Foundations and Trends® in
Signal Processing
Vol. 5, No. 4 (2011) 265–389

DOI: 10.1561/2000000015

now
the essence of knowledge

Minimum Probability of Error Image Retrieval: From Visual Features to Image Semantics

By Nuno Vasconcelos and Manuela Vasconcelos

Contents

Foundations and Trends® in
Signal Processing
Vol. 5, No. 4 (2011) 265–389

DOI: 10.1561/2000000015

now
the essence of knowledge

Minimum Probability of Error Image Retrieval: From Visual Features to Image Semantics

Nuno Vasconcelos[1] and Manuela Vasconcelos[2]

[1] *University of California, San Diego, 9500 Gilman Drive, MC 0407, La Jolla, CA 92093, USA, nuno@ece.ucsd.edu*
[2] *University of California, San Diego, 9500 Gilman Drive, MC 0407, La Jolla, CA 92093, USA, maspcv@gmail.com*

Abstract

The recent availability of massive amounts of imagery, both at home and on the Internet, has generated substantial interest in systems for automated image search and retrieval. In this work, we review a principle for the design of such systems, which formulates the retrieval problem as one of decision-theory. Under this principle, a retrieval system searches the images that are likely to satisfy the query with *minimum probability of error* (MPE). It is shown how the MPE principle can be used to design optimal solutions for practical retrieval problems. This involves a characterization of the fundamental performance bounds of the MPE retrieval architecture, and the use of these bounds to derive optimal components for retrieval systems. These components include a feature space where images are represented, density estimation methods to produce this representation, and the similarity function to be used for image matching. It is also

shown that many alternative formulations of the retrieval problem are closely related to the MPE principle, typically resulting from simplifications or approximations to the MPE architecture. The MPE principle is then applied to the design of retrieval systems that work at different levels of abstraction. *Query-by-visual-example* (QBVE) systems are strictly visual, matching images by similarity of low-level features, such as texture or color. This is usually insufficient to produce perceptually satisfying results, since human users tend to make similarity judgments on the basis of image semantics, not visual attributes. This problem is addressed by the introduction of MPE labeling techniques, which associate descriptive keywords with images, enabling their search with text queries. This involves computing the probabilities with which different concepts explain each image. The query by example paradigm is then combined with these probabilities, by performing MPE image matching in the associated probability simplex. This is denoted *query-by-semantic-example* (QBSE), and enables example-based retrieval by similarity of semantics.

1

From Pixels to Semantic Spaces: Advances in Content-Based Image Search

We are currently living through a confluence of three technological revolutions – the advent of digital imaging, broadband networking, and inexpensive storage – that allow millions of people to communicate and express themselves by sharing media. It could be argued, however, that a few pieces are still missing. While it is now trivial to acquire, store, and transmit images, it is significantly harder to manipulate, index, sort, filter, summarize, or search through them. Significant progress has, without doubt, happened in domains where the visual content is tagged with text descriptions, due to the advent of modern search engines and their image/video search off-springs. Nevertheless, because they only analyze *metadata*, not the images per se, these are of limited use in many practical scenarios. For example the reader can, at this moment, use one of the major image search engines to download $7,860,000$ pictures of "kids playing soccer", most served from Internet sites across the world. Yet, these are all useless, to *the reader*, when he/she is looking for pictures of *his/her* kids playing soccer. Although the latter are stored in the reader's hard-drive, literally at "hand's reach", they are completely inaccessible in any organized manner. The reader could, of course, take the time to manually label them, enabling the computer to

perform more effective searches, but this somehow feels wrong. After all, the machine should be working for the user, not the other way around.

The field of *content-based image search* aims to develop systems capable of retrieving images because they understand them and are able to represent their content in a form that is intuitive to humans. It draws strongly on computer vision and machine learning, and encompasses many sub-problems in image representation and intelligent system design. These include the evaluation of image similarity, the automatic annotation of images with descriptive captions, the ability to understand user feedback during image search, and support for indexing structures that can be searched efficiently. In this monograph, we review the progress accomplished in this field with a formulation of the problem as one of decision theory. We note that the decision theoretic view is not the only possible solution to the retrieval problem and that many alternatives have been proposed in the literature. These alternatives are covered by recent extensive literature reviews [24, 68, 105, 115] and will not be discussed in what follows, other than in context of highlighting possible similarities or differences to MPE retrieval.

1.1 Query by Visual Example

Query by visual example (QBVE) is the classical paradigm for content-based image search. It is based on strict visual matching, ranking database images by similarity to a user-provided query image. The steps are as follows: user provides query, retrieval system extracts a signature from it, this signature is compared to those previously computed for the images in the database, and the closest matches are returned to the user. There are, of course, many possibilities for composing image signatures or evaluating their similarity, and a rich literature has evolved on this topic [105]. While early solutions, such as the pioneering *query-by-image-content* system [80], were based on very simple image processing (e.g., matching of histograms of image colors), modern systems (1) rely on more sophisticated representations, and (2) aim for provably optimal retrieval performance.

In what follows, we review one such approach, usually denoted as *minimum probability of error* (MPE) retrieval. The retrieval problem is

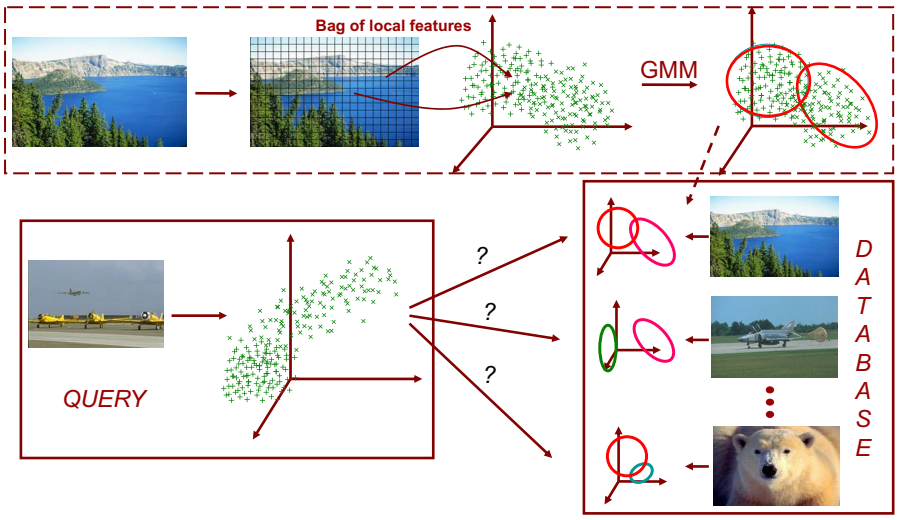


Fig. 1.1 MPE retrieval architecture. Images are decomposed into bags of local features, and characterized by their distributions on feature space. Database images are ranked by posterior probability of having generated the query features.

formulated as one of classification, and all components of the retrieval system are designed to achieve optimality in the MPE sense. This leads to the retrieval architecture depicted in Figure 1.1. Images are first represented as bags of local features (that measure properties such as texture, edginess, color, etc.), and a probabilistic model (in the figure a Gaussian mixture) is learned from the bag extracted from each image. The image signature is, therefore, a compact probabilistic representation of how it populates the feature space. When faced with a query, the retrieval system extracts a bag of features from it, and computes how well this bag is explained by each of the probabilistic models in the database. In particular, it ranks the database models according to their posterior probability, given the query. As we will see later on, this is optimal in the MPE sense.

Note that, besides finding the closest matches, the system assigns a probability of match to all images in the database. This allows the combination of visual matching with other sources of information that may impact the relevance of each database image. For example, the text in an accompanying web page [92], how well the image matches previous

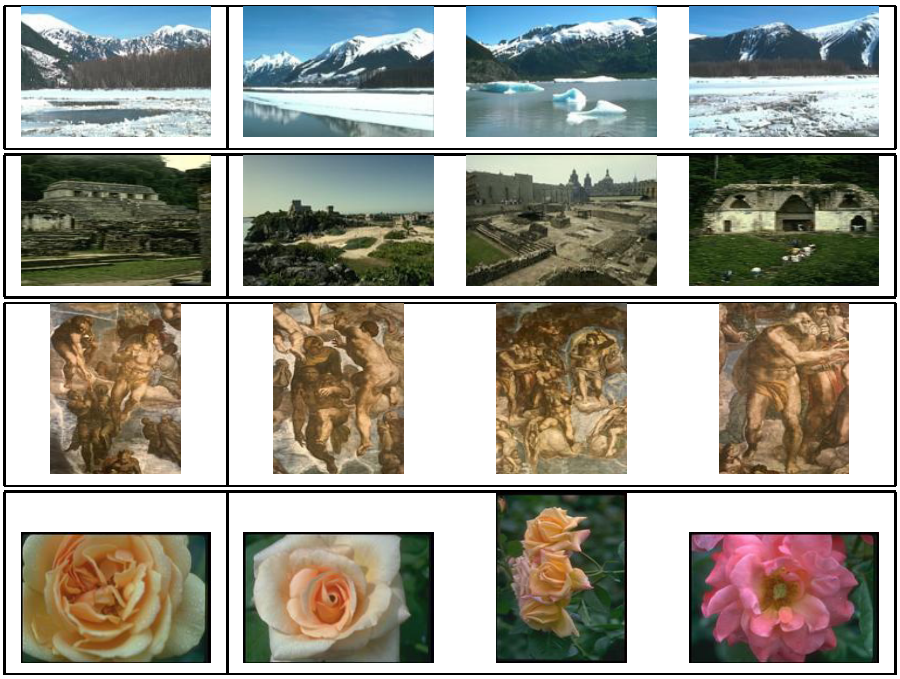


Fig. 1.2 MPE retrieval results. Each row shows the top three matches (among 1,500) to the query on the left.

queries [127, 128], external events that could increase the relevance of certain images on certain days (e.g., high demand for football images on Sunday night), etc.

The retrieval architecture of Figure 1.1 is currently among the top performers in QBVE [124]. These systems work well when similarity of visual appearance correlates with human judgments of similarity. This is illustrated by Figure 1.2, which presents the top matches, from a database of 1500 images, to four queries. Note that the database is quite diverse, and the images are basically unconstrained in terms of lighting conditions, object poses, etc. (even though they are all good quality images taken by professional photographers). The system is able to identify the different visual attributes that, in each case, contribute to the perception of image similarity. For example, similar color distributions seem to be determinant in the matches of the first row, while texture appears to play a more significant role in the third, shape (of the

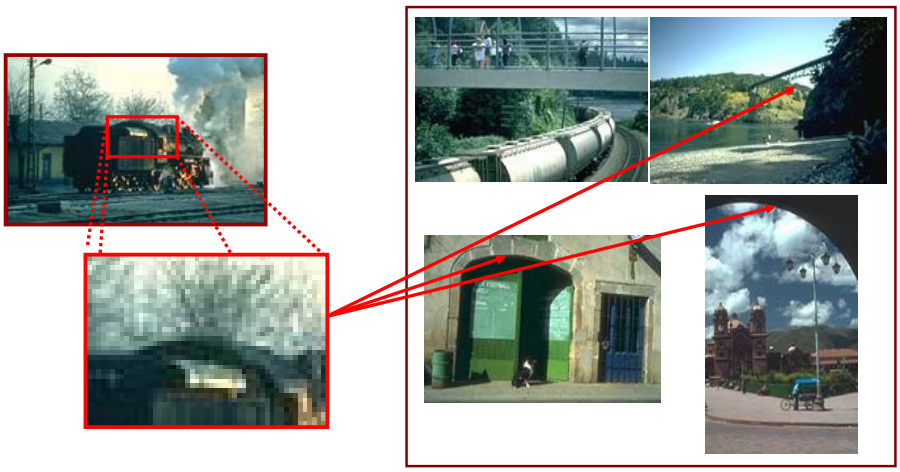


Fig. 1.3 A query image (left) and its top four matches by a QBVE system (right). Humans frequently discard strong visual cues in their similarity judgments. Failure to do this can lead to severe QBVE errors. For example, the visually distinctive arch-like structure in the train query induces the QBVE system to retrieve images of bridges or other arch-like structures.

flower petals) is probably the strongest cue for the results of the fourth, and the matches of the second row are likely due to the commonality of edge patterns in the building structures present in all images.

There are, nevertheless, many queries for which visual similarity does not correlate strongly with human similarity judgments. Figure 1.3 presents an example of how people frequently discard very strong visual cues in their similarity judgments. As can be seen from the close-up, the "train" query contains a very predominant arch-like structure. From a strictly visual standpoint, this makes it very compatible with concepts such as "bridges" or "arches". A QBVE system will fall in this trap, returning as top matches the four images also shown. Note that three of these do contain bridges or arch-like structures. Yet, the "train" interpretation of the query is completely dominant for humans, which assign very little probability to the alternative interpretations, and expect images of trains among the retrieved results.

The mismatch between the similarity judgments of user and machine can make the retrieval operation very unsatisfying. In the "train" example, most people would not be to able justify the matches

returned by the retrieval system, despite the obvious similarities of the visual stimuli. This is the nightmare scenario for image retrieval, since users not only end up unhappy with the retrieval results, but also acquire the feeling that the system just "does not get it". This can be an enormous source of user frustration.

1.2 Semantic Retrieval

The discussion above reveals what is often called a *semantic gap* between user and machine. Unlike QBVE systems, people seem to first classify images as belonging to a number of semantic classes, and then make judgments of similarity in the higher level semantic space where those classes are defined. This has motivated significant interest, over the last decade, in semantic image retrieval. A semantic retrieval system aims for the two complementary goals of image *annotation* and *search*. The starting point is a training image database, where each image is annotated with a natural language caption, from which the retrieval system learns a *mapping between words and visual features*. This mapping is then used to (1) annotate unseen images with the captions that best describe them, and (2) find the database images that best satisfy a natural language query.

Usually, the training corpus is only *weakly labeled*, in the sense that (1) the absence of a label from a caption does not necessarily mean that the associated visual concept is absent from the image, and (2) it is not known which image regions are associated with each label. For example, an image containing "sky" may not be explicitly annotated with that label and, when it is, no indication is available regarding which image pixels actually depict sky. Note that the implementation of a semantic retrieval system does not require individual users to label training images. While this can certainly be supported, to personalize the vocabulary, the default is to rely on generic vocabularies, shared by many systems.

Under the MPE retrieval framework, a semantic retrieval system is a simple extension of a QBVE system. As shown in Figure 1.4, it can be implemented by learning probabilistic models from *image sets*, instead of single images. In particular, the set of training images labeled with

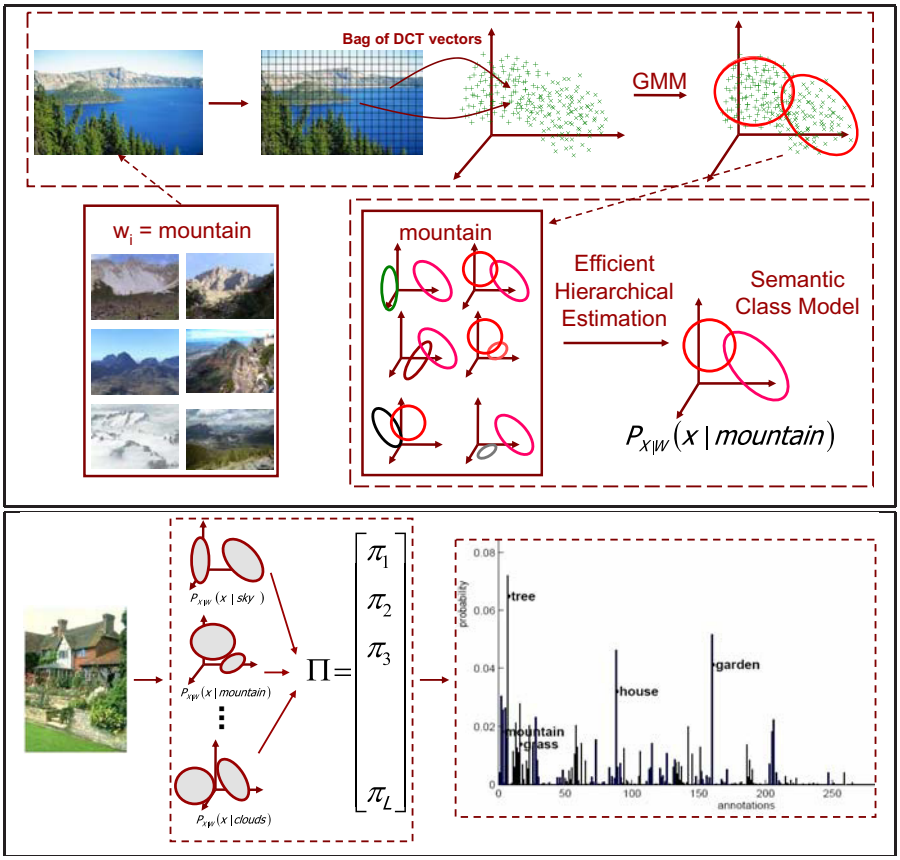


Fig. 1.4 Semantic MPE labeling. Top: images are grouped by semantic concept, and a probabilistic model learned for each concept. Bottom: each image is represented by a vector of posterior concept probabilities.

a particular keyword ("mountain", in the figure) is used to learn the model for the associated visual concept. As discussed in Section 6, this procedure converges to the true concept distribution plus a background uniform component that has small amplitude, if the set of training images is very diverse [16]. Given a set of models for different visual concepts, any image can be optimally labeled, in the MPE sense, by computing how well its features are explained by each model. In particular, the concepts are ordered by posterior probability, given the image, and the image is annotated with those of largest probability.

This is shown in Figure 1.4 where, among a vocabulary of more than 350 semantic concepts, an image of a country house receives, as most likely, the labels "tree", "garden", and "house".

It turns out that, under the MPE framework, it is possible to learn semantic models very efficiently, when individual image models are already available, i.e., when QBVE is also supported. In fact, it can be shown that the design of a semantic MPE retrieval system has complexity equivalent to that of an MPE system that only supports QBVE [16, 17]. Some examples of retrieval and annotation are shown in Figures 1.5 and 1.6. Note that the system recognizes concepts as diverse as "blooms", "mountains", "swimming pools", "smoke", or "woman". In fact, the system has learned that these classes can exhibit a wide diversity of patterns of visual appearance, e.g., that smoke can be both

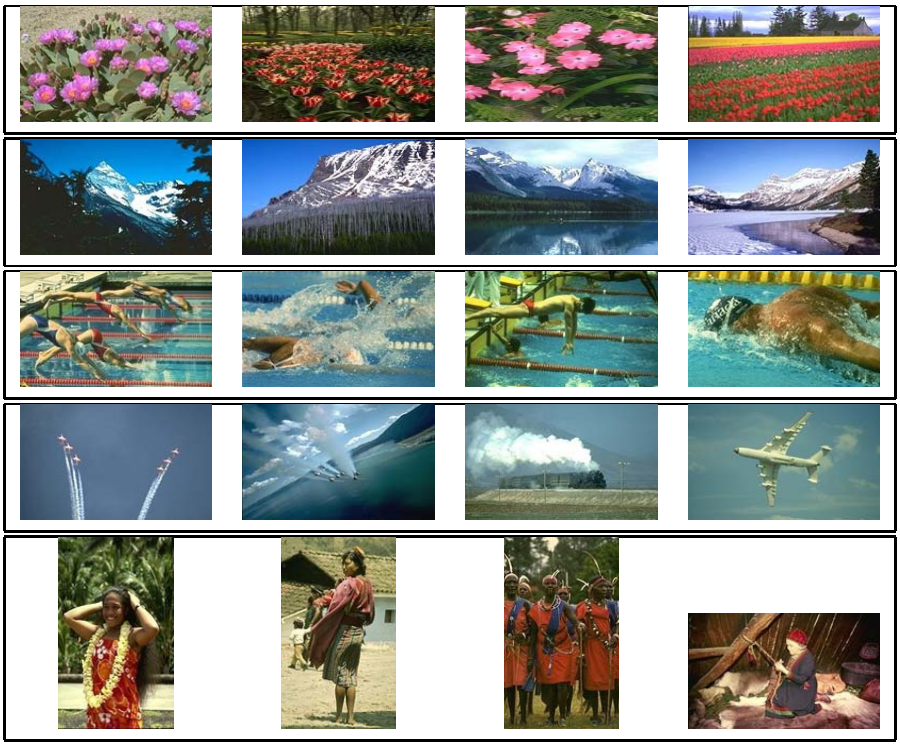


Fig. 1.5 Semantic retrieval results. Each row shows the top four matches to a semantic query. From first to fifth row: *'blooms'*, *'mountain'*, *'pool'*, *'smoke'*, and *'woman'*.

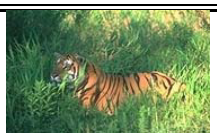
			
Human Annotation	sky jet plane smoke	snow fox arctic	sky buildings street cars
Automated Annotation	plane jet smoke flight prop	arctic snow polar fox ice	street buildings bridge sky arch
			
Human Annotation	grass forest cat tiger	bear polar snow tundra	coral fish ocean reefs
Automated Annotation	cat tiger plants leaf grass	polar tundra bear snow ice	reefs coral ocean fan fish
			
Human Annotation	water bridge train railroad	buildings clothes shops street	mountain sky clouds tree
Automated Annotation	sky bridge locomotive water train	buildings street shops people skyline	mountain valley sky clouds tree

Fig. 1.6 Comparison of the annotations produced by the system with those of a human subject.

white or very dark, that both blooms and humans can come in multiple colors, multiple sizes (depending on image scale), and multiple poses, or that pools can be mostly about water, mostly about people (swimmers), or both. This type of *generalization* is impossible for QBVE systems, where each image is modeled independently of the others.

The annotation results of Figure 1.6 illustrate a second form of generalization, based on *contextual relationships*, that humans also regularly exploit. For example, the fact that stores usually contain

people, makes us more prone to label an image of a store (where no people are visible) with the "people" keyword, than an image that depicts an animal in the wild. This is also the case for the MPE semantic retrieval system, whose errors tend to be (in significant part) due to this type of contextual associations. Note, for example, that the system erroneously associates the concept "prop" with a jet fighter, the concept "leaf" with grass, the concepts "people" and "skyline" with a store display, and so forth. Of course, there are also many situations in which these associations are highly beneficial and allow the correct identification of concepts that would otherwise be difficult to detect (due to occlusion, poor imaging conditions, etc.).

The ability to make such contextual generalizations stems from the weakly supervised nature of the training of the labeling system. Because concept models are learned from unsegmented images, most positive examples of "shop" are also part of the positive set for "people" (even though the latter will include many non-shopping related images as well). Hence, an image of a shop will originate some response from the "people" model, even when it does not contain people. That response will be weaker than that of an image of a shop that contains people, but stronger than the response of the "shop" model to a picture of people on a non-shopping context, e.g., fishing in a lake. These asymmetries are routine in human reasoning and, therefore, appear natural to users, making the errors of a semantic retrieval system less annoying than those of its QBVE counterpart. In fact, informal surveys conducted in our lab have shown that (1) humans frequently miss the labeling errors, and (2) even when the error is noted, the user can frequently find an explanation for it (e.g., "it confused a jet for a propeller plane"). This creates the sense that, even in making errors, the semantic retrieval system "gets it".

1.3 Exploring Semantic Feature Spaces

Despite all its advantages, semantic retrieval is not free of limitations. An obvious difficulty is that most images have multiple semantic interpretations. Since training images are usually labeled with a short caption, some concepts may never be identified as present. This reduces

the number of training examples and can impair the learning of concepts that (1) have high variability of visual appearance, or (2) are relatively rare. Furthermore, the semantic retrieval system is limited by the size of its vocabulary. Since it is still difficult to learn massive vocabularies, this can severely compromise generalization. It is, in fact, important to distinguish two types of generalization. The first is with respect to the concepts on which the system is trained, or *within the semantic space*. The second is with respect to all other concepts, or *outside the semantic space*.

While, as discussed in the previous section, semantic retrieval generalizes better (than QBVE) inside the semantic space, this is usually not true outside of it. One possibility, to address this problem, is to return to the query-by-example paradigm, but now at the semantic level, i.e., to adopt *query by semantic example* (QBSE) [91]. The idea is to represent each image by its vector of posterior concept probabilities (the π vector of Figure 1.4), and perform query by example in the simplex of these probabilities. Because the probability vectors are multinomial distributions over the space of semantic concepts, we refer to them as *semantic multinomials*. A similarity function between these objects is defined, the user provides a query image, and the images in the database are ranked by the distance of their semantic multinomials to that of the query. The process is illustrated in Figure 1.7.

When compared to semantic retrieval, a QBSE system is significantly less affected by the problems of (1) multiple semantic interpretations, and (2) difficult generalization outside of the semantic space. This follows from the fact that the system is not faced with a definitive natural language query, but an image that it expands into its internal semantic representation. For example, a system not trained with images of the concept "fishing", can still expand a query image of this subject into a number of alternative concepts, such as "water", "boat", "people", and "nets", in its vocabulary. This is likely to produce high scores for other images of fishing.

When compared to QBVE, QBSE has the advantage of a feature space where it is much easier to generalize. This is illustrated by Figure 1.8, which shows the QBSE matches to the query image of Figure 1.3. Note how these correlate much better with human

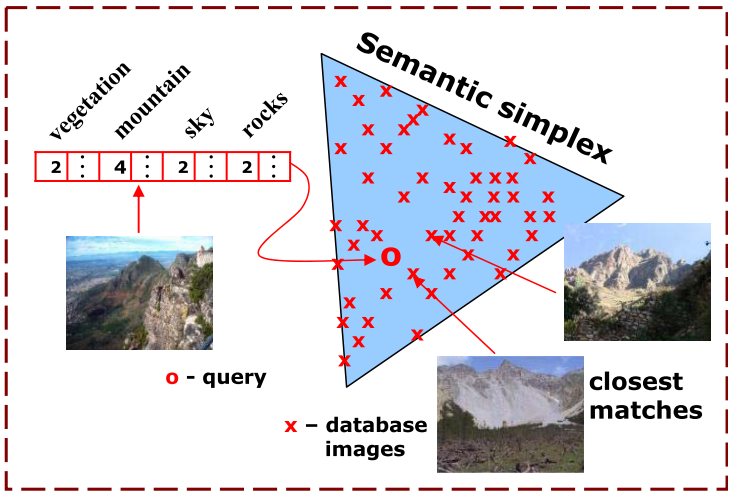


Fig. 1.7 Query by semantic example. Images are represented as vectors of concept probabilities, i.e., points on the semantic probability simplex. The vector computed from a query image is compared to those extracted from the images in the database, using a suitable similarity function. The closest matches are returned by the retrieval system.

Fig. 1.8 Top four matches to the QBSE query derived from the image shown on the left. Because good matches require agreement along various dimensions of the semantic space, QBSE is significantly less prone to the errors made by QBVE. This can be seen by comparing this set of image matches to those of Figure 1.3.

judgments of similarity that the QBVE matches of that figure. Inspection of the semantic multinomials associated with all images shown reveals that, although the query image receives a fair amount

of probability for the concept "bridge", it receives only slightly inferior amounts of probability for concepts such as "locomotive", "railroad", and "train". The latter are consistent with the semantic multinomials of other images depicting trains, but not necessarily with those of images depicting bridges. Hence, while the erroneous "bridge" label is individually dominant, it looses this dominance when the semantic multinomials are matched as a whole.

1.4 Organization of the Manuscript and Acknowledgments

In the following sections, we study in greater detail the fundamental properties of MPE retrieval. We start by laying out its theoretical foundations in Section 2. The sources of error of a retrieval system are identified, and upper and lower bounds on the resulting probability of error are derived. In Section 3, MPE retrieval architectures are related to a number of other approaches in literature. It is shown that many of the latter are special cases of the former, under simplifying assumptions that are not always sensible. In Section 4, we start to address the practical design of retrieval systems, by proposing a particular MPE implementation. This architecture is shown to have a number of interesting properties, and perform well in QBVE retrieval experiments. In Section 5, we consider the problem of semantic retrieval, by introducing MPE techniques for image annotation, and showing how they can be used to retrieve images with keyword-based queries. Some core technical issues in automated image annotation are then discussed in Section 6, where we study the possibility of learning image labels from weakly annotated training sets. The issue of generalization beyond the semantic space is introduced in Section 7, where we discuss QBSE. Finally, some conclusions are drawn in Section 8.

At this point, we would like to acknowledge the contributions of a number of colleagues that, over the last 10 years, have helped shape the research effort from which this work has resulted. Gustavo Carneiro has played an instrumental role in the development of the early ideas, from the design of multiple feature representations, to the first generation of our image annotation system. This work was then pursued by Antoni Chan, in a collaboration that also involved Pedro Moreno

at Google. This allowed us to evaluate the experimental performance of the theoretical ideas, at a scale that would not be possible in an academic laboratory. Nikhil Rasiwasia then took over, and developed most of the QBSE framework, as well as a number of more recent contributions that are not discussed here, mostly for lack of space. Since this manuscript follows closely a number of papers that we have co-written with all these colleagues, we will not include a more extensive discussion of who-did-what here. If interested, please refer to [16, 91, 119, 120, 124, 125]. Instead, we would like to thank a number of other people who were instrumental in the development of many of the ideas discussed here, including Andrew Lippman at MIT, and several students at the Statistical Visual Computing Laboratory at UCSD. These include Dashan Gao, Hamed Masnadi-Shirazi, Sunhyoung Han, and Vijay Mahadevan, among others. The many discussions that we have had over the years, about retrieval and related topics, have made our ideas much more clear and effective.

2

Theoretical Foundations of MPE Retrieval

In this section, we introduce the fundamental theoretical concepts underlying MPE retrieval. While, for simplicity, we concentrate on the QBVE paradigm, all concepts are equally applicable to semantic annotation and retrieval, or QBSE.

2.1 Minimum Probability of Error Retrieval Systems

A retrieval system is a mapping

$$g: \mathcal{X} \to \mathcal{Y} = \{1, \ldots, M\}$$

from a feature space $\mathcal{X}$ to the index set, $\mathcal{Y}$, of the M classes in the database. The retrieval system is optimal, under some suitable cost, if $\mathcal{X}$ and the similarity function $g(\cdot)$ are jointly optimized with respect to that cost. In this monograph we adopt the minimization of the probability of retrieval error as the goal for this optimization.

Definition 2.1. A retrieval system

$$g^*: \mathcal{X} \to \mathcal{Y}$$

that, for all $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$, minimizes

$$\text{Prob}[g(\mathbf{X}) \neq Y],$$

where $\mathbf{X}$ is the random variable from which the feature vector $\mathbf{x}$ is drawn and Y is the random variable that assigns $\mathbf{x}$ to its database class, is denoted a minimum probability of error (MPE) retrieval system.

It follows from this definition that an MPE-retrieval system is an optimal classifier [28, 30, 39, 57]. In fact, the MPE paradigm is a special case of the minimum average Bayes risk approach to classification, where the "0–1" loss or Hamming distortion is used as cost or risk function. The set $\mathcal{Y}$ of class labels depends on the desired retrieval functionality. For example, QBVE assumes that each image in the database is a class by itself. On the other hand, for image annotation, $\mathcal{Y}$ is a set of concepts defined as important for the semantic representation of the images to retrieve. Both the formulation and algorithms presented in this section are independent of the structure of $\mathcal{Y}$.

2.1.1 Minimum Probability of Error Classifiers

To analyze the fundamental limits of retrieval performance, we start by recalling a well known result on MPE classification [28, 39]: given a feature space $\mathcal{X}$ and query feature vector $\mathbf{x}$, the decision rule that minimizes the probability of retrieval error is the *Bayes* classifier

$$g^*(\mathbf{x}) = \arg\max_i P_{Y|\mathbf{X}}(i|\mathbf{x}), \tag{2.1}$$

where $P_{Y|\mathbf{X}}(i|\mathbf{x})$ is the posterior probability of class i given $\mathbf{x}$. This decision rule is well known in the communications literature as the maximum a posteriori probability classifier, and defines the similarity function of an MPE retrieval system. The probability of error of the Bayes classifier is the *Bayes error* (BE)

$$L^*_{\mathcal{X}} = 1 - E_{\mathbf{X}}[\max_i P_{Y|\mathbf{X}}(i|\mathbf{X})], \tag{2.2}$$

where $E_{\mathbf{X}}$ means expectation with respect to $P_{\mathbf{X}}$. This is a lower bound on the probability of error achievable with any other similarity function.

Although the Bayes classifier is the *optimal similarity function* for MPE retrieval, it assumes knowledge of the class-posterior probabilities $P_{Y|\mathbf{X}}(i|\mathbf{x})$. These are usually not available and must be estimated from a finite training sample. One popular solution is to rely on Bayes' rule to write (2.1) as

$$g^*(\mathbf{x}) = \arg\max_i P_{\mathbf{X}|Y}(\mathbf{x}|i)P_Y(i), \tag{2.3}$$

where $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ is the conditional probability density for the feature vectors drawn from the ith class and $P_Y(i)$ the prior probability for that class. The optimal decision rule is then approximated by

$$g(\mathbf{x}) = \arg\max_i \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)\hat{p}_Y(i), \tag{2.4}$$

where $\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)$ and $\hat{p}_Y(i)$ are estimates of the quantities in (2.3). If there is no a priori reason to favor any of the image classes in the database, it is acceptable to assume that the class priors $P_Y(i)$ are known and uniform, i.e., $\hat{p}_Y(i) = P_Y(i) = 1/M$. This leads to a decision function that depends only on estimates for the class-conditional densities $P_{\mathbf{X}|Y}(\mathbf{x}|i)$,

$$g(\mathbf{x}) = \arg\max_i \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i), \tag{2.5}$$

and is known as the *maximum likelihood* classifier. Because these class-conditional densities can be estimated independently for each class, the overall training complexity scales linearly in the number of classes, making this classifier architecture particularly appealing for problems, such as image retrieval or speech recognition [90], where that number is large. On the other hand, it should be emphasized that (2.5) is optimal, in the MPE sense, only insofar as the probability estimates are error-free. This is a requirement that is rarely met in practice, where density estimates are based on a finite data sample. In fact, when the feature space $\mathcal{X}$ is high-dimensional, the *density estimation error* can be substantial.

2.1.2 Impact of Density Estimation Errors on the Probability of Error

The impact of this error on the probability of error of the decision function of (2.5) can be quantified as follows.

Theorem 2.1. Consider a retrieval problem with equiprobable classes $P_Y(i) = 1/M, \forall i$, a feature space $\mathcal{X}$, unknown class conditional densities $P_{\mathbf{X}|Y}(\mathbf{x}|i)$, and the decision function of (2.5). For such a retrieval problem, the difference between the probability of error and the BE is upper bounded by

$$\text{Prob}[g(\mathbf{X}) \neq Y] - L^*_{\mathcal{X}} \leq \Delta_{g,\mathcal{X}}, \tag{2.6}$$

where

$$\Delta_{g,\mathcal{X}} = \frac{\sqrt{2\ln 2}}{M} \sum_i \sqrt{KL[P_{\mathbf{X}|Y}(\mathbf{x}|i)||\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)]} \tag{2.7}$$

is the density estimation error, and

$$\text{KL}[P_{\mathbf{X}|Y}(\mathbf{x}|i)||\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)] = \int P_{\mathbf{X}|Y}(\mathbf{x}|i) \log \frac{P_{\mathbf{X}|Y}(\mathbf{x}|i)}{\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)} d\mathbf{x} \tag{2.8}$$

the relative entropy or Kullback–Leibler (KL) divergence [22, 58, 88] between the true, $P_{\mathbf{X}|Y}(\mathbf{x}|i)$, and estimated, $\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)$, densities for class i.

Proof. See the Appendix. □

We note that (2.6) is a bound on the distance between the actual probability of error and the Bayes error, and substantially different from various bounds available in the information theoretic literature (see e.g., [12, 56]) that relate BE to the KL divergence between class densities. In these bounds, the KL divergence appears as a measure of discrimination and the bounds formalize the intuition that the BE decreases when the separation between class-conditional densities increases, i.e., with the increase of the KL divergence between classes. In the theorem above, the KL divergence does not take the role of a measure of discrimination but, instead, it appears as a measure of density estimation error. In particular, instead of the KL divergence between class-conditional densities, (2.6) is a function of the KL divergence between the true class-conditional densities and their estimates.

2.2 Impact of the Representation on the Bayes and Estimation Errors

So far, we have seen that *the probability of error of a retrieval system is lower bounded by its BE and upper bounded by the sum of its BE and estimation error*. The design of a retrieval system requires the specification of a feature space and a probability model for density estimation. These are denoted the *components of signal representation*. They affect the Bayes and estimation errors in very distinct ways. Since the BE only depends on the true densities, not their estimates, the only impact of the density model is on the estimation error of (2.7). The relationships between these two quantities have been extensively studied in the statistics literature, and are fairly well understood [30, 65, 102, 104].

For now, we concentrate on the dependence of the Bayes and estimation errors on the feature space. We assume the existence of a space of observations $\mathcal{Z}$, e.g., the space of $n \times n$ image blocks, and investigate the benefits of introducing a feature transformation $T: \mathcal{Z} \to \mathcal{X}$ in a retrieval system. A classical result [28] is that

$$L^*_{\mathcal{X}} \geq L^*_{\mathcal{Z}}, \tag{2.9}$$

where $L^*_{\mathcal{Z}}$ and $L^*_{\mathcal{X}}$ are, respectively, the BEs on $\mathcal{Z}$ and $\mathcal{X}$. Furthermore, equality is achieved if and only if T is an invertible transformation. This implies that the introduction of a feature transformation can *never* decrease the BE and seems to discourage the use of feature transformations. It turns out, however, that a feature transformation can also diminish the density estimation error. To show this, we start by considering sequences of nested vector spaces of increasing dimension, also known as sequences of embedded vector spaces [45].

Definition 2.2. A sequence of vector spaces $\{\mathcal{X}_1, \ldots, \mathcal{X}_d\}$, such that $\dim(\mathcal{X}_i) < \dim(\mathcal{X}_{i+1})$, is called embedded if there exists a sequence of one-to-one mappings

$$\epsilon_i : \mathcal{X}_i \to \mathcal{X}'_{i+1}, \quad i = 1, \ldots, d-1, \tag{2.10}$$

such that $\mathcal{X}'_{i+1} \subset \mathcal{X}_{i+1}$.

Embedded feature spaces enable a precise characterization of the trade-off between the Bayes and estimation error.

Theorem 2.2. Let

$$T{:}\mathbb{R}^d \to \mathcal{X} \subset \mathbb{R}^d$$

be a linear feature transformation, and

$$\pi_m^n : \mathbb{R}^n \to \mathbb{R}^m, \tag{2.11}$$

where $\pi_m^n(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) = (x_1, \ldots, x_m)$, the projection of the Euclidean space along the coordinate axes. Then,

$$\mathcal{X}_i = \pi_i^d(\mathcal{X}), i = 1, \ldots, d-1 \tag{2.12}$$

is a sequence of embedded feature spaces such that

$$L^*_{\mathcal{X}_{i+1}} \leq L^*_{\mathcal{X}_i}. \tag{2.13}$$

Furthermore, if $\mathbf{X}_1^d = \{\mathbf{X}_1, \ldots, \mathbf{X}_d\}$ is a sequence or random variables such that $\mathbf{X}_i \in \mathcal{X}_i$,

$$\mathbf{X}_i = \pi_i^d(\mathbf{X}), i = 1, \ldots, d, \tag{2.14}$$

and $\{g_i(\mathbf{x})\}_{i=1}^d$ a sequence of decision functions

$$g_i(\mathbf{x}) = \arg\max_k \hat{p}_{\mathbf{X}_i|Y}(\mathbf{x}|k), \tag{2.15}$$

then

$$\Delta_{g_{i+1},\mathcal{X}_{i+1}} \geq \Delta_{g_i,\mathcal{X}_i}, \tag{2.16}$$

where $\Delta_{g_{i+1},\mathcal{X}_{i+1}}$ is the density estimation error of (2.7).

Proof. See the Appendix. □

This theorem shows that any *linear feature transformation originates a sequence of embedded vector spaces with monotonically decreasing Bayes error, and monotonically increasing estimation error.* It follows that it is impossible to find a linear feature transformation that

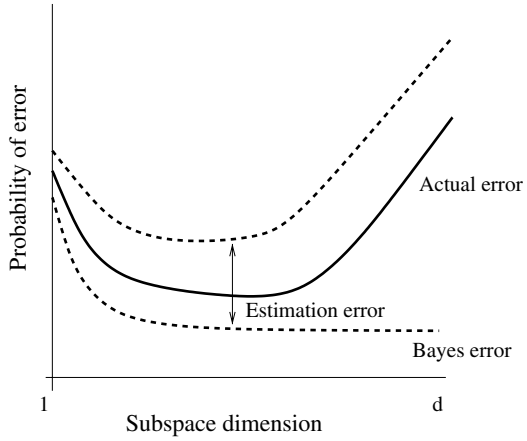


Fig. 2.1 Upper bound, lower bound, and probability of error as a function of subspace dimension.

can minimize the Bayes and estimation errors simultaneously. On one hand, given a feature space $\mathcal{X}$, it is possible to find a subspace where density estimates are more accurate. On the other hand, the projection onto this subspace will increase the BE. The practical result is that, for any feature transform used in a retrieval system, there is a need to reach a compromise between the two sources of error. This is illustrated by Figure 2.1, which shows the typical evolution of the upper and lower bounds on the probability of error as one considers successively higher-dimensional subspaces of a feature space $\mathcal{X}$. Since accurate density estimates can usually be obtained in low-dimensional spaces, the two bounds tend to be close when the subspace dimension is small. In this case, the probability of error is dominated by the BE. For higher-dimensional subspaces, the decrease in BE is cancelled by an increase in estimation error, and the actual probability of error increases. Overall, the curve of the probability of error exhibits the convex shape depicted in the figure, where an inflection point marks the subspace dimension for which BE ceases to be dominant. Different feature transforms will originate different curves.

3

A Unified View of Image Similarity

The MPE principle advocates a similarity function to be used in image retrieval. However, the measurement of similarity between signals has a very long history in communications and signal processing (see, e.g., [116]). Many of the classical similarity functions have been rediscovered in the context of image retrieval and classification. In this section, we present an overview of several similarity functions that are currently popular in the literature and can be seen as approximations, or simplifications, of that of MPE retrieval.

3.1 Approximations to MPE Similarity

Figure 3.1 illustrates how various similarity functions commonly used for image retrieval are special cases of MPE retrieval. While these functions do not exhaust the set of decisions rules that can be derived from or shown to be sub-optimal when compared to the MPE criteria (see Chapter 3 of [28] for several others), we concentrate on them for two reasons: (1) they *have been* proposed as similarity functions, and (2) when available, derivations of their relationships to MPE similarity are scattered around the literature.

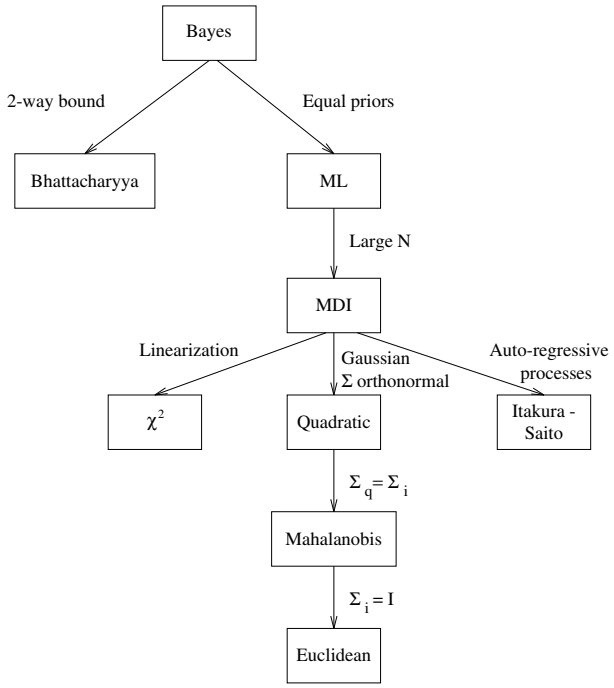


Fig. 3.1 Relations between different image similarity functions.

The figure illustrates that, if an upper bound on the Bayes error of a collection of two-way classification problems is minimized instead of the probability of error of the original problem, the MPE criteria reduces to the *Bhattacharyya distance* (BD). On the other hand, if the original criteria is minimized, but the different image classes are assumed to be equally likely a priori, we have the *maximum likelihood* (ML) retrieval criteria. As the number of query vectors approaches infinity, the ML criteria tends to the *minimum discrimination information* (MDI), which in turn can be approximated by the χ^2 test by performing a simple first-order Taylor series expansion. Alternatively, MDI can be simplified by assuming that the underlying probability densities belong to a pre-defined family. For *auto-regressive sources*, it reduces to the *Itakura–Saito* distance that has received significant attention in the speech literature. In the Gaussian case, further assumption of orthonormal covariance matrices leads to the *quadratic*

distortion (QD) measure. The next possible simplification is to assume that all classes share the same covariance matrix, leading to the *Mahalanobis distortion* (MD). Finally, assuming identity covariances results in the *Euclidean distortion* (ED) measure. We next discuss in more detail all these relationships.

3.1.1 Bhattacharyya Distance

If there are only two classes in the classification problem, (2.2) can be written as [39]

$$\begin{aligned} L^* &= E_{\mathbf{X}}[\min(P_{Y|\mathbf{X}}(0|\mathbf{X}), P_{Y|\mathbf{X}}(1|\mathbf{X}))] \\ &= \int P_{\mathbf{X}}(\mathbf{x}) \min[P_{Y|\mathbf{X}}(0|\mathbf{x}), P_{Y|\mathbf{X}}(1|\mathbf{x})] d\mathbf{x} \\ &= \int \min[P_{\mathbf{X}|Y}(\mathbf{x}|0)P_Y(0), P_{\mathbf{X}|Y}(\mathbf{x}|1)P_Y(1)] d\mathbf{x} \\ &\leq \sqrt{P_Y(0)P_Y(1)} \int \sqrt{P_{\mathbf{X}|Y}(\mathbf{x}|0)P_{\mathbf{X}|Y}(\mathbf{x}|1)} d\mathbf{x}, \end{aligned}$$

where we have used the bound $\min[a, b] \leq \sqrt{ab}$. The last integral is usually known as the Bhattacharyya distance between $P_{\mathbf{X}|Y}(\mathbf{x}|0)$ and $P_{\mathbf{X}|Y}(\mathbf{x}|1)$ and has been proposed (e.g., [20, 77]) for image retrieval where, for a query density $P_{\mathbf{X}}(\mathbf{x})$, it takes the form

$$g(\mathbf{x}) = \arg\min_i \int \sqrt{P_{\mathbf{X}}(\mathbf{x}) P_{\mathbf{X}|Y}(\mathbf{x}|i)} d\mathbf{x}. \tag{3.1}$$

The resulting classifier can thus be seen as the one which finds the lowest upper-bound on the Bayes error for the collection of two-class problems involving the query and each of the database classes.

3.1.2 Maximum Likelihood

It is straightforward to see that when all image classes are equally likely a priori, $P_Y(i) = 1/M$, the application of (2.3) to a sample $\mathcal{D} = \{\mathbf{x_1}, \ldots \mathbf{x}_N\}$ of independent observations reduces to the maximum likelihood classifier

$$g(\mathcal{D}) = \arg\max_i \frac{1}{N} \sum_{j=1}^{N} \log P_{\mathbf{X}|Y}(\mathbf{x}_j|i). \tag{3.2}$$

While class priors $P_Y(i)$ can be used to (1) account for the context in which the retrieval operation takes place, (2) integrate information from multiple content modalities that may be available in the database [121], and (3) design algorithms for learning from user feedback [127, 128], in this work we assume that there is no a priori reason to prefer any given image over the rest. In this case, MPE and maximum likelihood retrieval are equivalent and we will use the two terms indiscriminately.

3.1.3 Minimum Discrimination Information

If $H_i, i = 1, 2$, are the hypotheses that $\mathbf{x}$ is drawn from the statistical population with density $P_i(\mathbf{x})$, the KL divergence

$$\mathrm{KL}[P_2(\mathbf{x})||P_1(\mathbf{x})] = \int P_2(\mathbf{x}) \log \frac{P_2(\mathbf{x})}{P_1(\mathbf{x})} d\mathbf{x} \tag{3.3}$$

measures the mean information per observation from $P_2(\mathbf{x})$ for discrimination in favor of H_2 against H_1. Because it measures the difficulty of discriminating between the two populations, and is (1) non-negative and (2) equal to zero when $P_1(\mathbf{x}) = P_2(\mathbf{x}), \forall \mathbf{x}$ [58], the KLD has been proposed as a measure of similarity for various compression and signal processing problems [23, 32, 33, 44, 62].

Given a density $P_1(\mathbf{x})$ and a family of densities $\mathcal{M}$, the minimum discrimination information criteria [58] seeks the density in $\mathcal{M}$ that is the "nearest neighbor" of $P_1(\mathbf{x})$ in the KLD sense

$$P_2^*(\mathbf{x}) = \arg \min_{P_2(\mathbf{x}) \in \mathcal{M}} KL[P_2(\mathbf{x})||P_1(\mathbf{x})].$$

If $\mathcal{M}$ is a large family, containing $P_1(\mathbf{x})$, this problem has the trivial solution $P_2(\mathbf{x}) = P_1(\mathbf{x})$, which is not always the most interesting. In other cases, a sample from $P_2(\mathbf{x})$ is available but the explicit form of the distribution is not known. In these situations, it may be more useful to seek for the distribution that minimizes the KLD subject to a stricter set of constraints. Kullback suggested the problem

$$P_2^*(\mathbf{x}) = \arg \min_{P_2(\mathbf{x}) \in \mathcal{M}} KL[P_2(\mathbf{x})||P_1(\mathbf{x})]$$

subject to

$$\int T(\mathbf{x}) P_2(\mathbf{x}) = \theta,$$

where $T(\mathbf{x})$ is a measurable statistic (e.g., the mean when $T(\mathbf{x}) = \mathbf{x}$) and θ can be computed from a sample (e.g., the sample mean). He showed that the minimum is (1) achieved by

$$P_2^*(\mathbf{x}) = \frac{1}{Z} e^{-\lambda T(\mathbf{x})} P_1(\mathbf{x}),$$

where Z is a normalizing constant, $Z = \int e^{-\lambda T(\mathbf{x})} P_1(\mathbf{x}) d\mathbf{x}$, and λ a Lagrange multiplier [10] that weighs the importance of the constraint; and (2) equal to

$$KL[P_2^*(\mathbf{x})||P_1(\mathbf{x})] = -\lambda\theta - \log Z.$$

Gray et al. have studied extensively the case in which $P_1(\mathbf{x})$ belongs to the family of *auto-regressive moving average* (ARMA) processes [33, 44] and showed, among other things, that in this case the optimal solution is a variation of the Itakura–Saito distance commonly used in speech analysis and compression. Kupperman [58, 59] has shown that when all densities are members of the exponential family (a family that includes many of the common distributions of interest such as the Gaussian, Poisson, binomial, Rayleigh, and exponential among others [30]), the constrained version of MDI is equivalent to maximum likelihood.

The KLD has only been recently considered in the retrieval literature [26, 53, 89, 126, 129], where attention has focused on the unconstrained MDI problem

$$g(\mathbf{x}) = \arg\min_i KL[P_{\mathbf{X}}(\mathbf{x})||P_{\mathbf{X}|Y}(\mathbf{x}|i)] \tag{3.4}$$

where $P_{\mathbf{X}}(\mathbf{x})$ is the density of the query and $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ that of the ith image class. Similarly to the constrained case, it is possible to derive a connection between unconstrained MDI and maximum likelihood. However, the connection is much stronger in the unconstrained case since there is no need to make any assumptions regarding the type of densities involved. In particular, by simple application of the strong law of large numbers to (3.2), as $N \to \infty$ $g(\mathcal{D})$ converges almost surely to

$$\begin{aligned} g(\mathbf{X}) &= \arg\max_i E_{\mathbf{X}}[\log P_{\mathbf{X}|Y}(\mathbf{X}|i)] \\ &= \arg\max_i \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}|Y}(\mathbf{x}|i) d\mathbf{x} \end{aligned}$$

$$
\begin{aligned}
&= \arg\min_i \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} - \int P_{\mathbf{X}}(\mathbf{x}) \log P_{\mathbf{X}|Y}(\mathbf{x}|i) d\mathbf{x} \\
&= \arg\min_i \int P_{\mathbf{X}}(\mathbf{x}) \log \frac{P_{\mathbf{X}}(\mathbf{x})}{P_{\mathbf{X}|Y}(\mathbf{x}|i)} d\mathbf{x} \\
&= \arg\min_i KL[P_{\mathbf{X}}(\mathbf{x}) || P_{\mathbf{X}|Y}(\mathbf{x}|i)], \qquad (3.5)
\end{aligned}
$$

where $E_{\mathbf{X}}$ is the expectation with respect to the query density $P_{\mathbf{X}}$. This means that, independent of the type of densities, MDI is simply the asymptotic limit of the ML criteria as the cardinality of the query tends to infinity.[1] This relationship confirms that the MPE criteria converges to a meaningful global similarity function as the cardinality of the query grows. It also establishes a connection between MPE retrieval and several similarity functions that can be derived from MDI.

3.1.4 χ^2 Distance

The first of such similarity functions is the χ^2 statistic. Using a first-order Taylor series approximation for the logarithmic function about $x = 1$, $\log(x) \approx x - 1$, we obtain

$$
\begin{aligned}
KL[P_1(\mathbf{x}) || P_2(\mathbf{x})] &= \int P_1(\mathbf{x}) \log \frac{P_1(\mathbf{x})}{P_2(\mathbf{x})} d\mathbf{x} \\
&\approx \int \frac{P_1(\mathbf{x})^2 - P_1(\mathbf{x}) P_2(\mathbf{x})}{P_2(\mathbf{x})} d\mathbf{x} \\
&= \int \left(\frac{P_1(\mathbf{x})^2 - P_1(\mathbf{x}) P_2(\mathbf{x})}{P_2(\mathbf{x})} - P_1(\mathbf{x}) + P_2(\mathbf{x}) \right) d\mathbf{x} \\
&= \int \frac{(P_1(\mathbf{x}) - P_2(\mathbf{x}))^2}{P_2(\mathbf{x})} d\mathbf{x},
\end{aligned}
$$

where we have used the fact that $\int P_i(\mathbf{x}) d\mathbf{x} = 1, i = 1, 2$. In the retrieval context, this means that MDI can be approximated by

$$
g(\mathbf{x}) \approx \arg\min_i \int \frac{(P_{\mathbf{X}}(\mathbf{x}) - P_{\mathbf{X}|Y}(\mathbf{x}|i))^2}{P_{\mathbf{X}|Y}(\mathbf{x}|i)} d\mathbf{x}. \qquad (3.6)
$$

[1] Notice that this result only holds when the true distribution is that of the query. The alternative version of the divergence, where the distribution of the database image class is assumed to be true, does not have an interpretation as the asymptotic limit of a local metric of similarity.

The integral on the right is known as the χ^2 statistic and the resulting criteria a χ^2 test [84]. It has been proposed as a metric for image similarity in [26, 89, 100]. Since it results from the linearization of the KLD, it can be seen as an approximation to the asymptotic limit of the ML criteria. Obviously, this linearization can discard a significant amount of information.

3.2 The Gaussian Case

Several similarity functions of practical interest can be derived from the MPE retrieval criteria when the class likelihoods are assumed to be Gaussian with full rank covariance matrices. We now analyze the relationships for three such functions: quadratic, Mahalanobis, and Euclidean. Given the asymptotic convergence of ML to MDI, these results could also been derived from the expression for the KLD between two Gaussians [58], by replacing expectations with respect to the query distribution by sample means.

3.2.1 Quadratic Distortion

When the image features are Gaussian distributed, (3.2) becomes

$$\begin{aligned} g(\mathbf{x}) &= \arg\min_i \log|\mathbf{\Sigma}_i| + \frac{1}{N}\sum_n (\mathbf{x}_n - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}_n - \mu_i) \\ &= \arg\min_i \log|\mathbf{\Sigma}_i| + \hat{\mathcal{L}}_i, \end{aligned} \tag{3.7}$$

where

$$\hat{\mathcal{L}}_i = \frac{1}{N}\sum_n (\mathbf{x}_n - \mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}_n - \mu_i)$$

is the *quadratic distortion* (QD) measure commonly found in the perceptually weighted compression literature [40, 63, 66, 81] and quadratic discriminant analysis [30]. As a similarity measure, the QD can thus be seen as the result of imposing two stringent restrictions on the generic ML criteria. First, that all image sources are Gaussian and, second, that their covariance matrices are orthonormal ($|\mathbf{\Sigma}_i| = 1, \forall i$).

3.2.2 Mahalanobis Distortion

Furthermore, because

$$\begin{aligned}
\hat{\mathcal{L}}_i &= \frac{1}{N}\sum_n (\mathbf{x}_n - \mu_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_n - \mu_i) \\
&= \frac{1}{N}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}} + \hat{\mathbf{x}} - \mu_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_n - \hat{\mathbf{x}} + \hat{\mathbf{x}} - \mu_i) \\
&= \frac{1}{N}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}})^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_n - \hat{\mathbf{x}}) - 2(\hat{\mathbf{x}} - \mu_i)^T \boldsymbol{\Sigma}_i^{-1} \frac{1}{N}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}}) \\
&\quad + (\hat{\mathbf{x}} - \mu_i)^T \boldsymbol{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T \\
&= \frac{1}{N}\mathrm{trace}[\boldsymbol{\Sigma}_i^{-1}\sum_n (\mathbf{x}_n - \hat{\mathbf{x}})(\mathbf{x}_n - \hat{\mathbf{x}})^T] + (\hat{\mathbf{x}} - \mu_i)^T \boldsymbol{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T \\
&= \mathrm{trace}[\boldsymbol{\Sigma}_i^{-1}\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}] + (\hat{\mathbf{x}} - \mu_i)^T \boldsymbol{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T \\
&= \mathrm{trace}[\boldsymbol{\Sigma}_i^{-1}\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}] + \mathcal{M}_i, \qquad (3.8)
\end{aligned}$$

where

$$\hat{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n$$

is the sample mean of $\mathbf{x}_n$,

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \hat{\mathbf{x}})(\mathbf{x}_n - \hat{\mathbf{x}})^T$$

the sample covariance, and

$$\mathcal{M}_i = (\hat{\mathbf{x}} - \mu_i)^T \boldsymbol{\Sigma}_i^{-1}(\hat{\mathbf{x}} - \mu_i)^T$$

the Mahalanobis distortion (MD, squared Mahalanobis distance), we see that the MD results from complementing Gaussianity with the assumption that all classes have the same covariance ($\boldsymbol{\Sigma}_{\mathbf{x}} = \boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}, \forall i$).

3.2.3 Euclidean Distortion

Finally, if this covariance is the identity ($\boldsymbol{\Sigma} = \mathbf{I}$), we obtain the Euclidean distortion (ED, squared Euclidean distance)

$$\mathcal{E}_i = (\hat{\mathbf{x}} - \mu_i)^T (\hat{\mathbf{x}} - \mu_i). \qquad (3.9)$$

The MD, the ED, and variations on both, have been widely used in the retrieval literature [4, 7, 18, 25, 47, 54, 69, 69, 73, 80, 86, 87, 89, 97, 99, 101, 103, 106, 108, 113, 130, 131].

3.2.4 Some Intuition for the Advantages of MPE Retrieval

The Gaussian case is a good example of why, even if minimization of error probability is not considered to be the right goal for an image retrieval system, there seems to be little justification to rely on any criteria for image similarity other than MPE. Recall that, under MPE retrieval, the similarity function is

$$g(\mathbf{x}) = \arg\min_i \log|\boldsymbol{\Sigma}_i| + \overbrace{\mathrm{trace}[\boldsymbol{\Sigma}_i^{-1}\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}] + \underbrace{(\hat{\mathbf{x}}-\mu_i)^T\boldsymbol{\Sigma}_i^{-1}(\hat{\mathbf{x}}-\mu_i)}_{\mathrm{MD}}}^{\mathrm{QD}} \tag{3.10}$$

and all three other criteria are approximations that arbitrarily discard covariance information.

As illustrated by Figure 3.2, this information is important for the detection of subtle variations, such as rotation and scaling in feature space. In (a) and (b), we show the distance, under both QD and MD, between a Gaussian and a replica rotated by $\theta \in [0, \pi]$. Plot b) clearly illustrates that, while the MD has no ability to distinguish between the rotated Gaussians, the inclusion of the trace$[\boldsymbol{\Sigma}_i^{-1}\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}]$ term leads to a much more intuitive measure of similarity: minimum when both Gaussians are aligned and maximum when they are rotated by $\pi/2$.

As illustrated by (c) and (d), further inclusion of the term $\log|\boldsymbol{\Sigma}_i|$ (full ML retrieval) penalizes mismatches in scaling. In plot c), we show two Gaussians with covariances $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbf{I}$ and $\boldsymbol{\Sigma}_i = \sigma^2\mathbf{I}$, centered on zero. In this example, MD is always zero, while trace$[\boldsymbol{\Sigma}_i^{-1}\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}] \propto 1/\sigma^2$ penalizes small σ and $\log|\boldsymbol{\Sigma}_i| \propto \log\sigma^2$ penalizes large σ. The total distance is shown as a function of $\log\sigma^2$ in plot d) where, once again, we observe an intuitive behavior: the penalty is minimal when both Gaussians have the same scale ($\log\sigma^2 = 0$), increasing monotonically with the amount of scale mismatch. Notice that, if the $\log|\boldsymbol{\Sigma}_i|$ term is not included, large changes in scale may not be penalized at all.

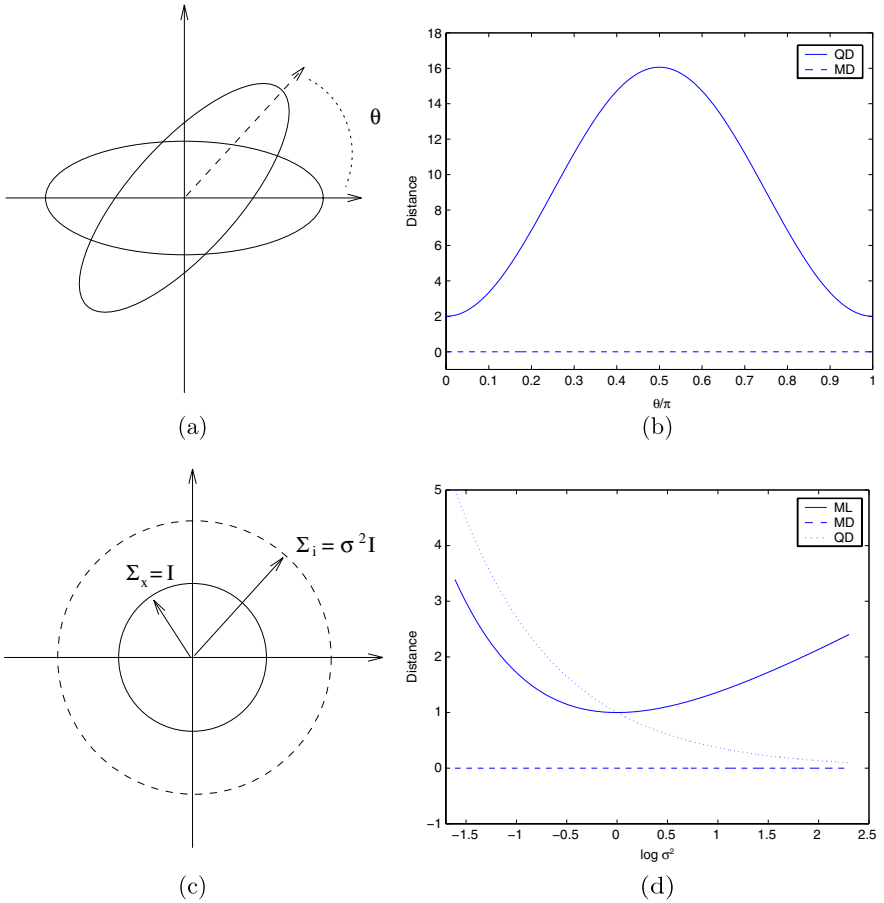


Fig. 3.2 (a) A Gaussian with mean $(0,0)^T$ and covariance $\mathrm{diag}(4, 0.25)$ and its replica rotated by θ. (b) Distance between the Gaussian and its rotated replicas as a function of θ/π under both the QD and the MD. (c) Two Gaussians with different scales. (d) Distance between them as a function of $\log \sigma^2$ under ML, QD, and MD.

3.2.5 L^p Norms

A popular metric of similarity is the L^p norm of the difference between densities

$$g(\mathbf{X}) = \arg\min_i \left(\int_{\mathcal{F}} |P_{\mathbf{X}}(\mathbf{x}) - P_{\mathbf{X}|Y}(\mathbf{x}|i)|^p d\mathbf{x} \right)^{\frac{1}{p}}, \quad p \geq 1. \tag{3.11}$$

These norms are particularly common as metrics of similarity between histograms. Consider a partition of the feature space $\mathcal{X}$ (with some sensible tie braking rule) into a collection of p disjoint cells $\{\mathcal{R}_1, \ldots, \mathcal{R}_p\}$ of representative vectors $\mathbf{c}_i \in \mathbb{R}^n$, and $\mathcal{D}$ a set of features vectors such that f_r vectors land on cell $\mathcal{R}_r$. Let $\mathbf{f} = \{f_1, \ldots, f_p\}$ be the histogram associated with the density

$$P_{\mathbf{X}}(\mathbf{x}) = \sum_k \frac{f_k}{\sum_i f_i} \delta_k(\mathbf{x} - \mathbf{c}_k), \tag{3.12}$$

where $\delta_k(\mathbf{x} - \mathbf{c}_k)$ is some probability density function supported in $\mathcal{R}_k$.

Defining $\mathbf{q}$ to be the histogram of Q query vectors, $\mathbf{p}^i$ the histogram of P^i vectors from the ith image class, and substituting (3.12) into (3.11)

$$\begin{aligned}
g(\mathbf{x}) &= \arg\min_i \left(\int_{\mathcal{X}} \left| \sum_r \left(\frac{q_r}{\sum_k q_k} - \frac{p_r^i}{\sum_k p_k^i} \right) \delta_r(\mathbf{x} - \mathbf{c}_r) \right|^p d\mathbf{x} \right)^{\frac{1}{p}} \\
&= \arg\min_i \left(\int_{\mathcal{X}} \sum_r \left| \frac{q_r}{\sum_k q_k} - \frac{p_r^i}{\sum_k p_k^i} \right|^p \delta_r^p(\mathbf{x} - \mathbf{c}_r) d\mathbf{x} \right)^{\frac{1}{p}} \\
&= \arg\min_i \left(\sum_r \left| \frac{q_r}{\sum_k q_k} - \frac{p_r^i}{\sum_k p_k^i} \right|^p \int_{\mathcal{R}_r} \delta_r^p(\mathbf{x} - \mathbf{c}_r) d\mathbf{x} \right)^{\frac{1}{p}} \\
&= \arg\min_i \left(\sum_r \omega_r \left| \frac{q_r}{\sum_k q_k} - \frac{p_r^i}{\sum_k p_k^i} \right|^p \right)^{\frac{1}{p}}
\end{aligned} \tag{3.13}$$

where we have used the fact that the cells $\mathcal{R}_r$ are disjoint and

$$\omega_r = \int_{\mathcal{R}_r} \delta_r^p(\mathbf{x} - \mathbf{c}_r) d\mathbf{x}.$$

As shown in [112], the minimization of the L^1 distance is equivalent to the maximization of the *histogram intersection* (HI)

$$g(\mathbf{x}) = \arg\max_i \frac{\sum_r \min(q_r, p_r^i)}{\sum_k q_k}, \tag{3.14}$$

a similarity function that has become the de-facto standard for color-based retrieval [4, 14, 52, 54, 69, 83, 85, 89, 96, 97, 106, 107, 109, 110, 112].

While (3.2) minimizes the classification error, measures such as the HI minimize pointwise dissimilarity between density estimates. Clearly, for this criterion to work, it is necessary that the estimates be close to the true densities. However, it is known (e.g., see Theorem 6.5 of [28]) that the probability of error of rules of the type of (3.2) tends to the Bayes error orders of magnitude faster than the associated density estimates tend to the right distributions. This implies that accurate density estimates are not required everywhere for the classification criteria to work. In fact, accuracy is required only in the regions near the boundaries between the different classes because these are the only regions that matter for the classification decisions. On the other hand, HI is dependent on the quality of the density estimates all over $\mathcal{X}$. It, therefore, places a much more stringent requirement on the quality of these estimates and, since density estimation is know to be a difficult problem [118, 104], there seems to be no reason to expect it to be a better retrieval criterion than (3.2). We next provide some experimental evidence for this claim, through retrieval experiments on real image databases.

3.3 Experimental Evaluation

A series of retrieval experiments was conducted to evaluate the performance of the ML criteria. As benchmarks, we selected two popular representatives of the similarity functions discussed above: the Mahalanobis distortion for texture-based retrieval and the histogram intersection for color-based retrieval. In order to isolate the contribution of the similarity function from those of the features and the feature representation, the comparison was performed with the feature sets and representations that are commonly used for each of the domains: color-based retrieval was implemented by combining the color histogram with (3.2) and texture-based retrieval by combining the features derived from the *multi-resolution simultaneous auto-regressive* (MRSAR) model [74] with (3.10). Texture retrieval experiments were performed on the Brodatz texture database [87], while color-based retrieval was evaluated on the Columbia object image library [79].

The MRSAR features were computed using a window of size 21×21 sliding over the image with increments of two pixels in both the horizontal and vertical dimensions. Each feature vector consists of four SAR parameters plus the error of the fit achieved by the SAR model at three resolutions, in a total of 15 dimensions. This is a standard implementation of this model [67, 73, 74]. For color histogramming, the 3D YBR color space was quantized by finding the bounding box for all the points in the query and retrieval databases, and then dividing each axis in b bins. This leads to b^3 cells. Experiments were performed with different values of b.

To evaluate the retrieval performance, we relied on standard precision/recall curves. In all the databases considered there is clear ground truth regarding which images are relevant to a given query (e.g., different views of the same object on Columbia) and we used it to measure precision and recall. Each database was split into training and test sets, the images in the test set serving as queries for performance evaluation. We refer to this set as the *query database*. The remaining images composed the *retrieval database*. The specific organization of the databases was as follows. The 1008 images in Brodatz were divided into a query database of 112, and a retrieval database of 896 images. The Columbia database was also split into a query containing a single view of each of the 100 objects available, and a retrieval database containing nine views (separated by 40°) of each object.

Figure 3.3 presents precision/recall curves for the two databases. As expected, texture-based retrieval (MRSAR/MD) performs better on Brodatz, while color-based retrieval (color histogramming) does better on Columbia. Furthermore, due to their lack of spatial support, histograms do poorly on Brodatz while, being a model specific for texture, MRSAR does poorly on Columbia.[2]

More informative is the fact that, when the best performing features and representation are used for the specific database, the ML criteria always leads to a clear improvement in retrieval performance. In particular, for the texture database, combining ML with the MRSAR features

[2] Notice that this would not be evident if we were only looking at classification accuracy, i.e., the percentage of retrievals for which the first match is from the correct class.

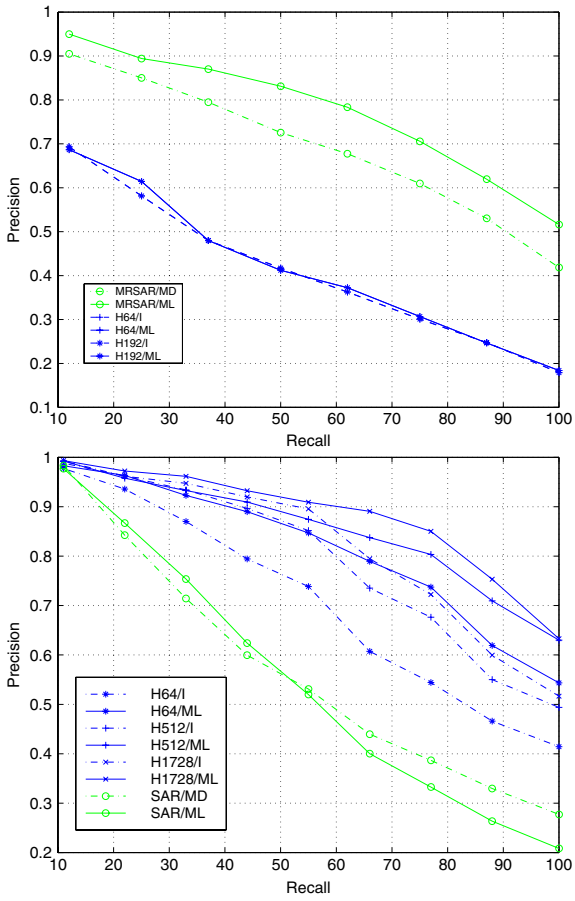


Fig. 3.3 Precision/recall curves for Brodatz (top) and Columbia (bottom). MRSAR, MRSAR features; H, histograms; ML, maximum likelihood; MD, Mahalanobis distortion; and I, intersection. The total number of bins in each histogram is indicated after the H.

and the Gaussian representation leads to an improvement in precision from 5% to 10% (depending on the level of recall) over that achievable with the Mahalanobis distortion. Similarly, on Columbia, replacing histogram intersection by the ML criteria leads to an improvement that can be as high as 20%.[3]

[3] Notice that, for these databases, 100% recall means retrieving the eight or nine images in the same class as the query, and it is important to achieve high precision at this level. This may not be the case for databases with hundreds of images in each class, since it is unlikely that users may want to look at that many images.

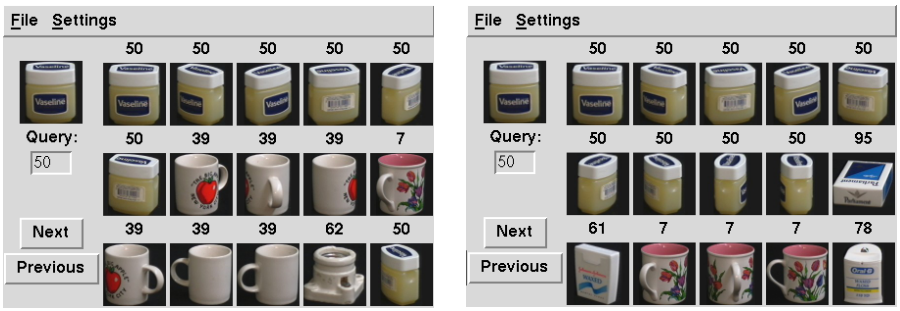


Fig. 3.4 Results for the same query under HI (left) and ML (right). In both images, the query is shown in the top left corner, and the returned images in raster-scan order (left to right, top to bottom) according to their similarity rank. The numbers displayed above the retrieved images indicate the class to which they belong.

The latter observation provides evidence in favor of the arguments of Section 3.2.5, where we argued that, while the ML criteria only depends on the class boundaries, HI requires good estimates throughout the feature space. This also suggests that, whenever there is a change in the imaging parameters (lighting, shadows, object rotation, etc.) and the densities change slightly, the impact on HI should be higher than on ML. An example of this behavior is given in Figure 3.4, where we present the results of the same query under the two similarity criteria. Notice that, as the object is rotated, the relative percentages of the different colors in the image change. HI changes accordingly and, when the degree of rotation is significant, views of other objects are preferred. On the other hand, because the color of each individual pixel is always better explained by the density of the rotated object than by those of other objects, ML achieves a perfect retrieval. This increased invariance to changes in imaging conditions explains why, for large recall, the precision of ML is consistently and significantly higher than that of HI.

4

An MPE Architecture for Image Retrieval

In this section, we build on the discussion above to design an MPE architecture for image retrieval based on QBVE. The general form of this architecture is illustrated in Figure 1.1. Image i in the database is represented as a bag of features $\mathcal{D}^i = \{\mathbf{x}_1^i, \ldots, \mathbf{x}_n^i\}$, which is used to learn an estimate $\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)$ of the feature distribution $P_{\mathbf{X}|Y}(\mathbf{x}|i)$ for that image. Given a query image, represented by the bag $\mathcal{Q} = \{\mathbf{x}_1^q, \ldots, \mathbf{x}_n^q n\}$, the closest match is found by evaluating the log-probability of the query bag under all database models, and choosing the database image for which this log-probability is largest,

$$g(\mathcal{Q}) = \arg\max_i \sum_k \log \hat{p}_{\mathbf{X}|Y}(\mathbf{x}_k^q|i). \tag{4.1}$$

In this section, we discuss the implementation of the density estimation and feature selection modules.

4.1 Density Estimation

We start by considering the problem of density estimation. Density estimates are usually obtained by choosing a parametric probability density function and learning its parameters from a training sample.

Algorithm 1 EM algorithm (Gaussian mixtures)

Input: training set $\mathcal{D} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, and initial mixture parameters $\{\pi_1, \ldots, \pi_p\}$, $\mu_1, \ldots, \mu_p$, $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_p$.

repeat

E-step: for each $\mathbf{z}_k \in \mathcal{D}$ and mixture component c, compute

$$h_{ck} = P_{\Omega|\mathbf{Z}}(c|\mathbf{z}_k) = \frac{\mathcal{G}(\mathbf{z}_k, \mu_c, \mathbf{\Sigma}_c)\pi_c}{\sum_l \mathcal{G}(\mathbf{z}_k, \mu_l, \mathbf{\Sigma}_l)\pi_l} \tag{4.2}$$

M-step: update the mixture parameters with

$$\pi_c^{\text{new}} = \frac{1}{N}\sum_k h_{ck} \tag{4.3}$$

$$\mu_c^{\text{new}} = \frac{\sum_k h_{ck}\mathbf{z}_k}{\sum_k h_{ck}} \tag{4.4}$$

$$\mathbf{\Sigma}_c^{\text{new}} = \frac{\sum_k h_{ck}(\mathbf{z}_k - \mu_c^{new})(\mathbf{z}_k - \mu_c^{new})^T}{\sum_k h_{ck}} \tag{4.5}$$

until convergence

In this monograph, we consider mixture densities [11, 95, 114]

$$P_{\mathbf{Z}}(\mathbf{z}) = \sum_{c=1}^{C} P_{\mathbf{Z}|\Omega}(\mathbf{z}|c)P_{\Omega}(c), \tag{4.6}$$

where C is a number of mixture components, $\{P_{\mathbf{Z}|\Omega}(\mathbf{z}|c)\}_{c=1}^{C}$ a sequence of *mixture components*, and $\{P_{\Omega}(c)\}_{c=1}^{C}$ a sequence of *component probabilities*. These densities model processes with hidden structure: one among the C components is first selected, according to the probabilities $P_{\Omega}(c)$, and observations are then drawn from the respective mixture component. These components can be any valid probability density functions, i.e., any set of non-negative functions integrating to one. Most densities of practical interest can be well approximated by mixtures with a relatively small number of components.

Mixture parameters are learned by maximum likelihood, using the *expectation-maximization* (EM) algorithm [27, 46, 95]. This iterates between two steps, an expectation step and a maximization step. The

expectation step computes the expectation of the data likelihood with respect to the hidden variable Ω. The maximization step then finds the parameters of the model that maximize this expected data likelihood. The precise equations of the two steps depend on the details of the mixture components $P_{\mathbf{X}|\Omega}(\mathbf{x}|c)$ in (4.6). Algorithm 1 summarizes the EM steps for learning the parameters of a Gaussian mixture

$$P_{\mathbf{Z}}(\mathbf{z}) = \sum_c \pi_c \mathcal{G}(\mathbf{z}, \mu_c, \mathbf{\Sigma}_c) \tag{4.7}$$

of class probabilities $P_Y(c) = \pi_c$, and Gaussian components of mean μ_c and covariance $\mathbf{\Sigma}_c$. This is the model that we will use in the remainder of this work.

One interesting property of this model is that it supports *parallel* evaluation of multiple MPE rules over a set embedded spaces. This follows from a well known property of Gaussian random variables [30].

Property 4.1. If $\mathbf{T}: \mathcal{X} \to \mathcal{X}'$ is a linear feature transformation $\mathbf{X} \in \mathcal{X}$ and $\mathbf{X}' \in \mathcal{X}'$ are two random variables such that $\mathbf{X}$ is distributed according to

$$P_{\mathbf{X}}(\mathbf{x}) = \sum_i \lambda_i \mathcal{G}(\mathbf{x}, \mu_i, \mathbf{\Sigma}_i), \tag{4.8}$$

where $0 \leq \lambda_i \leq 1, \forall i, \sum_i \lambda_i = 1$ and $\mathbf{X}' = \mathbf{T}\mathbf{X}$, then

$$P_{\mathbf{X}'}(\mathbf{x}) = \sum_i \lambda_i \mathcal{G}(\mathbf{x}, \mathbf{T}\mu_i, \mathbf{T}\mathbf{\Sigma}_i\mathbf{T}^T). \tag{4.9}$$

Consider a linear transformation $\mathbf{T}$ and associated feature space $\mathcal{X} \subset \mathbb{R}^d$. From Theorem 2.2, the sequence $\mathcal{X}_j = \pi_i^d(\mathcal{X})$ is a sequence of embedded subspaces of $\mathcal{X}$. Denoting by $\mathbf{\Pi}_j$ the projection matrix associated with π_j^d, i.e., $\mathbf{\Pi}_j = [\mathbf{I}_j, \mathbf{0}_{d-j}]$, where $\mathbf{I}_j$ is the identity matrix of order j and $\mathbf{0}_{d-j}$ the $j \times d - j$ zero matrix, it follows from the property above that, if $\mathbf{X} \in \mathcal{X}$ is distributed according to (4.8), the random variables $\mathbf{X}_j = \pi_j^d(\mathbf{X})$ are distributed according to

$$P_{\mathbf{X}_j}(\mathbf{x}) = \sum_i \lambda_i \mathcal{G}(\mathbf{x}, \mathbf{\Pi}_j\mu_i, \mathbf{\Pi}_j\mathbf{\Sigma}_i\mathbf{\Pi}_j^T). \tag{4.10}$$

The collection of densities in (4.10) is the family of *embedded mixture models* associated with $\mathbf{X}$. It has two properties of significant practical interest. The first is that, once an estimate is available for $\{\lambda_i, \mu_i, \mathbf{\Sigma}_i\}$, the parameters of $P_{\mathbf{X}_j}(\mathbf{x})$ can be obtained for any j by simply extracting the first j components of the mean vectors μ_i and the upper-left $j \times j$ sub-matrix of the covariances $\mathbf{\Sigma}_i$. This implies that it is not necessary to repeat the density estimation for each of the subspace dimensions under consideration. Hence, the complexity of estimating all $P_{\mathbf{X}_j}(\mathbf{x})$ is the same as that of estimating $P_{\mathbf{X}}(\mathbf{x})$. The second is a similar result for the complexity of evaluating image queries. It is based on the fact that the complexity of (4.8) is dominated by the computation of $||\mathbf{x} - \mu||_{\mathbf{\Sigma}}$ and $|\Sigma|$.

Lemma 4.1. Consider the contribution to $P_{\mathbf{X}_j}(\mathbf{\Pi}_j\mathbf{x}), j = 1, \ldots, d$ of a mixture component with mean $\mathbf{\Pi}_j\mu$ and covariance $\mathbf{S}_j = \mathbf{\Pi}_j\mathbf{\Sigma}\mathbf{\Pi}_j^T$. The terms $\mathcal{M}_j = ||\mathbf{\Pi}_j\mathbf{x} - \mathbf{\Pi}_j\mu||_{\mathbf{S}_j}$ and $\mathcal{D}_j = |\mathbf{S}_j|$ are given by the following recursion.

Initial conditions: $\mathcal{M}_1 = (x_1 - \mu_1)^2/\sigma_{1,1}$, $\mathcal{D}_1 = \mathbf{S}_1 = \sigma_{1,1}$.

Recursion:

$$\psi_j^T = (\mathbf{u}_{j-1}^T\mathbf{S}_{j-1}^{-1}, -1) \tag{4.11}$$

$$p_j = -(\mathbf{u}_{j-1}^T, \sigma_{j,j})\,\psi_j \tag{4.12}$$

$$\mathbf{S}_j^{-1} = \Gamma(\mathbf{S}_{j-1}^{-1}) + \frac{1}{p_j}\psi_j\psi_j^T \tag{4.13}$$

$$\mathcal{M}_j = \mathcal{M}_{j-1} + \frac{(\psi_j^T\mathbf{\Pi}_j\mathbf{d})^2}{p_j} \tag{4.14}$$

$$\mathcal{D}_j = p_j\mathcal{D}_{j-1}, \tag{4.15}$$

where $\Gamma(\cdot)$ is a mapping that adds to matrix $\cdot$ a row and a column (which become the last row and column, respectively) of zeros, $\mathbf{d} = \mathbf{x} - \mu$, $\sigma_{i,j}$ is the (i,j)th element of Σ, and $\mathbf{u}_{j-1} = (\sigma_{1,j}, \ldots, \sigma_{j-1,j})^T$ the vector containing the $j-1$ first elements of the jth column of $\mathbf{\Sigma}$. The complexity of evaluating all $\mathcal{M}_j$ and $\mathcal{D}_j$ is $O(d^3)$.

Proof. See the Appendix. □

It follows from this lemma that the complexity of evaluating all $P_{\mathbf{X}_j}(\mathbf{\Pi}_j\mathbf{x})$ is $O(d^3)$ and, since this is also the cost of computing $\mathbf{\Sigma}^{-1}$, this complexity is the same as that required to compute $P_{\mathbf{X}}(\mathbf{x})$. Hence, the computations required for MPE retrieval can be performed in parallel across a collection of embedded subspaces, without any additional computational cost. This property can be explored to develop computationally efficient *feature selection* algorithms for image retrieval [124].

4.2 Embedded Multi-Resolution Mixture Models

The parallelism of embedded mixture models is particularly useful when combined with multiresolution feature transformations. Such transformations have a number of interesting properties. For example, they are justified by what is known about biological vision. Ever since the work of Hubel and Wiesel [49], it has been established that (1) human visual processing is local, and (2) different cells in primary visual cortex (i.e., area V1) are tuned for detecting different types of stimulus (e.g., bars of different size). This indicates that, at the lowest level, the architecture of the human visual system can be approximated by a multi-resolution representation localized in space and frequency, and several "biologically plausible" models of early vision are based on this principle [8, 9, 38, 70, 98, 111]. More recently, it has been shown that filters remarkably similar to the receptive fields of cells found in V1 [6, 82] can be learned from training images, by imposing requirements of sparseness [37, 82] or independence [6] to a multiresolution transformation.

A second interesting property is invariance. When the feature transform T is a multi-resolution decomposition, embedded mixture densities can be interpreted as families of densities defined over multiple image scales, each adding higher resolution information to the characterization provided by those before it. In fact, disregarding the dimensions associated with high-frequency basis functions is equivalent to modeling densities of low-pass filtered images. In the extreme case where only the first, or DC, coefficient is considered, the representation is equivalent to the histogram of a smoothed version of the original image. This is illustrated in Figure 4.1.

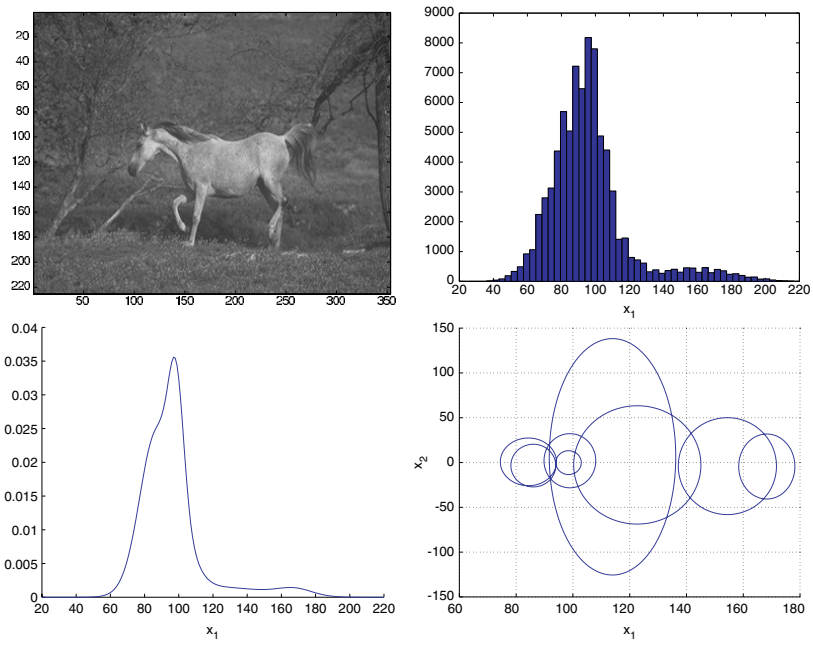


Fig. 4.1 A natural image (top left), its histogram of image intensities (top right), and projections of the corresponding 64-dimensional embedded mixture onto the DC subspace (bottom left), and the subspace of the two lower frequency coefficients (bottom right). The embedded mixture describes the probability density of the discrete cosine transform coefficients derived from a collection of 8×8 blocks extracted from the image.

This observation suggests that a natural ordering for the subspaces generated by a multiresolution decomposition is by increasing frequency of the basis functions associated with those subspaces. The resulting *embedded multi-resolution mixture* (EMM) model (embedded mixtures on a multi-resolution feature space) is a generalization of the color histogram, where the additional dimensions capture the spatial dependencies that are crucial for fine image discrimination. Figure 4.2 illustrates this point by presenting two images that have the exact same color histogram but are perceptually quite distinct. The advantage of the EMM generalization is that it enables fine control over the invariance properties of the representation. Since the histogram is approximately invariant to scaling, rotation, and translation, when only the DC subspace is considered the EMM representation is also invariant to all these

Fig. 4.2 Two images that, although visually very dissimilar, have the same color histogram.

transformations. However, by including high-frequency coefficients, it is possible to trade-off invariance for Bayes error.

4.3 Multiresolution Transforms

A number of multiresolution feature transformations can be used with MPE retrieval. In fact, it is possible to efficiently combine many transformations, so as to explicitly select the best feature subset, in the MPE sense. This is discussed in detail in [120, 124]. In this section, we simply review some of the most popular transformations that are available in the literature. The discussion is, by design, brief. More details can be found on a number of image processing textbooks [19, 50, 72].

Definition 4.1. The discrete cosine transform (DCT) [19, 55] of size n is the orthogonal transform whose basis functions are defined by:

$$A(i,j) = \alpha(i)\alpha(j)\cos\frac{(2x+1)i\pi}{2n}\cos\frac{(2y+1)j\pi}{2n}, \quad 0 \leq i,j,x,y < n, \tag{4.16}$$

where $\alpha = \sqrt{1/n}$ for $i = 0$, and $\alpha = \sqrt{2/n}$ otherwise.

The DCT is widely used in image compression, and recognition experiments have shown that DCT features can lead to recognition rates comparable to those of many features proposed in the recognition literature [121]. It is also possible to show that, for certain classes of stochastic processes, the DCT converges asymptotically to the following transform [55].

Definition 4.2. Principal components analysis (PCA) is the orthogonal transform defined by

$$\mathbf{T} = \mathbf{D}^{-1/2}\mathbf{E}^T, \tag{4.17}$$

where $\mathbf{EDE}^T$ is the eigenvector decomposition of the covariance matrix $E[\mathbf{z}\mathbf{z}^T]$.

It is well known (and straightforward to show) that PCA generates uncorrelated features, i.e., $E[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$. In this context, PCA is the optimal redundancy reduction transform, i.e., the one that produces the most parsimonious description of the input observations. For this reason, PCA has been widely used in both compression and recognition [79, 117]. An alternative multi-resolution representation is provided by wavelet decompositions.

Definition 4.3. A wavelet transform (WT) [71, 72] is the orthogonal transform whose basis functions are defined by

$$A(i,j) = \sqrt{2^{k+l}}\Psi\left(2^k x - i\right)\Psi\left(2^l y - j\right)_{(0,0)\le(i,j)<(2^k,2^l)}^{0\le k,l<\log_2 n}, \tag{4.18}$$

where $\Psi(x)$ is a function (wavelet) that integrates to zero.

Like the DCT, wavelets have been shown empirically to achieve good decorrelation. However, natural images exhibit a significant amount of higher-order dependencies that cannot be captured by orthogonal components [82]. Eliminating such dependencies is the goal of independent component analysis.

Definition 4.4. Independent component analysis (ICA) [15, 50] is a feature transform $T\colon \mathcal{Z} \to \mathcal{X}$ such that, for a general $P_{\mathbf{Z}}(\mathbf{z})$, the random variable $\mathbf{X} = (X_1, \ldots, X_d)$ from which feature vectors are drawn has independent components

$$P_{\mathbf{X}}(\mathbf{x}) = \prod_i P_{\mathbf{X}_i}(\mathbf{x}_i). \tag{4.19}$$

The exact details of ICA depend on the particular algorithm used to learn the basis from a training sample. Since independence is usually

difficult to measure and enforce if d is large, ICA techniques tend to settle for less ambitious goals. The most popular solution is to minimize a contrast function, which is guaranteed to be zero if the inputs are independent. Examples of such contrast functions are higher order correlations and information-theoretic objective functions [15]. Popular representatives from the two types are (1) the method developed by Comon [21], which uses a contrast function based on high-order cumulants, and (2) the FastICA algorithm [51], that relies on the negative entropy of the features.

Figure 4.3 presents sets of basis functions learned from a sample of 100,000 examples extracted randomly from the Brodatz texture database. The figure presents the functions learned for PCA, ICA with the method of Comon, and ICA with the FastICA algorithm, as well as the DCT basis (wavelet basis do not have block-based support and

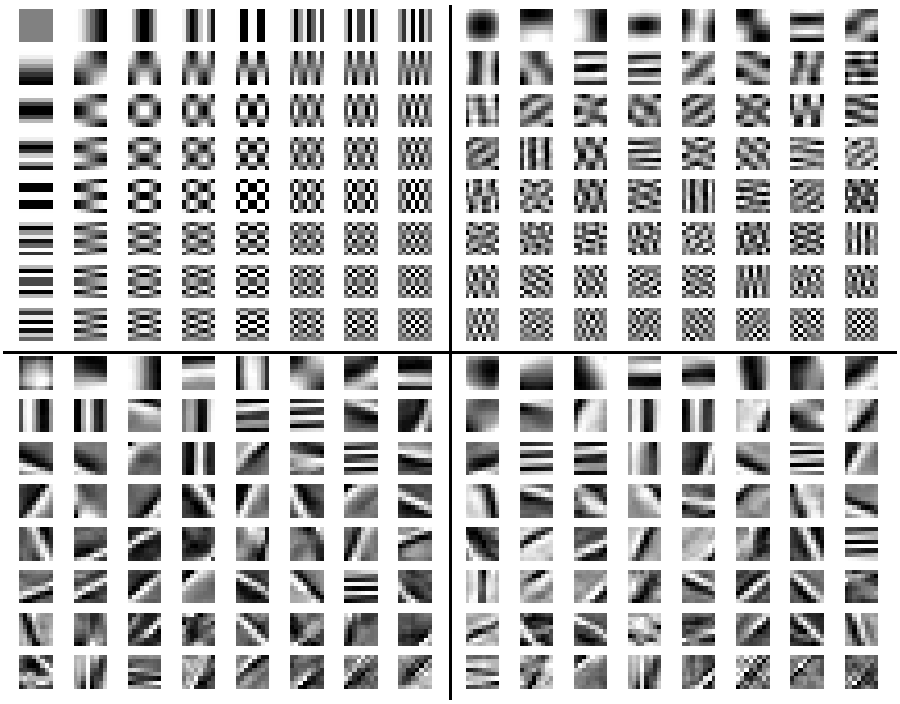


Fig. 4.3 Basis functions for DCT (top left), PCA (top right) ICA learned with Comon's method (bottom left) and ICA learned with the fastICA method (bottom right).

are not shown). In principle, any of these features can be used in the EMM representation. In Section 4.5, we present a comparison of the resulting retrieval performance.

4.4 Localized Similarity

While we have, so far, focused on the problem of similarity between entire images, which we refer to as *holistic image similarity*, a good retrieval architecture should also provide support for *localized* 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 localized 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 much more robust to occlusion, object deformation, and changes of imaging parameters. This is likely to improve retrieval accuracy even for holistic queries. Second, localized queries are much more revealing of the user's interests than global ones. Consider a retrieval system faced with the query image of Figure 4.4. 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,

Fig. 4.4 Example of a query image with multiple interpretations.

bookshelves, painting on the wall, flower baskets, white table, sofas, carpet, rooms with light painted walls) and the query is ambiguous. By allowing the user to indicate the relevant regions of the image, the ambiguity can be significantly reduced.

One of the main attractions of MPE retrieval is that it makes local queries straightforward. This follows from the fact that the optimality of (4.1) does not require the query set $\mathcal{Q}$ to have the same cardinality as the sets $\mathcal{D}_i$ used to estimate the class-conditional densities $P_{\mathbf{X}|Y}(\mathbf{x}|i)$. In fact, it is completely irrelevant if $\mathcal{Q}$ 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 MPE 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 explicit segmentation of the images in the database. The only segmentation that is required are the image regions which make up the query, and which are provided by the user himself/herself. Second, since MPE 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 localized 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.

4.5 Experiments

In this section, we present an experimental study of the retrieval architecture discussed above. In addition to the Brodatz and Columbia datasets used in the previous section, we have also conducted experiments on the Corel dataset of stock photography. While Brodatz is a texture database and color-based methods tend to do well on Columbia,

Corel contains generic imagery and requires retrieval algorithms that can account for both color and texture.

4.5.1 Experimental Set-Up

All images were normalized to the sizes 240×360 or 360×240. The image observations were 8×8 patches obtained with a sliding window moved by two pixels in a raster scan fashion (with a vertical interval of two lines), leading to a sample of about $20,000$ observations per image. Mixtures of 8 Gaussians with diagonal covariance were learned for all images with the EM algorithm [27], initialized with the generalized Lloyd algorithm [42] according to the codeword splitting procedure discussed in [43]. After learning the initial set of means, all the vectors in the training set were assigned to the closest (in the Euclidean sense) mean vector, the sample covariances resulting from this assignment were used as initial estimate for the covariances, and the relative frequencies of the assignments as initial estimates for the mixture probabilities. Each image in the retrieval database was considered as a different class.

The specific organization of the Brodatz and Columbia databases was as in the previous section. From Corel, we selected 15 concepts[1] leading to a total of 1500 images. Of these, 10% were used on the query database, leaving the remaining 90% for retrieval. In terms of benchmark techniques, we used those of the previous section: MRSAR features and the Mahalanobis distortion on Brodatz, and color histograms with HI on Columbia. In addition to the texture and color-based approaches, Corel allowed the comparison of MPE retrieval against a popular empirical approach that jointly models the two attributes: the color correlogram of [48].

4.5.2 Feature Transformation

We start with a set of results that illustrate (1) the importance of relying on a diverse dictionary of feature transforms as a means to achieve

[1] "Arabian horses", "auto racing", "coasts", "divers and diving", "English country gardens", "fireworks", "glaciers and mountains", "Mayan and Aztec ruins", "oil paintings", "owls", "land of the pyramids", "roses", "ski scenes", and "religious stained glass".

high retrieval accuracy over a diverse set of databases, and (2) how the performance of a given transform can vary significantly with both the type of database and the selected subspace dimension. For each query, we measured precision at various levels of recall. The precision/recall (PR) curves were then averaged over all queries to generate an average PR curve for each feature transform. Figure 4.5 presents the curves of precision, as a function of subspace dimension, at 30% recall on Brodatz and 10% recall on Corel (the relative precision values obtained

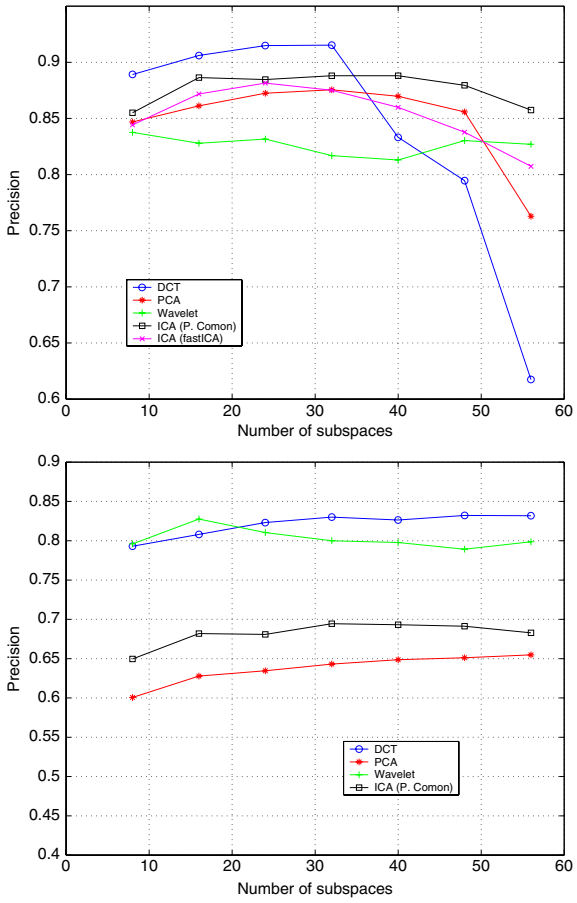


Fig. 4.5 Top: precision, at 30% recall, on Brodatz. Bottom: precision, at 10% recall, on Corel.

with the various transformations did not vary significantly with the level of recall).

The precision curves comply with the theoretical arguments of Section 2.2. Since precision is inversely proportional to the probability of error, one would expect, from those arguments, the precision curves to be concave. This is indeed the case for all transformations (there is a large increase in precision from 1 to 8 dimensions on both cases that we omit for clarity of the graph). Other than this, there are two other interesting observations. The first is that, for a given database, a poor choice of transformation can lead to significant degradation of retrieval performance. For example, the peak precision of the worst transformation (wavelet) on Brodatz is 10% below that of the best (DCT) and on Corel the variation is almost 20%. Furthermore, while the wavelet basis has the worst performance on Brodatz, it is one of the top two feature sets on Corel. On the other hand, ICA does better on Brodatz than on Corel. Second, even for a given feature transformation, precision can vary dramatically with the number of embedded subspaces. For example, the precision of the DCT features on Brodatz drops from the peak value of about 92% to about 62% when all the subspaces are included. Overall, the use of the first 32 coefficients of the DCT seems to achieve good performance across all datasets. We, therefore, adopted this value in all remaining experiments.

4.5.3 Comparison in the Texture and Color Domains

We next compared the performance of MPE retrieval with those of MRSAR and HI, in the specific databases where the latter work best: texture (Brodatz) for MRSAR and color (Columbia) for HI. Figure 4.6 presents the resulting PR curves, showing that MPE retrieval achieves equivalent performance or actually outperforms the best of the two other approaches in each image domain. This indicates that the MPE architecture performs well for both color and texture and should therefore do well on a large spectrum of databases. Visual inspection of the retrieval results suggests that, also along the dimension of perceptual relevance, MPE retrieval clearly outperforms the MRSAR

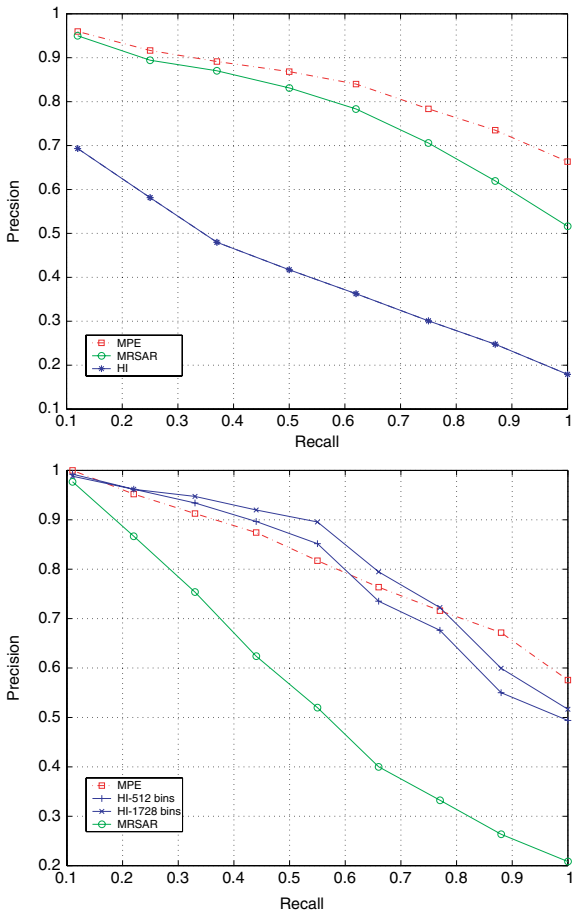


Fig. 4.6 PR measured for the MPE, MRSAR, and HI retrieval architectures. Top: curves from Brodatz, where the best results for HI (which are shown) were obtained with histograms of 192 bins. Bottom: curves from Columbia where best HI results were obtained with histograms of 1728 bins. There was, however, a wide range of number of bins where the performance was nearly constant, as illustrated by the second curve, obtained with histograms of 512 bins.

and histogram-based approaches. Figure 4.7 presents representative examples of the three of major advantages of the MPE retrieval system: (1) when it makes errors, these tend to be perceptually less disturbing than those of the other approaches, (2) when there are several visually similar classes in the database, images from these classes tend to be retrieved together, and (3) even when the performance is worse than

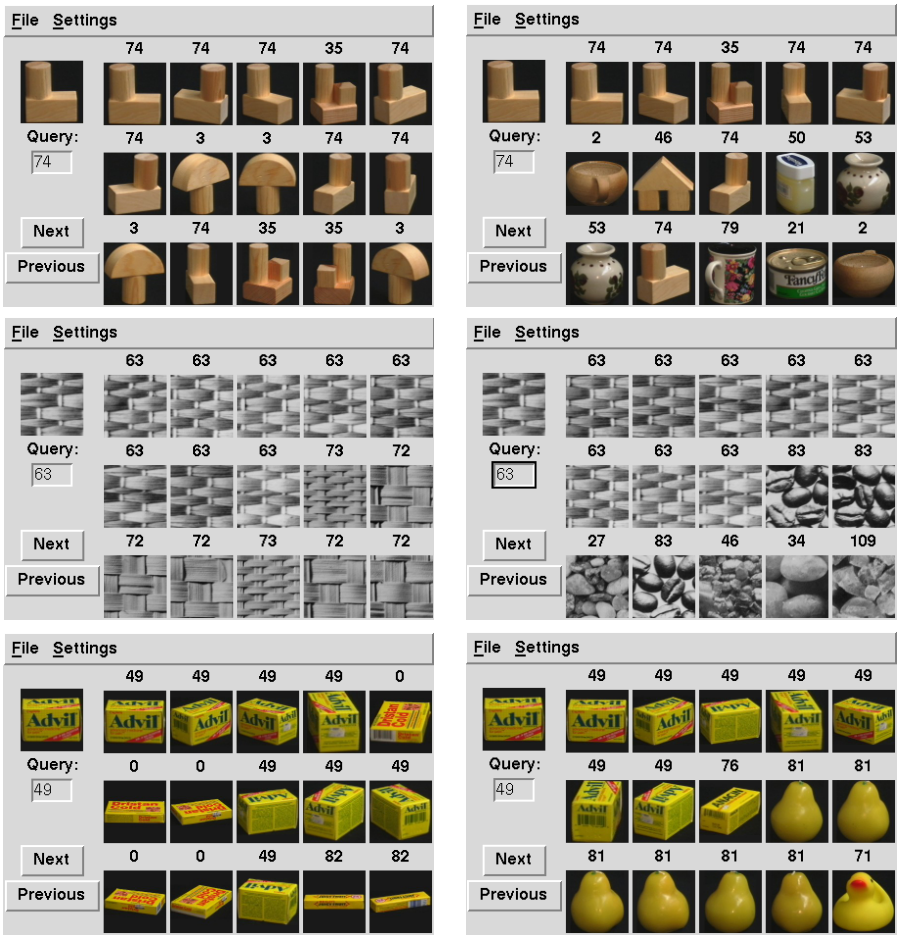


Fig. 4.7 Comparison of MPE retrieval results (left) with those of HI on Columbia and MRSAR on Brodatz (right).

that of the other approaches in terms of PR, the results are frequently better from a perceptual standpoint.

The two pictures on the top row exemplify how MPE retrieval can lead to perceptually pleasing retrieval results, even when the PR performance is only mediocre. In this case, while HI retrieves several objects unrelated to the query, MPE only returns objects that, like the query, are made of wood blocks. This is due to the fact that, by relying on features with spatial support, the embedded multiresolution

mixture representation is able to capture the local appearance of the object surface. Hence, it tends to match surfaces with the same shape, texture, and reflectance properties. This is not possible with color histograms.

The two images on the center exemplify situations where both approaches perform perfectly in terms of PR, yet the perceptual retrieval quality is very different. MRSAR ranks all the images in the query class at the top, but produces poor matches after that. On the other hand, MPE retrieves images that are visually similar to the query after all the images in its class are exhausted. This observation is frequent and derives from the fact that the MRSAR features have no perceptual justification. On the other hand, because a good match under MPE retrieval implies that the query and retrieved images should populate the space of spatial frequencies in a similar fashion, this approach tends to group images that have energy along the same orientations and a frequency spectrum with the same types of periodicities. These characteristics are known to be relevant for human judgments of similarity [67].

Finally, the pictures on the bottom row illustrate how, even when it has higher PR, HI can lead to perceptually poorer results than the MPE approach. In this case, images of a pear and a duck are retrieved by HI after the images in the right class ("Advil box"), even though there are several boxes with colors similar to those of the query in the database. On the other hand, MPE retrieval only retrieves boxes, although not in the best possible order.

4.5.4 Generic Retrieval

In addition to color and texture, we performed experiments on the Corel database, where a combination of the two cues is usually needed to evaluate image similarity. Figure 4.8 presents a comparison, in terms of PR, of MRSAR, HI, the color correlogram, and MPE retrieval. It is clear that the texture model alone performs very poorly, color histogramming does significantly better, and the correlogram further improves performance by about 5%. However, all the empirical approaches are significantly less effective than MPE retrieval.

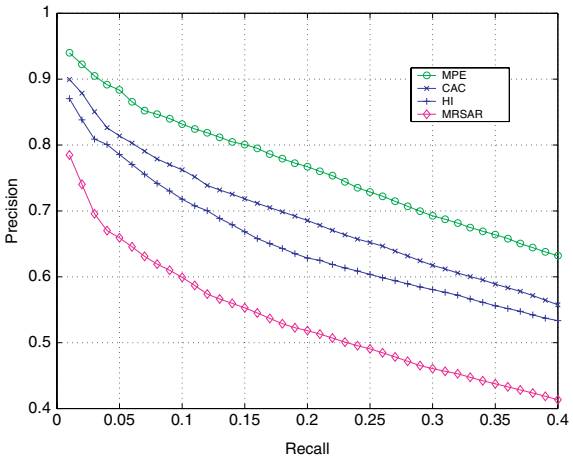


Fig. 4.8 PR on Corel for MRSAR, HI (512 bin histograms), color correlogram (CAC), and MPE retrieval. The features selected by MPE were the DCT set with 46 subspaces.

4.5.5 Localized Queries

We next considered region-based queries. For this, we started by replicating the experiments above 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%).[2] Blocks were selected starting from the center in an outward spiral fashion.

Figure 4.9 presents PR 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

[2] 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.

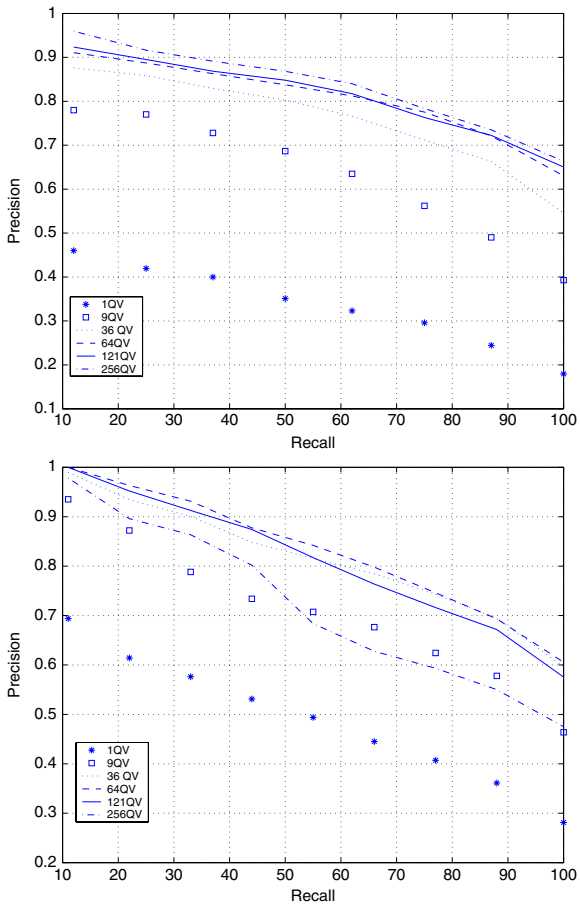


Fig. 4.9 PR curves of EMM/ML on Brodatz (top) and Columbia (bottom). X QV means that only X feature vectors from the query image were actually included in query.

due to the fact that, in Columbia, all objects appear over a common black background that can cover a substantial amount of the image area. As Figure 4.10 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.

Notice that these are two natural interpretations of similarity (prefer objects similar to the query and presented in the same pose vs. prefer

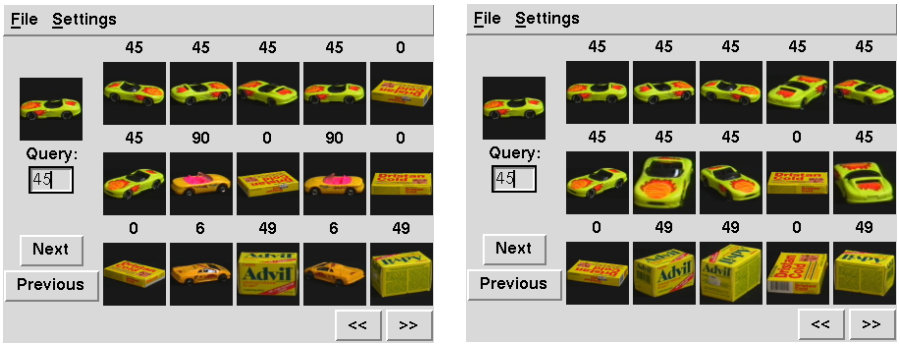


Fig. 4.10 Global similarity (left) can lead to worse precision/recall than localized similarity (right) on Columbia due to the large black background common to all objects.

the query object in different poses) and MPE 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 (localized query), this object becomes preferred independently of its pose. Obviously, PR cannot account for these types of subtleties and the former interpretation is heavily penalized. In any case, these experiments show that MPE retrieval has some robustness against missing data and can therefore handle localized queries.

5

MPE Image Annotation and Semantic Retrieval

So far, we have considered architectures for QBVE. This is not always a natural retrieval paradigm. For example, the user may not even have a good query image at hand. The natural next step is to consider the design of semantic retrieval systems. These are systems where database images are annotated with semantic labels, enabling the user to specify the query through a natural language description of the visual concepts of interest. The central problem is how to automatically extract semantic descriptors from images. We next consider this problem.

5.1 Semantic Labeling and Retrieval

Consider a database $\mathcal{T} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$ of images $\mathcal{I}_i$ and a semantic vocabulary $\mathcal{L} = \{w_1, \ldots, w_T\}$ of semantic labels w_i. The goal of semantic image annotation is to, given an image $\mathcal{I}$, extract the set of semantic labels, or caption,[1] $\mathbf{w}$ that best describes it. The goal of semantic retrieval is to, given a semantic label w_i, extract the images in the database that contain the associated visual concept. In both cases,

[1] A caption is represented by a binary vector $\mathbf{w}$ of T dimensions whose kth entry is 1 when w_k is a member of the caption and 0 otherwise.

learning is based on a training set $\mathcal{D} = \{(\mathcal{I}_1, \mathbf{w}_1), \ldots, (\mathcal{I}_D, \mathbf{w}_D)\}$ of image–caption pairs. The training set is said to be weakly labeled if (1) the absence of a semantic label from caption $\mathbf{w}_i$ does not necessarily mean that the associated concept is not present in $\mathcal{I}_i$, and (2) it is not known which image regions are associated with each label. For example, an image containing "sky" may not be explicitly annotated with that label and, when it is, no indication is available regarding which image pixels actually depict sky. Weak labeling is expected in practical retrieval scenarios since (1) each image is likely to be annotated with a small caption that only identifies the semantics deemed as most relevant to the labeler, and (2) users are rarely willing to manually annotate image regions.

Under the MPE criteria, image labeling is defined as a multiclass classification problem. The classes are the elements of the semantic vocabulary $\mathcal{L}$, which compete for the image to label at annotation time. This form of *supervised multiclass labeling* (SML) requires the introduction of a random variable W, which takes values in $\{1, \ldots, T\}$, so that $W = i$ if and only if $\mathbf{x}$ is a sample from concept w_i. Each class is characterized by a class-conditional distribution $P_{\mathbf{X}|W}(\mathbf{x}|i)$. Given a set of feature vectors $\mathcal{F}$ extracted from a (previously unseen) test image $\mathcal{I}$, the image's MPE label is

$$i^*(\mathcal{F}) = \arg\max_i P_{W|\mathbf{X}}(i|\mathcal{F}). \tag{5.1}$$

Similarly, given a query concept w_i, MPE semantic retrieval consists of returning the database image of index

$$j^*(w_i) = \arg\max_j P_{\mathbf{X}|W}(\mathcal{F}_j|i), \tag{5.2}$$

where $\mathcal{F}_j$ is the set of feature vectors extracted from the jth database image, $\mathcal{I}_j$. The posterior probabilities $P_{W|\mathbf{X}}(i|\mathcal{F})$ are computed by application of Bayes rule

$$P_{W|\mathbf{X}}(i|\mathbf{x}) = \frac{P_{\mathbf{X}|W}(\mathbf{x}|i)P_W(i)}{P_{\mathbf{X}}(\mathbf{x})} \tag{5.3}$$

where $P_W(i)$ is the prior probability of class i. Note that, at annotation time, SML produces an ordering of the semantic classes by posterior probability $P_{W|\mathbf{X}}(i|\mathcal{F})$. This ordering is optimal in the MPE sense.

5.2 Estimation of Semantic Class Distributions

The ith semantic class density is estimated from a training set $\mathcal{D}_i$ containing all feature vectors extracted from images labeled with concept w_i. However, many of the concepts only occupy a fraction of the images that contain them. Since most images are a combination of various concepts, the assembly of a training set for each semantic class should be preceded by (1) careful semantic segmentation, and (2) identification of the image regions containing the associated visual feature vectors. In practice, the manual segmentation of all database images with respect to all concepts of interest is infeasible. Existing segmentation algorithms are also unable to produce a decomposition of each image into a plausible set of semantic regions. A pressing question is then whether it is possible to estimate the densities of a semantic class without prior semantic segmentation, i.e., from a training set containing a significant percentage of feature vectors from other semantic classes. This question has been studied in the machine learning literature, and it is usually referred to as *multiple instance* learning [1, 3, 29, 75, 76].

Unlike classical learning, which is based on sets of positive and negative examples, multiple instance learning addresses the problem of how to learn models from positive and negative *bags of examples*. A bag is a collection of examples and is considered positive if at least one of those examples is positive. Otherwise, the bag is negative. The key property is that, for sufficiently large bags, the empirical distribution of feature vectors in the positive bag tends to approximate the distribution of the positive class. Although this has been mostly demonstrated experimentally, the experimental evidence is substantial. For example, [75] has shown that the peak of the empirical distribution tends to occur in the region of support of the positive examples, [123] has shown that the empirical distribution performs well when used as the concept's class conditional distribution for image classification, and [36] has shown that clustering the collection of feature vectors produces a codebook of parts which are representative of the positive class (e.g., eyes, mouth, or nose for a face concept).

The intuition for this behavior is simple: while the negative examples present in positive bags tend to be spread all over the feature space,

the positive examples are much more likely to be concentrated within a small region of the latter. Hence, the empirical distribution of positive bags is well approximated by a mixture of two components: a uniform component from which negative examples are drawn, and the distribution of positive examples. The key insight is that, because it must integrate to one, the uniform component tends to have small amplitude (in particular if the feature space is high dimensional). It follows that, although the density of the common concept may not be dominant in any individual image, the consistent appearance in all images makes it dominant over the entire positive bag. Hence, a density estimate produced from the entire bag should be a close approximation to the density of the target concept. In the following section, we provide a more precise characterization of this intuition, and some theoretical results on the learnability of concepts using the multiple instance learning paradigm. For now, we consider the problem of learning semantic class densities efficiently.

5.3 Efficient Density Estimation

Since concept densities must be learned from large numbers of images, the direct estimation of $P_{\mathbf{X}|W}(\mathbf{x}|i)$ from the set of *all* feature vectors extracted from *all* training images that contain concept w_i is usually infeasible. A more effective strategy is to reuse computation. Given a training set $\mathcal{D}_i$ of D_i images, a density estimate is first produced for each image in $\mathcal{D}_i$, originating a sequence $P_{\mathbf{X}|L,W}(\mathbf{x}|l,i), l \in \{1,\ldots D_i\}$, where L is a hidden variable that indicates the image number. The concept density is then estimated in a second step, by combining the individual image density estimates. This can be done in two ways: *model averaging* and *hierarchical estimation*.

Under the model averaging strategy, the concept density is obtained by averaging the individual image densities

$$P_{\mathbf{X}|W}(\mathbf{x}|i) = \frac{1}{D_i}\sum_{l=1}^{D_i} P_{\mathbf{X}|L,W}(\mathbf{x}|l,i). \tag{5.4}$$

The direct application of (5.4) is feasible when the densities $P_{\mathbf{X}|L,W}(\mathbf{x}|l,i)$ are defined over a (common) partition of the feature

space. For example, if all densities are histograms defined on a partition of the feature space $\mathcal{S}$ into Q cells $\{\mathcal{S}_q\}, q = 1, \ldots, Q$, and $h_{i,l}^q$ the number of feature vectors from class i that land on cell $\mathcal{S}_q$ for image l, then the average class histogram is simply

$$\hat{h}_i^q = \frac{1}{D_i} \sum_{l=1}^{D_i} h_{i,l}^q.$$

However, when (1) the partition is not the same for all histograms or (2) more sophisticated models (e.g., mixture or non-parametric density estimates) are adopted, model averaging is not as simple.

Consider, for example, the Gauss mixture model

$$P_{\mathbf{X}|L,W}(\mathbf{x}|l,i) = \sum_k \pi_{i,l}^k \mathcal{G}(\mathbf{x}, \mu_{i,l}^k, \Sigma_{i,l}^k), \tag{5.5}$$

where $\sum_k \pi_{i,l}^k = 1$. Direct application of (5.4) leads to

$$P_{\mathbf{X}|W}(\mathbf{x}|i) = \frac{1}{D_i} \sum_k \sum_{l=1}^{D_i} \pi_{i,l}^k \mathcal{G}(\mathbf{x}, \mu_{i,l}^k, \Sigma_{i,l}^k), \tag{5.6}$$

i.e., a D_i-fold increase in the number of Gaussian components per mixture. Since, at annotation time, this probability has to be evaluated for each semantic class, it is clear that straightforward model averaging will lead to an extremely slow annotation process.

An efficient alternative is to adopt a hierarchical density estimation method first proposed in [122] for image indexing. This method is based on a mixture hierarchy, where children densities consist of different combinations of subsets of the parents' components. In the semantic labeling context, image densities are children and semantic class densities their parents. As shown in [122], it is possible to estimate the parameters of class mixtures directly from those available for the individual image mixtures, using a two-stage procedure. The first stage is the model averaging of (5.6). Assuming that each image mixture has K components, this leads to a class mixture of $D_i K$ components with parameters

$$\{\pi_j^k, \mu_j^k, \Sigma_j^k\}, \quad j = 1, \ldots, D_i, \;\; k = 1, \ldots, K. \tag{5.7}$$

Algorithm 2 Hierarchical EM algorithm (Gaussian mixtures)

Input: set of parameters, $\{\pi_j^k, \mu_j^k, \Sigma_j^k\},\ j = 1, \ldots, D_i, k = 1, \ldots, K$, of the average model of (5.4), and initial mixture parameters $\{\pi_c^m, \mu_c^m, \Sigma_c^m\}, m = 1, \ldots, M$ of the class mixture.

repeat

E-step: for each component $\{\pi_j^k, \mu_j^k, \Sigma_j^k\}$ of the average model and each component $\{\pi_c^m, \mu_c^m, \Sigma_c^m\}$ of the class mixture, compute

$$h_{jk}^m = \frac{\left[\mathcal{G}(\mu_j^k, \mu_c^m, \mathbf{\Sigma}_c^m) e^{-\frac{1}{2}\text{trace}\{(\mathbf{\Sigma}_c^m)^{-1}\mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k} \pi_c^m}{\sum_l \left[\mathcal{G}(\mu_j^k, \mu_c^l, \mathbf{\Sigma}_c^l) e^{-\frac{1}{2}\text{trace}\{(\mathbf{\Sigma}_c^l)^{-1}\mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k} \pi_c^l} \tag{5.8}$$

M-step: update the class mixture parameters with

$$(\pi_c^m)^{\text{new}} = \frac{\sum_{jk} h_{jk}^m}{D_i K} \tag{5.9}$$

$$(\mu_c^m)^{\text{new}} = \sum_{jk} w_{jk}^m \mu_j^k, \quad \text{where } w_{jk}^m = \frac{h_{jk}^m \pi_j^k}{\sum_{jk} h_{jk}^m \pi_j^k} \tag{5.10}$$

$$(\mathbf{\Sigma}_c^m)^{\text{new}} = \sum_{jk} w_{jk}^m \left[\mathbf{\Sigma}_j^k + (\mu_j^k - (\mu_c^m)^{\text{new}})(\mu_j^k - (\mu_c^m)^{\text{new}})^T\right]. \tag{5.11}$$

until convergence

The second is an extension of EM which clusters the Gaussian components into a M-component mixture, where M is the number of components desired at the class level. This hierarchical extension of EM is presented in Algorithm 2.

Note that the number of parameters in each image mixture is orders of magnitude smaller than the number of feature vectors in the image itself. Hence, the complexity of estimating the class mixture parameters is negligible when compared to that of estimating the individual mixture parameters for all images in the class. It follows that the overall training complexity is dominated by the latter task, i.e., only marginally superior to that of model averaging. On the other hand, the

Algorithm 3 Learning concept mixtures

Input: Training image database $\mathcal{T}$, number of components in image K_1 and concept K_2 mixture models.

for each image $\mathcal{I}_i \in \mathcal{T}$ **do**

extract a set $\mathcal{F} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ of features from $\mathcal{I}_i$

estimate the parameters $\{\pi_i^k, \mu_i^k, \Sigma_i^k\}_{k=1}^{K_1}$ of a Gauss mixture model

$$P_{\mathbf{X}|L}(\mathbf{x}|i) = \sum_{k=1}^{K_1} \pi_i^k \mathcal{G}(\mathbf{x}, \mu_i^k, \Sigma_i^k) \tag{5.12}$$

that maximize the likelihood of $\mathcal{F}$, using the EM procedure of Algorithm 1.

end for

for each semantic class $\omega \in \mathcal{L}$ **do**

build a training image set $\tilde{\mathcal{T}} \subset \mathcal{T}$, where $\omega \in \mathbf{w}_i$ for all $\mathcal{I}_i \in \tilde{\mathcal{T}}$.

set

$$P_W(\omega) = |\tilde{\mathcal{T}}|/|\mathcal{T}|$$

learn a concept-mixture

$$P_{\mathbf{X}|W}(\mathbf{x}|\omega) = \sum_{k=1}^{K_2} \pi_w^k \mathcal{G}(\mathbf{x}, \mu_w^k, \Sigma_w^k) \tag{5.13}$$

by applying the hierarchical EM procedure of Algorithm 2 to the image-level mixtures of (5.12) associated with the images $\mathcal{I}_i \in \tilde{\mathcal{T}}$.

end for

Output: concept level mixtures $P_{\mathbf{X}|W}(\mathbf{x}|\omega)$ and concept probabilities $P_W(\omega)$.

complexity of evaluating likelihoods is significantly smaller than that of model averaging.

5.4 Algorithms

In this section, we summarize the three algorithms used for MPE concept learning, annotation, and retrieval. The learning and annotation

Algorithm 4 Image annotation

Input: Image $\mathcal{I}$ to annotate, concept models $P_{\mathbf{X}|W}(\mathbf{x}|\omega)$, and probabilities $P_W(\omega)$.
extract a set $\mathcal{F} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ of features from $\mathcal{I}$
for each semantic class $\omega \in \mathcal{L}$ **do**
compute (unnormalized) posterior probabilities

$$\gamma_\omega = \sum_k \log P_{\mathbf{X}|W}(\mathbf{x}_k|\omega) + \log P_W(\omega)$$

end for
annotate the test image with the five classes ω_i of largest posterior probability, γ_{ω_i}.
Output: image annotations ω_i and posterior probabilities γ_{ω_i}.

Algorithm 5 Image retrieval

Input: database of test images $\mathcal{T}_T$, concept models $P_{\mathbf{X}|W}(\mathbf{x}|\omega)$ and probabilities $P_W(\omega)$, and query word ω_q.
for each image $\mathcal{I}_t \in \mathcal{T}_T$ **do**
annotate $\mathcal{I}_t$ using Algorithm 4 with models $P_{\mathbf{X}|W}(\mathbf{x}|\omega)$, and probabilities $P_W(\omega)$.
end for
rank the images labeled with the query word ω_q by decreasing posterior probability γ_{ω_q}.
Output: image ranking.

processes are also illustrated by Figure 1.4. For the algorithm that learns concept models, we assume a training set $\mathcal{D} = \{(\mathcal{I}_1, \mathbf{w}_1), \ldots, (\mathcal{I}_D, \mathbf{w}_D)\}$ of image-caption pairs, where $I_i \in \mathcal{T}$ with $\mathcal{T} = \{\mathcal{I}_1, \ldots, \mathcal{I}_D\}$, and $\mathbf{w}_i \subset \mathcal{L}$, with $\mathcal{L} = \{w_1, \ldots, w_T\}$. The learning procedure is presented in Algorithm 3. A mixture model is learned for each image in $\mathcal{T}$ and the mixtures associated with each concept are then pooled to learn the concept density. The concept probabilities $P_W(\omega)$ are set to the ratios of the number of images labeled with each concept ω and the total number of images.

The annotation algorithm uses the concept models and probabilities to identify the concepts that best describe a given image $\mathcal{I}$ to annotate.

This is done with the MPE rule, i.e., by selecting the concepts of largest posterior probability. Finally, the retrieval algorithm has inputs (a) a query concept ω_q, and (b) a database of test images $\mathcal{T}_T$, such that $\mathcal{T}_T \bigcap \mathcal{T}_D = \emptyset$. It returns a set of images from $\mathcal{T}_T$, ordered by posterior probability of depicting the concept. We have found, experimentally, that the restriction to the images for which the query is a top label increases the robustness of the ranking (as compared by the simple ranking by label posterior).

5.5 Experiments

A number of proposals for semantic image annotation and retrieval have appeared in the literature. In general, it is difficult to compare different algorithms, unless their performance has been evaluated with a common experimental protocol. A popular protocol, here referred to as Corel5K, has been adopted by a number of research groups [31, 35, 60]. There are, nevertheless, two significant limitations associated with the Corel5K protocol. First, because it is based on a relatively small database, many of the semantic labels in Corel5K have a very small number of examples. This makes it difficult to guarantee that the resulting annotation systems have good generalization. Second, because the size of the caption vocabulary is also relatively small, Corel5K does not test the scalability of annotation/retrieval algorithms.

Some of these limitations are corrected by the Corel30K protocol, which is an extension of Corel5K based on a substantially larger database. Neither of the two protocols is, however, easy to apply to massive databases since both require the manual annotation of each training image. The protocol proposed by Li and Wang [64] (which we refer to as PSU) is a suitable alternative for testing large-scale labeling and retrieval systems. Because each of the three protocols has been used to characterize a non-overlapping set of semantic labeling/retrieval techniques, we performed an evaluation on all three.

5.5.1 The Corel5K and Corel30K Protocols

The evaluation of a semantic annotation/labeling and retrieval system requires three components: an image database with manually produced

annotations, a strategy to train and test the system, and a set of measures of retrieval and annotation performance. The Corel5K benchmark [31, 35, 60] is based on the Corel image database: 5000 images from 50 Corel Stock Photo CDs were divided into a training set of 4000 images, a validation set of 500 images, and a test set of 500 images. An initial set of model parameters is learned on the training set. Parameters that require cross-validation are then optimized on the validation set, after which this set is merged with the training set to build a new training set of images. Non-cross-validated parameters are then tuned with this training set. Each image has a caption of 1–5 semantic labels, and there are 371 labels in the data set.

Image annotation performance is evaluated by comparing the captions automatically generated for the test set with the human-produced ground-truth. Similar to [35, 60], we define the automatic annotation as the five semantic classes of largest posterior probability, and compute the recall and precision of every word in the test set. For a given semantic descriptor, assuming that there are w_H human annotated images in the test set, and the system annotates w_{auto}, of which w_C are correct, recall and precision are given by $\text{recall} = \frac{w_C}{w_H}$, and $\text{precision} = \frac{w_C}{w_{\text{auto}}}$, respectively. As suggested in previous works [35, 60], the values of recall and precision are averaged over the set of words that appear in the test set. Finally, we also consider the number of words with non-zero recall (i.e., words with $w_C > 0$), which provides an indication of how many words the system has effectively learned.

The performance of semantic retrieval is also evaluated by measuring precision and recall. Given a query term and the top n image matches retrieved from the database, recall is the percentage of all relevant images contained in the retrieved set, and precision the percentage of the n which are relevant (where relevant means that the ground-truth annotation of the image contains the query term). Once again, we adopted the experimental protocol of [35], evaluating retrieval performance by the mean average precision (MAP). This is defined as the average precision, over all queries, at the ranks where recall changes (i.e., where relevant items occur).

The Corel30K protocol is similar to Corel5K but substantially larger, containing 31,695 images and 5,587 words. Of the 31,695 images,

90% were used for training (28,525 images) and 10% for testing (3170 images). Only the words (950 in total) that were used as annotations for at least 10 images were trained.

5.5.2 The PSU Protocol [64]

For very large image sets, it may not even practical to label each training image with ground-truth annotations. An alternative approach, proposed by Li and Wang [64], is to assign images to loosely defined categories, where each category is represented by a set of words that characterize the category as a whole, but may not accurately characterize each individual image. For example, a collection of images of tigers running in the wild may be annotated with the words "tiger", "sky", "grass", even though some of the images may not actually depict sky or grass. We refer to this type of annotation as noisy supervised annotation. While it reduces the time required to produce ground-truth annotations, it introduces noise in the training set, where each image in some category may contain only a subset of the category annotations.

Li and Wang [64] relied on noisy supervised annotation to label very large databases, by implementing a two-step annotation procedure, which we refer to as supervised category-based labeling (SCBL). The image to label is first processed with an image category classifier that identifies the five image categories to which the image is most likely to belong. The annotations from those categories are then pooled into a list of candidate annotations with frequency counts for re-occurring annotations. The candidate annotations are then ordered based on the hypothesis test that a candidate annotation has occurred randomly in the list of candidate annotations.

More specifically, the probability that the candidate word appears at least j times in k randomly selected categories is

$$P(j,k) = \sum_{i=j}^{k} I(i \leq m) \frac{\binom{m}{i}\binom{n-m}{k-i}}{\binom{n}{k}}$$

where $I(.)$ is the indicator function, n the total number of image categories, and m the number of image categories containing the word. For

$n, m \gg k$, the probability can be approximated by

$$P(j,k) \approx \sum_{i=j}^{k} \binom{k}{i} p^i (1-p)^{k-i},$$

where $p = m/n$ is the frequency with which the word appears in the annotation categories. A small $P(j,k)$ indicates a low probability that the candidate word occurred randomly (i.e., the word has high significance as an annotation). Hence, candidate words with $P(j,k)$ below a threshold value are selected as the annotations.

Li and Wang [64] also proposed an experimental protocol, based on noisy supervised annotation, for the evaluation of highly scalable semantic labeling and retrieval systems. This protocol, which we refer to as PSU, is also based on the Corel image set, containing 60,000 images with 442 annotations. The image set was split into 600 image categories of 100 images each, which were then annotated with a general description that reflects the image category as a whole. For performance evaluation, 40% of the PSU images were reserved for training (23,878 images), and the remainder (35,817 images) used for testing. Note that Li and Wang [64] only used 4,630 of the 35,817 possible test images, whereas all the test images were used in the experiments reported here. Annotation and retrieval performance were evaluated with the same measures used in Corel5K and Corel30K.

5.6 Experimental Results

In this section, we compare the performance of SML with various previous approaches. We start with a comparison against methods that have been evaluated on Corel5K. We then compare SML to SCBL on the larger PSU benchmark. Finally, we perform a study of the scalability and robustness of SML.

5.6.1 Image Representation

All experiments were based on the image representation previously used in the QBVE experiments of Section 4. In all cases, we used DCT features. For the implementation of SML, a Gaussian mixture of 64

components was fit to the entire collection of images associated with each annotation. We will refer to this class representation as GMM-DCT. The images from the PSU database were annotated using both the SML and SCBL methods. In the latter case, the classifiers for the image categories had 64 mixture components and used the GMM-DCT representation (i.e., in SML, equivalent to setting the image categories as the annotation words).

5.6.2 Comparison of SML and Unsupervised Labeling

Table 5.1 presents the results obtained for SML and various previously proposed methods (results from [60, 35]) on Corel5K. Specifically, we considered the co-occurrence model of [78], the translation model of [31], the continuous-space relevance model of [35, 60], and the multiple-Bernoulli relevance model (MBRM) of [35]. Overall, SML achieves the best performance, exhibiting a gain of 16% in recall for an equivalent level of precision when compared to the next best results (MBRM). Furthermore, the number of words with positive recall increases by 15%. Figure 1.6 presents some examples of the annotations produced. Note that, when the system annotates an image with a descriptor not contained in the human annotation groundtruth, this annotation is frequently plausible.

Table 5.2 shows that, for ranked retrieval on Corel, SML produces results superior to those of MBRM. In particular, it achieves a gain

Table 5.1. Performance comparison of automatic annotation on Corel5K.

Models	Co-occurrence	Translation	CRM	MBRM	SML
#words with recall > 0	19	49	107	122	137
	Results on all 260 words				
Mean per-word recall	0.02	0.04	0.19	0.25	0.29
Mean per-word precision	0.03	0.06	0.16	0.24	0.23

Table 5.2. Retrieval results on Corel5K.

Mean average precision for corel dataset		
Models	All 260 words	Words with recall > 0
SML	0.31	0.49
MBRM	0.30	0.35

of 40% mean average precision on the set of words that have positive recall. Figure 1.5 illustrates the retrieval results obtained with one word queries for challenging visual concepts. Note the diversity of visual appearance of the returned images, indicating that SML has good generalization ability.

5.6.3 Comparison of SML and SCBL

We next compared the image categorization performance of the GMM-DCT class representation with that of the representation of [64]. In [64], an image category is represented by a two-dimensional multi-resolution hidden Markov model (2D-MHMM) defined on a feature space of localized color and wavelet texture features at multiple scales. An image was considered to be correctly categorized if any of the top r categories is the true category. Table 5.3 shows the accuracy of image categorization using the two class representations. GMM-DCT outperformed the 2D-MHMM of [64] in all cases, with an improvement of about 0.10 (from 0.26 to 0.36). Figure 5.1 (left) shows the categorization accuracy of GMM-DCT versus the dimension of the DCT feature space. It can be seen that the categorization accuracy increases with the dimension of the feature space, but remains fairly stable over a significant range of dimensions.

We next compared the annotation performance of the two steps of SCBL, using the GMM-DCT representation (we denote this combination by SCBL-GMM-DCT) and [64]. Following [64], the performance was measured using "mean coverage", which is the percentage of ground-truth annotations that match the computer annotations. Table 5.4 shows the mean coverage of SCBL-GMM-DCT and of [64], using a threshold of 0.0649 on $P(j,k)$, as in [64], and without using a threshold. Annotations using GMM-DCT outperform those of [64] by

Table 5.3. Accuracy of image categorization on PSU database.

Class representation	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
GMM-DCT	0.2090	0.2701	0.3094	0.3379	0.3615
2D-MHMM	0.1188	0.1706	0.2076	0.2324	0.2605

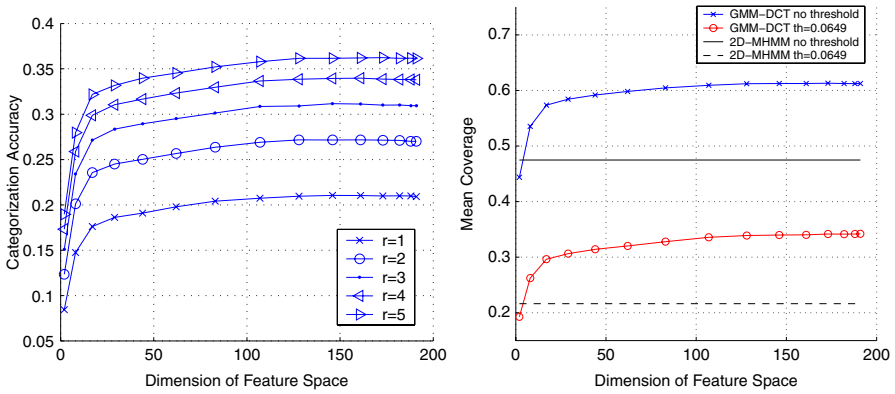


Fig. 5.1 Left: image categorization accuracy on PSU using GMM-DCT versus the dimension of the DCT feature space. Right: mean coverage of annotation on PSU using SCBL-GMM-DCT versus the dimension of the DCT feature space.

Table 5.4. Mean coverage for annotation on PSU database.

Method	Threshold = 0.0649	No threshold
SCBL-GMM-DCT	0.3420	0.6124
Li and Wang [64]	0.2163	0.4748

about 0.13 (from 0.22 to 0.34 using a threshold, and 0.47 to 0.61 for no threshold). Figure 5.1 (right) shows the mean coverage versus the dimension of the DCT feature space. Again, performance increases with feature space dimension, but remains fairly stable over a large range of dimensions.

Finally, we compared SCBL and SML when both methods used the GMM-DCT representation. SCBL annotation was performed by thresholding the hypothesis test (SCBL-GMM-DCT threshold), or by selecting a fixed number annotations (SCBL-GMM-DCT fixed). Figure 5.2 presents the precision-recall (PR) curves produced by the two methods. The SML curve has the best overall precision at 0.23, and its precision is clearly superior to that of SCBL at most levels of recall. There are, however, some levels where SCBL-GMM-DCT leads to a better precision. This is due to the coupling of words within the same image category, and to the noise in the ground-truth annotations of PSU. A more detailed discussion is available in [16].

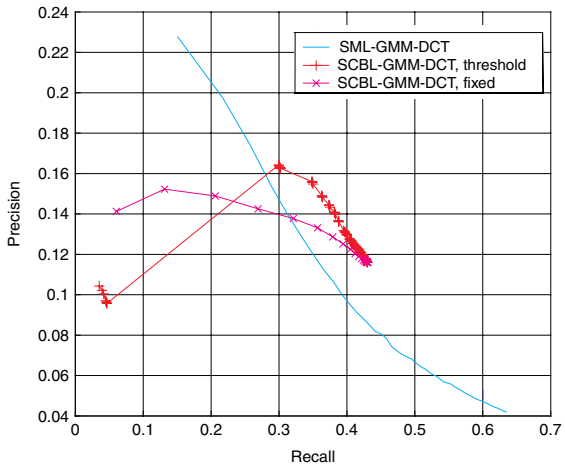


Fig. 5.2 Precision-recall for SCBL and SML using GMM-DCT on the PSU database.

In summary, the experimental results show that the GMM-DCT representation substantially outperforms the 2D-MHMM of [64] in both image categorization and annotation using SCBL. When comparing SML and SCBL based on the GMM-DCT representation, SML achieves the best overall precision, but for some recall levels SCBL can achieve a better precision due to coupling of annotation words and noise in the annotation ground truth.

6

Weakly Supervised Estimation of Probability Densities

In the previous section, we have seen that effective image classifiers can be learned hierarchically and with very weak supervision. To estimate class-conditional feature distributions, we simply pooled all the feature vectors from all the image associated with each class. In principle, this might appear to be a bad idea. After all, many of the features extracted from any image of a given class may not even belong to that class. This is illustrated by the top portion of Figure 6.1. The image shown contains a number of visual concepts, and has been manually annotated as belonging to the classes "people", "beach", "sand", "palm trees", "hut", and "vacation". As shown on the right, feature vectors extracted from this image will cover a large region of the feature space. In particular, we illustrate how various sub-regions of the space are populated by feature vectors from different classes, e.g., "hut", "palm trees", "sand" (shown in red), or "people" (shown in green). When the image is used as an example of the "people" class, most of the feature vectors will actually fall outside the region of the space associated with this concept (the green area). How is it, then, that the classifier can learn to disregard all the red areas, and declare the presence of the "people" concept only when faced with feature vectors from the green region?

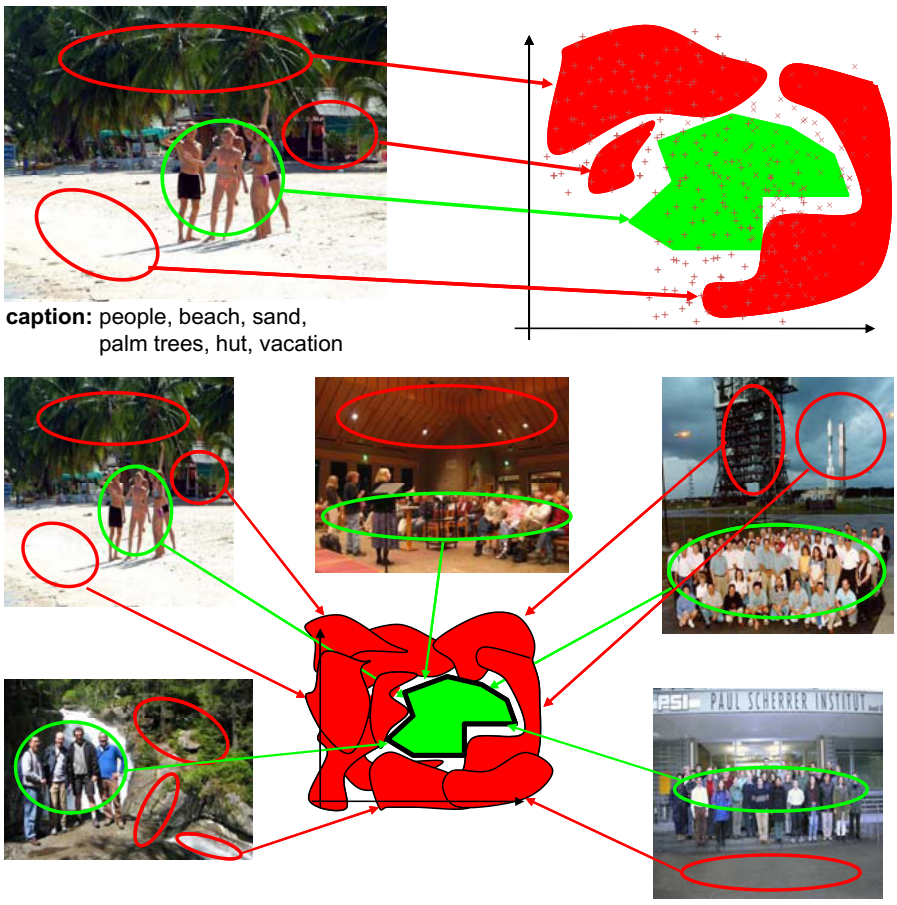


Fig. 6.1 Top: a typical image contains feature vectors associated with various visual concepts. Only some of these will correspond to a particular interpretation of the image, e.g., "people". Bottom: when a diverse set of images labeled with a common concept (again "people") is assembled, the resulting feature distribution is dominated by the distribution of that concept.

We have already hinted that the answer is multiple instance learning. While all this training noise makes it impossible to learn the people concept from a single image, this becomes possible from a collection of images. The key requirement is that this collection be *diverse*. By this it is meant that, while all images will depict "people", the remaining concepts must be random. For example, people should appear indoors

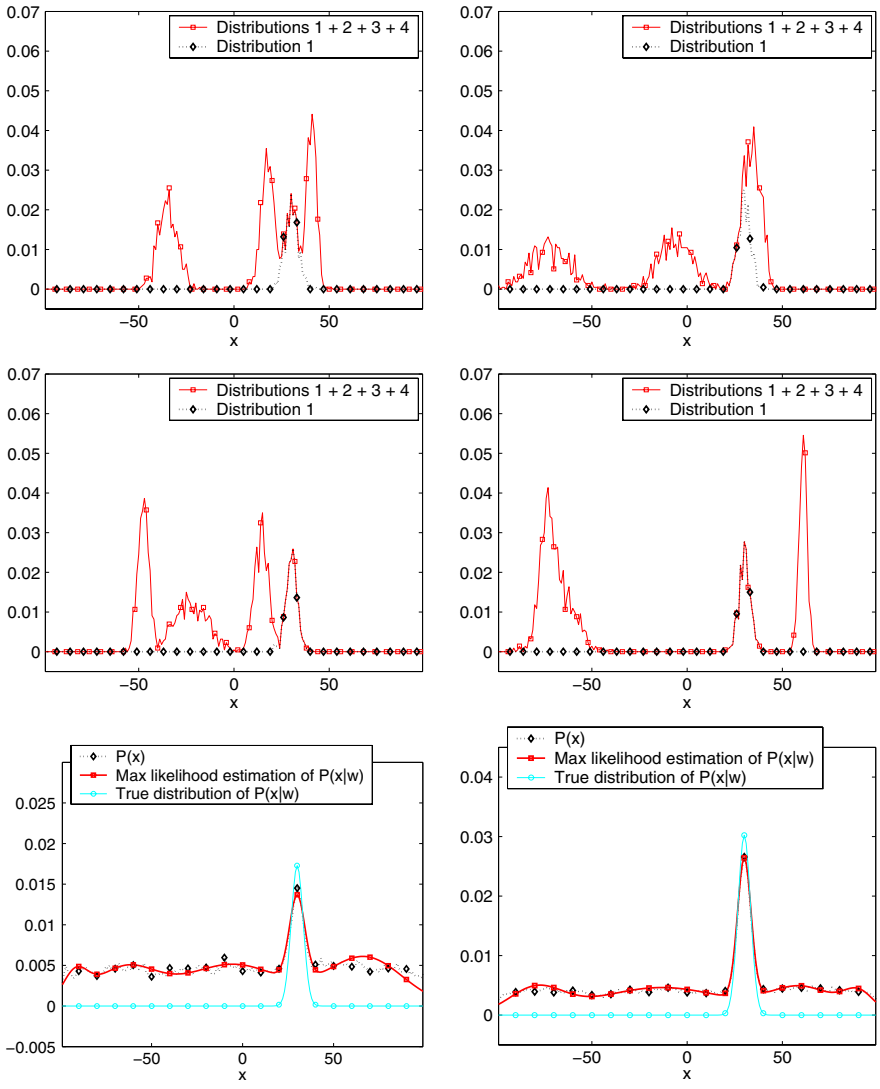
in some images, outdoors in others, sometimes on the beach, others on mountains, others in urban environments, and so forth. This is illustrated by the bottom portion of Figure 6.1. Note that all images in this figure contribute feature vectors to the region populated by the people concept (again shown in green). However, since the backgrounds are diverse, each image contributes to a different set of red regions. If the space is large, there will be many such regions, and each will get a few feature vectors. Hence, a histogram computed over the entire space will contain a very strong peak for the green cell, and much smaller feature vector counts for the red cells. This implies that the probability distribution is dominated by the green region, and a density estimate learned from the entire feature set will approximate that learned from the "people" features alone. In this section, we attempt to quantify some of these statements by considering the following questions. Is the distribution learned from the whole image set really dominated by the distribution of the common concept? What are the variables that affect the convergence to this distribution as the training set grows? How can one quantify the statement that the backgrounds should be diverse? What is meant when it is said that the space should be large enough? We derive theoretical answers to these questions that help understand the effectiveness of the classifiers introduced in the previous section.

6.1 Weakly Supervised Density Estimation

We start by analyzing the simple synthetic example of Figure 6.2. This example illustrates the problem of learning semantic class densities for a hypothetical set of images containing four semantic Gaussian concepts. Each concept has probability $\pi_i \in [0,1]$, i.e., it occupies π_i of the image area, on average. Introducing a hidden variable L for the image number, the distribution of each image can be written as

$$P_{X|L}(x|l) = \sum_{i=1}^{4} \pi_i \mathcal{G}(x, \mu_i^l, \sigma_i^l), \tag{6.1}$$

where $\sum_{i=1}^{4} \pi_i = 1, (\mu_i^l, \sigma_i^l)$ are the mean and variance of the ith Gaussian associated with the lth image, with $\mathcal{G}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$,



(a) $\pi_1 = 0.3$ (b) $\pi_1 = 0.4$

Fig. 6.2 Synthetic example of multiple instance learning of semantic class densities. Top and center rows: probability distributions of individual images ($P_{X|L}(\mathbf{x}|l)$). Each image distribution is simulated by a mixture of the distribution of the concept of interest (dashed line) and three distributions of other visual concepts present in the image (solid line). All concepts are simulated as Gaussians of different mean and variance. Bottom row: empirical distribution $P_X(\mathbf{x})$ obtained from a bag of $D = 1000$ simulated images, the estimated class conditional distribution (using maximum likelihood parameter estimates under a mixture of Gaussians model) $\hat{P}_{X|W}(x|w)$, and the true underlying distribution $P_{X|W}(x|w) = \mathcal{G}(x, \mu_w, \sigma_w)$ of the common concept w. Each column is associated with a different value of π_1 in (6.1).

and the distribution of the bag of D images is

$$P_X(x) = \sum_{l=1}^{D} P_{X|L}(x|l)P_L(l) = \frac{1}{D}\sum_{l=1}^{D}\sum_{i=1}^{4}\pi_i\mathcal{G}(x,\mu_i^l,\sigma_i^l)$$

where we have assumed that all images are equally likely.

If one of the four components (e.g., the first, for simplicity) is always the density of concept w, e.g., $\mu_1^l = \mu_w$ and $\sigma_1^l = \sigma_w, \forall l$, and the others are randomly selected from a pool of Gaussians of uniformly distributed mean and standard deviation, then

$$\begin{aligned} P_X(x) &= \sum_{i=1}^{4}\frac{1}{D}\sum_{l=1}^{D}\pi_i\mathcal{G}(x,\mu_i^l,\sigma_i^l) \\ &= \pi_1\mathcal{G}(x,\mu_w,\sigma_w) + \sum_{i=2}^{4}\frac{\pi_i}{D}\sum_{l=1}^{D}\mathcal{G}(x,\mu_i^l,\sigma_i^l) \end{aligned}$$

and, from the law of large numbers, as $D \to \infty$

$$P_X(x) = \pi_1\mathcal{G}(x,\mu_w,\sigma_w) + (1-\pi_1)\int \mathcal{G}(x,\mu,\sigma)p_{\mu,\sigma}(\mu,\sigma)d\mu d\sigma,$$

where $p_{\mu,\sigma}(\mu,\sigma)$ is the joint distribution of the means and variances of the components other than that associated with w. Hence, the distribution of the positive bag for concept w is a mixture of (1) the concept's density, and (2) the average of many Gaussians of different mean and covariance. The latter converges to a distribution that is approximately uniform and, in order to integrate to one, must have small amplitude, i.e.,

$$\lim_{D\to\infty} P_X(x) = \pi_1\mathcal{G}(x,\mu_w,\sigma_w) + (1-\pi_1)\kappa,$$

with $\kappa \approx 0$.

Figure 6.2 presents a simulation of this effect, when $\mu \in [-100, 100]$, $\sigma \in [0.1, 10]$, $\mu_w = 30$, $\sigma_w = 3.3$, and the bag contains $D = 1{,}000$ images. Figure 6.3 presents a comparison between the estimate of the distribution of w, $\hat{P}_{X|W}(\mathbf{x}|w)$, obtained by fitting (in the maximum likelihood sense) a mixture of five Gaussians (using the EM algorithm) to the entire bag, and the true distribution $P_{X|W}(\mathbf{x}|w) = \mathcal{G}(\mathbf{x},\mu_w,\sigma_w)$. The

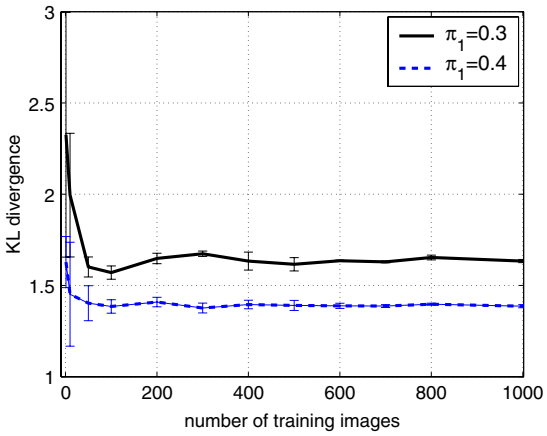


Fig. 6.3 KL divergence between estimated, $\hat{P}_{X|W}(x|w)$, and actual, $P_{X|W}(x|w)$, class conditional density of concept w as a function of the number of training images D, for different values of π_1. Error bars illustrate the standard deviation over a set of 10 experiments for each combination of $D = \{1, \ldots, 1000\}$ and $\pi_1 = 0.3, 0.4$.

comparison is based on the Kullback–Leibler (KL) divergence,

$$\mathrm{KL}[\hat{P}_{X|W}(\mathbf{x}|w)||P_{X|W}(\mathbf{x}|w)] = \int \hat{P}_{X|W}(\mathbf{x}|w) \log \frac{\hat{P}_{X|W}(\mathbf{x}|w)}{P_{X|W}(\mathbf{x}|w)} d\mathbf{x},$$

and shows that, even when π_1 is small (e.g., $\pi_1 = 0.3$), the distribution of concept w dominates the empirical distribution of the bag, as the number D of images increases.

Figure 6.4 shows that the same type of behavior is observed in real image databases. In this example, semantic densities were learned over a set of training images from the Corel database, using the methods of the previous section. A set of test images were then semantically segmented by (1) extracting a feature vector from each location in the test image, and (2) classifying this feature vector into one of the semantic classes present in the image (semantic classes were obtained from the caption provided with the image [31]). Figure 6.4 depicts the indexes of the classes to which each image location was assigned (class indexes shown in the color bar on the right of the image) according to

$$i^*(\mathcal{F}) = \begin{cases} \arg\max_i P_{W|\mathbf{X}}(i|\mathcal{F}), & \text{if } P_{W|\mathbf{X}}(i|\mathcal{F}) > \tau \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

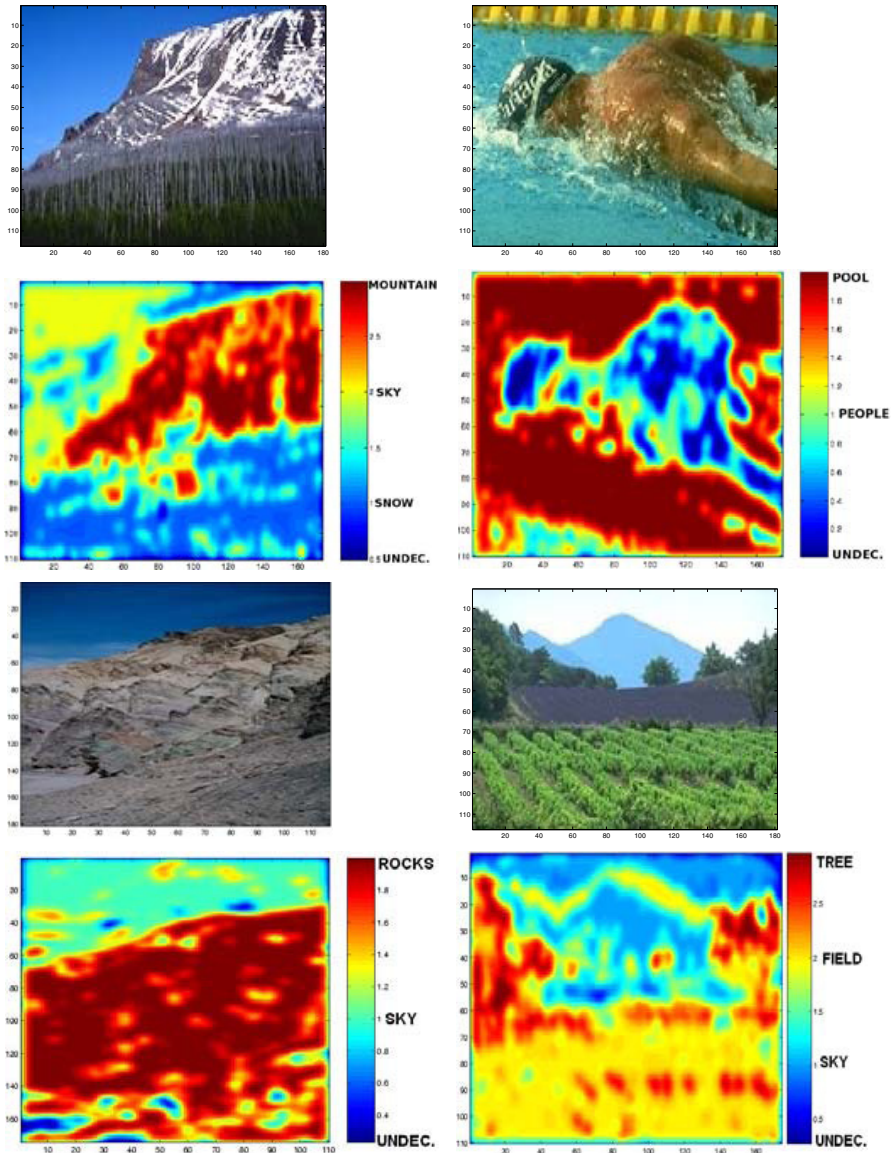


Fig. 6.4 Original images (top row) and posterior assignments (bottom row) for each image neighborhood (Undec. means that no class has a posterior bigger that τ in (6.2)).

where $\mathcal{F}$ is the set of feature vectors extracted from the image to segment, $\tau = 0.5$,

$$P_{W|\mathbf{X}}(i|\mathcal{F}) = \frac{P_{\mathbf{X}|W}(\mathcal{F}|i)P_W(i)}{P_X(\mathcal{F})}$$

with

$$P_{\mathbf{X}|W}(\mathcal{F}|i) = \prod_k P_{\mathbf{X}|W}(\mathbf{x}_k|i), \tag{6.3}$$

$P_W(i)$ uniform,

$$P_{\mathbf{X}}(\mathcal{F}) = P_{\mathbf{X}|W}(\mathcal{F}|i)P_W(i) + P_{\mathbf{X}|W}(\mathcal{F}|\neg i)P_W(\neg i),$$

and the density for "no class i" ($\neg i$) learned from all training images that did not contain class i in their caption. In order to facilitate visualization, the posterior maps were reproduced by adding a constant, the index of the class of largest posterior, to that posterior. Regions where all posteriors were below threshold were declared "undecided". Finally, the segmentation map was smoothed with a Gaussian filter. Note that, while coarse, the segmentations do (1) split the images into regions of different semantics, and (2) make correct assignments between regions and semantic descriptors. This shows that the learned densities are close to the true semantic class densities.

6.2 Concept Learnability

Having provided some empirical evidence for the convergence of multiple instance learning, we turn to the derivation of theoretical results on the learnability of semantic concepts. As before, images are represented as collections of feature vectors, i.e., $I_i = \{\mathbf{x}_1^i, \ldots, \mathbf{x}_n^i\}$ for the ith image, and concepts are drawn from a random variable W that assigns a probability distribution to a concept vocabulary $\mathcal{L}$. The goal is to learn the probability distribution associated with a certain concept c, $P_{\mathbf{X}|W}(\mathbf{x}|c)$, which we will refer to as $P_c(\mathbf{x})$ for simplicity. Learning is based on a training set $\mathcal{D}_c$. Each image $I \in \mathcal{D}_c$ is a sample of feature vectors from a distribution

$$P_{\mathbf{X}}(\mathbf{x}) = \pi P_c(\mathbf{x}) + (1 - \pi)P_B(\mathbf{x}). \tag{6.4}$$

The probability π is the percentage of the image area covered by c, on average, and $P_B(\mathbf{x})$ a background distribution that accounts for everything else. Since any probability distribution can be approximated arbitrarily well by a (potentially infinite) mixture of Gaussians, we assume that the background density is of this form. We further assume that it is a mixture of $K-1$ equal probability $(1/K)$ components[1] and that $\pi = 1/K$.

Definition 6.1. Image I_i in the training set $\mathcal{D}$ is a sample from a random variable of probability density function

$$P^i_{\mathbf{X}}(\mathbf{x}) = \frac{1}{K}\left(P_c(\mathbf{x}) + \sum_{j=1}^{K-1} \mathcal{G}(\mathbf{x}, \mu^i_j, \mathbf{\Sigma}^i_j)\right). \tag{6.5}$$

The training set $\mathcal{D}$ is denoted *diverse* if the background distributions are themselves a diverse set. This can be formalized by making the Gaussian parameters μ^i_j and $\mathbf{\Sigma}^i_j$ samples from some random variable.

Definition 6.2. $\mathcal{D}$ is a diverse training set if μ^i_j, and $\mathbf{\Sigma}^i_j$ are independent samples from two independent random variables with probability density functions

$$P_\mu(\mu) = \mathcal{G}(\mu, \mu_0, \mathbf{\Sigma}_0)$$

and $P_{\mathbf{\Sigma}}(\mathbf{\Sigma})$, such that $E_{\mathbf{\Sigma}}[\Sigma] = \mathbf{S}$, and (for some $\epsilon \geq 0$)

$$|E_{\mathbf{\Sigma}}[\mathcal{G}(\mathbf{x}, \mu_0, \mathbf{\Sigma} + \mathbf{\Sigma}_0)] - \mathcal{G}(\mathbf{x}, \mu_0, \mathbf{S} + \mathbf{\Sigma}_0)]| \leq \epsilon. \tag{6.6}$$

The assumption of a Gaussian distribution for μ is not crucial for the discussion that follows. In particular, all results could be generalized to the case of a Gaussian mixture and, therefore, any $P_\mu(\mu)$ of practical interest. The Gaussian assumption makes the notation much simpler, and enables a simple characterization of the diversity of the means, by

[1] This is mostly to simplify notation, all results that follow could be extended to the case where each component has an individual weight.

making the differential entropy of μ a simple function of $\mathbf{\Sigma}_0$, namely

$$H(\mu) = \frac{d}{2}\ln(2\pi e) + \frac{1}{2}\ln|\det(\mathbf{\Sigma}_0)|. \tag{6.7}$$

We refer to $\mathbf{\Sigma}_0$ as the *diversity parameter* of $\mathcal{D}$. The condition of (6.6) is a technical condition, required by the proofs derived in the remainder of this section. We note, however, that it is a very mild restriction on $P_{\mathbf{\Sigma}}(\mathbf{\Sigma})$. If, for example, the Gaussian components of (6.5) are produced by a kernel density estimator, it is common practice for all covariances to be identical, i.e., $\mathbf{\Sigma}_j^i = \mathbf{S}$. In this case, $P_{\mathbf{\Sigma}}(\mathbf{\Sigma})$ is a delta function centered at $\mathbf{S}$, and (6.6) holds with $\epsilon = 0$. In general, the condition will hold if $\mathbf{\Sigma} + \mathbf{\Sigma}_0 \approx \mathbf{S} + \mathbf{\Sigma}_0$ for all $\mathbf{\Sigma}$ such that $P_{\mathbf{\Sigma}}(\mathbf{\Sigma}) > 0$, i.e., if the spread of $P_{\mathbf{\Sigma}}(\mathbf{\Sigma})$ around the mean value $\mathbf{S}$ is small compared to $\mathbf{S} + \mathbf{\Sigma}_0$. This is true whenever $\mathbf{\Sigma}_0$ is large, which (as we will see below) is a necessary condition for the concept distribution to be learnable. Note that, if the support of $P_{\mathbf{\Sigma}}(\mathbf{\Sigma})$ is bounded, it is possible, by making $\mathbf{\Sigma}_0$ arbitrarily large, to make (6.6) hold with arbitrarily small ϵ.

The following theorem shows that the distribution of a diverse set of images of concept c converges to a mixture of the concept distribution and a background component of spread determined by the diversity parameter $\mathbf{\Sigma}_0$.

Theorem 6.1. If $\mathcal{D}$ is a diverse training set, according to Definitions 6.1 and 6.2, and

$$P_N(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} P_{\mathbf{X}}^i(\mathbf{x}) \tag{6.8}$$

then, with probability one, for all $\mathbf{x}$

$$\lim_{N\to\infty} |P_N(\mathbf{x}) - f(\mathbf{x})| \leq \delta \tag{6.9}$$

with

$$f(\mathbf{x}) = \frac{1}{K}P_{\mathbf{C}}(\mathbf{x}) + \left(1 - \frac{1}{K}\right)\mathcal{G}(\mathbf{x}, \mu_0, \mathbf{S} + \mathbf{\Sigma}_0) \tag{6.10}$$

and

$$\delta = (1 - 1/K)\,\epsilon. \tag{6.11}$$

Proof. See the Appendix. □

The theorem enables a number of insights on the feasibility of learning the concept c under the weakly supervised learning paradigm. For a large training set, the empirical distribution learned from $\mathcal{D}$ will be a good approximation to $f(\mathbf{x})$. If the first component of (6.10) dominates the second, it follows that the empirical distribution will be close to the concept distribution for most values of $\mathbf{x}$. Hence, the *learnability* of the concept can be evaluated by measuring how dominant the first component is. One possible measure of such dominance is the ratio between the amplitudes of the concept and background components.

Definition 6.3. The learnability of concept c is

$$\kappa_c = \sup_{\mathbf{x}} \left(\frac{P_c(\mathbf{x})}{(K-1)\mathcal{G}(\mu_0, \mu_0, \mathbf{S} + \mathbf{\Sigma}_0)} \right) \tag{6.12}$$

If $\kappa_c \approx 1$, the concept component of (6.10) has amplitude similar to that of the background component and is not likely to be easy to identify. One the other hand, κ_c is large when (6.10) is a mixture of the concept component and a background component of much smaller amplitude.

This definition enables the formal characterization of a number of properties of concept learnability. We start by noting that the amplitude of the background distribution is

$$\mathcal{G}(\mu_0, \mu_0, \mathbf{S} + \mathbf{\Sigma}_0) = [(2\pi)^d \det(\mathbf{S} + \mathbf{\Sigma}_0)]^{-1/2}$$

and, from (6.7), $|\det(\mathbf{\Sigma}_0)| = (2\pi)^{-d} e^{2H(\mu)-d}$. It follows that, when $\mathbf{\Sigma}_0 \gg \mathbf{S}$,[2]

$$\mathcal{G}(\mu_0, \mu_0, \mathbf{S} + \mathbf{\Sigma}_0) \approx e^{d/2 - H(\mu)}.$$

Hence, it is possible to arbitrarily decrease the amplitude of the background distribution by raising its differential entropy. The interesting insight provided by the theorem is therefore that, no matter how small the percent of the area that it covers individually on each image ($1/K$),

[2] The notation $\mathbf{\Sigma}_0 \gg \mathbf{S}$ is equivalent to $\det(\mathbf{S} + \mathbf{\Sigma}_0) \approx \det(\mathbf{\Sigma}_0)$.

the concept is learnable if this entropy is sufficiently large. In fact, when $\mathbf{\Sigma}_0 \gg \mathbf{S}$,

$$\kappa_c = \frac{1}{(K-1)} e^{H(\mu)-d/2} \sup_{\mathbf{x}} P_c(\mathbf{x}), \tag{6.13}$$

and the linear decrease of learnability with concept area is dominated by an exponential increase with differential entropy.

The theorem also provides insight on how concept learnability depends on the dimension d of the feature space. Consider the case where $P_c(\mathbf{x})$ is a Gaussian of covariance $\sigma_c^2\mathbf{I}$, $\mathbf{S} = s^2\mathbf{I}$, and $\mathbf{\Sigma}_0 = \sigma_0^2\mathbf{I}$. Then, from (6.12),

$$\kappa_c = \frac{1}{K-1} \left(\frac{s^2 + \sigma_0^2}{\sigma_c^2} \right)^{d/2}$$

and, as long as $s^2 + \sigma_0^2 > \sigma_c^2$, the learnability of c increases exponentially with d. That is, *concepts become exponentially easier to learn as the dimension of the space increases*. Note, once again, that the linear decrease of learnability with the decrease of the image area covered by the concept is overwhelmed by this exponential dependence on dimensionality. The case of a non-Gaussian concept is more difficult to analyze, but qualitatively similar. Note that, under the assumption that $\sup_{\mathbf{x}} P_c(\mathbf{x})$ decreases exponentially with d (due to the requirement that $P_c(\mathbf{x})$ integrates to one),

$$\sup_{\mathbf{x}} P_c(\mathbf{x}) = O(a^{-d})$$

with $a > 0$, it follows from (6.12) that

$$\kappa_c \propto O\left(\left[\frac{\sqrt{2\pi(s^2 + \sigma_0^2)}}{a} \right]^d \right)$$

and, as long as $a < \sqrt{2\pi(s^2 + \sigma_0^2)}$, *concept learnability increases exponentially with the dimension d*.

In summary, concepts can be learned with weak supervision if (1) the background distribution is diverse enough, or (2) the dimension of the feature space is large enough. Furthermore, learnability

increases exponentially with these two parameters. This effect dominates any potential decrease in learnability due to the limited area of the concept within the training images. Weakly supervised learning is thus, in principle, possible even for concepts that occupy a small area in all training images. It suffices that concept training sets are large, present the concept against a large diversity of backgrounds, and learning is performed on high-dimensional feature spaces.

7

Query By Semantic Example

In the previous sections we have studied two retrieval paradigms: one based on visual queries, denoted as *query-by-visual-example* (QBVE), and the other based on text, denoted as *semantic retrieval* (SR). Under QBVE, each image is decomposed into a number of low-level visual features and image retrieval is formulated as the search for the MPE match to the collection of feature vectors extracted from a query image. Under SR, images are annotated with semantic keywords, enabling users to specify their queries through a natural language description of the visual concepts of interest. Both paradigms have their strengths and weaknesses. SR has the advantage of evaluating image similarity at a higher level of abstraction and, therefore, better generalization than what is possible with QBVE. On the other hand, the performance of SR systems tends to degrade for semantic classes that they were not trained to recognize. Since it is still difficult to learn appearance models for massive concept vocabularies, this could compromise the generalization gains due to abstraction. This problem is seldom considered in the literature, where most evaluations are performed with query concepts that are known to the retrieval system [5, 13, 16, 31, 35, 61].

In fact, it is not even straightforward to compare the two retrieval paradigms because they assume different levels of query specification. While a semantic query is usually precise (e.g., 'the White House'), a visual example (a picture of the 'White House') will depict various concepts that are irrelevant to the query (e.g., the street that surrounds the building, cars, people, etc.). It is, therefore, possible that better SR results could be due to a better interface (natural language) rather than an intrinsic advantage of representing images semantically. In this section, we introduce a framework for the objective comparison of the two formulations, by extending the query-by-example paradigm to the semantic domain. This consists of defining a semantic feature space, where each image is represented by the vector of posterior concept probabilities assigned to it by an SR system, and performing query-by-example in this space. We refer to the combination of the two paradigms as *query-by-semantic-example* (QBSE), and present a comparison of its performance with that of QBVE. It is shown that QBSE has significantly better performance for both concepts known and unknown to the retrieval system, i.e., it can generalize beyond the vocabulary used for training. It is also shown that the performance gain is intrinsic to the semantic nature of image representation.

7.1 Query by Visual Example vs Semantic Retrieval

Both QBVE and SR have advantages and limitations. Because concepts are learned from collections of images, SR can *generalize* significantly better than QBVE. For example, by using a large training set of images labeled with the concept 'sky', containing both images of sky at daytime (when it is mostly blue) and sunsets (when it is mostly orange), an SR system can learn that 'sky' is sometimes blue and others orange. This is not easy to accomplish with QBVE, which only has access to two images (the query and that in the database) and can only perform direct matching of visual features. We refer to this type of abstraction as *generalization inside the semantic space*, i.e., inside the space of concepts that the system has been trained to recognize.

While better generalization is a strong advantage for SR, there are some limitations associated with this paradigm. An obvious difficulty

Fig. 7.1 An image containing various concepts: 'train', 'smoke', 'road', 'sky', 'railroad', 'sign', 'trees', 'mountain', 'shadows', with variable degrees of presence.

is that most images have multiple semantic interpretations. Figure 7.1 presents an example, identifying various semantic concepts as sensible annotations for the image shown. Note that this list, of relatively salient concepts, is a small portion of the keywords that could be attached to the image. Other examples include colors (e.g., 'yellow' train), or objects that are not salient in an abstract sense but could become very relevant in some contexts (e.g., the 'paint' of the markings on the street, the 'letters' in the sign, etc.). In general, it is impossible to predict all annotations that may be relevant for a given image. This is likely to compromise the performance of an SR system. Furthermore, because queries are specified as text, an SR system is usually limited by the size of its vocabulary.[1] In summary, SR can generalize poorly *outside the semantic space*.

[1] It is, of course, always possible to rely on text processing ideas based on thesauri and ontologies like WordNet [34] to mitigate this problem. For example, query expansion can be used to replace a query for 'pollution' by a query for 'smoke', if the latter is in the vocabulary and the former is not. While such techniques are undeniably useful for practical implementation of retrieval systems, they do not reflect an improved ability, by the retrieval system, to model the relationships between visual features and words. They are simply an attempt to fix these limitations a posteriori (i.e., at the language level). In practice, it is not always easy to perform text-based query expansion when the vocabulary is small, as is the case for most SR systems, or when the queries report to specific instances (e.g., a person's name).

Since visual retrieval has no notion of semantics, it is not constrained by either vocabulary or semantic interpretations. When compared to SR, QBVE systems can generalize better outside the semantic space. In the example of Figure 7.1, a QBVE would likely return the image shown as a match to a query depicting an industrial chimney engulfed in dark smoke (a more or less obvious query prototype for images of 'pollution') despite the fact that the retrieval system knows nothing about 'smoke', 'pollution', or 'chimneys'. Obviously, there are numerous examples where QBVE correlates much worse with perceptual similarity than SR. We have already seen that when the latter is feasible, i.e., inside the semantic space, it has better generalization. Overall, it is sensible to expect that SR will perform better inside the semantic space, while QBVE should fare better outside of it. QBSE aims to achieve good generalization both within and outside the semantic space.

7.2 Query by Semantic Example

A QBSE system operates at the semantic level, representing an image $\mathcal{I}$ by a vector of concept counts $\mathcal{C} = (c_1, \ldots, c_L)^T$. Each feature vector $\mathbf{x}_i$ of the image is assumed to be sampled from the probability distribution of a semantic class (concept). Concept probabilities are learned with a semantic labeling system, as discussed in the previous sections, and the probability of the ith concept, given the observed feature vectors in $\mathcal{I}$, is

$$\pi^i = P_{W|\mathbf{X}}(i|\mathcal{I}). \tag{7.1}$$

c_i is the number of feature vectors drawn from the ith concept. The count vector for the yth image is drawn from a multinomial variable $\mathbf{T}$ of parameters $\pi_y = (\pi_y^1, \ldots, \pi_y^L)^T$

$$P_{\mathbf{T}|Y}(\mathcal{C}|y;\pi_y) = \frac{n!}{\prod_{k=1}^{L} c_k!} \prod_{j=1}^{L} (\pi_y^j)^{c_j}. \tag{7.2}$$

The random variable $\mathbf{T}$ can be seen as the result of a feature transformation from the space of visual features $\mathcal{X}$ to the L-dimensional probability simplex $\mathcal{S}_L$. This mapping, $\mathbf{\Pi}: \mathcal{X} \to \mathcal{S}_L$ such that $\mathbf{\Pi}(\mathbf{X}) = \mathbf{T}$, establishes a correspondence between images and points $\pi_y \in \mathcal{S}_L$, as illustrated by Figure 1.7. Since the entries of π_y

are the posterior probabilities of the semantic concepts $\omega_i, i = 1, \ldots, L$ given the yth image, we refer to the probability simplex $\mathcal{S}_L$ as the *semantic simplex*, and to the probability vector π_y itself as the *semantic multinomial* (SMN) that characterizes the image.

7.3 The Semantic Multinomial

As is usual in probability estimation, the posterior concept probabilities of (7.1) can be inaccurate for concepts with a small number of training images. Of particular concern are cases where some of the π_i are very close to zero and can become ill-conditioned during retrieval, where noisy estimates are amplified by ratios or logs of probabilities. A common solution is to introduce a prior distribution to regularize these parameters. For this, it is worth considering an alternative procedure for the estimation of the π_i. Instead of (7.1), this consists of computing the posterior concept probabilities $P_{W|\mathbf{X}}(w|\mathbf{x}_k), w \in \{1, \ldots, L\}$ of *each* feature vector $\mathbf{x}_k$, assign $\mathbf{x}_k$ to the concept of largest probability, and count the number c_w of vectors assigned to each concept. The maximum likelihood estimate of the probabilities is then given by [30]

$$\pi_w^{\mathrm{ML}} = \arg\max_{\pi_w} \prod_{j=1}^{L} \pi_j^{c_j} = \frac{c_w}{\sum_j c_j} = \frac{c_w}{n}. \tag{7.3}$$

Regularization can then be enforced by adopting a Bayesian parameter estimation viewpoint, where the parameter π is considered a random variable and a prior distribution $P_{\mathbf{\Pi}}(\pi)$ introduced to favor parameter configurations that are, a priori, more likely.

Conjugate priors are frequently used, in Bayesian statistics [41], to estimate parameters of distributions in the exponential family, as is the case of the multinomial. They lead to a closed-form posterior (which is in the family of the prior), and *maximum a posteriori probability* parameter estimates which are intuitive. The conjugate prior of the multinomial is the Dirichlet distribution

$$\pi \sim \mathbf{Dir}(\alpha) = \frac{\Gamma\left(\sum_j^L \alpha_j\right)}{\prod_{j=1}^L \Gamma(\alpha_j)} \prod_{j=1}^{L} \pi_j^{\alpha_j - 1} \tag{7.4}$$

of *hyper-parameters* α_i, and where $\Gamma(.)$ is the Gamma function. Setting[2] $\alpha_i = \alpha$, the maximum aposteriori probability estimates are

$$\begin{aligned}\pi_w^{\text{posterior}} &= \arg\max_{\pi_w} P_{\mathbf{T}|\mathbf{\Pi}}(c_1,\ldots,c_L|\pi)P_{\mathbf{\Pi}}(\pi) \\ &= \arg\max_{\pi_w} \prod_{j=1}^{L} \pi_j^{c_j} \prod_{j=1}^{L} \pi_j^{\alpha-1} \\ &= \frac{c_w + \alpha - 1}{\sum_{j=1}^{L}(c_j + \alpha - 1)}. \end{aligned} \tag{7.5}$$

This is identical to the maximum likelihood estimates obtained from a sample where each count is augmented by $\alpha - 1$, i.e., where each image contains $\alpha - 1$ more feature vectors from each concept. The addition of these vectors prevents zero counts, regularizing π. As α increases, the multinomial distribution tends to uniform.

Thresholding the individual feature vector posteriors and counting is likely to produce worse probability estimates than those obtained, with (7.1), directly from the entire collection of feature vectors. Nevertheless, the discussion above suggests a strategy to regularize the probabilities of (7.1). Noting, from (7.3), that $c_w = n\pi_w^{\text{ML}}$, the regularized estimates of (7.5) can be written as

$$\pi_w^{\text{posterior}} = \frac{\pi_w^{\text{ML}} + \pi_0}{\sum_j^L (\pi_j^{\text{ML}} + \pi_0)},$$

with $\pi_0 = \frac{\alpha-1}{n}$. Hence, regularizing the estimates of (7.1) with

$$\pi_w^{\text{reg}} = \frac{\pi_w + \pi_0}{1 + L\pi_0} \tag{7.6}$$

is equivalent to using maximum aposteriori probability estimates, in the thresholding plus counting paradigm, with the Dirichlet prior of (7.4). We have found that values of $L\pi_0 \in [0.001, 0.1]$ perform best in retrieval experiments (see [91] for details).

[2] Different hyper-parameters could also be used for the different concepts.

7.4 Image Similarity

A QBSE system operates on the simplex $\mathcal{S}_L$, according to a similarity mapping $f : \mathcal{S}_L \to \{1, \ldots, D\}$ such that

$$f(\pi) = \arg\max_y s(\pi, \pi_y), \tag{7.7}$$

where π is the query SMN, π_y the SMN that characterizes the yth database image, and $s(\cdot, \cdot)$ an appropriate similarity function.

We have compared various similarity functions. The KL divergence between two semantic multinomials π and π' is

$$s_{\mathrm{KL}}(\pi, \pi') = \mathrm{KL}(\pi||\pi') = \sum_{i=1}^{L} \pi_i \log \frac{\pi_i}{\pi'_i}. \tag{7.8}$$

We have already seen that it is the asymptotic limit of (4.1) when Y is uniformly distributed. A symmetric version can be defined as

$$s_{\mathrm{symmKL}}(\pi, \pi') = \mathrm{KL}(\pi||\pi') + \mathrm{KL}(\pi'||\pi) \tag{7.9}$$

$$= \sum_{i=1}^{L} \pi_i \log \frac{\pi_i}{\pi'_i} + \sum_{i=1}^{L} \pi'_i \log \frac{\pi'_i}{\pi_i}. \tag{7.10}$$

The Jensen–Shannon divergence (JS) is a measure of whether two samples, as defined by their empirical distributions, are drawn from the same source distribution [22]. It is defined as

$$s_{\mathrm{JS}}(\pi, \pi') = KL(\pi||\hat{\pi}) + KL(\pi'||\hat{\pi}), \tag{7.11}$$

where $\hat{\pi} = \frac{1}{2}\pi + \frac{1}{2}\pi'$. This divergence can be interpreted as the average distance (in the KL sense) between each distribution and the average of all distributions.

It is also possible to rely on L^p distances

$$s_{L^p}(\pi, \pi') = \left(\sum_{i=1}^{L} |\pi_i - \pi'_i|^p \right)^{\frac{1}{p}}, \quad p \geq 1. \tag{7.12}$$

For $p = 2$, we have the Euclidean distance, whose minimization is equivalent to maximizing the correlation

$$s_{\mathrm{CO}}(\pi, \pi') = \pi^T \pi' = \sum_{i}^{L} \pi_i \times \pi'_i \tag{7.13}$$

whenever $||\pi|| = 1$. This is not the case for semantic multinomials, which motivates an alternative correlation measure, the *normalized correlation*,

$$s_{\text{NC}}(\pi, \pi') = \frac{\pi^T \pi'}{||\pi||||\pi'||} = \frac{\sum_i^L \pi_i \times \pi'_i}{\sqrt{\sum \pi_j^2}\sqrt{\sum \pi_j'^2}}. \tag{7.14}$$

Finally, we have already seen that the minimization of the L^1 norm is equivalent to the maximization of the histogram intersection (HI) [112],

$$s_{\text{HI}}(\pi, \pi') = \sum_{i=1}^{L} \min(\pi_i, \pi'_i). \tag{7.15}$$

7.5 Properties of QBSE

As a query paradigm, QBSE has a number of interesting properties. First, the mapping of the visual features to the probability simplex $\mathcal{S}_L$ can be seen as an abstract mapping of the image to a *semantic space*, where each concept probability $\pi_y^i, i = 1, \ldots, L$, is a *semantic feature*. Semantic features, or concepts, outside the vocabulary simply define directions orthogonal to the learned semantic space. This implies that, by projecting these dimensions onto the simplex, the QBSE system can generalize beyond the known semantic concepts. In the example of Figure 7.1, the mapping of the image onto the semantic simplex assigns high probability to (known) concepts such as 'train', 'smoke', 'railroad', etc. This makes the image a good match for other images containing large amounts of 'smoke', such as those depicting industrial chimneys or 'pollution' in general. The system can therefore establish a link between the image of Figure 7.1 and 'pollution', despite the fact that it has no *explicit* knowledge of the 'pollution' concept.[3] Second, when compared to QBVE, QBSE complements all the advantages of query by example with the advantages of a semantic representation. Moreover, since in both cases queries are specified by the same examples, any differences in their performance can be directly attributed to the semantic vs.

[3] Note that this is different from text-based query expansion, where the link between 'smoke' and 'pollution' must be *explicitly* defined. In QBSE, the relationship is instead inferred automatically, from the fact that both concepts have commonalities of visual appearance.

visual nature of the associated image representations.[4] This enables the objective comparison of QBVE and QBSE.

7.6 Multiple Image Queries

Semantic image labeling is, almost by definition, a noisy endeavor. This is a consequence of the fact that various interpretations are usually possible for a given arrangement of image intensities. An example is given in Figure 7.2, where we show the query image of Figure 1.3 and the associated SMN. While most of the probability mass is assigned to concepts that are present in the image ('railroad', 'locomotive', 'train', 'street', or 'sky'), two of the concepts of largest probability are 'bridge' and 'arch'. We already saw that the locomotive's roof resembles the arch of a bridge. This visual feature seems to be highly discriminant since, when used as a query in a QBVE system, most of the top matches are images with arch-like structures, not trains (see Figure 1.3). While these types of errors are difficult to avoid, they are *accidental*. In particular, the arch-like structure of Figure 7.2 is the result of viewing a particular

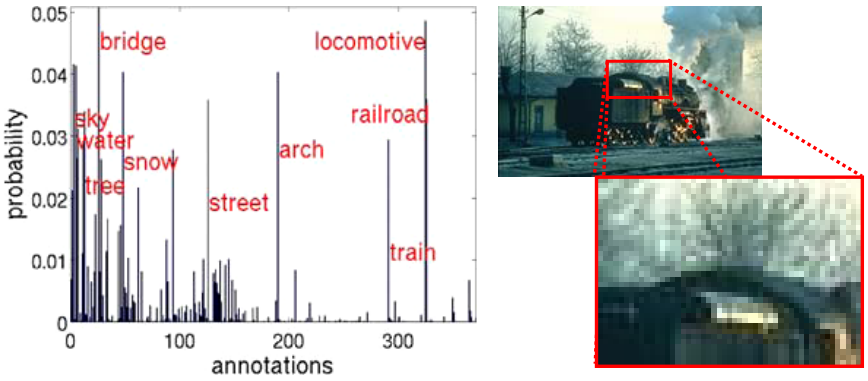


Fig. 7.2 An image and its associated SMN. Note that, while most of the concepts of largest probability are present in the image, the SMN assigns significant probability to 'bridge' and 'arch'. These are due to the geometric structure shown on the image close-up.

[4] This assumes, of course, that a common framework, such as MPE, is used to implement both the QBSE and QBVE systems.

type of train, at a particular viewing angle, and a particular distance. It is unlikely that similar structures will emerge consistently over a set of train images. A pressing question is then whether it is possible to exploit the lack of consistency of these errors to obtain a better characterization of the query image set?

Once again, we resort to the *multiple instance* learning paradigm, formulating the problem as one of learning from *bags of examples*. In QBSE, each image is modeled as a bag of feature vectors, which are drawn from the different concepts according to the probabilities π_i. When the query consists of multiple images, or bags, the negative examples that appear across those bags are inconsistent (e.g., the feature vectors associated with the arch-like structure which is prominent in Figure 7.2), and tend to be spread over the feature space (because they also depict background concepts, such as roads, trees, mountains, etc., which vary from image to image). On the other hand, feature vectors corresponding to positive examples are likely to be concentrated within a small region of the space. It follows that, although the distribution of positive examples may not be dominant in any individual bag, the consistent appearance in all bags makes it dominant over the entire query ensemble. This suggests that a better estimate of the query SMN should be possible by considering a set of multiple query images.

Under MPE retrieval, query combination is relatively straightforward to implement by QBVE systems. Given two query images $\mathcal{I}_q^1 = \{\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_n^1\}$ and $\mathcal{I}_q^2 = \{\mathbf{x}_1^2, \mathbf{x}_2^2, \ldots, \mathbf{x}_n^2\}$, the probability of the composite query $\mathcal{I}_q^C = \{\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_n^1, \mathbf{x}_1^2, \mathbf{x}_2^2, \ldots, \mathbf{x}_n^2\}$ given class $Y = y$ is

$$\begin{aligned} P_{\mathbf{X}|Y}(\mathcal{I}_q^C|y) &= \prod_{k=1}^{n} P_{\mathbf{X}|Y}(\mathbf{x}_k^1|y) \prod_{l=1}^{n} P_{\mathbf{X}|Y}(\mathbf{x}_l^2|y) \\ &= P_{\mathbf{X}|Y}(\mathcal{I}_q^1|y) P_{\mathbf{X}|Y}(\mathcal{I}_q^2|y). \end{aligned} \tag{7.16}$$

The MPE decision for the composite query is obtained by using the Gaussian mixture of (4.8) as $P_{\mathbf{X}|Y}(x|y)$ in (7.16), and combining with (2.3). Under QBSE, there are at least three possibilities for query combination. The first is equivalent to (7.16), but based on the

probability of the composite query $\mathcal{I}_q^C$ given semantic class $W = w$,

$$\begin{aligned} P_{\mathbf{X}|W}(\mathcal{I}_q^C|w) &= \prod_{k=1}^{n} P_{\mathbf{X}|W}(\mathbf{x}_k^1|w) \prod_{l=1}^{n} P_{\mathbf{X}|W}(\mathbf{x}_l^2|w) \\ &= P_{\mathbf{X}|W}(\mathcal{I}_q^1|w) P_{\mathbf{X}|W}(\mathcal{I}_q^2|w), \end{aligned} \tag{7.17}$$

which is combined with (5.13) and Bayes rule to compute the posterior concept probabilities of (7.1). We refer to (7.17) as the 'LKLD combination' strategy for query combination. It is equivalent to taking a geometric mean of the probabilities of the individual images given the class.

A second possibility is to represent the query as a mixture of SMNs. This relies on a different generative model than that of (7.17): the ith query is first selected with probability λ_i and a count vector is then sampled from the associated multinomial distribution. It can be formalized as

$$P_{\mathbf{T}}(\mathcal{C}_q^C; \pi_q) = \frac{n!}{\prod_{k=1}^{L} c_k!} \prod_{j=1}^{L} (\lambda_1 \pi_1^j + \lambda_2 \pi_2^j)^{c_j}, \tag{7.18}$$

where $P_{\mathbf{T}}(\mathcal{C}_q^C; \pi_q)$ is the multinomial distribution for the query combination, of parameter $\pi_q = \lambda_1 \pi_1 + \lambda_2 \pi_2$. π_1 and π_2 are the parameters of the individual multinomial distribution, and $\lambda = (\lambda_1, \lambda_2)^T$ the vector of query selection probabilities. If $\lambda_1 = \lambda_2$, the two SMNs are simply averaged. We adopt the uniform query selection prior, and refer to this strategy as 'SMN combination'. Geometrically, it sets the combined SMN to the centroid of the simplex that has the SMNs of the query images as vertices. This ranks highest the database SMN which is closest to this centroid.

The third possibility, henceforth referred to as 'KL combination', is to execute the multiple queries separately, and combine the resulting image rankings. For example, when similarity is measured with the KL divergence, the divergence between the combined image SMN, π_q, and database SMNs, π_y, is

$$s_{\mathrm{KL}}(\pi_q, \pi_y) = \frac{1}{2} \mathrm{KL}(\pi_1 || \pi_y) + \frac{1}{2} \mathrm{KL}(\pi_2 || \pi_y). \tag{7.19}$$

It is worth noting that this combination strategy is closely related to that used in QBVE. Note that the use of (7.16) is equivalent to using the arithmetic average (mean) of log-probabilities which, in turn, is identical to combining image rankings, as in (7.19). For QBVE, the two combination approaches are identical.

7.7 Experimental Evaluation

To evaluate QBSE, we have used the image annotation system trained with the Corel5K set-up of Section 5. Overall, there are 371 keywords in the data set, leading to a 371-dimensional semantic simplex. With respect to image representation, all images were normalized to size 181×117 or 117×181 and converted from RGB to the YBR color space. Image observations were derived from 8×8 patches obtained with a sliding window, moved in a raster-scan fashion. A feature transformation was applied to this space by computing the 8×8 DCT of the three color components of each patch. The parameters of the semantic class mixture hierarchies were learned in the subspace of the resulting 192-dimension feature space composed of the first 21 DCT coefficients from each channel. In all experiments, the SMN associated with each image was computed with these semantic class-conditional distributions. Various datasets were used as query and retrieval databases, so as to enable the evaluation of retrieval performance both inside and outside the semantic space.

Inside the semantic space. Retrieval performance inside the semantic space was evaluated by using *Corel5K* (here referred to as *Corel50*) as both retrieval and query database. More precisely, the 4,500 training images served as the *retrieval database* and the remaining 500 as the *query database*. This experiment relied on clear ground-truth regarding the relevance of the retrieved images, based on the theme of the CD to which the query belonged.

Outside the semantic space. To test performance outside the semantic space, we relied on two additional databases. The first, *Corel15*,

consisted of 1500 images from 15[5] previously unused Corel CDs. Again, the CD themes (non-overlapping with those of *Corel50*) served as the ground truth. We also collected a database from `www.flickr.com`. The images in this database were extracted by placing queries on the flickr search engine, and manually pruning images that appeared irrelevant to the specified queries. Note that the judgments of relevance did not take into account how well a content-based retrieval system would perform on the images, simply whether they appeared to be search errors (by flickr) or not. The images are shot by flickr users, and hence differ from the Corel Stock photos, which have been shot professionally. This database, *Flickr18*, contains 1,800 images divided into 18 classes according to the manual annotations provided by the online users. For both databases, 20% of randomly selected images served as *query images* and the remaining 80% as the *retrieval database*.

QBVE only requires a query and a retrieval database. In all experiments, these were made identical to the query and retrieval databases used by QBSE. Since the performance of QBVE does not depend on whether queries are inside or outside the semantic space, this establishes a benchmark for evaluating the generalization of QBSE.

7.7.1 Effect of the Similarity Function

Table 7.1 presents a comparison of the seven similarity functions discussed in the text. It is clear that L^2 distance and histogram intersection do not perform well. All information theoretic measures, KL divergence, symmetric KL divergence, and Jensen–Shannon divergence have superior performance, with an average improvement of 15%. Among these, the KL divergence performs the best. Its closest competitors are the correlation and normalized correlation metrics. Although they outperform KL divergence inside the semantic space (*Corel50*), their performance is inferior for databases outside the semantic space (*Flickr18, Corel15*). This suggests that the KL divergence has better generalization. We thus adopted the KL divergence for all remaining experiments.

[5] 'Adventure Sailing', 'Autumn', 'Barnyard Animals', 'Caves', 'Cities of Italy', 'Commercial Construction', 'Food', 'Greece', 'Helicopters', 'Military Vehicles', 'New Zealand', 'People of World', 'Residential Interiors', 'Sacred Places', and 'Soldier'.

Table 7.1. Effect of the similarity function on the MAP score of QBSE.

Similarity function	MAP score		
	Corel50	Corel15	Flickr18
KL divergence	0.1768	**0.2175**	**0.1615**
Symmetric KL	0.1733	0.2164	0.1602
Jensen–Shannon	0.1740	0.2158	0.1611
Correlation	**0.2108**	0.1727	0.1392
Normalized correlation	0.1938	0.2041	0.1595
L2 distance	0.1461	0.1830	0.1408
Histogram intersection	0.1692	0.2119	0.1600

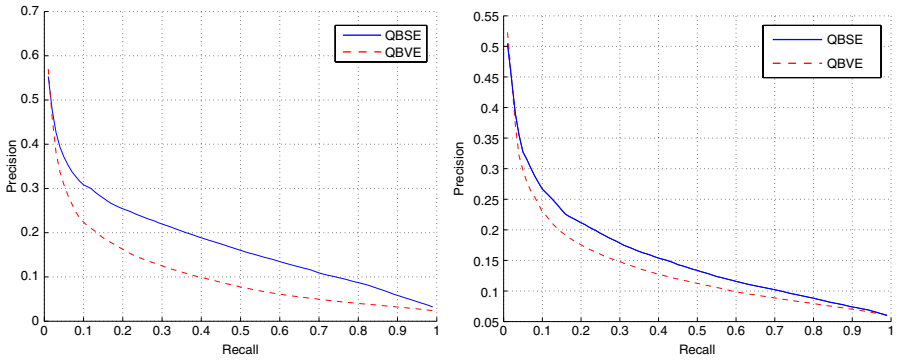


Fig. 7.3 Average precision–recall of single-query QBSE and QBVE; left: inside the semantic space (*Corel50*); right: outside the semantic space (*Flickr18*).

7.7.2 Performance Within the Semantic Space

Figure 7.3 (left) presents the PR curves on *Corel50* with QBVE and QBSE. The precision of QBSE is significantly higher than that of QBVE, at most levels of recall. At low recall, there are always some database images which are visually similar to the query and QBVE is competitive with QBSE. However, performance decreases much more dramatically than that of QBSE as recall increases, confirming the better generalization of the latter. The MAP scores for QBSE and QBVE are 0.1665 and 0.1094, respectively, and the chance MAP performance is 0.0200. Figure 7.4 shows that QBSE outperforms QBVE for almost all classes.

The advantages of QBSE are also illustrated by Figure 7.5, where we present the results of some queries, under both QBVE and QBSE.

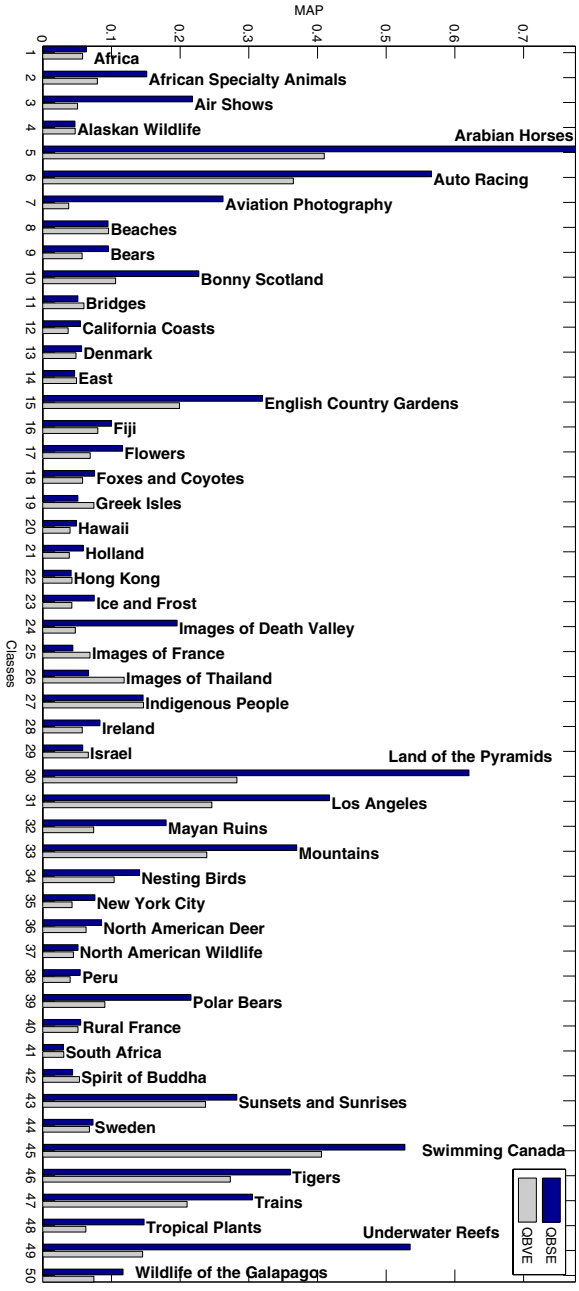


Fig. 7.4 MAP scores of QBSE and QBVE across the 50 classes of *Corel50*.

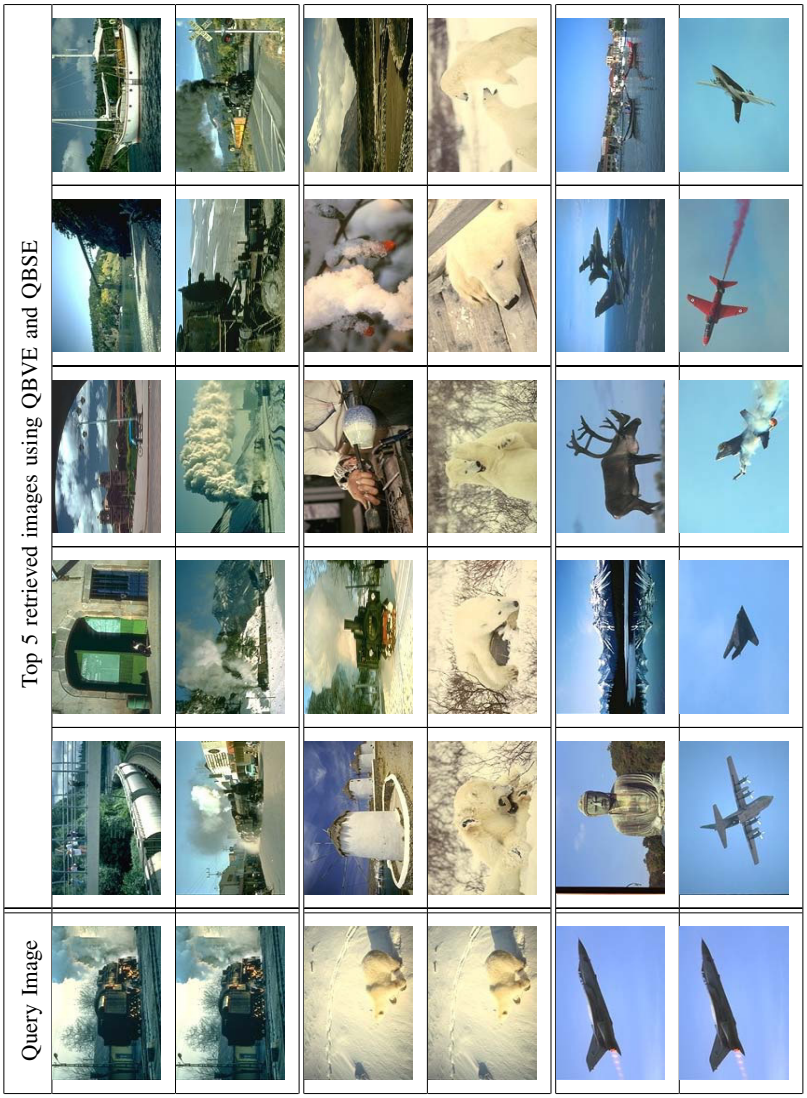


Fig. 7.5 Some examples where QBSE performs better than QBVE. The second row of every query shows the images retrieved by QBSE.

Note, for example, that for the query containing "white smoke" and a large area of "dark train", QBVE tends to retrieve images with *whitish* components, mixed with *dark* components, that have little connection to the "train" theme. Furthermore, the arch-like structure highlighted

in Figure 7.2 seems to play a prominent role in visual similarity since three of the five top matches contain arches. Due to its higher level of abstraction, QBSE is successfully able to generalize the main semantic concepts of "train", "smoke" and "sky", realizing that the white color is an irrelevant attribute to this query (as can be seen in the last column, where an image of a "train with black smoke" is successfully retrieved).

7.7.3 Multiple Image Queries

Figure 7.6 (left) shows the MAP values for multiple image queries, as a function of query cardinality, under both QBVE and QBSE for *Corel50*. In the case of QBSE, we also compare the three possible query combination strategies: '*LKLD*', '*SMN*', and '*KL Combination*'. It is clear that, inside the semantic space, the gains achieved with multiple QBSE queries are unparalleled on the visual domain. Among the various combination methods, combining SMNs yields best results, with a gain of 29.8% over single image queries. '*LKLD*' and '*KL Combination*' exhibit a gain of 17.3% and 26.4%, respectively.

For QBSE-SMN, MAP increases with query cardinality for 76% of the classes. For the remaining classes, poor performance can be explained by (1) significant inter-concept overlap (e.g., 'Air Shows' vs. 'Aviation Photography'), (2) incongruous concepts that would be difficult even for a human labeler (e.g., 'Holland' and 'Denmark'), or

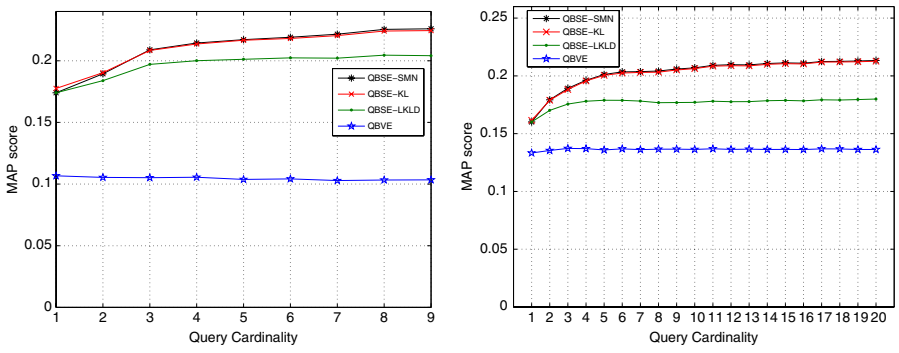


Fig. 7.6 MAP as a function of query cardinality for multiple image queries. Comparison of QBSE, with various combination strategies, and QBVE. Left: inside the semantic space (*Corel50*); right: outside the semantic space (*Flickr18*).

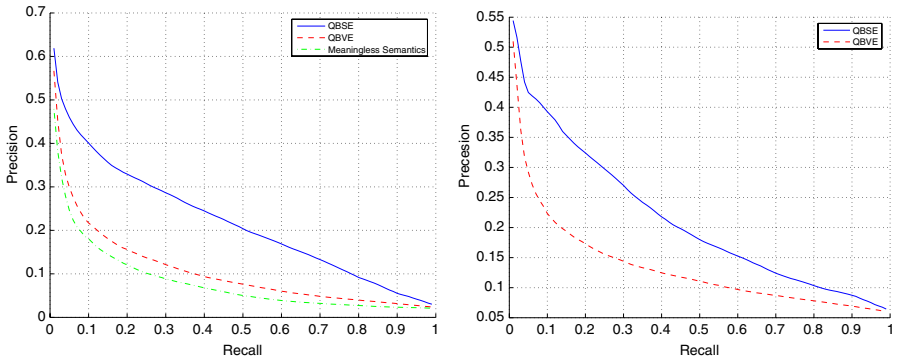


Fig. 7.7 Best precision–recall curves achieved with QBSE and QBVE on *Corel50*. Left: inside the semantic space (*Corel50*), also shown is the performance with meaningless semantic space. Right: outside the semantic space (*Flickr18*).

(3) failure to learn semantic homogeneity among the images, e.g., 'Spirit of Buddha'. Nevertheless, for 86% of the classes QBSE outperforms QBVE by an average MAP score of 0.136. On the remaining QBVE is only marginally better than QBSE, by an average MAP score of 0.016. Figure 7.7 (left) presents the average precision–recall curves, obtained with the number of image queries that performed best, for QBSE and QBVE on *Corel50*. It is clear that QBSE significantly outperforms QBVE at all levels of recall, the average MAP gain being of 111.73%.

7.7.4 Performance outside the semantic space

Figure 7.3 (right) presents PR curves on *Flickr18*, showing that outside the semantic space single-query QBSE is marginally better than QBVE. When combined with Figure 7.3 (left), it confirms that, overall, single-query QBSE has better generalization than visual similarity: it is substantially better inside the semantic space, and has slightly better performance outside of it. As was the case for *Corel50*, multiple image queries benefit QBSE substantially but have no advantage for QBVE. This is shown in Figure 7.6 (right). Regarding combination strategies, 'SMN' once again outperforms 'KL' (slightly) and 'LKLD Combination' (significantly).

An illustration of the benefits of multiple image queries is given in Figure 7.8. The two top rows present query images from the class

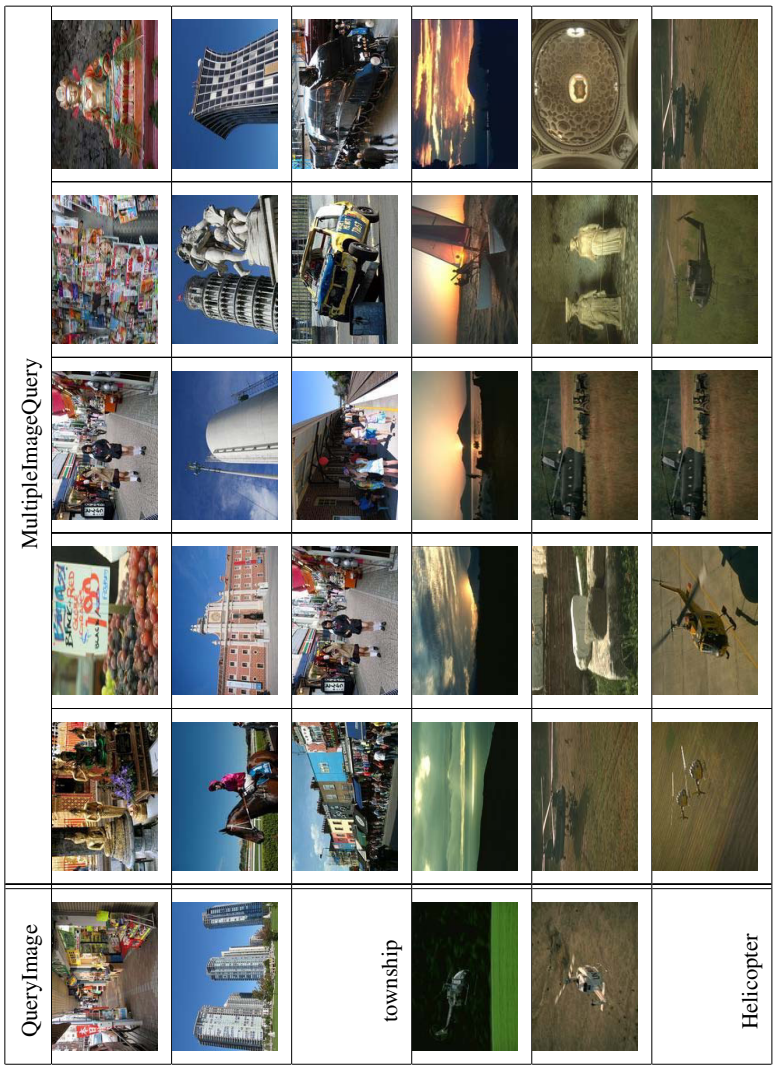


Fig. 7.8 Examples of multiple-image QBSE queries. Two queries (for "Township" and "Helicopter") are shown, each combining two examples. In each case, two top rows presents the single-image QBSE results, while the third presents the combined query.

'Township' (*Flickr18*) and single-query QBSE retrieval results. The third row presents the result of combining the two queries by 'SMN *combination*'. It illustrates the wide variability of visual appearance of the images in the 'Township' class. While single-image queries

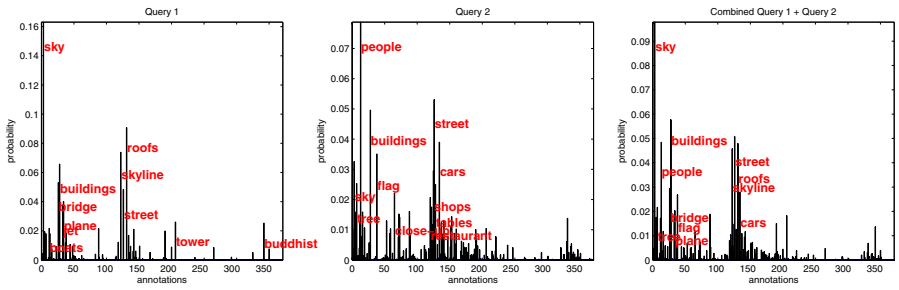


Fig. 7.9 SMN of individual and combined queries from class 'Township' of Figure 7.8. Left column shows the first query SMN, center the second and, right the combined query SMN.

Table 7.2. MAP of QBVE and QBSE on all datasets considered.

Database	Chance	QBVE	QBSE	% increase
Corel50	0.0200	0.1067	0.2259	111.73
Corel15	0.0667	0.2176	0.2980	36.95
Flickr18	0.0556	0.1373	0.2134	55.47

fail to express the semantic richness of the class, the combination of the two images allows the QBSE system to expand 'indoor market scene' and 'buildings in open air' to an 'open market street' or even a 'railway platform'. This is revealed, by the SMN of the combined query, presented in Figure 7.9 (right), which is a semantically richer description of the visual concept 'Township', containing concepts (like 'sky', 'people', 'street', 'skyline') from both individual query SMNs. The remaining three rows of Figure 7.8 present a similar result for the class 'Helicopter' (*Corel15*).

Finally, Figure 7.7 presents the best results obtained with multiple queries under both the QBSE and QBVE paradigms. It shows that QBSE significantly outperforms QBVE, even outside the semantic space. Table 7.2 summarizes the MAP gains of QBSE, over QBVE, for all datasets considered. In *Flickr18*, the gain is of 55.47%.

Overall, QBSE significantly outperforms QBVE, both inside and outside the semantic space. Since the basic visual representation (DCT features and Gaussian mixtures) is shared by the two approaches, this is a strong indication that *there is a benefit* to the use of semantic representations in image retrieval. To further investigate this hypothesis,

we performed a final experiment based on QBSE with a semantically meaningless space. Building on the fact that all semantic models are learned by grouping images with a common semantic concept, this was achieved by replicating the QBSE experiments with random image groupings. That is, instead of a semantic space composed of concepts like 'sky' (learned from images containing sky), we created a 'semantic space' of nameless concepts learned from random collections of images. Figure 7.7 (left) compares (on *Corel50*) the precision–recall obtained with QBSE on this 'meaningless semantic space', with the previous results of QBVE and QBSE. It is clear that, in the absence of semantic structure, QBSE has *very poor* performance, and is *clearly inferior* to QBVE.

8

Conclusions

In this monograph, we have reviewed the MPE principle for image retrieval, and shown how it can be used to design optimal solutions for practical retrieval problems. We have characterized the fundamental performance bounds of the MPE retrieval architecture, and used these bounds to derive optimal components for retrieval systems. We have also shown that many alternative formulations of the retrieval problem are closely related to the MPE principle, typically resulting from simplifications or approximations to the MPE architecture. The MPE principle was then applied to the design of retrieval systems that work at different levels of abstraction. QBVE systems are strictly visual, matching images by similarity of low-level features, such as texture or color. With MPE image labeling techniques, it is possible to start representing images in terms of more abstract visual concepts, producing semantic descriptions. QBSE represents images in the resulting concept spaces, enabling example-based retrieval by similarity of semantics.

We finish by emphasizing some important points. First, it should be clear that the design of systems which understand images well enough to enable effective search of large databases remains a challenging problem, and current retrieval systems are not useful for

all applications. The trend is positive, however, as shown by the improvements of retrieval performance from QBVE to QBSE. The retrieval community has also only just begun to explore avenues of tremendous potential, such as the use of semantic taxonomies [115, 132]. Second, the representations discussed here are far from exhausting what is possible in image analysis. In fact, recent work has already shown that (1) MPE image classifiers designed in the space of semantic features have improved performance over what is possible at visual level [93], (2) contextual relationships between visual concepts can be extracted by modeling probability distributions in semantic space [94], and (3) it is even possible to model the joint statistics of image and text, so as to enable seamless retrieval *across* the two modalities [92].

Finally, it should also be stressed that an image retrieval system is much more than an image similarity engine. In addition to image matching, it should address the problems of indexing to enable fast searches [122]; accounting for prior information, which can be used to weigh some images more strongly than others; and exploring the users presence in the retrieval loop. Information about the users preferences is usually collected by relevance feedback algorithms [127], operating at both short and long time scales [128]. Within a single session, the retrieval system can exploit user feedback to refine particular searches. As the user provides more information, the system becomes more confident about the users needs, and retrieval accuracy increases. Across sessions, the system can use relevance feedback to build user profiles or improve semantic labeling of the database images. All of these operations can be formulated under the MPE retrieval framework, and optimal solutions are available for a number of them [121, 122, 127, 128].

A

Proofs

In this appendix, we include the proofs of all mathematical results discussed in this monograph.

A.1 Proof of Theorem 2.1

The theorem follows from the application of two bounds. The first is that, for a problem with class conditional densities $P_{\mathbf{X}|Y}(\mathbf{x}|i)$, equiprobable classes $P_Y(i) = 1/M, \forall i$, class-conditional density estimates $\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)$, and a feature space $\mathcal{X}$,

$$\text{Prob}[g(\mathbf{X}) \neq Y] - L^*_{\mathcal{X}} \leq \frac{1}{M} \sum_i \int |P_{\mathbf{X}|Y}(\mathbf{x}|i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)| d\mathbf{x}. \quad \text{(A.1)}$$

The second is a well known bound in information theory, usually referred to as Pinsker's inequality[1] [88], see e.g., Lemma 12.6.1 of [22] or Theorem 7.11.1 of [12],

$$\int |P_{\mathbf{X}}(\mathbf{x}) - Q_{\mathbf{X}}(\mathbf{x})| d\mathbf{x} \leq \sqrt{2 \ln 2 \, KL[P_{\mathbf{X}}(\mathbf{x}) || Q_{\mathbf{X}}(\mathbf{x})]}.$$

[1] The general inequality relates relative entropy to variational distance, only the version for continuous distributions is considered here.

The proof of (A.1) is an extension of that given in [28] for $M = 2$. To extend this proof to multiple classes we note that

$$\begin{aligned}\text{Prob}[g(\mathbf{X}) \neq Y] - L^*_{\mathcal{X}} &= E_{\mathbf{X}}[\sum_i (\delta_{g^*(\mathbf{X}),i} - \delta_{g(\mathbf{X}),i}) P_{Y|\mathbf{X}}(i|\mathbf{X})] \\ &= \int_E \Delta(\mathbf{x}) P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x},\end{aligned}$$

where

$$E = \{\mathbf{x} | \mathbf{x} \in \mathcal{X}, P_{\mathbf{X}}(\mathbf{x}) > 0, \quad g(\mathbf{x}) \neq g^*(\mathbf{x})\}$$

and

$$\Delta(\mathbf{x}) = \sum_i (\delta_{g^*(\mathbf{x}),i} - \delta_{g(\mathbf{x}),i}) P_{Y|\mathbf{X}}(i|\mathbf{x}).$$

Defining the sets

$$\begin{aligned}E^*_i &= \{\mathbf{x} | \mathbf{x} \in E, g^*(\mathbf{x}) = i\} \\ E_i &= \{\mathbf{x} | \mathbf{x} \in E, g(\mathbf{x}) = i\},\end{aligned}$$

it follows that, $\forall \mathbf{x} \in E^*_i \cap E_j$,

$$\Delta(\mathbf{x}) = P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}).$$

We next note that, from (2.1),

$$P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) \geq 0, \quad \forall \mathbf{x} \in E^*_i, \ \forall j \neq i$$

and, from (2.5) and the fact that $P_{\mathbf{X}}(\mathbf{x}) > 0, \quad \forall \mathbf{x} \in E$,

$$\frac{\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|j)\hat{p}_Y(j)}{P_{\mathbf{X}}(\mathbf{x})} - \frac{\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)\hat{p}_Y(i)}{P_{\mathbf{X}}(\mathbf{x})} \geq 0, \quad \forall \mathbf{x} \in E_j, \ \forall i \neq j.$$

Defining

$$\hat{p}_{Y|\mathbf{X}}(i|\mathbf{x}) = \frac{\hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i)\hat{p}_Y(i)}{P_{\mathbf{X}}(\mathbf{x})},$$

it follows that, $\forall \mathbf{x} \in E^*_i \cap E_j$,

$$\begin{aligned}\Delta(\mathbf{x}) &= P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) \\ &\leq P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) + \hat{p}_{Y|\mathbf{X}}(j|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(i|\mathbf{x}) \\ &= |P_{Y|\mathbf{X}}(i|\mathbf{x}) - P_{Y|\mathbf{X}}(j|\mathbf{x}) + \hat{p}_{Y|\mathbf{X}}(j|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(i|\mathbf{x})| \\ &\leq |P_{Y|\mathbf{X}}(i|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(i|\mathbf{x})| + |P_{Y|\mathbf{X}}(j|\mathbf{x}) - \hat{p}_{Y|\mathbf{X}}(j|\mathbf{x})|\end{aligned}$$

and

$$\int_{E_i^* \cap E_j} \Delta(\mathbf{x}) P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \leq \int_{E_i^* \cap E_j} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} + \int_{E_i^* \cap E_j} |P_{\mathbf{X}|Y}(\mathbf{x}|j) P_Y(j) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|j) \hat{p}_Y(j)| d\mathbf{x}.$$

Using the fact that both collections of sets E_i^* and E_j partition E, we obtain

$$\begin{aligned}
\int_E \Delta(\mathbf{x}) P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} &= \sum_{i,j} \int_{E_i^* \cap E_j} \Delta(\mathbf{x}) P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \\
&\leq \sum_i \int_{E_i^*} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} \\
&\quad + \sum_j \int_{E_j} |P_{\mathbf{X}|Y}(\mathbf{x}|j) P_Y(j) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|j) \hat{p}_Y(j)| d\mathbf{x} \\
&= \sum_i \left[\int_{E_i^*} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} \right. \\
&\quad \left. + \int_{E_i} |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x} \right] \\
&\leq \sum_i \int |P_{\mathbf{X}|Y}(\mathbf{x}|i) P_Y(i) - \hat{p}_{\mathbf{X}|Y}(\mathbf{x}|i) \hat{p}_Y(i)| d\mathbf{x},
\end{aligned}$$

where we have also used the fact that $E_i^* \cap E_i = \emptyset$.

A.2 Proof of Theorem 2.2

The fact that the sequence of vector spaces is embedded follows from (2.12) since, $\forall i \in \{1, \ldots, d-1\}$,

$$\mathcal{X}_i = \pi_i^{i+1}(\mathcal{X}_{i+1}) \tag{A.2}$$

and, consequently, there is a sequence of one-to-one mappings

$$\epsilon_i(\mathbf{x}) = (\mathbf{x}, 0) \tag{A.3}$$

for which

$$\epsilon_i(\mathcal{X}_i) \subset \mathcal{X}_{i+1}. \tag{A.4}$$

Inequality (2.13) then follows from (A.2), (2.9) and the fact that the mappings $\pi_i^{i+1}(\mathbf{x})$ are non-invertible. To prove (2.16), we start from Theorem 2.1, i.e.,

$$\Delta_{g_i,\mathcal{X}_i} = \frac{\sqrt{2\ln 2}}{M}\sum_k \sqrt{\mathrm{KL}[P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)||\hat{p}_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)]}, \tag{A.5}$$

where $P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)$ is the class-conditional likelihood function for $\mathbf{X}_i$ under class k. Since, from (A.2), $\mathbf{X}_{i+1} = (\mathbf{X}_i, X_{i+1})$, where X_{i+1} is the $i+1$th coordinate of $\mathbf{X}_{i+1}$, we have

$$\begin{aligned}
&\mathrm{KL}[P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)||\hat{p}_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)] \\
&= \int P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)\log\frac{P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)}{\hat{p}_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)}d\mathbf{x}_{i+1} \\
&= \int P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)\log\frac{P_{X_{i+1}|\mathbf{X}_i,Y}(x_{i+1}|\mathbf{x}_i,k)}{\hat{p}_{X_{i+1}|\mathbf{X}_i,Y}(x_{i+1}|\mathbf{x}_i,k)}dx_{i+1}d\mathbf{x}_i \\
&\quad + \int P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)\log\frac{P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)}{\hat{p}_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)}dx_{i+1}d\mathbf{x}_i \\
&= \int P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)\log\frac{P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)}{\hat{p}_{X_{i+1}|\mathbf{X}_i,Y}(x_{i+1}|\mathbf{x}_i,k)P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)}dx_{i+1}d\mathbf{x}_i \\
&\quad + \int P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)\log\frac{P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)}{\hat{p}_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)}d\mathbf{x}_i \\
&= \mathrm{KL}[P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)||\hat{p}_{X_{i+1}|\mathbf{X}_i,Y}(x_{i+1}|\mathbf{x}_i,k)P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)] \\
&\quad + \mathrm{KL}[P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)||\hat{p}_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)] \\
&\geq \mathrm{KL}[P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)||\hat{p}_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)],
\end{aligned}$$

where we have used the non-negativity of the KL divergence [22]. It follows from the fact that the square root is a monotonically increasing function that

$$\begin{aligned}
&\sqrt{\mathrm{KL}[P_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)||\hat{p}_{\mathbf{X}_{i+1}|Y}(\mathbf{x}_{i+1}|k)]} \\
&\qquad \geq \sqrt{KL[P_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)||\hat{p}_{\mathbf{X}_i|Y}(\mathbf{x}_i|k)]}
\end{aligned}$$

which, combined with (A.5), leads to (2.16).

A.3 Proof of Lemma 4.1

From the properties of symmetric block matrices [2], it is known that if

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix}$$

where $\mathbf{A}$ and $\mathbf{D}$ are symmetric matrices, then

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A}^{-1}(\mathbf{I} + \mathbf{B}\mathbf{P}^{-1}\mathbf{B}^T\mathbf{A}^{-1}) & -\mathbf{A}^{-1}\mathbf{B}\mathbf{P}^{-1} \\ -\mathbf{P}^{-1}\mathbf{B}^T\mathbf{A}^{-1} & \mathbf{P}^{-1} \end{bmatrix} \tag{A.6}$$

$$= \Gamma(\mathbf{A}^{-1}) + \mathbf{E}\mathbf{P}^{-1}\mathbf{E}^T \tag{A.7}$$

and $|\mathbf{M}| = |\mathbf{A}||\mathbf{P}|$ with

$$\Gamma(\mathbf{A}^{-1}) = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \mathbf{A}^{-1}\mathbf{B} \\ -\mathbf{I} \end{bmatrix}$$

and $\mathbf{P} = \mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$. Hence, for any vector $\mathbf{z}^T = [\mathbf{x}^T\mathbf{y}^T]$, where $\mathbf{x}$ and $\mathbf{y}$ have the appropriate lengths for $||\mathbf{z}||_{\mathbf{M}}$ to make sense,

$$\begin{aligned} ||\mathbf{z}||_{\mathbf{M}} &= ||\mathbf{x}||_{\mathbf{A}} + (\mathbf{B}^T\mathbf{A}^{-1}\mathbf{x} - \mathbf{y})^T\mathbf{P}^{-1}(\mathbf{B}^T\mathbf{A}^{-1}\mathbf{x} - \mathbf{y}) \\ &= ||\mathbf{x}||_{\mathbf{A}} + ||\mathbf{B}^T\mathbf{A}^{-1}\mathbf{x} - \mathbf{y}||_{\mathbf{P}}. \end{aligned} \tag{A.8}$$

Using the decomposition

$$\mathbf{\Pi}_j = \begin{bmatrix} \mathbf{\Pi}_{j-1} \\ \mathbf{e}_j^T \end{bmatrix},$$

where $\mathbf{e}_j$ is the jth vector of the canonical basis of $\mathbb{R}^p$ (jth coordinate equal to 1, all others to 0), and defining $\mathbf{S}_j = \mathbf{\Pi}_j\mathbf{\Sigma}\mathbf{\Pi}_j^T$, it follows that

$$\mathbf{S}_j = \begin{bmatrix} \mathbf{S}_{j-1} & \mathbf{u}_{j-1} \\ \mathbf{u}_{j-1}^T & \sigma_{j,j} \end{bmatrix} \quad \text{and} \quad \mathbf{\Pi}_j\mathbf{d} = \begin{bmatrix} \mathbf{\Pi}_{j-1}\mathbf{d} \\ d_j \end{bmatrix},$$

where d_j is the jth element of $\mathbf{d}$. Making $\mathbf{M} = \mathbf{S}_j$, $\mathbf{A} = \mathbf{S}_{j-1}$, $\mathbf{B} = \mathbf{u}_{j-1}$, $\mathbf{D} = \sigma_{j,j}$, and defining $p_j = \mathbf{P}$, and $\psi_j = \mathbf{E}$, it follows that

$$\begin{aligned} \psi_j^T &= (\mathbf{u}_{j-1}^T\mathbf{S}_{j-1}^{-1}, -1) \\ p_j &= \sigma_{j,j} - ||\mathbf{u}_{j-1}||_{\mathbf{S}_{j-1}}, \\ &= -(\mathbf{u}_{j-1}^T, \sigma_{j,j})\,\psi_j \\ \mathbf{S}_j^{-1} &= \Gamma(\mathbf{S}_{j-1}^{-1}) + \frac{1}{p_j}\psi_j\psi_j^T. \end{aligned}$$

Letting $\mathbf{z} = \mathbf{\Pi}_j \mathbf{d}$, $\mathbf{x} = \mathbf{\Pi}_{j-1}\mathbf{d}$, $\mathbf{y} = d_j$, and applying (A.8),

$$||\mathbf{\Pi}_j \mathbf{d}||_{\mathbf{S}_j} = ||\mathbf{\Pi}_{j-1}\mathbf{d}||_{\mathbf{S}_{j-1}} + \frac{1}{p_j}(\psi_j^T \mathbf{\Pi}_j \mathbf{d})^2.$$

Since $\mathcal{M}_j = ||\mathbf{\Pi}_j \mathbf{d}||_{\mathbf{S}_j}$, this leads to (4.11)–(4.14). Furthermore, from $|\mathbf{M}| = |\mathbf{A}||\mathbf{P}|$, it follows that $|\mathbf{S}_j| = p_j|\mathbf{S}_{j-1}|$, which leads to (4.15). Finally, since the steps of (4.11)–(4.15) have complexity $O(j)$ or $O(j^2)$, the overall complexity is $O(\sum_{j=1}^{d} j^2) = O(d(d+1)(2d+1)/6 = O(d^3)$.

A.4 Proof of Theorem 6.1

We start by noting that

$$\begin{aligned} P_N(\mathbf{x}) &= \frac{1}{K} P_{\mathbf{C}}(\mathbf{x}) + \frac{1}{NK} \sum_{i=1}^{N} \sum_{j=1}^{K-1} \mathcal{G}(\mathbf{x}, \mu_j^i, \mathbf{\Sigma}_j^i) \\ &= \frac{1}{K} P_{\mathbf{C}}(\mathbf{x}) + \frac{1}{K} \sum_{j=1}^{K-1} \frac{1}{N} \sum_{i=1}^{N} \mathcal{G}(\mathbf{x}, \mu_j^i, \mathbf{\Sigma}_j^i) \end{aligned}$$

For any $\mathbf{x}$, it follows, from the strong the law of large numbers that, as $N \to \infty$, $P_N(\mathbf{x})$ converges almost surely to

$$\begin{aligned} P(\mathbf{x}) &= \frac{1}{K} P_{\mathbf{C}}(\mathbf{x}) + \frac{1}{K} \sum_{j=1}^{K-1} E_{\mu,\mathbf{\Sigma}}[\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})] \\ &= \frac{1}{K} P_{\mathbf{C}}(\mathbf{x}) + \left(1 - \frac{1}{K}\right) E_{\mu,\mathbf{\Sigma}}[\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})]. \end{aligned}$$

Using the independence of μ and $\mathbf{\Sigma}$,

$$\begin{aligned} &E_{\mu,\mathbf{\Sigma}}[\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma})] \\ &= \int P_{\mathbf{\Sigma}}(\mathbf{\Sigma}) \int \mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma}) \mathcal{G}(\mu, \mu_0, \mathbf{\Sigma}_0) d\mu d\Sigma \\ &= \int P_{\mathbf{\Sigma}}(\mathbf{\Sigma}) \mathcal{G}(\mathbf{x}, \mu_0, \mathbf{\Sigma} + \mathbf{\Sigma}_0) d\Sigma \\ &= E_{\mathbf{\Sigma}}[\mathcal{G}(\mathbf{x}, \mu_0, \mathbf{\Sigma} + \mathbf{\Sigma}_0)] \end{aligned}$$

and, with probability one, for all $\mathbf{x}$

$$\begin{aligned}
&\lim_{N\to\infty} P_N(\mathbf{x}) \\
&= \frac{1}{K} P_{\mathbf{C}}(\mathbf{x}) + \left(1 - \frac{1}{K}\right) E_{\boldsymbol{\Sigma}}[\mathcal{G}(\mathbf{x}, \mu_0, \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_0)] \\
&= f(\mathbf{x}) + \left(1 - \frac{1}{K}\right) [E_{\boldsymbol{\Sigma}}[\mathcal{G}(\mathbf{x}, \mu_0, \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_0)] - \mathcal{G}(\mathbf{x}, \mu_0, \mathbf{S} + \boldsymbol{\Sigma}_0)].
\end{aligned}$$

Using (6.6), leads to (6.9).

References

[1] A.Kalai and A.Blum., "A note on learning from multiple instance examples," *Artificial Intelligence*, vol. 30, pp. 23–30, 1998.

[2] M. Artin, *Algebra*. Prentice Hall, 1991.

[3] P. Auer, "On learning from multi-instance examples: Empirical evaluation of a theoretical approach," in *Proceedings of the International Conference on Machine Learning*, 1997.

[4] J. Bach, "The virage image search engine: An open framework for image management," in *SPIE Storage and Retrieval for Image and Video Databases*, San Jose, California, 1996.

[5] K. Barnard and D. Forsyth, "Learning the semantics of words and pictures," in *International Conference on Computer Vision,* vol. 2, pp. 408–415, Vancouver, 2001.

[6] A. Bell and T. Sejnowski, "The independent components of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3328, December 1997.

[7] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color-and texture-based image segmentation using EM and its application to content-based image retrieval," in *International Conference on Computer Vision*, pp. 675–682, Bombay, India, 1998.

[8] J. Bergen and E. Adelson, "Early vision and texture perception," *Nature*, vol. 333, no. 6171, pp. 363–364, 1988.

[9] J. Bergen and M. Landy, "Computational modeling of visual texture segregation," in *Computational Models of Visual Processing*, (M. Landy and J. Movshon, eds.), MIT Press, 1991.

[10] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning,* vol. 4. New York: Springer, 2006.
[12] R. Blahut, *Principles and Practice of Information Theory*. Addison Wesley, 1991.
[13] D. Blei and M. Jordan, "Modeling Annotated Data," in *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.
[14] J. Boreczky and L. Rowe, "Comparison of video shot boundary detection techniques," in *Proceedings of SPIE Conference on Visual Communication and Image Processing*, 1996.
[15] J. Cardoso, "Blind signal separation: Statistical principles," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 2009–2026, October 1998.
[16] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 394–410, March 2007.
[17] G. Carneiro and N. Vasconcelos, "A database centric view of semantic image annotation and retrieval," in *Proceedings of ACM SIGIR*, 2005.
[18] S. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 602–615, September 1998.
[19] R. Clarke, *Transform Coding of Images*. Academic Press, 1985.
[20] D. Comaniciu, P. Meer, K. Xu, and D. Tyler, "Retrieval performance improvement through low rank corrections," in *Workshop in Content-based Access to Image and Video Libraries*, pp. 50–54, Fort Collins, Colorado, 1999.
[21] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.
[22] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley, 1991.
[23] I. Csiszár, "I-divergence geometry of probability distributions and minimization problems," *Annals of Probability*, vol. 3, pp. 146–158, 1975.
[24] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, pp. 1–60, 2008.
[25] J. De Bonet and P. Viola, "Structure driven image database retrieval," in *Neural Information Processing Systems,* vol. 10, Denver, Colorado, 1997.
[26] J. De Bonet, P. Viola, and J. Fisher, "Flexible histograms: A multiresolution target discrimination model," in *Proceedings of SPIE,* vol. 3370-12, (E. G. Zelnio, ed.), 1998.
[27] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM Algorithm," *Journals of the Royal Statistical Society*, vol. B-39, 1977.
[28] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.
[29] T. Dietterich, R. Lathrop, and T. Lozano-Pere, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1, pp. 31–71, 1997.

[30] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley and Sons, 2001.

[31] P. Duygulu, K. Barnard, D. Forsyth, and N. Freitas, "Object recognition as machine translation: Learning a Lexicon for a fixed image vocabulary," in *European Conference on Computer Vision*, Copenhagen, Denmark, 2002.

[32] Y. Ephraim, A. Denbo, and L. Rabiner, "A minimum discrimination information approach for hidden Markov modeling," *IEEE Transactions on Information Theory*, vol. 35, no. 5, pp. 1001–1013, September 1989.

[33] Y. Ephraim, H. Lev-Ari, and R. M. Gray, "Asymptotic minimum discrimination information measure for asymptotically weakly stationary processes," *IEEE Trans. on Information Theory*, vol. 34, no. 5, pp. 1033–1040, September 1988.

[34] C. Fellbaum, *Wordnet: An Electronic Lexical Database*. MIT Press, 1998.

[35] S. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli relevance models for image and video annotation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington DC, 2004.

[36] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," *IEEE Conference in Computer Vision and Pattern Recognition*, 2003.

[37] D. Field, "What is the goal of sensory coding?," *Neural Computation*, vol. 6, no. 4, pp. 559–601, January 1989.

[38] I. Fogel and D. Sagi, "Gabor filters as texture discriminators," *Biological Cybernitics*, vol. 61, pp. 103–113, 1989.

[39] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[40] W. Gardner and B. Rao, "Theoretical analysis of the high-rate vector quantization of LPC parameters," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 367–376, September 1995.

[41] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*. Chapman Hall, 1995.

[42] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Press, 1992.

[43] R. M. Gray, "Vector quantization," *Signal Processing Magazine*, vol. 1, April 1984.

[44] R. M. Gray, A. Gray, G. Rebolledo, and J. Shore, "Rate-distortion speech coding with a minimum discrimination information distortion measure," *IEEE Transactions on Information Theory*, vol. IT-27, pp. 708–721, November 1981.

[45] V. Guillemin, *Differential Topology*. Pearson Education, 1974.

[46] M. Gupta and Y. Chen, "Theory and use of the EM method," *Foundations and Trends in Signal Processing, NOW Publishers*, vol. 4, pp. 223–296, 2010.

[47] N. Howe, "Percentile blobs for image similarity," in *Workshop in Content-based Access to Image and Video Libraries*, pp. 78–83, Santa Barbara, California, 1998.

[48] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Spatial color indexing and applications," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 245–268, December 1999.

[49] D. Hubel and T. Wiesel, "Brain mechanisms of vision," *Scientific American*, September 1979.
[50] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.
[51] A. Hyvarinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, 2000.
[52] F. Idris and S. Panchanathan, "Storage and retrieval of compressed sequences," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 937–941, August 1995.
[53] G. Iyengar and A. Lippman, "Clustering images using relative entropy for efficient retrieval," in *International workshop on Very Low Bitrate Video Coding*, Urbana, Illinois, 1998.
[54] A. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognition Journal*, vol. 29, pp. 1233–1244, August 1996.
[55] N. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice Hall, 1984.
[56] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, pp. 52–60, February 1967.
[57] V. Kotel'nikov, *The Theory of Optimum Noise Immunity*. New York: McGraw-Hill, 1959.
[58] S. Kullback, *Information Theory and Statistics*. New York: Dover Publications, 1968.
[59] M. Kupperman, "Probabilities of hypothesis and information-statistics in sampling from exponential-class populations," *Annals of Mathematical Statistics*, vol. 29, pp. 571–574, 1958.
[60] V. Lavrenko, R. Manmatha, and J. Jeon, "A model for learning the semantics of pictures," in *Neural Information Processing Systems*, Denver, Colorado, 2003.
[61] V. Lavrenko, R. Manmatha, and J. Jeon, "A model for learning the semantics of pictures," in *Neural Information Processing Systems*, 2003.
[62] H. Lev-Ari, S. Parker, and T. Kailath, "Multidimensional maximum-entropy covariance extension," *IEEE Transactions on Information Theory*, vol. 35, no. 3, pp. 497–508, May 1988.
[63] J. Li, N. Chadda, and R. M. Gray, "Asymptotic performance of vector quantizers with a perceptual distortion measure," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1082–1091, May 1999.
[64] J. Li and J. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on Pattern. Analysis and Machine Intelligence*, vol. 25, no. 10, 2003.
[65] Q. Li, "Estimation of mixture models," PhD thesis, Yale University, 1999.
[66] T. Linder and R. Zamir, "High-resolution source coding for non-difference distortion measures: The rate-distortion function," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 533–547, March 1999.
[67] F. Liu and R. Picard, "Periodicity, directionality, and randomness: Wold features for image modeling and retrieval," *IEEE Transactions on Pattern. Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 722–733, July 1996.

[68] Y. Liu, D. Zhang, G. Lu, and W. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[69] W. Ma and H. Zhang, "Benchmarking of image features for content-based retrieval," in *Asilomar Conference on Signals, Systems, and Computers*, Asilomar, California, 1998.

[70] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America*, vol. 7, no. 5, pp. 923–932, May 1990.

[71] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern. Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[72] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[73] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern. Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, August 1996.

[74] J. Mao and A. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, 1992.

[75] O. Maron and T. Lozano-Perez, "A framework for multiple-instance learning," in *Neural Information Processing Systems 10*, Denver, Colorado, 1998.

[76] O. Maron and A. Ratan, "Multiple-instance learning for natural scene classification," in *Proceedings of 15th International Conference on Machine Learning*, 1998.

[77] B. Moghaddam, H. Bierman, and D. Margaritis, "Defining image content with multiple regions-of-interest," in *Workshop in Content-based Access to Image and Video Libraries*, pp. 89–93, Fort Collins, Colorado, 1999.

[78] Y. Mori, H. Takahashi, and R. Oka, "Image-to-word transformation based on dividing and vector quantizing images with words," in *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[79] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearence," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.

[80] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Pektovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC project: Querying images by content using color, texture, and shape," in *SPIE Storage and Retrieval for Image and Video Databases*, pp. 173–181, San Jose, California, 1993.

[81] N. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Transactions on Communications*, vol. 33, pp. 551–557, June 1985.

[82] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[83] M. Ortega, Y. Rui, K. Chakrabarti, K. Porkaew, S. Mehrotra, and T. Huang, "Supporting ranked Boolean similarity queries in MARS," *IEEE Transactions*

on Knowledge and Data Engineering, vol. 10, no. 6, pp. 905–925, December 1998.

[84] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[85] G. Pass and R. Zabih, "Comparing images using joint histograms," *ACM Journal of Multimedia Systems*, vol. 7, no. 3, pp. 234–240, May 1999.

[86] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233–254, June 1996.

[87] R. Picard, T. Kabir, and F. Liu, "Real-time recognition with the entire brodatz texture database," in *Proceedings of IEEE Conference on Computer Vision*, New York, 1993.

[88] M. Pinsker, *Information and Information Stability of Random Variables and Processes*. San Francisco: Holden-Day, 1964.

[89] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," in *International Conference on Computer Vision*, pp. 1165–1173, Korfu, Greece, 1999.

[90] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[91] N. Rasiwasia, P. Moreno, and N. Vasconcelos, "Bridging the gap: Query by semantic example," *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 923–938, August 2007.

[92] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *ACM Proceedings of the International Conference on Multimedia*, 2010.

[93] N. Rasiwasia and N. Vasconcelos, "Scene classification with low-dimensional semantic spaces and weak supervision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[94] N. Rasiwasia and N. Vasconcelos, "Holistic context modeling using semantic co-occurrences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[95] R. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, April 1984.

[96] Y. Rui, T. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, pp. 39–62, March 1999.

[97] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 644–655, September 1998.

[98] D. Sagi, "The Psychophysics of Texture Segmentation," in *Early Vision and Beyond* , chapter 7, (T. Papathomas, ed.), MIT Press, 1996.

[99] H. Sakamoto, H. Suzuki, and A. Uemori, "Flexible montage retrieval for image data," in *SPIE Storage and Retrieval for Image and Video Databases*, San Jose, California, 1994.

[100] B. Schiele and J. Crowley, "Recognition without correspondence using multidimensional receptive field histograms," *International Journal of Computer Vision*, vol. 36, no. 1, pp. 31–50, January 2000.
[101] C. Schmid and R. Mohr, "Local greyvalue invariants for image retrieval," *IEEE Transactions on Pattern. Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–535, May 1997.
[102] D. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley, 1992.
[103] C. Shyu, C. Brodley, A. Kak, A. Kosaka, A. Aisen, and L. Broderick, "Local versus global features for content-based image retrieval," in *Workshop in Content-based Access to Image and Video Libraries*, pp. 30–34, Santa Barbara, California, 1998.
[104] J. Simonoff, *Smoothing Methods in Statistics*. Springer-Verlag, 1996.
[105] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval: The end of the early years," *IEEE Transactions on Pattern. Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.
[106] J. Smith, "Integrated spatial and feature image systems: Retrieval, compression and analysis," PhD thesis, Columbia University, 1997.
[107] J. Smith and S. Chang, "VisualSEEk: A fully automated content-based image query system," in *ACM Multimedia*, pp. 87–98, Boston, Massachussetts, 1996.
[108] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," in *SPIE Storage and Retrieval for Image and Video Databases*, pp. 29–40, San Jose, California, 1996.
[109] M. Stricker and M. Orengo, "Similarity of color images," in *SPIE Storage and Retrieval for Image and Video Databases*, San Jose, California, 1995.
[110] M. Stricker and M. Swain, "The capacity of color histogram indexing," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 704–708, 1994.
[111] A. Sutter, J. Beck, and N. Graham, "Contrast and spatial variables in texture segregation: Testing a simple spatial-frequency channels model," *Perceptual Psychophysics*, vol. 46, pp. 312–332, 1989.
[112] M. Swain and D. Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
[113] L. Taycher, M. Cascia, and S. Sclaroff, "Image digestion and relevance feedback in the image rover WWW search engine," in *Visual*, San Diego, California, 1997.
[114] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. John Wiley, 1985.
[115] A. Tousch, S. Herbin, and J. Audibert, "Semantic hierarchies for image annotation: A survey," *Pattern Recognition*, vol. 45, pp. 333–345, 2012.
[116] H. V. Trees, *Detection, Estimation, and Modulation Theory*. Wiley, 1968.
[117] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, 1991.
[118] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
[119] M. Vasconcelos, G. Carneiro, and N. Vasconcelos, "Weakly supervised top-down image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[120] M. Vasconcelos and N. Vasconcelos, "Natural image statistics and low-complexity feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 228–244, 2009.

[121] N. Vasconcelos, "Bayesian models for visual information retrieval," PhD thesis, Massachusetts Institute of Technology, 2000.

[122] N. Vasconcelos, "Image indexing with mixture hierarchies," in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, Kawai, Hawaii, 2001.

[123] N. Vasconcelos, "Exploiting group structure to improve retrieval accuracy and speed in image databases," in *Proceedings of International Conference Image Processing*, Rochester, NY, 2002.

[124] N. Vasconcelos, "Minimum probability of error image retrieval," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2322–2336, 2004.

[125] N. Vasconcelos, "On the efficient evaluation of probabilistic similarity functions for image retrieval," *IEEE Transactions on Information Theory*, vol. 50, pp. 1482–1496, 2004.

[126] N. Vasconcelos and A. Lippman, "Library-based coding: A representation for efficient video compression and retrieval," in *Proceedings of Data Compression Conference*, Snowbird, Utah, 1997.

[127] N. Vasconcelos and A. Lippman, "Learning from user feedback in image retrieval systems," in *Neural Information Processing Systems*, Denver, Colorado, 1999.

[128] N. Vasconcelos and A. Lippman, "Learning over multiple temporal scales in image databases," in *Proceedings of European Conference on Computer Vision*, Dublin, Ireland, 2000.

[129] N. Vasconcelos and A. Lippman, "A probabilistic architecture for content-based image retrieval," in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, Hilton Head, North Carolina, 2000.

[130] X. Wan and C. Kuo, "A new approach to image retrieval with hierarchical color clustering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 628–643, September 1998.

[131] J. Wang, G. Wiederhold, O. Firschein, and A. Wei, "Content-based image indexing and searching using daubechies' wavelets," *International Journal of Digital Libraries*, vol. 1, pp. 311–328, 1997.

[132] L. Xie, R. Yan, J. Tesic, A. Natsev, and J. R. Smith, "Probabilistic visual concept trees," in *ACM Multimedia*, 2010.