Modeling Images as Mixtures of Reference Images

Florent Perronnin and Yan Liu*
Textual and Visual Pattern Analysis (TVPA)
Xerox Research Centre Europe (XRCE), France
Florent.Perronnin@xerox.com, Yan.Liu@xerox.com

Abstract

A state-of-the-art approach to measure the similarity of two images is to model each image by a continuous distribution, generally a Gaussian mixture model (GMM), and to compute a probabilistic similarity between the GMMs. One limitation of traditional measures such as the Kullback-Leibler (KL) divergence and the Probability Product Kernel (PPK) is that they measure a global match of distributions.

*This paper introduces a novel image representation. We propose 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. 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. 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.*

We apply this novel representation to the image categorization problem and compare its performance to traditional kernel-based methods. We demonstrate on the PASCAL VOC 2007 dataset a consistent increase in classification accuracy.

1. Introduction

We consider the image categorization problem which consists in assigning to an image one or multiple labels based on its semantic content. The most successful image representation to date for this task is certainly the *bag-of-patches* which describes an image as an unordered set of low-level local feature vectors. While the bag-of-patches discards important information about the image structure, it demonstrated state-of-the-art performance in recent evaluations [4, 5].

Kernel-based image classification requires the definition of a measure of similarity between images. *Model-based* similarities are a state-of-the-art approach to defining a measure of similarity between vector sets. They consist in (i) modeling each vector set as a distribution and (ii) defining the measure of similarity between the vector sets as the similarity between their respective distributions. There are two leading model-based methods in the case of bag-of-patches representations.

The first one, the *bag-of-visual-words* (BOV), models an image as a discrete distribution [20, 2]. The BOV is based on an intermediate representation, the *visual vocabulary*, which is estimated offline on a large set of low-level feature vectors. An image is characterized as a histogram of visual-word counts, *i.e*. a multinomial. The most commonly used measure of similarity between bag-of-words representations is the χ^2 kernel [23]. One limitation of the BOV is the assumption that the distribution of feature vectors in any image can be known a priori.

The second one – which is the focus of this paper – models an image as a continuous distribution, generally a Gaussian mixture model (GMM) [6, 15, 21, 22, 12]. The most commonly used measures of similarity between two GMMs are the Kullback-Leibler (KL) divergence [6, 15, 22, 21, 7] or the Probability Product Kernel (PPK) [9, 10]. As there is generally no closed-form formula for the KL or PPK between two GMMs, one should resort to approximations.

While many of the previously cited papers focus on how to best approximate the true KL or PPK, none of them addresses their inherent limitation. Indeed, traditional measures of similarity between distributions give a high similarity when two distributions match globally but a low similarity when they match only partially (this effect is exacerbated in the case of KL because of the $\log$ function). 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.

*Yan Liu is a Ph.D. student in the Laboratoire d'Informatique en Image et Systmes d'Information (LIRIS) at the Ecole Centrale de Lyon (ECL).

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

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.

We thus propose to approximate an image, modeled as a GMM, as a convex combination of K reference image GMMs 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 between individual Gaussian components) which is significantly different from traditional measures. The vector of mixture weights may then be used as input to a discriminative classifier for categorization.

Our work can be related to dissimilarity-based learning which is an alternative to traditional kernel-based learning. In [16], 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 our approach is that in [16] 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 image representation. This results in a measure of similarity which better takes into account strong matches.

Our work can also be related to [17, 1, 18, 19]. 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 [17, 1], in which case there is no guarantee that they are semantically meaningful, or in a supervised manner [18, 19] 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. While reference images might be less semantically meaningful than concepts learned in a supervised manner, they are more meaningful than visual words.

The remainder of the article is organized as follows. In section 2, we briefly review the KL and PPK. We provide definitions, show how they can be approximated in the case of mixture models and analyze their limitations. This leads us to introduce our novel image representation in section 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 4 we provide experimental results showing that the proposed framework outperforms a standard kernel-based classifier employing the Kullback-Leibler Kernel (KLK) or the PPK.

2. Measures between Probability Distributions

Throughout this article, we focus on the KL divergence and the PPK. We first provide definitions and show how these measures can be approximated in the case of mixture models. We then discuss the limitations of such measures. In the following, $f(x) = \sum_{i=1}^{M} \alpha_i f_i(x)$ and $g(x) = \sum_{j=1}^{N} \beta_i g_i(x)$ are the two mixture models to be compared.

2.1. Kullback-Leibler divergence

The KL is defined as:

$$KL(f,g) = \int_x f(x) \log\left(\frac{f(x)}{g(x)}\right) dx. \tag{1}$$

It can be approximated using Monte-Carlo sampling [15, 22], the unscented transform [6], a mapping of Gaussian components [6, 21] or variational bound methods [7]. We now focus on the latter approximation. The idea of Hershey and Olsen is to write:

$$KL(f,g) = H(f,g) - H(f,f). \tag{2}$$

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

$$H(f,g) \leq -\sum_{i=1}^{M} \alpha_i \log\left(\sum_{j=1}^{N} \beta_j \exp(-H(f_i,g_j))\right). \tag{3}$$

A closed-form formula exists for the cross-entropy $H(f_i,g_j)$ between Gaussians. Since this KL approximation is the difference of two bounds, it is not a bound. Because the KL is asymmetric, one generally considers its symmetrized version:

$$SKL(f,g) = KL(f,g) + KL(g,f). \tag{4}$$

2.2. Probability Product Kernel

The PPK is defined as:

$$PPK_\rho(f,g) = \int_x (f(x)g(x))^\rho \, dx. \tag{5}$$

In this article we focus on the Bhattacharyya similarity:

$$B(f,g) = PPK_{1/2}(f,g). \tag{6}$$

An approximation similar to that used for the KL was proposed in [8] by Hershey and Olsen. It leads to the following bound:

$$B(f,g) \geq \sqrt{\sum_{i,j} \alpha_i \beta_j B^2(f_i,g_j)} \tag{7}$$

A closed form formula exists for the Bhattacharyya similarity $B(f_i,g_j)$ between two Gaussians.

2.3. Limitations

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:

$$q = \frac{1}{2}\mathcal{N}(+2, 1) + \frac{1}{2}\mathcal{N}(-2, 1). \quad (8)$$

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

$$p_1 = \mathcal{N}(-2, 1)\,, \quad (9)$$
$$p_2 = \mathcal{N}(2, 1)\,, \quad (10)$$
$$p_3 = \frac{1}{2}\mathcal{N}(2+\delta, 1) + \frac{1}{2}\mathcal{N}(-2-\delta, 1). \quad (11)$$

where δ is a non-negative value (c.f. Figure 1). To study the limitations of the SKL and PPK (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 $B(q, p_1) = B(q, p_2)$ by symmetry. $SKL(q, p_3)$ (resp. $B(q, p_3)$) is an increasing (resp. decreasing) function of δ with $SKL(q, p_3) = 0$ (resp. $B(q, p_3) = 1$) 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 $B(q, p_1) = B(q, p_2) = B(q, p_3)$. 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 1. 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.

3. Images as 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

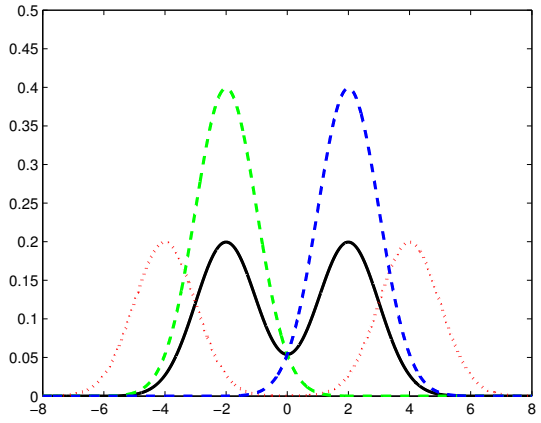


Figure 1. The SKL between q (black straight line) and p_1 or p_2 (green and blue dashed lines respectively) is approximately the same as the SKL between q and p_3 (dotted red line).

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:

$$E = \int_x q(x) \log \left(\sum_{k=1}^{K} \omega_k p_k(x) \right) dx\,. \quad (12)$$

under the constraints

$$\omega_k \geq 0\,, \forall k \text{ and } \sum_{k=1}^{K} \omega_k = 1. \quad (13)$$

This is a convex optimization problem which can be solved iteratively using the Expectation-Maximization (EM) algorithm [3]. 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:

$$\gamma_k(x) = \frac{\omega_k p_k(x)}{\sum_{j=1}^{K} \omega_j p_j(x)}\,. \quad (14)$$

The M-step leads to the following estimate:

$$\hat{\omega}_k = \int_x q(x)\gamma_k(x)dx. \quad (15)$$

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.

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 (12) as follows:

$$E \approx \frac{1}{T} \sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} \omega_k p_k(x_t) \right). \quad (16)$$

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:

$$\hat{\omega}_k = \frac{1}{T} \sum_{t=1}^{T} \gamma_k(x_t). \quad (17)$$

We note that we would have obtained the same re-estimation formula if we had applied the law of large numbers on equation (15) directly.

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 (16) 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 [12], Liu and Perronnin 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 [12].

We rewrite the objective function (12) as follows:

$$E = \sum_{i=1}^{N} \pi_i \int_x q_i(x) \log \left(\sum_{k=1}^{K} \omega_k \sum_{j=1}^{N_k} \pi_{k,j} p_{k,j}(x) \right) dx. \quad (18)$$

We use the idea proposed by Hershey and Olsen [7] 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:

$$0 \leq \gamma_{i,k,j} \leq 1 \text{ and } \sum_{k=1}^{K} \sum_{j=1}^{N_k} \gamma_{i,k,j} = 1. \quad (19)$$

The function (18) becomes:

$$E = \sum_{i} \pi_i \int_x q_i(x) \log \left(\sum_{k,j} \gamma_{i,k,j} \frac{\omega_k \pi_{k,j} p_{k,j}(x)}{\gamma_{i,k,j}} \right) dx. \quad (20)$$

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

$$E \geq \sum_{i} \pi_i \int_x q_i(x) \sum_{k,j} \gamma_{i,k,j} \log \left(\frac{\omega_k \pi_{k,j} p_{k,j}(x)}{\gamma_{i,k,j}} \right) dx \,. \quad (21)$$

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

$$E \geq \sum_{i} \pi_i \log \left(\sum_{k,j} \omega_k \pi_{k,j} \exp(-H_{i,k,j}) \right). \quad (22)$$

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

$$H_{i,k,j} = H(q_k, p_{k,j}) = -\int_x q_k(x) \log p_{k,j}(x) dx. \quad (23)$$

$H_{i,k,j}$ 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:

$$\gamma_{i,k,j} = \frac{\omega_k \pi_{k,j} \exp(-H_{i,k,j})}{\sum_{k,j} \omega_k \pi_{k,j} \exp(-H_{i,k,j})} \quad (24)$$

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

$$\hat{\omega}_k = \sum_{i,j} \pi_i \gamma_{i,k,j}. \quad (25)$$

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 (3)). 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 [12], 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.

3.3. Convergence Issues

Let us go back to our toy example of section 2.3. 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 (18) 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 (12) is maximized by $\omega_j = 1$ and $\omega_i = 0, \forall i \neq j$. This undesired effect is prevented by early stopping.

3.4. Beyond KL

As explained in section 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 $B_{i,k,j} = B(q_i, p_{k,j})$ in the E-step (24).

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. 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 now provide the E- and M-step.

E-step:

$$\gamma_{i,k,j} = \frac{\omega_k \pi_{k,j} B_{i,k,j}^2}{\sum_{k,j} \omega_k \pi_{k,j} B_{i,k,j}^2} \,. \tag{26}$$

M-step:

$$\hat{\omega}_k = \frac{(\sum_i \pi_i \sum_j \sqrt{\pi_{k,j}\gamma_{i,k,j}} B_{i,k,j})^2}{\sum_k (\sum_i \pi_i \sum_j \sqrt{\pi_{k,j}\gamma_{i,k,j}} B_{i,k,j})^2} \,. \tag{27}$$

Details are left in the appendix. Preliminary experiments showed that the principled computation of weights always outperformed the heuristic approach.

4. Experimental Results

We now apply this representation to an image categorization task. We first describe the database, then the experimental setup and finally the results.

4.1. PASCAL VOC 2007

We used the PASCAL VOC 2007 database [4] 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. During the VOC 2007 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.

4.2. Experimental Setup

Low-level feature vectors are extracted on regular grids at multiple scales. There is an average of 1,000 feature vectors per image. We make use of two types of low-level features: local histograms of orientations as described in [13] (later referred to as ORH) and simple RGB statistics (later

referred to as COL). In both cases, the dimensionality of the feature vectors was reduced through Principal Component Analysis.

We evaluated three baseline systems:

- A standard BOV with χ^2 kernel [23].
- The method of [12] with the KL kernel (KLK). The KLK between two distributions p and q is defined as:

$$K_{klk}(p,q) = \exp\left(-\gamma SKL(p,q)\right) . \qquad (28)$$

  To set parameter γ we followed [23]: γ is equal to the inverse of the mean of the SKL between two GMMs as estimated on a subset of the whole training set.

- The method of [12] with the PPK.

For the second and third 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. We used the fast scoring described in [12] as it was shown to have little influence on the classification accuracy.

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

- MOM KL sampling: c.f. section 3.1.
- MOM KL lower-bound: c.f. section 3.2.
- MOM PPK: c.f. section 3.4.

For a fair comparison, all image GMMs are also estimated using the adaptation framework of [12]. We used as reference images the 5,011 training images.

For all categorization systems, we used the Sparse Logistic Regression (SLR) [11] as a discriminative classifier. One classifier is trained per class in a one-versus-all manner. For the three proposed approaches, we apply SLR directly to the vectors of mixture weights. For a fair comparison, we use the generalized kernel framework of [16] for the three baseline systems: an image is represented as a vector of similarities / distances to the set of training images and we apply SLR to these vectors. In our own experience, using (i) a regularized kernel classifier such as the Support Vector Machine (SVM) with a given kernel $K(.,.)$ or (ii) the framework of [16] with $K(.,.)$ as similarity measure and a regularized linear classifier such as SLR lead to similar results.

In all cases, 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).

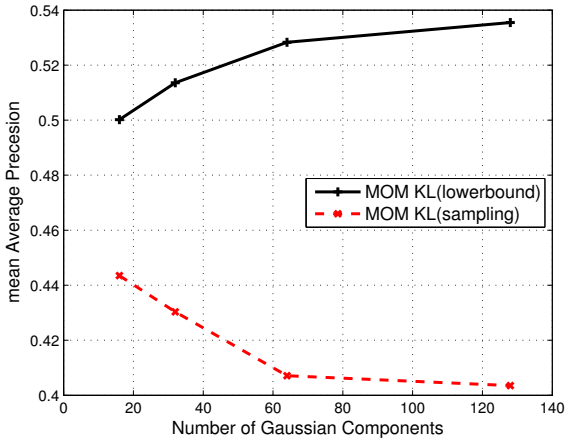


Figure 2. Mean AP for the sampling and lower-bound approximations of MOM KL for the system based on ORH features only.

4.3. Results

Lower-bound vs sampling. We start with the comparison of the sampling and lower-bound approximations for MOM KL. Results are shown on Figure 2 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 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 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.

Comparative evaluation. The results of the comparison of KLK with MOM KL and PPK with MOM PPK are shown on Figure 4 for the different features. We did not represent the performance of BOV on these figures because BOV typically requires a larger number of Gaussians. The best results we obtained with BOV was a mean AP of 52.6% with approximately 4000 Gaussians (for ORH + COL). 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 be-

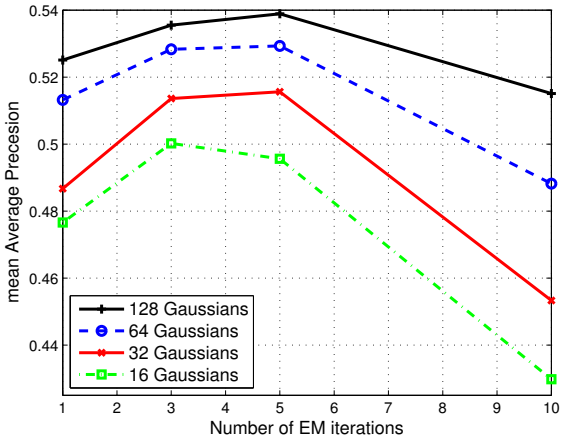


Figure 3. Influence of the number of EM iterations on the mean AP for MOM KL lower-bound for different numbers of Gaussian components (system based on ORH features only).

lieve 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 2.3).

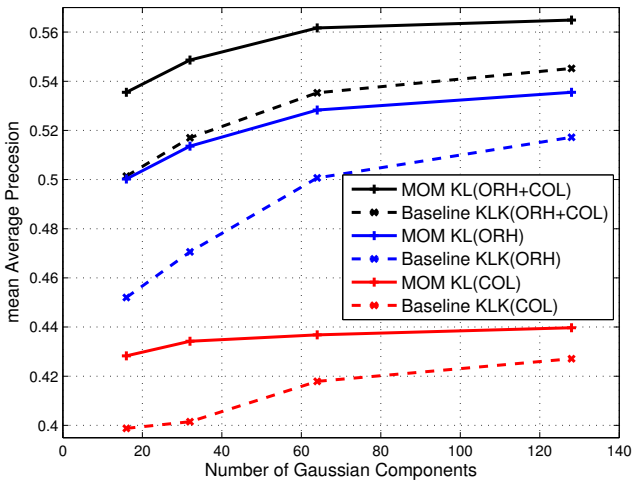
As we used the standard VOC07 protocol, our results can be compared to those published in the literature. The best results reported on this dataset during the challenge was 59.4% (INRIA-genetic) [14]. We would like to outline that the cost of training and testing our system is significantly lower compared 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.

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 the framework of [12] (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 $K = 128$.

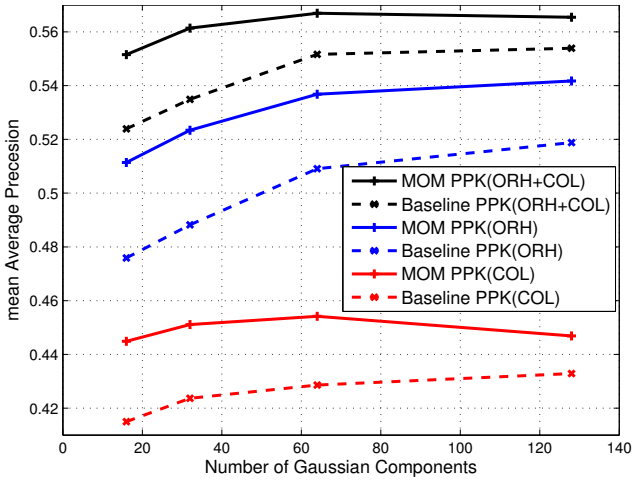
5. Conclusion

We presented in this article 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 an image classification task and showed on the PASCAL VOC07 dataset a consistent increase in classification accuracy.



(a)



(b)

Figure 4. Comparison of the proposed algorithms to traditional kernel methods: (a) MOM KL versus KLK and (b) MOM PPK versus PPK. 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).

A. Alternative Objective Function

Instead of minimizing the KL between q and $\sum_{k=1}^{K} \omega_k p_k$, we can maximize their Bhattacharyya

similarity:

$$E = \int_x \sqrt{q(x)} \sqrt{\left(\sum_{k=1}^{K} \omega_k p_k(x) \right)} dx \quad (29)$$

$$= \int_x \sqrt{\sum_{i=1}^{N} \pi_i q_i(x) \sum_{k=1}^{K} \omega_k \sum_{j=1}^{N_k} \pi_{k,j} p_{k,j}(x)} dx \quad (30)$$

We apply a first time Jensen's inequality and write:

$$E \geq \sum_i \pi_i \int_x \sqrt{q_i(x) \sum_k \omega_k \sum_j \pi_{k,j} p_{k,j}(x)} dx. \quad (31)$$

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:

$$\sum_i \pi_i \int_x \sqrt{q_i(x) \sum_{k,j} \gamma_{i,k,j} \frac{\omega_k \pi_{k,j} p_{k,j}(x)}{\gamma_{i,k,j}}} dx. \quad (32)$$

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

$$E \geq \sum_i \pi_i \sum_{k,j} \sqrt{\omega_k \pi_{k,j} \gamma_{i,k,j}} B_{i,k,j}. \quad (33)$$

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 equations (26) and (27).

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via PLSA. In *IEEE ECCV*, 2006.

[2] 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.

[3] A. Dempster, N. Laird, and D. Ruin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society*, 39(Series B):1–38, 1977.

[4] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge 2007 results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[5] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[6] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures. In *IEEE ICCV*, 2003.

[7] J. Hershey and P. Olsen. Approximating the kullback-leibler divergence between gaussian mixture models. In *IEEE ICASSP*, 2007.

[8] J. Hershey and P. Olsen. Variational Bhattacharyya divergence for Hidden Markov Models. In *IEEE ICASSP*, 2008.

[9] T. Jebara and R. Kondor. Bhattacharyya and expected likelihood kernels. In *COLT*, 2003.

[10] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *JMLR, Special Topic on Learning Theory*, 5:819–944, 2004.

[11] B. Krishnapruam, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Trans. on PAMI*, 27(6):957–968, 2005.

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

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[14] M. Marszalek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning representations for visual object class recognition. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/workshop/marszalek.pdf, 2007.

[15] P. Moreno, P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia application. In *NIPS*, 2003.

[16] E. Pekalska, P. Paclik, and R. Duin. A generalized kernel approach to dissimilarity-based classification. *JMLR, special issue on kernel methods*, 2(2):175–211, 2002.

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

[18] N. Rasiwasia, P. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *IEEE Trans. on MM*, 9(5):923–938, 2007.

[19] N. Rasiwasia and N. Vasconcelos. Scene classification with low-dimensional semantic spaces and weak supervision. In *IEEE CVPR*, 2008.

[20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[21] N. Vasconcelos. On the efficient evaluation of probabilistic similarity functions for image retrieval. *IEEE Trans. on IT*, 50(7):1482–1496, 2004.

[22] N. Vasconcelos, P. Ho, and P. Moreno. The Kullback-Leibler kernel as a framework for discriminant and localized representations for visual recognition. In *IEEE ECCV*, 2004.

[23] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: an in-depth study. Technical Report RR-5737, INRIA, 2005.