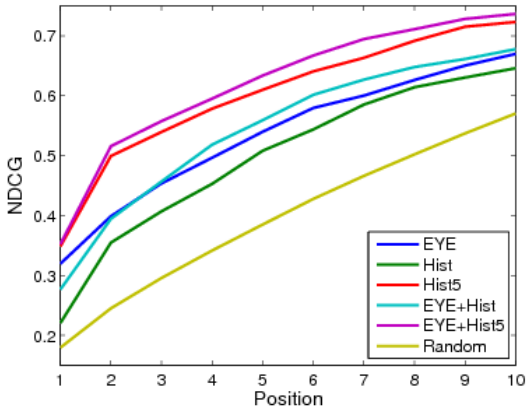


Figure 7: Global model - new user, the average NDCG at each position across all users using five different sets of features.

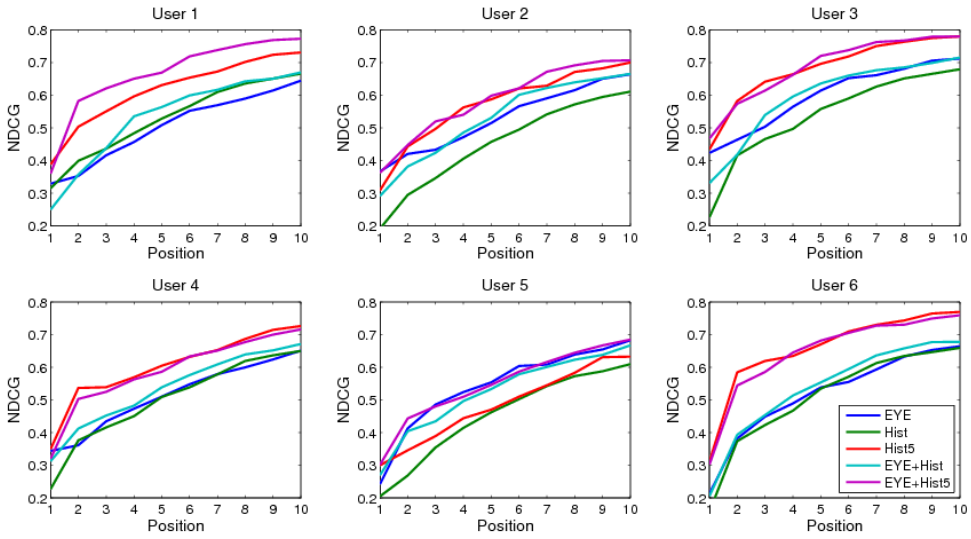


Figure 8: Global model - new user, it shows results of NDCG at each position for individual user using five different sets of features.

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 11. 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.

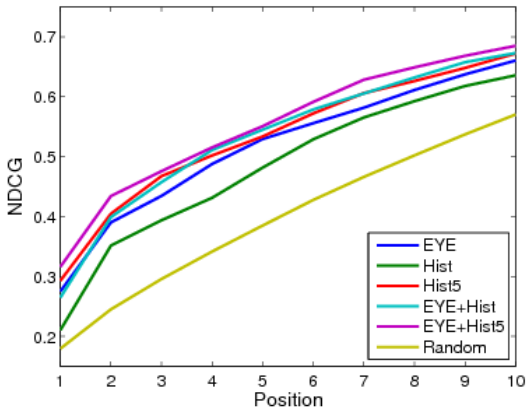


Figure 9: User-specific model, the average NDCG at each position across all users using five different sets of features.

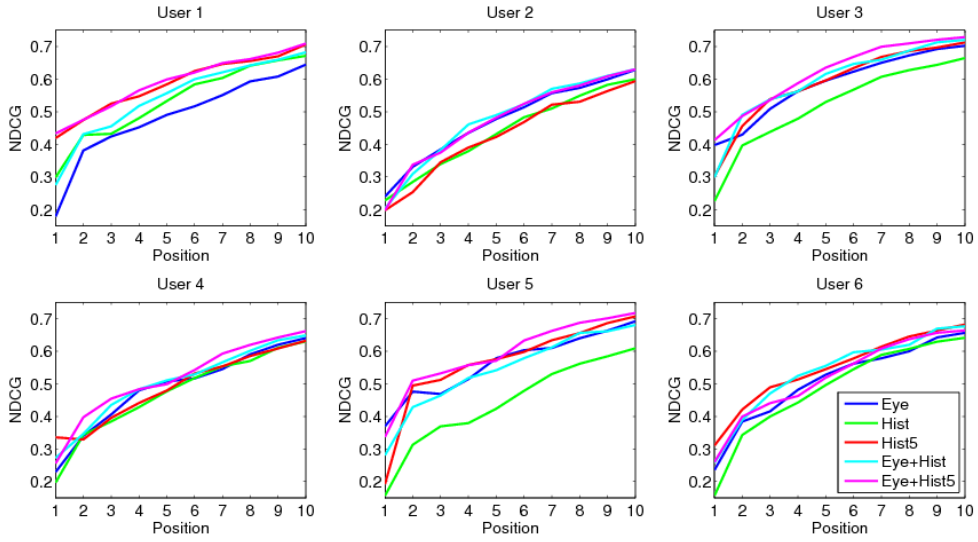


Figure 10: User-specific model, it shows results of NDCG at each position for individual user using five different sets of features.

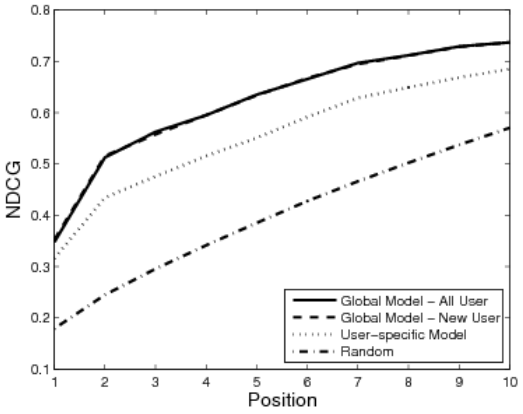


Figure 11: A comparison of NDCG at each position on three different models using Eye+Hist5 features set.

4 Image Ranking with Implicit Feedback from Eye Movements

Despite previous section encouraging results, the proposed approach is largely unrealistic as image and eye features are combined for both training and testing. Whereas in a real scenario no eye movements will be presented a-priori for new images. In other words, only after the ranked images are presented to a user, would one be able to measure the user's eye movements on them. Furthermore, earlier studies of Hardoon et al. [10] and Ajanki et al. [2] explored the problem of where an implicit information retrieval query is inferred from eye movements measured during a reading task. The result of their empirical study is that it is possible to learn the implicit query from a small set of read documents, such that relevance predictions for a large set of unseen documents are ranked significantly better than by random guessing.

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 used in the machine learning literature as a means of predicting edges in a protein interaction or co-complex network by using the tensor product transformation to derive a kernel on protein pairs from a kernel on individual proteins [4, 16, 22]. In this study we use the tensor product to construct a joined semantic space by combining eye movements and image features. Furthermore, we continue to show that the combined learnt semantic space can be efficiently decomposed into its contributing sources (i.e. images and eye movements), which in turn can be used independently.

The section is organised as follows. In subsection 4.1 we give a brief introduction to the ranking SVM methodology and continue to develop in subsection 4.2 our proposed tensor ranking SVM and the efficient decomposition of the joint semantic space into the individual sources. In subsection 4.3 we bring forward our experiments on page ranking for individual users as well as a feasibility study on user generalisation. Finally, we conclude our study with discussion on our present methodology and results in subsection 4.4.

4.1 Ranking SVM as a Baseline SVM

According to the subsection 3.1.1. Linear ranking function can be placed in a binary SVM classification framework where let c_k be the new label indicating the quality of rank k,

$$\langle \boldsymbol{w}, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle = \begin{cases} c_k = +1 & \text{if } r_i \succ r_j \\ c_k = -1 & \text{if } r_j \succ r_i \end{cases}, \tag{9}$$

which can be solved by the following optimisation problem,

$$\min \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_k \xi_k \tag{10}$$

subject to the following constrains:

$$\begin{aligned} \forall (i,j) \in \boldsymbol{r}^{(k)} &: c_k(\langle \boldsymbol{w}, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle + b) \geq 1 - \xi_k \\ \forall (k) &: \xi_k \geq 0 \end{aligned}$$

where $\boldsymbol{r}^{(k)} = [r_1, r_2, \ldots, r_m]$, C is a hyper-parameter which allows trade-off between margin size and training error, and ξ_k is training error. Alternatively, we can represent the ranking SVM as a vanilla SVM where we re-represent our samples as

$$\phi(\boldsymbol{x})_k = \boldsymbol{x}_i - \boldsymbol{x}_j$$

with label c_k and m being the total number of new samples. Finally, we quote from Cristianini and Shawe-Taylor [6] the general dual SVM optimisation as

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j c_i c_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \tag{11}$$

$$\text{subject to } \textstyle\sum_{i=1}^{m} \alpha_i c_i = 0 \text{ and } \alpha_i \geq 0 \quad i = 1, \ldots, m,$$

where we again use c_i to represent the label and $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ to be the kernel function between $\mathbf{x}_i$ and $\mathbf{x}_j$.

4.2 Tensor Ranking SVM

In the following section we propose to construct a tensor kernel on the ranked image and eye movements features, i.e. following equation (9), 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, $\boldsymbol{x}$ and $\boldsymbol{y}$, 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 [20] 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 (11). 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

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

4.2.1 Decomposition

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

$$W \approx W^T = \sum_{t=1}^{T} \mathbf{w}_x^t \mathbf{w}_y^{t\prime}, \tag{12}$$

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

$$WW' = \sum_{i,j}^{m} \alpha_i \alpha_j c_i c_j \kappa_y(\mathbf{y}_i,\mathbf{y}_j)\phi_x(\mathbf{x}_i)\phi_x(\mathbf{x}_j)' \tag{13}$$

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 $K_{ij}^y = \kappa_y(\mathbf{y}_i,\mathbf{y}_j)$. Substituting back into equation (13) 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 $\mathbf{h}_k = \sum_{i=1}^m \alpha_i c_i \mathbf{u}_i^k \phi_x(\mathbf{x}_i)$, hence, 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

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

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

$$H'H = Z\Upsilon V'V\Upsilon Z' = Z\Upsilon^2 Z' \tag{14}$$

with Υ a matrix with υ_t on the diagonal truncated after 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 $\forall i \beta_i$ where we train and test an SVM. i.e. we compute $\tilde{\phi}(\boldsymbol{x}) = K^x \beta$, for the training samples, and $\tilde{\phi}(\boldsymbol{x}_t) = K_t^x \beta$ for our test samples. We explore both these approaches in our experiments.

4.3 Experiments

We evaluate two different scenarios for learning the ranking of image based on image (256-bin grey scale histogram) and eye features;

- Predicting rankings on a page given only other data from a single specific user.
- A global model using data from other users to predict rankings for a new unseen user.

We compare our proposed tensor Ranking SVM algorithm which combines both information from eye movements and image histogram features to a Ranking SVM using histogram features and to a Ranking SVM using eye movements alone. 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 (assuming that $\phi_x(\mathbf{x}_i)$ are taken as the image features in equation (13)).

4.3.1 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:

- $T1$: 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).
- $T2$: 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.
- $T1^{all}$: similar to $T1$ although here we use all t weight vectors and take the mean value across as the final score.
- $T1^{opt}$: 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 NDCG across 10 positions.

We plot the user specific leave-page-out NDCG performances in figure 12 where we are able to observe that $T2$ consistently outperforms the image feature Ranking SVM across all users, demonstrating that it is indeed possible to improve on the image ranking with the incorporation of eye movement features during training. Furthermore, it is interesting to observe that for certain users $T1^{opt}$ significantly improves on the ranking performance, suggesting that there is an optimal combination of the decomposed features that may further improve on the results.

In figure 13 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.

4.3.2 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, i.e;

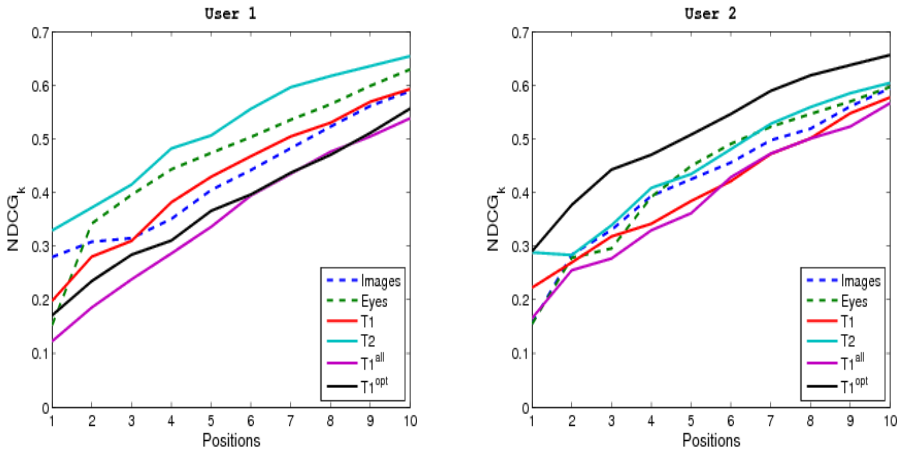
```
For all users
   Withhold data from user i
   For all pages
      Withhold page j from all users
      Train on all pages-j from all users-i
      Test on page j from user i
   Endfor
Endfor
```

Therefore we only use the users from table 1 who viewed the same number of pages, i.e. users 1, 2, 3 and 6, which we refer to henceforth as users 1-4.

We evaluate the proposed approach with the following two setting:

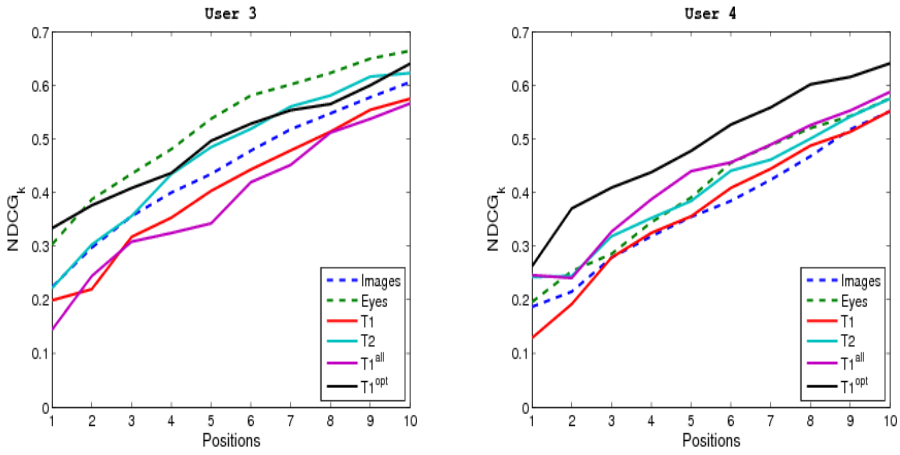
- $T1$: using the largest component of tensor decomposition in the form of a linear discriminator. We use the weight vector corresponding to the largest eigenvalue (as we have a t weights).
- $T2$: we project the image features into the learnt semantic space (i.e. the decomposition on the image source) and train and test within the projected space a secondary Ranking SVM.

We plot in figure 14 the resulting NDCG performance for the leave-user-out routine. We are able to observe, with the exclusion of user 2 in figure 14(b), that $T2$ is able to significantly outperform the Ranking SVM on image features. Indicating that it is possible to generalise our proposed approach across new unseen users. Furthermore, it is interesting to observe that $T2$ achieves a similar performance to that of a Ranking SVM trained and tested on the eye features. Finally, even though we do not improve when testing on data from user 2, we are able to observe that we perform as-good-as the baselines. In figure 14(e) we plot the average NDCG performance on the leave-user-out routine, demonstrating that on average we improve on the ranking of new images for new users.



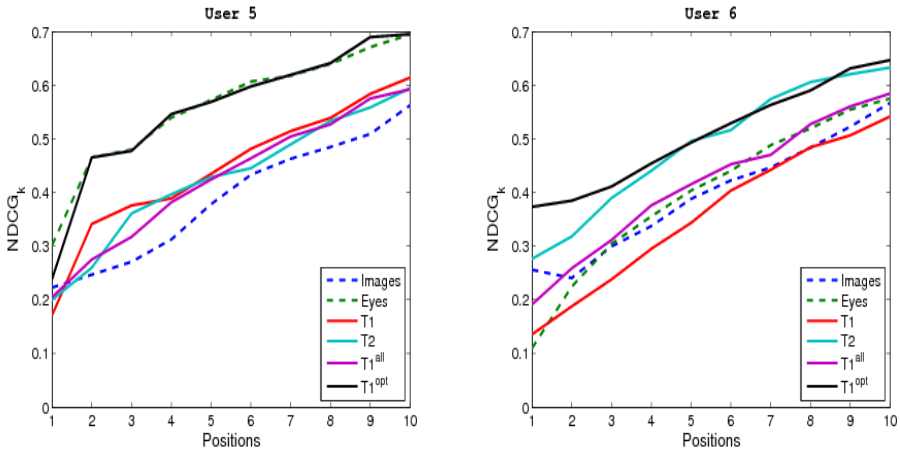
(a) NDCG performance within user 1

(b) NDCG performance within user 2



(c) NDCG performance within user 3

(d) NDCG performance within user 4



(e) NDCG performance within user 5

(f) NDCG performance within user 6

Figure 12: In the following sub-figures 12(a)-12(f) we illustrate the NDCG performance for each user in a leave-page-out routine, i.e. here we aim to generalise over new pages rather new users. We are able to observe that $T2$ and $T1^{opt}$ routinely outperform the ranking with only using image features. The 'Eyes' plot in all the figures demonstrates how the ranking (only using eye-movements) would perform if eye-features were indeed available a-priori for new images.

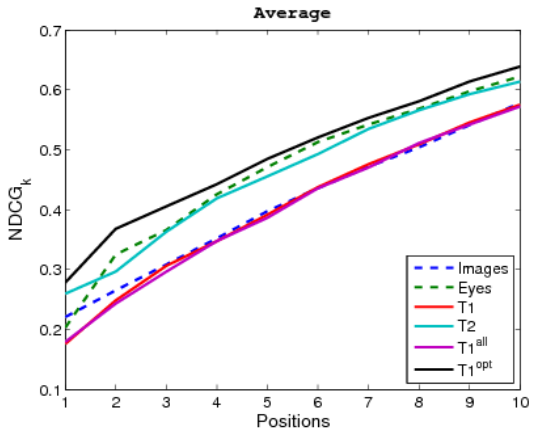


Figure 13: Average NDCG performance across all users for predicting rankings on a page given only other data from a single specific user.

4.4 Discussion

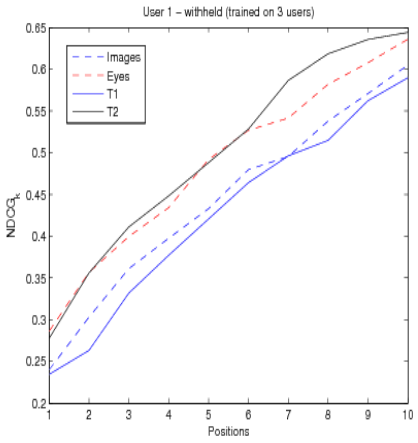
In this section we presented a novel search strategy for combining eye movements and image features with a tensor product kernel used in a Ranking SVM framework. We showed that the joint learnt semantic space of eye and image features can be efficiently decomposed into its independent sources allowing us to further test or train only using images. We explored two different search scenarios for learning the ranking of images based on image and eye features. The first was predicting ranking on a page given only other data from a single specific user. This experiment was to test the fundamental question of whether eye movement are able to improve ranking for a user. Demonstrating that it was indeed possible to improve in the single subject setting, we then proceeded to our second setting where we constructed a global model across users in attempt to generalise on data from a new user. Again our results demonstrated that we are able to generalise our model to new users. Despite these promising results, it was also clear that using a single direction (weight vector) does not necessarily improve on the baseline result. Therefore motivating the need for a more sophisticated combination of the resulting weights. This, as well as extending our experiment to a much larger number of users, will be addressed in a future study. Finally, we would also explore the notion of image segmentation and the use of more sophisticated image features that are easily computable.

5 Tensor Kernelised LinRel

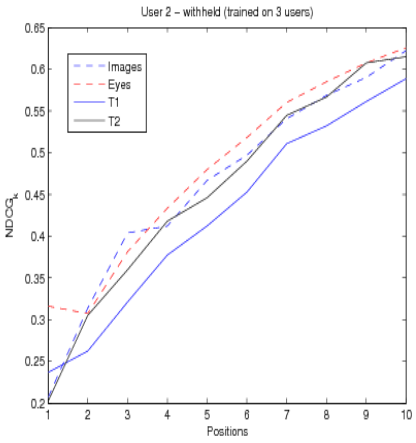
With the balance of exploration and exploitation, LinRel provides an efficient means to choose relevant images to the user. Kernelised LinRel makes use of a kernel matric as a distance matric for on-line learning. Thus, we can accommodate LinRel to learn from multiple sources using tensor which provides an accurate distance metric with image and eye movements informations in sementic space.

The random bandit problem formalises an exploration-exploitation trade-off to maximise the cumulative reward. In 1995, it was shown that a simple algorithm can achieve the optimal performance based on upper confidence bounds for the expected rewards [1].

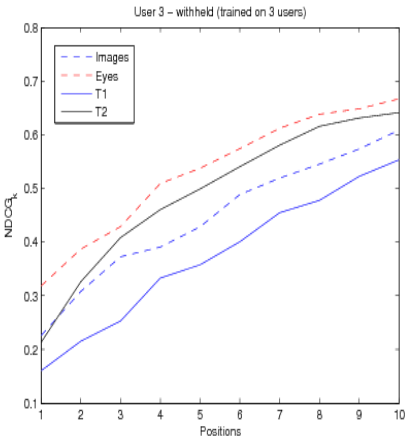
In each trial, t, the algorithm selects the alternative with the largest upper confidence bound $\mu_i(t) + \sigma_i(t)$, where $\mu_i(t)$ and $\sigma_i(t)$ is the expected value and the derivation in the upper confidence bound, respectively. As $\sigma_i(t)$ decreases rapidly with each choice of alternative, i, the number of exploration trials is limited. The use of upper confidence bounds automatically trades off between exploration and exploitation. We use the LinRel algorithm [3] which considers an additional feature vector provided to the learning algorithm. For each



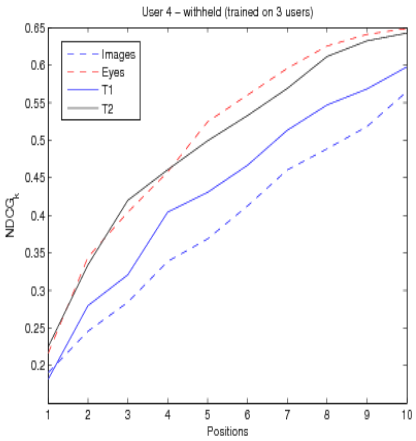
(a) NDCG performance on user 1



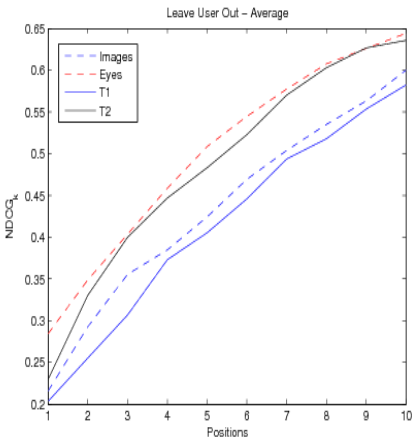
(b) NDCG performance on user 2



(c) NDCG performance on user 3



(d) NDCG performance on user 4



(e) Average NDCG performance

Figure 14: In the following sub-figures 14(a)-14(d) we illustrate the NDCG performance in a leave-user-out (leave-page-out) routine. The average NDCG performance is given in sub-figure 14(e) where we are able to observe that $T2$ outperforms the ranking of only using image features. The 'Eyes' plot in all the figures demonstrates how the ranking (only using eye-movements) would perform if eye-features were indeed available a-priori for new images.

alternative, i, a feature vector $\mathbf{z}_i(t) \in \mathbb{R}^d$ is given. We have a linear model, $E[\mathbf{x}_i] = \mathbf{f}\mathbf{z}_i$ where $\mathbf{f}$ is an unknown fixed weight vector on the features and $\mathbf{z}_i \in \mathbb{R}^{n\times 1}$ is the feature vector. $\mathbf{x}_i$ represents the user feedback. We calculate UC-bound for the expected feedback, $\mathbf{f}\mathbf{z}_i$, to balance the exploration-exploitation. Let $Z_t = \begin{bmatrix}\mathbf{z}(1) & \mathbf{z}(2) & \dots & \mathbf{z}(t-1)\end{bmatrix} \in \mathbb{R}^{n\times(t-1)}$, and $X_t = \begin{bmatrix}\mathbf{x}(1) & \mathbf{x}(2) & \dots & \mathbf{x}(t-1)\end{bmatrix} \in \mathbb{R}^{1\times(t-1)}$, hence, the UC-bound for $\mathbf{f}\mathbf{z}_i$ is as follow,

$$\mathrm{UCB}_i(t) = X_t Z_t'(Z_t Z_t')^{-1}\mathbf{z}_i + ||\mathbf{a}_i||C$$

where C is a free parameter to control the importance of the variance compared to the expected value in the upper confidence bound and $\mathbf{a}_i = \mathbf{z}_i'(Z_t Z_t')^{-1} Z_t$. To avoid overfitting, $(Z_t Z_t' + \mu I)^{-1}$ is used for linear regression with regularization instead of $(Z_t Z_t')^{-1}$.

6 Conclusions

The objective of Task 5.1 is to develop algorithms which use implicit relevance feedback namely eye movements from the user. All the experiments were conducted on "Transport Rank Five" Dataset which were previously collected in Task 8.3.

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 [10, 2] 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.

In section 3 we have adapted and improved Ranking SVM through a perceptron-style algorithm for on-line 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 experiments 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 image 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.

Although, still, the proposed approach in section 3 requires eye features for the test images which would not be practical in a real system. In section 4 we presented a novel search strategy for combining eye movements and image features with a tensor product kernel used in a Ranking SVM framework. We continued to show that the joint learnt semantic space of eye and image features can be efficiently decomposed into its independent sources allowing us to further test or train only using images.

Experience with this task showed that it actually took 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 rank the images internally before clicking on the radio buttons. In some cases mistakes were made and the user had to return and re-rank or add missing ranks, so post-processing of this data will need to be done with care.

Acknowledgements

We wish to thank Dr. Jorma Laaksonen of Teknillinen korkeakoulu for his valuable contributions in the writing of this report and for comments on its draft versions.

References

[1] Rajeev Agrawal. Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

[2] 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.

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2003.

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

[5] Georg Buscher, Andreas Dengel, and Ludger 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.

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

[7] 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.

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

[9] David R. Hardoon and Kitsuchart Pasupa. Image ranking with implicit feedback from eye movements. In *Proceedings of the 6th Biennial Symposium on Eye Tracking Research & Applications (ETRA'2010)*, 2010.

[10] David R. Hardoon, John Shawe-Taylor, Antti Ajanki, Kai Puolamäki, and Samuel Kaski. Information retrieval by inferring implicit queries from eye movements. In *AISTATS '07: Proceeding of International Conference on Artificial Intelligence and Statistics*, 2007.

[11] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In Smola, Bartlett, Schoelkopf, and Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 2000.

[12] Kalervo Järvelin and Jaana 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.

[13] Thorsten 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.

[14] Arto Klami, Craig Saunders, Teófilo E. de Campos, and Samuel 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.

[15] Jorma Laaksonen, Markus Koskela, Sami Laakso, and Erkki Oja. PicSOM–content-based image retrieval with self-organizing maps. *Pattern Recognition Letter*, 21(13-14):1199–1207, 2000.

[16] Shawn Martin, Diana Roe, and Jean-Loup Faulon. Predicting protein-protein interactions using signature products. *Bioinformatics*, 21:218–226, 2005.

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

[18] 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.

[19] Kitsuchart Pasupa, Sandor Szedmak, and David R. Hardoon. Learning to rank images from eye movements. In *Proceedings of the Neural Information Processing Systems (NIPS'2009) Workshop on Advances in Ranking*, pages 37–42, 2009. http://web.mit.edu/shivani/www/Ranking-NIPS-09/Proceedings/proceedings-nips09workshop-ranking.pdf.

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

[21] Kai Puolamäki, Jarkko Salojärvi, Eerika Savia, Jaana Simola, and Samuel 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, NY, USA, 2005.

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

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

[24] Jarkko Salojärvi, Kai Puolamäki, Jaana Simola, Lauri Kovanen, Ilpo Kojo, and Samuel Kaski. Inferring relevance from eye movements: Feature extraction. Technical Report A82, Computer and Information Science, Helsinki University of Technology, 2005.

[25] Craig Saunders and Arto Klami. Database of eye-movement recordings. Technical Report Deliverable D8.3, PinView, European Comunity project FP7-216529, 2008. http://www.pinview.eu.

[26] Sidney Siegel and N. John Castellan Jr. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill, Singapore, 1988.

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