Deliverable D3.1

Basic metric learning

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 1.03
Contractual date of delivery	31.12.2008
Actual date of delivery	31.12.2008
Deliverable number	D3.1
Deliverable title	Basic metric learning
Nature	report
Dissemination level	PU – Public
WP contributing to the deliverable	WP3 Learning the comparison metrics
Task contributing to the deliverable	Task 3.1 Definition of extended metric class
WP responsible	University College London
Task responsible	University College London
Editor	Zakria Hussain, <z.hussain@cs.ucl.ac.uk >
Editor address	Department of Computer Science, University College London, WC1E 6BT, United Kingdom
Authors in alphabetical order	Zakria Hussain, Kitsuchart Pasupa, Craig J. Saunders and John Shawe-Taylor
EC Project Officer	Pierre-Paul Sondag
Keywords	multiple kernel learning, 1-class support vector machine, content-based image retrieval with relevance feedback
Abstract	This report presents a a novel Multiple Kernel Learning (MKL) algorithm for the 1-class support vector machine. The emphasis is placed on viewing the CBIR task with relevance feedback as a metric learning problem, where each image has 11 different feature extraction methods applied to it. Our method attempts at finding the most compact ball amongst the 11 different feature representations using a novel 1- and 2-norm regularisation technique for the 1-class SVM under the MKL framework. We also devise a simple way of including the set of negative examples whilst still utilising the 1-class SVM implementation.

List of annexes

none

Contents

1 Overview

This is the first Deliverable of Work Package 3 of the *Personal Information Navigator Adapting Through Viewing*, PinView, project, funded by the European Community's Seventh Framework Programme under Grant Agreement n° 216529. The report constitutes the output of Task 3.1 *Definition of extended metric class.*

The description of work makes mention of two different types of activities for Task 3.1 and the associated deliverable D3.1. The first is concerned with the definition of the set of metrics and their extension to include information gleaned from eye tracking, while the second is focused on the main topic of the work package, 'Learning the Metric', and addresses the question of how linear combinations of basis metrics can be isolated as part of the learning process. The learned combination would replace the reweighting carried out implicitly by the PicSOM algorithm. The first topic proved more dependent on the output of the other work packages than anticipated as much of the eye tracking analysis is still in progress. Nonetheless we include a section describing some very preliminary results combining image features with eye-tracking movements to perform an image retrieval task. These results suggest that making use of the eye-tracking information will require careful modelling of the interaction between user and content, as a naive inclusion of the extra information does not improve performance. The main part of the deliverable addresses the second question investigating alternative kernel-based approaches to learning the metric. It is argued that this corresponds to learning the kernel and if we ignore negative examples the problem can be posed as multiple kernel learning for 1-class support vector machines (SVMs). We apply a standard framework taken from 2-class SVMs with disappointing results both in terms of the small number of kernels involved in the optimal solution as well as the quality of the resulting average precision. We extend the form of the regularisation to counter the over sparsification with positive results. This novel implementation can also be extended to 2-class SVMs with further improved performance. The work provides a framework for learning the metric that can be readily generalised to the large sets of kernels anticipated from other work packages.

The involvement of TKK in this Task has consisted of the preparation of a new programming interface in their PicSOM CBIR system that will be used in forthcoming on-line experiments in this Work package. MUL's involvement in this task was considering the similarity of images through different metrics. This will be continued in future deliverables of WP3. The involvement of SOTON-ECS was conducting the preliminary experiments of combining image features with eye-tracking movements in order to perform an image retrieval task. This has been included at the start of the report.

2 Learning from eye movements

Using the data collection gathered from the data collection campaign of D8.3, we provide some initial experiments based on a simple linear combination of a standard image metric (namely histograms) and features gained from the eye movements. These are used both to show that metric information based on eye movements can be useful, and to provide additional motivation for the use of more complex metric combination methods (such as those used in PicSOM and further outlined in Section 3 using support vector machines (SVMs)).

2.1 Transport task data

In this task (for full details see D8.3), users are shown 10 images on a page and they are asked to rank the top five images in order of relevance to the topic of "transport". Each page contains 1–3 clearly relevant images, 2–3 either borderline or marginally relevant images, and the rest are non-relevant images. The experiment has 30 pages which consist of 300 images from the Pascal Visual Objects Challenge 2007 database [6]. An example of each page is shown in figure 2.1.

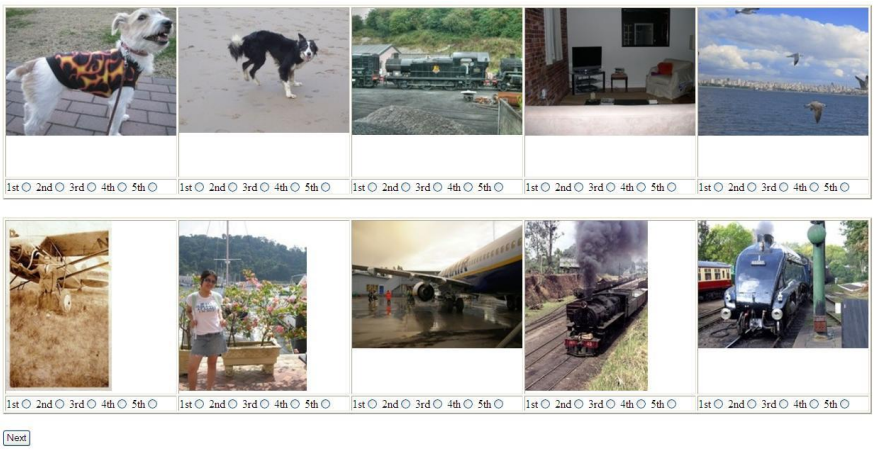

Figure 1: An Example of page. There are five clearly and one marginally relevant images.

The experiment was performed by seven different users. Eye movements were recorded by a Tobii X120 eye tracker which was connected to a PC with 19-inch monitor (resolution of 1280x1024). Any pages that contained less than five images with fixations (for example due to the subject moving and the eye-tracker temporarily losing track of the subject's eyes) were discarded.

2.2 Feature extraction

In these experiments we use standard image histograms and also features obtained from the eye-tracking. The task is then to predict relevant images based on individual image or eye-track features only, or simple combinations including a basic linear sum and using histograms from sub-parts of an image in which the user focussed. First let us discuss the features obtained from the output of the eye-tracking device.

2.2.1 Eye movement (EYE)

Feature vectors are pre-processed from eye gaze and fixation information from each image, using the standard ClearView fixation filter provided with the Tobii eye-tracking software. This is a standard approach to obtaining features (c.f.[8]) where the fixation threshold was set at 100 ms and a 30 px radius[1]. The features are listed in table 1, and in practice each feature is normalised by a total value of that features from all images in the same page.

Table 1: Eye movement extracted features.

#	Feature
1	Number of fixations
2	Length of fixations
3	Average length of fixations
4	Length of maximum fixation
5	Length of last fixation
6	Number of time looking at image (based on fixation data)
7	Length of gaze point
8	Number of time looking at image (based on gaze data)

2.2.2 Histogram Image Features

As a baseline for simple image features we used a 256-bin grayscale histogram as image-only features. We also however produced histograms on sub-parts of an image which corresponded to areas on which the user fixated – thus enabling an eye-driven combination of features. Each image is divided into five segments: one horizontal split, one vertical split, and one intersection of four segments as shown in figure 2.2.2. The feature vector is therefore a combination of five 256-bin grey scale histograms. Any segment which has no gaze information from the user is set to zero, thus incorporating both image and eye-movement features.

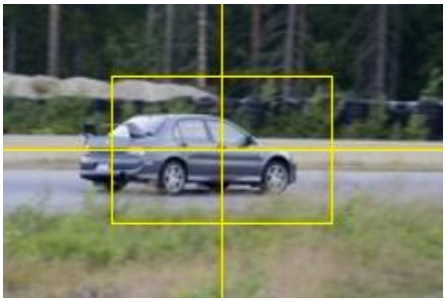

Figure 2: Each image is divided into five segments.

2.3 Experimental results

Experimentation has been carried out on the above features. To evaluate the task, we applied two methods which are linear discriminant analysis (LDA) and support vector machine for

[1]These are also the settings recommended for media with mixed content [16]

optimising mean average for precision (SVM^{MAP}).

- Linear discriminant analysis: the task will be considered as 2-class classification problem (relevant/non-relevant). Images will be ranked in descending order based on the computed score. The higher the score, the more likely an image is relevant. The top five ranked images are predicted by ordering the scores output by the classifier and labeling the top five scores appropriately.
- Support vector machine for optimising mean average for precision (SVM^{MAP}): The algorithm was first proposed to tackle the ranking problem by [19]. It is a structural SVM which directly optimises mean average for precision, and will have the full ranking information given by each user available in order to predict a ranking.

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

In order to compare results, we used the ranking performance measure of normalised discount cumulative gain (NDCG). It can be computed by,

$$\mathrm{NDCG}_k(y,q) = \frac{1}{N_q}\sum_{i=1}^{k} D(y_i)\varphi(g_{qi}) \tag{1}$$

with $D(r) = \frac{1}{\log_2 1+r}$ and $\varphi(g) = 2^g - 1$, where q is a page number, r is rank position, k is a truncation level, here $k = 5$, N is a normalised constant which makes the perfect ranking (based on g_{qi}) equal to one, and g_{qi} is the categorical grade, here grade is equal to 5 for 1$^{\text{st}}$ rank and 0 for non-relevant.

We found that although the topic was left deliberately vague, and that some images were chosen specifically to be ambiguous the amount of agreement in the rankings between users was large. In order to test the model, we used a leave-one-page-out cross validation approach. There is a separate model for each user, and the parameter of SVM^{MAP} was also set via cross-validation. The experimental results of LDA and SVM^{MAP} are shown in Table 2 and 3, respectively. The NDCG_5 value when the ranks are selected randomly is 0.36, so even using the simple baseline methods presented here we are able to achieve higher performance.

Table 2: NDCG Value from LDA and number of page used for each user.

User	nPage	EYE	HIST	HIST5	EYE+HIST	EYE+HIST5
1	30	0.4080	0.3962	0.4092	0.4128	0.3624
2	29	0.3923	0.4118	0.4243	0.4208	0.4250
3	30	0.5084	0.4016	0.5262	0.3998	0.4929
4	19	0.5304	0.3849	0.4494	0.4415	0.4420
5	2	0.2954	0.2889	0.7017	0.2405	0.7017
6	21	0.4543	0.3267	0.3712	0.2964	0.3408
7	30	0.4506	0.3444	0.4703	0.3523	0.4638
Average	161	0.4509	0.3786	0.4485	0.3866	0.4277

From the tables one can observe that in general using information from eye movements is often better than classifying purely based on image histograms. Although this is not always the case, the histogram approach may be slightly misleading in that transport images often

Table 3: NDCG Value from SVM^{MAP} and number of page used for each user.

User	nPage	EYE	HIST	HIST5	EYE+HIST	EYE+HIST5
1	30	0.4304	0.4998	0.5787	0.4895	0.5792
2	29	0.4534	0.4614	0.4635	0.4474	0.4290
3	30	0.5106	0.5162	0.5472	0.5266	0.5572
4	19	0.5590	0.4936	0.4863	0.5436	0.5063
5	2	0.5736	0.3012	0.8508	0.3012	0.8508
6	21	0.4529	0.4897	0.5493	0.4814	0.5489
7	30	0.4700	0.5005	0.5157	0.4832	0.5067
Average	161	0.4768	0.4916	0.5290	0.4906	0.5254

contain a large portion of sky (as they are often taken outside), and for more complex tasks this metric is likely to not be so suitable. Simply breaking up the image histogram into the five segments and only using those areas which the user looked at (HIST5) nearly always increases performance against whole-image histograms (and in many cases this increase is very large). This gives some evidence that eye-movement information could potentially be used to guide image features. The results from linearly combining the eye-movement and histogram-based features are less conclusive. It is likely that the small number of fixation-based features are dominated by the histogram counterparts and more advanced methods of combining them (and other metrics) would lead to increased performance. In general however, these simple initial results on the data from D8.3 provide a baseline result which shows the potential advocacy of both using eye-movements and learning image metrics beyond the simple histogram and 5-region features considered here. We now turn our attention to the main focus of task 3.1: basic metric learning (kernel learning).

3 Multiple Kernel Learning with 1-class SVM

Content-Based Image Retrieval (CBIR) looks for relevant images based on the contents of images, such as for example, colour, shape, texture, *etc.*[5]. The development of CBIR systems has become a major area of research in the image retrieval, information science and machine learning communities. The main idea is to extract relevant images from a query based only on the content of the image rather than its label or other associated information such as the surrounding text. CBIR systems broaden image search to unlabelled corpora but the task of selecting relevant images is rendered correspondingly more demanding.

We will consider a specific type of CBIR that is based on user feedback. Presented with an image or set of images the user indicates which are relevant. This feedback is used to drive the CBIR system in what we will refer to as Relevance-Feedback Content-Based Image Retrieval. Given an image obtained the user can indicate how relevant the image is. This relevance may be a discrete yes or no, or could be a larger range of relevances such as *e.g.*, highly relevant, moderately relevant, highly irrelevant, *etc.*. In this report we only consider the case where the user's feedback is a yes or no (*i.e.*, relevant or irrelevant).

Recently PicSOM [9] has been proposed as a CBIR system that can use Relevance-Feedback. The system makes use of Self-organising maps (SOMs) that map feature vectors extracted from images into 2-dimensional grids that preserve the local geometry of the feature vectors. The system then looks for regions of the SOM where the density of relevant images is high. In this report we apply the 1-class SVM in place of the SOMs in order to identify similar regions of the feature space. Our problem setting is the following. We are given several thousand images and ask users to identify relevant images. The number of relevant

images is typically very small when compared to the number of irrelevant images. We then use the 11 different feature extraction methods considered in PicSOM [18], including SIFT, Histogram, colour, texture, *etc.*. While PicSOM generates a SOM for each type of feature using all of the available images, we use the features to construct 11 kernel functions using a Gaussian kernel over the corresponding feature vector.

Given these 11 different kernels we consider only the relevant images to be useful in distinguishing the important characteristics for the task at hand. For instance, some images may contain cats and we would ask the user to identify when they have seen a cat in an image. All images without cats would be considered irrelevant by the user. Therefore, we can use only the positive (relevant) images in a 1-class SVM to construct the tightest ball enclosing most of the relevant images and almost none of the irrelevant ones. This is the route proposed by Chen *et al.*[3] though their method uses a single kernel which in this case will be the sum of the 11 kernels.

The main contribution of this report is the observation that in different CBIR searches different features are important (*e.g.*colour histograms for sunsets, colour histograms and texture for sunsets over snowscapes, *etc.*). In order to improve the focus of a search it is important to learn a weighted combination of the features that emphasises those critical for the search. PicSOM does not do this explicitly but its use of a density map on the feature SOMs results in areas where relevant images are concentrated receiving higher scores. In contrast we will include adaptation of a weighting of the features within the learning process using so-called Multiple Kernel Learning [10, 2, 1].

We therefore propose a method for Multiple Kernel Learning in the 1-class case, where a weighted combination of kernels is found that finds the tightest ball around the relevant images. In our initial experiments the optimisation proposed for 2-class SVMs in [1] results in single kernels being selected in the 1-class case, hence we adapt the objective to create a non-trivial combination of features. The weights obtained tell us what aspects (features) of images make them relevant, because we have learnt a specific weighting based on the search images.

The SOM implements an embedding into a 2-dimensional space, while the 1-class SVM searches for an enclosing sphere in a very high dimensional space (in the case of the Gaussian kernel that we use, the space is in fact an infinite-dimensional Hilbert space). Despite the high dimensionality we will see that the regularisation implicit in both learning the 1-class and the combination weightings enable us to obtain excellent results. We leave open the possibility that other dimension reduction techniques could be combined with our approach in future work.

3.1 Background

The PicSOM method [9] is based on building SOMs for each of the 11 different features extracted from the images. These are listed in Table 4 and are described in detail in [18]. In creating the SOMs the complete repository of (unlabelled) images is used but once computed the SOMs can be used for all types of searches.

Each SOM is a grid of 64×64 points each of which has an associated feature vector at the end of the learning phase. The SOM algorithm enforces that nodes close in the grid have similar feature vectors.

New images can be mapped onto the SOM by finding the node whose feature vector is most similar to that of the image. When performing a search the m^+ images identified as relevant are mapped onto each of the 11 SOMs as are the m^- irrelevant images. For each SOM each relevant image contributes a score of $1/m^+$ to its node, while an irrelevant image contributes $-1/m^-$. A low pass filter is then applied to each SOM resulting in a score for each of its nodes.

Feature	dimensions
DCT coefficients of average colour in rectangular grid	12
CIE L*a*b* colour of two dominant colour clusters	6
Histogram of local edge statistics	80
Haar transform of quantised HSV colour histogram	256
Histogram of interest point SIFT features	256
Average CIE L*a*b* colour	15
Three central moments of CIE L*a*b* colour distribution	45
Histogram of four Sobel edge directions	20
Co-occurrence matrix of four Sobel edge directions	80
Magnitude of the 16×16 FFT of Sobel edge image	128
Histogram of relative brightness of neighbouring pixels	40

Table 4: Visual features extracted from images

We can now score a new image by reading off the score its node has in each of the feature SOMs and summing these values. Higher values indicate higher relevance. Note that the features are not explicitly weighted in PicSOM, but that regions of a feature SOM where a large number of relevant (and few irrelevant) images are mapped will have a high score, hence giving the feature a bigger impact on the overall score of the image. This implies an implicit weighting of the features.

3.1.1 1-class support vector machines

Let $\mathbf{w} \in \mathbb{R}^n$ be the weight vector and $C \in \mathbb{R}$ the parameter controlling the number of mistakes made. We define the feature mapping $\phi : \mathbf{x} \mapsto \mathcal{F}$ that maps the input $\mathbf{x} \in \mathbb{R}^n$ to a higher dimensional space $\mathcal{F}$, known as the *feature space*. The primal formulation of the one-class support vector machine (SVM) [14, 15] can be given as:

$$\begin{array}{ll} \min_{\mathbf{w},\xi} & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\,\|\xi\|_1 \\ \text{subject to} & \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle \geq 1 - \xi_i \\ & \xi_i \geq 0, \;\; i = 1, \ldots, m. \end{array}$$

It is well-known that the dual of this optimisation problem can be expressed in terms of the kernel function $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z})\rangle$ as

$$\begin{array}{ll} \max_{\boldsymbol{\alpha}} & W(\boldsymbol{\alpha}) = \sum_{i=1}^m \alpha_i - \sum_{i,j=1}^m \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \\ & 0 \leq \alpha_i \leq C, \;\; i = 1, \ldots, m \end{array}$$

with the corresponding evaluation function $\langle \mathbf{w}, \phi(\mathbf{x})\rangle = \sum_{i=1}^m \alpha_i \kappa(\mathbf{x}_i, \mathbf{x})$. The one-class SVM is a good candidate algorithm for novelty detection as for normalised data (when using a Gaussian kernel the data is automatically normalised in the feature space) it finds the weight vector $\mathbf{w}$ that defines the centre of the smallest enclosing ball containing most of the training points. The C parameter controls the number of mistakes that may be made and so also indirectly controls the size of the ball. A test point is considered novel if it does not fall close to the ball *i.e.*, it falls outside the estimate for the support of the distribution.

3.1.2 Multiple Kernel Learning

Now let us assume that we have a set of kernel functions $\mathbb{K} = \{\kappa_1, \ldots, \kappa_K\}$. Given a vector $\mathbf{z} = (z_1, \ldots, z_K)$ of coefficients and kernel functions $\kappa_i(\cdot, \cdot)$ for $i = 1, \ldots, K$, we can define the

following linear combination of kernel functions:

$$\kappa_{\mathbf{z}}(\cdot,\cdot) = \sum_{k=1}^{K} z_k \kappa_k(\cdot,\cdot).$$

In our case the kernels will correspond to the 11 feature extraction methods and we can think of the vector $\mathbf{z}$ as weighting the features. The aim of multiple kernel learning is to learn the weighting $\mathbf{z}$ of the kernels at the same time as the 1-class SVM in the feature space corresponding to $\kappa_{\mathbf{z}}$. In the sequel we follow an approach similar to that proposed in [2] and [1] for SVMs but with a novel twist.

3.2 Multiple Kernel Learning for 1-Class SVM

We first derive the algorithm and then give pseudo code in subsection 3.2.1. We propose to constrain a convex combination of the 2-norm and the 1-norm of the 2-norms of the weight vectors in each kernel's feature space (assuming data normalised in each space):

$$\begin{array}{ll} \min_{\mathbf{w}_k,\xi} & \frac{\mu}{2}\left(\sum_{k=1}^{K}\|\mathbf{w}_k\|_2\right)^2 + \frac{1-\mu}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|_2^2 + C\|\xi\|_1 \\ \text{subject to} & \sum_{k=1}^{K}\langle \mathbf{w}_k, \phi_k(\mathbf{x}_i)\rangle \geq 1-\xi_i \\ & \xi_i \geq 0, \;\; i=1,\ldots,m. \end{array}$$

When $\mu \to 0$ then we retrieve the 2-norm regularisation and use the full set of kernels. In the case that $\mu \to 1$ we move towards the 1-norm solution and a sparser set of kernels in the final combination.

In order to keep the document in a more digestible form we have placed most of the derivations of converting the above primal into its dual form in the Appendix. However, we need the following definitions (see Appendix) in order to state the dual optimisation problem of our MKL formulation. For k satisfying $z_k \neq 0$

$$\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i,\mathbf{x}_j) = (\mu + (1-\mu)z_k)^2 D^2, \tag{2}$$

where $D = \frac{\sum_{k\in J}\sqrt{\beta_k}}{1-\mu+\mu|J|}$ and $J = \{k : z_k \neq 0\}$, that is the set of indices k, for which

$$\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i,\mathbf{x}_j) > \mu^2 D^2, \tag{3}$$

since if $\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i,\mathbf{x}_j) < \mu^2 D^2$ then Equation (2) cannot hold.

The resulting dual optimisation is:

$$\begin{array}{ll} \max_{\boldsymbol{\alpha}} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{A}{2}\sum_{k\in J}\beta_k + \frac{B}{2}\left(\sum_{k\in J}\sqrt{\beta_k}\right)^2 \\ \text{subject to} & \beta_k = \sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j) \\ & 0 \leq \alpha_i \leq C, \;\; i=1,\ldots,m \end{array}$$

where $A = 1/(1-\mu)$ and $B = ((|J|-1)\mu^2+\mu)/((1-\mu)(1-\mu+\mu|J|)^2)$.

We now consider performing coordinate-wise descent in the α vector. Writing

$$g_i(\alpha_i) = \frac{\partial W(\alpha)}{\partial \alpha_i},$$

where α_i is the i-th coordinate of α in the argument of $W(\cdot)$, we seek the solution of $g_i(\alpha_i) = 0$ as the new value for α_i. We expand $g_i(\alpha_i)$ in a Taylor series around the current values α^0:

$$g_i(\alpha_i) \approx \frac{\partial W(\alpha^0)}{\partial \alpha_i} + \frac{\partial W^2(\alpha^0)}{\partial \alpha_i^2}(\alpha_i - \alpha_i^0) = 0$$

and solve for α_i.

First we compute the partial derivatives:

$$\frac{\partial W(\alpha^0)}{\partial \alpha_i} = 1 - A\sum_{k=1}^{K} f_k(\mathbf{x}_i) + B\left(\sum_{k\in J}\sqrt{\beta_k}\right)\sum_{k\in J}\frac{f_k(\mathbf{x}_i)}{\sqrt{\beta_k}}$$

where $f_k(\mathbf{x}) = \sum_{j=1}^{m}\alpha_j\kappa_k(\mathbf{x},\mathbf{x}_j)$ and

$$\frac{\partial W^2(\alpha^0)}{\partial \alpha_i^2} = -A\sum_{k=1}^{K}\kappa_k(\mathbf{x}_i,\mathbf{x}_i) + B\left(\sum_{k\in J}\sqrt{\beta_k}\right)\sum_{k\in J}\frac{\beta_k\kappa_k(\mathbf{x}_i,\mathbf{x}_i) - f_k(\mathbf{x}_i)^2}{\beta_k^{3/2}} + B\left(\sum_{k\in J}\frac{f_k(\mathbf{x}_i)}{\sqrt{\beta_k}}\right)^2.$$

Finally we update α_i using the equation

$$\alpha_i = \alpha_i^0 - \frac{\frac{\partial W(\alpha^0)}{\partial \alpha_i}}{\frac{\partial W^2(\alpha^0)}{\partial \alpha_i^2}}.$$

3.2.1 Algorithm

The training procedure for the MKL for 1-class SVM is described in Algorithm 1 given below.

Input: set of kernel matrices $\{\mathbf{K}_1,\ldots,\mathbf{K}_K\}$, $\boldsymbol{\alpha}^0$ vector to zero with one element, say $\alpha_1^0 > 0$, μ and C.

Output: decision function of the form $f(\mathbf{x}) = \sum_{i=1}^{m}\alpha_i\sum_{k\in J}\frac{z_k}{\mu+(1-\mu)z_k}\kappa_k(\mathbf{x}_i,\mathbf{x})$

1: **repeat**

2: compute update rule for each component of $\boldsymbol{\alpha}$ using:

$$\alpha_i = \alpha_i^0 - \frac{\frac{\partial W(\alpha^0)}{\partial \alpha_i}}{\frac{\partial W^2(\alpha^0)}{\partial \alpha_i^2}}.$$

3: update

$$z_k = \frac{1}{1-\mu}\left(\frac{\sqrt{\beta_k}}{D} - \mu\right),$$

where $z_k = 0$ if $z_k < 0$.

4: update

$$D = \frac{\sum_{k\in J}\sqrt{\beta_k}}{1-\mu+\mu|J|}$$

where J can be found using Equation (3)

5: **until** $\|\boldsymbol{\alpha}^n - \boldsymbol{\alpha}^{n-1}\|_2 < \epsilon$, where ϵ is a small positive real number

Algorithm 1: Algorithm for Multiple Kernel Learning with 1-class Support Vector Machine (MKL 1-class SVM).

3.3 Experiments

The features that we used can be found in [18]. We list them in Table 4 with an extra feature extraction method known as Histogram of interest point SIFT (row 5). These interest points were first detected with a Harris-Laplace detector [13], then a histogram formed of the SIFT

descriptors [12] – based on local gradient orientation – of the interest points. The histogram bins were chosen by clustering the SIFT descriptors of all the interest points in the training images with the Linde-Buzo-Gray (LBG) algorithm [11]. We limit ourselves to use histograms with 256 bins. We do not describe the remaining feature extraction methods, as descriptions of these can be found in [18] and references therein.

We use the VOC2007 challenge database [7] which contains 9963 images, each with at least 1 object. The number of objects in each image ranges from 1 to 20, with, for instance, objects of people, sheep, horses, cats, dogs *etc.*. For a complete list of the objects, and description of the data set see the VOC2007 challenge website: `http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`.

The challenge organisers split the data into training, validation and testing sets. We only use the training (2501 images) and testing splits (4952 images) and ignore the validation splits in these experiments. A further experiment is conducted with users, who are asked to state whether an image is relevant or not for a given task. The relevant images in each training set are then used in our training phase. We use the relevance feedback results from cats, dogs and cows. We can view the relevance information of the training set as a supervised learning task. Where relevant images are viewed as positive examples and irrelevant images as negatives. However, in the 1-class SVM case we make use of only the relevant images.

The first experimental results are given in Figure 3. The results are only given for cat objects. We had 166 relevant images (*i.e.*, cats) in the training set of size 2501 and trained the MKL 1-class SVM on the 11 feature extraction methods from Table 4 using a Gaussian kernel[2] – hence giving us 11 kernels to learn. After training we test the resulting function by calculating the decision functions for test points and ordering them in descending order and calculating the Recall-Precision of each of the relevant test images retrieved. Throughout the experiments we fix the value of $C = 0.01$ as these worked well during training. The number of relevant test images was 332 from a test set of size 4952. The top plot of Figure 3 depicts the MKL 1-class SVM proposed in this report for varying values of μ between 0 and 1 with increments of 0.1. The x-axis shows the different values of μ and the y-axis shows the number of kernels used for a particular value of μ. As $\mu = 0$ we retrieve a 2-norm regularisation of the MKL problem and hence all of the kernels in the final linear combination. In contrast when $\mu = 0.7$ (and onwards) we retrieve the sparsest solution with only 1 kernel being chosen. For values of μ between 0 and 0.7 we find between 11 to 1 (inclusive) kernels in the final convex combination. Note that when 1 kernel is chosen through a particular choice of μ then any larger values of μ will not change the solution of the algorithm. This is not the case for μ values for which more than 1 kernel is chosen.

The bottom plot of Figure 3 depicts three methods: the MKL 1-class SVM proposed in this report, PicSOM [9] and 1-class SVM resulting from the method of Chen *et al.* [3]. The method of [3] only used a single 1-class SVM in order to carry out image retrieval. However, given that we have 11 different features, the Chen *et al.*method would need to concatenate all of these features and construct a single kernel and carry out 1-class SVM learning, which is equivalent to our MKL formulation of the 1-class with $\mu = 0$. This corresponds to using an unweighted combination of all the kernels. The plot gives the Average-Precision 20 (AP20) which is the Average-Precision for the first 20 relevant images retrieved from the test set. As we can see the SVM (Chen *et al.*) method yields the poorest results with PicSOM fairing slightly better at a retrieval rate of 0.25. However, as we vary μ (*i.e.*, $\mu \geq 0.3$) and use a smaller number of kernels the MKL 1-class SVM method outperforms both methods. This is quite surprising with respect to PicSOM as it utilises information from both classes: relevant and irrelevant. However, the results for Average Precision using all of the test examples was

[2]We calculated the Gaussian width parameter of each kernel by calculating the distance between relevant and irrelevant images, sorting the result in descending order and choosing the ith position for the width parameter. This heuristic seemed to work well for our experiments.

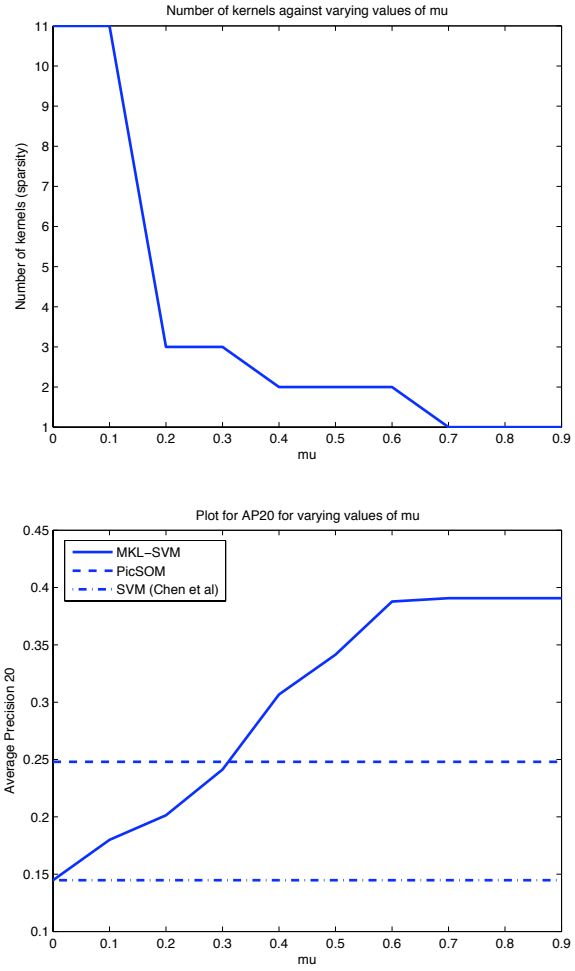

Figure 3: VOC2007 database retrieval results using 1-class SVM for AP20. Top: number of kernels found in the multiple kernel learning algorithm using 1-class SVM for varying values of $\mu \in [0, 1)$. Bottom: average precision 20 (AP20) retrieval rate on test images containing cats, using Multiple Kernel Learning with 1-class SVM, PicSOM and 1-class SVM (Chen et al).

disappointing and did not perform as well as those of AP20. Therefore, we now carry out further experiments that also utilise both relevant and irrelevant images in order to better mimic the behaviour of PicSOM and to help improve the results.

In order to add the additional information of negative images into our MKL formulation and take account of the PicSOM re-weighting scheme we carry out the following procedures: 1) by negating the kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ iff $y_i \neq y_j$ we can take into account the negative (irrelevant image) information without the need to derive a 2-class version of our 1-norm 2-norm variant for MKL. This change of sign results in solving the corresponding 2-class SVM problem [17, 4, 15]; 2) by using C^+ and C^- for positive and negative examples, respectively, we can upper bound the α_i for $y_i = +1$ and $y_i = -1$ separately according to the label of the image. We can re-weight the images in a similar way to PicSOM by fixing $C^+ = 0.01$ (as we do throughout the experiments) but setting $C^- = \frac{m^+}{m^-} C^+$ where m^+ is the number of relevant images and m^- the number of irrelevant images used in training.

In order to mimic real world CBIR tasks, where the number of irrelevant images is much larger than the number of relevant, we choose to use twice the number of relevant images.

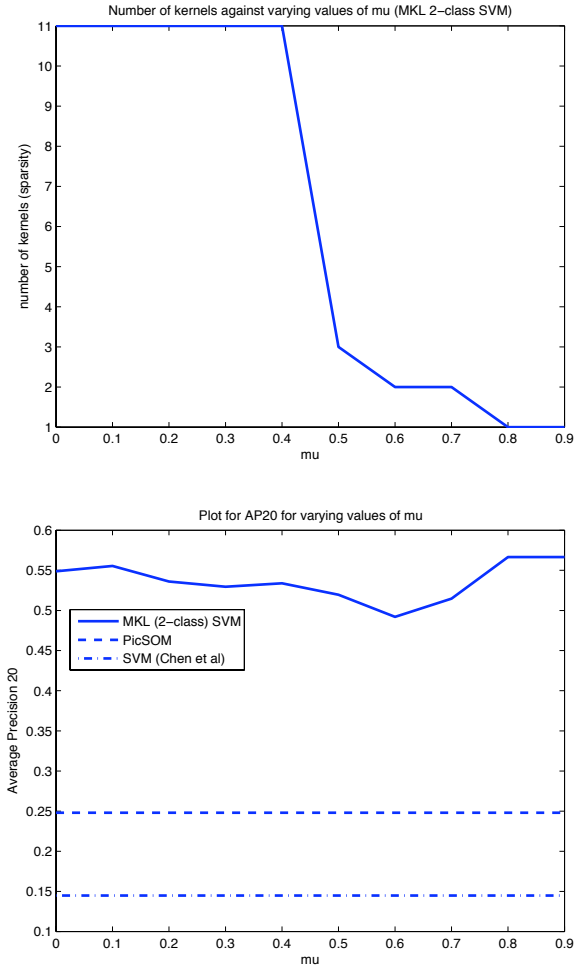

Figure 4: VOC2007 database retrieval results using 2-class SVM for AP20. Top: number of kernels found in the multiple kernel learning algorithm using 2-class SVM for varying values of $\mu \in [0, 1)$. Bottom: average precision retrieval rate on test images containing cats, using Multiple Kernel Learning with 2-class SVM, PicSOM and 1-class SVM (Chen et al).

Hence, in the cat data set containing 166 relevant images we randomly choose 332 irrelevant images from the remaining examples in the training set in order to train the MKL 1-class SVM, which when using the two procedures from above will be called the MKL 2-class SVM.[3] The results are given in Figure 4 with the top and bottom plots being analogous to the top and bottom plots of Figure 3 for the 1-class scenario. It is clear that the 2-class version outperforms both PicSOM and the 1-class SVM method of [3] by at least 25%.

We now fix μ to an intermediary value of 0.5 to conduct the remaining experiments. We now also use the cow objects, for which there are 77 relevant images in the training set and where we randomly find 144 non-relevant images for experiments to be conducted with the MKL 2-class SVM. The number of relevant test images to be retrieved for cow was 127 from 4952 test images. Similarly for dog objects there were 210 relevant images and 420 irrelevant images during training. The number of relevant images in testing was 433 from a total of 4952. Figure 5 shows Recall-Precision curves for MKL 2-class SVM, MKL 1-class SVM ($\mu = 0.5$), MKL 1-class SVM ($\mu = 0$) and PicSOM. Earlier the results reported for AP20

[3]Note that although this is equivalent to a 2-class SVM we use the 1-class formulation of the MKL to solve it (see Algorithm 1).

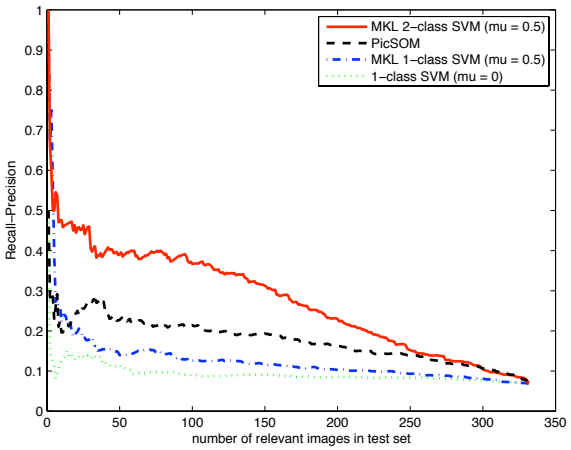

Figure 5: VOC2007 database: Recall-Precision curve for all test images containing cats, using Multiple Kernel Learning with 2-class SVM, PicSOM and 1-class SVM ($\mu = 0.5$) and 1-class SVM ($\mu = 0$) corresponding to Chen *et al.*.

with the MKL 1-class SVM were superior to those of PicSOM. However as the full set of images is used for ranking it is clear that the MKL 1-class SVM ($\mu = 0$) is the worst method followed by the improvement we gave using a weighted combination of kernels ($\mu = 0.5$) constructed from relevant images alone. PicSOM seems to fair better but the MKL 2-class SVM method outperforms all of the techniques and suggests that using irrelevant images in Relevance-Feedback CBIR can yield good results. Our final experiments are given in Table 5 with results for cat objects, cow objects and dog objects found in images, using the Average Precision 20 (AP20) and Average Precision 50 (AP50) measures. It is clear that the MKL 2-class SVM outperforms all other methods with dog and cat being the stand out results. We show the AP20 and AP50 results because we believe that the important feature of an image retrieval system is how well it performs at the start (*i.e.*, top of the ranked list). For instance when searching for a particular item one is usually interested in the first few hits and not those further down the ranked list. We also show the number (#) of kernels used for the MKL SVM methods, implying that several different features are needed in order to learn the relevance of images. The most common features found by both our methods were colour histogram and SIFT.

4 Conclusions

We have presented two sets of results. One where we carried out image retrieval using Relevance Feedback gleaned from eye movements. The results were inconclusive and demonstrated the need for more advanced techniques than the simple combination of image features and eye movement features presented in this report. The second set of results were concerned with basic metric learning. We only used the image feature data in order to learn the appropriate weighted combination of metrics that best capture the information required for successful image retrieval. We proposed a novel Multiple Kernel Learning algorithm for the 1-class support vector machine and presented encouraging results against the PicSOM CBIR system. However, the results for a full Recall-Precision curve were disappointing when we only made use of the relevant (1-class) images. Hence, when we turned our attention to using irrelevant information (and a similar rescaling technique used in PicSOM) we saw a dramatic increase in performance, outperforming the PicSOM system.

Object	MKL 2-class SVM			MKL 1-class SVM ($\mu = 0.5$)			1-class SVM ($\mu = 0$)			PicSOM	
	AP20	AP50	# kernels	AP20	AP50	# kernels	AP20	AP50	# kernels	AP20	AP50
Cat	**0.5196**	**0.4561**	3	0.3415	0.2361	2	0.1448	0.1333	11	0.2480	0.2518
Cow	**0.2863**	**0.2010**	5	0.1674	0.1394	3	0.1362	0.1158	11	0.2537	0.1999
Dog	**0.3653**	**0.3580**	11	0.1095	0.1297	2	0.1061	0.1151	11	0.2819	0.2778

Table 5: AP20 ad AP50 results for Cat, Cow and Dog objects using $\mu = 0.5$ and $C^+ = 0.01$. The 2-class SVM was trained using the 1-class implementation with the kernel entries of examples with opposite labels being negated.

For future work we would like to incorporate the eye movement data with the image feature information and carry out metric learning, to see if any advantage may be gained from more complex combinations of these two feature sets. We would also like to replace the

SOM algorithm in the PicSOM system with the MKL 2-class SVM algorithm and run more natural CBIR experiments in an on-line (adaptive) learning setting. Finally, an attempt at using the MKL algorithm to help provide information to the Exploration-Exploitation phase of learning is also an important direction to pursue.

Acknowledgements

We would like to thank Jorma Laaksonen for his help with the PicSOM system and experimental issues and Samuel Kaski for reviewing the report and suggesting improvements.

Appendix

In this Appendix we describe the derivation of the dual of the following primal problem that forms the basis of the report:

$$\begin{array}{ll} \min_{\mathbf{w}_k,\xi} & \frac{\mu}{2}\left(\sum_{k=1}^{K}\|\mathbf{w}_k\|_2\right)^2+\frac{1-\mu}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|_2^2+C\|\xi\|_1 \\ \text{subject to} & \sum_{k=1}^{K}\langle\mathbf{w}_k,\phi_k(\mathbf{x}_i)\rangle\geq 1-\xi_i \\ & \xi_i\geq 0,\ \ i=1,\ldots,m \end{array}$$

We form the Lagrangian to obtain the dual optimisation:

$$C\sum_{i=1}^{m}\xi_i-\sum_{i=1}^{m}\alpha_i\left[\sum_{k=1}^{K}\langle\mathbf{w}_k,\phi_k(\mathbf{x}_i)\rangle-1+\xi_i\right]+\frac{\mu}{2}\left(\sum_{k=1}^{K}\|\mathbf{w}_k\|_2\right)^2+\frac{1-\mu}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|_2^2-\sum_{i=1}^{m}\beta_i\xi_i.$$

Differentiating with respect to the primal variables and letting $D=\sum_{k=1}^{K}\|\mathbf{w}_k\|_2$:

$$\begin{aligned} \frac{\partial L(\mathbf{w}_k,\alpha,\xi)}{\partial\mathbf{w}_k} &= -\sum_{i=1}^{m}\alpha_i\phi_k(\mathbf{x}_i)+\left(\frac{D\mu}{\|\mathbf{w}_k\|_2}+(1-\mu)\right)\mathbf{w}_k=\mathbf{0}; \\ \frac{\partial L(\mathbf{w}_k,\alpha,\xi)}{\partial\xi_i} &= C-\alpha_i-\beta_i=0, \end{aligned}$$

giving the constraints $0\leq\alpha_i\leq C$. Letting $z_k=\|\mathbf{w}_k\|_2/D$, we obtain

$$\mathbf{w}_k=\frac{z_k}{\mu+(1-\mu)z_k}\sum_{i=1}^{m}\alpha_i\phi_k(\mathbf{x}_i)\ \text{with}\ \sum_{k=1}^{K}z_k=1.$$

Taking the inner product of the first equation with itself we obtain:

$$\|\mathbf{w}_k\|_2^2=\frac{z_k^2}{(\mu+(1-\mu)z_k)^2}\sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j)$$

implying for k satisfying $z_k\neq 0$

$$\sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j)=(\mu+(1-\mu)z_k)^2D^2. \tag{4}$$

Let $J=\{k:z_k\neq 0\}$, that is the set of indices k, for which

$$\sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j)>\mu^2D^2,$$

since if $\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j) < \mu^2 D^2$ then Equation (4) cannot hold. It follows that

$$(1-\mu)^2 D^2 \sum_{i=1}^{K} z_k^2 = \sum_{k \in J} \sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j) - |J| \mu^2 D^2 - 2\mu(1-\mu) D^2.$$

Substituting into the Lagrangian we obtain

$$\begin{aligned}
&\sum_{i=1}^{m} \alpha_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j \left[\sum_{k=1}^{K} \frac{z_k}{\mu + (1-\mu) z_k} \kappa_k(\mathbf{x}_j, \mathbf{x}_i) \right] + \frac{\mu D^2}{2} \left(\sum_{k=1}^{K} z_k \right)^2 + \frac{(1-\mu) D^2}{2} \sum_{k=1}^{K} z_k^2 \\
&= \sum_{i=1}^{m} \alpha_i - D^2 \sum_{k=1}^{K} \left(z_k (\mu + (1-\mu) z_k) - \frac{(1-\mu)}{2} z_k^2 \right) + \frac{\mu D^2}{2} \\
&= \sum_{i=1}^{m} \alpha_i - \frac{D^2 \mu}{2} - \frac{D^2 (1-\mu)}{2} \sum_{k=1}^{K} z_k^2 \\
&= \sum_{i=1}^{m} \alpha_i - \frac{1}{2(1-\mu)} \sum_{k \in J} \sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j) + \frac{\mu((|J|-1)\mu + 1)}{2(1-\mu)} D^2
\end{aligned}$$

We would like a a substitution of D^2 in terms of the dual variables. Hence rearranging primal variable $\|\mathbf{w}_k\|_2$ we get:

$$\begin{aligned}
\|\mathbf{w}_k\|_2 &= \frac{z_k}{\mu + (1-\mu) z_k} \sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j)} \\
&= \frac{\|\mathbf{w}_k\|_2}{D\mu + (1-\mu)\|\mathbf{w}_k\|_2} \sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j)},
\end{aligned}$$

where the second line follows from the identity $z_k = \|\mathbf{w}_k\|_2 / D$. Now rearranging to get an equation in terms of $\|\mathbf{w}_k\|_2$ we get for $k \in J$,

$$\begin{aligned}
\|\mathbf{w}_k\|_2 \left(D\mu + (1-\mu)\|\mathbf{w}_k\|_2 \right) &= \|\mathbf{w}_k\|_2 \sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j)} \\
\|\mathbf{w}_k\|_2 &= \frac{\sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j)} - D\mu}{1-\mu}
\end{aligned}$$

and using the identity $D = \sum_{k=1}^{K} \|\mathbf{w}_k\|_2$ to obtain

$$\begin{aligned}
\sum_{k=1}^{K} \|\mathbf{w}_k\|_2 &= D = \frac{\sum_{k \in J} \sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j)} - D\mu|J|}{1-\mu} \\
D((1-\mu) + \mu|J|) &= \sum_{k \in J} \sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j)}
\end{aligned}$$

which implies

$$D^2 = \frac{\left(\sum_{k \in J} \sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j)} \right)^2}{(1 - \mu + \mu|J|)^2}$$

Therefore, substituting this into the final equation of the dual we get

$$\sum_{i=1}^{m}\alpha_i - \frac{1}{2(1-\mu)}\sum_{k=1}^{K}\sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j) + \frac{\mu((|J|-1)\mu+1)}{2(1-\mu)}\frac{\left(\sum_{k\in J}\sqrt{\sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j)}\right)^2}{(1-\mu+\mu|J|)^2}$$
$$= \sum_{i=1}^{m}\alpha_i - \frac{1}{2(1-\mu)}\sum_{k\in J}\beta_k + \frac{\mu((|J|-1)\mu+1)}{2(1-\mu)(1-\mu+\mu|J|)^2}\left(\sum_{k\in J}\sqrt{\beta_k}\right)^2$$

where $\beta_k = \sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j)$, with the resulting dual optimisation:

$$\begin{array}{ll} \max_{\boldsymbol{\alpha}} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{A}{2}\sum_{k\in J}\beta_k + \frac{B}{2}\left(\sum_{k\in J}\sqrt{\beta_k}\right)^2 \\ \text{subject to} & \beta_k = \sum_{i,j=1}^{m}\alpha_i\alpha_j\kappa_k(\mathbf{x}_i,\mathbf{x}_j) \\ & 0 \le \alpha_i \le C,\ \ i = 1,\ldots,m \end{array}$$

where $A = 1/(1-\mu)$ and $B = ((|J|-1)\mu^2+\mu)/((1-\mu)(1-\mu+\mu|J|)^2)$.

References

[1] A. Argyriou, C. A. Micchelli, and M. Pontil. Learning convex combinations of continuously parameterized basic kernels. In *Computational Learning Theory*, volume 3559 of *Lecture Notes in Computer Science*, pages 338–352. Springer, 2005.

[2] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6, New York, NY, USA, 2004. ACM.

[3] Y. Chen, X. S. Zhou, and T. Huang. One-class SVM for learning in image retrieval. *Proceedings of International Conference on Image Processing 2001*, 1:34–37, 2001.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, U.K., 2000.

[5] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40:5:1–5:60, 2008.

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

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

[8] A. Klami, C. Saunders, T. de Campos, and S. Kaski. Can relevance of images be inferred from eye movements? In *MIR'08: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, Vancouver, British Columbia, Canada, October 2008.

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

[10] 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.

[11] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, Jan 1980.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal Computer Vision*, 60(2):91–110, 2004.

[13] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[14] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[15] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

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

[17] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[18] V. Viitaniemi and J. Laaksonen. Techniques for image classification, object detection and object segmentation applied to voc challenge 2007. Technical Report 2, Department of Information and Computer Science, Helsinki University of Technology (TKK), 2008.

[19] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'2007)*, pages 271–278, Amsterdam, Netherlands, 23–27 July 2007. ACM.