



Deliverable D4.2.1

Report on using side information for exploration-exploitation trade-offs

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

Project ref. no.	FP7-216529
Project acronym	PinView
Status and version	Final, Revision: 2.00
Contractual date of delivery	31.12.2009
Actual date of delivery	4.1.2010
Deliverable number	D4.2.1
Deliverable title	Report on using side information for exploration-exploitation trade-offs
Nature	report
Dissemination level	PU – Public
WP contributing to the deliverable	WP4 Exploration vs. exploitation
Task contributing to the deliverable	Task 4.2 Algorithms to deal with side information without delayed feedback
WP responsible	Montanuniversität Leoben
Task responsible	Montanuniversität Leoben
Editor	Peter Auer <auer@unileoben.ac.at>
Editor address	University of Leoben, Franz-Josef-Straße 18, A-8700 Leoben, Austria
Authors in alphabetical order	Peter Auer, Alex Leung, Zakria Hussain, John Shawe-Taylor
EC Project Officer	Pierre-Paul Sondag
Keywords	CBIR with Relevance Feedback, Exploration-Exploitation trade-off, Bandit problem, Side information
Abstract	We have developed an algorithm which makes use of available side information to retrieve relevant images from a database, based on relevance feedback. The algorithm deals with the exploration-exploitation trade-off caused by imperfect relevance estimates by using upper confidence values of these estimates.

List of annexes

none

Contents

1 Overview	4
2 Introduction	5
3 The user model for the filtering task	5
3.1 Modeling the relevance scores	6
4 The LINREL algorithm	6
4.1 Regularized LINREL for $n = 1$	7
4.2 Kernelized LINREL	8
4.3 Integration of multiple kernel learning into LINREL	8
4.4 LINREL for presenting collages of images	8
5 Experiments	9
5.1 Experiments with several kernels	10
5.2 Experiments with collages	10
6 Conclusion	11
A Derivation of equation (7)	12

1 Overview

This deliverable reports on how the LINREL algorithm can be used for CBIR tasks with relevance feedback, when the user wants to retrieve relevant images from a database. The LINREL algorithm makes use of side information — typically various image features — to estimate the relevance of an image. Since these estimates are imperfect, when selecting possibly relevant images the algorithm needs to trade off between high estimated relevance and possible gain of information to improve the estimates. The original LINREL algorithm is based on linear estimates of relevance and selects a single image in each iteration. The original algorithm has been extended in several ways: it has been kernelized to allow non-linear relevance estimates, a regularization to avoid overfitting has been devised, and several methods to select collages instead of single images have been considered. The algorithm has been evaluated and compared with the original PicSOM system, using the VOC'2007 Challenge image dataset.

This deliverable is a main output of Task 4.2, *Algorithms to deal with side information without delayed feedback*, of the PinView project. It is accompanied with Deliverable D4.2.2, *Implementation of using side information for exploration-exploitation trade-offs*, which makes the developed algorithms available to the PinView consortium. This Deliverable D4.2.2 is not being submitted separately, but it is included in the PinView system submitted as Deliverable D8.5.1, *Prototype framework implementation, stage I*.

This deliverable is partly based on Deliverable D4.1, *Models of exploration-exploitation trade-offs*, which proposed two different models of CBIR with relevance feedback. The first of these models has been the focus of Deliverable D4.1. This Deliverable D4.2.1 focuses on the second model and develops the LINREL algorithm for this model. The LINREL algorithm relies on relevance feedback, which is provided by methods developed in work packages WP1, WP2, and WP5. The kernel functions for the LINREL algorithm, which encode a learned metric for images, will be provided by methods of WP3. This work will be continued in Task 4.3, *Development of algorithms to deal with delayed feedback*, to include also delayed feedback.

2 Introduction

In Deliverable D4.1, *Models of Exploration-Exploitation Trade-offs* [4] of the PinView project, we have formalized two models for CBIR with relevance feedback. The first model assumes that the user is searching for a particular image, and that in several rounds the user gives feedback by selecting the best match in a collage of images, until a satisfactory image is found. Some algorithms and experiments for this first model have already been reported in Deliverable D4.1. The algorithms which have been tested are: robust binary search, calculating representative cluster centers, and random sampling according to a Bayesian prior. In our experiments random sampling according to a Bayesian prior performed best.

The second model is motivated by a filtering task, where a user wants to find a set of images relevant to his query. When presented with a collage of images, the user marks¹ all images which are relevant to his query. The goal of the search algorithm is to present many relevant images in the first few iterations of the search.

This deliverable reports on algorithms and experiments for this second model. The algorithms in this deliverable build upon the LINREL algorithm proposed in [2, Section 4]. The extensions to the LINREL algorithm are the following:

- replacing the explicit linear model by a kernel feature space,
- adding regularization to avoid overfitting, and
- accommodating the presentation of a collage of images instead of a single image.

The algorithms described in this deliverable have been implemented and are available in the PinView prototype framework, details are given in Deliverable D8.5.1, *Prototype framework implementation, stage I*.

The remaining deliverable is organized as follows: Section 3 revisits the filtering user model already described in Deliverable D4.1. This model is extended by considering that not single images are presented to the user but collages of images. In Section 4 the LINREL algorithm is discussed. The regularized LINREL algorithm is described in Section 4.1. The kernelized version of LINREL is described in Section 4.2. The integration of the multiple-kernel learning algorithm developed in WP3 is described in Section 4.3. In Section 4.4 methods for dealing with collages instead of single images are discussed. Experimental results are reported in Section 5, where the LINREL algorithm is compared with the original PicSOM system [8].

3 The user model for the filtering task

We assume that the user is looking for a set of relevant images I in an image database \mathcal{D} . The relevance of an image is determined by the user query, about which the search engine is informed only by relevance feedback. The goal of the search engine is to present mostly relevant images to the user, and only a small number of irrelevant images. The feedback of the user is given by a relevance score $y \in [0, 1]$, with 0 meaning not relevant, 1 meaning relevant, and possible degrees of relevance in between. The relevance score can be given by an explicit binary feedback (e.g. mouse clicks), or implicitly (e.g. by recorded eye movements). The formal protocol for this model is the following:

- In each iteration $t = 1, 2, \dots$:
 - The search engine selects an image $I_t \in \mathcal{D}$ and presents it to the user.
 - The user’s feedback is given by the relevance score $y_t \in \{0, 1\}$.

¹This “marking” might be either explicit, e.g. by mouse clicks, or it might be implicit by recorded eye movements.

The performance of the search engine is determined by the number of relevant images returned in a certain number of iterations. We note that in this protocol only a single image is presented in each round and that the relevance score is binary. This is the original protocol used to develop the LINREL algorithm in [2]. For the CBIR tasks considered in the PinView project we need to extend this protocol:

- In each iteration $t = 1, 2, \dots$:
 - The search engine selects n images $I_{t,1}, \dots, I_{t,n} \in \mathcal{D}$ and presents them to the user.
 - For each presented image the search engine receives a relevance score $y_{t,i} \in [0, 1]$, $i = 1, \dots, n$.

Thus the search engine needs to select a fixed number n of images, and the relevance scores might be in between 0 and 1.

3.1 Modeling the relevance scores

To be able to learn about the user’s query from the relevance scores, we are making assumptions about how the relevance scores are generated. We assume that an image I is represented by a normalized vector \mathbf{x}_I of features, $\mathbf{x}_I \in \mathbb{R}^d$, $\|\mathbf{x}_I\| = 1$. Furthermore, we assume that the relevance score y_I of an image I is a random variable with expected value $\mathbf{E}[y_I] = \mathbf{x}_I \cdot \mathbf{w}$, $\mathbf{w} \in \mathbb{R}^d$, such that the expected relevance score is a linear function of the image features. The unknown weight vector \mathbf{w} is essentially the representation of the user’s query and determines the relevance of images.

4 The LINREL algorithm

In [2] the LINREL algorithm was devised for a slight variant of the model described in the previous section. In this section we describe the LINREL algorithm with the necessary modifications to accommodate our model. We restrict ourselves to the case $n = 1$, i.e. only one image is presented to the user in each iteration. The general case, i.e. image collages, is discussed in Section 4.4.

The user model for the filtering tasks (in particular the model for the relevance scores) confronts the search engine with an exploration-exploitation trade-off. Typically the search engine will maintain an implicit or explicit representation of an estimate $\hat{\mathbf{w}}$ of the unknown weight vector \mathbf{w} . When selecting the next image for presentation to the user, the search engine might simply select the image with highest estimated relevance score based on $\hat{\mathbf{w}}$. But since the estimate $\hat{\mathbf{w}}$ might be inaccurate, this exploitative choice might be suboptimal. Alternatively, the search engine might exploratively select an image for which the user feedback improves the accuracy of the estimate $\hat{\mathbf{w}}$, enabling better image selections in subsequent iterations.

In each iteration t , LINREL obtains an estimate $\hat{\mathbf{w}}_t$ by solving the linear regression problem

$$\mathbf{y}_t \approx \mathbf{X}_t \cdot \hat{\mathbf{w}}_t,$$

where

$$\mathbf{y}_t = \begin{pmatrix} y_1 \\ \vdots \\ y_{t-1} \end{pmatrix}$$

is the column vector of relevance scores received so far, and

$$\mathbf{X}_t = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{t-1} \end{pmatrix}$$

is the matrix of row feature vectors of the images presented so far. Based on the estimated weight vector $\hat{\mathbf{w}}$, LINREL calculates an estimated relevance score $\hat{y}_I = \mathbf{x}_I \cdot \hat{\mathbf{w}}$ for each image I that has not already been presented to the user. To deal with the exploration-exploitation trade-off, LINREL selects for presentation not the image with largest estimated relevance score, but the image with the largest upper confidence bound for the relevance score. The upper confidence bound for an image I is calculated as $\hat{y}_I + c\hat{\sigma}_I$, where $\hat{\sigma}_I$ is an upper bound on the standard deviation of the relevance estimate \hat{y}_I . The constant c is used to adjust the confidence level of the upper confidence bound.

The rationale for using upper confidence bounds is that an image gets selected (a) if its relevance score is indeed large, or (b) if the estimated relevance score is rather unreliable and the resulting confidence interval is large. Case (a) gives an exploitative choice, while case (b) improves the estimates and thus is explorative. It has been shown that upper confidence bounds are a versatile tool to balance exploration and exploitation in online selection problems [1, 3, 2]. In [2] rigorous bounds on the performance of LINREL are proven. The details of our variant of the LINREL algorithm are given in the next section.

4.1 Regularized LINREL for $n = 1$

For solving the regression problem $\mathbf{y}_t \approx \mathbf{X}_t \cdot \hat{\mathbf{w}}_t$ we are using regularized linear regression such that

$$\hat{\mathbf{w}}_t = (\mathbf{X}_t^\top \mathbf{X}_t + \mu \mathbf{I})^{-1} (\mathbf{X}_t^\top \mathbf{y}_t) \quad (1)$$

where \mathbf{I} is the identity matrix and $\mu > 0$ is the regularization parameter. This solution is obtained from the optimization problem

$$\hat{\mathbf{w}}_t = \arg \min_{\mathbf{w}} \{ \|\mathbf{y}_t - \mathbf{X}_t \cdot \mathbf{w}\|^2 + \mu \|\mathbf{w}\|^2 \}. \quad (2)$$

To bound the variance $\hat{\sigma}_I^2 = \mathbf{V}[\hat{y}_I]$ of the estimate $\hat{y}_I = \mathbf{x}_I \cdot \hat{\mathbf{w}}_t$ for a feature vector \mathbf{x}_I , we define $\mathbf{a}_I \in \mathbb{R}^{t-1}$ as

$$\mathbf{a}_I = \mathbf{x}_I \cdot (\mathbf{X}_t^\top \mathbf{X}_t + \mu \mathbf{I})^{-1} \mathbf{X}_t^\top \quad (3)$$

such that

$$\hat{y}_I = \mathbf{x}_I \cdot \hat{\mathbf{w}}_t = \mathbf{x}_I \cdot (\mathbf{X}_t^\top \mathbf{X}_t + \mu \mathbf{I})^{-1} (\mathbf{X}_t^\top \mathbf{y}_t) = \mathbf{a}_I \cdot \mathbf{y}_t$$

and

$$\mathbf{V}[\hat{y}_I] \approx \sum_{i=1}^{t-1} a_i^2 \mathbf{V}[y_i] \leq \frac{1}{4} \sum_{i=1}^{t-1} a_{I,i}^2 = \frac{1}{4} \|\mathbf{a}_I\|^2 \quad (4)$$

since the variance of any random variable bounded in $[0, 1]$ is at most $1/4$. The first approximation in (4) relies on the fact that the coefficients a_j are fixed and do not depend on the observed relevance scores y_i . This is actually not the case since the images selected for presentation do depend on the relevance scores. In [2] this problem is formally fixed by using a more complex algorithm SUPLINREL, which guarantees that the variance $\mathbf{V}[\hat{y}]$ is indeed bounded by the right hand side of (4). An alternative, closely related and also formally correct method has been devised in [5]. For practical purposes these more complex algorithms seem unnecessary and very good results can be obtained by the simple version of the LINREL algorithm proposed in this section², using the variance bound in (4).

Concluding, in each iteration t the regularized LINREL algorithm for $n = 1$, calculates

$$\mathbf{a}_I = \mathbf{x}_I \cdot (\mathbf{X}_t^\top \mathbf{X}_t + \mu \mathbf{I})^{-1} \mathbf{X}_t^\top \quad (5)$$

for each image I and selects for presentation the image I_t which maximizes

$$I_t = \arg \max_I \left\{ \mathbf{a}_I \cdot \mathbf{y}_t + \frac{c}{2} \|\mathbf{a}_I\| \right\} \quad (6)$$

for some specified constant $c > 0$.

²It is actually likely that the more complex algorithms perform worse, since the theoretically better performance bounds come with large constant factors which render these algorithms less useful for practical applications.

4.2 Kernelized LINREL

To accommodate the use of implicit image features given by kernel functions, we need to modify the LINREL algorithm appropriately. We assume that the feature vector $\mathbf{x}_I = \Phi(I)$ of an image I is not given explicitly, but can be accessed only implicitly through a kernel function $k(\cdot, \cdot)$ which returns the inner product of two images in feature space, $k(I_1, I_2) = \Phi(I_1) \cdot \Phi(I_2)$. Using kernel functions allows us to be very flexible about distance measures between images and even to adapt the distance measure on the fly. As an example, the integration of multiple kernel learning (an outcome of WP3) into the kernelized LINREL algorithm is described in Section 4.3.

To kernelize LINREL, we revisit the optimization problem (2) and its solution (1). This solution can alternatively be written as

$$\hat{\mathbf{w}}_t = X_t^\top (X_t X_t^\top + \mu \mathbf{I})^{-1} \mathbf{y}_t \quad (7)$$

(for completeness the simple derivation is given in Appendix A), such that \mathbf{a}_I in (5) can be rewritten as

$$\mathbf{a}_I = \mathbf{x}_I \cdot \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{X}_t^\top + \mu \mathbf{I})^{-1} = (k(I, I_1) \cdots k(I, I_{t-1})) \cdot (\mathbf{K}_t + \mu \mathbf{I})^{-1},$$

where I_1, \dots, I_{t-1} are the images selected in iterations $i = 1, \dots, t-1$ and \mathbf{K}_t is the Gram matrix

$$\mathbf{K}_t = (k(I_i, I_j))_{1 \leq i, j \leq t-1}.$$

Thus \mathbf{a}_I can be calculated by using only the kernel function $k(\cdot, \cdot)$. Since the selection rule (6) remains unchanged, this gives the kernelized version of LINREL.

4.3 Integration of multiple kernel learning into LINREL

In this section we describe how the Multiple Kernel Learning (MKL) algorithm proposed in Deliverable D3.1, *Basic metric learning* [7], can be integrated into the LINREL algorithm.

The MKL algorithm learns a suitable metric between images in respect to a user query, by finding a good linear combination of given kernel functions k_1, \dots, k_P ,

$$k_\eta(\cdot, \cdot) = \sum_{p=1}^P \eta_p k_p(\cdot, \cdot),$$

where $\eta = (\eta_1, \dots, \eta_P)$, $\eta_p \geq 0$. Such a linear combination again forms a kernel function [9].

In the current version, the MKL algorithm works only for binary relevance scores $y \in \{0, 1\}$. The MKL algorithm receives as input a list $\mathcal{L} = \langle (I_\tau, y_\tau) \rangle$ of images I_τ labeled with their relevance scores y_τ , and a set \mathcal{U} of unlabeled images. Then the MKL algorithm calculates suitable linear weights η and returns the kernel matrix $\mathbf{K} = (k_\eta(I_i, I_j))_{i, j}$, for all $I_i \in \mathcal{L}$ and $I_j \in \mathcal{L} \cup \mathcal{U}$.

Since only in each iteration the kernelized LINREL algorithm relies on a fixed kernel function, the integration of the MKL algorithm into LINREL is very simple: LINREL calls the MKL algorithm at the beginning of each iteration and then uses the kernel matrix returned by the MKL algorithm.

4.4 LINREL for presenting collages of images

An obvious way to adapt LINREL to present $n > 1$ images in each iteration, is to present those n images with the highest upper confidence bounds (6). While this is a reasonable approach, there are considerations which are advocating also other options.

A drawback of relying only on the upper confidence values is that the similarities of the selected images are not considered. It might be the case that all selected images are very similar explorative choices, which means that the feedback for the n selected images is not much more useful than the feedback for a single image. A simple method to avoid this problem is to use just one image for exploration — by selecting the image with the maximum upper confidence bound $\hat{y}_I + c\hat{\sigma}_I$ — and selecting the remaining $n - 1$ images exploitatively just according to the maximum estimated

relevance scores \hat{y}_I . Thus this second method does as little exploration as possible, while the first method does the maximal possible exploration.

The third method we consider covers the middle ground between these two extreme methods. It selects the images I_1, \dots, I_n for presentation sequentially, still relying on the upper confidence bounds, but when selecting image I_k it takes into account the exploration already done by the images I_1, \dots, I_{k-1} . Formally, this is achieved by replacing \mathbf{X}_t by $\mathbf{X}_{t,k}$ and \mathbf{y}_t by $\mathbf{y}_{t,k}$ in (5) and (6) when selecting the k -th image $I_{t,k}$ in iteration t . The matrix $\mathbf{X}_{t,k}$ augments \mathbf{X}_t by the feature vectors of the already selected images $I_{t,1}, \dots, I_{t,k-1}$,

$$\mathbf{X}_t = \begin{pmatrix} \Phi(I_{1,1}) \\ \vdots \\ \Phi(I_{t-1,n}) \end{pmatrix}, \quad \mathbf{X}_{t,k} = \begin{pmatrix} \mathbf{X}_t \\ \Phi(I_{t,1}) \\ \vdots \\ \Phi(I_{t,k-1}) \end{pmatrix}. \quad (8)$$

The vector $\mathbf{y}_{t,k}$ augments the vector of previous outcomes \mathbf{y}_t by the *expected outcomes* for the already selected images,

$$\mathbf{y}_t = \begin{pmatrix} y_1 \\ \vdots \\ y_{t-1} \end{pmatrix}, \quad \mathbf{y}_{t,k} = \begin{pmatrix} \mathbf{y}_t \\ \hat{y}_{t,1} \\ \vdots \\ \hat{y}_{t,k-1} \end{pmatrix}, \quad (9)$$

where $\hat{y}_{t,j} = \hat{y}_{I_{t,j}}$ is calculated according to (3). The rationale for this construction is that the confidence terms are modified according to the selected images, while the solution of the optimization problem (2) is not changed by replacing \mathbf{X}_t by $\mathbf{X}_{t,k}$ and \mathbf{y}_t by $\mathbf{y}_{t,k}$. The solution is not changed since $\hat{y}_{t,j} = \Phi(I_{t,j}) \cdot \hat{\mathbf{w}}_t$. In contrast, augmenting \mathbf{X}_t does change the vector \mathbf{a}_I in (5), and thus also changes the confidence term $\frac{c}{2} \|\mathbf{a}_I\|$ in (6). Except for the initial phase of the search, the confidence term decreases by augmenting \mathbf{X}_t .³

Summing up we have three methods for selecting collages of images:

1. Select the images $I_{t,1}, \dots, I_{t,n}$ with maximal upper confidence bounds $\mathbf{a}_I \cdot \mathbf{y}_t + \frac{c}{2} \|\mathbf{a}_I\|$, where $\mathbf{a}_I = \mathbf{x}_I \cdot (\mathbf{X}_t^\top \mathbf{X}_t + \mu \mathbf{I})^{-1} \mathbf{X}_t^\top$ with \mathbf{X}_t from (8) and \mathbf{y}_t from (9).
2. Select image $I_{t,1}$ with the maximal upper confidence bound $\mathbf{a}_I \cdot \mathbf{y}_t + \frac{c}{2} \|\mathbf{a}_I\|$.
Select the images $I_{t,2}, \dots, I_{t,n}$ with maximal estimated relevance scores $\mathbf{a}_I \cdot \mathbf{y}_t$.
3. For $k = 1, \dots, n$ select image $I_{t,k}$ which maximizes $\mathbf{a}_I \cdot \mathbf{y}_{t,k} + \frac{c}{2} \|\mathbf{a}_I\|$, where $\mathbf{a}_I = \mathbf{x}_I \cdot (\mathbf{X}_{t,k}^\top \mathbf{X}_{t,k} + \mu \mathbf{I})^{-1} \mathbf{X}_{t,k}^\top$.⁴

5 Experiments

In this section we are reporting results of comparing several variants of the LINREL algorithm as well as comparing LINREL with the selection mechanism originally implemented in the PicSOM system [8]. As in Deliverable D4.1 [4] we are using the data of the VOC'2007 Challenge [6] for the experiments. The results for the PicSOM system have been provided by Jorma Laaksonen (TKK), using the images of the VOC'2007 training dataset and presenting collages of $n = 15$ images.

The VOC'2007 training dataset consists of 2501 images which contain objects from 20 categories. An image may contain objects from several categories. For each class we are evaluating the performance in respect to a user query which is looking for images with objects from this class. The user feedback y is assumed to be binary, $y \in \{0, 1\}$, correctly marking all relevant images. Performance

³This is because the eigenvalues of $\mathbf{X}_t^\top \mathbf{X}_t$ increase. The analysis in [2] can be used to calculate $\|\mathbf{a}_I\|$.

⁴Combining this third method with multiple kernel learning is a bit more involved, since for each selection $I_{t,k}$, $k = 1, \dots, n$, the MKL algorithm needs to be called.

is measured by average precision \bar{p} , where the precision p_t after iteration t is defined by the fraction of relevant images presented in the first t iterations,

$$p_t = \frac{1}{tn} \sum_{i=1}^t \sum_{k=1}^n y_{t,k},$$

and

$$\bar{p} = \frac{1}{T} \sum_{t=1}^T p_t,$$

when n is the size of the collages and T is the total number of iterations.

We conducted two sets of experiments. One set of experiments used collages of size $n = 15$ and a total of $T = 10$ iterations, which seems realistic for an actual filtering task. For evaluating different kernels, we also conducted experiments with the presentation of single images ($n = 1$) and $T = 150$ iterations. As raw features we used the 11 feature types implemented in the PicSOM system, each represented by a 2-dimensional SOM value (from the self-organizing maps which PicSOM learns unsupervised for each feature type). We concatenated these SOM values into a 22-dimensional raw feature vector. The resulting raw feature vectors \mathbf{x} were normalized such that $\|\mathbf{x}\| = 1$.

In all experiments we set the regularization parameter $\mu = 1$ and the confidence parameter $c = 0.1$. Since the feature vectors are normalized, $\mu = 1$ seems a reasonable choice. The confidence parameter was set to a small value since we were expecting a variance of the confidence score y that is far smaller than the maximum possible one (which was used to derive (4)). All reported numbers are averages over 100 repetitions of the experiments.

5.1 Experiments with several kernels

Since the implementation the MKL algorithm for learning a query specific metric will become available only in year 3 of the project (Task 3.3), we used two standard kernel functions for testing the kernelized LINREL algorithm: the polynomial kernel function and the Gaussian kernel function. For comparison we also used the linear kernel. More specifically, we used

$$\begin{aligned} k_{\text{lin}}(\mathbf{x}_1, \mathbf{x}_2) &= \mathbf{x}_1 \cdot \mathbf{x}_2, \\ k_{\text{pol}}(\mathbf{x}_1, \mathbf{x}_2) &= (\mathbf{x}_1 \cdot \mathbf{x}_2 + 1)^2, \\ k_{\text{exp}}(\mathbf{x}_1, \mathbf{x}_2) &= \exp \left\{ -\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2} \right\}. \end{aligned}$$

The results of the experiments are reported in Table 1, including as baseline p_{bas} a random selection of images. Besides the average precision we are also giving the improvement factor over random guessing, \bar{p}/p_{bas} . The results show that the choice of the kernel function does indeed influence the performance of the LINREL algorithm, and that the performance of the linear kernel can be exceeded. We expect that a more significant improvement can be achieved using a learned kernel.

5.2 Experiments with collages

Since the Gaussian kernel function performed best in the experiments reported in the previous section, we are using this kernel function for the experiments in this section. We are comparing the performance of the three selection methods for collages described in Section 4.4 with the performance of the selection mechanism of the PicSOM system. The collage size is set to $n = 15$ and $T = 10$ iterations are performed. The results in Table 2 show that LINREL outperforms PicSOM, in particular when the second selection method is used, i.e. when relatively little exploration is done. In Figure 1 we plot the precision (i.e. fraction of relevant images presented until iteration t) versus the number of iterations t for two categories, “airplane” and “boat”, using the second selection method. In the first

Table 1: Average precision of LINREL for the VOC'2007 training data using different kernel functions, $n = 1$, $T = 150$, $\mu = 1$, $c = 0.1$. The baseline p_{bas} is the total fraction of relevant images.

Category	p_{bas}	Average precision \bar{p} [%]			\bar{p}/p_{bas}		
		k_{lin}	k_{pol}	k_{exp}	k_{lin}	k_{pol}	k_{exp}
aeroplane	4.52	39.12	46.59	42.73	8.66	10.31	9.46
bicycle	4.88	8.58	10.63	7.42	1.76	2.18	1.52
bird	7.28	18.03	18.73	21.15	2.48	2.57	2.91
boat	3.48	23.10	22.48	21.01	6.64	6.46	6.04
bottle	6.12	9.73	7.18	11.96	1.59	1.17	1.96
bus	4.00	4.89	6.15	8.20	1.22	1.54	2.05
car	16.07	33.14	33.83	35.42	2.06	2.10	2.20
cat	6.64	12.58	15.52	15.19	1.90	2.34	2.29
chair	11.28	28.25	26.87	26.34	2.51	2.38	2.34
cow	2.84	10.02	6.95	6.50	3.53	2.45	2.29
diningtable	5.20	7.21	11.98	13.49	1.39	2.30	2.60
dog	8.40	12.25	12.52	14.85	1.46	1.49	1.77
horse	5.76	19.57	18.65	13.59	3.40	3.24	2.36
motorbike	4.92	8.96	7.76	9.16	1.82	1.58	1.86
person	36.15	46.98	48.27	49.13	1.30	1.34	1.36
pottedplant	6.12	8.48	7.31	8.99	1.39	1.19	1.47
sheep	1.96	11.95	7.13	8.78	6.10	3.64	4.48
sofa	7.52	12.55	14.14	13.55	1.67	1.88	1.80
train	5.12	14.13	13.74	15.60	2.76	2.68	3.05
tvmonitor	5.76	11.46	14.14	19.61	1.99	2.46	3.41
Average	7.70	17.05	17.53	18.13	2.78	2.77	2.86

iteration the precision equals the baseline, since the first collage is essentially selected randomly. For both categories the precision increases initially, as LINREL learns the relevant category. For category “airplane” the precision starts to drop after iteration 7, since only the more difficult images with airplanes are left. Eventually, after exhausting the whole database, the precision will drop back to the base line. For category “boat” learning still continues after iteration 10, but eventually the precision will also drop back to the base line.

6 Conclusion

We have developed the LINREL algorithm for CBIR based on relevance feedback. LINREL makes use of available side information to predict the relevance of images. It deals with the exploration-exploitation trade-off by using upper confidence values of the relevance estimates. The LINREL algorithm is able to accommodate arbitrary kernel functions as similarity measures of images. Initial experiments with standard kernels show that appropriate kernel functions do improve the performance of LINREL. We expect that kernel functions learned by methods developed in WP3, will further improve the performance.

The LINREL algorithm has also been adapted to select collages of images for presentation to the user. Several methods for selecting collages have been considered. In our experiments all methods improve over the selection mechanism of the original PicSOM system, and the method with most emphasis on exploitation performed best. This advantage of increased exploitation will be investigated further.

Table 2: Average precision of PicSOM and LINREL for the VOC’2007 training data, using different collage selection methods, with $n = 15$, $T = 10$, $\mu = 1$, $c = 0.1$. The baseline p_{bas} is the total fraction of relevant images.

Category	p_{bas}	Average precision \bar{p} [%]				\bar{p}/p_{bas}			
		PicSOM	Meth 1	Meth 2	Meth 3	PicSOM	Meth 1	Meth 2	Meth 3
aeroplane	4.52	31.18	34.39	31.36	33.20	6.90	7.61	6.94	7.35
bicycle	4.88	12.07	9.41	11.82	8.93	2.47	1.93	2.42	1.83
bird	7.28	20.33	16.61	17.76	17.82	2.79	2.28	2.44	2.45
boat	3.48	20.65	19.55	15.02	19.01	5.94	5.62	4.32	5.46
bottle	6.12	9.66	11.22	11.01	10.65	1.58	1.83	1.80	1.74
bus	4.00	7.49	7.58	9.26	7.97	1.87	1.90	2.32	1.99
car	16.07	24.12	33.55	33.27	32.68	1.50	2.09	2.07	2.03
cat	6.64	24.74	13.92	14.65	13.53	3.73	2.10	2.21	2.04
chair	11.28	20.46	26.07	26.83	25.82	1.81	2.31	2.38	2.29
cow	2.84	6.17	5.71	5.60	6.16	2.17	2.01	1.97	2.17
diningtable	5.20	12.29	11.31	15.78	13.17	2.36	2.18	3.04	2.53
dog	8.40	9.13	14.15	15.10	14.72	1.09	1.69	1.80	1.75
horse	5.76	11.30	16.96	19.90	14.23	1.96	2.95	3.46	2.47
motorbike	4.92	12.65	7.75	7.55	9.15	2.57	1.58	1.54	1.86
person	36.15	54.59	47.51	47.36	45.61	1.51	1.31	1.31	1.26
pottedplant	6.12	15.53	9.75	9.60	9.46	2.54	1.59	1.57	1.55
sheep	1.96	2.61	7.58	5.86	7.95	1.33	3.87	2.99	4.06
sofa	7.52	11.14	13.71	12.77	13.49	1.48	1.82	1.70	1.79
train	5.12	9.67	16.64	19.41	14.18	1.89	3.25	3.79	2.77
tvmonitor	5.76	9.32	13.89	15.58	15.27	1.62	2.41	2.71	2.65
Average	7.70	16.25	16.86	17.27	16.65	2.46	2.62	2.64	2.60

The LINREL algorithm has been implemented and made available in the PinView prototype system.

Acknowledgments

We thank Kitsuchart Pasupa for very helpful comments on an earlier version of this report, and Jorma Laaksonen for conducting experiments with the PicSOM system.

A Derivation of equation (7)

Omitting the subscripts t , we get from (1) that

$$\mathbf{X}^\top \mathbf{X} \hat{\mathbf{w}} + \mu \hat{\mathbf{w}} = \mathbf{X}^\top \mathbf{y}$$

and

$$\hat{\mathbf{w}} = \mathbf{X}^\top \mathbf{z}$$

where $\mathbf{z} = \mu^{-1}(\mathbf{y} - \mathbf{X} \hat{\mathbf{w}})$. Thus

$$(\mathbf{X} \mathbf{X}^\top + \mu \mathbf{I}) \mathbf{z} = \mu^{-1} \mathbf{X} (\mathbf{X}^\top \mathbf{y} - \mathbf{X}^\top \mathbf{X} \hat{\mathbf{w}}) + \mathbf{y} - \mathbf{X} \hat{\mathbf{w}} = \mu^{-1} \mathbf{X} (\mu \hat{\mathbf{w}}) + \mathbf{y} - \mathbf{X} \hat{\mathbf{w}} = \mathbf{y}$$

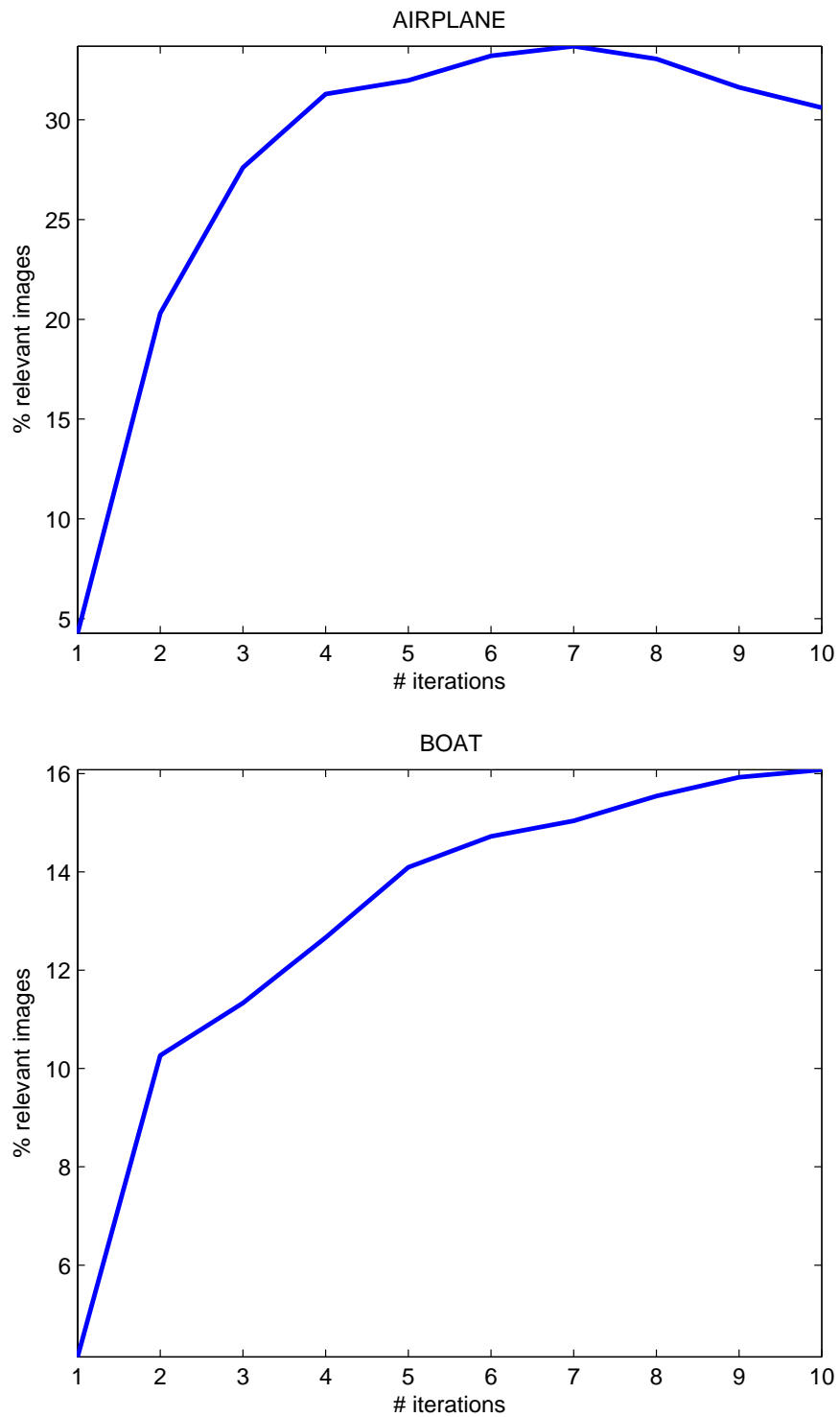


Figure 1: Fraction of relevant images presented until iteration t for two categories using LINREL with the second selection method for collages.

and

$$\hat{\mathbf{w}} = \mathbf{X}^\top \mathbf{z} = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \mu\mathbf{I})^{-1} \mathbf{y},$$

which gives equation (7).

References

- [1] R. 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] P. Auer, “Using Confidence Bounds for Exploration-Exploitation Trade-offs”. *Journal of Machine Learning Research*, 3:397–422, 2002.
- [3] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite time analysis of the multiarmed bandit problem”. *Machine learning*, 47(2-3):235–256, 2002.
- [4] P. Auer and A. Leung, “Models of Exploration-Exploitation Trade-offs”. PinView Deliverable D4.1, www.pinview.eu/deliverables, 2009.
- [5] V. Dani, T.P. Hayes, and S.M. Kakade, “Stochastic Linear Optimization under Bandit Feedback”. *Proc. 21st Ann. Conf. on Learning Theory*, pp. 355–366, 2008.
- [6] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 Results”. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop>, 2007.
- [7] Z. Hussain, K. Pasupa, C.J. Saunders, and J. Shawe-Taylor, “Basic metric learning”. PinView Deliverable D3.1, www.pinview.eu/deliverables, 2008.
- [8] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, “Self-organizing maps as a relevance feedback technique in content-based image retrieval”. *Pattern Analysis & Applications*, 4(2+3):140–152, 2001.
- [9] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan, “Learning the kernel matrix with semidefinite programming”. *Journal of Machine Learning Research*, 5:27–72, 2004.