Chapter 6

From local to global similarity

We have mentioned, in Chapter 2, that probabilistic retrieval establishes a common framework for local and global queries. In this chapter, we analyze the issue of local queries in greater detail. We start by discussing their importance and reviewing the most popular approaches for their implementation. We then show why probabilistic retrieval provides a natural solution to the problem and present experimental evidence of its robustness against incomplete queries.

The major limitation of the straightforward implementation of probabilistic retrieval is then shown to be the linear growth of retrieval complexity with the cardinality of the query. When the goal is to evaluate global similarity, this makes such an implementation much more expensive than standard methods, such as the histogram intersection or MRSAR techniques. We derive an alternative implementation which is competitive with these techniques from a complexity point of view, while achieving performance similar to that of the straightforward implementation.

At the core of this implementation is the evaluation of the KL divergence between two Gaussian mixtures. This is an interesting problem on its own, with applications that go well beyond the CBIR problem. We show that this divergence can be computed exactly when the mixture models are vector quantizers and introduce an asymptotic approximation for generic Gaussian mixtures. This approximation significantly reduces the computational complexity of the KL divergence without any significant impact on the resulting similarity judgments.

6.1 Local similarity

In addition to evaluating holistic similarity between images, a good retrieval architecture should also provide support for local queries, i.e. queries consisting of user-selected image regions. The ability to satisfy such queries is of paramount importance for two fundamental reasons. First, a retrieval architecture that supports local similarity will be much more tolerant to incomplete queries than an architecture that can only evaluate global similarity. In particular, it will be able to perform partial matches and therefore deal with events involving occlusion, object deformation, and changes of camera parameters. This is likely to improve retrieval accuracy even for global queries.

Figure 6.1: Example of a query image with multiple interpretations.

Second, local queries are much more revealing of the user's interests than global ones. Consider a retrieval system faced with the query image of Figure 6.1. Given the entire picture, the only possible inference is that the user may be looking for any combination of the objects in the scene (fireplace, bookshelves, painting on the wall, flower baskets, white table, sofas, carpet, rooms with light painted walls) and the query is too ambiguous.

By allowing the user to indicate the relevant regions of the image, the ambiguity can be significantly reduced.

6.1.1 Previous solutions

The standard solution for handling local queries is to rely on image segmentation and then perform retrieval on the individual segments, i.e. evaluate the similarity of each query region against all the regions extracted from the images in the database. This approach suffers from two fundamental problems: 1) segmentation is a difficult problem, and 2) there is a combinatorial explosion of the number of similarity evaluations to be performed.

Despite the difficulty of automatic image segmentation, several retrieval systems have relied on it for determining image regions [5, 7, 65, 164, 165]. While, theoretically, precise image segmentation enables shape-based retrieval, in practice it is not uncommon for a segmentation algorithm to break a single object into several regions or unify various objects into one region, making shape-based similarity close to hopeless. Hence, even when automated segmentation is used, shape representations tend to be very crude. Therefore, it is not clear that precise segmentation is an advantage for region-based queries. In fact, the use of sophisticated segmentation can be more harmful than beneficial: for example, in the context of "blob-world", Howe [65] reports significant improvements by replacing the sophisticated segmentation algorithm used by Belongie et al. [7] with a much simpler variation.

The only clear exceptions to this observation seem to be applications where it is possible to manually pre-segment all the imagery because 1) there is an economic incentive to do this, and 2) it is very clear what portions of each database image will be relevant to the queries posed to the retrieval system. An example of such application domain is that of medical imaging, in particular what concerns to lesion diagnostics [160]. On the contrary, for generic databases there is usually too much imagery to allow manual processing and it is rarely known what specific objects may be of interest to the users of such databases.

Since precise segmentations are difficult, several authors have adopted the simplifying view of relying on arbitrary image partitions to obtain local information [5, 110, 111, 141, 153, 166, 175]. While this solves the problem of segmentation complexity without noticeable degradation of performance (in fact it does not even seem clear at this point that segmentation works better than arbitrary image partitioning), it still does not address the second problem, i.e. the combinatorial explosion associated with matching all image segments.

In order to overcome this difficulty, several mechanisms have been proposed in the literature. The simplest among these is to make the individual regions large enough and their feature representation compact enough so that each image can still be represented by a simple feature vector (concatenation of the individual region features) of manageable dimensions [166, 175]. Such approaches are of limited use for local queries since 1) several objects or visual concepts may fall on a single image region, 2) feature representations are not expressive enough to finely characterize each region, and 3) it is hard to guarantee invariance to image transformations when dealing with regions of large spatial support.

An alternative view is not to worry with compactness and simply deal with the combinatorics of region-based retrieval at the level of traditional database indexing [110, 111, 141, 164]. Minka and Picard [110] propose clustering of the individual image regions as a database organization step that significantly reduces query time (since query regions are matched against cluster representatives instead of all the members). The use of clustering as an indexing tool has the major disadvantage that the entire database must be re-clustered (an expensive operation) when images are included in or deleted from the database.

An alternative to clustering, proposed by Ravela et al. [141] and Smith and Chang [164], is to rely on indexing mechanisms derived from those traditionally used with text databases. The idea is to consider all the dimensions of the feature space independent, create one dimensional indices (which can be searched quickly) for each of them, and then use standard database operations, such as joins, during retrieval. The main problem with these approaches is that, for the high-dimensional spaces required for meaningful image characterization, the indexing savings vanish as the database grows. The problem is, therefore, particularly acute for databases of image regions.

In summary, the downside of approaches based on indexing is that region databases complicate the indexing problem by orders of magnitude. Since, at this point, indexing is still an open question (even for the simpler case of non-region based representations) this can be a significant hurdle.

6.1.2 Probabilistic retrieval

One of the main attractives of probabilistic retrieval is that, conceptually, it makes local queries straightforward. Recall, from Chapter 2, that for a query composed by a collection of N vectors $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the Bayesian retrieval criteria is

$$g^*(\mathbf{x}) = \arg\max_i \sum_{j=1}^{N} \log P_{\mathbf{X}|Y}(\mathbf{x}_j|i) + \log P_Y(i).$$

Notice that there is, under this criteria, no constraint for the query set to have the same cardinality as the set used to estimate the class-conditional densities $P_{\mathbf{X}|Y}(\mathbf{x}_j|i)$. In fact, it is completely irrelevant if $\mathbf{x}$ consists of one query vector, a collection of vectors extracted from a region of a query image, or all the vectors that compose that query image. Hence, there is no difference between local and global queries.

The ability of Bayesian similarity to explain individually each of the vectors that compose the query is a major advantage over criteria based on measures of similarity between entire densities, such as L^p norms or the KL divergence, for two fundamental reasons. First, it enables local similarity without requiring explicit segmentation of the images in the database. The only segmentation information that is required are the image regions which make up the query and which are provided by the user himself. The indexing complexity is therefore not increased. Second, since probabilistic retrieval relies on a generative model (a probability density) that is compact independently of the number of elemental regions that compose each image, these can be made as small as desired, all the way down to the single pixel size. Our choice of local 8×8 neighborhoods is motivated by concerns that are not driven by the feasibility of the representation per se, but rather by the desire to achieve a good trade-off between invariance, the ability to model local image dependencies, and the ability to allow users to include regions of almost arbitrary size and shape in their queries.

6.1.3 Experimental evaluation

We are now ready to evaluate the accuracy of Bayesian retrieval with region-based queries. For this, we start by replicating the experiments of section 5.4.1 but now considering incomplete queries, i.e. queries consisting only of a subset of the query image. All parameters were set to the values that were previously used to evaluate global similarity and a series of experiments conducted for query sets of different cardinalities. From a total of 256 non-overlapping blocks, the number of vectors contained in the query varied from 1 (0.3% of the image) to 256 (100%)[1]. Blocks were selected starting from the center in an outward spiral fashion.

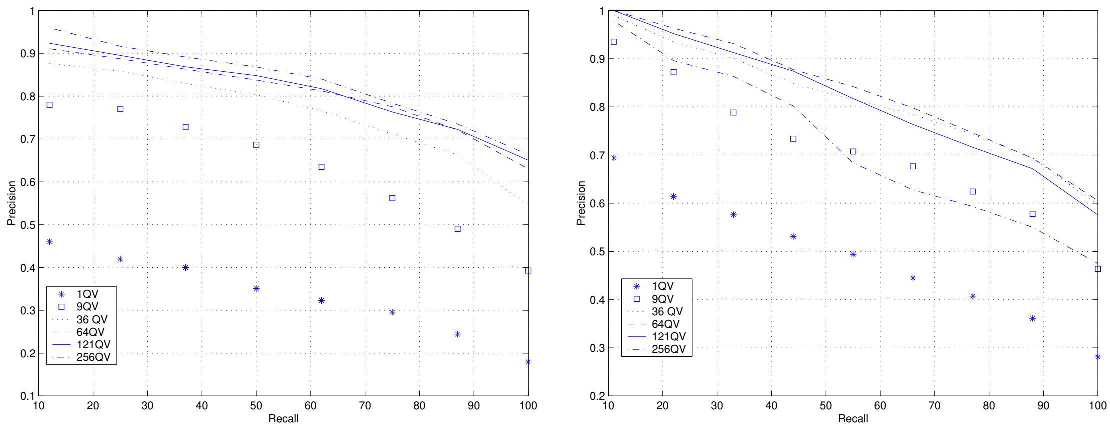

Figure 6.2: Precision/recall curves of EMM/ML on Brodatz (left) and Columbia (right). X QV means that only X feature vectors from the query image were actually included in query.

Figure 6.2 presents precision/recall curves for these experiments. The figure clearly shows that it only takes a small subset of the query feature vectors to achieve retrieval performance identical to the best possible. In both cases, 64 query vectors, 0.4% of the total number that could be extracted from the image and covering only 25% of its area, are enough. In fact, for Columbia, performance is significantly worse when all 256 vectors are considered than when only 64 are used. This is due to the fact that, in Columbia, all objects appear over a common black background that can cover a substantial amount of the

[1]Notice that even 256 vectors are a very small percentage (1.5%) of the total number of blocks that could be extracted from the query image if overlapping blocks were allowed.

image area. As Figure 6.3 illustrates, when there are large variations in scale among the different views of the object used as query, the consequent large differences in uncovered background can lead to retrieval errors. In particular, images of objects in a pose similar to that of the query are preferred to images of the query object in very different poses.

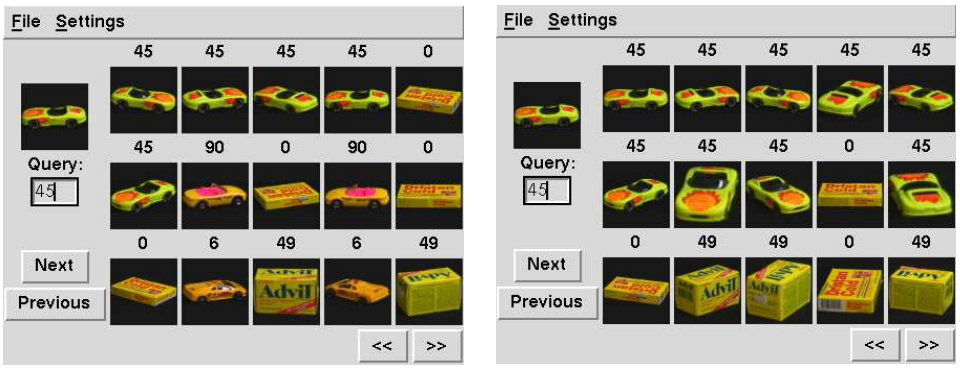

Figure 6.3: Global similarity (left) can lead to worse precision/recall than local similarity (right) on Columbia due to the large black background common to all objects.

Notice that these are two natural interpretations of similarity (prefer objects similar to the query and presented in the same pose vs. prefer the query object in different poses) and Bayesian retrieval seems to oscillate between the two. Under global similarity, the more generic interpretation (pictures of box-shaped objects in a particular orientation) is favored. When the attention of the retrieval system is focused explicitly on the query object (local query), this object becomes preferred independently of its pose. Obviously, precision/recall cannot account for these types of subtleties and the former interpretation is heavily penalized. In any case, these experiments show that, on databases like Brodatz and Columbia, Bayesian retrieval is very robust against missing data and can therefore handle local queries very easily.

A more challenging situation is that in which all images are composed by multiple visual stimulae, e.g. the mosaic databases presented in the appendix where each image is a mosaic of four Columbia or Brodatz images. The goal here is to, given a query image containing only one texture or object, to find all the images in the mosaic database that contain that query image. Figure 6.4 presents precision/recall curves for this case. Once again, retrieval is performed for query sets of various cardinalities. For comparison, we also show the performance based on global similarity, i.e. where one image of the mosaic database containing the texture or object of interest is used as query, and all feature vectors are considered in the retrieval operation.

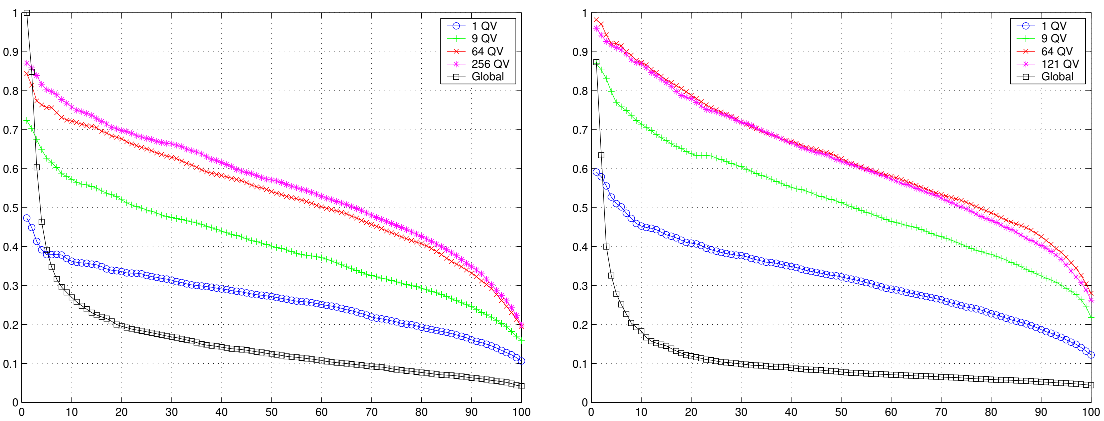

Figure 6.4: Precision/recall curves of EMM/ML on Brodatz mosaics (left) and Columbia mosaics (right). X QV means that only X feature vectors from the query image were actually included in query. For comparison, the curves obtained with global similarity are also shown.

The figure clearly shows that 1) retrieval based on local queries is significantly better than that based on global similarity, and 2) a small sample of the texture or object of interest (64 query vectors covering 25% of its area and containing 0.4% of the total number of vectors that could be extracted from it) is sufficient to achieve performance similar to the best. These results confirm the argument that Bayesian retrieval leads to effective region-based queries even for imagery composed by multiple visual stimulae.

The overwhelming superiority of local over global queries is explained by Figure 6.5, where we present the results of the two types of query for a particular object (yellow onion). When global similarity is employed, the retrieval system returns mosaics that have objects in common with the query with high probability. While this may be satisfactory in certain contexts, typically the user is interested in only one of the objects. There is however no way for the retrieval system to know this from the query alone. By selecting the region of interest, the user reduces the ambiguity of the query, enabling significantly higher retrieval precision.

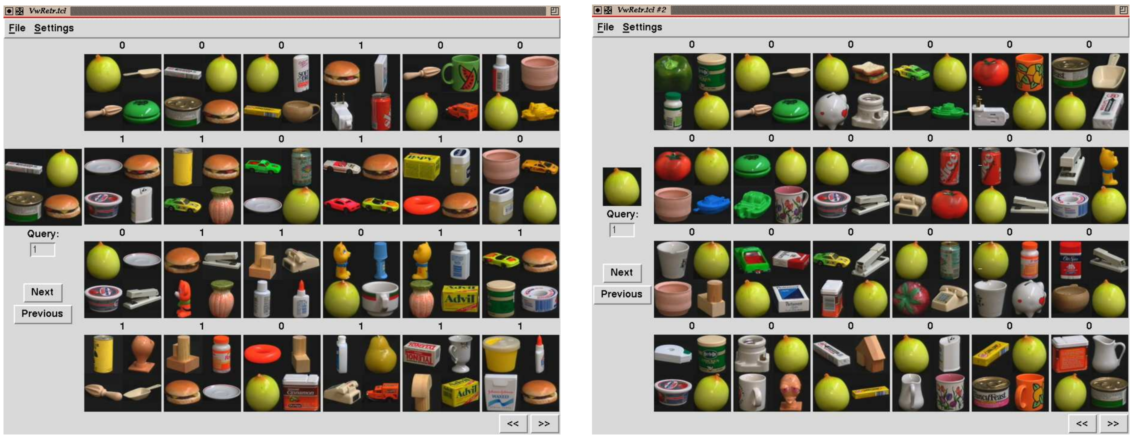

Figure 6.5: Examples of retrieval based on global (left) and local (right) similarity. In this case, the user is looking for images containing yellow onions. The number on top of each retrieved image is a flag indicating retrieval errors.

6.2 The complexity of global similarity

The results above show that, in addition to capturing both color and texture, the combination of EMM models with Bayesian retrieval is an elegant solution to the problem of local similarity. There is, however, one aspect in which this retrieval architecture is still not competitive with standard solutions like MRSAR/MD and HI: the computational cost of global similarity. We now investigate solutions to this problem.

6.2.1 Computational cost

The main limitation of (2.16) is that, because it evaluates the relevance of each query vector individually, its complexity is linear in the cardinality of the query. While this is not a significant problem for local queries since these consider only a small subset of the query image, it may become a major problem for the evaluation of global similarity, where all vectors must be taken into account. Ideally, one would like the complexity of global Bayesian similarity to be equivalent to that of standard approaches like MRSAR/MD and HI.

Table 6.1 presents a comparison of the computational complexity of the various ap-

Representation	Similarity function	Expression	Complexity
histogram	L^p norms	$\left(\sum_{r=1}^{R}\left\|\frac{q_r}{Q}-\frac{p_r^i}{P^i}\right\|^p\right)^{\frac{1}{p}}$	$O(R)=O(k^n)$
Gaussian	Mahalanobis	$(\hat{\mathbf{x}}-\mu_i)^T\mathbf{\Sigma}_i^{-1}(\hat{\mathbf{x}}-\mu_i)^T$	$O(n^2)$
Gaussian	ML or MDI	$\log\|\mathbf{\Sigma}_i\|+trace[\mathbf{\Sigma}_i^{-1}\hat{\mathbf{\Sigma}}_{\mathbf{x}}]$ $+(\hat{\mathbf{x}}-\mu_i)^T\mathbf{\Sigma}_i^{-1}(\hat{\mathbf{x}}-\mu_i)^T$	$O(n^2)$
EMM	ML	$\sum_{i=1}^{N}\log\sum_{c=1}^{C}\pi_c\mathcal{G}(\mathbf{x}_i,\mu_c,\mathbf{\Sigma}_c)$	$O(NCn^2)$

Table 6.1: The complexity of various retrieval solutions. See Chapters 2 and 5 for the meaning of all the symbols in the expressions of the third column. On the fourth column, n is the dimension of the feature space, k the number of cells per coordinate axis of the histogram, N the number of feature vectors on the Bayesian query, and C the number of classes of the EMM.

proaches discussed so far. While the complexity of the histogram is exponential in the dimension n of the feature space, the complexity of the Gaussian model is only quadratic. This is substantially less than the linear complexity of EMM/ML on the product of the cardinality of the query with the number of classes in the EMM. In practice, a few tricks can be used to reduce this gap.

6.2.2 Practical ways to reduce complexity

It is well known from speech research that, because the Gaussian components act together to model the overall density, full covariance matrices are usually not required by a mixture model even when the features are not independent [145, 169, 106, 107]. In fact, a combination of Gaussians of diagonal covariance can model correlations between the elements of the feature vector. Since the DCT coefficients are already approximately uncorrelated [29, 74], this is particularly true in the case of EMMs.

The use of diagonal covariances has two important consequences: 1) it reduces complexity from $O(NCn^2)$ to $O(NCn)$, and 2) significantly reduces the number of parameters to be estimated (and consequently the sample sizes required for estimation) and inherent complexity of learning the database models [107]. We rely on diagonal covariances in our implementation and all the results presented in the thesis were obtained in this way. Notice that using full covariance matrices is significantly more important under the rigid single-Gaussian model, where the diagonal covariance approximation can lead to a significant loss in performance [107, 145].

The embedded nature of the representation also makes it suitable for the implementation of filtering strategies, similar to those proposed in [60, 24], to minimize the computational requirements of retrieval. These strategies start by finding the K_1 best matches considering only the first DCT coefficient. Next, among these matches, the K_2 best matches are found using the first two coefficients. The search can continue in this way until the best K_n matches are obtained at full resolution. The average complexity of the similarity evaluation will then be $O(NC(1 + 2K_1/S + 3K_2/S + \ldots + nK_{n-1}/S))$, where S is the database size. If $K_i << S$, this complexity will only be marginally larger than $O(NC)$. Since these type of strategies can also be used for histograms and the standard Gaussian model, we do not investigate this issue any further. Instead, our goal is to derive a global similarity function of complexity competitive with that of the standard retrieval approaches.

For this, it is useful to compare the cost of the different solutions for typical values of the parameters involved. For color histograming, the standard approach is to quantize the luminance axis into 8 bins and the two color components into 16 each [172]. This leads to a total of $R = 2,048$ cells. With respect to texture, the dimension of the feature space depends strongly on the particular feature transformation employed. We use the 15 dimensional vectors of the MRSAR transformation as a benchmark. In both cases, complexity is significantly smaller than that of the direct implementation of Bayesian retrieval with EMMs.

In addition to using diagonal covariances, this complexity can be reduced by considering only a small number of feature classes (between 8 and 16) and embedded subspaces (typically 16) in the EMMs. We saw in the previous chapter that this restriction does not hurt the retrieval performance in any way. Nevertheless, the complexity per feature vector of EMM/ML ($128 < Cn < 256$) is still equivalent to the total complexity of MRSAR ($d^2 = 256$) and only a few orders of magnitude smaller than the total complexity of the histogram methods (2,048). This means that if $Cost_{hist/L^p}$, $Cost_{mrsar/md}$, and $Cost_{emm/ml}$ are, respectively, the total complexities for histogram/L^p norms, MRSAR/MD and EMM/ML, then

$$\frac{N}{2}Cost_{mrsar/md} < Cost_{emm/ml} < NCost_{mrsar/md}$$

and

$$Cost_{emm/ml} \propto \frac{N}{8}Cost_{hist/L^p}.$$

Since the number of feature vectors N can be as large as the number of pixels in the query image, this is very problematic.

6.2.3 Asymptotic approximations for global similarity

In section 2.3.3, we saw that, as the cardinality of the query grows, Bayesian retrieval converges asymptotically to the MDI criteria, i.e. the minimization of the KL divergence between the densities of the query and retrieved image. This suggests an alternative to (2.16) for the evaluation of global similarity: start by estimating the density of the query and then evaluate the distance between that density and those in the database. If the density estimates are based on a compact feature representation, this procedure will have much smaller complexity than the direct application of (2.16). The main problem with this strategy is that there is no easy way to evaluate the KL divergence between mixtures. We next investigate when this is possible and devise approximations for when it is not.

Start by recalling (see section 2.3.3) that, when the cardinality of the query is large, the Bayesian criteria (2.16) converges to

$$g(\mathbf{x}) = \arg\max_i \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}|Y}(\mathbf{x}|i) d\mathbf{x}. \tag{6.1}$$

To simplify the notation, we adopt the following conventions in the remainder of this chapter

$$P_{\mathbf{X}}(\mathbf{x}) = P(\mathbf{x}) = \sum_{j=1}^{C} P(\mathbf{x}|\omega_j) P(\omega_j) \tag{6.2}$$

and

$$P_{\mathbf{X}|Y}(\mathbf{x}|i) = P_i(\mathbf{x}) = \sum_{j=1}^{C_i} P_i(\mathbf{x}|\omega_j) P_i(\omega_j) \tag{6.3}$$

where the subscript i refers to the image class and the ω_j to the feature classes within each image class.

Given no constraints on the feature class-conditional densities $P(\mathbf{x}|\omega_j)$ and $P_i(\mathbf{x}|\omega_j)$, it is only possible to derive a generic expression for global similarity.

Lemma 2 *For a retrieval problem with the query and database densities of (6.2) and (6.3),*

$$\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \tag{6.4}$$

$$= \sum_{j,k} P(\omega_j) \left[\log P_i(\omega_k) + \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log \frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})} d\mathbf{x} \right] \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}$$

where

$$\chi_k(\mathbf{x}) = \begin{cases} 1, & \text{if } P_i(\omega_k|\mathbf{x}) \geq P_i(\omega_l|\mathbf{x}) \, \forall l \neq k \\ 0, & \text{otherwise,} \end{cases} \tag{6.5}$$

$\chi_k = \{\mathbf{x} : \chi_k(\mathbf{x}) = 1\}$, *and*

$$P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) = \begin{cases} \frac{P(\mathbf{x}|\omega_j)}{\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}}, & \text{if } \mathbf{x} \in \chi_k \text{ and } \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Proof: From (6.2) and (6.3),

$$\begin{aligned}\int P(\mathbf{x})\log P_i(\mathbf{x})d\mathbf{x} &= \sum_j P(\omega_j)\int P(\mathbf{x}|\omega_j)\log\sum_l P_i(\mathbf{x}|\omega_l)P_i(\omega_l)d\mathbf{x}\\ &= \sum_j P(\omega_j)\sum_k\int_{\chi_k} P(\mathbf{x}|\omega_j)\log\sum_l P_i(\mathbf{x}|\omega_l)P_i(\omega_l)d\mathbf{x}.\end{aligned}$$

Using Bayes rule

$$P_i(\omega_k|\mathbf{x}) = \frac{P_i(\mathbf{x}|\omega_k)P_i(\omega_k)}{\sum_l P_i(\mathbf{x}|\omega_l)P_i(\omega_l)}, \tag{6.6}$$

we have, $\forall k$ such that $P_i(\omega_k|\mathbf{x}) \neq 0$,

$$\sum_l P_i(\mathbf{x}|\omega_l)P_i(\omega_l)d\mathbf{x} = \frac{P_i(\mathbf{x}|\omega_k)P_i(\omega_k)}{P_i(\omega_k|\mathbf{x})}.$$

Since $\sum_k P_i(\omega_k|\mathbf{x}) = 1$, from the definition of χ_k we obtain $P_i(\omega_k|\mathbf{x}) > 0$, $\forall \mathbf{x} \in \chi_k$, and

$$\begin{aligned}&\int P(\mathbf{x})\log P_i(\mathbf{x})d\mathbf{x} =\\ &= \sum_j P(\omega_j)\sum_k\int_{\chi_k} P(\mathbf{x}|\omega_j)\log\frac{P_i(\mathbf{x}|\omega_k)P_i(\omega_k)}{P_i(\omega_k|\mathbf{x})}d\mathbf{x}\\ &= \sum_j P(\omega_j)\sum_k\left[\log P_i(\omega_k)\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} + \int_{\chi_k} P(\mathbf{x}|\omega_j)\log\frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})}d\mathbf{x}\right]\\ &= \sum_j P(\omega_j)\sum_k\left[\log P_i(\omega_k) + \int_{\chi_k} P(\mathbf{x}|\omega_j,\chi_k(\mathbf{x})=1)\log\frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})}d\mathbf{x}\right]\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x}.\ \square\end{aligned}$$

Equation (6.4) reveals that there are two fundamental components to global similarity. The first

$$\sum_{j,k} P(\omega_j)\log P_i(\omega_k)\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x}$$

is a function of the feature class probabilities, the second

$$\sum_{j,k} P(\omega_j)\int_{\chi_k} P(\mathbf{x}|\omega_j,\chi_k(\mathbf{x})=1)\log\frac{P_i(\mathbf{x}|\omega_k)}{P_i(\omega_k|\mathbf{x})}d\mathbf{x}\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x}$$

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

$$\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x}$$

weighting the contribution of each cell according to the fraction of the query probability that it contains. In particular, if $\mathcal{S}(\omega_j)$ is the support set of $P(\mathbf{x}|\omega_j)$, then

$$\begin{array}{ll}\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} = 0, & \text{if } \mathcal{S}(\omega_j)\cap\chi_k = \emptyset\\ \int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} = 1, & \text{if } \mathcal{S}(\omega_j)\subset\chi_k\\ \int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x} \in (0,1), & \text{otherwise,}\end{array}$$

and $\int_{\chi_k} P(\mathbf{x}|\omega_j)d\mathbf{x}$ can be seen as a measure of overlap between $P(\mathbf{x}|\omega_j)$ and the cell χ_k determined by $P_i(\mathbf{x}|\omega_k)$.

Histograms

When all image classes share the same feature class-conditional densities and the feature space is divided into a collection of disjoint cells, the evaluation of global similarity is straightforward. This is the case of the standard histogram model and VQ label histograms for a fixed quantization of the feature space.

Lemma 3 *If all mixture densities define the same hard partition*

$$\chi_k(\mathbf{x}) = \begin{cases} 1, & \textit{if } P(\omega_l|\mathbf{x}) = P_i(\omega_l|\mathbf{x}) = \delta_{k,l} \, \forall i \\ 0, & \textit{otherwise}, \end{cases} \tag{6.7}$$

where $\delta_{k,l}$ is the Kronecker delta function (2.3), then

$$\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \sum_j P(\omega_j) \log P_i(\omega_j) + \sum_j P(\omega_j) \int_{\chi_j} P(\mathbf{x}|\omega_j) \log P_i(\mathbf{x}|\omega_j) d\mathbf{x}. \tag{6.8}$$

Proof: Using the same argument as in the proof of Theorem 5, we assume without loss of generality that all the classes in all mixture models have non-zero probability, i.e.

$$P(\omega_l) > 0 \text{ and } P_i(\omega_l) > 0, \ \forall l, i.$$

It follows from (6.6) that $P(\omega_k|\mathbf{x}) = 1$ if and only if

$$\sum_{l \neq k} P_i(\mathbf{x}|\omega_l) P_i(\omega_l) = 0.$$

Since all the terms in the summation are non-negative, this implies

$$P_i(\mathbf{x}|\omega_l) = 0 \ \forall l \neq k.$$

I.e., for a hard partition such as (6.7), the support sets of $P(\mathbf{x}|\omega_k)$ and $P_i(\mathbf{x}|\omega_k)$ are contained in χ_k. Hence

$$\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} = \delta_{k,j}$$

and, since $P_i(\omega_k|\mathbf{x}) = 1, \forall \mathbf{x} \in \chi_k$, (6.8) follows from Lemma 2. $\square$

Because when all image classes share the same feature-class conditional densities, the second term of (6.8) does not depend on i, this lemma implies that

$$\begin{aligned}\arg\max_i \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} &= \arg\max_i \sum_j P(\omega_j) \log P_i(\omega_j) \\ &= \arg\min_i \sum_j P(\omega_j) \log \frac{P(\omega_j)}{P_i(\omega_j)}.\end{aligned} \tag{6.9}$$

Hence, if the feature space is first vector quantized and all image classes represented by label histograms, Bayesian retrieval is equivalent to minimizing the KL divergence between those label histograms. This is an interesting result from the computational point of view, since the complexity of this operation is $O(C)$, where C is the number of VQ cells, as opposed to the $O(NCn^2)$ complexity inherent to the straightforward application of the Bayesian criteria.

Gaussian mixtures

A much more challenging case occurs when we lift the restrictions of a common hard partition and consider generic Gaussian mixtures. We now concentrate in this case, starting with a preliminary result.

Lemma 4 *For any probability density $P(\mathbf{x})$, $\mathbf{x} \in R^n$, $\alpha \in R^n$, $B \in R^{n \times n}$ and set χ, if*

$$\int_\chi P(\mathbf{x}) d\mathbf{x} = 1,$$

then

$$\int_\chi P(\mathbf{x})(\mathbf{x}-\alpha)^T B(\mathbf{x}-\alpha) d\mathbf{x} = trace[B\hat{\Sigma}_\mathbf{x}] + (\hat{\mu}_\mathbf{x} - \alpha)^T B(\hat{\mu}_\mathbf{x} - \alpha),$$

where

$$\hat{\mu}_\mathbf{x} = \int_\chi P(\mathbf{x})\mathbf{x}\, d\mathbf{x}$$

$$\hat{\Sigma}_\mathbf{x} = \int_\chi P(\mathbf{x})(\mathbf{x} - \hat{\mu}_\mathbf{x})(\mathbf{x} - \hat{\mu}_\mathbf{x})^T d\mathbf{x}.$$

Proof:

$$\int_\chi P(\mathbf{x})(\mathbf{x}-\alpha)^T B(\mathbf{x}-\alpha) d\mathbf{x} =$$

$$
\begin{aligned}
&= \int_{\chi} P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}} + \hat{\mu}_{\mathbf{x}} - \alpha)^T B(\mathbf{x} - \hat{\mu}_{\mathbf{x}} + \hat{\mu}_{\mathbf{x}} - \alpha) d\mathbf{x} \\
&= \int_{\chi} P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})^T B(\mathbf{x} - \hat{\mu}_{\mathbf{x}}) d\mathbf{x} + 2 \int_{\chi} P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})^T B(\hat{\mu}_{\mathbf{x}} - \alpha) d\mathbf{x} + (\hat{\mu}_{\mathbf{x}} - \alpha)^T B(\hat{\mu}_{\mathbf{x}} - \alpha) \\
&= trace\left[B \int_{\chi} P(\mathbf{x})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})(\mathbf{x} - \hat{\mu}_{\mathbf{x}})^T d\mathbf{x}\right] + 2(\hat{\mu}_{\mathbf{x}} - \hat{\mu}_{\mathbf{x}})^T B(\hat{\mu}_{\mathbf{x}} - \alpha) + (\hat{\mu}_{\mathbf{x}} - \alpha)^T B(\hat{\mu}_{\mathbf{x}} - \alpha) \\
&= trace[B\hat{\Sigma}_{\mathbf{x}}] + (\hat{\mu}_{\mathbf{x}} - \alpha)^T B(\hat{\mu}_{\mathbf{x}} - \alpha). \square
\end{aligned}
$$

This lemma allows us to specialize (6.4) to Gaussian mixtures.

Lemma 5 *For a retrieval problem with the query densities of (6.2) and Gaussian mixtures for the database densities (6.3),*

$$
P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \mathcal{G}(\mathbf{x}, \mu_{i,k}, \mathbf{\Sigma}_{i,k}) P_i(\omega_k),
$$

where $\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})$ *is as defined in (2.5),*

$$
\begin{aligned}
&\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \\
&\quad = \sum_{j,k} P(\omega_j) \log P_i(\omega_k) \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} + \\
&\quad + \sum_{j,k} P(\omega_j) \left[\log \mathcal{G}(\hat{\mu}_{q,j,k}, \mu_{i,k}, \mathbf{\Sigma}_{i,k}) - \frac{1}{2} trace[\mathbf{\Sigma}_{i,k}^{-1} \hat{\Sigma}_{q,j,k}]\right] \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} \\
&\quad - \sum_{j,k} P(\omega_j) \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log P_i(\omega_k|\mathbf{x}) d\mathbf{x} \int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}
\end{aligned}
\tag{6.10}
$$

where

$$
\hat{\mu}_{q,j,k} = \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \mathbf{x} \, d\mathbf{x}, \tag{6.11}
$$

$$
\hat{\Sigma}_{q,j,k} = \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1)(\mathbf{x} - \hat{\mu}_{q,j,k})(\mathbf{x} - \hat{\mu}_{q,j,k})^T d\mathbf{x}, \tag{6.12}
$$

and χ_k *and* $P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1)$ *are as defined in Lemma 2.*

Proof: Since $P_i(\mathbf{x}|\omega_k) = \mathcal{G}(\mathbf{x}, \mu_{i,k}, \mathbf{\Sigma}_{i,k})$ and $\int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) d\mathbf{x} = 1$, simple application of the previous lemma results in

$$
\begin{aligned}
&\int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) \log P_i(\mathbf{x}|\omega_k) d\mathbf{x} = \\
&\quad = \log \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_{i,k}|}} \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1) d\mathbf{x} \\
&\quad\quad -\frac{1}{2} \int_{\chi_k} P(\mathbf{x}|\omega_j, \chi_k(\mathbf{x}) = 1)(\mathbf{x} - \mu_{i,k})^T \mathbf{\Sigma}_{i,k}^{-1} (\mathbf{x} - \mu_{i,k})^T d\mathbf{x}
\end{aligned}
$$

$$
\begin{aligned}
&= \log \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_{i,k}|}} - \frac{1}{2} trace[\mathbf{\Sigma}_{i,k}^{-1} \hat{\Sigma}_{q,j,k}] - \frac{1}{2} (\hat{\mu}_{q,j,k} - \mu_{i,k})^T \mathbf{\Sigma}_{i,k}^{-1} (\hat{\mu}_{q,j,k} - \mu_{i,k}) \\
&= \log \mathcal{G}(\hat{\mu}_{q,j,k}, \mu_{i,k}, \mathbf{\Sigma}_{i,k}) - \frac{1}{2} trace[\mathbf{\Sigma}_{i,k}^{-1} \hat{\Sigma}_{q,j,k}].
\end{aligned}
$$

The lemma follows by simple algebraic manipulation of (6.4). □

It is interesting to analyze each of the terms in (6.10). Consider 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 measure of overlap $\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x}$. This term is equivalent to that appearing in (6.9) but accounts for the fact that the partitions defined by the query and image class densities are now not aligned.

Comparing the term in square brackets with (2.23), it is straightforward to show that this term is equivalent to the similarity, under Bayesian retrieval, between the Gaussian $P_i(\mathbf{x}|\omega_k)$ and a Gaussian with parameters $\hat{\mu}_{q,j,k}$ and $\hat{\mathbf{\Sigma}}_{q,j,k}$. From (6.11) and (6.12), these are simply the mean and covariance of $\mathbf{x}$ according to $P(\mathbf{x}|\omega_j)$ given that $\mathbf{x} \in \chi_k$. Hence, the second term is simply a measure of the similarity between the feature class conditional densities inside the cell defined by $P_i(\mathbf{x}|\omega_k)$. Once again, this measure is weighted by the amount of overlap between the two densities.

Finally, the third term weights the different cells χ_k according to the ambiguity of their ownership. Recall that, $\forall \mathbf{x} \in \chi_k$, $P_i(\omega_k|\mathbf{x}) > P_i(\omega_l|\mathbf{x})$, $\forall l \neq k$. If $P_i(\omega_k|\mathbf{x}) = 1$, the cell is uniquely assigned to ω_k and this term will be zero. If, on the other hand, $P_i(\omega_k|\mathbf{x}) < 1$, then the cell will also be assigned to other classes and the overall likelihood will increase.

While providing insight on the different factors involved in global similarity, (6.10) is not very useful from a computational standpoint since the integrals that it involves do not have a closed-form expression. There is, however, one particular case where a closed-form solution is available: the case where all mixture models are vector quantizers.

Vector quantizers

Using Theorem 5, the VQ case can be analyzed by assuming Gaussian feature class-conditional densities and investigating what happens when all covariances tend to zero. This leads to the following result.

Lemma 6 *For a retrieval problem with Gaussian mixtures for the query (6.2) and database densities (6.3)*

$$P(\mathbf{x}) = \sum_{j=1}^{C} \mathcal{G}(\mathbf{x}, \mu_{q,j}, \epsilon \mathbf{\Sigma}_{q,j}) P(\omega_j)$$

$$P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \mathcal{G}(\mathbf{x}, \mu_{i,k}, \epsilon \mathbf{\Sigma}_{i,k}) P_i(\omega_k)$$

where $\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})$ *is defined in (2.5), when* $\epsilon \to 0$

$$\begin{aligned}
&\int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \\
&= \sum_j P(\omega_j) \log P_i(\omega_{\alpha(j)}) \\
&+ \sum_j P(\omega_j) \lim_{\epsilon \to 0} \left[\log \mathcal{G}(\hat{\mu}_{q,j,\alpha(j)}, \mu_{i,\alpha(j)}, \epsilon \mathbf{\Sigma}_{i,\alpha(j)}) - \frac{1}{2\epsilon} trace[\mathbf{\Sigma}_{i,\alpha(j)}^{-1} \hat{\Sigma}_{q,j,\alpha(j)}] \right],
\end{aligned}$$

where χ_k *is as defined in Lemma 2,* $\hat{\mu}_{q,j,\alpha(j)}$ *and* $\hat{\Sigma}_{q,j,\alpha(j)}$ *as defined in Lemma 5, and*

$$\alpha(j) = k \text{ such that } ||\mu_{q,j} - \mu_{i,k}||^2_{\mathbf{\Sigma}_{i,k}} < ||\mu_{q,j} - \mu_{i,l}||^2_{\mathbf{\Sigma}_{i,l}} \; \forall l \neq k.$$

Proof: When $\epsilon \to 0$,

$$\mathcal{G}(\mathbf{x}, \mu_{q,j}, \epsilon \mathbf{\Sigma}_{q,j}) \to \delta(\mathbf{x} - \mu_{q,j})$$

and since, from the definition of the delta function,

$$\int f(\mathbf{x}) \delta(\mathbf{x} - \mu) d\mathbf{x} = f(\mu),$$

it follows that

$$\int_{\chi_k} P(\mathbf{x}|\omega_j) d\mathbf{x} \to \chi_k(\mu_{q,j}).$$

On the other hand, from Theorem 5 and the definition of χ_k, if $\epsilon \to 0$ then

$$P_i(\omega_k|\mathbf{x}) \to 1 \, \forall \mathbf{x} \in \chi_k,$$

and $\chi_k(\mu_{q,j}) = 1$ if and only if

$$(\mu_{q,j} - \mu_{i,k})\mathbf{\Sigma}_{i,k}^{-1}(\mu_{q,j} - \mu_{i,k}) < (\mu_{q,j} - \mu_{i,l})\mathbf{\Sigma}_{i,l}^{-1}(\mu_{q,j} - \mu_{i,l}), \forall l \neq k.$$

The lemma follows from the application of these results to (6.10). □

We are now ready to derive a closed-form expression for global similarity under Bayesian retrieval with VQ density estimates.

Theorem 6 *For a retrieval problem with VQ density estimates for the query (6.2) and database densities (6.3)*

$$P(\mathbf{x}) = \sum_{j=1}^{C} \delta(\mathbf{x}, \mu_{q,j})P(\omega_j)$$

$$P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \delta(\mathbf{x} - \mu_{i,k})P_i(\omega_k),$$

when evaluating global similarity the Bayesian retrieval criteria reduces to

$$\begin{aligned}
&\arg\max_i \int P(\mathbf{x}) \log P_i(\mathbf{x})d\mathbf{x} = \\
&= \arg\min_i \lim_{\lambda\to\infty} \left\{ \sum_j P(\omega_j) \log \frac{P(\omega_j)}{P_i(\omega_{\alpha(j)})} + \lambda \sum_j P(\omega_j)||\mu_{q,j} - \mu_{i,\alpha(j)}||^2 \right\} \quad (6.13)
\end{aligned}$$

where

$$\alpha(j) = k \text{ such that } ||\mu_{q,j} - \mu_{i,k}||^2 < ||\mu_{q,j} - \mu_{i,l}||^2, \forall l \neq k.$$

Proof: From (6.11) and (6.12), when $\epsilon \to 0$

$$\hat{\mu}_{q,j,\alpha(j)} \to \mu_{q,j}$$

$$\hat{\Sigma}_{q,j,\alpha(j)} \to \epsilon\mathbf{\Sigma}_{q,j}.$$

Using Lemma 6,

$$\begin{aligned}
&\arg\max_i \int P(\mathbf{x}) \log P_i(\mathbf{x})d\mathbf{x} = \\
&= \arg\max_i \left\{ \sum_j P(\omega_j) \log P_i(\omega_{\alpha(j)}) + \sum_j P(\omega_j) \lim_{\epsilon\to 0} \log \mathcal{G}(\mu_{q,j}, \mu_{i,\alpha(j)}, \epsilon\mathbf{\Sigma}_{i,\alpha(j)}) \right. \\
&\qquad \left. - \sum_j P(\omega_j) \frac{1}{2} trace[\mathbf{\Sigma}_{i,\alpha(j)}^{-1}\mathbf{\Sigma}_{q,j}] \right\}.
\end{aligned}$$

Since, for a vector quantizer, $\mathbf{\Sigma}_{i,k} = \mathbf{\Sigma}_{q,j} = \mathbf{I}, \forall k, j$, the third term on the right-hand side of the above equation does not depend on i, and setting $\lambda = 1/2\epsilon$ leads to

$$
\begin{aligned}
&\arg\max_i \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \\
&= \arg\max_i \left\{ \sum_j P(\omega_j) \left[\log P_i(\omega_{\alpha(j)}) - \lim_{\lambda \to \infty} \lambda ||\mu_{q,j} - \mu_{i,\alpha(j)}||^2 \right] \right\} \\
&= \arg\min_i \lim_{\lambda \to \infty} \left\{ \sum_j P(\omega_j) \log \frac{P(\omega_j)}{P_i(\omega_{\alpha(j)})} + \lambda \sum_j P(\omega_j) ||\mu_{q,j} - \mu_{i,\alpha(j)}||^2 \right\}. \square
\end{aligned}
$$

The theorem states that, for VQ-based density estimates, Bayesian retrieval is equivalent to a constrained optimization problem [11]. Given a query VQ and a VQ associated with a database image class, 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 terms resulting from this procedure: a term that accounts for the average distortion of the quantization ($\sum_j P(\omega_j)||\mu_{q,j} - \mu_{i,\alpha(j)}||^2$) and the KL divergence between the feature-class probability distributions. λ is a Lagrange multiplier that weighs the contribution of the two terms.

By making $\lambda \to \infty$, 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 becomes that of (6.9), i.e. the KL divergence between label histograms. Since equal quantizers with equal codewords define equal partitions of the feature space, this situation is equivalent to that of histogramming and the result is, therefore, not surprising.

If the quantizers have different codewords (and consequently define different partitions), the quantization distortion becomes predominant and the retrieval criteria becomes

$$\arg\max_i \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} = \arg\min_i \sum_j P(\omega_j) ||\mu_{q,j} - \mu_{i,\alpha(j)}||^2.$$

Computationally, this reduces the complexity of the retrieval operation from $O(NCn)$ to $O(C^2 n)$. Since C, the number of feature classes, is fixed and orders of magnitude smaller than the cardinality of the query, N, the resulting savings are very significant. In fact, using the typical values of section 6.2.2,

$$4Cost_{mrsar/md} < Cost_{emm/ml} < 16Cost_{mrsar/md},$$

$$Cost_{hist/L^p} < Cost_{emm/ml} < 2Cost_{hist/L^p},$$

rendering the complexity of Bayesian retrieval with EMM similar to that of the standard approaches.

The asymptotic likelihood approximation

Vector quantization is a case of particular interest not only because it has a closed-form solution for global similarity, but also because the analysis performed for VQ provides insight on how to approximate (6.10) for generic Gaussian mixtures. In particular, Lemma 6 suggests the following approximation.

Definition 5 *Given a retrieval problem with Gaussian mixtures for the query (6.2) and database densities (6.3)*

$$P(\mathbf{x}) = \sum_{j=1}^{C} \mathcal{G}(\mathbf{x}, \mu_{q,j}, \mathbf{\Sigma}_{q,j}) P(\omega_j)$$

$$P_i(\mathbf{x}) = \sum_{k=1}^{C_i} \mathcal{G}(\mathbf{x}, \mu_{i,k}, \mathbf{\Sigma}_{i,k}) P_i(\omega_k),$$

the asymptotic likelihood approximation (ALA) is defined by

$$\begin{aligned} \int P(\mathbf{x}) \log P_i(\mathbf{x}) d\mathbf{x} \approx & \\ \approx & \sum_j P(\omega_j) \log P_i(\omega_{\alpha(j)}) \\ + & \sum_j P(\omega_j) \left[\log \mathcal{G}(\mu_{q,j}, \mu_{i,\alpha(j)}, \mathbf{\Sigma}_{i,\alpha(j)}) - \frac{1}{2} trace[\mathbf{\Sigma}^{-1}_{i,\alpha(j)} \mathbf{\Sigma}_{q,j}] \right], \end{aligned}$$

where

$$\alpha(j) = k \textit{ such that } ||\mu_{q,j} - \mu_{i,k}||^2_{\mathbf{\Sigma}_{i,k}} < ||\mu_{q,j} - \mu_{i,l}||^2_{\mathbf{\Sigma}_{i,l}}, \forall l \neq k.$$

A comparison of the ALA with the true likelihood of (6.10) reveals two assumptions underlying this approximation.

Assumption 3 *Each cell χ_k of the partition determined by $P_i(\mathbf{x})$ is assigned to one feature class with probability one, i.e.*

$$P_i(\omega_k|\mathbf{x}) = 1, \;\; \forall \mathbf{x} \in \chi_k.$$

Assumption 4 *The support set of each feature class-conditional density of the query mixture is entirely contained in a single cell χ_k of the partition determined by $P_i(\mathbf{x})$. I.e.,*

$$\forall j\ \exists k : \mathcal{S}(\omega_j) \subset \chi_k.$$

Under Assumption 3, the third term of (6.10) vanishes. Under Assumption 4, $\int_{\chi_k} P(\mathbf{x}|\omega_j) = \delta_{k,\alpha(j)}$, $\hat{\mu}_{q,j,\alpha(j)} = \mu_{q,j}$, and $\hat{\Sigma}_{q,j,\alpha(j)} = \mathbf{\Sigma}_{q,j}$. 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 covariances. However, Assumption 3 will only be reasonable if the feature class-conditional densities of $P_i(\mathbf{x})$ have reduced overlap. This implies that the distance between each pair of $\mu_{i,q}$ should be larger than the spread of the associated Gaussians. A 1-D illustration of this effect is provided by Figure 6.6, where we show two Gaussians class-conditional likelihoods and the posterior probability function $P_i(\omega_0|\mathbf{x})$ for class 0. As the separation between the Gaussians increases, the posterior probability changes more abruptly and the partition becomes harder.

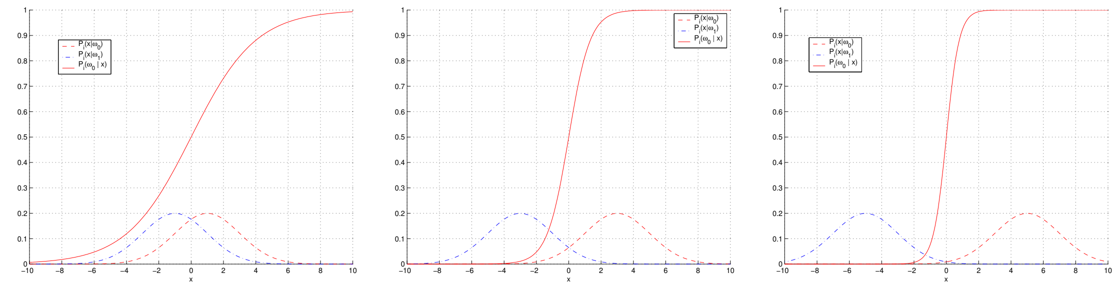

Figure 6.6: Impact of the separation between two Gaussian class conditional likelihoods on the partition of the feature space that they determine.

Assumption 4 never really holds for Gaussian mixtures, since Gaussians have infinite support. However, if Assumption 3 holds and, in addition, the spread of the Gaussians in $P(\mathbf{x})$ is much smaller than the cells χ_k, then

$$\int_{\chi_{\alpha(j)}} P(\mathbf{x}|\omega_j) d\mathbf{x} \approx 1$$

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. This is already visible in Figure 5.3, where it is clear that as the dimension of the space grows the Gaussians tend to have smaller overlap. To validate this point with more concrete evidence, we performed the following experiment

- a $10,000$ point sample was drawn from the mixture model of Figure 5.3;
- for each sample point $\mathbf{x}_i, i = 1, \ldots, 10,000$, we evaluated the maximum posterior class-assignment probability $max_k P(\omega_k|\mathbf{x}_i)$;
- the maximum posterior probabilities were histogramed.

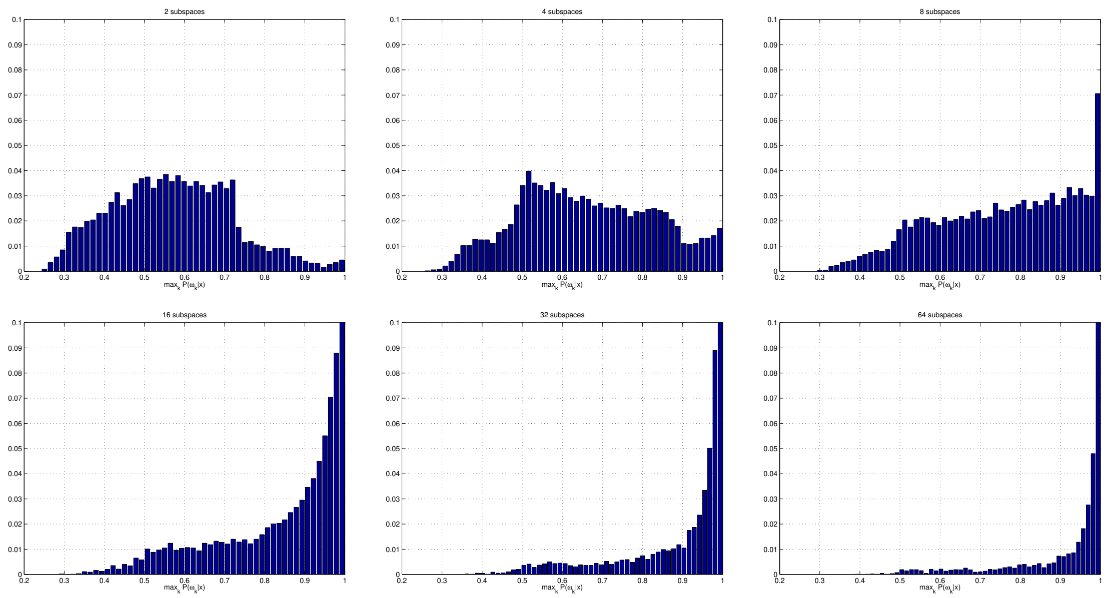

Figure 6.7: Maximum class posterior probability histograms for different numbers of subspaces of the EMM of Figure 5.3.

The experiment was repeated for several mixture models obtained by restricting the EMM of Figure 5.3 to an increasing number of subspaces. Figure 6.7 presents the histograms of the maximum posterior probability obtained with 2, 4, 8, 16, 32, and 64 subspaces. It is clear that, in high-dimensional spaces, Assumption 3 is realistic.

6.2.4 Experimental evaluation

We are now ready to conclude the experimental evaluation of probabilistic retrieval with EMMs. Since the Brodatz and Columbia databases contain only specific types of visual concepts (textures and objects), are organized into relatively unambiguous classes, and each of their images consists of only one concept, these databases provide a controlled environment that enables important insights on the different retrieval solutions. However, because realistic image retrieval rarely occurs under such controlled circumstances, it is important to validate the results obtained so far with evaluation on a generic database. In particular, it is important to consider databases that require joint modeling of texture and color. In this section, we consider one such database (Corel) and compare the performance of probabilistic retrieval against that of the domain-specific approaches discussed so far (HI and MRSAR/MD) and the two other approaches that, to the best of our knowledge, represent the state of the art in the joint modeling of the two attributes: color autocorrelograms and linear combinations of color and texture.

In these experiments, we used mixtures of 8 Gaussians and a spacing of four pixels between consecutive training samples. The implementations of MRSAR and HI were as discussed above, in the latter case we used a histogram with 512 bins. For color autocorrelograms, we followed the implementation of Huang et al. [66]. One of the limitations of the autocorrelogram is that, because each of its entries is a probability conditioned on a different event, it cannot be combined with probabilistic retrieval criteria such as ML. We therefore relied on the variation of the L^1 norm proposed in [66] as a similarity criteria. In order to make a fair comparison, we restricted its region of support to be the 8×8 pixel window also used by the embedded mixtures, i.e. the set of distances used for computing the autocorrelogram was $D = 1, 2, 3$. Overall, the autocorrelogram contained $2,048$ bins.

To combine linearly color and texture, we started by evaluating all the distances between query and database entries according to both HI and MRSAR/MD. For each query, we then normalized all distances by their mean and variance, clipped all values with magnitude larger than three standard deviations, and mapped the resulting interval into $[0, 1]$. This is a standard normalization to ensure that the distances relative to the two attributes are of the same order of magnitude [72, 44, 150]. An overall distance was then computed for each entry in the database, according to

$$d' = (1 - w)d_c + wd_t$$

where d_c and d_t are, respectively, the normalized distances according to HI and MRSAR/MD, and $w \in [0, 1]$ a pre-defined weight. These distances were then used to rank all the entries and measure precision/recall.

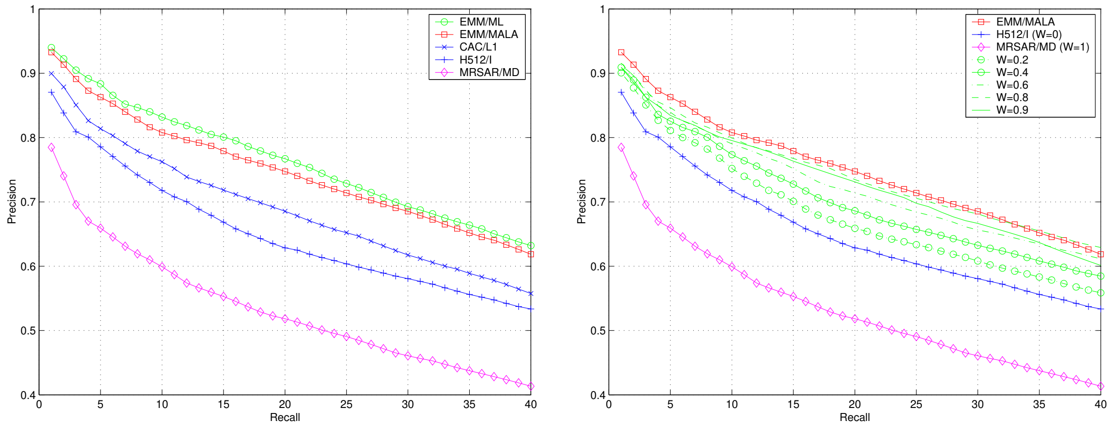

Figure 6.8: Left: precision/recall on Corel for MRSAR/MD, HI, color autocorrelogram (CAC), EMM/ML, and EMM/MALA. Right: comparison of precision /recall achieved with EM/MALA and linear weighting of MRSAR/MD and HI for different weights.

The left plot on Figure 6.8 presents the precision/recall curves for the different retrieval solutions. It is clear that the texture model alone performs very poorly, color histogramming does significantly better, and the autocorrelogram further improves performance by about 5%. However, all these approaches are significantly less effective than either EMM/ML or EM/MALA (where we maximize the asymptotic likelihood approximation discussed in the previous section). Furthermore, there is no significant difference between the two EMM approaches. This confirms the argument that, for global queries, 1) ALA is a good approximation to the true likelihood, and 2) EMM/MALA is the best overall solution when one takes computational complexity into account.

Finally, the right plot on the figure compares the precision/recall curves of EMM/MALA with those obtained by linear weighting of the color and texture distances. Several curves are shown for values of $w \in [0, 1]$. It is clear that the performance of the latter approach is never better than that of EM/MALA. Given that, in a realistic retrieval scenario, the value of the optimal weight is not known, there are no intuitive ways to determine it, and the linear combination always requires an increase in complexity (distances have to be computed according to the two representations), we see no reason to prefer these types of solutions to probabilistic retrieval.

Figure 6.9: Outcome of queries on Corel for horses and cars.

We conclude this chapter by giving some visual examples of the outcome of queries in the Corel database. Figures 6.9 to 6.11 present typical results for queries with horses, cars, diving scenes, gardens, and paintings. These pictures illustrate some of the nice properties

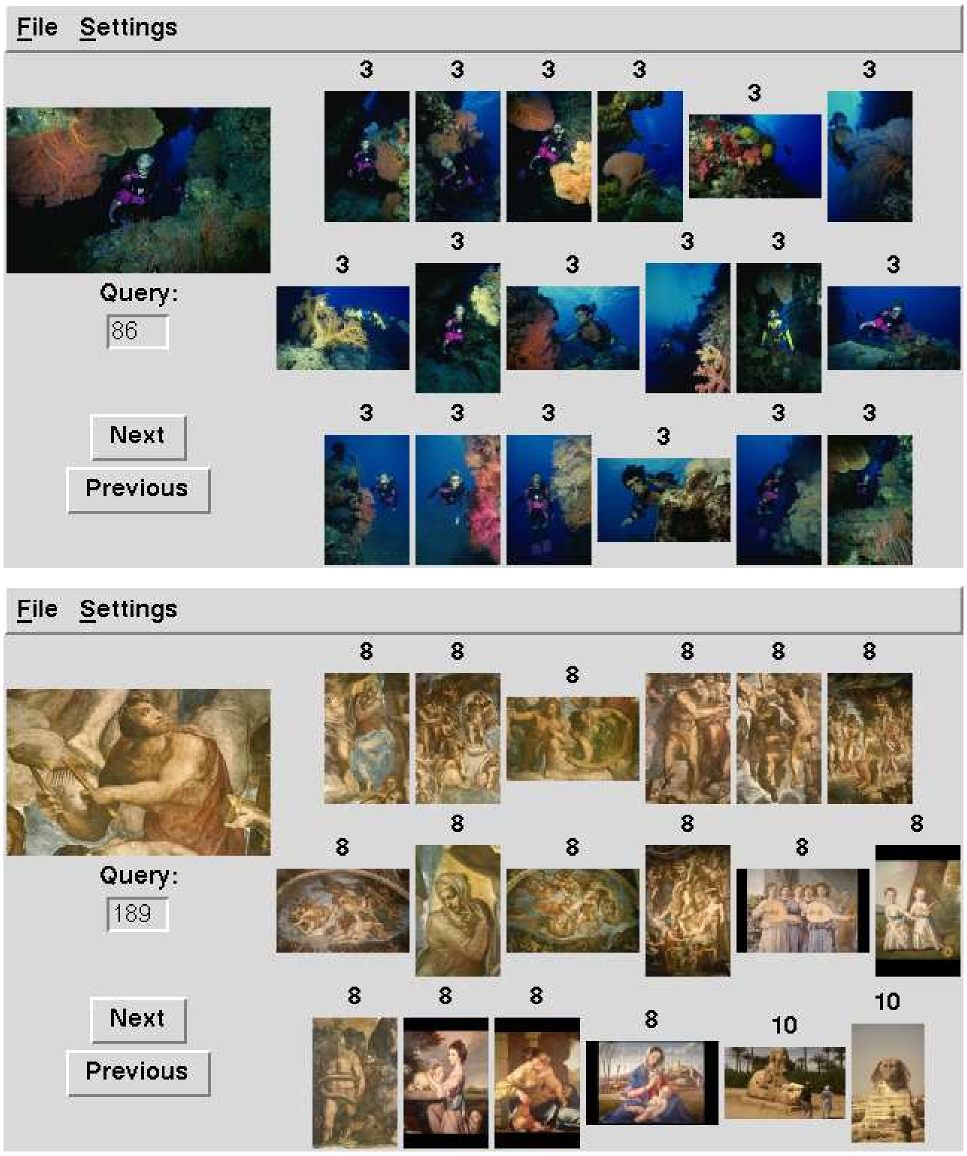

Figure 6.10: Outcome of queries on Corel for diving scenes and paintings.

Figure 6.11: Outcome of queries on Corel for gardens.

of the probabilistic retrieval formulation: robustness to changes in object position and orientation, robustness against the presence of distracting objects in the background, good performance even when there are large chunks of missing data in the query (notice that, in the diving example, even though almost no sea is visible in the query, the retrieved images are all from the right class and most contain large patches of blue), and perceptually intuitive errors (in the painting example, two pictures of the sphinx - pyramids class - are returned after all the paintings of human figures).