On the Complexity of Probabilistic Image Retrieval

Nuno Vasconcelos
Compaq Computer Corporation
Cambridge Research Laboratory
nuno.vasconcelos@compaq.com

Abstract

Probabilistic image retrieval approaches can lead to significant gains over standard retrieval techniques. However, this occurs at the cost of a significant increase in computational complexity. In fact, closed-form solutions for probabilistic retrieval are currently available only for simple representations such as the Gaussian and the histogram. We analyze the case of mixture densities and exploit the asymptotic equivalence between likelihood and Kullback-Leibler divergence to derive solutions for these models. In particular, 1) we show that the divergence can be computed exactly for vector quantizers and, 2) has an approximate solution for Gaussian mixtures that introduces no significant degradation of the resulting similarity judgments. In both cases, the new solutions have closed-form and computational complexity equivalent to that of standard retrieval approaches, but significantly better retrieval performance.

1. Introduction

The design of an architecture for content-based image retrieval (CBIR) requires careful analysis of the interplay between three fundamental components: a feature transformation, a feature representation, and a similarity function. While such an analysis exposes several conflicting requirements, it also reveals that it is possible to achieve good compromise solutions [10, 9]. Among these, one of the most promising is to rely on probabilistic retrieval principles: given a feature transformation, 1) an estimate of the density associated to each image class (e.g. the histogram of its feature values) is computed off-line, 2) a collection of feature vectors is extracted from each query image, 3) the likelihood of these vectors is evaluated for each class, and 4) the maximum-likelihood (ML) class is selected. The appeal of probabilistic retrieval derives from several interesting properties. In particular, it can be shown that 1) it minimizes the probability of retrieval error, 2) it generalizes a significant number of known retrieval approaches, 3) it establishes a common framework for handling global (based on entire images) and local (based on image regions) similarity, 4) it provides a natural foundation for the design of learning (relevance feedback) algorithms through Bayesian inference, and 5) it allows the natural integration of multiple content modalities (e.g. queries taking into account both images and the text of associated captions).

There is, nevertheless, one significant hurdle to the practical implementation of probabilistic retrieval: evaluating the likelihood of Q query features (complexity O(Q)) can be overwhelming when the query set has large cardinality. This complexity is always significantly higher than that of standard solutions (e.g. histogram intersection (HI)) that, operating directly on feature representations characterized by a small number P of parameters (e.g. histogram bins), have complexity O(P) where P is orders of magnitude smaller than Q (while there may be many thousands of feature vectors in an image, the histogram typically has less than 256 bins).

In this paper, we investigate solutions to the computational complexity of probabilistic retrieval. The starting point is the well known fact that, for a given image class in the database, the likelihood of the query converges asymptotically to the Kullback-Leibler (KL) divergence between the underlying query density and the density of that class. Since, for P-parameter models, one would expect the KL divergence to have computational cost O(P), there is no a priori reason to believe that it should be more expensive than that of the standard solutions. In practice, however, the use of the KL divergence as a similarity function for image retrieval has been limited to systems based on very simple density models, such as the Gaussian or the histogram [4, 7]. These models are either too simplistic to accurately describe the densities associated with real images (Gaussian) or too rigid to be usable in the high-dimensional features spaces required for accurate image discrimination (histogram). The fundamental reason behind this restriction is the fact that, in general, the KL divergence between two densities does not have a closed-form solution. The Gaussian and histogram cases are, in fact, among the few cases...
for which a closed-form expression is available [5, 4].

We consider two models, vector quantizers (VQ) and mixture densities that, being more generic than the Gaussian and more flexible than the histogram, overcome the fundamental limitations of these representations. Unlike the histogram, both models partition the feature space according to the distribution of the data, leading to density estimates whose complexity is determined by the complexity of this distribution (number of clusters that it contains) and not the dimension of the space itself. This enables estimates of reasonable accuracy on high dimensional spaces, something that is impossible with histograms. The only, yet significant, difference between the two models is that while (like regular histograms) VQ-based density estimation procedures partition the space into mutually exclusive cells, mixture densities rely on soft partitions. These partitions lead to better estimates but make the theoretical analysis significantly harder.

In fact, the main theoretical contribution of this work is to show that a closed form solution for the KL divergence does exist in the VQ case. Curiously, this result is obtained by exploiting the relationships between the VQ and the Gaussian mixture and therefore also provides new insights on the KL divergence between mixtures. In particular, it suggests a closed form approximation for this quantity that works well in high dimensions. In practice, these theoretical results have the important consequence of enabling probabilistic retrieval with sophisticated density models at a computational cost similar to that of standard approaches such as HI. We present experimental evidence illustrating that, from the point of view of retrieval accuracy, probabilistic retrieval significantly outperforms the most popular among these approaches.

2. Feature representation

We start by briefly introducing the four models under consideration and analyzing the relationships between them. Because all models can be seen as particular cases of mixture densities, we consider these first.

2.1. Mixture densities

A mixture density has the form

\[ P(x) = \sum_{c=1}^{C} P(x|\omega_c)P(\omega_c), \tag{1} \]

where \( C \) is a number of feature classes\(^3\), \( \{P(x|\omega_c)\}_{c=1}^{C} \) a sequence of feature class-conditional densities or mixture components, and \( \{P(\omega_c)\}_{c=1}^{C} \) a sequence of feature class probabilities. Mixture densities model processes with hidden structure: one among the \( C \) feature classes is first selected according to the \( \{P(\omega_c)\} \), and the observed data is then drawn according to the respective feature class-conditional density. These densities can be any valid probability density functions, i.e., any set of non-negative functions integrating to one.

2.2. Relationships to other models

By simply making \( C = 1 \), it is obvious from (1) that any parametric density is a particular case of the mixture model. In particular, we obtain a Gaussian of mean \( \mu \) and covariance \( \Sigma \) when

\[ P(x|\omega_1) = \mathcal{G}(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{-\frac{1}{2}||x - \mu||^2_\Sigma}, \tag{2} \]

where

\[ ||x - \mu||^2_\Sigma = (x - \mu)^T \Sigma^{-1} (x - \mu). \tag{3} \]

To analyze the relationship with VQ, we start by noticing that any mixture model induces a soft partition of the feature space. In particular, given an observation \( x \), the a posteriori probability assignment of that observation to each of the feature classes is

\[
P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{\sum_{k=1}^{C} P(x|\omega_k)P(\omega_k)} \tag{4}
\]

\[ = \begin{cases} 
1 + \sum_{j \neq i} \frac{1}{P(x|\omega_j)P(\omega_j)}P(\omega_i), & \text{if } P(x|\omega_i)P(\omega_i) > 0 \\
0, & \text{otherwise.}
\end{cases}
\]

This leads to an explicit connection with VQ.

**Theorem 1** If \( x \) is a random vector distributed according to a Gaussian mixture

\[ P_\epsilon(x) = \sum_c P(\omega_c)\mathcal{G}(x, \mu_c, \Sigma_c(\epsilon)) \]

with covariance matrices

\[ \Sigma_c(\epsilon) = \epsilon \mathbf{I}, \forall c, \]

then

\[ \lim_{\epsilon \to 0} P_\epsilon(\omega_i|x) = \begin{cases} 
1, & \text{if } ||x - \mu_i||^2 \leq ||x - \mu_k||^2 \forall k \neq i \\
0, & \text{otherwise}
\end{cases} \tag{5} \]

and

\[ \lim_{\epsilon \to 0} P_\epsilon(x) = \sum_{i=1}^{C} \delta(x - \mu_i)P(\omega_i). \tag{6} \]

\(^1\)While these relationships are mostly known, we include them here because they play an important role in the new results to be introduced in subsequent sections.

\(^2\)For brevity, we make implicit the dependence on the image class, e.g., we write \( P_i(x|\omega_c) \) instead of \( P_i(x|\omega_{Y = i}) \) where \( Y = i \) is the true class label. In this section we also drop the subscripts, since all results are valid for any density.

\(^3\)We use the term “feature class” to describe the different feature sub-classes that may exist within each image class.
where \( \delta(x) \) is the Dirac delta function.\(^4\)

Equations (5) and (6) are a generative model for a VQ. In the VQ literature [2], (5) is known as the nearest neighbor condition (each point is assigned to the feature class associated with the nearest-neighbor codebook entry), and (6) as the centroid condition (this codebook entry is the mean of the cell associated with the feature class). The quantization operation consists of replacing each point by the codebook entry of the feature class to which it is assigned.

Since the histogram can be defined as

\[
P(x) = \sum_{i=1}^{C} \delta(x - c_i)P(\omega_i),
\]

where \( c_i \) are the centroids of the associated histogram cells (bins), the histogram is also a particular case of (6). The only difference with respect to VQ is that, in the histogram case, the cells are defined arbitrarily and not learned from a training sample. While, conceptually, this is a small difference, in practice it can lead to substantially different retrieval performance. Because histogram partitions are arbitrary, it makes small sense to have different ones for the different image classes. Hence, a universal partition (e.g. square cells of uniform size) is usually adopted by all classes. This solution, which we refer to as fixed partition or quantization, is clearly not ideal: for any given image class there will be many empty cells and a few strongly populated ones. A better approach, which is possible with VQ and mixture estimates, is to rely on an adaptive partition scheme where a different partition is learned for each class. This is, in the context of this work, the true distinction between a VQ and an histogram.

While VQ-based representations have been frequently used in the retrieval literature [4, 7, 8, 6], they typically rely on a fixed partition learned from a sample of all the image classes in database. This is only marginally different from using histograms and does not really address the major limitations of the histogram model. The problem is that it is not clear how to evaluate the distance between two VQs or histograms defined on different partitions of the feature space. It is this problem that we now address.

3. KL divergence between mixtures

The KL divergence between a query density \( P(x) \) and the density of the \( i \)th image class in the database \( P_i(x) \) (which we will call the “database density” for brevity) is defined as

\[
KL(P||P_i) = \int P(x) \log \frac{P(x)}{P_i(x)} dx.
\]

Since, in the retrieval context, we are only interested in finding the image class that minimizes this divergence and \( P(x) \) is independent of \( i \), it suffices to maximize the expected log of \( P_i(x) \) under \( P(x) \)

\[
i^* = \arg \max_i \int P(x) \log P_i(x) dx \equiv \arg \max_i EL(P||P_i).
\]

From now on we will refer to \( EL(P||P_i) \) as the retrieval similarity function. Unless otherwise noted, all densities are assumed to be mixtures of the form (1). Given no constraint on the feature class-conditional densities \( P(x|\omega_j) \) and \( P_i(x|\omega_j) \), it is only possible to derive a generic expression for the similarity function.

**Lemma 1** For a retrieval problem with query and database densities \( P(x) \) and \( P_i(x) \),

\[
EL(P||P_i) = \sum_{j,k} P(\omega_j) \int_{\chi_k} P(x|\omega_j) dx \log P_i(\omega_k)
\]

\[
+ \int_{\chi_k} P(x|\omega_j, \chi_k(x) = 1) \log \frac{P_i(\omega_k)}{P_i(\omega_j|x)} dx
\]

(10)

where

\[
\chi_k(x) = \begin{cases} 1, & \text{if } P_i(\omega_k|x) \geq P_i(\omega_l|x) \forall l \neq k \\ 0, & \text{otherwise.} \end{cases}
\]

(11)

\( \chi_k = \{ x : \chi_k(x) = 1 \} \) defines the partition of the feature space, and

\[
P(x|\omega_j, \chi_k(x) = 1) =
\]

\[
\left\{ \begin{array}{ll} \frac{P(x|\omega_j)}{\int_{\chi_k} P(x|\omega_j) dx}, & \text{if } x \in \chi_k \text{ and } \int_{\chi_k} P(x|\omega_j) dx > 0 \\ 0, & \text{otherwise.} \end{array} \right.
\]

Equation (10) reveals two fundamental components of similarity. The first

\[
\sum_{j,k} P(\omega_j) \log P_i(\omega_k) \int_{\chi_k} P(x|\omega_j) dx
\]

is a function of the feature class probabilities, the second

\[
\sum_{j,k} P(\omega_j) \int_{\chi_k} P(x|\omega_j) \log \frac{P_i(x|\omega_k)}{P_i(x|\omega_j)} dx
\]

is a function of the class-conditional densities. The overall similarity is strongly dependent on the partition \( \{ \chi_1, \ldots, \chi_C \} \) of the feature space determined by the database density \( P_i(x) \), the term

\[
\int_{\chi_k} P(x|\omega_j) dx
\]
weighting the contribution of each cell according to the fraction of the query probability that it contains. In particular, if \( S(\omega_j) \) is the support set of \( P(x|\omega_j) \), then
\[
\begin{align*}
\int_{\chi_k} P(x|\omega_j) dx &= 0, & & \text{if } S(\omega_j) \cap \chi_k = \emptyset \\
\int_{\chi_k} P(x|\omega_j) dx &= 1, & & \text{if } S(\omega_j) \subseteq \chi_k \\
\int_{\chi_k} P(x|\omega_j) dx &\in (0,1), & & \text{otherwise,}
\end{align*}
\]
and \( \int_{\chi_k} P(x|\omega_j) dx \) can be seen as a measure of overlap between \( P(x|\omega_j) \) and the cell \( \chi_k \) determined by \( P_i(x|\omega_k) \).

### 3.1. Histograms

When all image classes share the same feature class-conditional densities and the partition of the feature space is fixed, determining the similarity in closed-form is straightforward. This is the case of the histogram.

**Lemma 2** If all mixture densities define the same hard partition
\[
\chi_k(x) = \begin{cases} 1, & \text{if } P(\omega_i|x) = P_i(\omega_i|x) = \delta_{k,i} \forall i \\ 0, & \text{otherwise,} \end{cases} \tag{12}
\]
where \( \delta_{k,i} \) is the Kronecker delta function
\[
\delta_{k,l} = \begin{cases} 1, & \text{if } k = l, \\ 0, & \text{otherwise,} \end{cases} \tag{13}
\]
then
\[
EL(P||P_i) = \sum_j P(\omega_j) \log P(\omega_j) + \sum_j P(\omega_j) \int_{\chi_k} P(x|\omega_j) \log P_i(x|\omega_j) dx. \tag{14}
\]

Because, when all image classes share the same feature-class conditional densities, the second term of (14) does not depend on \( i \), this lemma implies that
\[
\arg \max_i EL(P||P_i) = \arg \max_j \sum_j P(\omega_j) \log P_i(\omega_j) = \arg \min_i \sum_j P(\omega_j) \log \frac{P(\omega_j)}{P_i(\omega_j)}
\]
and we obtain the standard expression for the KL divergence between histograms.

### 3.2. Gaussian mixtures

Lifting the restriction to a common hard partition, makes the computation of the KL divergence significantly more challenging. We next concentrate on this case, starting with a preliminary result.

**Lemma 3** For any probability density \( P(x) \), \( x \in \mathbb{R}^n \), \( \alpha \in \mathbb{R}^n \), symmetric positive definite matrix \( B \in \mathbb{R}^{n \times n} \), and set \( \chi \), if
\[
\int_{\chi} P(x) dx = 1,
\]
then
\[
\int_{\chi} P(x)||x - \alpha||_B^2 dx = \text{trace}[B^{-1}\hat{\Sigma}_x] + \|\hat{\mu}_x - \alpha\|_B^2,
\]
where
\[
\hat{\mu}_x = \int_{\chi} P(x)x dx
\]
and
\[
\hat{\Sigma}_x = \int_{\chi} P(x)(x - \hat{\mu}_x)(x - \hat{\mu}_x)^T dx.
\]

This lemma allows us to specialize (10) to Gaussian mixtures.

**Lemma 4** For a retrieval problem with query density \( P(x) \) given by (1) and Gaussian mixtures for the database densities \( P_i(x) \),
\[
P_i(x) = \sum_{k=1}^{C_i} G(x, \mu_{i,k}, \Sigma_{i,k}) P_i(\omega_k),
\]
where \( G(x, \mu, \Sigma) \) is as defined in (2),
\[
EL(P||P_i) = \sum_{j,k} P(\omega_j) \int_{\chi_k} P(x|\omega_j) dx \{ \log P_i(\omega_k) \\
+ \left[ \log G(\hat{\mu}_{q,j,k}, \mu_{i,k}, \Sigma_{i,k}) - \frac{1}{2} \text{trace}[\Sigma_{i,k}^{-1}\hat{\Sigma}_{q,j,k}] \right] \\
- \int_{\chi_k} P(x|\omega_j, \chi_k(x) = 1) \log P_i(\omega_k|x) dx \} \tag{16}
\]
where
\[
\hat{\mu}_{q,j,k} = \int_{\chi_k} P(x|\omega_j, \chi_k(x) = 1)x dx, \tag{17}
\]
\[
\hat{\Sigma}_{q,j,k} = \int_{\chi_k} P(x|\omega_j, \chi_k(x) = 1)(x - \hat{\mu}_{q,j,k})(x - \hat{\mu}_{q,j,k})^T dx, \tag{18}
\]
and \( \chi_k \) and \( P(x|\omega_j, \chi_k(x) = 1) \) are as defined in Lemma 1.

Equation (16) reveals that, for Gaussian mixtures, there are three components of the similarity function. Consider, without loss of generality, the query feature class \( w_j \) and the database feature class \( w_k \).

The first term in the equation is simply a measure of the similarity between the class probabilities \( P(\omega_j) \) and \( P_i(\omega_k) \) weighted by the measure of overlap \( \int_{\chi_k} P(x|\omega_j) dx \). This
term is a generalization of the one appearing in (15) that
defines the lack of alignment between the partitions defined by
the query and database densities.

The term in square brackets is, up to constants that do not
depend on \( i \), the KL divergence between the Gaussian
\( P_i(x|\omega_k) \) and a Gaussian with parameters \( \bar{\mu}_{q,j,k} \) and
\( \bar{\Sigma}_{q,j,k} \) [5]. From (17) and (18), these are simply the mean
and covariance of \( x \) according to \( P(x|\omega_j) \) given that \( x \in \chi_k \). Hence, the second term is simply a measure of the sim-
ilarity between the feature class conditional densities inside
the cell defined by \( P_i(x|\omega_k) \). Once again, this measure is
weighted by the amount of overlap between the two densities.

Finally, the third term weighs the different cells \( \chi_k \) ac-
cording to the ambiguity of their ownership. Recall that,\( \forall x \in \chi_k, P_i(\omega_k|x) > P_i(\omega_l|x) \), \( \forall l \neq k \). If \( P_i(\omega_k|x) = 1 \),
the cell is uniquely assigned to \( \omega_k \) and this term will be zero.
If, on the other hand, \( P_i(\omega_k|x) < 1 \), then the cell will also
be assigned to other classes and the overall similarity will
increase.

While providing insight on the different components of
similarity, (16) does not have a closed-form solution. There
is, however, one particular case where such a solution exits:
the case where all mixture models are vector quantizers.

### 3.3. Vector quantizers

Using Theorem 1, the VQ case can be analyzed by assum-
ing Gaussian feature class-conditional densities and in-
vestigating what happens when all covariance matrices tend
to zero. This leads to the following result.

**Lemma 5** For a retrieval problem with Gaussian mixtures
for the query and database densities

\[
P_i(x) = \sum_{j=1}^{C_i} g(x, \mu_{q,j,k}, \epsilon \Sigma_{q,j}) P(\omega_j)
\]

\[
P_i(x) = \sum_{j=1}^{C_i} g(x, \mu_{l,j,k}, \epsilon \Sigma_{l,j}) P_l(\omega_k)
\]

where \( g(x, \mu, \Sigma) \) is as defined in (2), when \( \epsilon \to 0 \)

\[
EL(P_i|P_l) = \sum_j P(\omega_j) \log P_i(\omega_{a(j)})
\]

\[
\quad + \lim_{\epsilon \to 0} \log (g(\bar{\mu}_{q,j,a(j)}, \epsilon \Sigma_{q,j,a(j)}))
\]

\[
\quad - \frac{1}{2\epsilon} \text{trace}(\Sigma_{q,j,a(j)}^{-1} \tilde{\Sigma}_{q,j,a(j)})
\]

where \( \chi_k \) is as defined in Lemma 1, \( \bar{\mu}_{q,j,a(j)} \) and
\( \bar{\Sigma}_{q,j,a(j)} \) as defined in Lemma 4, and

\[
\alpha(j) = k \iff \|\mu_{q,j} - \mu_{l,k}\|_{\Sigma_{l,k}}^2 < \|\mu_{q,j} - \mu_{l,l}\|_{\Sigma_{l,l}}^2, \forall l \neq k.
\]

We are now ready to derive a closed-form expression for
the similarity between VQ-based density estimates.

**Theorem 2** For a retrieval problem with VQ-based esti-
mates for the query and database densities

\[
P(x) = \sum_{j=1}^{C} \delta(x - \mu_{q,j})P(\omega_j)
\]

\[
P_l(x) = \sum_{k=1}^{C_l} \delta(x - \mu_{l,k})P_l(\omega_k),
\]

the KL similarity criteria reduces to

\[
\arg\max_i EL(P||P_i) =
\]

\[
\arg\min_i \lim_{\lambda \to \infty} \left\{ \sum_j P(\omega_j) \log \frac{P(\omega_j)}{P_l(\omega_{a(j)})} \right\} + \lambda \sum_j P(\omega_j) \|\mu_{q,j} - \mu_{l,a(j)}\|^2 \]

(19)

where

\[
\alpha(j) = k \iff \|\mu_{q,j} - \mu_{l,k}\|^2 < \|\mu_{q,j} - \mu_{l,l}\|^2, \forall l \neq k.
\]

The theorem states that, for VQ-based density estimates,
the minimization of the KL divergence is a constrained opti-
imization problem [1]. Given a query VQ and a database
VQ, one starts by vector quantizing the codewords of the
former according to the latter, i.e. each codeword of the
query VQ is assigned to the cell of the database VQ whose
centroid is closest to it. The best database VQ is the
one that minimizes a sum of two resulting terms: a term
that accounts for the average distortion of the quantization
(\( \sum_j P(\omega_j) \|\mu_{q,j} - \mu_{l,a(j)}\|^2 \)) and the KL divergence
between the feature-class probability distributions. \( \lambda \) is a La-
grange multiplier that weighs the contribution of the two
terms.

By taking the limit \( \lambda \to \infty \), all the emphasis is placed
on the average quantization distortion. This leads to two
distinct situations of practical interest. The first is when
the two quantizers share the same codewords. In this case,
the quantization distortion is null and the cost function be-
comes that of (15), i.e. the KL divergence between label
histograms. Since quantizers with equal codewords define
equal partitions of the feature space, this situation is equiva-
 lent to that of histograms and the result is, therefore, not
surprising.

The second is when the quantizers have different code-
words (and consequently define different partitions). In this
case, the quantization distortion becomes predominant and the retrieval criteria reduces to
\[
\arg \max_i \text{EL}(P || P_i) = \arg \min_i \sum_j P(\omega_j) || \mu_{q,j} - \mu_{i,\alpha(j)} ||^2.
\]
Notice that, even in this case, the complexity of the retrieval operation is only \(O(C^2 n)\), where \(n\) the dimension of the space. \(C^2 n\) is typically orders of magnitude smaller than the cardinality of the query set \(Q\) leading to significant savings over the direct evaluation of the query likelihood. Compared to the complexity of histogram-based techniques \(O(b^n)\), where \(b\) is the number of bins per axis, (19) trades off exponential growth in the number of dimensions for quadratic growth in the number of classes. Since \(C\) is usually small, this enables significantly more accurate estimates in high-dimensional spaces. Consider for example a space with \(n = 16\): the complexity of (19) with 64 mixture components (more than enough to accommodate the typical number of clusters in densities of practical interest) is equal to the complexity of HI with only two bins per axis (i.e. each feature quantized in a binary fashion). It is natural to expect that the coarse histogram estimates will lead to worse retrieval performance.

### 3.4. The asymptotic likelihood approximation

Vector quantization has particular interest not only because it leads to a closed-form similarity expression, but also because it provides insights on how to approximate (16) in the case of generic Gaussian mixtures. In particular, Lemma 5 suggests the following definition.

**Definition 1** Given a retrieval problem with Gaussian mixtures for the query and database densities

\[
P(x) = \sum_{j=1}^{C} G(x; \mu_{q,j}, \Sigma_{q,j}) P(\omega_j)
\]

\[
P_i(x) = \sum_{k=1}^{C_i} G(x; \mu_{i,k}, \Sigma_{i,k}) P_i(\omega_k),
\]

the asymptotic likelihood approximation (ALA) is defined by

\[
\text{ALA}(P || P_i) \equiv \sum_j P(\omega_j) \left\{ \log P_i(\omega_{\alpha(j)}) + \log G(\mu_{q,j}, \mu_{i,\alpha(j)}, \Sigma_{q,j}, \Sigma_{i,\alpha(j)}) - \frac{1}{2} \text{trace} \left[ \Sigma_{i,\alpha(j)}^{-1} \Sigma_{q,j} \right] \right\}, \tag{20}
\]

where \(\alpha(j)\) is as defined in Lemma 5.

Comparing (20) with (16) reveals that \(\text{ALA}(P || P_i)\) is a good approximation to \(\text{EL}(P || P_i)\) if the following two assumptions hold.

**Assumption 1** Each cell \(\chi_k\) of the partition determined by \(P_i(x)\) is assigned to one feature class with probability one, i.e.

\[
P_i(\omega_k | x) = 1, \ \forall x \in \chi_k.
\]

**Assumption 2** The support set of each feature class-conditional density of the query mixture is entirely contained in a single cell \(\chi_k\) of the partition determined by \(P_i(x)\). I.e.,

\[
\forall j \exists k : S(\omega_j) \subset \chi_k.
\]

Under Assumption 1, the third term of (16) vanishes. Under Assumption 2, \(\int_{\chi_k} P(x | \omega_j) = \delta_{k,\alpha(j)}, \hat{\mu}_{q,j,\alpha(j)} = \mu_{q,j}, \text{ and } \hat{\Sigma}_{q,j,\alpha(j)} = \Sigma_{q,j}\). Taken together, these equalities lead to the ALA. While both assumptions are valid in the VQ case, the ALA does not necessarily imply a VQ model. In particular, all feature class-conditional densities are allowed to have non-zero covariance matrices. However, Assumption 1 will only be reasonable if the feature class-conditional densities of \(P_i(x)\) have reduced overlap. This implies that the distance between each pair of \(\mu_{i,k}\) should be larger than the spread of the associated Gaussians. A 1-D illustration of this effect is provided by Figure 1, where we show two Gaussian class-conditional likelihood functions and the posterior probability function \(P_i(\omega_k | x)\) for class 0. As the separation between the Gaussians increases, the posterior probability changes more abruptly and the partition becomes harder.

**Figure 1.** Impact of the separation between two Gaussian densities on the partition that they define.

Assumption 2 never really holds for Gaussian mixtures, since Gaussians have infinite support. However, if Assumption 1 holds and, in addition, the spread of the Gaussians in \(P(x)\) is also much smaller than the size of the cells \(\chi_k\), then

\[
\int_{\chi_{\alpha(j)}} P(x | \omega_j) dx \approx 1
\]

with high probability.
In summary, the crucial assumption for the validity of the ALA is that the Gaussian feature class-conditional densities within each model have reduced overlap. The plausibility of this assumption grows with the dimension of the feature space, since high-dimensional spaces are more sparsely populated than low-dimensional ones. To illustrate this point, we performed the following experiment:

- a mixture model with 8 feature classes was fitted, using EM, to a training sample containing feature vectors from a random image in the Corel database;
- a 10,000 point sample was drawn from this mixture model;
- for each sample point $x_i$, $i = 1, \ldots, 10,000$, we evaluated the maximum posterior class-assignment probability $\max_k P(\omega_k|x_i)$;
- the maximum posterior probabilities were histogramed.

The experiment was repeated with various images. For each, several mixture models were obtained by projecting the mixture model of the original space ($n = 64$) into lower-dimensional subspaces. The results were similar across images, a typical example being presented in Figure 2, where we show the histograms of the maximum posterior probability for 2, 4, 8, 16, 32, and 64 subspaces. It is clear that, as the dimension of the space increases Assumption 1 becomes more realistic.

3.5. Experimental evaluation

The main contribution of this paper has been to show that probabilistic retrieval and the asymptotically equivalent KL divergence criteria can be computed exactly or closely approximated with computational cost equivalent to that of standard retrieval solutions. This contribution is only valuable if the probabilistic criteria outperforms the ones associated with these solutions. While theoretical evidence that this is true has been presented elsewhere [10], this evidence should be validated experimentally. In this section, we report the results of experiments on a dataset containing 1,500 images from 15 classes of the Corel database, comparing the performance of probabilistic retrieval against two popular histogram-based approaches: HI and color autocorrelograms [3].

For these experiments, we have used mixtures of 8 Gaussians on a 48-dimensional feature space based on the discrete-cosine transform (see [9, 11] for details), 512-bin color histograms and 2,048-bin color autocorrelograms (512-bin base histograms times 4 distances, see [3] for details). While, to be completely fair (exact same amount of computation), we should use 3,072 bins for the histogram techniques, preliminary experiments had shown no performance increase over the results obtained with the number of parameters above. In order to evaluate the goodness of the ALA, we evaluated the retrieval performance under both MALA (where we maximize the ALA) and the, much more expensive, true ML criteria.

The plot on Figure 3 presents the precision/recall curves for the four different retrieval solutions. It is clear that the performance of HI is not very good, and autocorrelograms
only improve performance by about 5%. Both approaches are significantly less effective than either ML or MALA, that show no significant differences in precision/recall. This confirms the argument that 1) ALA is a good approximation to the true likelihood, and 2) MALA is the best overall solution when one takes computational complexity into account.

We conclude by presenting some visual examples of the retrieval outcome. Figure 4 presents typical results for queries with horses, cars, and oil paintings. These results illustrate some of the nice properties of the probabilistic retrieval formulation: robustness to changes in object position and orientation, robustness against the presence of distracting objects in the background, and perceptually intuitive errors (in the painting example, two pictures of the sphinx - pyramids class - are returned after all the paintings of human figures).

References


Figure 3. Precision/recall on Corel for HI, color autocorrelogram (CAC), ML, and MALA.

Figure 4. Queries for horses, cars, and paintings.