Representations and Similarities of Images for Visual Classification

Yan Liu

ECOLE CENTRALE DE LYON

THÈSE

pour obtenir le grade de
DOCTEUR DE L'ÉCOLE CENTRALE DE LYON
Spécialisté: Informatique

dans le cadre de l'École Doctorale Informatique et Information pour la Société
présentée et soutenue

par
Yan LIU
Soutenue le 22 mars 2010

Représentations et Similarités d'Images Pour la Classification Visuelle

---

Directeur de thèse: Liming CHEN
et
Co-directeur de thèse: Florent PERRONNIN

JURY

Prof. Stéphane CANU	Rapporteur
Prof. Frédéric JURIE	Rapporteur
Prof. Eric GAUSSIER	Président
Dr. Gabriela CSURKA	Examinateur
Prof. Liming CHEN	Directeur de thèse
Dr. Florent PERRONNIN	Co-directeur de thèse

Remerciements

Je tiens à remercier en tout premier lieu mon directeur de thèse Liming Chen pour m'avoir accueilli au sein du Laboratoire, pour m'avoir fait confiance, pour son soutien. Dans les périodes difficiles, il a su prendre du temps pour m'aider à avancer. Je lui exprime ma profonde reconnaissance.

Je tiens tout particulièrement à remercier mon co-directeur de thèse Florent Perronnin pour avoir su me guider avec gentillesse pendant ces trois années. Ses capacités scientifiques, son honnêteté humaine, son encouragement et ses remarques ont largement contribué à l'aboutissement de cette thèse.

J'adresse évidemment un grand merci à Chris Dance et Marco Bressan pour m'avoir donné l'opportunité de réaliser cette thèse dans le Centre de Recherche Européen de Xerox (XRCE).

Je tiens à remercier les rapporteurs Frédéric Jurie et Stéphane Canu qui ont pris le temps de lire et évaluer mes travaux ainsi que pour leurs remarques constructives qu'ils ont faites. J'adresse également des remerciements aux membres du jury pour l'intérêt qu'ils ont porté à mes travaux.

Cette thèse s'est déroulée successivement au sein du XRCE et du Laboratoire d'InfoRmatique en Image et Systèmes d'information. Je tiens à remercier tous mes collègues et mes amis du XRCE et du LIRIS pour l'ambiance chaleureuse, amicale et passionnée.

Enfin, je souhaite exprimer toute ma reconnaissance à ma chérie Xiaodong, mes parents et beaux-parents pour leur soutien constant et leur patience.

Contents

Chapter 1

Introduction

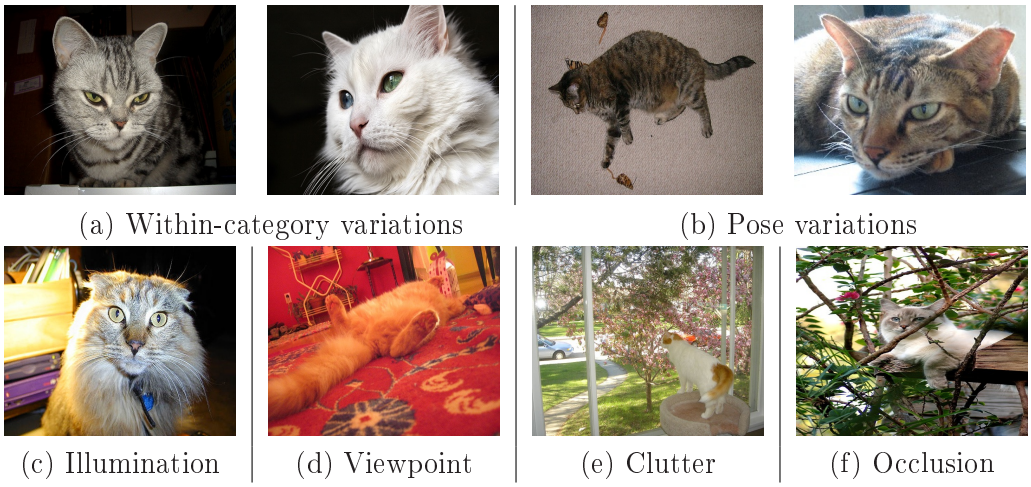
Contents

1.1 Context

With the rapid development of digital cameras and camera phones, the world is currently experiencing a digital imaging revolution. Many online photo sharing websites such as Flickr [Flickr] or Facebook [Facebook] have enabled millions of users to upload and share their photo albums over the Internet. For instance, as of October 2009, Flickr claimed to host more than 4 billion images [Flickr 2009]. In April 2009, Facebook announced that it hosted 15 billion photos with a growth rate of 220 million photos per week [Facebook 2009].

However, accessing these huge repositories remains an open problem. Search engines like Google Images [Google], Yahoo! Image Search [Yahoo!] or Picsearch [Picsearch] still rely only on the textual metadata associated with images (*e.g.* file name, title and tags if any) and disregard the most important information: the *image content*. There has been a great interest in the last decade in automatic image annotation, *i.e.* assigning to an image one or multiple labels based on its semantic content. Although it is an easy problem for humans, it has proved to be very difficult for computers to cope with variations in view, lighting and occlusion and with typical object and scene variations as illustrated by the cat images in Figure 1.1.

In the field of image annotation, representations based on the bag-of-patches (BOP) - which describe an image as an unordered set of low-level local feature vectors - combined with kernel-based learning have become the state-of-the-art. Kernel-based algorithms require the definition of a proper measure of similarity between BOP representations. The focus of this thesis has been on the definition of probabilistic representations of images based on the BOP assumption and the



(a) Within-category variations (b) Pose variations

(c) Illumination (d) Viewpoint (e) Clutter (f) Occlusion

Figure 1.1: Sample images for the category "cat" from the PASCAL VOC2007 Dataset.

computation of similarities between such representations with application to kernel learning of image classifiers.

1.2 Problems and Contributions

We now list the four problems that we addressed in this thesis with the proposed solutions.

1.2.1 Modeling Image as Adapted Gaussian Mixture Model

As already mentioned, the most successful image representation to date in computer vision is certainly the BOP. BOP representations are very simple, yet effective and demonstrated state-of-the-art performance in several evaluations [Everingham *et al.* 2005, Everingham *et al.* 2006, Everingham *et al.* 2007, Everingham *et al.* 2008, Everingham *et al.* 2009] although they disregard all information about the spatial layout of the feature descriptors. In this thesis, we will work only with BOP models.

As measuring directly the similarity between BOPs (model-free approaches) is costly, model-based approaches have been proposed. They consist in (i) modeling each set of feature vectors as a distribution and (ii) defining a measure of similarity between the vector sets as the similarity between their respective distributions.

There are two leading model-based approaches in the case of the BOP representation:

- The first one, the *bag-of-visual-words* (BOV), models an image as a discrete distribution [Sivic & Zisserman 2003, Csurka *et al.* 2004]. The BOV is based on an intermediate representation, the *visual vocabulary*, which is estimated offline on a large set of low-level feature vectors. Each image is characterized

by a histogram of visual words frequencies. One of the main limitations of this approach is the assumption that the distribution of feature vectors in any image can be known a priori.

- The second one models an image as a continuous distribution, generally a *Gaussian Mixture Model* (GMM) [Jebara & Kondor 2003, Jebara *et al.* 2004, Goldberger *et al.* 2003, Moreno *et al.* 2003, Vasconcelos 2004, Vasconcelos *et al.* 2004] and measures the similarity as the Kullback-Leibler divergence (KLD) or the probability product kernel (PPK) between GMMs. However, these methods have two main shortcomings. First, the robust estimation of the GMM parameters may be difficult as the cardinality of the vector set is small. Second, computing the similarity between two GMMs is expensive.

In chapter 4, we propose a novel GMM-based representation of images which is based on the maximum a posteriori (MAP) adaptation of a "universal" GMM. The advantages of this approach are twofold. MAP provides a more accurate estimate of the GMM parameters compared to standard maximum likelihood estimation (MLE). Moreover, there is a correspondence between the Gaussians of two GMMs adapted from a common distribution and one can take advantage of this fact to compute efficiently the probabilistic similarity. This work presented in chapter 4 was published in CVPR'08 [Liu & Perronnin 2008].

1.2.2 Modeling Images as Mixtures of Reference Images

We then address what we believe is an inherent limitation of traditional measures of similarity between distributions, such as the KLD or PPK. They give a high similarity when two distributions match globally but a low similarity when they match partially. This implies that two GMMs will typically have a high similarity if all their Gaussians match (at least approximately) but may have a low similarity because few Gaussians in one of the GMMs match poorly the Gaussians of the other GMM. If we translate this assertion into the image domain, this means that two images will have a high similarity if they match completely, *e.g.* same object in the same background, but may have a low similarity because they match partially.

In chapter 5 we thus propose to approximate an image, modeled as a GMM, as a convex combination of K reference image GMMs (later referred to as **MOM** for mixture of mixtures) and to characterize this image as a K-dimensional vector of mixture weights. These mixture weights measure a soft count of matching Gaussian components between the image to be described and each reference image. Hence, they encode a similarity which favors local matches (i.e. strong matches of individual Gaussians) and is therefore fundamentally different from traditional KLD or PPK. The vector of mixture weights may be used as input to a discriminative classifier for categorization.

Although the computation of the mixture weights is a convex optimization problem, its direct optimization is difficult. We propose two approximate optimization algorithms: the first one based on traditional sampling methods, the second one

based on a variational bound approximation of the true objective function. This work presented in chapter 5 was published in CVPR'09 [Perronnin & Liu 2009].

1.2.3 Contextual Similarity based on MOM

We say that a similarity between two distributions p and q is absolute if it only depends on p and q. We say that the similarity is contextual in the other case. We can see that the similarity defined between two images in the previous subsection is contextual as it depends on a set of reference images: *the context*. However, defining the similarity between two images using an explicit set of context images is both costly and cumbersome.

In chapter 6, we propose an alternative measure of contextual similarity between distributions: the similarity between two distributions p and q is measured in the context of a third distribution u. The similarity is computed following similar approximate optimization techniques proposed for the MOM approach. Although we do not observe a significant increase of accuracy, an advantage of the proposed similarity is that the context does not depend explicitly on a set of reference images. An extension, applying ideas proposed in chapter 6 to discrete distribution with application to image retrieval, was published in CVPR'09 [Perronnin *et al.* 2009].

1.2.4 Explicit Data Embedding for Large-Scale Learning

We focus on the problem of large-scale kernel learning on BOV representations. Linear classifiers are fast to learn and to evaluate but their accuracy is generally limited. Non-linear classifiers outperform significantly linear classifiers on BOV representations but are difficult to scale to large training sets.

However, non-linear kernel classifiers rely on an implicit mapping of the data such that non-linear classification in the original space corresponds to linear classification in the new space. It has been proposed to perform an explicit (possibly approximate) mapping of the data and to learn directly linear classifiers in the new space.

In chapter 7, we experiment with three approaches to BOV embedding: 1) kernel PCA (kPCA) [Schölkopf *et al.* 1998], 2) a modified kPCA we propose for additive kernels and 3) random projections for shift-invariant kernels [Rahimi & Recht 2007]. An important conclusion is that simply square-rooting BOV vectors - which corresponds to an exact mapping for the Bhattacharyya kernel -already leads to large improvements, often quite close to the best results obtained with additive kernels. Another conclusion is that, although it is possible to go beyond additive kernels, the embedding for non-additive kernels comes at a much higher cost. The work presented in chapter 7 was published in CVPR'10 [Perronnin *et al.* 2010].

1.3 Outline of thesis

The structure of this thesis is as follows.

- In chapter 2, we give more detailed reviews of related work, especially on BOP approaches and the similarity between BOPs.

- In chapter 3, we introduce the datasets used in this thesis.
- In chapter 4, we describe the estimation of universal and adapted image models. We present two similarity measures between distributions and explain how they can be approximated in our case.
- In chapter 5, we show that the vector of mixture weights can be computed through the optimization of a convex objective function. We compare our novel image representation with that standard kernel-based classifier *i.e.* the KLK or the PPK proposed in chapter 4 and demonstrate its efficiency.
- In chapter 6, we propose an alternative measure of contextual similarity between distributions.
- In chapter 7, we explain explicit data embedding for large-scale learning.
- In chapter 8, we summarize our conclusions and propose future directions.

Chapter 2

Literature Review

Contents

In this chapter, we review BOP approaches to image categorization. The literature review will be split into three parts. In the first part, we review very briefly the local feature detection and extraction. In the second part, we review the measures of similarity between BOP representations. In the third part, we review briefly kernel based classification.

2.1 Patch Detection and Extraction

There are mainly three strategies for patch detection:

- Interest Points: We just provide a few examples of interest points based methods. [Lindeberg 1998] developed a scale invariant blob detector, where a blob is defined by a maximum of the normalized Laplacian in scale-space. The original Harris corner detector [Harris & Stephens 1998] is invariant to rotation but is not scale-invariant. Multi-scale Harris by [Mikolajczyk & Schmid 2004] was adapted to solve this problem by selecting the points in the multi-scale representation. The Harris-Laplace detector [Mikolajczyk & Schmid 2001] is invariant to rotation and scale changes. The points are detected by the scale-adapted Harris function and selected in scale-space by the Laplacian of Gaussian operator. (see [Mikolajczyk *et al.* 2005] for a detailed overview)

- Random Sampling: [Marée *et al.* 2005] and [Nowak *et al.* 2006] showed experimentally that random sampling could outperform interest points detectors.

- Dense Sampling: similarly, [Winn *et al.* 2005, Fei-Fei & Perona 2005] showed experimentally that using regular grids to select patches could outperform interest points detectors as well.

It is possible to combine different strategies. The winning system of the PASCAL VOC 2007 Challenge which used the combination of interest points detectors and dense sampling was shown to outperform each method taken separately [Everingham *et al.* 2007].

One or multiple feature descriptors are computed for each extracted patch. We provide a few examples among the most well-known descriptors:

- The SIFT [Lowe 2004] descriptor subdivides the patch into a set of regions and computes a histogram of local oriented gradients in each subregion. Typically, there are $4 \times 4 = 16$ subregions and each histogram contains 8 bins which results in a 128 dimensional descriptor.

- The PCA-SIFT [Ke & Sukthankar 2004] consists in computing the gradient in horizontal and vertical directions at each pixel of the patch and then reducing the dimensionality of the resulting representation using Principal Component Analysis (PCA). This typically results in a 36 dimensional feature.

- The SURF [Bay *et al.* 2006] descriptor is similar to the SIFT in the sense that the patch is subdivided regularly into a set of subregions. In each subregion, four-dimensional features are computed based on Haar wavelets. This results in a 64 dimensional descriptor.

- Some of the previous descriptors can be extended to color features. This includes color SIFT features. RGB-SIFT [van de Sande *et al.* 2010] computes the SIFT feature for each RGB channel independently. HueSIFT [van de Weijer & Schmid 2006] concatenates the hue histogram with the SIFT descriptor.

In this thesis, we extract low-level features on regular grids at multiple scales and make use of two types of low-level features: the first features are SIFT descriptors and the second ones are based on simple RGB statistics.

2.2 Measuring the Similarity between Bags-of-Patches

We first review model-free approaches, *i.e.* the methods that directly compare sets of patches. We then review model-based approaches which consist in estimating a model (typically probabilistic) from each BOP representation and then computing the similarity between BOPs as a similarity between the models. These approaches can be divided into two sub-classes as a BOP can be modeled with a *discrete* or *continuous* distribution.

2.2.1 Model-Free Approaches

In this section we review those approaches which measure directly the similarity between two unordered feature sets. We assume that we have two feature sets $X = \{x_i, i = 1...T_X\}$ and $Z = \{z_j, j = 1...T_Z\}$. The simplest approach to defining a measure of similarity between such sets is the sum of the similarities between all

possible pairs of feature vector. Let $k(.,.)$ be a positive semi-definite kernel (psd). The summation kernel [Haussler 1999] is defined as

$$K_S(X,Z) = \frac{1}{T_X}\frac{1}{T_Z}\sum_{i=1}^{T_X}\sum_{j=1}^{T_Z} k(x_i, z_j). \tag{2.1}$$

However, its discriminative ability is compromised as all possible matchings between features are combined with equal weights. The good matchings could be easily swamped by the bad ones.

[Wallraven *et al.* 2003] and [Boughhorbed *et al.* 2004] both proposed a matching kernel that only considered the similarities of the best matched local features:

$$K_M(X,Z) = \frac{1}{2}\left[\frac{1}{T_X}\sum_{i=1}^{T_X}\max_{j=1\ldots T_Z} k(x_i, z_j) + \frac{1}{T_Z}\sum_{j=1}^{T_Z}\max_{i=1\ldots T_X} k(z_j, x_i)\right]. \tag{2.2}$$

Unfortunately, the "max" operator makes this kernel non-Mercer (not psd).

Lyu [Lyu 2005] proposed a Mercer kernel to quantify the similarities between feature sets. The kernel is a linear combination of the p-exponentiated kernels between local features:

$$K(X,Z) = \frac{1}{T_X}\frac{1}{T_Z}\sum_{i=1}^{T_X}\sum_{j=1}^{T_Z}[k(x_i, z_j)]^p. \tag{2.3}$$

p is the kernel parameter and $p > 1$ gives more influence to good matchings.

The Earth Mover's Distance (EMD) [Rubner *et al.* 1998] is a measure of similarity between sets of features and aims at finding an optimal matching that would be required to transform one set into the other. It is defined as

$$EMD = \max_{\substack{f_{ij}\\ i=1\ldots T_X\\ j=1\ldots T_Z}} \sum_{i=1}^{T_X}\sum_{j=1}^{T_Z} k(x_i, z_j) f_{ij}, \tag{2.4}$$

subject to the following constraints:

$$f_{ij} \geq 0, \tag{2.5}$$

$$\sum_{j=1}^{T_Z} f_{ij} \leq 1, \tag{2.6}$$

$$\sum_{i=1}^{T_X} f_{ij} \leq 1, \tag{2.7}$$

$$\sum_{i=1}^{T_X}\sum_{j=1}^{T_Z} f_{ij} = \min(T_X, T_Z). \tag{2.8}$$

f_{ij} is the flow between x_i and z_j. The computation of the EMD requires the computation of a similarity between all pairs of components of the two sets and the

optimization of a transportation problem whose complexity is cubic in the number of features.

To address the computational issue, [Grauman & Darrell 2005a] makes use of an embedding of the EMD based on the work of [Indyk & Thaper 2003]. However, the approximation suffers from a high error when the feature dimension increases.

All the previous approaches have high computational complexity: typically $O(T_X T_Z)$ with T_X and T_Z varying from a few hundreds to a few thousands.

[Grauman & Darrell 2005b] proposed the Pyramid Match Kernel (PMK) to address the computational cost issue. It partitions the feature space in a hierarchical manner, from fine to coarse and counts the number of correspondences between two feature sets at each level of the hierarchy. While the PMK is efficient for a small number of dimensions (*e.g.* 2), its cost grows exponentially with the number of dimensions.

2.2.2 Model-Based Approaches: Discrete Distributions

In this section, we review the approaches which model an image as a discrete distribution, *i.e.* bag-of-visual-words representation (BOV). There are three main steps in a BOV based system: offline visual vocabulary generation, histogram computation and histogram similarity computation.

2.2.2.1 Visual Vocabulary Generation

The visual vocabulary can be obtained offline on a set of BOPs by unsupervised or supervised learning methods.

The k-means [Sivic & Zisserman 2003, Csurka *et al.* 2004] is an unsupervised clustering algorithm which proceeds by iterated assignments of points to their closest cluster centers and re-computation of the cluster centers. The advantage of k-means is its simple and efficient implementation. One drawback of k-means is that most of the centroids will end-up in dense regions which do not necessarily correspond to discriminative patches. [Jurie & Triggs 2005] proposed a radius-based clustering, which avoids setting all cluster centers into high density areas and assigns all features within a fixed radius of similarity r to one cluster, where r is a parameter of the algorithm. [Wu & Rehg 2009] proposed to use the Histogram Intersection Kernel (HIK) instead of the popular Euclidean distance to cluster features into visual words. They also proposed to use the one-class SVM for the clustering and demonstrated higher accuracy.

In [Farquhar *et al.* 2005, Zhang *et al.* 2005], category specific vocabularies are trained and agglomerated into a single vocabulary. Although substantial improvements were obtained, this approach is impractical for a large number of categories as the size of the agglomerated vocabulary and the histogram representation grow linearly with the number of categories. Especially when the number of categories is large, this makes the histogram computation costly and the classifier learning challenging. Therefore, a compact visual vocabulary can provide a lower-dimensional representation and effectively avoid these difficulties. [Winn *et al.* 2005, Fulkerson *et al.* 2008, Lazebnik & Raginsky 2009] make use of

the mutual information between the features and category information to reduce the number of visual words without sacrificing the discriminative power of the vocabulary. [Moosmann *et al.* 2006] proposed an efficient alternative, in which training examples are recursively divided using a randomized decision forest and where the splits in the decision trees are comparisons of a descriptor dimension to a threshold. [Perronnin *et al.* 2006] characterize images using a set of category specific histograms, where each histogram describes whether the content can be best modeled by the universal vocabulary or by its corresponding category vocabulary. [Yang *et al.* 2008] proposed to unify the vocabulary construction with classifier training, and then to encode an image by a sequence of visual bits that constitute the semantic vocabulary.

The visual vocabulary generated by the unsupervised approaches is universal *i.e.* it is independent of the category set, but its discriminative ability is generally poor as it ignores label information. While supervised approaches explicitly incorporate the category information and can be more discriminative, we believe that they are not scalable. Indeed, every time a category is added to the system all image histograms have to be recomputed and all classifiers to be retrained.

2.2.2.2 Histogram Computation

The BOV approach characterizes an image by a histogram of visual word frequencies. There are two strategies to assign features to visual words: *hard assignment* and *soft assignment*. We assume that we have a set of features extracted from an image $X = \{x_t, t = 1...T\}$ where T is the number of features.

In hard assignment a feature vector is assigned to the single best representative visual word. The advantages of hard assignment are the simplicity and the fact that it leads to sparser histograms compared to soft assignment. The main problem is that two close patches might be assigned to two different centroids and therefore lead to two different representations.

There are two approaches to soft assignment. The first one consists in making the assignment a decreasing function of the Euclidean distance of the descriptor to the centroid. [van Gemert *et al.* 2008] uses for this purpose the Gaussian kernel. However, this is somewhat heuristic and requires the hand tuning of the bandwidth parameter. A more principled alternative consists in performing probabilistic clustering using typically a Gaussian Mixture Model (GMM) [Farquhar *et al.* 2005, Winn *et al.* 2005, Perronnin *et al.* 2006]. The soft assignment is then computed as the posterior probability of the Gaussian knowing the observations.

Let us call $\gamma_t(i)$ the hard/soft assignment of x_t to centroid/Gaussian i, then the BOV is computed as:

$$\frac{1}{T}\sum_{t=1}^{T}\gamma_t(i). \tag{2.9}$$

2.2.2.3 Similarity Computation

We assume that we have two BOV histograms $a = \{a_i, i = 1...N\}$ and $b = \{b_i, i = 1...N\}$. Among the most popular kernels are the Bhattacharyya kernel (BHA), the χ^2 kernel (CHI2) and the Intersection kernel (INT), all of which are positive semi-definite kernels (psd):

$$K_{bha}(a,b) = \sum_{i=1}^{N} \sqrt{a_i b_i}, \tag{2.10}$$

$$K_{chi2}(a,b) = 2\sum_{i=1}^{N} \frac{a_i b_i}{a_i + b_i}, \tag{2.11}$$

$$K_{int}(a,b) = \sum_{i=1}^{N} \min(a_i, b_i). \tag{2.12}$$

If K is a psd kernel, $\exp(\gamma K)$ with $\gamma > 0$ is also guaranteed to be a psd kernel. Hence, all the previous kernels can be exponentiated as $K^{exp} = \exp(\gamma(K-1))$. The -1 in the kernel is to ensure that the exponentiated kernel values are upper-bounded by 1. Also it makes these kernels easier to relate to well-known kernels: K^{exp}_{bha} is the rbf kernel on the square-rooted vectors, K^{exp}_{chi2} is a popular kernel used for instance in [Zhang *et al.* 2005, Vedaldi *et al.* 2009], and K^{exp}_{int} is the Laplacian kernel:

$$K^{exp}_{bha}(a,b) = \exp\left(-\frac{\gamma}{2}\sum_{i=1}^{N}(\sqrt{a_i} - \sqrt{b_i})^2\right), \tag{2.13}$$

$$K^{exp}_{chi2}(a,b) = \exp\left(-\frac{\gamma}{2}\sum_{i=1}^{N}\frac{(a_i - b_i)^2}{a_i + b_i}\right), \tag{2.14}$$

$$K^{exp}_{int}(a,b) = \exp\left(-\gamma\sum_{i=1}^{N}|a_i - b_i|\right). \tag{2.15}$$

The parameter γ controls the neighborhood size around the training samples and therefore the non-linearity of the classifier.

2.2.3 Model-Based Approaches: Continuous Distributions

In this section we review the approaches which model an image as a continuous distribution.

[Moreno *et al.* 2003, Farquhar *et al.* 2005] proposed to model an image with a single Gaussian with full covariance. However, the monomodal assumption is generally too restrictive. [Goldberger *et al.* 2003, Vasconcelos 2004, Vasconcelos *et al.* 2004] proposed to model an image as a mixture of Gaussians, generally with diagonal covariance.

2.2.3.1 Similarity Computation

A probabilistic kernel is defined between the distributions, such as the *Probability Product Kernel* (PPK) and the *Kullback-Leibler Kernel* (KLK). There is a closed

form solution for the PPK or the KLK between two Gaussians. However there is generally no closed form solution for them in the case of GMMs. We only provide definitions of the kernels and will discuss in chapter 4 how these measures can be approximated in the case of mixture models.

We assume that we have two continuous distributions p and q defined on the space $\mathbb{R}^D$ (D is the dimensionality of the low-level features). Jebara *et al.* proposed the PPK [Jebara & Kondor 2003, Jebara *et al.* 2004] as a kernel between distributions:

$$K^{\rho}_{ppk}(p,q) = \int_{x\in\mathbb{R}^D} p(x)^{\rho} q(x)^{\rho} dx \tag{2.16}$$

where ρ is a parameter. The PPK is trivially psd and has two special cases. When $\rho = 1$, the PPK takes the form of the expectation of one distribution under the other. This is referred to as the *Expected Likelihood Kernel* (ELK):

$$K_{elk}(p,q) = \int_{x\in\mathbb{R}^D} p(x)q(x)dx = E_p[q(x)] = E_q[p(x)] \tag{2.17}$$

When $\rho = 1/2$, it is known as the *Bhattacharyya Kernel* (BHA):

$$K_{bha}(p,q) = \int_{x\in\mathbb{R}^D} \sqrt{p(x)}\sqrt{q(x)}dx. \tag{2.18}$$

This is an upper bound on the Bayesian error.

The Kullback-Leibler Divergence (KLD) [Kullback 1968] is defined as follows:

$$KL(p||q) = \int_{x\in\mathbb{R}^D} p(x)\log\frac{p(x)}{q(x)}dx. \tag{2.19}$$

The symmetric KL is given by:

$$SKL(p,q) = KL(p||q) + KL(q||p). \tag{2.20}$$

The KLK [Moreno *et al.* 2003] can then be defined by exponentiating the symmetric KLD (not necessarily psd):

$$K_{klk} = \exp(-\gamma SKL(p,q)). \tag{2.21}$$

where $\gamma > 0$ is a kernel parameter.

2.3 Kernel-Based Learning

We assume that we have a set of N labeled training samples $\{(x_i, y_i), i = 1...N\}$ where each x_i is an image representation (not to be mistaken for the x_i of BOP in section 2.2.1) and each $y_i \in \{-1, +1\}$ is a binary label.

In kernel based classification, the decision function has the form:

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + \beta. \tag{2.22}$$

with $\alpha = [\alpha_1, ..., \alpha_N]$ an N-dimensional vector of parameters, β an offset parameter and $K(x, x_i)$ the similarity between x and training sample x_i.

The most popular kernel classifier is certainly the Support Vector Machine (SVM) [Vapnik 1995]. The parameters α and β are learned by minimizing the following loss function:

$$\frac{1}{N}\sum_{i=1}^{N} \ell_{hinge}(f(x_i), y_i) + \lambda||f||_2^2 \tag{2.23}$$

where $\ell_{hinge}(f(x_i), y_i) = \max(0, 1 - f(x_i)y_i)$ is the hinge loss and λ is the regularization parameter for penalization. The major advantage of the SVM is that it leads to a sparse solution in α, thus reducing the cost of classification.

An alternative classifier is Sparse Logistic Regression (SLR) [Krishnapruam *et al.* 2005] which minimizes the following loss function:

$$\frac{1}{N}\sum_{i=1}^{N} \ell_{log}(f(x_i), y_i) + \lambda||\alpha||_1 \tag{2.24}$$

where $\ell_{log}(f(x_i), y_i) = \log(1 + \exp(-f(x_i)y_i))$. The L_1 regularization leads to a sparse solution, as is the case of the SVM. Both SVM and SLR lead to similar results as observed experimentally in [Krishnapruam *et al.* 2005]. We used SLR in most of our experiments.

Chapter 3

Datasets

Contents

3.1 PASCAL VOC2007

We used the PASCAL VOC2007 database [Everingham *et al.* 2007] which contains a total of 9,963 images: 5,011 images for training and 4,952 for testing. There are 20 different object classes: person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa and tv monitor. Sample images can be seen on Figure 3.1.

3.2 Scene-15

The Scene-15 database provided by several research groups [Oliva & Torralba 2001, Fei-Fei & Perona 2005, Lazebnik *et al.* 2006] is composed of 4,485 images falling into 15 categories: bedroom, suburb, industrial, kitchen, living room, coast, forest, highway, inside city, mountain, open country, street, tall building, office and store. The number of images in each category ranges from 200 to 400. The major sources of the pictures in the dataset include the COREL collection, personal photographs, and Google image search. Sample images can be seen on Figure 3.2.

3.3 Caltech-256

The Caltech-256 [Griffin *et al.* 2007] database has a total of 29,780 images containing 256 categories. Each category contains at least 80 images. We show the taxonomic tree of all the 256 categories proposed by Greg Griffin [Griffin 2007] on Figure 3.3.

3.4 ImageNet

ImageNet is an image database organized according to the WordNet hierarchy and collects approximately 9.4M images corresponding to almost 15K concepts. Images of each concept are quality-controlled and human-annotated using the Amazon

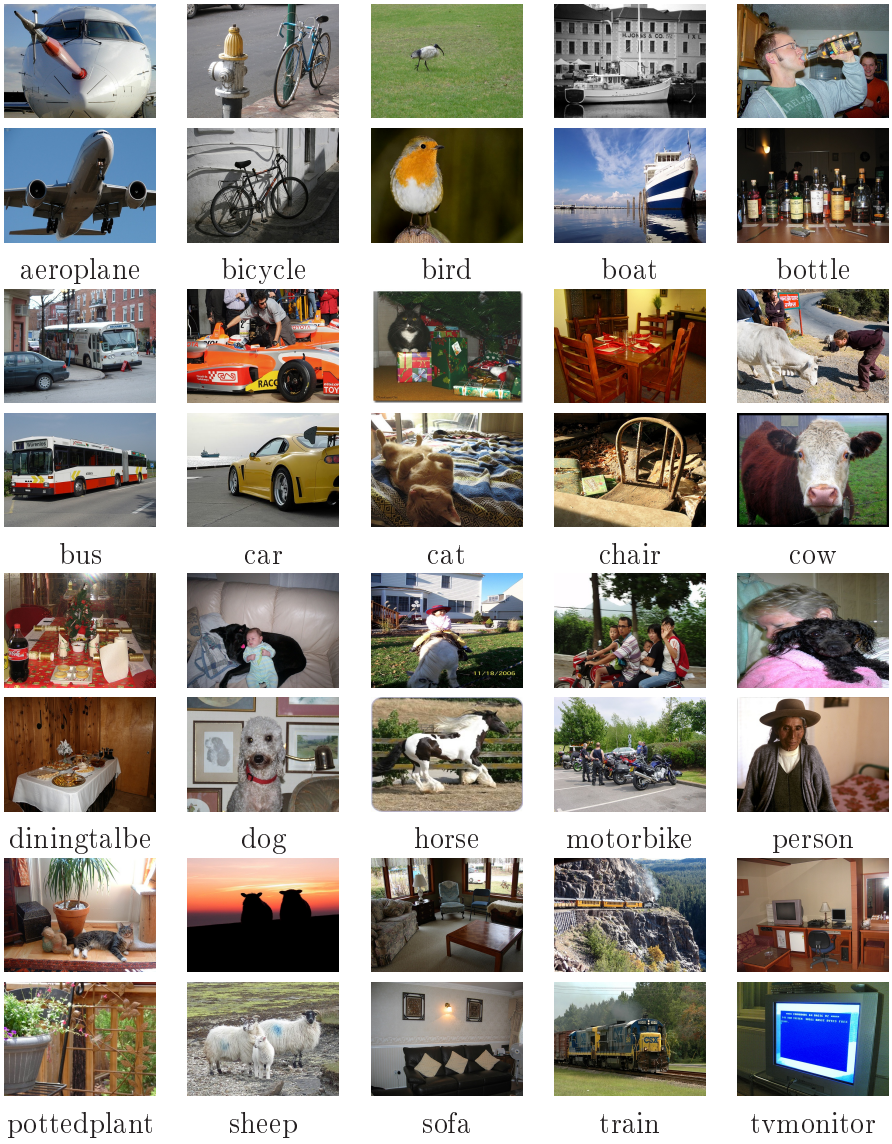


Figure 3.1: Sample images from the PASCAL VOC2007 database for all 20 categories.

Figure 3.2: Sample images from Scene-15 database.

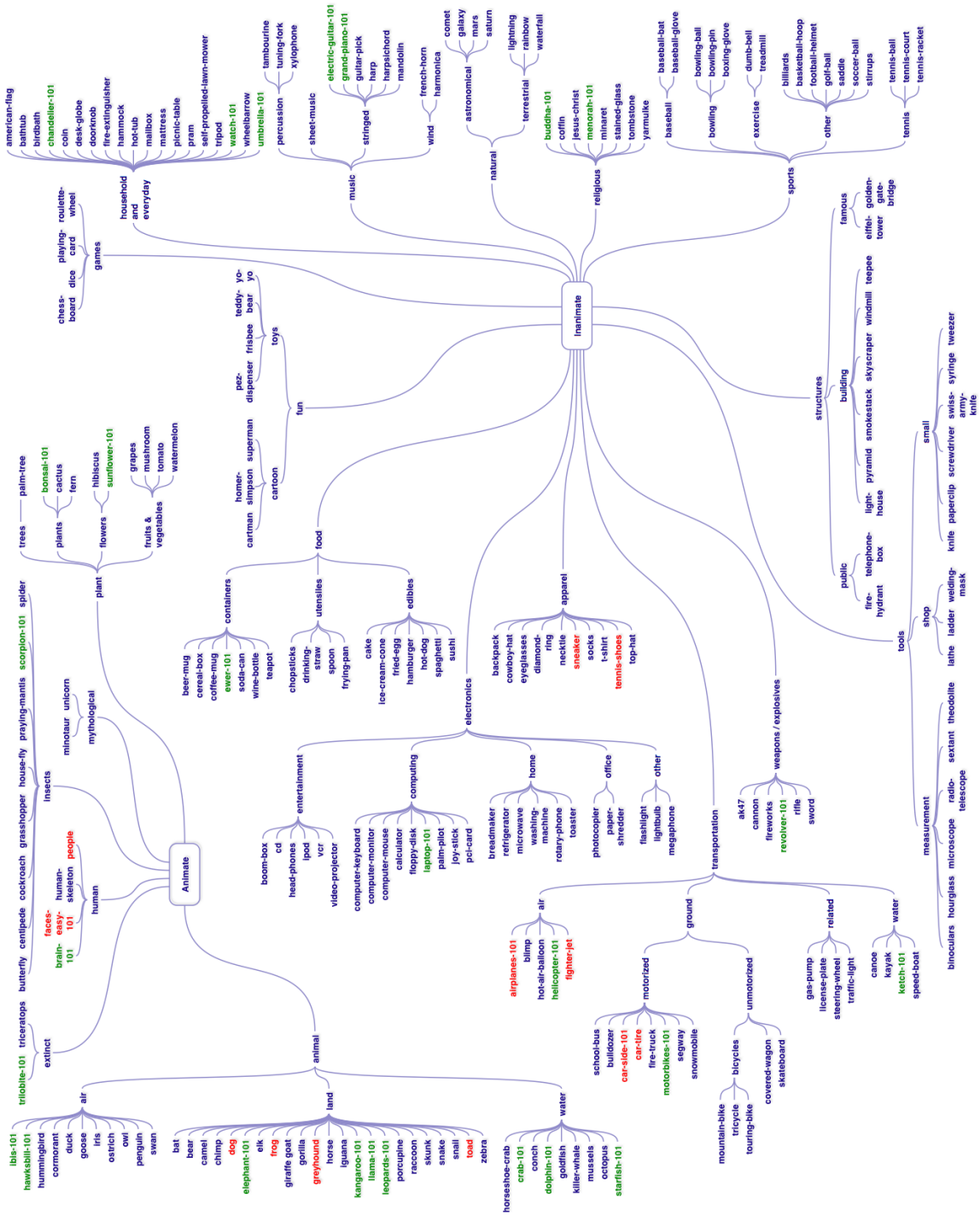


Figure 3.3: Taxonomic tree for Caltech-256 database taken from [Griffin 2007].

Mechanical Turk (AMT:www.mturk.com). The database is publicly available at http://www.image-net.org.

Chapter 4

Images as Adapted Mixtures of Gaussians

Contents

In this chapter, we focus on those methods which model images as continuous distributions (GMMs) [Jebara & Kondor 2003, Jebara *et al.* 2004, Goldberger *et al.* 2003, Moreno *et al.* 2003, Vasconcelos 2004, Vasconcelos *et al.* 2004]. We note that these approaches have two main shortcomings:

- First, the robust estimation of the GMM parameters may be difficult if the cardinality of the feature set is small. For instance, the number of features extracted from an image typically varies from a few hundreds up to a few thousands. One could increase this number, *e.g.* by using a denser grid in the case of regular extraction or by lowering the detection threshold in the case of interest points detectors, but this would also significantly increase the cost of the feature extraction and the GMM estimation.

- Second, as there is no closed form solution for the KLK or the PPK between two GMMs, the cost of traditional approximations is typically quadratic in the number of Gaussian components.

In this chapter, we present a novel approach to compute the similarity between two unordered variable-sized vector sets with application to image categorization.

It consists in representing one image by a GMM obtained through maximum a posteriori (MAP) adaptation of a common "universal" GMM. This offers two main advantages:

- First, *MAP estimation is more accurate than MLE* in the challenging case where the training data is scarce as the universal model provides a priori information on the location of the parameters in the whole parameter space. We will show experimentally that this a priori information needs not to be exact: even if the universal model is learned on a set of images which is not directly related to the task at hand, excellent performance is obtained.
- Second, if two GMMs are adapted from a common distribution, there is a one-to-one correspondence between their Gaussians. We make use of this correspondence to derive approximations of the PPK and KLK with a *cost linear in the number of Gaussians*.

Note that the idea of learning visual vocabularies - modeled as GMMs - through the adaptation of a common universal vocabulary has already been used in [Perronnin *et al.* 2006]. However, in [Perronnin *et al.* 2006] the adapted vocabularies are *class-GMMs* and images are modeled with histograms of visual-word occurrences while in this chapter the adapted vocabularies are *image-GMMs*.

The remainder of this chapter is organized as follows. In section 4.1 we describe the estimation of universal and adapted GMMs. In section 4.2 we first present two similarity measures between distributions, the PPK and the KLD, review how they were approximated in the state-of-the-art and then propose our approximations. In section 4.3 we provide experimental results on three databases: PASCAL VOC2007, Scene-15 and Caltech-256 and compare to the state-of-the-art. Finally we draw a brief conclude in section 4.4.

4.1 Image Representation

Let us first introduce our notations. The parameters of a GMM are denoted $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1...N\}$ where w_i, μ_i and Σ_i are respectively the weight, mean vector and covariance matrix of Gaussian i and N denotes the number of Gaussians. Let x_t be an observation vector and q_t its associated hidden variable, *i.e.* the variable indicating which Gaussian emitted x_t.

The likelihood that observation x_t was generated by the GMM is:

$$p(x_t|\lambda) = \sum_{i=1}^{N} w_i p_i(x_t|\lambda) \ . \tag{4.1}$$

where $p_i(x_t|\lambda) = p(x_t|q_t = i, \lambda)$. Finally, $\gamma_i(x_t) = p(q_t = i|x_t, \lambda)$ is the occupancy probability, *i.e.* the probability that observation x_t was generated by Gaussian i. It is computed using Bayes formula:

$$\gamma_i(x_t) = \frac{w_i p_i(x_t|\lambda)}{\sum_{j=1}^{N} w_j p_j(x_t|\lambda)} \ . \tag{4.2}$$

We now describe the training of the universal model and the adapted image models.

4.1.1 Training the Universal Model

The universal GMM is supposed to describe the content of any image and, therefore, it should be trained offline on a varied set of images. Let λ^u denote the parameters of the universal GMM. Let $X = \{x_t, t = 1...T\}$ be the set of training vectors. The estimation of λ^u may be performed by maximizing the log-likelihood function $\log p(X|\lambda^u)$. The standard procedure for MLE is the Expectation Maximization (EM) algorithm [Dempster *et al.* 1977]. For the E-step, the values $\gamma_i(x_t)$ are computed. We provide here for completeness the M-step re-estimation equations [Bilmes 1998]:

$$\hat{w}_i^u = \frac{1}{T}\sum_{t=1}^{T}\gamma_i(x_t) \ , \tag{4.3}$$

$$\hat{\mu}_i^u = \frac{\sum_{t=1}^{T}\gamma_i(x_t)x_t}{\sum_{t=1}^{T}\gamma_i(x_t)} \ , \tag{4.4}$$

$$\hat{\Sigma}_i^u = \frac{\sum_{t=1}^{T}\gamma_i(x_t)x_t x_t'}{\sum_{t=1}^{T}\gamma_i(x_t)} - \hat{\mu}_i^u \hat{\mu}_i^{u\prime} \ . \tag{4.5}$$

4.1.2 Training Adapted Image Models

Our primary motivation for learning the image GMMs through the adaptation of a universal model is to overcome the scarcity of the training material. Indeed, only a small number of low-level feature vectors (typically from a few hundreds up to a few thousands) are extracted from one image. We will observe in section 4.3 that this is insufficient to train robustly a mixture with a large number of Gaussians (*e.g.* 100) for each image. In the following, λ^a denotes the parameters of an adapted model.

Let $X = \{x_t, t = 1...T\}$ now denote the set of adaptation samples extracted from one image. We use the MAP criterion to adapt a GMM. The goal of MAP estimation is to maximize the posterior probability $p(\lambda^a|X)$ or equivalently $\log p(X|\lambda^a) + \log p(\lambda^a)$. Hence, the difference with MLE is in the assumption of a prior distribution $p(\lambda^a)$. To perform MAP learning, one has to (i) choose the prior distribution family and (ii) specify the parameters of the prior distribution.

It was shown in [Gauvain & Lee 1994] that the prior densities for GMM parameters could be adequately represented as a product of Dirichlet (prior on weight parameters) and normal-Wishart densities (prior on Gaussian parameters). When adapting a universal model with MAP to more specific conditions, it is natural to use the parameters of the universal model as a priori information on the location of the adapted parameters in the parameter space. As shown in [Gauvain & Lee 1994], one can also apply the EM procedure for MAP estimation. The M-step re-estimation

equations are provided here for completeness:

$$\hat{w}_i^a = \frac{\sum_{t=1}^{T} \gamma_i(x_t) + \tau}{T + N \times \tau} , \tag{4.6}$$

$$\hat{\mu}_i^a = \frac{\sum_{t=1}^{T} \gamma_i(x_t) x_t + \tau \mu_i^u}{\sum_{t=1}^{T} \gamma_i(x_t) + \tau} , \tag{4.7}$$

$$\hat{\Sigma}_i^a = \frac{\sum_{t=1}^{T} \gamma_i(x_t) x_t x_t' + \tau [\Sigma_i^u + \mu_i^u \mu_i^{u\prime}]}{\sum_{t=1}^{T} \gamma_i(x_t) + \tau} - \hat{\mu}_i^a \hat{\mu}_i^{a\prime} . \tag{4.8}$$

The *relevance factor* τ keeps a balance between the a priori information contained in the generic model λ^u and the new evidence contained in X. If a mixture component i was estimated with a small number of observations $\sum_{t=1}^{T} \gamma_i(x_t)$, then more emphasis is put on the a priori information. On the other hand, if it was estimated with a large number of observations, more emphasis is put on the new evidence. Hence MAP provides a more robust estimate than MLE when little training data is available. The parameter τ is generally set manually [Gauvain & Lee 1994, Reynolds *et al.* 2000].

For a given number of Gaussians, the cost of one EM iteration is (almost) identical for MLE and MAP. The only difference is the addition in the M-step of MAP of the a priori information in the statistics (compare equations 4.3, 4.4 and 4.5 to 4.6, 4.7 and 4.8 resp.) However, as MAP uses some a priori information on the location of the parameters, it requires a smaller number of EM iterations to reach an accurate estimate. Therefore, it is *significantly faster* compared to MLE. This statement will be verified experimentally.

We finally note that an adapted model contains the same number of Gaussians as the universal model from which it is adapted.

4.2 Measures of Similarity of GMMs

In the following, we first present two measures of similarity between distributions and review how they were approximated in state-of-the-art. We then propose our approximations.

4.2.1 Probability Product Kernel

4.2.1.1 Previous Approximatins

We recall that the probability product kernel (PPK) [Jebara & Kondor 2003] between probability distributions p and q is defined as follows:

$$K_{ppk}^{\rho}(p, q) = \int_{x \in \mathbb{R}^D} p(x)^{\rho} q(x)^{\rho} dx . \tag{4.9}$$

There is a closed form solution for the PPK between two Gaussians:

$$K_{ppk}^{\rho}(p, q) = (2\pi)^{(1-2\rho)D/2} \mid \Sigma \mid^{1/2} \mid \Sigma_p \mid^{-\rho/2} \mid \Sigma_q \mid^{-\rho/2} \exp\left(-\frac{\rho}{2} \mu_p^{\top} \Sigma_p^{-1} \mu_p - \frac{\rho}{2} \mu_q^{\top} \Sigma_q^{-1} \mu_q + \frac{1}{2} \mu^{\top} \Sigma \mu \right) , \tag{4.10}$$

where μ_p, Σ_p and μ_q, Σ_q are respectively the mean and covariance matrix of Gaussian p and q, $\Sigma = (\rho\Sigma_p^{-1} + \rho\Sigma_q^{-1})^{-1}$, $\mu = \rho(\Sigma_p^{-1}\mu_p + \Sigma_q^{-1}\mu_q)$ and D is the dimensionality of the feature vectors.

However there is no closed form solution for the PPK in the case of mixtures of Gaussians except for the special case $\rho = 1$ (which led to poor results in preliminary experiments). In the case of a mixture model, we have $p(x) = \sum_{i=1}^{N} \alpha_i p_i(x)$ and $q(x) = \sum_{j=1}^{M} \beta_j q_j(x)$. In [Jebara & Kondor 2003] (section 4) the following approximation is suggested:

$$K_{ppk}^{\rho}(p, q) \approx \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j K_{ppk}^{\rho}(p_i, q_j) \ . \tag{4.11}$$

When $\rho \leq 1$ this approximation corresponds to an upper-bound on the true value of $K_{ppk}^{\rho}(p, q)$ and when $\rho \geq 1$ it is a lower-bound.

[Hershey & Olsen 2008] proposed a variational bound method to approximate the Bhattacharyya Kernel (BHA), *i.e.* when $\rho = 1/2$ (*c.f.* formula 2.18). It leads to the following bound:

$$K_{bha}(p, q) \geq \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j K_{bha}^2(p_i, q_j)}. \tag{4.12}$$

In all experiments, we used 4.11 as an approximation of K_{bha} [1].

The evaluation of the PPK between two GMMs which contain respectively M and N Gaussians requires the computation of $M \times N$ PPKs between individual Gaussians. This cost may be a handicap in the case of large values of M and N.

4.2.1.2 Proposed Approximation

We make use of the fact that two mixtures of Gaussians have been adapted from the same generic model to speed-up the computation. Indeed, [Reynolds *et al.* 2000] first noticed that there is a one-to-one correspondence between the i-th Gaussian of an adapted GMM and the i-th Gaussian of the GMM it is adapted from. By transitivity, it means that there is a one-to-one correspondence between the i-th Gaussians of two GMMs adapted from the same GMM (we recall that we necessarily have $M = N$ in our adaptation framework). Consequently, in our case, the terms $K_{ppk}^{\rho}(p_i, q_i)$ dominate the previous sum and the PPK may be further approximated as follows:

$$K_{ppk}^{\rho}(p, q) \approx \sum_{i=1}^{N} \alpha_i \beta_i K_{ppk}^{\rho}(p_i, q_i) \ . \tag{4.13}$$

This evaluation requires only the computation of N PPKs between individual Gaussians.

[1] We became aware of [Hershey & Olsen 2008] more than a year after this work was done. However, we do not believe that 4.12 would lead to significantly better results than 4.11.

4.2.2 Kullback-Leibler Kernel

4.2.2.1 Previous Approximations

We recall that the Kullback-Leibler Divergence (KLD) between two continuous distributions is defined as follows:

$$KL(p||q) = \int_{x\in\mathbb{R}^D} p(x)\log\frac{p(x)}{q(x)}dx \ . \tag{4.14}$$

There is also a closed form solution for the KLD between two Gaussians:

$$KL(p||q) = \frac{1}{2}\left[\log\frac{\mid\Sigma_q\mid}{\mid\Sigma_p\mid} + \mathrm{Tr}(\Sigma_q^{-1}\Sigma_p) + (\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q) - D\right] \ . \tag{4.15}$$

However, no closed-form expression exists for the KLD between two GMMs.

[Moreno *et al.* 2003] and [Vasconcelos *et al.* 2004] approximate the KLD using Monte Carlo (MC) sampling:

$$KL(p||q) = E_p[\log\frac{p}{q}] \approx \frac{1}{T}\sum_{t=1}^{T}\log\frac{p(x_t)}{q(x_t)} \tag{4.16}$$

where $x_1, ..., x_T$ are sampled from p.

[Goldberger *et al.* 2003] used the unscented transform mechanism to obtain an approximation for KLD between two GMMs:

$$KL(p||q) \approx \frac{1}{2d}\sum_{i=1}^{N}\alpha_i\sum_{k=1}^{2d}\log\frac{p(x_{i,k})}{q(x_{i,k})} \tag{4.17}$$

such that:

$$x_{i,k} = \mu_{p_i} + (\sqrt{d\Sigma_{p_i}})_k \qquad k = 1, ...d, \tag{4.18}$$

$$x_{i,d+k} = \mu_{p_i} - (\sqrt{d\Sigma_{p_i}})_k \qquad k = 1, ...d. \tag{4.19}$$

where d is the number of chosen "sigma" points, μ_{p_i} is the mean of Gaussian p_i and $(\sqrt{\Sigma_{p_i}})_k$ is the k-th column of the matrix square root of Σ_{p_i}. The unscented transform is similar to the Monte Carlo technique except that the samples are chosen deterministically. However, the cost of both methods can be prohibitive as one has to draw a large number of samples to obtain a reasonable estimate.

[Goldberger *et al.* 2003] and [Vasconcelos 2004] proposed two very similar approximations of the KLD. They are based on a two-step approach: it first consists in finding a mapping π from the Gaussians of p to the Gaussians of q as follows:

$$\pi(i) = \arg\min_j \left(KL(p_i||q_j) - \log\beta_j\right) \ . \tag{4.20}$$

Then π is used to approximate the KLD:

$$KL(p||q) \approx \sum_{i=1}^{N}\alpha_i\left(KL(p_i||q_{\pi(i)}) + \log\frac{\alpha_i}{\beta_{\pi(i)}}\right) \ . \tag{4.21}$$

This approximation is well motivated when Gaussians have little overlap, *e.g.* when the dimensionality D of the feature space is high. In our experiments, $D = 50$ (*c.f.* section 4.3.1). If two GMMs contain respectively M and N Gaussians, computing the mapping function π requires the computation of $M \times N$ KLDs between individual Gaussians.

[Hershey & Olsen 2007] also proposed a variational approximation for the KLD. The idea is to write:

$$KL(p||q) = H(p,q) - H(p,p). \tag{4.22}$$

where $H(p,q)$ is the cross-entropy between p and q and to compute a variational bound on H:

$$H(p,q) \leq -\sum_{i=1}^{N} \alpha_i \log \left(\sum_{j=1}^{M} \beta_j \exp(-H(p_i,q_j)) \right). \tag{4.23}$$

Since this KL approximation is the difference of two bounds, it is not a bound.

In our experiments we used 4.20 and 4.21 to approximate the KLD [2].

4.2.2.2 Proposed Approximation

Once again, we can make use of the fact that there is a one-to-one correspondence between the Gaussians of two GMMs adapted from the same model to perform the following approximation: $\pi(i) = i$. Under this assumption, the KLD can be rewritten:

$$KL(p||q) \approx \sum_{i=1}^{N} \alpha_i \left(KL(p_i||q_i) + \log \frac{\alpha_i}{\beta_i} \right). \tag{4.24}$$

Hence, the computation of the KLD requires only N Gaussian computations in our case. We recall that the Kullback-Leibler Kernel (KLK) is defined as:

$$K_{klk} = \exp(-\gamma(KL(p||q) + KL(q||p)). \tag{4.25}$$

4.3 Experiments

We first describe our experimental setup. We then report results on three datasets: the PASCAL VOC2007, Scene-15 and Caltech-256 databases.

4.3.1 Experimental Setup

Low-level feature vectors are extracted on regular grids at multiple scales in our experiments. There is an average of 1,000 feature vectors extracted per image per feature type. We make use of two types of low-level features. The first features are based on local histograms of orientations as described in [Lowe 2004] (128 dimensional features, later referred to as ORH). The second ones are based on RGB

[2] Similarly, we became aware of [Hershey & Olsen 2007] more than a year after this work was done. Again, we do not believe that the variational approximation would lead to significantly better results.

statistics (96 dimensional features, later referred to as COL). In both cases, the dimensionality of the feature vectors is reduced to 50 through Principal Component Analysis (PCA).

The universal GMM is trained using the following iterative strategy inspired by HTK [Young *et al.* 2002]. We first train a GMM with a single Gaussian. We then split it into two by introducing a small perturbation in the mean parameter and retrain the GMM using several iterations of EM. The process of splitting and retraining is repeated until the desired number of Gaussians is obtained. To train the adapted image GMMs with MAP, the default value for the relevance factor is $\tau = 10$.

For the PPK, we choose $\rho = 1/2$ (*i.e.* the Bhattacharyya Kernel) as this value leads to the best results in preliminary experiments. To set parameter γ for the KLK (*c.f.* equation 2.21) we followed [Zhang *et al.* 2005]: γ is equal to the inverse of the mean of the symmetric KL divergence $SKL(p, q)$ between two GMMs (*c.f.* equation 2.20) as estimated on a subset of the whole training set.

For the classification, we used Sparse Logistic Regression (SLR) [Krishnapruam *et al.* 2005] as a discriminative classifier. One linear classifier is trained per class in a one-versus-all manner.

We have two separate systems: one for each feature type. The end result is the average of the scores of the two systems (later referred to as ORH+COL).

4.3.2 PASCAL VOC2007

During the VOC2007 competition, the accuracy was primarily measured with the Average Precision (AP). Therefore, we use the mean of AP (averaged over the 20 categories) to make our results easily comparable to the state-of-the-art.

In the following, we start with a comparative evaluation of the proposed approach. We then proceed with the analysis of the impact of parameter τ. We also carry out cross-database experiments showing that, even if the universal model is learned on a different database, the performance does not vary significantly. Finally, we analyze the computational cost of the proposed method on this database.

4.3.2.1 Results

We compare the performance of the following systems:

(i) A system which learns the image BOVs with χ^2 kernel. This system is later referred to as BOV_CHI2.

(ii) The proposed approach with MAP adaptation and the fast one-to-one mapping of Gaussian components (*c.f.* formula (4.13) for PPK and formula (4.24) for KLK). This system is later referred to as MAP_OTO.

(iii) A system which learns the image GMMs with MLE (using the same iterative strategy which was employed to train the universal model) and the slow one-to-many mapping of Gaussian components (*c.f.* formula (4.11) for PPK and formulae (4.20) and (4.21) for KLK). This system is later referred to as MLE_OTM.

(iv) An intermediate system which makes use of MAP adaptation as is the case of (ii) but which uses the slow one-to-many scoring of (iii). This system is later referred to as MAP_OTM.

Hence, when comparing (i) and (ii), we can measure the advantage of image GMMs compared to image BOVs. When comparing (iii) and (iv), we can measure the benefit of MAP compared to MLE. When comparing (ii) and (iv) we can measure the impact on the accuracy of the fast one-to-one scoring versus the slow one-to-many scoring.

Results are provided on Figure 4.1. We did not represent the performance of BOV_CHI2 on this figure because BOV typically requires a larger number of Gaussians. The best results we obtained with BOV_CHI2 was a mean AP of 52.6% with approximately 4000 Gaussians (for ORH + COL). Then the following conclusions could be drawn from the figures.

- First, the proposed method consistently outperforms the BOV_CHI2.
- Second, because of computationally expensive cost of one-to-many scoring for both MLE and MAP, we only show the performances with 16, 32 and 64 Gaussians. From the results we have shown, we can see that MAP clearly outperforms MLE for both PPK and KLK. Especially the performance of the MLE_OTM system starts to drop for more Gaussians while for MAP it continues to increase. This shows that we can learn robustly a larger number of Gaussians with MAP than with MLE.
- Third, the accuracy of PPK_OTO is superior to that of PPK_OTM. This observation came as a surprise as we first thought that by dropping terms in equation 4.11, we would lose information. Our best explanation is that the bound 4.11 is too coarse an approximation of the PPK. This suggests an alternative approach for computing the PPK similar to that used for KLK: first find a matching between the Gaussians of p and q and then approximate the PPK as a weighted sum of PPKs between the matched Gaussians (this approximation might be worth testing in the future).
- Fourth, the accuracy of KLK_OTO is inferior to that of KLK_OTM, but not significantly so, showing that our one-to-one approximation is a good one.

As we used the standard VOC2007 protocol, our best result is 55.4% which can be compared to those published in the literature. The best result reported on this dataset during the challenge was 59.4% (INRIA-genetic) [Marszalek *et al.* 2007]. We would like to outline that the cost of training and testing our system is significantly lower compared to that of the winning system as it made use of 21 "channels" (while we make use of only 2: ORH + COL) and a sophisticated approach to combine them.

4.3.2.2 Influence of Parameters

We now analyze the influence of the relevance factor τ. τ impacts two competing aspects of our system:

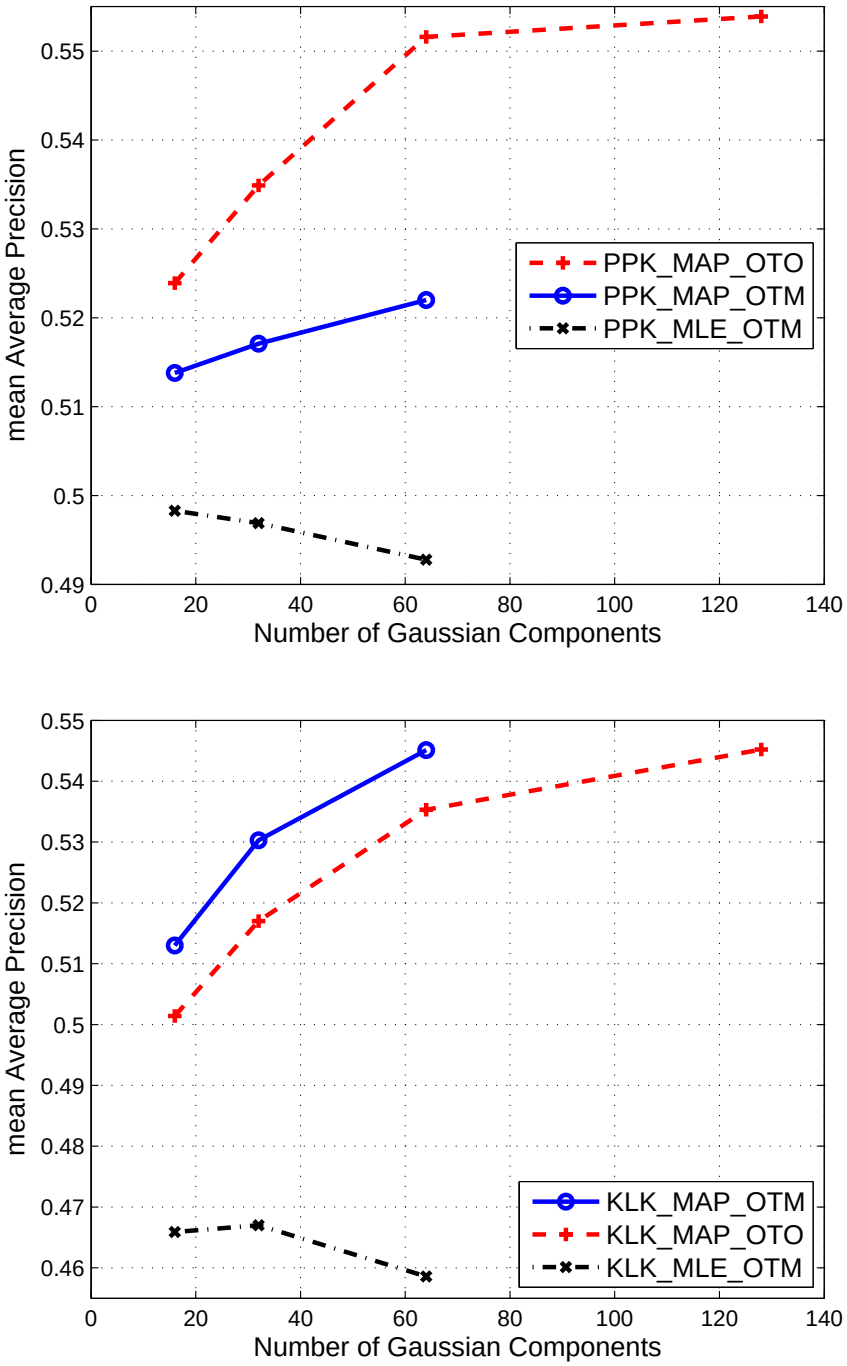


Figure 4.1: Performance on the PASCAL VOC2007 database of the PPK (top) and KLK (bottom) for a varying number of Gaussian components. The mean AP is shown for system based on ORH+COL features.

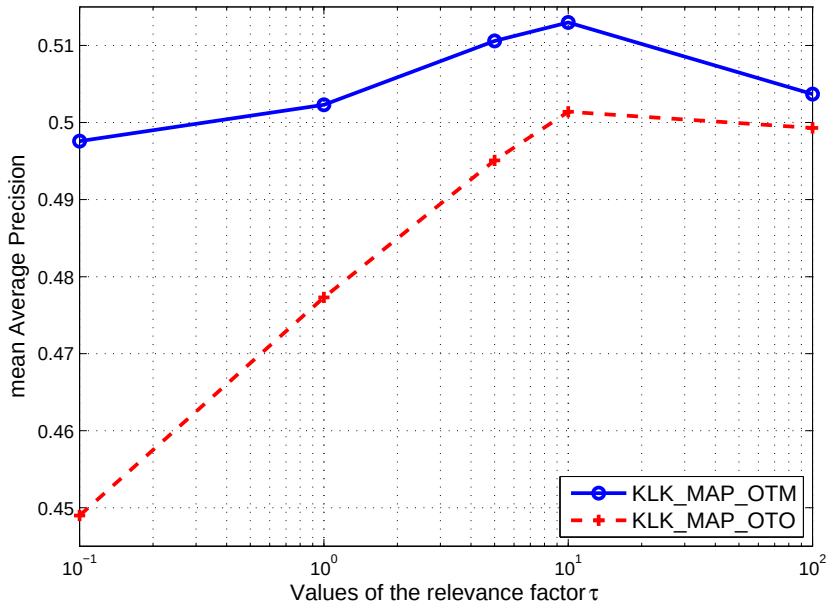


Figure 4.2: The mean AP based on ORH+COL features with KLK on the VOC2007 database as a function of parameter τ. We used GMMs with 16 Gaussians for this experiment.

- First τ *influences the robustness of the estimation.* We can consider two extreme cases. When $\tau = 0$, MAP turns into MLE and the parameters are not estimated robustly as was shown in the previous experiments. When $\tau = \infty$, the image GMMs remain equal to the universal model. As the distance between any pair of images is constant no kernel classifier can be learned. The best performance will thus be obtained when an intermediate value between these two extremes is chosen.

- Second τ *impacts the proposed fast scoring.* Indeed, our fast scoring is only possible if there is a one-to-one correspondence between the Gaussians of two adapted GMMs. The strength of the correspondence will depend on τ. If $\tau = \infty$, the correspondence is maximized and the one-to-one mapping is exact. When $\tau = 0$, the correspondence is weaker.

Hence, the τ which optimizes the robustness ($0 < \tau < \infty$) is necessarily different from the τ which optimizes the Gaussian correspondence ($\tau = \infty$).

We present the result in Figure 4.2. This analysis was performed on the 16 Gaussians model using the KLK kernel. We can see that for small values of τ MAP_OTM outperforms MAP_OTO. This shows that, when τ is small the correspondence between the Gaussian of two adapted models is loose and that our one-to-one assumption is too naïve. However, as expected, as τ increases to more reasonable values, the difference between the two systems becomes narrower. For

both systems the best performance is obtained for $\tau = 10$.

4.3.2.3 Cross database experiments

As the estimation of the image models with MAP relies on the a priori information contained in the universal model, it is important to understand how the performance of our approach is affected when the universal model is learned on another dataset. The alternate dataset we used to learn the visual vocabulary contains 120,000 unannotated images from a printing workflow of photo albums. We had a look at a small sample of these images to try to understand whether they were representative of the 20 categories found in the PASCAL VOC2007 database. While this set of images contains a very large number of photos of persons, it seems to contain very few (if no) occurrences of the 19 other classes. Hence, we believe that there is a strong mismatch between this dataset and VOC2007. To learn a universal vocabulary, we took a random sub-sample of 2,000 images. This experiment was repeated 10 times with 10 different subsamples. We restricted this analysis to the case where we employ the fast one-to-one scoring.

For the PPK and the KLK kernels, the average precision with the standard deviation were 0.542±0.002 and 0.538±0.002 respectively. Clearly, the proposed approach does not seem to be sensitive to the set of images used to train the universal model. Hence the same universal model can be used across different category sets. This is a clear advantage when one grows a category set incrementally as one does not need to relearn the universal GMMs, and thus the image GMMs, every time a new category is added.

4.3.2.4 Computational Cost

We now perform a brief analysis of the computational cost of the proposed approach. For this analysis, we considered GMMs containing 128 Gaussians. The following durations were measured on a 3.1 GHz Xeon^{TM}machine.

The cost of training the GMM of one image with MLE using the iterative strategy of [Young *et al.* 2002] is approximately $430ms$ while it is only $15ms$ for MAP. We recall that this difference is due to the greater number of EM iterations required for MLE compared to MAP. Note that, instead of the iterative approach of [Young *et al.* 2002], we could have used the alternative strategy which consists in starting from multiple random initializations of the parameters and picking the best one, *i.e.* the one which leads to the highest log-likelihood. However, the cost of this alternative would have been even greater.

We now consider the cost of the kernel computations. On the VOC2007 database, classifying one image takes approximately $240s$ for the PPK and $31s$ for the KLK. We estimated it would take almost 28 days to run the experiments with the PPK (128 Gaussians) on this dataset (9,963 images in total). As for the proposed fast scoring based on the one-to-one correspondence, the classification cost is reduced to $2.0s$ for PPK and $0.4s$ for KLK. These figures are consistent with the fact that, for both kernels, we expect the proposed one-to-one scoring to be 128 times faster than the one-to-many scoring when GMMs contain 128 Gaussians (linear versus quadratic cost).

suburb (98.5%) | forest (94.1%) | mountain (89.9%) | kitchen (70.0%) | livingroom (62.5%) | industrial (59.7%)

Figure 4.3: Image examples of the best and the worst Scene-15 classes. On the left the three classes with the best classification rates and on the right those with the lowest rates.

4.3.3 Scene-15

For the Scene-15, we only consider the ORH features as the database is only available in grayscale. We choose 100 images per class for training and used the rest for testing which is the standard evaluation procedure [Lazebnik *et al.* 2006]. We made use of 128 Gaussians to run our PPK and KLK systems. For the BOV baseline with χ^2 kernel, we considered 4096 Gaussians. The final result was reported as the mean and standard deviation of the results from 10 individual runs.

We obtain a 80.4% accuracy (0.7% standard deviation) on the KLK system and 79.0% (0.8% standard deviation) on the PPK system. Both results are better than that of BOV baseline: 78.4% (0.9% standard deviation). Our proposed systems are close to the state-of-the-art reported in [Lazebnik *et al.* 2006] (81.4% with 0.5% standard deviation) which makes use of the Spatial Pyramid Matching. We do not use geometric information. Our results can also be compared to the very recent results of [Zhou *et al.* 2009]. They report a 75.8% of classification accuracy using a GMM-based modeling of images and up to 85.2% using additional geometric information. Figure 4.3 presents the categories with the best and worst classification rates according to the result of our KLK approach.

4.3.4 Caltech-256

For the Caltech-256, we only considered the ORH features as well because many works report results only in grayscale. Our results are evaluated by using 10, 20 and 50 images per class respectively for training and 25 images per class for testing. We consider GMMs containing 128 Gaussian components with our proposed systems and compare to the baseline BOV_CHI2 (4096 Gaussians). We report the mean and standard deviation of classification rate over 5 individual runs.

Figure 4.4 shows clearly the advantage of using our PPK approach over the BOV

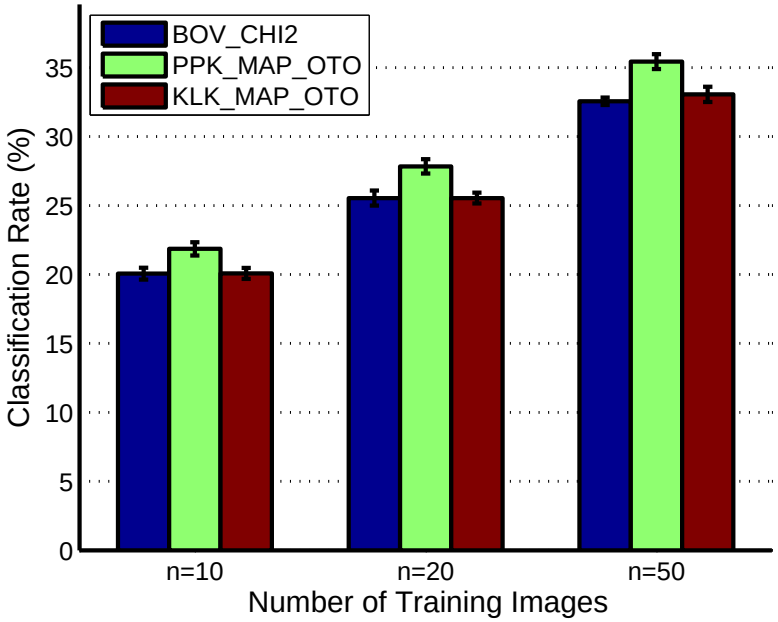


Figure 4.4: Classification rate (%) on Caltech-256 database with 128 Gaussian components by varying the number of training images.

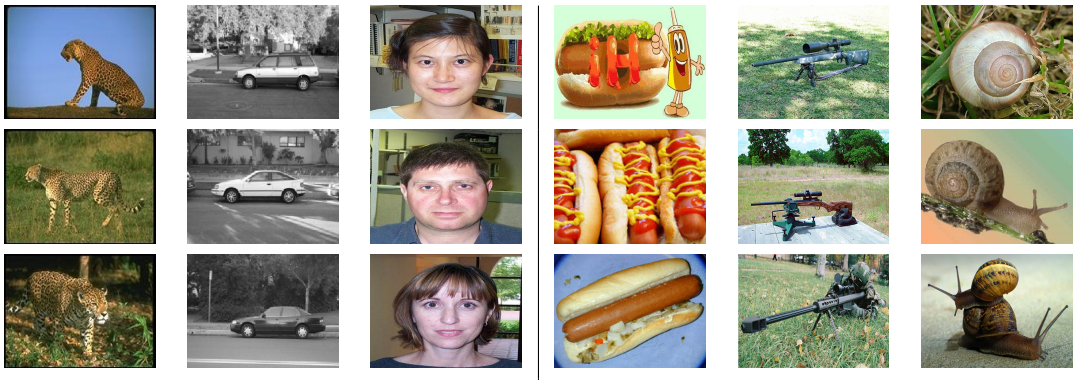


Figure 4.5: Image examples of the best and the worst Caltech-256 classes. On the left the three classes with the best classification rates and on the right those with the lowest rates.

baseline. However, the KLK approach gains no improvement over the baseline. We obtain a 35.4% (0.5% standard deviation) with 50 training images on PPK which is close to the result reported by Lazebnik *et al.* (39.0% of accuracy in the challenge of [Griffin 2007]). Figure 4.5 shows sample images for the categories with the best and worst classification rate based on the PPK approach.

4.4 Conclusion

In this chapter, we introduced a novel approach to compute the similarity between two unordered vector sets. The main contribution was to model each vector set with a generative model – a GMM in our case – adapted from a common universal model using MAP. We showed that this adaptation framework offers two major advantages compared to the case where the distributions are trained with MLE. First MAP provides a more accurate estimate compared to MLE when the cardinality of the vector sets is small. Second, there is a one-to-one correspondence between the components of adapted mixture models which may be used for fast scoring. This correspondence was used to derive efficient approximations for two kernels on distributions: the probability product kernel and the Kullback-Leibler kernel.

This approach was applied to the image categorization problem. We showed that this approach is practical. First, the classification cost is reasonable. Second, the a priori information contained in the universal model needs not to be perfectly representative of the category set under consideration to obtain good results.

Future work could consider the use adaptation techniques other than MAP. Especially, techniques such as maximum likelihood linear regression (MLLR) [Leggetter & Woodland 1995, Gales 1998], cluster adaptive training (CAT) [Gales 2000] or "eigenvoices" [Kuhn *et al.* 2000] have been shown to yield significantly better results than MAP in the speech recognition literature when the amount of adaptation data is extremely scarce. We also note that the PPK outperforms the KLK on VOC2007 and Caltech-256 and the KLK outperforms the PPK on Scene-15. This seems to indicate that no kernel always outperforms the other. To avoid the per-database selection of kernel, we could consider principled combinations of kernels using Multiple Kernel Learning (*e.g.* [Varma & Ray 2007]).

Chapter 5

Images as Mixtures of Reference Images

Contents

5.1 Introduction

One limitation of previous probabilistic kernels between two GMMs such as the KL and the PPK is that they measure a global match of distributions. In this chapter, we propose to approximate an image, modeled as a GMM, as a convex combination of K reference image GMMs and then to describe the image as the K-dimensional vector of mixture weights. The computed weights encode a similarity that favors local matches (*i.e.* matches of individual Gaussians) and is therefore fundamentally different from the KL or PPK. The vector of mixture weights may then be used as input to a discriminative classifier for categorization.

Our approach is related to dissimilarity-based learning which is an alternative to traditional kernel-based learning. In [Pekalska *et al.* 2002], Pekalska *et al.* propose to represent an object as a vector of distances with respect to a set of reference objects. The main difference with ours is that in [Pekalska *et al.* 2002] each reference object contributes independently to the representation. For instance, if we use the KL as a measure of distance, the distance-based representation will be plagued with the limitations of KL. In our case the reference images contribute jointly to the

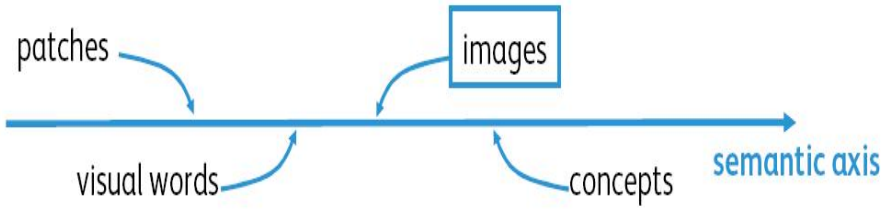


Figure 5.1: Comparison of different representations on the semantic axis.

image representation. This results in a measure of similarity which better takes into account strong matches.

Our approach can also be related to [Quelhas *et al.* 2005, Bosch *et al.* 2006, Rasiwasia *et al.* 2007, Rasiwasia & Vasconcelos 2008]. While the BOV represents an image as a vector of posterior visual word probabilities (when using probabilistic vocabularies), these papers propose to represent an image as a vector of posterior *concept* probabilities. The assumption is that concepts are more semantically meaningful than visual words. These concepts may be learned in an unsupervised fashion [Quelhas *et al.* 2005, Bosch *et al.* 2006], in which case there is no guarantee that they are semantically meaningful, or in a supervised manner [Rasiwasia *et al.* 2007, Rasiwasia & Vasconcelos 2008] which requires large amounts of training material. Our work is significantly different from those as we score images with respect to other images, not reference concepts. Our intuition is that reference images might be less semantically meaningful than concepts learned in a supervised manner, but more meaningful than visual words. Figure 5.1 displays this intuition.

The rest of this chapter is organized as follows. In section 5.2, we analyze the limitations of traditional approaches between GMMs through a toy example. This leads us to introduce our novel image representation in section 5.3. We show that the vector of mixture weights can be computed through the optimization of a convex objective function. As the direct optimization is difficult, we propose two possible approximations: the first one based on sampling, the second one based on a variational bound of the objective function. We also discuss convergence issues. In section 5.4 we provide experimental results showing that the proposed framework outperforms a standard kernel-based classifier employing the KLK or the PPK. Finally, we draw our conclusions in section 5.5.

5.2 Toy Example

Let $\mathcal{N}(\mu, \sigma)$ denote the one dimensional Gaussian with mean μ and standard deviation σ. Let us consider the following toy example. Let q be a mixture of two Gaussians:

$$q = \frac{1}{2}\mathcal{N}(+2, 1) + \frac{1}{2}\mathcal{N}(-2, 1). \tag{5.1}$$

We will compare the SKL and PPK between q and three distributions:

$$p_1 = \mathcal{N}(-2,1) \ , \tag{5.2}$$
$$p_2 = \mathcal{N}(2,1) \ , \tag{5.3}$$
$$p_3 = \frac{1}{2}\mathcal{N}(2+\delta,1) + \frac{1}{2}\mathcal{N}(-2-\delta,1). \tag{5.4}$$

where δ is a non-negative value (*c.f.* Figure 5.2). To study the limitations of the SKL (*c.f.* formula 2.20) and the BHA (*c.f.* formula 2.18) (and not of their approximations), we estimated the measures numerically, replacing the integral by a sum over many tiny intervals.

We have $SKL(q,p_1) = SKL(q,p_2)$ and $BHA(q,p_1) = BHA(q,p_2)$ by symmetry. $SKL(q,p_3)$ (resp. $BHA(q,p_3)$) is an increasing (resp. decreasing) function of δ with $SKL(q,p_3) = 0$ (resp. $BHA(q,p_3) = 1$) if $\delta = 0$. We are interested in the value δ_{SKL} such that $SKL(q,p_1) = SKL(q,p_2) = SKL(q,p_3)$ and δ_{BHA} such that $BHA(q,p_1) = BHA(q,p_2) = BHA(q,p_3)$. We found numerically $\delta_{SKL} \approx 2.0$ and $\delta_{BHA} \approx 1.5$. The value δ_{SKL} was chosen to represent p_3 on Figure 5.2. We can see that with such a value, while q and p_3 share a similar shape (bimodal) they are significantly different. On the other hand p_1 and p_2 perfectly match one of the Gaussian components of q but are strongly penalized because they match a single component.

Let us now try to translate what this toy example means in the image domain. Even if there is a strong match between the components of two images, *e.g.* the two images contain the same object, the SKL (resp. the PPK) might be large (resp. small) because the object occurs in different backgrounds or because it is occluded in one of the two images.

5.3 Mixtures of Mixtures

Let $q = \sum_{i=1}^{N} \pi_i q_i$ be the GMM that models the image we want to describe. N denotes the number of Gaussian components, π_i is the mixture weight for Gaussian i and q_i is the i-th Gaussian component. Let $\{p_k, k = 1...K\}$ be a set of K reference GMMs, each one modeling a reference image. We write $p_k = \sum_{j=1}^{N_k} \pi_{k,j} p_{k,j}$ where N_k denotes the number of Gaussian components in p_k, $\pi_{k,j}$ is the mixture weight for Gaussian j and $p_{k,j}$ is the j-th Gaussian component.

Our goal is to approximate q as a convex combination of p_k's. Let ω_k denote the mixture weight associated with p_k. We choose the optimal ω_k's as those which minimize the KL between q and $\sum_{k=1}^{K} \omega_k p_k$. This is equivalent to maximizing the following objective function:

$$E = \int_{x \in \mathbb{R}^D} q(x) \log \left(\sum_{k=1}^{K} \omega_k p_k(x) \right) dx \ . \tag{5.5}$$

under the constraints

$$\omega_k \geq 0 \ , \forall k \text{ and } \sum_{k=1}^{K} \omega_k = 1. \tag{5.6}$$

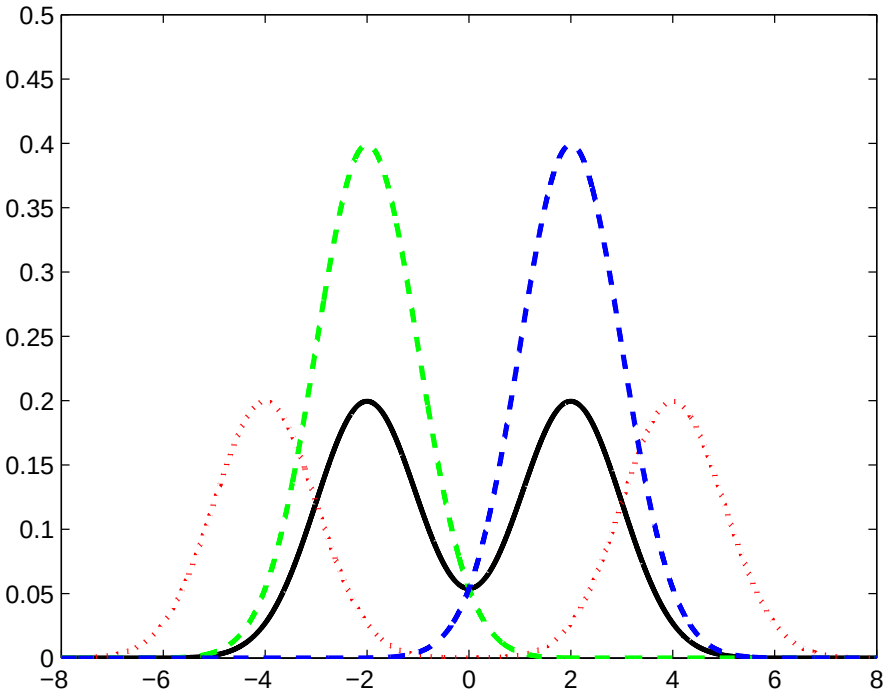


Figure 5.2: The SKL between q (black straight line) and p_1 or p_2 (green and blue dashed lines respectively) is approximately the same as the SKL between q and p_3 (dotted red line).

This is a convex optimization problem which can be solved iteratively using the Expectation-Maximization (EM) algorithm [Dempster *et al.* 1977]. The E-step consists in computing the occupancy probability $\gamma_k(x)$ *i.e.* the probability that observation x was generated by the k-th reference image:

$$\gamma_k(x) = \frac{\omega_k p_k(x)}{\sum_{j=1}^{K} \omega_j p_j(x)} \ . \tag{5.7}$$

The M-step leads to the following estimate:

$$\hat{\omega}_k = \int_{x \in \mathbb{R}^D} q(x) \gamma_k(x) dx. \tag{5.8}$$

However, the computation of the previous integral is difficult as there is no closed form formula for ratios of GMMs. We consider two possible approximations using: (i) a sampling method and (ii) a lower-bound method.

5.3.1 Sampling Approximation

Let $\{X = x_t, t = 1...T\}$ be a set of T vectors distributed according to q. This might be a set of feature vectors drawn from q (Monte-Carlo sampling). This might also

be the set of low-level feature vectors directly extracted from the image we want to characterize.

If the number of samples T is large enough, we can use the law of large numbers and approximate the objective function (5.5) as follows:

$$E \approx \frac{1}{T} \sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} \omega_k p_k(x_t) \right). \tag{5.9}$$

This remains a convex objective function which can be optimized with respect to the ω_k's using the EM algorithm. The E-step consists in computing the values $\gamma_k(x_t)$ for each sample x_t and each reference GMM p_k. The M-step gives the following estimates:

$$\hat{\omega}_k = \frac{1}{T} \sum_{t=1}^{T} \gamma_k(x_t). \tag{5.10}$$

We note that we would have obtained the same re-estimation formula if we had applied the law of large numbers on equation 5.8 directly.

5.3.2 Lower-Bound Approximation

As explained in the previous sub-section, the mixture weights ω_k can be estimated directly from the low-level features extracted from the image to be described as in a Maximum Likelihood Estimation (MLE) framework the samples used to estimate q are supposed to be distributed according to q. In such a case there is no need to estimate q, which might be seen as an advantage of the sampling approximation. However, we will see that it can be beneficial to estimate q for two main reasons:

- The first reason is a practical one. If we want the approximation (5.9) to be reasonably good, T should be large enough which can result in a high computational cost during the E-step at the number of Gaussian computations grows linearly with T.
- Secondly, one can incorporate a-priori information in the model q. In chapter 4, we proposed to estimate the per-image GMMs through the adaptation of a "universal" GMM using the Maximum a Posteriori (MAP) criterion. This was shown to have two advantages. First MAP estimation leads to a more robust estimate of parameters than MLE in the case of scarce training data. Second, there is a correspondence between the Gaussians of two GMMs adapted from a common distribution and one can take advantage of this fact to speed-up the similarity computation.

We first present the estimation of the ω_k's in the general case, *i.e.* whatever the criterion used to estimate q. We then show how it can be speeded-up using the framework of chapter 4.

We rewrite the objective function (5.5) as follows:

$$E = \sum_{i=1}^{N} \pi_i \int_{x \in \mathbb{R}^D} q_i(x) \log \left(\sum_{k=1}^{K} \omega_k \sum_{j=1}^{N_k} \pi_{k,j} p_{k,j}(x) \right) dx. \tag{5.11}$$

We use the idea proposed by Hershey and Olsen [Hershey & Olsen 2007] to approximate the KL divergence between two GMMs and introduce a set of variational parameters $\gamma_{i,k,j}$ which are subject to the constraints:

$$0 \leq \gamma_{i,k,j} \leq 1 \text{ and } \sum_{k=1}^{K} \sum_{j=1}^{N_k} \gamma_{i,k,j} = 1. \tag{5.12}$$

The function (5.11) becomes:

$$E = \sum_i \pi_i \int_{x \in \mathbb{R}^D} q_i(x) \log \left(\sum_{k,j} \gamma_{i,k,j} \frac{\omega_k \pi_{k,j} p_{k,j}(x)}{\gamma_{i,k,j}} \right) dx. \tag{5.13}$$

Applying Jensen's inequality to the concave log-function, we obtain the following lower-bound:

$$E \geq \sum_i \pi_i \int_{x \in \mathbb{R}^D} q_i(x) \sum_{k,j} \gamma_{i,k,j} \log \left(\frac{\omega_k \pi_{k,j} p_{k,j}(x)}{\gamma_{i,k,j}} \right) dx \ . \tag{5.14}$$

Maximizing the lower-bound with respect to $\gamma_{i,k,j}$'s leads to the following bound:

$$E \geq \sum_i \pi_i \log \left(\sum_{k,j} \omega_k \pi_{k,j} \exp(-H_{i,k,j}) \right). \tag{5.15}$$

where $H_{i,k,j}$ is defined as follows:

$$H_{i,k,j} = H(q_i, p_{k,j}) = -\int_{x \in \mathbb{R}^D} q_i(x) \log p_{k,j}(x) dx. \tag{5.16}$$

$H_{i,k,j}$ is the cross-entropy between q_i and $p_{k,j}$ and we recall that it can be computed in closed form in the case where q_i and $p_{k,j}$ are Gaussians.

We propose to compute the set of ω_k's which optimize the bound on E rather than E. One more time, this is a convex optimization problem which can be solved with an EM-like algorithm. The E-step consists in computing the values $\gamma_{i,k,j}$ that maximize the bound:

$$\gamma_{i,k,j} = \frac{\omega_k \pi_{k,j} \exp(-H_{i,k,j})}{\sum_{k,j} \omega_k \pi_{k,j} \exp(-H_{i,k,j})} \tag{5.17}$$

Taking the derivative with respect to ω_k and equating it to zero leads to the M-step:

$$\hat{\omega}_k = \sum_{i,j} \pi_i \gamma_{i,k,j}. \tag{5.18}$$

This shows that our similarity computation takes into account the cross-entropy between the individual Gaussians, as is the case of the KL approximation between two GMMs (*c.f.* equation 4.23). However, our measure of similarity is fundamentally different. $\gamma_{i,k,j}$ is a measure of soft-matching between the Gaussian components q_i and $p_{k,j}$. Hence, the optimal ω_k measures the number of soft matches between the

components of q and the components of p_k. This point will be made clearer in the next subsection.

The cross-entropy computations dominate the cost of the EM algorithm. EM requires the computation of $N \times \sum_{k=1}^{K} N_k$ cross-entropies which is comparable to the cost of computing K KL divergences between GMMs. If we make use of the framework of chapter 4, all GMMs are trained through the adaptation of a common GMM which contains N Gaussians ($N_k = N$). In such a case, we can use the fact that there is a correspondence between the Gaussian components of two GMMs adapted from the same GMM, *i.e.* that $H_{i,k,j}$ is small if $i = j$ and large if $i \neq j$. This means that $\gamma_{i,k,j} \approx 0$ if $i \neq j$. The previous approximation reduces the cost to $N \times K$ cross-entropy computations.

5.3.3 Convergence Issues

Let us go back to our toy example of section 5.2. We want to approximate q as a convex combination of p_1, p_2 and p_3. As we have $q = \frac{1}{2}p_1 + \frac{1}{2}p_2$, it is trivial to see that the optimal weights that maximize the objective function (5.11) are $\omega_1 = \omega_2 = \frac{1}{2}$ and $\omega_3 = 0$ in the case where $\delta > 0$ (if $\delta = 0$ there is an infinite number of solutions). Hence, $\omega_3 = 0$ whether δ is very large, meaning that q and p_3 are very different, or δ is very small, meaning that q and p_3 are near-identical. Although the perfect matching of Gaussian components, as is the case of our toy example, happens seldom, this shows that our objective function might give too much weight to the near perfect matching of Gaussians, as opposed to SKL or PPK which give too much weight to a global match. Clearly, *the optimal solution is a balance between global and local matching.*

A simple solution that we found to be very effective to find a middle-ground between these two extreme situtations is early stopping, *i.e.* stopping EM after few iterations. An important fact is that early stopping does not change the ranking of the ω's (this property was observed empirically and a formal proof is under investigation). The larger δ, the faster ω_3 will converge to zero.

Early stopping solves also the problematic case where q belongs to the reference distributions. This happens in our image categorization scenario when the reference images are the set of labeled images. If $q = p_j$, then our objective function (5.5) is maximized by $\omega_j = 1$ and $\omega_i = 0, \forall i \neq j$. This undesired effect is prevented by early stopping.

Note that we experimented with more elaborate strategies than early stopping. For instance, adding a Dirichlet prior on the parameters ω and performing MAP estimation of the ω's rather than MLE would also avoid the ω's to go to zero. However, we did not find this approach to be more effective than early stopping.

5.3.4 Beyond KL

As explained in section 5.3.2, the mixture weights ω_k are based on the cross-entropy between individual Gaussians. It would be interesting to extend this framework to other measures such as the Bhattacharyya similarity. A heuristic would for instance consist in replacing $\exp(-H_{i,k,j})$ by $B_{i,k,j} = K_{bha}(q_i, p_{k,j})$ in the E-step (5.17).

A more principled approach consists in modifying the objective function. Instead of minimizing the KL between q and $\sum_{k=1}^{K} \omega_k p_k$, we propose to maximize their Bhattacharyya similarity:

$$E = \int_{x \in \mathbb{R}^D} \sqrt{q(x)} \sqrt{\sum_{k=1}^{K} \omega_k p_k(x)} dx \tag{5.19}$$

$$= \int_{x \in \mathbb{R}^D} \sqrt{\sum_{i=1}^{N} \pi_i q_i(x) \sum_{k=1}^{K} \omega_k \sum_{j=1}^{N_k} \pi_{k,j} p_{k,j}(x)} dx. \tag{5.20}$$

This leads to a convex objective function which is difficult to optimize directly. One more time, we can optimize a bound on the true objective function rather than the objective function itself. We apply a first time Jensen's inequality and write:

$$E \geq \sum_i \pi_i \int_{x \in \mathbb{R}^D} \sqrt{q_i(x) \sum_k \omega_k \sum_j \pi_{k,j} p_{k,j}(x)} dx. \tag{5.21}$$

We then introduce a set of variables $\gamma_{i,k,j}$ which are subject to the constraints: $0 \leq \gamma_{i,k,j} \leq 1$ and $\sum_{k,j} \gamma_{i,k,j} = 1$. The bound becomes:

$$\sum_i \pi_i \int_{x \in \mathbb{R}^D} \sqrt{q_i(x) \sum_{k,j} \gamma_{i,k,j} \frac{\omega_k \pi_{k,j} p_{k,j}(x)}{\gamma_{i,k,j}}} dx. \tag{5.22}$$

Applying again Jensen's inequality we obtain the following lower-bound:

$$E \geq \sum_i \pi_i \sum_{k,j} \sqrt{\omega_k \pi_{k,j} \gamma_{i,k,j}} B_{i,k,j}. \tag{5.23}$$

where $B_{i,k,j}$ is the Bhattacharyya similarity between the two Gaussians q_i and $p_{k,j}$.

Computing derivatives with respect to $\gamma_{i,k,j}$ and ω_k and equating them to zero leads respectively to the E- and M-step.

E-step:

$$\gamma_{i,k,j} = \frac{\omega_k \pi_{k,j} B_{i,k,j}^2}{\sum_{k,j} \omega_k \pi_{k,j} B_{i,k,j}^2} \ . \tag{5.24}$$

M-step:

$$\hat{\omega}_k = \frac{(\sum_i \pi_i \sum_j \sqrt{\pi_{k,j} \gamma_{i,k,j}} B_{i,k,j})^2}{\sum_k (\sum_i \pi_i \sum_j \sqrt{\pi_{k,j} \gamma_{i,k,j}} B_{i,k,j})^2} \ . \tag{5.25}$$

Preliminary experiments showed that the principled computation of weights always outperformed the heuristic approach.

Again, we use the one-to-one correspondence to speed-up the scoring and set $B_{i,k,j} = 0$ for $i \neq j$.

5.4 Experiments

5.4.1 Experimental Setup

We make use of the same experimental setup as in section 4.3.1: including the extraction of features, dimension reduction through PCA, classification using SLR classifier.

We evaluated two baseline systems:

- The method of chapter 4 with the KLK.
- The method of chapter 4 with the Bhattacharyya Kernel (PPK with $\rho = 1/2$).

For two baselines, a "universal" GMM is first estimated with all training images. Then the per-image GMMs are estimated through MAP adaptation of the universal GMM. For both the baselines KLK and PPK and the method proposed in this chapter, we used the one-to-one fast scoring described in chapter 4. For the two baselines, an image is represented as a vector of similarities/distances to the set of training images.

5.4.2 PASCAL VOC2007

We compared these two baselines to the three versions of our approach (later referred to as MOM for mixture of mixtures):

- MOM KL sampling: *c.f.* section 5.3.1.
- MOM KL lower-bound: *c.f.* section 5.3.2.
- MOM PPK: *c.f.* section 5.3.4.

We used as reference images the 5,011 training images. For the three proposed approaches, we apply SLR classifier directly to the vectors of mixture weights.

Lower-bound vs sampling. We start with the comparison of the sampling and lower-bound approximations for MOM KL. For the sampling approximation, we used directly the low-level features extracted from the image to be described as samples. Results are shown on Figure 5.3 for the system based on ORH features as a function of the number of Gaussian components in the per-image GMMs. Similar results were obtained for the COL features. The lower-bound approximation clearly outperforms the sampling one. We believe that this difference can be explained by the a priori information incorporated in q in the case of the lower-bound approximation. In the following, we will not consider the MOM KL sampling approximation anymore.

Influence of the number of EM iterations. We now study the influence of the number of EM iterations on the performance of our algorithm. This is shown on Figure 5.4 for the system based on ORH features as a function of the number of Gaussian components in the per-image GMMs. Similar results were obtained for MOM PPK or for COL features. In all cases, the best results are obtained for 3 to 5 iterations. With more than 5 iterations, the accuracy decreases quite rapidly. This demonstrates the importance of early stopping. This shows that the best measure should not only consider global or local similarity but a mix between the two.

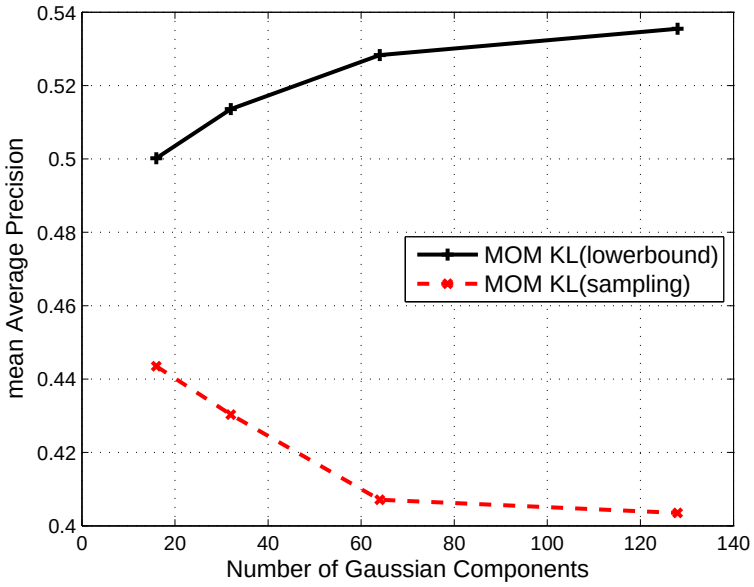


Figure 5.3: Mean AP for the sampling and lower-bound approximations of MOM KL for the system based on ORH features only.

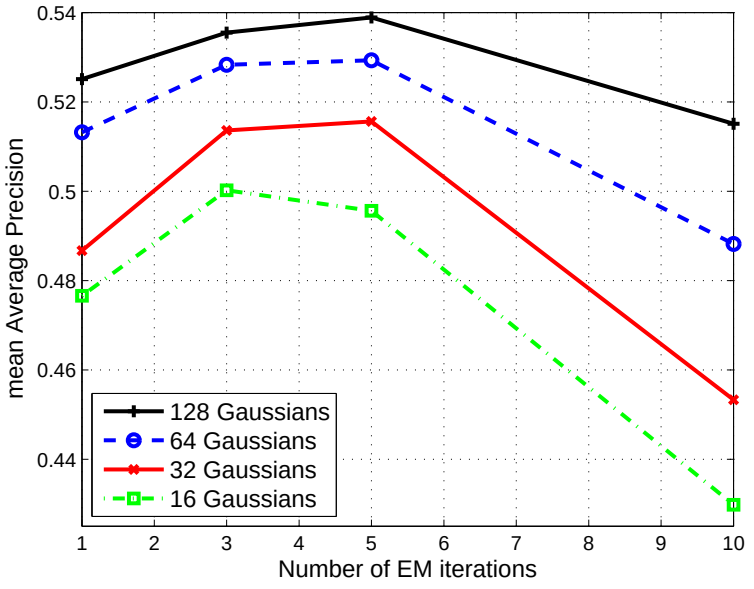


Figure 5.4: Influence of the number of EM iterations on the mean AP for MOM KL lower-bound for different numbers of Gaussian components (system based on ORH features only).

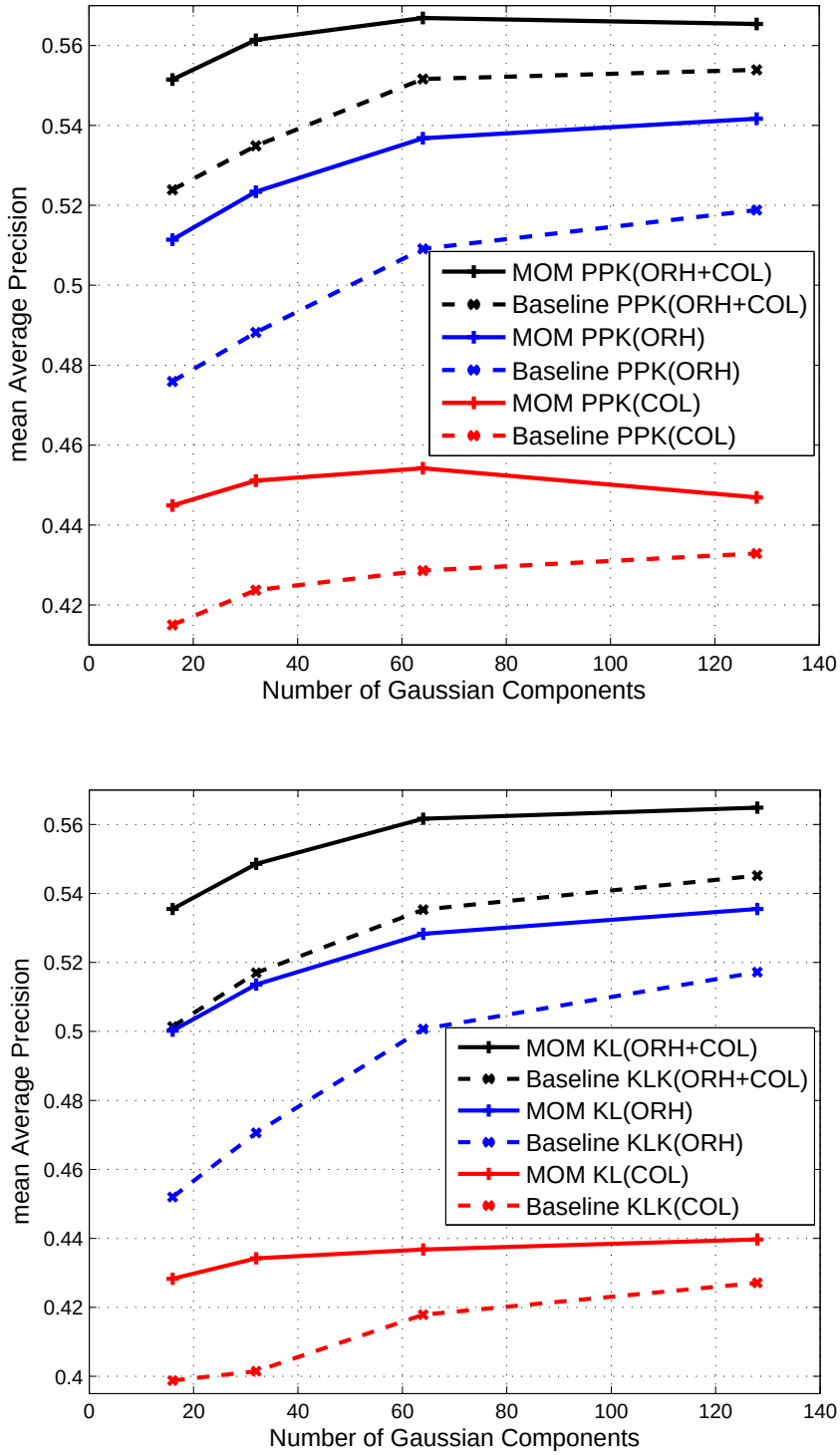


Figure 5.5: Comparison of the proposed algorithms to traditional kernel methods: MOM PPK versus PPK (top) and MOM KL versus KLK (bottom). The mean AP is shown as a function of the number of Gaussian components in the per-image GMMs for the different features (ORH, COL, ORH+COL).

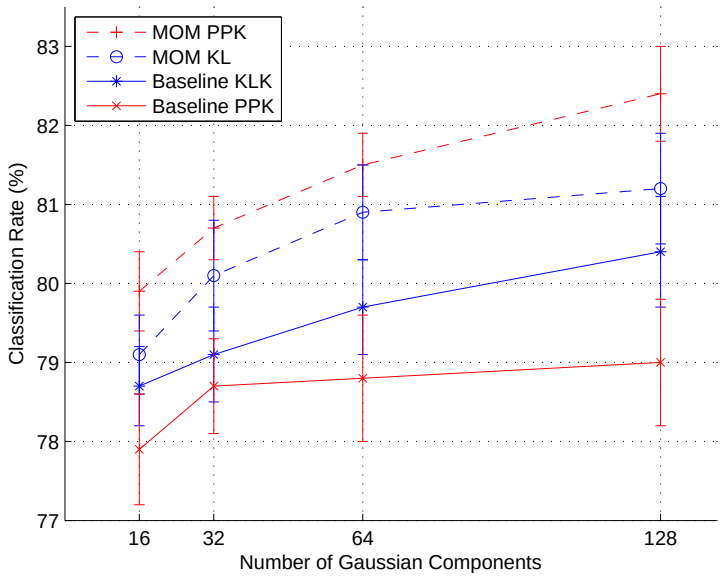


Figure 5.6: Classification rate (%) on Scene-15 database by varying the number G of Gaussian components

Comparative evaluation. The results of the comparison of KLK with MOM KL and PPK with MOM PPK are shown on Figure 5.5 for the different features.

We can see that the proposed method consistently outperforms the baseline for all feature types, for both KL and PPK and for various numbers of Gaussians. We note that the difference is more pronounced for KL than it is for PPK. We believe that this is because PPK is more resilient than KLK to the poor matching of individual Gaussian components (*c.f.* the toy example in section 5.2).

We note that an alternative to the proposed approach would have been to model an image, not as vector of similarities/distances to N reference/training images, but as a vector of $K^2 \times N$ similarities/distances between the K components of the image to be described and the $K \times N$ components of the N reference images. Using our proposed framework of chapter 4 (i.e. taking into account the correspondence between adapted Gaussians), we can reduce the vector size to $K \times N$. Our initial intuition was that, since this representation contains K times more information than the proposed representation, it should perform better. However in practice, this approach performed worse than the proposed approach. Our best explanation is the very high dimensionality of the vectors: 640,000 dimensions for $N = 5,000$ and $K = 128$.

5.4.3 Scene-15

With the same experimental setup as in 4.3.3, we evaluate our approaches based on lower-bound approximation on Scene-15. Figure 5.6 shows the average classification

rate over 10 individual runs. We compare our results with the two baseline systems and observe that our lower-bound approximations outperform the baselines. Our best classification rate is 82.4% (0.6% standard deviation)with the MOM PPK and it is better than the result reported 81.4% in [Lazebnik *et al.* 2006] and still below [Zhou *et al.* 2009] which makes use of geometric information (85.2%).

5.4.4 Caltech-256

We report the average classification rate over 5 individual runs on Caltech-256 database. Figure 5.7 shows that MOM KL outperforms the baseline KLK and obtains a 38.4% accuracy (0.4% standard deviation) which is much closer to the state-of-the-art of Lazebnik *et al.* (39.0% of accuracy) reported in the challenge [Griffin 2007]. We also observe that MOM PPK gives better result than the baseline PPK when using more training images. In the case of lower-bound approximation, the MOM KL is better than the MOM PPK.

5.5 Conclusion

We presented in this chapter a novel image representation. The idea was to approximate an image, modeled by a GMM, as a convex combination of K reference image GMMs and then to describe the image as the K-dimensional vector of mixture weights. We explained that these mixture weights encode a similarity which favors strong local matches of Gaussians components rather than a global match of the distribution, as is the case of traditional distance/similarity measures such as the SKL or PPK.

We applied this framework to image classification and showed on several databases a consistent increase for classification accuracy. This increase seems more important on those datasets where the background influence is reduced (*e.g.* Scene-15 and Caltech-256). In the next chapter, we discuss the limitations of our previous work and propose a solution to go beyond those limitations.

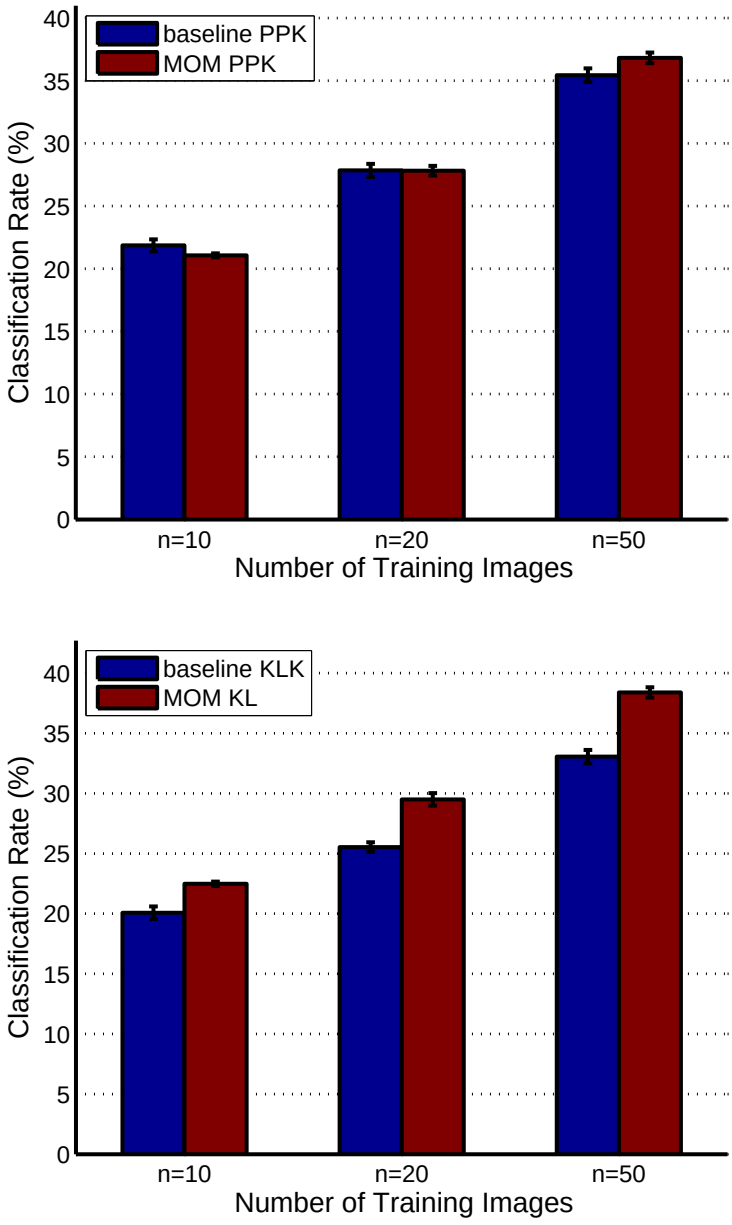


Figure 5.7: Classification rate (%) on Caltech-256 database of the MOM PPK (top) and MOM KL (bottom) with 128 Gaussian components with different number of training images.

Chapter 6

A Family of Contextual Measures of Similarity Between Distributions

Contents

6.1 Introduction

In the previous chapter, we modeled images as mixtures of reference images. We explained that each mixture weight expressed a similarity between the image to be described and each reference image. We can say that this measure of similarity is contextual as it depends on the set of reference images: the context. However, finding the similarity between two images using such an explicit set of context images is both costly and cumbersome.

We note that several authors have already proposed contextual measures of similarity. [Jégou *et al.* 2007] proposes a contextual measure of distance between points which consists in symmetrizing the K-NN relationship. The initial distance is contextualized by adding a multiplicative penalty term which can be computed iteratively. In effect, it downweights those images which are located in a dense region of the image space. [Zhao *et al.* 2007] proposes a contextual distance between data points which is defined as the difference of their contributions to the integrity

of the structure of the contextual set (defined as the neighboring points). Two measures of structural integrity were proposed: a geometric one and an informative one respectively based on the center of gravity and the coding length of the contextual set.

Of highest interest to us are those measures which were proposed in the information retrieval community. [Ponte & Croft 1998] proposed the so-called language modeling (LM) approach. If p, q and u are respectively a template, a query and a context multinomial of dimension D, one can measure the dissimilarity of q and p in the context of u as the Kullback-Leibler (KL) divergence between q and a *smoothed* version of p:

$$\sum_{i=1}^{D} q_i \log\left(\frac{q_i}{\omega p_i + (1-\omega)u_i}\right). \tag{6.1}$$

Smoothing has two benefits over a standard KL divergence between q and p. First, it avoids log(0) effects in the case of sparse vectors. Second, by rewriting equation 6.1 as:

$$-\sum_{i=1}^{D} q_i \log\left(1 + \frac{\omega}{1-\omega}\frac{p_i}{u_i}\right) + C \tag{6.2}$$

where C is independent of p, we can see that it downweights the influence of frequent words (indices i with large values u_i) as is the case of TF-IDF. A major issue is the sensitivity to the choice of ω (see *e.g.* [Zhai & Lafferty 2001] for a study of the impact of ω as well as different smoothing schemes).

In this chapter, we introduce a novel family of contextual measures of similarity between distributions: the similarity between two distributions q and p is measured in the context of a third distribution u. In our framework any traditional measure of similarity/dissimilarity has its contextual counterpart. We show that for two important families of divergences (Bregman and Csiszár), the contextual similarity computation consists in solving a convex optimization problem. We apply the contextual similarity to the case of continuous distributions and report the comparison of performance with our previous approach MOM KL in chapter 5 on several datasets.

This chapter is organized as follows. In section 6.2 we give the definition of our contextual measure and discuss its properties. In our framework any traditional measure of similarity/dissimilarity has its contextual counterpart. We show that when the measure to be contextualized belongs to one of two important families of divergences (Bregman and Csiszár), the contextual similarity computation consists in solving a convex optimization problem. In section 6.3 we specialize our method to the case of continuous mixture distributions. The idea is similar in spirit of that used in chapter 5. We provide in section 6.4 categorization results. Finally, we draw conclusions in section 6.5.

6.2 Contextual Similarity

We first introduce a broad definition of contextual similarities which is valid for discrete or continuous distributions, parametric or non-parametric distributions,

etc.

6.2.1 Definition

Let p and q be two distributions to be compared and let u be the distribution that models the context. Typically, in our case, p and q will be two GMMs describing two images. Let f be a "traditional" (*i.e.* non-contextual) measure of similarity between distributions. We introduce the following function:

$$\phi_f(\omega; q, p, u) = f\left(q, \omega p + (1-\omega)u\right). \tag{6.3}$$

As we are dealing with distributions, ϕ_f is defined over the interval $0 \leq \omega \leq 1$. We note that in the case where $f(q,p) = E_q[\log p]$, where E_q denotes the expectation under q, $\phi_f(\omega; q, p, u)$ is the distance used in the LM approach to retrieval [Ponte & Croft 1998] (*c.f.* the introduction).

We define the contextual similarity cs_f as:

$$cs_f(q, p|u) = \arg\max_{0\leq\omega\leq 1} \phi_f(\omega; q, p, u). \tag{6.4}$$

cs_f is ill-defined for $p = u$ and we choose the convention $cs_f = 1/2$ in such a case.

The intuition behind this measure of similarity is the following one. By maximizing $\phi_f(\omega; q, p, u)$ over ω, we estimate the mixture of p and u that best approximates q. The weight ω which maximizes $\phi_f(\omega; q, p, u)$ reflects how much p contributes to the approximation, *i.e.* whether q is best modeled by the broad domain information contained in u or the specialized information contained in p. Our similarity is fundamentally different from the traditional LM approach. Especially, there is no parameter tuning required. The difference with the contextual similarity defined in the previous chapter 5 is the following one. While in the previous chapter the similarity depends on an explicit set of reference image, here the similarity depends only implicitly on the set of reference images through u (the reference images are used to train u).

By definition cs_f is guaranteed to have values in the interval $[0, 1]$. We note that $q = p \Rightarrow cs_f(\omega; q, p, u) = 1$ but that the converse does not hold. ϕ_f and thus cs_f are asymmetric in p and q even if f is symmetric, *i.e.* $cs_f(q, p|u) \neq cs_f(p, q|u)$ in general. There exist various ways to symmetrize the contextual similarity if needed. One way is to combine $cs_f(p, q|u)$ and $cs_f(q, p|u)$ using for instance a sum or product rule. Another way is to symmetrize ϕ_f, *e.g.* as follows:

$$\begin{aligned} \phi_f(\omega; q, p, u) &= f\left(q, \omega p + (1-\omega)u\right) \\ &+ f\left(p, \omega q + (1-\omega)u\right). \end{aligned} \tag{6.5}$$

In our experiments, we always made us of the symmetric contextual measure.

6.2.2 Choice of the Function f

We have not yet defined a similarity measure but a family of similarity measures parametrized by the particular choice of the function f. cs_f can thus be understood

as a contextualized version of f. f can be virtually any measure of similarity between distributions. Obviously, f can be a dissimilarity instead of a similarity: this just requires changing the max by a min in (6.4).

Interestingly, not all measures f are good candidates for contextualization. A simple counter-example is the Expected Likelihood (EL) kernel [Jebara & Kondor 2003]: $EL(q,p) = E_q[p] = E_p[q]$. Except in the case where $E_q[p-u] = 0$, it is easy to show that cs_{EL} gives binary values (0/1).

It is advantageous to choose ϕ to be concave (resp. convex) in ω if f is a similarity (resp. dissimilarity) as one is thus guaranteed to have a unique optimum which simplifies the optimization process. In the following, we consider the case of continuous distributions. We show that when f belongs to one of two important families of divergences, ϕ_f is convex in ω.

Bregman divergences. The Bregman divergence between two distributions x and y for a convex function $h : \Omega \to \mathbb{R}$ is defined as:

$$B_h(x,y) = h(x) - h(y) - \langle \nabla h(y), (x-y) \rangle \tag{6.6}$$

where ∇h denotes the gradient vector of h and $\langle .,. \rangle$ the dot product. Intuitively, $B_h(x,y)$ can be understood as the difference between the value of h at point x and the value of the first-order Taylor expansion of h around y evaluated at x. Special cases of Bregman divergences include the Euclidean distance, the Mahalanobis distance, the Kullback-Leibler divergence or the Itakura-Saito divergence.

If $\phi(\omega; q, p, u) = B_h\left(\omega p + (1-\omega)u, q\right)$, then $\phi(\omega; q, p, u)$ is convex in ω. To prove this assertion, it is sufficient to show that the second order derivative is positive. We have:

$$\begin{aligned} &\frac{\partial^2}{\partial \omega^2} B_h\left(\omega p + (1-\omega)u, q\right) \\ &= (p-u)^T \nabla^2 h\left(\omega p + (1-\omega)u\right)(p-u) \end{aligned} \tag{6.7}$$

where $\nabla^2 h$ denotes the Hessian matrix of h and ^{T} the transposition. As h is convex, this quantity is positive by definition and thus ϕ is convex in ω.

We note however that if $\phi(\omega; q, p, u) = B_h\left(q, \omega p + (1-\omega)u\right)$, we cannot conclude on the convexity of ϕ (the second order derivative with respect to ω includes third order derivatives of h).

Csiszár divergences. The Csiszár divergence between two distributions x and y for a convex function $h : \mathbb{R} \to \mathbb{R}$ is given by:

$$f_h(x,y) = \int_{z \in \mathbb{R}^D} x(z) h\left(\frac{y(z)}{x(z)}\right) dz. \tag{6.8}$$

where z is an observation vector. Special cases of Csiszár divergences include the Manhattan distance, the Kullback-Leibler divergence, the Hellinger distance or the Rényi divergence.

If $\phi(\omega; q, p, u) = f_h\left(q, \omega p + (1-\omega)u\right)$, then $\phi(\omega; q, p, u)$ is convex in ω. One more time, it is sufficient to show that the second order derivative is positive. We

have:

$$\begin{aligned} &\frac{\partial^2}{\partial \omega^2} f_h\left(q, \omega p + (1-\omega)u\right) \\ &= \int_{z \in \mathbb{R}^D} \frac{(p(z)-u(z))^2}{q(z)} h''\left(\frac{\omega p(z) + (1-\omega)u(z)}{q(z)}\right) dz \end{aligned} \tag{6.9}$$

where h'' is the second order derivative of h. As h is convex, $h'' \geq 0$ and the previous quantity is positive.

Similarly, if $\phi(\omega; q, p, u) = f_h\left(\omega p + (1-\omega)u, q\right)$, ϕ is convex in ω as:

$$\begin{aligned} &\frac{\partial^2}{\partial \omega^2} f_h\left(\omega p + (1-\omega)u, q\right) \\ &= \int_{z \in \mathbb{R}^D} \frac{q^2(z)(p(z)-u(z))^2}{(\omega p(z) + (1-\omega)u(z))^3} h''\left(\frac{q(z)}{\omega p(z) + (1-\omega)u(z)}\right) dz \end{aligned} \tag{6.10}$$

is a positive quantity.

6.3 Continuous Distributions

We now assume that q, p and u are continuous distributions and let z be an observation vector. In this chapter, we focus on the case of the KL divergence although we could also apply our framework to the Bhattacharyya similarity.

6.3.1 Kullback Leibler (KL)

By definition, we have:

$$\phi_{KL}(\omega; q, p, u) = \int_{z \in \mathbb{R}^D} q(z) \log\left(\frac{q(z)}{\omega p(z) + (1-\omega)u(z)}\right) dz. \tag{6.11}$$

Minimizing the equation 6.11 is equivalent to maximizing:

$$E = \int_{z \in \mathbb{R}^D} q(z) \log(\omega p(z) + (1-\omega)u(z)) dz. \tag{6.12}$$

We will focus on the case where q, p and u are mixture models: $q = \sum_i \pi_i q_i$, $p = \sum_j \pi_{p_j} p_j$ and $u = \sum_j \pi_{u_j} u_j$ where π_i, π_{p_j} and π_{u_j} are respectively the mixture weight for Gaussian q_i, Gaussian p_j and Gaussian u_j. Using previous lower-bound approximation proposed in chapter 5, we have the following lower bound:

$$E \geq \sum_i \pi_i \log\left(\omega\Big(\sum_j \pi_{p_j} \exp(-H_{q_i,p_j})\Big) + (1-\omega)\Big(\sum_j \pi_{u_j} \exp(-H_{q_i,u_j})\Big)\right). \tag{6.13}$$

where H_{q_i,p_j} and H_{q_i,u_j} are defined as follows (cross-entropies):

$$H_{q_i,p_j} = -E_{q_i}[\log p_j] \ , \tag{6.14}$$

$$H_{q_i,u_j} = -E_{q_i}[\log u_j]. \tag{6.15}$$

In the case where q, p and u are Gaussian mixture models (GMMs), there is a closed-form formula for H_{q_i,p_j} and H_{q_i,u_j}. Equation 6.13 can thus be optimized iteratively using the expectation maximization algorithm. At iteration $(k+1)$, we have:

E-step:

$$\gamma_{i,j}^{(k+1)} = \frac{\omega^{(k)}\pi_{p_j}\exp(-H_{q_i,p_j})}{\omega^{(k)}\pi_{p_j}\exp(-H_{q_i,p_j}) + (1-\omega^{(k)})\pi_{u_j}\exp(-H_{q_i,u_j})} \tag{6.16}$$

M-step:

$$\omega^{(k+1)} = \sum_{i,j}\pi_i\gamma_{i,j}^{(k+1)} \tag{6.17}$$

Note that in this case, the cost of the contextual KL is largely dominated by the cost of computing H_{q_i,p_j} and H_{q_i,u_j}, and not by the optimization process itself. As those values also have to be computed in the case of the KL, the added cost of the contextual KL compared to KL can be neglected. As is the case in the previous chapter 5, we can speed-up the computation by using the one-to-one correspondence discussed in chapter 4 and this leads to $\gamma_{i,j}^{(k+1)} = 0$ if $i \neq j$.

6.4 Experiments

6.4.1 Experimental Setup

We make use of the same experimental setup as in section 5.4.1: including feature extraction, PCA reduction, per-image GMMs through MAP adaptation of the universal GMM and classification using SLR classifier.

We choose the method of chapter 5 with the lower-bound MOM KL as the baseline system. We compare the baseline with our symmetric contextual measure based on the KL (later referred to as CTXT KL).

For the cross-entropy computations, we make use of the one-to-one fast scoring described in chapter 4 which efficiently reduces the computational cost. For the early stopping of EM algorithm, we vary the number of iteration from 1 to 5 and report the best accuracy. Each image is represented by the vector of similarities to the set of training images. As opposed to the previous chapter, in general these weights (similarities) do not sum to 1. We apply the SLR classifier directly to the vectors of mixture weights for both systems.

6.4.2 PASCAL VOC2007

We report the mean Average Precision on VOC2007 dataset. From the Figure 6.1, we can see that the CTXT KL approach outperforms the MOM KL for various numbers of Gaussians. We obtain a 57.1% AP which is comparable to the performance reported by the leading participants during the challenge (59.4% [Everingham *et al.* 2007]).

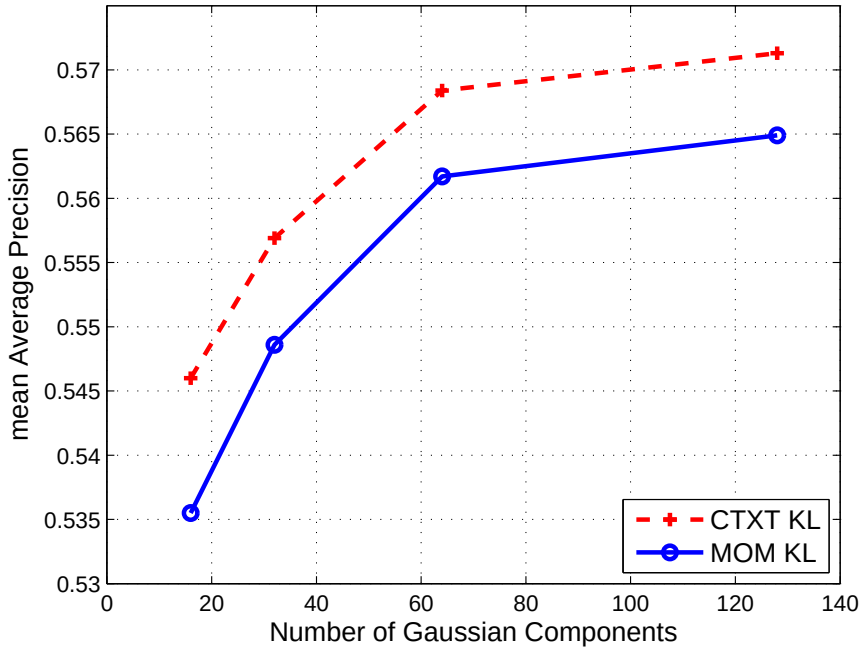


Figure 6.1: The mean AP on the PASCAL VOC2007 of the CTXT KL based on the ORH+COL features for a varying number of Gaussian components.

6.4.3 Scene-15

With the same experimental setup as in 5.4.3, we report the average classification rate over 10 individual runs to evaluate the CTXT KL on Scene-15. Figure 6.2 shows that the CTXT KL obtains more improvement of performance than the MOM KL baseline with a small number of Gaussians. However, both approaches reach the same accuracy (81.2%) when the number of Gaussians is 128.

6.4.4 Caltech-256

We report the average classification rate over 5 individual runs on the Caltech-256 database. Figure 6.3 shows that the CTXT KL approach obtains a 39.0% accuracy (0.4% standard deviation) which gives a very slight improvement over the MOM KL approach (38.4% with 0.4% standard deviation).

6.5 Conclusion

In this chapter, we presented a novel family of contextual measures of similarity between distributions. We explained that in our framework any measure of similarity or dissimilarity had its contextual counterpart. We showed that for two important families of divergences (Bregman and Csiszár) the contextual similarity computation is a convex optimization problem. We focused on the case of continuous distribution

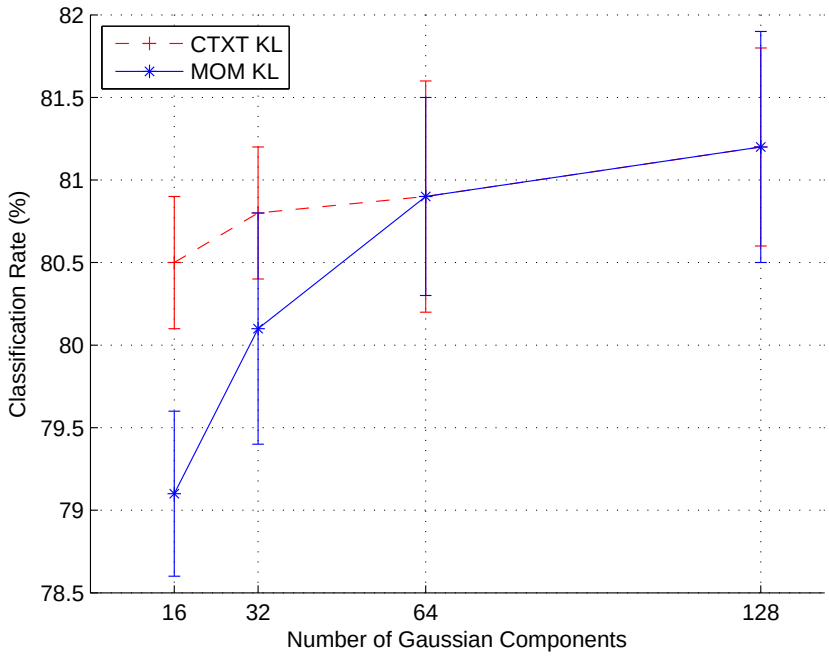


Figure 6.2: Classification rate (%) on Scene-15 database by varying the number of Gaussian components.

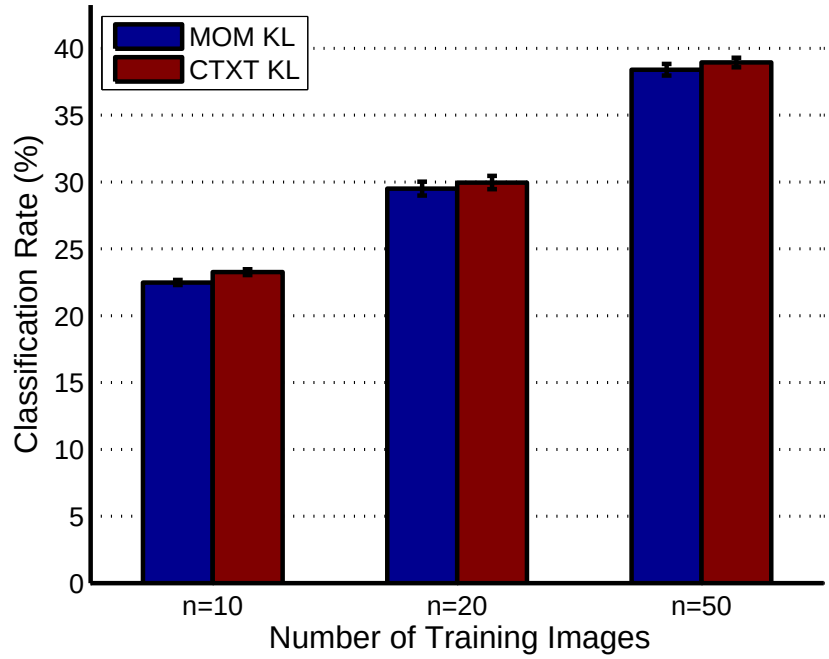


Figure 6.3: Classification rate (%) on Caltech-256 database with 128 Gaussian components by varying the number of training images.

and explained how to compute in practice the similarity for several well-known measures.

Note that this contextual similarity is not significantly better than the one introduced in the previous chapter 5 but constitutes an interesting alternative. Especially the implicit dependence on the set of reference images makes this similarity much more effective for retrieval.

In the future, we intend to focus on the application of this framework to clustering. Indeed, clustering consists in grouping "similar" images where the notion of similarity depends on the other images contained in the dataset. For instance, while it might make sense to group images of different breeds of cats in a general dataset of animal images, it might not in a dataset of cat images. Hence, we believe that clustering is a problem that could benefit greatly from the proposed family of measures.

Chapter 7

Explicit Data Embedding for Large-Scale Learning

Contents

7.1 Introduction

Within the computer vision community, annotated data for training and evaluating image categorization algorithms has long been viewed as a scarce resource. This is because the traditional approach to building such datasets is a time consuming process. As a consequence, the datasets which have been (and are still currently) employed to benchmark image categorization algorithms are fairly modest in size. For instance, Scene-15 [Lazebnik *et al.* 2006], PASCAL VOC2007 [Everingham *et al.* 2007] and CalTech-256 [Griffin *et al.* 2007], three of the most popular benchmarks, contain 4,485, 9,963 and 29,780 images respectively.

However, this perception is drastically changing as large quantities of quality labeled data are becoming available. The ImageNet dataset (www.image-net.org) is certainly the best example of this trend [Deng *et al.* 2009]. As a consequence, the problem is shifting from "how to learn a category from a single

image?" [Fei-Fei *et al.* 2006] to "how to handle these large quantities of data?" [Wang *et al.* 2009, Maji & Berg 2009, Li *et al.* 2009].

As explained in chapter 2, one of the leading approaches to image categorization has been to describe images with bag-of-visual-words (BOV) histograms and to classify them using Support Vector Machines (SVM) classifiers (see [Csurka *et al.* 2004, Zhang *et al.* 2007, Vedaldi *et al.* 2009] for a few examples). While non-linear SVMs are perfectly suited to the small datasets which are typically employed, they do not scale well with the number N of training samples: between $O(N^2)$ and $O(N^3)$. This is to be contrasted with linear SVMs which can be learned in $O(N)$ using, for instance, a cutting-plane algorithm [Joachims 2006] or Stochastic Gradient Descent (SGD) [Shalev-Shwartz *et al.* 2007]. It is even shown in [Shalev-Shwartz & Srebro 2008] that, to achieve a target loss, the runtime of the SGD solver PEGASOS [Shalev-Shwartz *et al.* 2007] *decreases* with the training set size (see also section 2 of [Shalev-Shwartz & Srebro 2008] for additional background on the cost of training non-linear and linear SVMs). This is a very compelling argument for using linear SVMs. Yet, it has been repeatedly reported that non-linear SVMs outperform linear SVMs for image categorization (see *e.g.* [Maji *et al.* 2008, Wang *et al.* 2009, Maji & Berg 2009, Vedaldi *et al.* 2009]).

However, *non-linear kernel classifiers are linear-classifiers*. Indeed, if K is a positive semi-definite (psd) kernel, there exists an *implicit* mapping φ in a new (possibly infinite-dimensional) space such that $K(x,z) = \varphi(x)'\varphi(z)$. Let $X = \{x_i, i = 1...N\}$ denote the set of training samples. The kernel classifier $f(z) = \sum_{i=1}^{N} a_i K(z, x_i) + b$ can be re-written as $f(z) = \left(\sum_{i=1}^{N} \alpha_i \varphi(x_i)\right)' \varphi(z) + b$ which is a linear classifier in the new space. Since we know that there exist linear classifiers which perform well in this new space, it is extremely tempting to perform an *explicit* (possibly approximate) mapping of the data and learn linear classifiers *directly* in this new space.

In previous chapters, we introduced novel image representations and similarity measures between such representations. In this chapter, our goal is somewhat different, yet related. Given an image representation (the BOV) and a kernel, the question is how to transform the representation such that the dot product in the new space is equivalent to the kernel similarity in the original space. Several approaches have been proposed along this line. Kernel Principal Component Analysis (kPCA) [Schölkopf *et al.* 1998], which was successfully applied in [Williams & Seeger 2001], can be employed with any kernel. [Rahimi & Recht 2007] proposed an embedding based on random projections (RPs) for shift-invariant kernels. Recently, [Maji & Berg 2009] proposed an embedding for the intersection kernel. We note that the accuracy of an embedding approach depends directly on the accuracy of the kernel which is approximated: good kernels lead to good embeddings as verified in our experiments.

In this chapter we explore several approaches to performing explicit embedding on various kernels. More precisely, our contributions are the following ones:

- We show that square-rooting BOV vectors, which corresponds to an exact mapping in the case of the Bhattacharyya kernel, already leads to large improvements.

- To go beyond this simple strategy, we propose addkPCA: a novel kPCA-based embedding for additive kernels. We demonstrate that it leads to additional improvements at a very affordable cost.
- We go one step further and experiment with embedding for non-additive kernels. We show that the fairly small gain in classification accuracy over additive kernels is obtained at the expense of a very significant increase of the computational cost.
- Using ImageNet data, we show experimentally that we continuously improve the classification accuracy using up to 10,000 training images per class. This clearly justifies the use of very large training sets.

The remainder of this chapter is organized as follows. In section 7.2, we discuss the choice of kernels for BOV histograms (*i.e.* multinomials). In section 7.3, we briefly review *kPCA*. In section 7.4, we show how to adapt this framework to additive kernels. In section 7.5, we review embedding with RPs for shift-invariant kernels. In section 7.6.2, we present experimental results on datasets: PASCAL VOC2007 and ImageNet. We also give a very brief introduction to the SGD solver we use in our experiments in Appendix A.1. The remainder of the relevant literature will be reviewed throughout the chapter [1].

7.2 Kernels for BOV Histograms

7.2.1 Additive Kernels

Additive kernels can be written as $K(x,z) = \sum_{d=1}^{D} k(x(d), z(d))$ where $x(d)$ denotes the d-th dimension of vector x. They have gained considerable popularity since [Maji *et al.* 2008] showed that the classification of a sample by an additive kernel SVM can be done approximately at a cost which is independent of the number of support vectors. This is because the classification function $f(z) = \sum_{i=1}^{N} a_i K(z, x_i) + b$ can be rewritten as $f(z) = \sum_{d=1}^{D} f_d(z(d)) + b$ with $f_d(z(d)) = \sum_{i=1}^{N} a_i k(z(d), x_i(d))$ and each of the 1-D f_d functions can be approximated by piecewise-constant or -linear functions. The most popular additive kernels are the Bhattacharyya kernel, the χ^2 kernel and the intersection kernel (*c.f.* equations 2.10, 2.11 and 2.12).

While the Bhattacharyya kernel has been successfully applied to continuous distributions [Jebara & Kondor 2003, Farquhar *et al.* 2005] in our experience, it remains under-exploited in the case of BOV histograms. K_{bha} can be directly related to K_{chi2}. Indeed, we have the following bounds (*c.f.* Appendix A.2):

$$K_{chi2}(x,z) \leq K_{bha}(x,z) \leq \frac{1}{2}\left(1 + K_{chi2}(x,z)\right) \tag{7.1}$$

and if $x \approx z$, $K_{bha}(x,z) \approx \frac{1}{2}(1+K_{chi2}(x,z))$. Now, going back to our data embedding problem, we see that in the case of K_{bha} the mapping is trivial: $\varphi(z) = \sqrt{z}$ (where

[1] While in the previous chapters we used SLR for linear classification, we switched to linear SVMs in this chapter. This is because our SLR code did not scale to the large amounts of images employed in section 7.6.2 (up to 140k) as opposed to the SGD code of [Bottou] which dealt easily with such large quantities of data.

the square-root of a vector should be understood as a term-by-term operation). Hence, in the case of K_{bha} the mapping can be done exactly at (almost) zero cost, simply by square-rooting the BOV vectors.

7.2.2 Exponential Kernels

All the previous additive kernels can be exponentiated and their exponential versions are also guaranteed to be psd kernels (*c.f.* equations 2.13, 2.14 and 2.15). Recently, [Vedaldi *et al.* 2009] reported on an object segmentation task that K^{exp}_{chi2} outperformed K_{chi2}. In section 7.6.2, we systematically compare the accuracy of the additive kernels with their exponential counterparts and confirm the superiority of exponential kernels.

7.3 Embedding with Kernel PCA

We just provide a brief introduction to embedding with kPCA. For more background on kPCA, please refer to [Schölkopf *et al.* 1998]. Let $\{x_i, i = 1 \dots M\}$ be a set of samples in $\mathbb{R}^D$ provided to learn the embedding. Given a kernel $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, we seek a mapping $\varphi : \mathbb{R}^D \to \mathbb{R}^E$ such that $K(x_i, x_j) \approx \varphi(x_i)'\varphi(x_j)$, $\forall(i, j)$. A possible criterion to find an optimal φ is to minimize:

$$\sum_{i=1}^{M} \sum_{j=1}^{M} \left(K(x_i, x_j) - \varphi(x_i)'\varphi(x_j)\right)^2 \tag{7.2}$$

$$\text{under} \qquad \sum_{i=1}^{M} \varphi_e(x_i)\varphi_f(x_i) = 0 \text{ for } e \neq f \tag{7.3}$$

where $\varphi_e(x_i)$ denotes the e-th dimension of $\varphi(x_i)$. The constraints (7.3) translate the fact that we want the dimensions of φ to be uncorrelated. Let $\mathbf{K}$ be the $M \times M$ kernel matrix and $\phi_e = [\varphi_e(x_1), \dots, \varphi_e(x_M)]$. We write $\phi_e = \sigma_e \psi_e$ where σ_e is the L2 norm of ϕ_e (and consequently ψ_e is the L2-normalized version of ϕ_e). Taking the derivatives of the equation 7.2 with respect to the $\varphi_e(x_i)$'s leads to:

$$\mathbf{K}\psi_e = \sigma_e^2 \psi_e \quad \text{for} \quad e = 1 \dots E. \tag{7.4}$$

This is a standard eigenvalue problem. The solution to minimizing the equation 7.2 is to take the E eigenvectors ψ_e corresponding to the E largest eigenvalues σ_e^2 and to set $\varphi_e(x_i) = \sigma_e \psi_e(i)$.

We define $K(z, .) = [K(z, x_1), \dots, K(z, x_M)]$. To extend the mapping to a new sample z, we use the Nyström approximation [Williams & Seeger 2001]:

$$\varphi_e(z) = \frac{K(z, .)'\psi_e}{\sigma_e}. \tag{7.5}$$

The cost of computing the kernel matrix $\mathbf{K}$ is in $O(M^2 D)$. The cost of the eigenvalue decomposition – typically based on a Cholesky decomposition – is in $O(M^3)$. The cost of embedding a new sample is in $O(M(D + E))$. In all our kPCA experiments, we set $M = E$.

7.4 Embedding for Additive Kernels: addkPCA

While kPCA can be applied as is to additive kernels, it does not exploit their additive structure. We now propose addkPCA: an efficient kPCA embedding for additive kernels.

7.4.1 Learning the Embedding

As noted in [Maji & Berg 2009], a possible way to find an embedding φ for additive kernels is to find separate embeddings φ_d for each dimension d and then to concatenate them. Therefore, we propose to apply the kPCA framework separately in each dimension. Given an additive kernel $K(x,z) = \sum_{d=1}^{D} k(x(d), z(d))$ and a set of samples $\{x_i, i = 1 \dots M\}$, the embedding is learned as follows:

1) For each dimension d:

   a) Compute the $M \times M$ kernel matrix $\mathbf{k_d}$ with entries $k(x_i(d), x_j(d))$

   b) Compute the F largest eigenvalues $\sigma_{d,1}^2, \dots, \sigma_{d,F}^2$ of $\mathbf{k_d}$ (*e.g.* $F = 10$) and their associated eigenvectors $\psi_{d,1}, \dots, \psi_{d,F}$.

2) Sort the $D \times F$ eigenvalues $\sigma_{d,e}^2$ for $d = 1 \dots D$ and $e = 1 \dots F$ and keep the E largest ones with their associated eigenvectors.

Although the function k is the same for all dimensions, we learn different embedding functions $\varphi_{d,e}$ in each dimension because the distribution of samples is different in each dimension. Also, a variable number of functions $\varphi_{d,e}$ is selected automatically by the algorithm in each dimension d (*c.f.* step 2)). This enables to put more emphasis on those dimensions which contain more energy.

The cost of computing the D kernel matrices is in $O(M^2 D)$ and the cost of the D eigenvector decompositions is in $O(M^3 D)$. The crucial difference between kPCA and addkPCA is that, in the first case, one approximates a kernel $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ while, in the second case, one approximates D times a kernel $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Since K operates in a much higher dimensional space than k, the number M of vectors necessary to learn a good approximation of K is typically much larger than for k. In our experiments, we found that $M = 128$ was sufficient to learn addkPCA as opposed to several thousands for kPCA. Learning addkPCA on BOV histograms of dimensionality $D = 4,000$ with $M = 128$ takes less than 30s on a single CPU of a 3GHz Xeon machine.

We show in Figure 7.1 the first eigenfunctions in the case of K_{chi2}. An important question is: how many dimensions E will typically be needed to obtain a good embedding? We show that K_{chi2} can already be correctly approximated using a single eigenfunction. We will show in section 7.6.2 that with $E = 2D$ (*i.e.* by doubling the dimensionality of our representations) we can do as well as additive kernel SVMs.

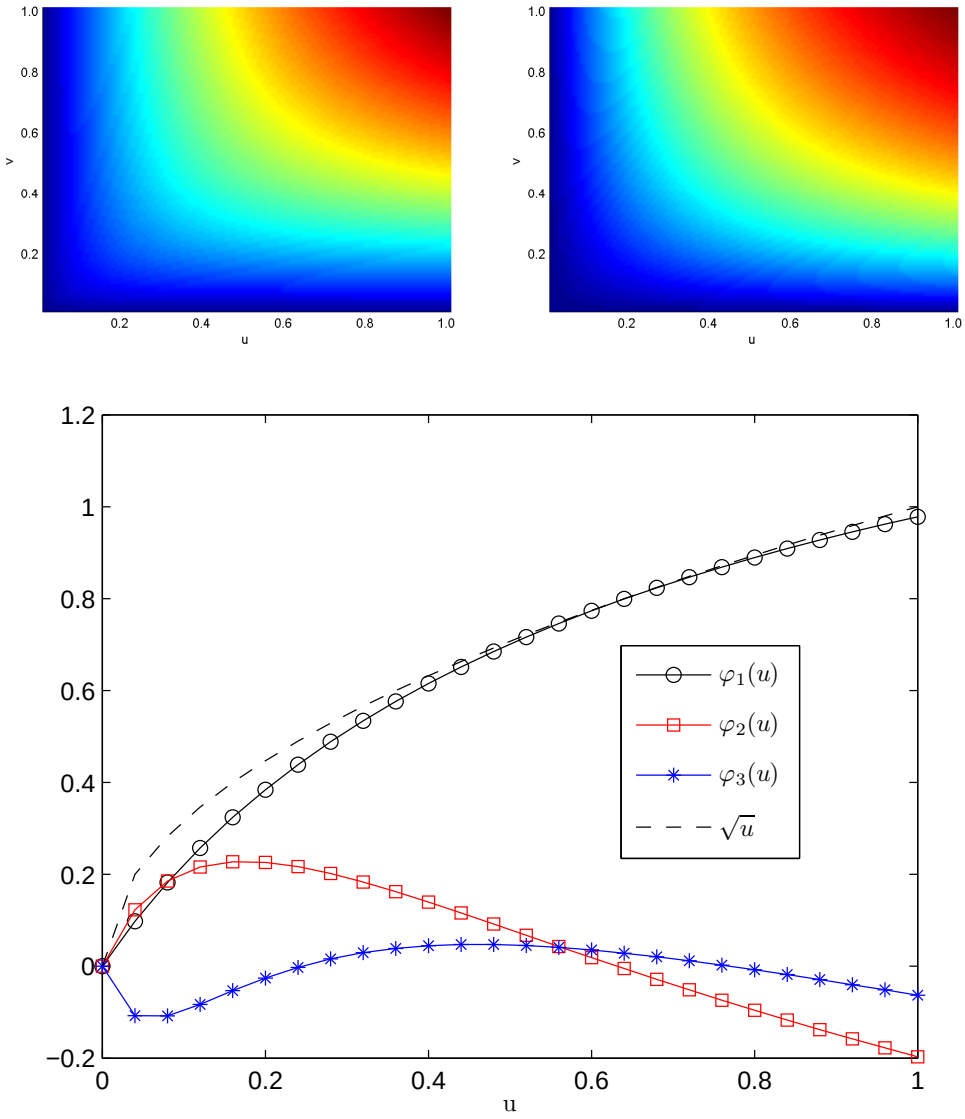


Figure 7.1: Top-left: the function $k(u, v) = 2\frac{uv}{u+v}$ which corresponds to K_{chi2}. Top-right: its approximation $\varphi_1(u)'\varphi_1(v)$. Using a single eigenfunction already leads to a very good approximation of k. Bottom: the first 3 eigenfunctions of k. For this example, we learned the embedding using samples regularly spaced on the [0,1] interval. Notice that $\varphi_1(u) \approx \sqrt{u}$.

7.4.2 Embedding a new sample

We can embed a sample using the Nyström approximation:

$$\varphi_{d,e}(z(d)) = \frac{k(z(d),.)'\psi_{e,d}}{\sigma_{e,d}} \tag{7.6}$$

with $k(z(d),.) = [k(z(d),x_1(d)),\ldots,k(z(d),x_M(d))]$. In this case, the embedding cost is in $O(M(D+E))$.

However, we can make the embedding cost independent of M using the same trick as [Maji *et al.* 2008]. We split each of the D dimensions into a set of B bins (by regularly sampling the interval between the smallest and largest values). Let $c_{d,b}$ be the center of the b-th bin in dimension d. We pre-compute the values $\varphi_{d,e}(c_{d,b})$ for the E dimensions of the embedding using Nyström. Given a new sample z, we compute for each dimension the bin index ($O(D)$) and then use a piecewise-constant or -linear approximation of $\varphi_{d,e}$ ($O(E)$).

7.4.3 Classification of a sample

We assume that we have learned a set of C linear classifiers on the embedded samples: $f_c(z) = w_c'\varphi(z) + b_c$. We can compute $f_c(\varphi(z))$ in two different ways:

1. The simplest way is to perform the explicit embedding $\varphi(z)$ ($O(D+E)$) and then to do C dot-products ($O(CE)$).

2. Using – again – the trick of [Maji *et al.* 2008], we can rewrite the classifiers f_c as non-linear classifiers in the original space. $f_c(z) = \sum_{d=1}^{D} f_{c,d}(z(d)) + b$ where each 1D function $f_{c,d}$ can be approximated by binning the d-th dimension and then by using a piecewise-constant or -linear approximation. The cost of computing the bin indices in each dimension is in $O(D)$ and the cost of the C classifications is in $O(CD)$.

As already mentioned, to obtain top accuracy with our approach, we typically set $E = 2D$ which may lead to think that the second option is cheaper. Actually, this is incorrect as the $O()$ notation hides constants. For the first option, the computation of $f_c(z)$ relies on dot-products which are very efficient operations while for the second option it relies on look-up table accesses which are much slower [2]. In practice, using a C++ implementation and setting $E = 2D$, we found that a standard dot-product was approximately *5 times faster* than look-up tables. Using the SSE2 [3] implementation of the dot-product of [Bottou], the speed-up was a *factor of 15*. Note that SSE2 instructions cannot speed-up look-up table accesses.

[2] On modern processors, the main bottleneck is often not the execution of the instructions but the access to the data. To reduce the access time, processors perform *prefetching*, *i.e.* when accessing the data at a given position in RAM, they also retrieve the data at subsequent positions. Prefetching results in a very significant speed-up provided that data is accessed *sequentially*. This is the case when performing dot-products and this is generally not the case when accessing look-up tables.

[3] Modern processors, support *single instruction with multiple data* (SIMD) commands. SSE2 instructions perform multiple simple operations, such as additions or multiplications, in parallel.

To learn the linear SVM for class c we use an SGD solver (*c.f.* Appendix A.1). At each iteration, the value $f_c(x_i)$ is evaluated with a different training sample x_i (*c.f.* the term δ_i in equation A.4). To train our linear classifiers, we also prefer the direct approach to computing $f_c(x_i)$ (embedding + dot-product) over the approach based on look-up tables. This is much more efficient when the same sample is used multiples times by the SGD solver (5 times on average in our experiments).

7.4.4 Related Work

Closest to our approach is [Maji & Berg 2009] which proposes an explicit embedding for the intersection kernel. Each dimension of the BOV vector is embedded into a B-dimensional space. Per dimension, only 2 values are non-zero. This leads to a BD-dimensional embedding with $2D$ non-zero values. As is the case of [Maji & Berg 2009], addkPCA is as accurate as the additive kernel SVM it approximates. The main advantages of the proposed approach with respect to [Maji & Berg 2009] are the following ones:

- Our approach preserves data sparsity. Let z be a D-dimensional vector and let nz be the proportion of non-zero values in z. According to equation 7.6, if $z(d) = 0$, then $k(z(d), .)$ is the null vector (at least in the case of K_{int}, K_{chi2} or K_{bha}) and $\varphi_{d,e}(z(d)) = 0$. Hence, the proportion of non-zero values in $\varphi(z)$ is approximately nz (this is only approximate because we keep a variable number of eigenvectors per dimension). This is to be contrasted with [Maji & Berg 2009] where $2D$ values are non-zero (independently of nz). This can be a limitation for very sparse data. For instance, in the case of object segmentation with a sliding window classifier, the classifier typically has to consider many small windows described by very sparse histograms (each window containing a few patches).

- The training cost of [Maji & Berg 2009] is in $O(NBD)$ with B ranging from 30 to 100. Our training cost is in $O(ED)$ with $E = 2D$ typically.

- [Maji & Berg 2009] reports that their work makes "training and testing an additive classifier only a small (constant factor) slower than training a linear classifier". The classification as well as the SGD training in [Maji & Berg 2009] are based on look-up table accesses while our approach is based on dot-products. We explained in the previous subsection that the second type of operation is much more efficient: up to 15 times faster (hardly a "small" constant) with a good implementation of a dot-product.

[Wang *et al.* 2009], which proposes an algorithm to learn intersection kernel SVMs, is closely related to [Maji & Berg 2009] and to addkPCA. The advantages of our approach with respect to [Wang *et al.* 2009] are the same as the advantages with respect to [Maji & Berg 2009]: we preserve sparsity and classification as well as training are more efficient.

7.5 Embedding for Shift-Invariant Kernels

We provide a brief introduction to embedding with random projections (RPs) for shift-invariant kernels. For more details, please refer to [Rahimi & Recht 2007].

Let $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be a kernel. It is shift-invariant if it can be written under the form $K(x, z) = K(x - z)$. For instance K_{int} is shift-invariant since $K_{int}(x, z) = 1 - \frac{1}{2}|x - z|$ where $|.|$ is the L1 norm. Consequently K_{int}^{exp} is also shift-invariant. Similarly, $K_{bha}^{exp}(x, z)$ can be rewritten as an rbf kernel: $K_{bha}^{exp}(x, z) = K_{rbf}(\sqrt{x} - \sqrt{z})$. Therefore, K_{bha}^{exp} is shift-invariant in the space of the square-rooted BOV vectors. In our experiments, we will focus on K_{bha}^{exp}.

$K(x-z)$ is psd if and only if it is the Fourier transform of a non-negative measure p (Bochner's theorem). If K is properly scaled, p is a probability distribution. For instance if K is the rbf kernel, then p is the Gaussian distribution. We have:

$$K(x - z) = \int_{\omega \in \mathbb{R}^D} p(\omega) e^{j\omega'(x-z)} d\omega \tag{7.7}$$

As K and p are real, we can replace $e^{j\omega'(x-z)}$ by $\cos(\omega'(x - z))$. Introducing the 2D vector $\zeta_\omega(x) = [\cos(\omega' x) \sin(\omega' x)]$, we have:

$$K(x - z) = E_\omega \left[\zeta_\omega(x)' \zeta_\omega(z)\right] \tag{7.8}$$

In practice, to compute an embedding $\varphi : \mathbb{R}^D \to \mathbb{R}^E$, one draws a set of $E/2$ iid Gaussian vectors $\omega_1, \ldots, \omega_{E/2} \in \mathbb{R}^D$ and the mapping is $\varphi(x) = \frac{1}{\sqrt{D}}[\cos(\omega_1' x), \ldots, \cos(\omega_{E/2}' x), \sin(\omega_1' x), \ldots, \sin(\omega_{E/2}' x)]$. As $E \to \infty$, $\varphi(x)'\varphi(z) \to K(x - z)$.

We highlight that this approach does not require any learning. The cost of embedding a vector is in $O(DE)$.

7.6 Experiments

We follow the same experimental setup as in section 4.3.1 but with SIFT features only. We did not make use of color features in our experiments. This is because some of our large scale experiments were already time-consuming and running experiments with color features would have doubled the runtime. We train visual codebooks of 4,000 visual words. An image is described by the L1-normalized histogram of visual-word occurrences ($D = 4,000$). To learn linear SVMs, we use the SGD code available at [Bottou] (*c.f.* Appendix A.1). To learn non-linear SVMs, we use libsvm [Chang & Lin 2001].

7.6.1 Small-Scale Experiments

For the small-scale experiments, we focus on accuracy, not on the training time. Indeed, training non-linear SVMs on small datasets is not an issue. Our goal is to verify to which extent the 3 approaches to explicit data embedding can approximate the non-linear kernel classifiers.

Linear SVM [Bottou]					
44.1					
Non-Linear SVM [Chang & Lin 2001]					
Additive kernels			Exponential kernels		
K_{bha}	K_{chi2}	K_{int}	K_{bha}^{exp}	K_{chi2}^{exp}	K_{int}^{exp}
48.9	50.1	50.5	52.7	52.8	52.0

Table 7.1: PASCAL VOC2007 baseline (in % AP).

E =	4k	8k	12k
PASCAL VOC2007			
AP (in %)	49.9 (0.2)	50.6 (0.1)	50.7 (0.1)

Table 7.2: Accuracy (and standard deviation) of the proposed addkPCA applied to K_{chi2}. E is the dimensionality of the embedded vectors.

Datasets: We experimented with PASCAL VOC2007 [Everingham *et al.* 2007]. Classification accuracy is measured using Average Precision (AP). We report the average over the 20 classes. To tune the SVM regularization parameters, we trained them on the train set and used the val set for validation.

Baseline. We report baseline linear and non-linear SVM results in Table 7.1 for VOC2007. As expected non-linear SVMs perform significantly better than linear ones, and exponential kernels somewhat better than their additive counterparts. All additive kernels perform similarly and the same can be said about exponential kernels.

Embedding additive kernels. We start with linear classifiers on square-rooted BOV vectors (exact embedding for K_{bha}). We obtain a 49.0% AP on VOC2007. Compared to the linear baseline, this corresponds to absolute increases of 4.9%.

We now turn to addkPCA. We used a subset of $M = 128$ samples to learn the embedding. We repeated the experiments 10 times with different subsets to learn the embedding and we report the average for K_{chi2} in Table 7.2. With $E = D = 4k$, we get results which are slightly better than the simple square-rooting approach. With $E = 2D = 8k$, we get results as good as those obtained with the best additive kernels.

Embedding exponential kernels. We focus on K_{bha}^{exp} since (i) it gives the best (or very close to the best) results and (ii) it is based on the dot-product between square-rooted vectors which is very fast to compute. We report experiments with kPCA and RPs.

For kPCA, we repeated the experiments 10 times using different subsets of the training data to learn the embedding. We report the accuracy in Table 7.3. On both datasets, kPCA does its best job when all samples are used to learn the embedding. This not surprising given the high-dimensionality of the data and the small training set sizes.

E =	256	512	1024	2048	Full
PASCAL VOC2007					
AP (in %)	45.6 (0.3)	47.9 (0.4)	49.8 (0.2)	51.2 (0.2)	52.4 (0.0)

Table 7.3: kPCA accuracy (and standard deviation) with K^{exp}_{bha}. E is the number of training samples used to learn the embedding and therefore the dimensionality of the embedded vectors. "Full" corresponds to using all training samples to learn the embedding (5,011 for VOC2007).

E =	4k	8k	12k	16k	20k
PASCAL VOC2007					
AP (in %)	46.7 (0.5)	49.0 (0.3)	50.0 (0.2)	50.7 (0.2)	51.0 (0.2)

Table 7.4: Accuracy (and standard deviation) of the approach of [Rahimi & Recht 2007] applied to K^{exp}_{bha}. E is the dimensionality of the embedded vectors.

For RPs, we also repeated each experiment 10 times with different random matrices. Results are reported in Table 7.4 for various values E of the embedding dimensionality. On the challenging VOC2007 dataset, even with $E = 5D = 20k$, we do not reach the accuracy of the baseline K^{exp}_{bha}.

7.6.2 Large-Scale Experiments

We now evaluate how the various embedding techniques scale to large training sets. To train classifiers, we used a subset of ImageNet [Deng *et al.* 2009]. Our goal was to create a training set with the same 20 classes as PASCAL VOC2007. At the time when we downloaded the data, only 18 classes had at least one synset in ImageNet: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, sheep, sofa, train and tvmonitor. We downloaded images from these 18 categories as well as their sub-categories (*i.e.* children synsets). In our experiments, we trained classifiers with up to 10k images per class. Half of the classes have less than 10k images: bicycle (6,653 images), bus (5,224), cow (1,572), dining table (4,061), motorbike (3,779), sheep (9,702), sofa (7,474), train (9,982) and tvmonitor (1,202).

To avoid the bias which typically occurs when training classifiers and evaluating their accuracy on the same dataset, we used as test data the VOC2007 test set. Training and testing on datasets collected separately reduces over-fitting effects and offers a much more realistic (and interesting) challenge. As is the case of VOC2007, we use AP (averaged over the 18 classes) to measure classification accuracy.

Each experiment was repeated 5 times on 5 different training subsets (later referred to as folds) and we report the average value as well as the standard deviation.

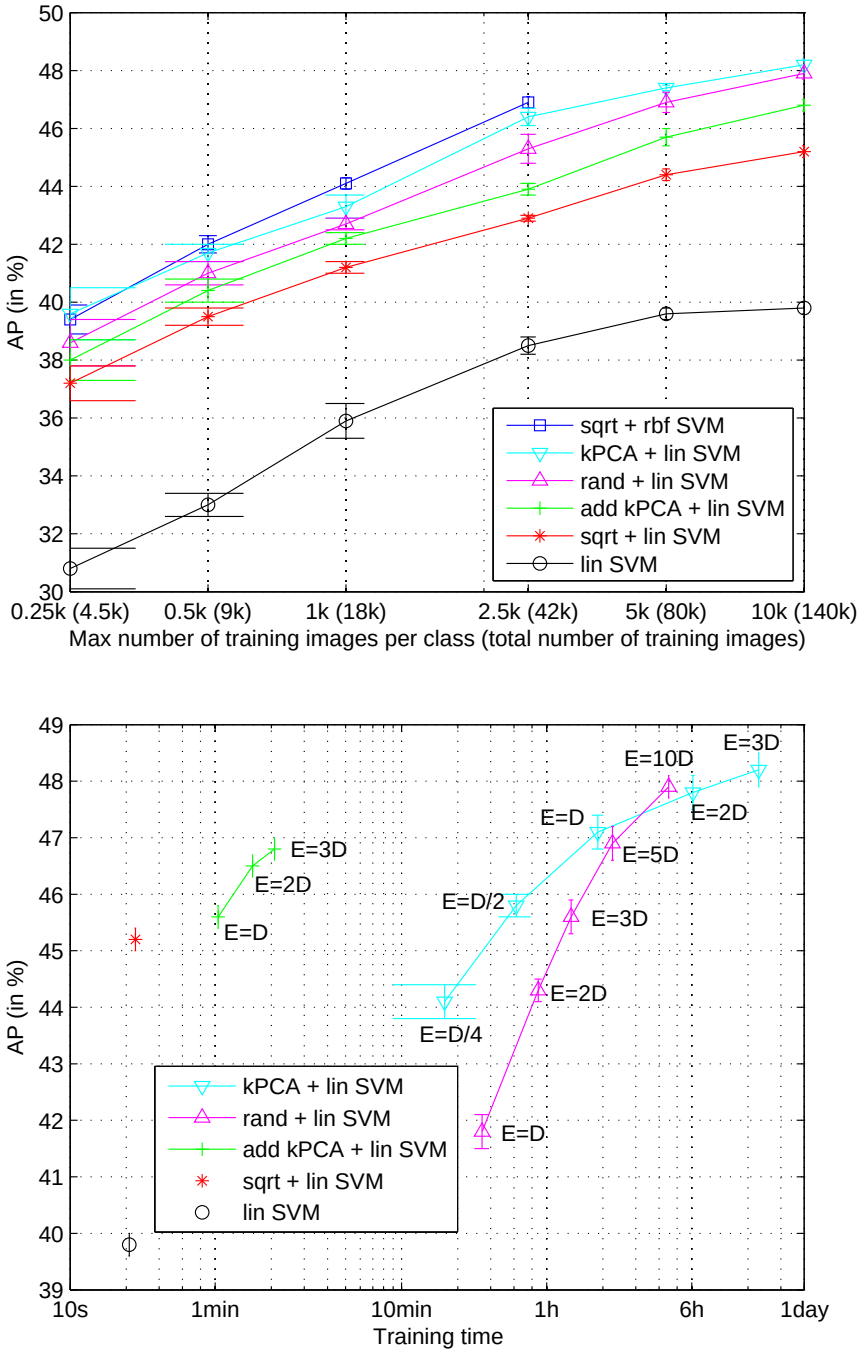


Figure 7.2: Top: classification accuracy as a function of the number of training images. Bottom: classification accuracy as a function of the training time (*i.e.* for various sizes E of the embedding space) using a training set of 140k images. "lin SVM" = linear SVM baseline. "sqrt + lin SVM" = linear SVM on square-rooted vectors. "add kPCA + lin SVM" = proposed addkPCA for K_{chi2}. "rand + lin SVM" = random projections to approximate K_{bha}^{exp}. "kPCA + lin SVM" = standard kPCA with K_{bha}^{exp}. "sqrt + rbf SVM" = baseline kernel SVM with K_{bha}^{exp}. For the 3 embedding techniques, we used on the top plot the best system, *i.e.* the one with the highest value E on the bottom plot (and therefore the highest training cost) .

To tune the SVM regularization parameters, we trained the classifiers on 50% of the training data of the first fold and used the second 50% of this fold as validation set. The same parameters were used for the 5 folds.

Does more data help? We first evaluate the influence of the training set size on the classification accuracy. This is an important question that remains largely unexplored. For example, [Wang *et al.* 2009, Maji & Berg 2009, Li *et al.* 2009] report experiments on large training sets but do not quantify the benefit of using the full training set as opposed to only a fraction. Results are reported in Figure 7.2 (top). As a comparison, the smallest training set (250 images per class = 4.5k images in total) is comparable in size to the train+val set of VOC2007. We do not report results with libsvm beyond 42k training images as training the 18 categories on 42k images already takes approx. 38h of CPU time per fold. We observe a continuous improvement of the AP up to 10k images per class (140k images in total). This clearly justifies the use of very large training sets.

On the 140k image training sets, square-rooting BOV vectors already leads to a large improvement: +5.4% AP (39.8% vs 45.2%). Using addkPCA leads to an additional 1.6% increase (46.8%). Finally, the largest increase we obtained with exponential kernels was another 1.4% (48.2%).

Training cost. We report in Figure 7.2 (bottom) the accuracy as a function of the training time on the 140k training sets. The training time includes the combined time of learning the embedding, embedding the training vectors and training the 18 classifiers. All the CPU times correspond to training on a single fold and were measured on a 3GHz Xeon machine using a single CPU. As a comparison, it takes less than 4h to compute SIFT features and extract BOV histograms from 140k images (approx. 100ms per image). While kPCA or RPs can outperform addkPCA, the benefit comes at a high cost: several hours of training as opposed to 2 min. Note that the cost of kPCA and RPs is largely dominated by the embedding, not by the linear SVM training.

Memory requirements should also be taken into account. For instance, with RPs, one has to set $E = 10D$ to obtain a mere 1% AP improvement over the best addkPCA results and the 140k image signatures already take approx. 20GB. We did not experiment with $E = 20D$ for instance, as the training data would not have fitted in the RAM of our 32GB machine. An important conclusion is that RPs for shift-invariant kernels seem better suited to problems where the input vectors are small dimensional as is the case of the datasets employed in [Rahimi & Recht 2007] (127 dimensions max.).

Testing cost. We report the testing time of the systems trained on the 140k images. As a comparison, the extraction of the BOV histograms from the 4,952 test images takes approx. 8 min. The baseline linear SVM and the linear SVM on the square-rooted vectors classify the whole test set in approx. 0.45s. addkPCA with $E = 2D$ takes approx. 2s using explicit embedding (1.4s) + dot products (0.65s). If we were to use look-up tables instead (*c.f.* section 7.4.3) the test time would be approx. 10s. The benefit of dot-products as opposed to look-up tables increases with the number of categories (*i.e.* when the embedding cost becomes negligible with respect to dot-products). kPCA with $E = 3D$ takes approx. 22 min and RPs with $E = 10D$ approx. 10 min. For both kPCA and RPs, the classification time is

fully dominated by the embedding cost.

7.7 Conclusion

In this chapter, we considered the problem of learning image classifiers with large training sets. We explored explicit embedding as an approach to scale non-linear SVMs and experimented with several techniques. We demonstrated that simply square-rooting BOV vectors, which corresponds to an exact embedding for K_{bha}, already leads to large improvements. We then proposed addkPCA: a simple and efficient embedding based on kPCA for additive kernels. We showed that it improved over the square-rooting at a very affordable cost. Finally, we experimented with kPCA and random projections to approximate exponential kernels and showed that it led to additional improvements but at a much higher cost this time.

This study focused on large training sets but did not address the problem of dealing with a large number of categories (*e.g.* thousands) which is a very challenging problem on its own. While we focused on the BOV, extensions to more complex representations would also be interesting such as the GMM-based representations introduced in the previous chapters.

Chapter 8

Conclusion

Contents

8.1 Summary

This thesis addressed the problem of deriving novel image representations and computing similarities between such representations with application to image classification. The major contributions of this thesis are the following ones.

- In chapter 4 we presented a novel approach to compute the similarity between two unordered vector sets. We characterized each bag-of-patch model with a continuous GMM which was adapted from a common universal GMM using MAP. This adaptation provides a more accurate estimation than MLE when the number of features is small and allows a one-to-one correspondence between the Gaussians of image GMMs for fast scoring. We derived efficient approximations for two probabilistic kernels: the Probability Product Kernel and the Kullback-Leibler Kernel. We applied this approach to image classification and showed its efficiency in terms of classification accuracy and cost.

- In chapter 5 a novel image representation was introduced. It consists in approximating an image, modeled for instance by a GMM, as a convex combination of K reference image GMMs and then characterizing the image as a K-dimensional vector of mixture weights. We proposed two approximate optimization algorithms for the computation of the mixture weights: one based on sampling and one based on variational bounds. This approach was applied to image classification and it was shown to outperform the GMM approach introduced in chapter 4.

- In chapter 6 we also introduced a novel family of contextual measures of similarity between distributions. The idea was to measure the similarity between two distributions p and q in the context of a third distribution u. We showed that for two important families of divergences (Bregman and Csiszár) the contextual similarity computation is a convex optimization problem. We focused on the case of continuous distributions, typically GMMs, and applied this similarity to the problem of kernel-based image classification. With respect to the

approach of chapter 5, we did not observe a significant increase of accuracy. However, an advantage of this contextual similarity is that the context does not depend explicitly on a set of reference images.

- Finally, in chapter 7 we explored explicit embedding as an approach to scale non-linear SVMs on BOV representations. We experimented with two state-of-the-art techniques to embedding: kernel PCA (kPCA) and random projection for shift-invariant kernels. We also proposed a simple and efficient embedding based on kPCA for additive kernels, addkPCA. We observed that a simple square-rooting of BOV vectors (which corresponds to an exact embedding in the case of the Bhattacharyya kernel) already led to significant improvements. We also showed that additional improvements could be gained with addkPCA at a very affordable cost. Finally, we observed that, while it is possible to improve over additive kernels, the gain in classification accuracy was obtained at a very high computational cost.

8.2 Future work

We have tried to outline some future work at the end of chapters 4, 6 and 7. We now summarize them here:

- **Adaptation Techniques.** The proposed MAP adaptation provides accurate estimation and improves the performance. We believe that other adaptation techniques could also be employed, such as Maximum Likelihood Linear Regression (MLLR) [Leggetter & Woodland 1995, Gales 1998], Cluster Adaptive Training (CAT) [Gales 2000] or Eigenvoices [Kuhn *et al.* 2000]. They have been shown to yield significantly better results than MAP in the speech recognition literature for smaller adaptation set.

- **Multiple Kernel Learning.** From the experimental results with PPK and KLK, we note that there is no clear winner between these two kernels. It might be interesting to consider principled combinations of kernels using Multiple Kernel Learning (MKL) [Varma & Ray 2007] to avoid the per-dataset selection of kernels.

- **Clustering based on Contextual Similarity.** Clustering consists in grouping "similar" images where the notion of similarity depends on the other images contained in the dataset. For instance, while it might make sense to group images of different breeds of cats in a general dataset of animal images, it might not in a dataset of cat images. Hence, we believe that clustering is a problem that could benefit greatly from the proposed family of contextual measures of similarity.

- **Extension of Explicit Data Embedding.** In this thesis we have shown that GMM-based representation obtain significantly improvements over the BOV. We believe that it would be interesting to extend explicit embedding methods to these more complex GMM-based representations.

- More generally, we believe that the ability to scale on algorithms to a large number of classes is necessarily to make image classification useful for real world applications. Doing so efficiently requires the ability to deal with class hierarchies, a problem we did not address in this thesis.

Appendix A

Appendix

A.1 SGD for linear SVMs

We give a brief overview of the SGD solver available at [Bottou] inspired by [Shalev-Shwartz *et al.* 2007]. Given a training set of labeled samples $\{(x_i, y_i), i = 1, \ldots, N\}$, with $y_i \in \{-1, +1\}$, we consider the problem of minimizing the regularized loss:

$$E(w) = \frac{\lambda}{2}||w||^2 + \frac{1}{N}\sum_{i=1}^{N}\lambda(w'x_i, y_i). \tag{A.1}$$

where λ is the regularization parameter. In the SVM case, $\lambda(w'x_i, y_i) = \max\{0, 1 - y_i w'x_i\}$. We did not include a bias term to simplify the presentation. Assuming that at iteration t the SGD algorithm considers only sample x_i, the loss (A.1) is approximated by:

$$E_i(w) = \frac{\lambda}{2}||w||^2 + \lambda(w'x_i, y_i). \tag{A.2}$$

w is updated as follows:

$$\begin{aligned} w_{t+1} &= w_t - \eta_t \nabla_{w=w_t} E_i(w) && \text{(A.3)} \\ &= (1 - \lambda\eta_t)w_t + \eta_t y_i x_i \delta_i && \text{(A.4)} \end{aligned}$$

with $\delta_i = 1$ if $\lambda(w'x_i, y_i) > 0$ and $\delta_i = 0$ otherwise. The learning rate has the form $\eta_t = \frac{1}{\lambda(t+t_0)}$ where t_0 is set heuristically in [Bottou].

A.2 Relationship between K_{chi2} and K_{bha}

Let x and z be two D-dimensional multinomials. Let us first show that $K_{bha}(x, z) \geq K_{chi2}(x, z)$. Let $u \in [0, 1]$ and $v \in [0, 1]$. We have:

$$uv - \left(\frac{2uv}{u+v}\right)^2 = \frac{uv(u-v)^2}{(u+v)^2} \geq 0. \tag{A.5}$$

Hence:

$$\sqrt{uv} \geq \frac{2uv}{u+v} \tag{A.6}$$

which leads trivially to $K_{bha}(x, z) \geq K_{chi2}(x, z)$.

Let us now show that: $K_{bha}(x, z) \leq \frac{1}{2}(1 + K_{chi2}(x, z))$. Again, let $u \in [0, 1]$ and $v \in [0, 1]$. We introduce $\alpha = \sqrt{uv}$ and $\beta = \frac{u+v}{2}$. We have:

$$1 + \frac{2uv}{u+v} - 2\sqrt{uv} = 1 + \frac{\alpha^2}{\beta} - 2\alpha \tag{A.7}$$

Multiplying the previous expression by β we obtain:

$$\beta(1 + \frac{2uv}{u+v} - 2\sqrt{uv}) = \beta + \alpha^2 - 2\alpha\beta. \tag{A.8}$$

Since $\beta \in [0,1]$, $\beta \geq \beta^2$ and

$$\beta + \alpha^2 - 2\alpha\beta \geq \beta^2 + \alpha^2 - 2\alpha\beta = (\alpha - \beta)^2 \geq 0. \tag{A.9}$$

Therefore, we have:

$$1 + \frac{2uv}{u+v} - 2\sqrt{uv} \geq 0 \tag{A.10}$$

which leads trivially to $\frac{1}{2}(1 + K_{chi2}(x,z)) \geq K_{bha}(x,z)$.

We can actually show that $\frac{1}{2}(1 + K_{chi2}(x,z))$ is a first-order Taylor expansion of $K_{bha}(x,z)$. Let $m = (x+z)/2$ and $\varepsilon = x - z$.

$$K_{bha}(x,z) = \sum_{d=1}^{D} \sqrt{x(d)z(d)} \tag{A.11}$$

$$= \sum_{d=1}^{D} m(d)\sqrt{1 - \left(\frac{\varepsilon(d)}{2m(d)}\right)^2}. \tag{A.12}$$

Assuming that $\varepsilon(d)/(2m(d)) \ll 1 \ \forall d$ and using a first-order Taylor expansion we get:

$$K_{bha}(x,z) \approx \sum_{d=1}^{D} m(d)\left(1 - \frac{\varepsilon(d)^2}{8m(d)^2}\right) \tag{A.13}$$

$$\approx 1 - \frac{1}{8}\sum_{d=1}^{D} \frac{\varepsilon(d)^2}{m(d)}. \tag{A.14}$$

Replacing m and ε by their definitions and reshuffling the terms leads to:

$$K_{bha}(x,z) \approx \frac{1}{2}\left(1 + 2\sum_{d=1}^{D} \frac{x(d)z(d)}{x(d) + z(d)}\right) \tag{A.15}$$

$$\approx \frac{1}{2}\left(1 + K_{chi2}(x,z)\right). \tag{A.16}$$

Bibliography

[Bay *et al.* 2006] H. Bay, T. Tuytelaars and L. van Gool. *SURF: Speeded up robust features*. In ECCV, 2006. 8

[Bilmes 1998] J. Bilmes. *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models*. Rapport technique, Department of Electrical Engineering and Computer Science, U.C. Berkeley, 1998. 23

[Bosch *et al.* 2006] A. Bosch, A. Zisserman and X. Munoz. *Scene Classification via pLSA*. In ECCV, pages 517–530, 2006. 38

[Bottou] Léon Bottou. *Stochastic Gradient Descent*. http://leon.bottou.org/projects/sgd. 63, 67, 69, 70, 79

[Boughhorbed *et al.* 2004] S. Boughhorbed, J.-P. Tarel and F. Fleuret. *Non-mercer kernels for svm object recognition*. In BMVC, 2004. 9

[Chang & Lin 2001] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. http://www.csie.ntu.edu.tw/ cjlin/libsvm. 69, 70

[Csurka *et al.* 2004] G. Csurka, C. Dance, L. Fan, J. Willamowski and C. Bray. *Visual Categorization with Bags of Keypoints*. In ECCV Workshop on Statistical Learning for Computer Vision, 2004. 2, 10, 62

[Dempster *et al.* 1977] A. Dempster, N. Laird and D. Ruin. *Maximum likelihood from incomplete data via the EM algorithm*. J. of the Royal Statistical Society, vol. 39, no. Series B, pages 1–38, 1977. 23, 40

[Deng *et al.* 2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. In CVPR, 2009. 61, 71

[Everingham *et al.* 2005] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi and J. Zhang. *The 2005 PASCAL Visual Object Classes Challenge*. In In Selected Proceedings of the First PASCAL Challenges Workshop, 2005. 2

[Everingham *et al.* 2006] M. Everingham, A. Zisserman, C. Williams and L. Van Gool. *The PASCAL Visual Object Classes Challenge 2006 Results*. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf, 2006. 2

[Everingham *et al.* 2007] M. Everingham, L. Van Gool, C. 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, 2007. 2, 8, 15, 56, 61, 70

[Everingham *et al.* 2008] M. Everingham, L. Van Gool, C. Williams, J. Winn and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results*. http://www.pascalnetwork.org/challenges/VOC/voc2008/workshop/index.html, 2008. 2

[Everingham *et al.* 2009] 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", 2009. 2

[Facebook] Facebook. `http://www.facebook.com`. 1

[Facebook 2009] Facebook. *Needle in a haystack: efficient storage of billions of photos*. `http://www.facebook.com/note.php?note_id=76191543919`, 2009. 1

[Farquhar *et al.* 2005] J. Farquhar, S. Szedmak, H. Meng and J. Shawe-Taylor. *Improving "bag-ofkeypoints" image categorisation: Generative Models and PDF-Kernels*. Rapport technique, University of Southampton, 2005. 10, 11, 12, 63

[Fei-Fei & Perona 2005] L. Fei-Fei and P. Perona. *A Bayesian hierarchical model for learning natural scene categories*. In CVPR, 2005. 7, 15

[Fei-Fei *et al.* 2006] L. Fei-Fei, R. Fergus and P. Perona. *One-shot learning of object categories*. IEEE PAMI, vol. 28, no. 4, 2006. 62

[Flickr] Flickr. `http://www.flickr.com`. 1

[Flickr 2009] Flickr. *4 Billion Photos on Flickr*. `http://blog.flickr.net/en/2009/10/12/4000000000/`, 2009. 1

[Fulkerson *et al.* 2008] B. Fulkerson, A. Vedaldi and S. Soatto. *Localizing Objects With Smart Dictionaries*. In ECCV, 2008. 10

[Gales 1998] M. Gales. *Maximum likelihood linear transformations for HMM-based speech recognition*. Computer Speech and Language, vol. 12, no. 2, pages 75–98, 1998. 35, 76

[Gales 2000] M. Gales. *Cluster adaptive training of hidden Markov models*. IEEE Trans. on Speech and Audio Processing, vol. 8, no. 4, pages 417–428, 2000. 35, 76

[Gauvain & Lee 1994] J.-L. Gauvain and C.-H. Lee. *Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains.* IEEE Trans. on speech and Audio Processing, vol. 2, no. 2, pages 291–298, 1994. 23, 24

[Goldberger *et al.* 2003] J. Goldberger, S. Gordon and H. Greenspan. *An Efficient Image Similarity Measure based on Approximations of KL-Divergence Between Two Gaussian Mixtures.* In ICCV, volume 1, pages 487–493, 2003. 3, 12, 21, 26

[Google] Google. `http://images.google.com`. 1

[Grauman & Darrell 2005a] K. Grauman and T. Darrell. *Efficient Image Matching with Distributions of Local Invariant Features.* In CVPR, 2005. 10

[Grauman & Darrell 2005b] K. Grauman and T. Darrell. *The Pyramid Match Kernel: discriminative Classification with Sets of Image Features.* In ICCV, 2005. 10

[Griffin *et al.* 2007] G. Griffin, A. Holub and P. Perona. *Caltech-256 Object Category Dataset.* Rapport technique 7694, California Institute of Technology, 2007. 15, 61

[Griffin 2007] G. Griffin. `http://www.vision.caltech.edu/CaltechChallenge2007/results/`, 2007. 15, 18, 35, 49

[Harris & Stephens 1998] C. Harris and M. Stephens. *A Combined Corner and Edge Detector.* In Alvey Vision Conference, 1998. 7

[Haussler 1999] D. Haussler. *Convolution kernels on discrete structures.* Rapport technique Technical Report UCS-CRL-99-10, 1999. 9

[Hershey & Olsen 2007] J.R. Hershey and P.A. Olsen. *Approximating the kullback-leibler divergence between gaussian mixture models.* In ICASSP, 2007. 27, 42

[Hershey & Olsen 2008] J.R. Hershey and P.A. Olsen. *Variational Bhattacharyya Divergence for Hidden Markov Models.* In ICASSP, pages 4557–4560, 2008. 25

[Indyk & Thaper 2003] P. Indyk and N. Thaper. *Fast Image Retrieval via Embeddings.* In 3rd Intl Workshop on Statistical and Computational Theories of Vision, Nice, France, 2003. 10

[Jebara & Kondor 2003] T. Jebara and R. Kondor. *Bhattacharyya and expected likelihood kernels.* In COLT, pages 57–73, 2003. 3, 13, 21, 24, 25, 54, 63

[Jebara *et al.* 2004] T. Jebara, R. Kondor and A. Howard. *Probability product kernels.* JMLR, Special Topic on Learning Theory, vol. 5, pages 819–944, 2004. 3, 13, 21

[Jégou *et al.* 2007] H. Jégou, H. Harzallah and C. Schmid. *A Contextual Dissimilarity Measure for Accurate and Efficient Image Search.* In IEEE CVPR, 2007. 51

[Joachims 2006] T. Joachims. *Training Linear SVMs in Linear Time.* In KDD, 2006. 62

[Jurie & Triggs 2005] F. Jurie and B. Triggs. *Creating efficient codebooks for visual recognition.* In ICCV, 2005. 10

[Ke & Sukthankar 2004] Y. Ke and R. Sukthankar. *PCA-SIFT: A More Distinctive Representation for Local Image Descriptors.* In CVPR, 2004. 8

[Krishnapruam *et al.* 2005] B. Krishnapruam, L. Carin, M. Figueiredo and A. Hartemink. *Sparse multinomial logistic regression: Fast algorithms and generalization bounds.* IEEE Trans. on PAMI, vol. 27, no. 6, pages 957–968, 2005. 14, 28

[Kuhn *et al.* 2000] R. Kuhn, J.-C. Junqua, P. Nguyen and N. Niedzielski. *Rapid Speaker Adaptation in Eigenvoice Space.* IEEE Trans. on Speech and Audio Processing, vol. 8, no. 6, pages 695–707, 2000. 35, 76

[Kullback 1968] S. Kullback. Information theory and statistics. Dove Publications Inc., Mineola, New York, 1968. 13

[Lazebnik & Raginsky 2009] S. Lazebnik and M. Raginsky. *Supervised Learning of Quantizer Codebooks by Information Loss Minimization.* IEEE Transactions on In Pattern Analysis and Machine Intelligence, vol. 31, pages 1294–1309, 2009. 10

[Lazebnik *et al.* 2006] S. Lazebnik, C. Schmid and J. Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories.* In CVPR, 2006. 15, 33, 49, 61

[Leggetter & Woodland 1995] C. Leggetter and P. Woodland. *Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models.* Computer Speech and Language, vol. 9, pages 806–814, 1995. 35, 76

[Li *et al.* 2009] Y. Li, D. Crandall and D. Huttenlocher. *Landmark Classification in Large-scale Image Collections.* In ICCV, 2009. 62, 73

[Lindeberg 1998] T. Lindeberg. *Feature Detection with Automatic Scale Selection.* IJCV, 1998. 7

[Liu & Perronnin 2008] Y. Liu and F. Perronnin. *A similarity measure between unordered vector sets with application to image categorization.* In CVPR, 2008. 3

[Lowe 2004] D.G. Lowe. *Distinctive image features from scale-invariant keypoints.* IJCV, vol. 60, no. 2, pages 91–110, 2004. 8, 27

[Lyu 2005] S. Lyu. *Mercer Kernels for Object Recognition with Local Features.* In CVPR, 2005. 9

[Maji & Berg 2009] S. Maji and A. Berg. *Max-Margin Additive Classifiers for Detection.* In ICCV, 2009. 62, 65, 68, 73

[Maji *et al.* 2008] S. Maji, A. Berg and J. Malik. *Classification using intersection kernel support vector machines is efficient.* In CVPR, 2008. 62, 63, 67

[Marée *et al.* 2005] R. Marée, P. Geurts, J. Piater and L. Wehenkel. *Random subwindows for robust image classification.* In CVPR, 2005. 7

[Marszalek *et al.* 2007] M. Marszalek, C. Schmid, H. Harzallah and J. van deWeijer. *Learning representations for visual object class recognition.* Rapport technique, 2007. 29

[Mikolajczyk & Schmid 2001] K. Mikolajczyk and C. Schmid. *Indexing Based on Scale Invariant Interest Points.* In ICCV, 2001. 7

[Mikolajczyk & Schmid 2004] K. Mikolajczyk and C. Schmid. *Scale and Affine Invariant Interest Point Detectors.* IJCV, 2004. 7

[Mikolajczyk *et al.* 2005] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool. *A comparison of affine region detectors.* IJCV, 2005. 7

[Moosmann *et al.* 2006] F. Moosmann, B. Triggs and F. Jurie. *Fast Discriminative Visual Codebooks using Randomized Clustering Forests.* In NIPS, 2006. 11

[Moreno *et al.* 2003] P. Moreno, P. Ho and N. Vasconcelos. *A Kullback-Leibler divergence based kernel for SVM classification in multimedia application.* In Neural Information Proceeding Systems, 2003. 3, 12, 13, 21, 26

[Nowak *et al.* 2006] E. Nowak, F. Jurie and B. Triggs. *Sampling strategies for bag-of-features image classification.* In ECCV, 2006. 7

[Oliva & Torralba 2001] A. Oliva and A. Torralba. *Modeling the shape of the scene: a holistic representation of the spatial envelope.* In IJCV, 2001. 15

[Pekalska *et al.* 2002] E. Pekalska, P. Paclik and R. Duin. *A generalized kernel approach to dissimilarity-based classification.* Journal of Machine Learning Research, special issue on kernel methods, vol. 2, no. 2, pages 175–211, 2002. 37

[Perronnin & Liu 2009] F. Perronnin and Y. Liu. *Modeling Images as Mixtures of Reference Images.* In CVPR, 2009. 4

[Perronnin *et al.* 2006] F. Perronnin, C. Dance, G. Csurka and M. Bressan. *Adapted vocabularies for generic visual categorization.* In ECCV, 2006. 11, 22

[Perronnin *et al.* 2009] F. Perronnin, Y. Liu and J.M. Renders. *A Family of Conteuxtual Measures of Similarity between Distributions with Application to Image Retrieval.* In CVPR, 2009. 4

[Perronnin *et al.* 2010] F. Perronnin, J. Sanchez and Y. Liu. *Large-Scale Image Categorization with Explicit Data Embedding.* In CVPR, 2010. 4

[Picsearch] Picsearch. http://www.picsearch.com. 1

[Ponte & Croft 1998] J. Ponte and W. Croft. *A Language Modeling Approach to Information Retrieval.* In ACM SIGIR, 1998. 52, 53

[Quelhas *et al.* 2005] P. Quelhas, F. Monay, J.-M Odobez, D. Gatica-Perez, T. Tuytelaars and L. Van Gool. *Modeling scenes with local descriptors and latent aspects.* In ICCV, 2005. 38

[Rahimi & Recht 2007] A. Rahimi and B. Recht. *Random features for large-scale kernel machines.* In NIPS, 2007. 4, 62, 69, 71, 73

[Rasiwasia & Vasconcelos 2008] N. Rasiwasia and N. Vasconcelos. *Scene Classification with Low-Dimensional Semantic Spaces and Weak Supervision.* In IEEE CVPR, 2008. 38

[Rasiwasia *et al.* 2007] N. Rasiwasia, P. Moreno and N. Vasconcelos. *Bridging the Gap: Query by Semantic example.* IEEE Trans. on Multimedia, vol. 9, no. 5, pages 923–938, 2007. 38

[Reynolds *et al.* 2000] D.A. Reynolds, T.F. Quatieri and R.B. Dunn. *Speaker verification using adapted Gaussian mixture models.* Digital Signal Processing, vol. 10, no. 1-3, pages 19–41, 2000. 24, 25

[Rubner *et al.* 1998] Y. Rubner, C. Tomasi and L.J. Guibas. *A metric for distributions with applications to image databases.* In ICCV, 1998. 9

[Schölkopf *et al.* 1998] B. Schölkopf, A. Smola and K.-R. Müller. *Non-linear component analysis as a kernel eigenvalue problem.* In Neural Computation, 1998. 4, 62, 64

[Shalev-Shwartz & Srebro 2008] S. Shalev-Shwartz and N. Srebro. *SVM optimization: inverse dependence on training set size.* In ICML, 2008. 62

[Shalev-Shwartz *et al.* 2007] S. Shalev-Shwartz, Y. Singer and N. Srebro. *Pegasos: Primal Estimate sub-Gradient Solver for SVM.* In ICML, 2007. 62, 79

[Sivic & Zisserman 2003] J. Sivic and A. Zisserman. *Video Google: A Text Retrieval Approach to Object Matching in Videos.* In ICCV, volume 2, pages 1470–1477, 2003. 2, 10

[van de Sande *et al.* 2010] K. van de Sande, T. Gevers and C. Snoek. *Evaluating Color Descriptors for Object and Scene Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, no. in press, 2010. 8

[van de Weijer & Schmid 2006] J. van de Weijer and C. Schmid. *Coloring Local Feature Extraction.* In ECCV, 2006. 8

[van Gemert *et al.* 2008] J. van Gemert, J. Geusebroek, C. Veenman and A. Smeulders. *Kernel codebooks for scene categorization.* In ECCV, 2008. 11

[Vapnik 1995] V. Vapnik. The nature of statisticsal learning. Springer, N.Y., 1995. 14

[Varma & Ray 2007] M. Varma and D. Ray. *Learning The Discriminative Power-Invariance Trade-Off.* In ICCV, 2007. 35, 76

[Vasconcelos *et al.* 2004] N. Vasconcelos, P.P. Ho and P.J. Moreno. *The Kullback-Leibler kernel as a framework for discriminant and localized representations for visual recognition.* In ECCV, 2004. 3, 12, 21, 26

[Vasconcelos 2004] N. Vasconcelos. *On the efficient evaluation of probabilistic similarity functions for image retrieval.* IEEE Trans. on Information Theory, vol. 50, no. 7, pages 1482–1496, 2004. 3, 12, 21, 26

[Vedaldi *et al.* 2009] A. Vedaldi, V. Gulshan, M. Varma and A. Zisserman. *Multiple Kernels for Object Detection.* In ICCV, 2009. 12, 62, 64

[Wallraven *et al.* 2003] C. Wallraven, B. Caputo and A. Graf. *Recognition with Local Features: the Kernel Recipe.* In ICCV, 2003. 9

[Wang *et al.* 2009] G. Wang, D. Hoiem and D. Forsyth. *Learning Image Similarity from Flickr Groups Using Stochastic Intersection Kernel Machines.* In ICCV, 2009. 62, 68, 73

[Williams & Seeger 2001] C. Williams and M. Seeger. *Using the Nyström method to speed up kernel machines.* In NIPS, 2001. 62, 64

[Winn *et al.* 2005] J. Winn, A. Criminisi and T. Minka. *Object categorization by learned universal visual dictionary.* In ICCV, 2005. 7, 10, 11

[Wu & Rehg 2009] J. Wu and J.M. Rehg. *Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel.* In ICCV, 2009. 10

[Yahoo!] Yahoo! `http://images.search.yahoo.com`. 1

[Yang *et al.* 2008] L. Yang, R. Jin, R. Sukthankar and F. Jurie. *Unifying Discriminative Visual Codebook Generation with Classi¯er Training for Object Category Recognition.* In CVPR, 2008. 11

[Young *et al.* 2002] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, S. Povey, V. Valtchev and P. Woodland. *The HTK book (version 3.2.1).* Cambridge University Engineering Department, Dec 2002. 28, 32

[Zhai & Lafferty 2001] C. Zhai and J. Lafferty. *A Study of Smoothing Methods for Language Model Applied to Ad Hoc Information Retrieval.* In ACM SIGIR, 2001. 52

[Zhang *et al.* 2005] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid. *Local features and kernels for classification of texture and object categories: an in-depth study.* Rapport technique RR-5737, INRIA, 2005. 10, 12, 28

[Zhang *et al.* 2007] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik and Cordelia Schmid. *Local features and kernels for classification of texture and object categories: a comprehensive study.* IJCV, vol. 73, no. 2, 2007. 62

[Zhao *et al.* 2007] D. Zhao, Z. Lin and X. Tang. *Contextual Distance for Data Perception.* In IEEE ICCV, 2007. 51

[Zhou *et al.* 2009] X. Zhou, N. Cui, Z. Li, F. Liang and T.S. Huang. *Hierarchical Gaussianization for Image Classification.* In ICCV, 2009. 33, 49