

## Deliverable D2.3

### Report on potential improvements obtainable by data fusion

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.



## Identification sheet

<b>Project ref. no.</b>	<b>FP7-216529</b>
<b>Project acronym</b>	PinView
<b>Status and version</b>	Final, Revision: 1.0
<b>Contractual date of delivery</b>	30.6.2010
<b>Actual date of delivery</b>	31.8.2010
<b>Deliverable number</b>	D2.3
<b>Deliverable title</b>	Report on potential improvements obtainable by data fusion
<b>Nature</b>	report
<b>Dissemination level</b>	PU – Public
<b>WP contributing to the deliverable</b>	WP2 Learning relevance feedback from eye tracking
<b>Task contributing to the deliverable</b>	Task 2.3 Data fusion
<b>WP responsible</b>	Aalto-korkeakoulusaatio
<b>Task responsible</b>	Aalto-korkeakoulusaatio
<b>Editor</b>	Jussi Kujala <jussi.kujala@iki.fi>
<b>Editor address</b>	PO BOX 15400, FI-00076 AALTO, Finland
<b>Authors in alphabetical order</b>	Samuel Kaski, Arto Klami, Jussi Kujala, Shiau Hong Lim, Chiwei Wang
<b>EC Project Officer</b>	Pierre-Paul Sondag
<b>Keywords</b>	data fusion, image retrieval, implicit relevance feedback
<b>Abstract</b>	<p>This report studies different aspects of relevance prediction of images from eye movement feedback in content-based image retrieval sessions. The first result is that eye movement relevance predictors generalize well when different factors, such as the user, in the CBIR setting change. The second part of the work studies how image content not related to the CBIR task at hand, like an attractive face, influences the relevance prediction. The result is that there is an image specific component in relevance predictions but in our experiments combining it with eye movement feedback does not improve prediction accuracy. Next this work demonstrates how to construct latent features of images by factorizing an incomplete matrix of eye movement relevance predictions. The method works well in a small-scale experiment that has images with two topics. However, theory predicts that the number of required relevance samples is very large for more complicated objects. Finally, this report studies how to augment Hough transform with semantic tags in order to better locate interesting objects in images.</p>

**List of annexes:** none

## Contents

<b>1</b>	<b>Overview</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
<b>3</b>	<b>Predicting image relevance from online gaze patterns</b>	<b>5</b>
3.1	Relevance prediction . . . . .	6
3.2	Experimental setup . . . . .	6
3.3	Evaluation . . . . .	7
3.4	Experiment results . . . . .	8
<b>4</b>	<b>Effect of cognitive distractors in images on eye movements during CBIR</b>	<b>9</b>
4.1	Experimental setup . . . . .	9
4.2	Evaluation . . . . .	10
4.3	Experiment results and discussion . . . . .	11
<b>5</b>	<b>Deriving latent image features from gaze relevance information</b>	<b>12</b>
5.1	Previous work . . . . .	13
5.2	Algorithm details . . . . .	14
5.3	Theory: How much (eye movement) data is enough to infer useful features? .	15
5.3.1	The simplest scenario: binary topics and known queries . . . . .	16
5.3.2	Generalization to multiple topics . . . . .	17
5.4	Experiments . . . . .	19
<b>6</b>	<b>Enhancing visual object detection with semantic tags</b>	<b>21</b>
6.1	Semantic Tags . . . . .	21
6.2	Hough Transform as Inference . . . . .	22
6.3	Experiments . . . . .	23
<b>7</b>	<b>Conclusion</b>	<b>23</b>
<b>A</b>	<b>Approximation to expected squared error when inferring topics</b>	<b>26</b>

# 1 Overview

This deliverable constitutes the output of Task 2.3 *Data fusion* of the *Personal Information Navigator Adapting Through Viewing*, PinView, project, funded by the European Community's Seventh Framework Programme under Grant Agreement number 216529. The delivery date of this deliverable was changed from M30 to M32 to allow larger and more time-consuming experiments than originally planned.

The aim of WP2 is to study relevance prediction of images from eye movement data in content-based image retrieval settings. This work studies eye movement relevance predictions from several perspectives.

First, the generalization capability of eye movement predictors to different environments is studied. The result is that the relevance predictors generalize well.

The second part of the work studies how image content not related to the CBIR task at hand, like an attractive face, influences the relevance prediction. The result is that there is a statistically significant image specific component in relevance predictions but combining it with eye movement feedback does not improve prediction accuracy in our experiments.

The third line of work combines relevance feedback collected during several search sessions to construct latent features of images. The motivation is that semantic information of images is defined by how humans perceive the images and hence learning directly from humans could be advantageous. A small-scale experiment with images with two topics shows that it is possible to learn the features with a simple matrix factorization model to the extent possible given by noise in relevance predictions. Theoretical model predicts that the number of gaze samples needed to learn latent information in complicated images is quite large. Hence, an approach that uses visual content of images to guide the learning might be necessary.

The final part of the work considers augmenting Hough transform with semantic tags. The idea is that objects compose of sub-objects, for example cars have tires. These sub-objects could be used to better predict the location of the main object. The results show that, with semantic tagging, less training examples are needed to achieve the same level of performance.

This report describes contributions of two project partners: Aalto University School of Science and Technology (formerly Helsinki University of Technology) and University of Leoben. University College London did also work for this task, but it is not yet mature enough to be included.

## 2 Introduction

This work studies how to improve interactive content-based image retrieval (CBIR) systems using implicit relevance feedback inferred from eye movements. Previously, Task 2.1 of the PinView project studied how to predict relevance of a thumb-nail image on an image collage and Task 2.2 studied inferring relevance of parts of an image from eye movements and visual content. This work continues the previous work with a focus on combining information from different sources. The following four sections will cover in detail the different aspects of the work. We give a short introduction to them here and each section has a more detailed introduction.

Section 3 studies how well a pre-trained relevance predictor performs in an environment that is different from the one where the predictor was trained. Previous work evaluated relevance predictors on test data from the same experiment as the training data. The question is whether or not do these results generalize to different tasks, image databases, image layouts, and users in CBIR experiments?

Section 4 studies whether combining eye movement data with image specific data improves relevance prediction. The motivation is that images might contain distractors such as attractive faces that could bias relevance predictions on images that contain them. This line of work first looks for evidence that such bias exists using statistical significance tests. The accuracy of relevance prediction is the ultimate goal, hence the influence of the bias on the accuracy is also estimated.

Section 5 focuses on how to combine relevance predictions from several search sessions in order to find latent features of the images. This enables transferring relevance feedback from earlier search sessions to the current search session. The motivation is that semantic information in images is defined by how humans perceive the images, hence learning directly from humans could be an important route to this information. The feature construction is done by performing a factorization of an incomplete matrix that has queries and images as dimensions and relevance values as items. We also estimate how much data would be needed to accurately infer topics of images given the noise level of eye movements.

Finally, Section 6 studies how to augment Hough transform with semantic tags. The idea is that objects compose of sub-objects, for example cars have tires. The sub-objects could aid in predicting the location of the main object.

## 3 Predicting image relevance from online gaze patterns

This section studies performance of an image relevance predictor in a CBIR task from gaze-patterns. This work was started in the previous Tasks 2.1 and 2.2 of the PinView project. The difference of the present work to the task 2.1 is that it considered relevance predictor that was trained and tested on data from the same experiment. Here we study performance of a pre-trained predictor in various online tasks. The Task 2.2, on the other hand, concentrated on taking advantage of sub-image gaze patterns, i.e., predicting relevance of parts of an image, which is not discussed here.

The motivation of this work is to understand how much information eye movements contain under various conditions, and especially how well a pre-trained predictor performs in online experiments. The factors that affect eye movement data in CBIR tasks include at least:

1. The task the user performs.
2. User-specific factors.
3. The layout (size and position) and content of the shown images.

A change in these factors could change the eye movements and hence the performance of a relevance predictor that depends on eye movements. To study the impact of these effects, a relevance predictor is first trained on gaze data from an unrelated task and its performance is then studied in two different online experiments. The following sections present details on the relevance prediction and experiments.

### 3.1 Relevance prediction

The training of relevance predictor slightly differs from the one in Task 2.1, because we found that the following procedure has a slightly better empirical performance. This study uses a smaller set of features and a regularized logistic regression predictor instead of linear discriminant analysis (LDA). Each time an image is showed to the user the system collects 19 features computed from both raw eye movement samples and fixations. These 19 features are the subset of the 33 features described in [18] that have the highest correlation with ground-truth relevance values. The features include, e.g., the logarithm of the total time the image was looked at and regressions to already seen images. More precisely, the selected features are features  $\{1, 2, 3, 7, 8, 9, 14, 15, 16, 17, 18, 19, 20, 21, 22, 30, 31, 32, 33\}$  described in Table 1 of [18]. If the image was not seen by the user then the features are zero.

The relevance of an image is predicted from the features using a logistic regression model trained on a data set collected during online CBIR search sessions. Hence, for an eye movement feature vector  $e$  corresponding to an image seen by a user in some task the system computes the relevance score  $\text{rel}(e|w)$  as:

$$\text{rel}(e|w, b) = \frac{1}{1 + \exp(-w \cdot e + b)},$$

where  $w$  and  $b$  are learned weight vector and bias term. Eye-movement features that were zero because the user did not see the corresponding image were removed from the data set before training the predictor. The parameters  $w$  and  $b$  were learned with regularized logistic regression on features which were each standardized to zero mean and unit variance. Thus, the loss function to minimize was the sum of logarithms of inverse probabilities given by the above  $\text{rel}(e|w, b)$  at each observed data point plus the norm  $\sqrt{\|w\|^2 + b^2}$ . The constant of regularization was selected with 5-fold cross-validation from the set  $\{0, 0.01, 0.1, 1, 10, 100, 1000\}$ .

Note that the relevance prediction ignores the following factors: size of images, gaps between images (which affects for example how many images are directly looked at by the user), position of images (for example the location where the user looked before current images were shown probably affects eye movement features), whether eye movements are recorded from the left, right, or both eyes, how long it took to come into conclusion, which was the last image seen, and regressions between images just before user clicked an image.

### 3.2 Experimental setup

The experiments consist of two separate online experiments that resemble real information retrieval tasks. The most significant limitation in the experiment design is how to obtain ground-truth relevances of images in the tasks. In these experiments the relevance is derived from manually collected class labels. Both experiments contain tasks with heterogeneous image data sets and more uniform collections that mimic collections that have been limited with a tag-based search. Also, the collages contained a varying number of relevant images (images with the same category label as the search target) depending on the task. This has a significant effect on eye movements; if the number of relevant images is low, then eye movements distinguish non-relevant and relevant images very well. The following paragraphs give more detailed information on each of the experiments.

**Experiment A** has six users. There are 4 different tasks which each consist of 3 similar sub-tasks. In tasks 1 and 2 the dataset is full VOC2008 dataset (9963 images). In tasks 3 and 4 the dataset is limited to single cats (604) and single dogs (721) of VOC2008 dataset. The task of the user is to find as many images as possible that match either category aeroplane or motorbike in the first two tasks, and either a cat or dog in the tasks 3 and 4. Images are retrieved either randomly or with the PinView CBIR system [2] which receives user feedback. Hence, image collages include ones with small and large number of relevant images (because of the used image dataset and the use of CBIR algorithm). Compared to the next experiment B the size of images is smaller and gaps between images larger (120 pixel border length, gap is approximately 100 pixels on 1280x1024 pixel screen). The eye movement relevance predictor for this experiment is trained on data from an earlier experiment which is similar to the first two tasks of this experiment where the number of relevant images on each collage is high.

**Experiment B** has 10 users, of which only one is in common with the experiment A. Each user performs four different tasks which each consist of 3 similar sub-tasks. In all of the tasks the images come from a subset of images sampled from ImageNet [11] which is an image database containing URLs to images available on Internet together with semantic labels (synsets of WordNet) and a hierarchy between them. Hence, there is a large variety of images. In the four tasks the system instructs the user to:

1. Find images of a particular sport: the targets of the sub-tasks are ice hockey, gymnastics, and soccer.
2. Find images of aeroplanes in each of the sub-tasks.
3. Find flowers in each of the sub-tasks.
4. Find a particular mammal: the targets of sub-tasks are deer and twice cheetah.

In task 1 the image dataset contains 1006 images sampled from the sports sub-category of ImageNet, which has 89 ice hockey, 92 gymnastics, and 88 soccer images. In tasks 2 and 3 the image dataset is the same: 900 uniformly sampled images that are not flowers or aeroplanes, and additionally 150 images from both aeroplanes and flower category. In task 4 the image dataset contains 105 images sampled from deer category, 99 images sampled from cheetah category, and 612 images sampled from mammal category with the restriction that they are not deers or cheetahs. The fraction of relevant images in data sets is relatively small, but some collages contain a high number of relevant images due to the fact that images are retrieved by the PinView system [2]. The size of images is larger and gap between images is smaller than in experiment A (180 pixel border length, gap between images is about 60 pixels). The eye movement relevance predictor for this experiment is trained on data from the experiment A.

### 3.3 Evaluation

In evaluation each query is analysed separately. Images shown during a query are ranked according to relevance predictions from eye movements and the quality of the ranking is measured with mean average precision (MAP) computed from ground-truth relevance labels. An image is relevant in a query if it has the same category as the target image of the task. To compute MAP, the images are sorted according to the predicted relevance, each position with a relevant image is given a score  $(1 - i/n)$ , where  $i$  is the position (highest rank has position 0) and  $n$  is the number of retrieved images, and the scores are normalized so that the perfect ranking has a precision 1.0.

Table 1: A pre-trained eye movement relevance predictor performance in experiment A. The tuples contain MAP of relevance predictions and an expected MAP of a random permutation of the retrieved images. Rows correspond to tasks and the last row is an average over tasks. The first column gives the target category of each task and the the remaining columns correspond to users and the average over users.

query/user	user1	user2	user3	user4	user5	user6	average
motorbike	(0.67,0.02)	(0.24,0.05)	(0.40,0.03)	(0.47,0.04)	(0.48,0.05)	(0.75,0.06)	(0.50,0.04)
motorbike	(0.67,0.07)	(0.18,0.06)	(0.16,0.07)	(0.33,0.02)	(0.00,0.01)	(0.67,0.02)	(0.33,0.04)
motorbike	(0.33,0.03)	(0.14,0.06)	(0.29,0.05)	(0.67,0.02)	(0.67,0.02)	(0.00,0.01)	(0.35,0.03)
aeroplane	(1.00,0.01)	(0.80,0.03)	(0.20,0.03)	(0.38,0.05)	(0.47,0.24)	(0.62,0.05)	(0.58,0.07)
aeroplane	(0.00,0.01)	(0.14,0.05)	(0.59,0.45)	(0.50,0.03)	(0.44,0.42)	(0.53,0.41)	(0.37,0.23)
aeroplane	(0.28,0.07)	(0.00,0.02)	(0.00,0.02)	(0.58,0.07)	(0.05,0.05)	(0.50,0.03)	(0.23,0.04)
dog	(0.68,0.55)	(0.68,0.53)	(0.69,0.47)	(0.67,0.63)	(0.53,0.46)	(0.72,0.42)	(0.66,0.51)
dog	(0.56,0.49)	(0.63,0.51)	(0.72,0.65)	(0.70,0.44)	(0.61,0.52)	(0.75,0.61)	(0.66,0.54)
dog	(0.68,0.60)	(0.71,0.57)	(0.64,0.50)	(0.65,0.48)	(0.68,0.56)	(0.73,0.61)	(0.68,0.55)
cat	(0.66,0.55)	(0.65,0.48)	(0.72,0.52)	(0.67,0.56)	(0.59,0.47)	(0.63,0.48)	(0.65,0.51)
cat	(0.64,0.50)	(0.72,0.54)	(0.57,0.52)	(0.30,0.46)	(0.66,0.53)	(0.52,0.44)	(0.57,0.50)
cat	(0.59,0.46)	(0.70,0.48)	(0.63,0.45)	(0.61,0.59)	(0.66,0.59)	(0.60,0.49)	(0.63,0.51)
average	(0.56,0.28)	(0.47,0.28)	(0.47,0.31)	(0.55,0.28)	(0.49,0.33)	(0.58,0.30)	(0.52,0.30)

Table 2: A pre-trained eye movement relevance predictor performance in experiment B. The layout of the table is the same as in Table 1.

query/user	user1	user2	user3	user4	user5	user6	user7	user8	user9	user10	average
cheetah	(0.39,0.22)	(0.80,0.49)	(0.72,0.38)	(0.61,0.47)	(0.58,0.39)	(0.57,0.25)	(0.01,0.03)	(0.62,0.42)	(0.53,0.38)	(0.50,0.03)	(0.53,0.30)
cheetah	(0.41,0.29)	(0.52,0.28)	(0.33,0.15)	(0.50,0.28)	(0.44,0.33)	(0.16,0.12)	(0.54,0.42)	(0.38,0.38)	(0.36,0.14)	(0.62,0.42)	(0.43,0.28)
deer	(0.49,0.15)	(0.39,0.14)	(0.43,0.16)	(0.56,0.17)	(0.61,0.20)	(0.26,0.15)	(0.57,0.19)	(0.30,0.15)	(0.57,0.21)	(0.47,0.17)	(0.46,0.17)
aeroplane	(0.43,0.11)	(0.70,0.30)	(0.62,0.19)	(0.40,0.15)	(0.61,0.28)	(0.62,0.20)	(0.53,0.26)	(0.52,0.26)	(0.42,0.19)	(0.55,0.30)	(0.54,0.22)
aeroplane	(0.45,0.27)	(0.25,0.08)	(0.47,0.22)	(0.41,0.12)	(0.46,0.23)	(0.28,0.12)	(0.33,0.07)	(0.40,0.07)	(0.48,0.13)	(0.20,0.04)	(0.37,0.14)
aeroplane	(0.47,0.08)	(0.61,0.27)	(0.01,0.04)	(0.42,0.08)	(0.56,0.28)	(0.36,0.14)	(0.59,0.20)	(0.62,0.30)	(0.29,0.06)	(0.49,0.23)	(0.44,0.17)
flower	(0.39,0.22)	(0.45,0.23)	(0.42,0.11)	(0.42,0.19)	(0.43,0.26)	(0.33,0.08)	(0.53,0.28)	(0.40,0.21)	(0.27,0.09)	(0.60,0.28)	(0.42,0.19)
flower	(0.37,0.28)	(0.39,0.06)	(0.27,0.26)	(0.47,0.28)	(0.48,0.14)	(0.48,0.30)	(0.42,0.17)	(0.38,0.07)	(0.51,0.30)	(0.52,0.18)	(0.43,0.21)
flower	(0.35,0.31)	(0.48,0.22)	(0.41,0.27)	(0.57,0.28)	(0.44,0.17)	(0.49,0.29)	(0.40,0.07)	(0.47,0.19)	(0.49,0.30)	(0.20,0.07)	(0.43,0.22)
gymnastic	(0.39,0.12)	(0.43,0.06)	(0.30,0.03)	(0.42,0.15)	(0.13,0.07)	(0.01,0.02)	(0.01,0.01)	(0.01,0.05)	(0.86,0.05)	(0.20,0.03)	(0.27,0.06)
soccer	(0.59,0.22)	(0.54,0.14)	(0.47,0.30)	(0.26,0.10)	(0.40,0.17)	(0.43,0.27)	(0.32,0.18)	(0.05,0.11)	(0.19,0.16)	(0.41,0.26)	(0.37,0.19)
ice hockey	(0.57,0.40)	(0.42,0.24)	(0.58,0.37)	(0.43,0.33)	(0.59,0.14)	(0.47,0.38)	(0.44,0.12)	(0.30,0.03)	(0.57,0.36)	(0.49,0.15)	(0.49,0.25)
average	(0.44,0.22)	(0.50,0.21)	(0.42,0.21)	(0.45,0.22)	(0.48,0.22)	(0.37,0.19)	(0.39,0.17)	(0.37,0.19)	(0.46,0.20)	(0.44,0.18)	(0.43,0.20)

### 3.4 Experiment results

Tables 1 and 2 present the results of the experiments. Each item is a tuple ( $MAP_{eye}$ ,  $MAP_{rand}$ ), where  $MAP_{eye}$  is MAP of the relevance predictor and  $MAP_{rand}$  is an expected MAP of a random permutation on the retrieved images.

Results in the tables are similar and the numbers differ mainly because Table 1 has tasks where either the number of retrieved images is very low or very large. The relevance predictor is almost always clearly better than the random baseline. The results indicate that the eye movements distinguish relevant items better when their number is low. For example, in several queries where the system showed only one relevant image the relevance predictor has the top predicted relevance (MAP 1.00 vs MAP 0.01 of the random baseline). The average quality of relevance predictions does not differ much between users.

Hence, these results answer positively to the question given in introduction on whether eye movement relevance predictors generalize to new environments. More concretely, for the PinView project the results imply that using existing predictors in online CBIR tasks will provide sufficient performance.

## 4 Effect of cognitive distractors in images on eye movements during CBIR

This section studies whether eye movement relevance prediction depends on the seen image in a way that does not depend on the task at hand. For example, a bright region with high contrast might catch the attention of the user even if it has nothing to do with the search task. In the following image content that distracts eye movements will be referred to as *cognitive distractors*. The relevance predictors considered in this work package so far have depended only on the gaze patterns on the image under consideration. The ultimate goal of this line of work is to improve eye movement relevance prediction by taking into account image specific factors.

However, existing literature in eye movement and vision research report that a viewer with clear task is able to ignore low-level saliency almost completely, indicating that the gaze control is primarily top-down instead of bottom-up [17, 13]. Nevertheless, the results of these studies do not necessarily apply to information retrieval tasks studied in this work. The experimental setup in the previous studies was different from an information retrieval setting because of the following aspects:

- The images were presented one at a time.
- The tasks were for example counting people in images or inserting bull's-eye target to search for.
- Saliency was defined in a particular way, in [17] as the natural “intensity, contrast, and edge density at fixated scene regions”, and in [13] salient regions were as ones where the contrast had been increased artificially.
- Images were of a particular group: out door scenes, and in [13] the images also contain few or no man-made objects.

Earlier work in the PinView project has given evidence that the image content might affect gaze patterns even if the content is not related to the task at hand. There are several reasons why the eye movements might be affected by cognitive distractors despite the earlier results in the literature. For example, non-visual properties of images, like semantic information that there is an attractive face of a human, differ from saliency. Also, properties such as the size of the main object or the complexity of the image content might increase the time required to process it.

Previously in this work package the Task 2.2 considered predicting visual saliency of parts of new images from visual content if the task is *same*. This work focuses on whether there is a bias (visual saliency) specific to the whole image in relevance predictions of images that is constant over several tasks and users. The following sections describe data collection procedure, how to evaluate the data, and results of the analysis.

### 4.1 Experimental setup

The cognitive distractor effect can be quantified only with data from experiments. Ideally the test subjects would interact for a long time with a CBIR system with a large image database. Unfortunately, we are not able to do large-scale experiments because user experiments are time consuming. The limited experiments should fulfill the following requirements to the best possible extent:

1. Some of the shown images contain cognitive distractors, assuming that they exist.

2. The evaluation can infer if relevance predictions are biased. Hence, each image must be seen several times during different search sessions in order to average away the noise in the predictions. However, a user must not see the same image too many times, because then the user's memory could bias the results.
3. Task dependent differences in relevance prediction must be removed in the evaluation. Thus the evaluation should be able to infer whether an image is relevant or not in a search query.

We do not know for certain what cognitive distractors look like, so the experiments have as large variety of images as possible. To this end the image database is formed by sampling from the ImageNet [11]. Instead of using a CBIR system to fetch the images we sample images uniformly from the image dataset to avoid biasing the image collection to too similar images. The following paragraphs document the two online experiments that produce data from which we study the cognitive distractor effect.

**Description of the experiment A.** Four test subjects from the staff of Aalto University School of Science and Technology (not associated with eye movement research) performed the experiment. The image dataset consisted of 449 images from ImageNet and before each task 50 additional images were added to the image dataset from the relevant category (these were very rarely seen as non-relevant). Each user performed 12 tasks. The task descriptions instruct to look a particular category together with an example image of the category. The categories to look for were flower (3 copies), aeroplane (3 copies), deer, cheetah (2 copies), ice hockey, gymnastics, and soccer. During each task the system showed 8 collages of 15 random images. Hence, the system showed a total of 5760 images, so each image was showed over 10 times, or twice per each user.

**Description of the experiment B.** The experiment was similar to the experiment A except in the following two aspects. First, the search targets were slightly different: there was two copies of each of deer, cheetah, ice hockey, skating, flower (not sunflower), and sunflower. Second, the image database contained only items that were relevant in one of the tasks: there were 80 images from ImageNet of each category. The test subjects were different from the test subjects in the experiment A.

## 4.2 Evaluation

The evaluation of the outcome of the experiment has two related goals:

1. To identify if there is an image specific component that affects the relevance predictions and to quantify it.
2. To analyse if the image specific component is helpful in predicting relevance of an image during a search task.

Let us now go through one way to formalize the first goal. First, discard all data samples in which a user saw a relevant image. This should remove the component in relevance predictions which depends on the task. Second, group relevance predictions corresponding to the same image. This creates an empirical distribution of relevance values for each image. The final step is to use ANOVA on these groups.

ANOVA is a statistical test that lets us set a null hypothesis that the relevance predictions come from the same distribution. In it we group the samples in several different ways and ask whether the relevance values depend on the groups. For example, samples could be grouped by the image they belong to, by the user that saw the image, or by both. ANOVA assumes a normal distribution on data samples and hence evaluation also performs a normality test on the data samples.

Table 3: High p-values for the user-specific predictor suggest that the null hypothesis (relevance predictions from it have a normal distribution) can not be rejected. Low p-values for the pre-trained predictor suggest that its output does not fully conform to the normal distribution.

	p-value for pre-trained predictor	p-value for user-specific predictor
experiment A	0.00016	0.12
experiment B	0.00550	0.14

Table 4: P-values from ANOVA on the null hypothesis that a given source does not affect the population mean. Low p-values for the image source suggest that relevance predictions have an image-specific component. The effect is more clear with the user-specific predictor (p-value user). Only in experiment B the p-value for pre-trained predictor (p-value pre) is high.

source	experiment A			experiment B		
	df	p-value pre	p-value user	df	p-value pre	p-value user
images	110	0.0256	<0.0001	112	0.9076	0.039
users	3	0.0144	<0.0001	3	<0.0001	<0.0001

Samples on images which were seen less than 10 times were discarded in the analysis with ANOVA. The evaluation is performed on relevance values from two different predictors. The *pre-trained predictor* is trained on data from experiment A documented in Section 3.2 using the same procedure. The *user-specific predictor* is trained for each user separately on data from the experiment under study and hence it is less noisy.

The second goal is what we are ultimately after: improving relevance prediction. To this end, the evaluation performs several iterations of the following procedure. First it discards all samples on images that were looked at less than three times by some user when the image was non-relevant in the task at hand. Then it divides the data on each image to three sets that we call I, II, and III. On the set I the evaluation trains a *user-specific predictor* called A (using procedure similar to one described in Section 3.1). The set II consists only of non-relevant samples and it is used to compute the bias in the relevance prediction. More precisely, the bias is the mean predicted relevance of non-relevant samples. The relevance predictor is either the user-specific predictor A or a *pre-trained predictor* which is trained on data from a separate experiment and for which user-specific mean is removed from predictions. A new relevance predictor called B is again trained on the set I, but this time the image biases are appended to the eye movement features. Finally, the performance of the predictors A and B are compared on the test set III. The performance measure is area under the curve (AUC) which measures how frequently the predictor correctly ranks a random positive and negative example. This approach can tell whether it is possible to improve the relevance prediction on an average image, but it can not tell if the bias is significant for a small number of images.

### 4.3 Experiment results and discussion

Table 3 reports p-values of a normality test (one-sample Kolmogorov-Smirnov test `kstest` in matlab) for the relevance values. The p-values suggest that the normality assumption can not be rejected for the user-specific predictor. Table 4 gives the results of the ANOVA tests. From it we see that the p-values of ANOVA are low for the user-specific predictor. Hence,

Table 5: AUC of two relevance predictors: one that does not use an image-specific bias in relevance prediction and one that does.

	without bias	with bias
AUC	76.23	76.28

the experiment suggests that there is a statistically significant image-specific bias in relevance predictions.

Table 5 reports the AUC measures for two relevance predictors: one that uses only eye movement features, and another one that uses both eye movement features and an image-specific bias. Section 4.2 described this evaluation in a more detail. The effect of the image-specific bias is clearly minimal. Note that it is not possible to perform similar analysis for the experiment A, because in that experiment images are seen mostly as relevant or non-relevant over all tasks and users. Hence, the image-specific bias can not be computed for images that were seen only as relevant, and so it is not possible to train a relevance predictor that takes the bias into account.

On the other hand the image data set in experiment B does not contain images sampled uniformly from ImageNet. Hence, it contains less variability than experiment A. This difference could result the experiment B having less potential cognitive distractors which might also explain why p-values for the image source of ANOVA are generally smaller (better) in the experiment A.

The results depend on factors such what the tasks are, what kind of images are shown to users, sample sizes (how many images and how many times images were seen), how ground-truth relevance is defined, how common cognitive distractors are, and whether cognitive distractors are also user-specific. Hence, we can not generalize these results to every possible scenario. However, these results suggest that, although cognitive distractors appear to exist, we can safely ignore them in an average CBIR task because their impact is small.

## 5 Deriving latent image features from gaze relevance information

This section considers a specific way to to improve information retrieval performance of a CBIR system by combining relevance predictions from eye movements collected during several search sessions. Let us first think possible ways in which we could exploit the relevance predictions outside the search session they were collected. Figure 1 presents a very general setting in which a group of humans interact with objects such as text documents, images, physical places, and the system records some aspect of this interaction. For example, this work focuses on a CBIR system that records eye movements of the users. We could use the observed interactions to learn more about different concepts in Figure 1:

1. users,
2. objects, or
3. interactions between users and objects. For example learning to predict relevance from eye movements.

These categories are not always fully separated, because information about the users or objects could be used to guide, e.g., relevance prediction.

This work will concentrate on learning about the objects, even though all of the above concepts are important in a CBIR setting. Concretely this translates to constructing features

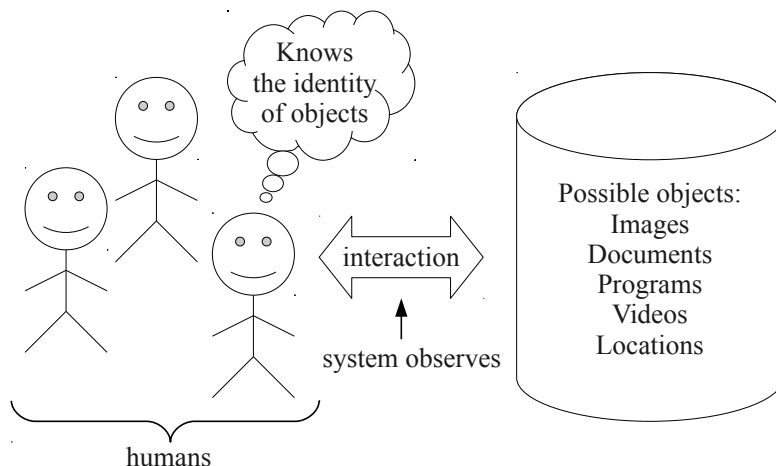


Figure 1: Humans interacting with objects. The interaction is observed by the system.

for images that contain as much useful information as possible. The motivation is that features that depend on visual image content of images are limited in the sense that they capture less semantic meaning than, e.g., bag-of-word representations for text documents. This *semantic gap* between visual information and semantic meaning is discussed in surveys such as in [10]. Humans define the semantic information of image content and hence, a method that learns directly from humans could provide an important source of information about images.

The following sections will review relevant previous work and describe in detail one possible approach how to infer semantic features on images from relevance information derived from gaze patterns. They will also discuss how much data is required for efficient learning.

## 5.1 Previous work

Feature construction from implicit relevance feedback collected during information retrieval sessions has recently received significant interest due to importance of text search and abundance of click-through data. Typically interest has focused on learning more about documents and queries. For example, Bai et al. [3] suggest mapping the bag of words presentations of the text documents and the queries to a space with smaller dimension in which it is easier to reason how relevant a document is in a query. More formally, for a document  $d$  and a query  $q$  they infer linear (they also considered higher order polynomial) mappings  $V$  and  $U$  such that

$$(Vq) \cdot (Ud)$$

is the estimated relevance of the document  $d$  in the query  $q$  (which are both bag of word feature vectors). They used implicit rank information contained in click-through data to infer the mappings  $V$  and  $U$ . The click-through data is collected in information retrieval sessions where the user first makes a textual query  $q$ , then an information retrieval engine shows an ordered set of documents, and finally the user may click one of them. It is possible to obtain enormous amount of information in this setting because the user is not required to do anything that he would not do during a real information retrieval task. The clicked document  $d_c$  should on average be more relevant than non-clicked documents  $d_n$ , which formally means that  $(Vq) \cdot (Ud_c) > (Vq) \cdot (Ud_n)$ . Bai et al. formulate a maximum margin loss function on the violation of that expression. They then minimize the loss function with an online stochastic gradient algorithm. The dimension of the output space was at most 200 in their experiments.

A similar line of work is studied by Grangier and Bengio [16] who study a CBIR setting where a user inputs a textual query and a CBIR engine responds with a list of images.

The differences to the work of Bai et al. [3] are that the image vectors (visual features) do not lie in the same space as the queries (bag-of-words text), they do not consider a low-rank approximation for the query, and they use an online passive aggressive algorithm [8] to optimize the loss function.

Chechik et al.[7] use similar tools to learn a similarity metric between images: they use an online algorithm to infer a matrix  $W$  that specifies a similarity  $s_W(p_i, p_j) = p_i^T W p_j$  between arbitrary images  $p_i$  and  $p_j$ . They learn the similarity using triplets  $(p, p_+, p_-)$  for which the pair  $(p, p_+)$  is more similar than the other pair  $(p, p_-)$ . For example, a triplet can be constructed using known category information on images: it is far more likely that a pair sharing a category label are more similar than if the other image is sampled uniformly at random from an image database.

Our work differs from the previous work in following ways:

- The relevance feedback is from eye movements which is even noisier than clicks. However, potentially eye movement data is even more abundant than click-through data.
- We do not assume to know the query (at least exactly) which will be inferred during the CBIR session. The setting is closer to browsing, because in image retrieval it is less feasible to assume that the user knows what words to use in order to find the image that interests him or her. Hence, the number of potential images to retrieve can not be pruned to the same extent as in a text search. Note that modeling queries also confers some advantages, for example the query can capture user-specific variation.
- Similarly, the documents are not modeled as a function of content. Hence, the performance of the system is not limited by the visual features of the image. However, this also is a disadvantage, because the amount of data needed to learn the features could be too large.

## 5.2 Algorithm details

This section describes one approach how to infer latent image features by combining data collected during several search sessions. The input data consists of queries in which users interact with a CBIR system. Each query consists of images retrieved by the CBIR engine and relevance feedbacks from the eye movements of the user. The high-level idea is to find latent features for all images and queries such that they model the relevance feedback. More formally, let  $p$  denote features for an image that has an estimated relevance  $r$  in a query with features  $q$ . The system tries to find  $p$  and  $q$  such that  $f(q, p) \approx r$  for some function  $f$ .

We use Euclidian vectors as a representation of the features for images and queries. At first, this appears rather limited, for example because there are so many words of the English language that the amount of dimensions needed to learn them accurately appears far too large. The number of concepts seems almost unlimited, we have places like “Colosseum”, “Helsinki”, and “lake” and more abstract information like “beautiful”, “red colours”, or “smooth edges”. However, our goal is not to find these concepts as such. The setting is more flexible than, e.g., in traditional regression, because both queries and images are inferred at the same time so a particular feature of an image is not bound to some concept set a priori. Thus, we can arbitrarily choose the features for images and queries which allows us to use the Johnson-Lindenstrauss lemma. In a nutshell, it says that features with relatively small dimension can approximately model arbitrary distances between images and queries.

The Johnson-Lindenstrauss lemma roughly says that we can embed  $m$  vectors of any dimension to a subspace with dimension  $8 \log m / \epsilon^2$  where distances are distorted by a multiplicative factor of  $\epsilon$  [9]. This lemma implies that, e.g., a data set with a billion items can be embedded to a space with 1600 dimensions where the distances would be distorted by at most a factor of 0.33. This dimension is still a large number, but it is much better than, e.g.,

100,000, and it is of the similar order of magnitude as experiments done in [3] in which the dimension is 200. Also, in practise we might perform better than the Johnson-Lindenstrauss claims, perhaps because it holds for arbitrary distances.

Let us now discuss how to infer the features. The input for the system consists of  $(q, p, r)$  triplets, where  $q$  is a query,  $p$  is an image showed during query  $q$ , and  $r$  is a predicted relevance value for the image. We misuse notation and denote by  $q$  and  $p$  both the query and the image and the inferred features. The algorithm to find the latent features is an incomplete matrix factorization (MF), because literature gives evidence that MF and its variants work well and are flexible in similar domains [19]. Hence, the link between features and relevance is the dot product  $q \cdot p \approx r$ . The MF assumes normally distributed noise on the output of the dot product and also assumes that the query vectors  $q$  and image vectors  $p$  have a normal prior. Maximizing the likelihood of this model corresponds to minimization of

$$\lambda (\|P\|^2 + \|Q\|^2) + \sum_{(q,p,r) \in D} (q \cdot p - r)^2,$$

where  $\lambda$  is a parameter of the prior distribution (the regularization constant),  $P$  and  $Q$  are matrices consisting of all features for images and queries, the matrix norm is the sum of all squares of values, and the sum is over all of our observations in the input data  $D$ .

We use an online stochastic gradient descent (SGD) algorithm to minimize the MF loss function. The SGD performs the minimization by iteratively choosing a single observation and taking a gradient step along the loss function computed only at that point. SGD is a standard algorithm in large-scale optimization tasks such as in recommendation systems [19].

SGD is a simple algorithm, but it is difficult to analyse how well it converges [5, 23, 26]. To use SGD one has to choose several hyper-parameters: a regularization parameter  $\lambda$ , a step size  $\eta$ , and a number of iterations to run. The results in [5, 23] show that if the step size is chosen carefully, for convex loss functions the SGD converges to a point close to the optima. For strongly convex functions (meaning that they are not linear) the SGD finds under expectation a solution with an additive error of  $\epsilon$  in  $O(1/\epsilon)$  iterations where the constant depends on how convex the function is. For example, Pegasos SVM solver [23] uses a step size of  $1/(\lambda t)$  where  $t$  is the iteration number, projects the weight vector to a feasible space after each gradient step, and converges under expectation in approximately  $O(d/(\lambda\epsilon))$  steps where  $d$  is the number of features.

Experiments are easier to perform if the optimization method is formally guaranteed to converge and the number of hyper-parameters is small. Unfortunately, the MF loss function is not convex, which is essential in the above theoretical convergence rates. However, if all features of queries are fixed then each image corresponds to a linear regression problem, and the same holds for queries if image features are fixed. Hence, this motivates us to rely on wishful thinking that updates similar to Pegasos work at least better than ones chosen at random. Algorithm 1 gives a more formal description of the algorithm we use. In it the regularization affects only parameters that belong to the chosen query or image. The step-size is computed from the time  $t$  which is the minimum of the number of times previous SGD steps have updated the chosen query or the image. The feasible space is heuristically set to a sphere with a radius of three times the mean of absolute values of relevances. The draw-back of the algorithm is that for large data sets it requires a large number of iterations before the step size is small.

### 5.3 Theory: How much (eye movement) data is enough to infer useful features?

Before proceeding to the experiments we will look into how much data a system needs to collect in order to produce meaningful results on the identity of images. Collecting a large

---

**Algorithm 1** SGD algorithm for matrix factorization.

---

Input of the algorithm consists of a set of observation triplets  $(q, p, r)$  in which  $q$  and  $p$  are indices of a query and an image, a regularization constant  $\lambda$ , and a number of iterations  $N$ . The algorithm uses the following data:  $Q$  and  $P$  are maps from indices to current estimates of queries and images, and  $t_q$  and  $t_p$  are maps from indices to how many times a query or an image has been processed. The feasible space is chosen by a heuristic that depends on the mean absolute value of input relevances.

---

Initialize estimates in  $Q$  and  $P$  to, e.g., samples from the standard normal distribution.

Initialize counts in  $t_q$  and  $t_p$  to all ones.

**for**  $N$  times **do**

Sample uniformly at random an observation  $(q, p, r)$ .

$t_c := \min(t_q(q), t_p(p))$

$t_q[q] := t_q[q] + 1$

$t_p[p] := t_p[p] + 1$

$\eta_t := \frac{1}{\lambda t_c}$

$e := r - Q(q) \cdot P(p)$

$q_t := Q[q]$

$Q[q] := (1 - 0.5 \eta_t \lambda)Q[q] + \eta_t e P[p]$

$P[p] := (1 - 0.5 \eta_t \lambda)P[p] + \eta_t e q_t$

Project vectors  $Q[q]$  and  $P[p]$  to the feasible space.

**end for**

---

amount of eye movement data is not currently feasible because eye movement trackers are expensive and not widely available. Hence, the motivation here is to study how useful it would be to collect more data. The next section provides theoretical lower bound on how much eye movement data we need in order to infer the identity of an image in a simple scenario. Section 5.3.2 then generalizes this result to a more complex scenario.

### 5.3.1 The simplest scenario: binary topics and known queries

We assume that all images have a binary topic (let us call them cats and dogs) and that queries are also similarly binary. The relevance feedback is binary (relevant or non-relevant). However, the feedback is noisy and hence there are  $tp$ ,  $fp$ ,  $tn$ , and  $fn$  (true positive, false positive, true negative, and false negative) rates on the relevance feedback.

The question we want to answer is to what extent we can infer the true topic of an image? However, this depends on how well the identity of the queries is known. Hence, to simplify the setting let us assume that for each query we know its identity, i.e., whether the object of the search was a cat or a dog. Note that this assumption can only help, so the number of times image must be seen can only get lower due to this assumption.

In this setting each time an image is seen is a binary random variable that votes for the topic of the image. We denote by  $p$  that a single vote is true. Note that  $p$  depends on the distribution of images that were seen (whether they were relevant or non-relevant):

$$p = f_{pos} \frac{tp}{tp + fn} + f_{neg} \frac{tn}{fp + tn},$$

where  $f_{pos}$  is the fraction of relevant examples and  $f_{neg}$  is the fraction of non-relevant examples. Hence, a predictor that outputs a constant label does not work for all label distributions, because it has zero accuracy if all images are from the opposite class even if it can have a high accuracy in special circumstances.

The likelihood to infer a correct topic for an image is equal to the probability of an event where the number of correct votes for the topic outnumbers the number of wrong votes. A normal approximation of the number of votes normalized to range  $[0, 1]$  is:

$$N\left(p, \sqrt{\frac{p(1-p)}{k}}\right),$$

where  $p$  was the accuracy of the classifier and  $k$  is the number of times an image is seen. If the random variable from the above approximative distribution is  $\geq 0.5$  then the topic of an image is correctly assigned.

Note that the accuracy  $p$  of the classifier should hold for all frequencies of relevant images in the test set. This accuracy corresponds to the intersection of the ROC curve and the line from top-left corner to bottom-right corner of the true positive rate vs. false positive rate chart. This empirical value is 60% for relevance prediction from eye movements if estimated from data from the experiments described in Section 3. We still assume that the distribution of eye movement features within relevant and non-relevant classes is always the same.

Figure 2 gives insight on how the probability of inferring the correct topic behaves as a function of accuracy of relevance prediction and the number  $k$  of search sessions in which the image was seen. We see that in a scenario, where the accuracy of the eye movement relevance predictor is at least 60%, we are able to infer the identity of the image if it has been seen ca. 100 times. However, already when an image has been seen ca. ten times the topic assignment accuracy is 75% which can be exploited in applications.

### 5.3.2 Generalization to multiple topics

Analysis in the previous section was simple because it assumed that images and queries had binary topics and the identities of the queries were known. This section tries to generalize the analysis to a setting where each image can have multiple topics.

Let us first consider a simpler scenario where each image can belong to several topics (they are binary vectors that indicate topic participation) and queries belong to only one of the topics. In this setting the performance depends on the distribution of queries, because to infer whether an image belongs to a particular topic we need to see that image in several queries of that type. More precisely, we need to see as many queries of one type as in the binary case. Thus, if the distribution of topics in queries is uniform then the number queries to learn all topics of an image to some accuracy would be roughly

$$d(\text{corresponding number for the binary problem}),$$

where  $d$  is the number of the topics and also the dimension of the image vector.

Let us generalize further to a case where both images and queries are complicated. More formally, assume that they are distributed according to a standard normal distribution. This corresponds to the assumption made by the MF model. Each relevance prediction on image  $p$  in query  $q$  is generated as

$$r_{p,q} = p \cdot q + \epsilon,$$

where  $\epsilon$  is a normally distributed noise term:  $\epsilon \propto N(0, \sigma_\epsilon^2)$ . The prior distributions of  $p$  and  $q$  are  $d$ -dimensional standard normal distributions  $N(0, \sigma_p^2 I)$  and  $N(0, \sigma_q^2 I)$ , where variances  $\sigma_p^2$  and  $\sigma_q^2$  are both  $1/\sqrt{d}$ . This guarantees that the distribution of the true relevance  $p \cdot q$  is  $N(0, 1)$  which does not depend on the dimension  $d$ . Thus, the relative magnitude of relevance and the noise is on the same scale for all values of the dimension  $d$ .

From queries and relevance predictions on an image  $p$  the Bayes rule gives the posterior distribution of the image vector. However, even if we would know the posterior distribution we must answer two questions before we obtain results comparable to ones in the binary problem:

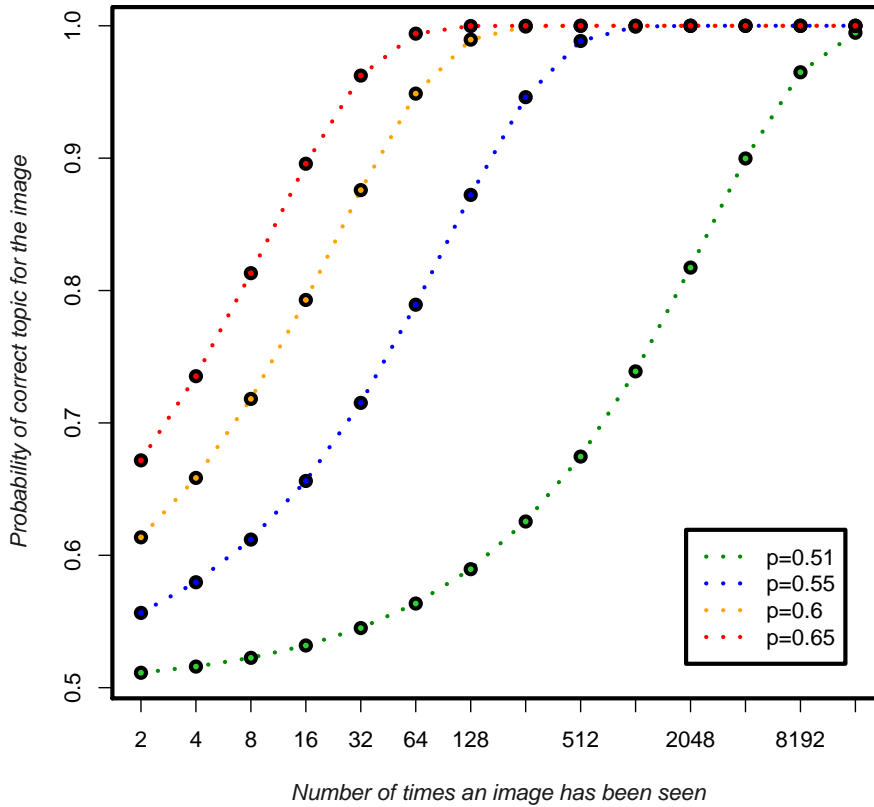


Figure 2: The probability of inferring a correct binary topic for an image given the number of times an image has been seen on the x-axis and four different possibilities for the accuracy of the relevance prediction.

1. How to estimate the variance of the noise term  $\epsilon$ ?
2. What is the connection between topic accuracies in the binary problem and the linear model considered here?

To compute the noise level  $\sigma_\epsilon$  we assume that the indicator  $(p \cdot q > 0)$  gives a binary relevance value, and  $(p \cdot q + \epsilon > 0)$  the observed noisy version of this. If the accuracy of the relevance predictor is 60% then this method gives a noise level of

$$\sigma_\epsilon \approx 3.2.$$

One possible answer to the second question is to associate to each topic accuracy a corresponding expected squared error rate  $\mathbb{E}(p \cdot q - \hat{p} \cdot q)^2$  resulting from approximating  $p$  with a sample  $\hat{p}$  from the posterior distribution. Hence, if we know the squared error rate for a single dimension, we can compute what kind of posteriors we will need in higher dimensions to obtain the same error rate. In particular, we could compute the error rates for a single dimension when the number of times seen is equal to the ones used in Figure 2.

Appendix A derives the following as an approximation to the expected squared error if  $\sigma_p^2$  and  $\sigma_q^2$  are  $1/\sqrt{d}$ :

$$\left(1 - \frac{1}{1 + \sigma_\epsilon^2 d/m}\right)^2 + \frac{1}{1 + m/(d\sigma_\epsilon^2)},$$

Table 6: In how many queries an image has to be seen in order to infer its identity from eye movements with the given accuracy. The rows correspond to how many dimensions the model has. The columns correspond to different accuracies. The relevance predictor is assumed to have 60% accuracy. The data is expected to conform to the model which is explained in the text.

model dimension	desired feature accuracy							
	0.62	0.65	0.73	0.79	0.88	0.94	0.98	0.9999
1	2	4	8	16	32	64	128	256
2	4	8	16	32	64	128	256	512
10	20	40	80	160	320	640	1280	2560
50	100	200	400	800	1600	3200	6400	12800
200	400	800	1600	3200	6400	12800	25600	51200
1000	2000	4000	8000	16000	32000	64000	128000	256000

where  $m$  is the number of times the image was seen. Unfortunately, it is clear from this expression that the dependence of the times seen  $m$  on the number of topics  $d$  is linear. In hindsight this is not that surprising because it means that learning  $d$  different things requires  $d$  times more data than learning a single thing. If this theoretical model holds in practise, then exploiting prior information, such as visual content of the images or user-provided labels, is necessary unless it is feasible to collect a large number of samples for each image. Table 6 gives in a convenient format the required number of times image has to be seen to reach different accuracies.

## 5.4 Experiments

This section describes a small-scale CBIR experiment and its analysis. The aim is to study how feasible matrix factorization is for providing information about the images in a real-world setting and how well do the results agree with theoretical bounds given in the previous sections.

**The data set** was collected using four experiment subjects. Each subject performed 30 tasks in which the target of the task was to find either a given dog or cat image. During each task, PicSOM image retrieval engine used relevance feedback from clicks to show three collages of 15 images from database containing 604 cat images and 721 dog images. Hence, each image was seen approximately 7 times.

**Analysis** was performed by setting the dimension of the latent vectors to two, which is a low value, but realistic because we had only two topics in the experiment. To derive the regularization constant data was split ten times into a training and test set (80%/20% of data), matrix factorization was performed on the training set, and the squared error was computed on the test set. The matrix factorization was then computed ten times on the whole data using the best regularization constant. Finally, to have an idea how good the resulting feature vectors were, we trained a classifier on the feature vectors using real topic labels and compared its accuracy against a classifier trained on random features.

**Results.** Figure 3 presents a visualization on the inferred features. Each point corresponds to either a query or on image. The coordinates are raw features computed by the matrix factorization. It is clear that the queries are very well separated due to fact that we have more information on each query than on images (visual inspection suggests that approximately 7 out of 30 queries are in a wrong place). The images are less so, but they are clearly clustered according to the topic. The best linear predictor is able to separate dog

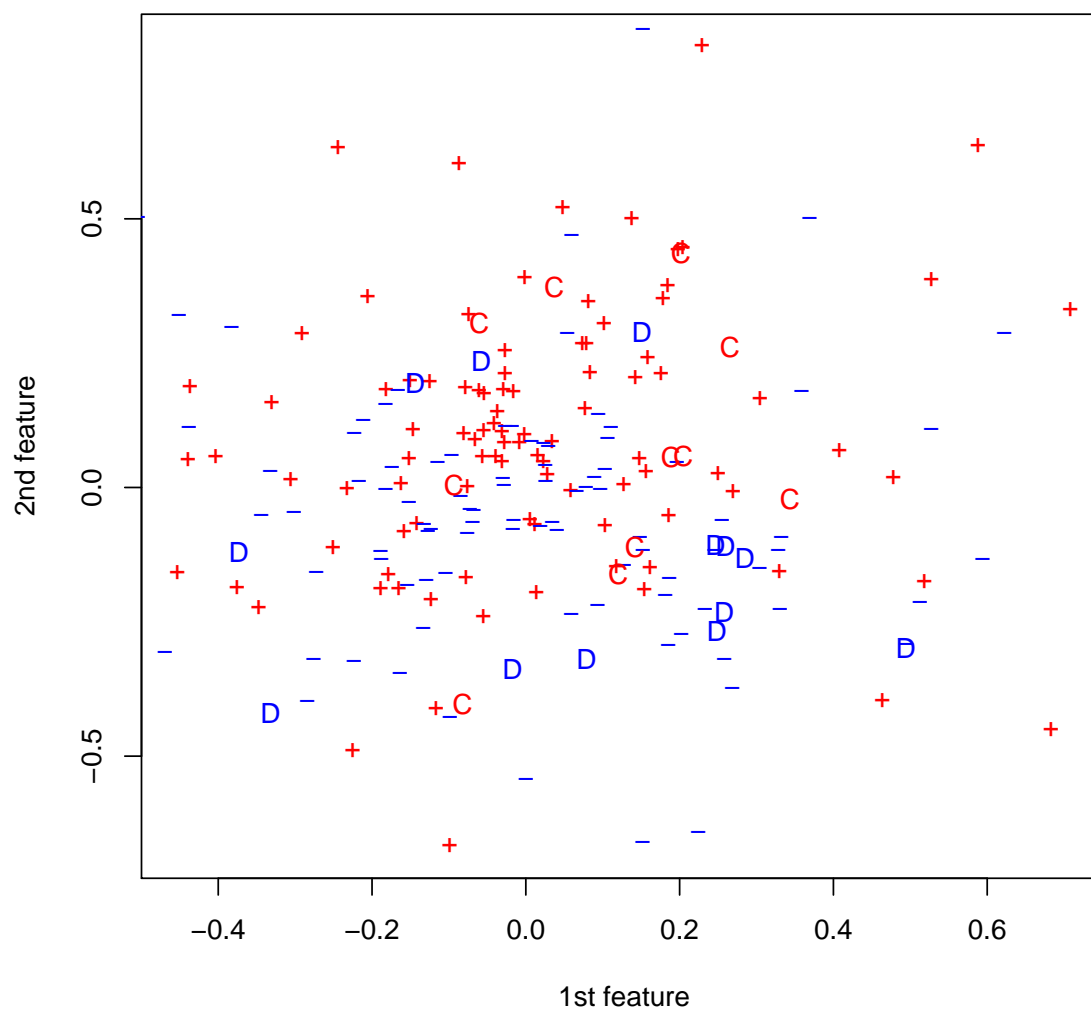


Figure 3: Visualization of the two dimensional inferred features for images and queries from relevance feedback data from a small scale CBIR experiment. Blue D corresponds to a query with dog target, red C to a query with cat target, blue - to an image with a dog in it, and red + to an image with a cat in it. Queries are better separated than the images, but the both are roughly divided so that dogs are in bottom left corner and the cats are in the top right corner.

and cat images with 65% mean accuracy, which is close to the lower bound in Figure 2. The mean accuracy of a predictor trained on random features was 53%.

## 6 Enhancing visual object detection with semantic tags

In learning vision tasks, the availability of ground truth and additional annotations can be extremely helpful. For example, in human detection, Bourdev and Malik [6] use 1000 3D annotations of human data in images to train “poselets”. Their approach achieves state-of-the-art results in person detection in PASCAL VOC 2007, 2008 and 2009 [14]. With the increasing availability of both automatic and manually created semantic annotations of various levels in image database [15, 25, 24], incorporating these extra information into the learning system in an effective and efficient manner becomes an important research question.

In this work, we consider systems that learn to detect visual objects from example images of a target object class. Given a test image, the system must detect the presence of objects in this class and provide the location for each detected instance in the form of a bounding box.

In the context of CBIR, the ability to detect high-level, semantically similar objects in images can be useful in deriving better overall description of images, which may then be used to enhance the relevance measures between images. This is especially important when the query is sensitive not only to presence of objects but also locations or configurations of multiple objects within the images.

A straightforward but costly approach to visual object detection is to first learn a binary classifier from some labeled examples, then apply the classifier over different parts of the test image using sliding windows of various sizes. One particularly attractive way to reduce the number of window locations that need to be evaluated is to first perform a generalized Hough transform [12] on image features, where each feature votes for potential locations of the target object. Only locations with high votes (i.e. peaks in the voting space) need to be evaluated. Maji and Malik [22] showed that a discriminatively trained probabilistic Hough transform can achieve state-of-the-art detection performance while evaluating on an order of magnitude fewer windows than the full sliding window approach.

We propose a different way to improve the quality of the Hough transform, by incorporating extra semantic information about the image features into the probabilistic Hough transform framework. By viewing the Hough transform as a form of probabilistic inference, we show that a better model can be learned from training examples that are enriched with semantic tagging, thus enhancing the quality of the Hough transform when applied to new examples. We evaluate our systems on both synthetic and real-world data and demonstrate that significant performance improvement can be obtained by incorporating simple, rather imperfect semantic information into the learning system.

### 6.1 Semantic Tags

Our approach makes use of information in the form of “semantic tags”. We consider the case where an expert has identified various parts of a target object in a meaningful way. For example, a car object can have semantic parts such as “window”, “wheel”, “door” etc. Assuming that a training image has been annotated where image regions corresponding to these parts have been identified, features extracted from the image can then be semantically tagged depending on the location from which they are extracted.

These semantic tags are unobservable “hidden” information, and are only available in the training set. Our approach incorporates this information through a probabilistic generalized Hough transform framework [12], where each feature votes for potential locations of a target

object. The semantic information improves the quality of the votes by linking together features through their higher level similarity. Figure 4 illustrates this.

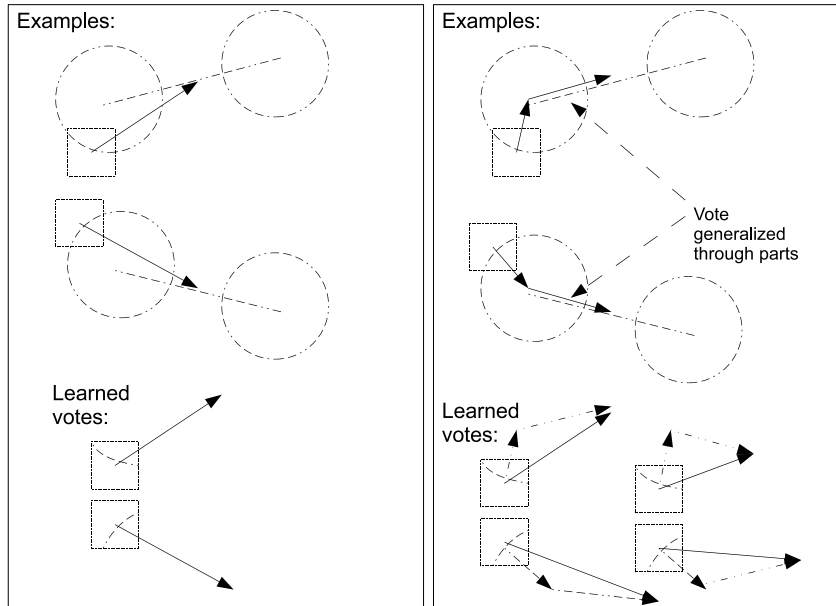


Figure 4: *Left*: Votes learned without using semantic tagging. Each feature votes for the object center directly. *Right*: Votes learned with semantic tagging. Each feature votes for the object center via the part center. Part-object relationships from other configurations propagate to every feature that belongs to the same semantic part

In the left figure, two training instances of “bicycles” are shown, where a feature (a local rectangular patch on the wheel) is observed in each instance. The system learns to vote for the object center relative to the location where the feature is observed. In the right figure, both features have been enriched with the same semantic tag “left wheel”, and the system can learn to vote for additional locations where the target object might be found since the relative location between the wheel and the bicycle center can be shared between features with the same tag.

In a sense, the semantic information creates new “virtual” examples that may help improve the model used for voting.

## 6.2 Hough Transform as Inference

We formulate the Hough transform as a probabilistic inference. In our framework, there exists three levels of information. The highest-level information is regarding the presence of the target object  $Q$  and its location  $x$ . The lowest-level information consists of the features observed in the input image. The intermediate-level information comes from the semantic tags. From the observed features, we wish to infer about  $Q$  and  $x$ . In this section, we show how this inference can be performed through a simple voting mechanism.

We make the simplifying assumption that each input example either contains exactly one target object, or none at all. Furthermore, exactly  $n$  features are extracted from each input example. We define a joint probability distribution from which every input example is drawn as

$$\Pr(Q, x, V_1, y_1, f_1, a_1, \dots, V_n, y_n, f_n, a_n)$$

where  $Q$  is a class label (e.g.  $Q = 1$  for “car” and  $Q = 0$  for “others”),  $x$  is an object

configuration (e.g. coordinates for center of object). Each 4-tuple  $(V_i, y_i, f_i, a_i)$  (for  $i = 1, \dots, n$ ) represents a randomly extracted feature from the instance. We assume that each feature  $f_i$ , observed at location  $a_i$ , is “generated” by some non-observable semantic part  $V_i$  (e.g. “front wheel”) at location  $y_i$ .

Given an input example, we extract features  $f_1, a_1, \dots, f_n, a_n$  and wish to evaluate

$$L(x) = \Pr(Q, x | f_1, a_1, \dots, f_n, a_n)$$

for all  $x$ . This is the probabilistic inference we need to perform through Hough transform. It can be shown that:

$$\log L(x) = c + \sum_{i=1}^n \Lambda(x, f_i, a_i)$$

where  $c$  is a constant. In the Hough voting scheme,  $\Lambda(x, f_i, a_i)$  represents the vote cast by feature  $f_i$  at location  $a_i$  onto target location  $x$ . The log-likelihood  $\log L(x)$  at  $x$  is proportional to the sum of the vote cast by each of the features  $f_1 \dots f_n$ . This “vote function”  $\Lambda(x, f_i, a_i)$  is given by:

$$\Lambda(x, f_i, a_i) = \log \sum_{V_i} \omega(V_i, f_i) \lambda(x, V_i, f_i, a_i)$$

where

$$\omega(V_i, f_i) = \frac{\Pr(V_i|Q) \Pr(f_i|V_i)}{\sum_{V_i} \Pr(V_i|Q) \Pr(f_i|V_i)}$$

and

$$\lambda(x, V_i, f_i, a_i) = \sum_{y_i} \Pr(x|Q, V_i, y_i) \Pr(y_i|V_i, f_i, a_i)$$

can be estimated from the training examples with semantic tags.

### 6.3 Experiments

We evaluate our approach on both synthetic as well as real-world problems. The target object for the synthetic problem is a simple “bicycle” with two wheels as in Fig. 4. Each training example contains tags for the left and the right wheel. Figure 5 show detection results with and without using the semantic tags. The results show that, with semantic tagging, less training examples are needed to achieve the same level of performance.

For real-world data, we use the UIUC Car database [1]. While it is simpler than some other databases, it remains challenging when only a small number of training examples are used. We use standard dense SIFT features [21]. For semantic tagging, the positive examples have been tagged with the following object parts: front, rear, front wheel, rear wheel, front window, rear window and car body. The best detection performance using Hough transform learned without using semantic tagging is 82.65% while with semantic tagging, this increases to 88.25%.

We further improve the system by adding a second-stage verifying classifier, trained using standard Support Vector Machine with the histogram intersection kernel [20] is added to the Hough peaks to improve the overall detection performance. Figure 6 shows the detection results. Notice that semantic tags help the most when the available training set is small.

## 7 Conclusion

This work studied relevance prediction from eye movements in CBIR settings.

The first part of the work studied how well an eye movement relevance predictor performs in an environment that is different from the one where it was trained. The result was that a

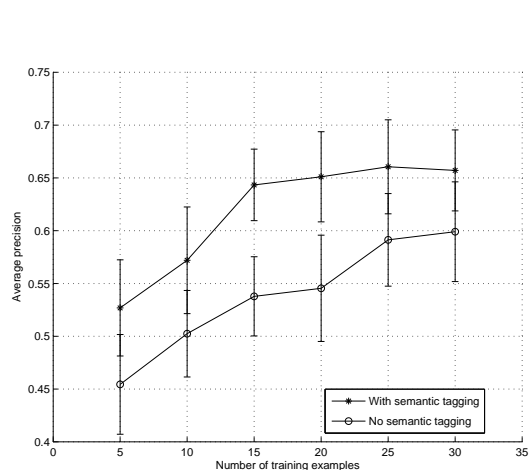


Figure 5: Detection performance for synthetic bicycles with different training set sizes.

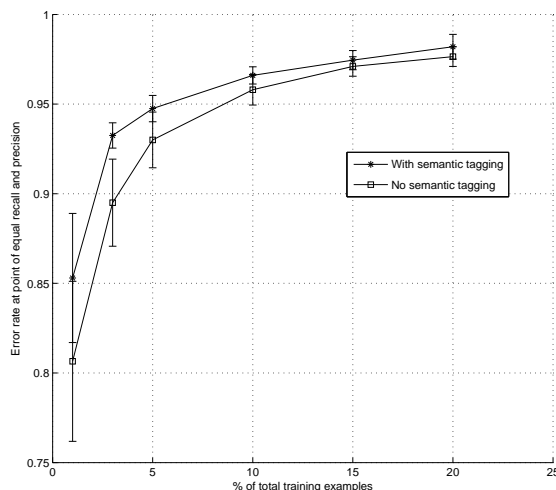


Figure 6: Detection performance for UIUC Car with different training set sizes.

pre-trained predictor ranks images well within queries. However, absolute value of predictions depends heavily on the user.

The second result was that in experiments the relevance predictions had a statistically significant component that depends on the image rather than task. However, taking image-specific biases into account in relevance prediction did not result in increased prediction accuracy.

The third part of the work studied how well it is possible to infer latent features of images by combining relevance predictions on images from several queries. The result was that a simple matrix factorization performed well on a small-scale data relative to noise levels of eye movements. A theoretical model suggests that the amount of samples needed to accurately learn complicated image content can be very large. Hence, an approach that uses visual content of images to guide the learning might be necessary.

Instead of eye movements the final part of the work studied how to efficiently find potential locations of interesting objects in images by augmenting Hough transform with semantic tagging. The results show that, with semantic tagging, less training examples are needed to achieve the same level of performance.

**Acknowledgments:** We wish to thank Craig Saunders of and Jorma Laaksonen for their valuable comments on the draft version of this deliverable.

## References

- [1] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.
- [2] Peter Auer, Zakria Hussain, Samuel Kaski, Arto Klami, Jussi Kujala, Jorma Laaksonen, Alex P. Leung, Kitsuchart Pasupa, and John Shawe-Taylor. Pinview: Implicit feedback in content-based image retrieval. In *Workshop on Applications of Pattern Analysis*, JMLR Workshop and Conference Proceedings, 2010.

- [3] Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Corinna Cortes, and Mehryar Mohri. Polynomial semantic indexing. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 64–72. NIPS Foundation (<http://books.nips.cc>), 2009.
- [4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [5] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. NIPS Foundation (<http://books.nips.cc>), 2008.
- [6] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *International Conference on Computer Vision*, sep 2009.
- [7] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.
- [8] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.
- [9] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. Technical report, ICSI, 1999. Technical Report TR-99-006.
- [10] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.
- [12] Richard O. Duda and Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [13] W. Eihäuser, U. Rutishauser, and C. Koch. Task-demands can immediately reverse the effects of sensory-driven saliency in complex visual stimuli. *Journal of Vision*, 8:1–19, 2008.
- [14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. <http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html>.
- [15] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.
- [16] David Grangier and Samy Bengio. A discriminative kernel-based approach to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1371–1384, 2008.
- [17] J. Henderson, J. R. Brockmole, M. S. Castelano, and M. Mack. Visual saliency does not account for eye movements during visual search in real-world scenes. In *Eye movements: A window on mind and brain*, pages 538–562. Elsevier, 2007.

- [18] Arto Klami, Kitsuchart Pasupa, Craig Saunders, and Teófilo E. de Campos. Prediction of relevance of an image from a scan pattern. PinView FP7-216529 Deliverable D2.1, 2008.
- [19] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [20] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.
- [21] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [22] S. Maji and J. Malik. Object detection using a max-margin Hough transform. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1038–1045, 2009.
- [23] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal Estimated sub-GrAdient sOlver for SVM. In *Proceedings of the 24th International Conference on Machine Learning*, pages 807–814, New York, NY, 2007. ACM.
- [24] Roger C. F. Wong and Clement H. C. Leung. Automatic semantic annotation of real-world web images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1933–1944, 2008.
- [25] Benjamin Yao, Xiong Yang, and Song-Chun Zhu. Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 169–183, 2007.
- [26] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 116, New York, NY, USA, 2004. ACM.

## A Approximation to expected squared error when inferring topics

This appendix computes an approximation to the squared error needed in Section 5.3.2. Let us first derive the posterior distribution of an image vector when it has been seen  $m$  times in the model described in Section 5.3.2. Recall that the image and query vectors had normal priors with distributions  $N(0, \sigma_p^2 I)$  and  $N(0, \sigma_q^2 I)$ . Let  $Q$  be a matrix with the queries  $[q_1, \dots, q_m]$  as columns. Let  $r$  be a column vector of observed relevance values in the queries. A standard textbook [4] shows that a sample  $\hat{p}$  from the posterior distribution of an image  $p$  has a normal distribution with a mean and an inverse variance of:

$$\begin{aligned}\mathbb{E} \hat{p} &= \frac{1}{\sigma_\epsilon^2} \mathbb{V}(\hat{p}) Q r \\ \mathbb{V}(\hat{p})^{-1} &= \frac{1}{\sigma_p^2} I + \frac{1}{\sigma_\epsilon^2} Q Q'\end{aligned}$$

The queries have a normal prior which we can use to approximate the formulas. The prior implies that  $\mathbb{E}QQ' = m\sigma_q^2 I$  which results into following formulas:

$$\begin{aligned}\mathbb{V}(\hat{p})^{-1} &\approx \left( \frac{1}{\sigma_p^2} + \frac{m\sigma_q^2}{\sigma_\epsilon^2} \right) I \\ \mathbb{E}\hat{p} &\approx \mathbb{E} \left( \frac{1}{\sigma_\epsilon^2/\sigma_p^2 + m\sigma_q^2} Qr \right) \\ &= \frac{1}{\sigma_\epsilon^2/\sigma_p^2 + m\sigma_q^2} (m\sigma_q^2 p + \mathbb{E}(Q\epsilon)) \\ &\approx \frac{m\sigma_q^2}{\sigma_\epsilon^2/\sigma_p^2 + m\sigma_q^2} p\end{aligned}$$

We were interested in computing the following expected squared error:

$$\mathbb{E}(\hat{p} \cdot q - p \cdot q)^2 = \mathbb{E}((\hat{p} - p) \cdot q)^2 = \sigma_q^2 \mathbb{E}\|\hat{p} - p\|^2.$$

Let us approximate that the expected norm squared  $\mathbb{E}\|\hat{p} - p\|^2$  is its bias squared plus variance. The bias squared is  $\mathbb{E}\|p - \mathbb{E}\hat{p}\|^2$  or more precisely:

$$\left( 1 - \frac{1}{1 + \sigma_\epsilon^2/(m\sigma_p^4)} \right)^2 \mathbb{E}\|p\|^2 = \left( 1 - \frac{1}{1 + \sigma_\epsilon^2/(m\sigma_p^4)} \right)^2 d\sigma_p^2.$$

The variance is:

$$\frac{d}{1/\sigma_p^2 + m\sigma_q^2/\sigma_\epsilon^2}.$$

Hence, if  $\sigma_p^2 = \sigma_q^2 = 1/\sqrt{d}$  the error squared is:

$$\left( 1 - \frac{1}{1 + \sigma_\epsilon^2 d/m} \right)^2 + \frac{1}{1 + m/(d\sigma_\epsilon^2)}.$$