Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 25 (2007) 1748–1758
www.elsevier.com/locate/imavis

On the Bayes fusion of visual features

Xiaojin Shi, Roberto Manduchi *

Department of Computer Engineering, University of California, Santa Cruz, Santa Cruz, CA 95064, USA

Received 17 January 2006; received in revised form 9 January 2007; accepted 9 January 2007

Abstract

We consider the problem of image classification when more than one visual feature is available. In such cases, Bayes fusion offers an attractive solution by combining the results of different classifiers (one classifier per feature). This is the general form of the so-called "naive Bayes" approach. This paper compares the performance of Bayes fusion with respect to Bayesian classification, which is based the joint feature distribution. It is well-known that the latter has lower bias than the former, unless the features are conditionally independent, in which case the two coincide. However, as originally noted by Friedman, the low variance associated with naive Bayes estimation may mitigate the effect of its inherent bias. Indeed, in the case of small training samples, naive Bayes may outperform Bayes classification in terms of error rate.

The contribution of this paper is threefold. First, we present a detailed analysis of the error rate of Bayes fusion assuming that the statistical description of the data is known. Second, we provide a qualitative justification of the small sample effect on the classifier's performance based on the bias/variance theory. Third, we present experimental results on three image data sets using color and texture features. Our experiments highlight the relationship between the error rate of the Bayes and the Bayes fusion classifiers as a function of the training sample size.

Keywords: Image classification; Bayes fusion; Color; Texture

1. Introduction

Classifying a scene into a number of known objects or surface materials is one of the most important tasks in computer vision. We consider here pixel-level classification, with a set of local features (such as color, texture, and optical flow) available at each pixel. Our work deals with the *statistical fusion* of such features, to obtain a classification that is hopefully more reliable than by using just one feature.

In principle, one may simply create a super-feature as the juxtaposition of all available feature vectors. Although, this is a straightforward procedure from a mathematical viewpoint, simply attaching feature vectors together may give the impression of "mixing apples with oranges". Indeed, in many practical cases one may prefer to analyze the different features separately and fuse together the classification results in a second stage. This strategy would allow one to use different classification methods for different features. Each classifier may be optimized for the characteristics of its input features, for example using available physical models.

Combining classifiers is the object of intense research in the machine learning and in the sensor fusion communities. In this article we consider a very simple combination method: the multiplicative rule on the posterior probabilities (or *Bayes fusion*) [17]. When the different features are single-dimensional, Bayes fusion of classifiers is usually called *naive Bayes* classification [21]. These two terms will be interchanged liberally in this paper.

It is well-known that naive Bayes classification coincides with Bayesian classification over the "composite" feature only if the individual features are conditionally independent for any given class. This is a strong assumption, which

* Corresponding author. Tel.: +1 831 459 1479.
E-mail address: manduchi@cse.ucsc.edu (R. Manduchi).

0262-8856/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2007.01.003

does not hold true in general. Yet, Bayes fusion is known to give good results even when features are correlated. This phenomenon was justified by Friedman [10] in light of the bias/variance theory, which applies when only small training samples are available. Since naive Bayes classifiers are "less complex" than Bayes classifiers, their variance is less affected by the training sample size. When the sample size is small, the contribution of the variance term dominates in the expression of the classification error. In such cases, a naive Bayes classifier can outperform a Bayes classifier (more precisely, a classifier designed with an algorithm that is asymptotically unbiased). Note that, whereas the bias/variance phenomenon for regression problems is well-known (in which case bias, variance and noise contribute as additive terms to the error variance [12]), for classification error under 0/1 loss no such simple relationship exists [3,8,10,19,33,35].

The contribution of this paper is threefold:

(1) An analytical expression is provided for the misclassification rate of Bayes fusion as a function of feature correlation in a simple case (equi-covariant 2-D Gaussians).
(2) The relationship between the Bayes and the Bayes fusion classifiers in the small sample case is described on the grounds of the bias/variance theory for classification. Based on Domingos' approach [8], we derive an expression that offers a simple intuition of the bias/variance trade-offs in the two cases.
(3) Quantitative empirical results on real image data sets using color and texture features are presented, showing the dependence of the error rate on the training sample size for both the Bayes and the Bayes fusion cases. Our experiments confirm that the difference in error rate in the two cases decreases as the sample size is reduced, and that in some cases a cross-over point is identified where the former outperforms the latter.

The naive Bayes classifier has been studied in pattern recognition, machine learning, sensor fusion and information retrieval for several decades. As noted by Lewis [21], there have been mostly two research directions in the field of naive Bayes. The first direction attempts to modify the feature set so that the resulting features are as independent as possible. Typical work includes [20], which uses a greedy algorithm to select a subset of features that are only weakly correlated. Friedman et al. [11] proposed an algorithm that allows adding edges to a Bayes network to approximate the correlation between features. The second research direction focuses on explaining why naive Bayes classifiers work well even when the conditional independent assumption does not hold [5,7,10]. In particular, J. Friedman's study of naive Bayes classifiers in the context of the bias/variance dilemma [10] inspired the work presented in this paper. The work of Raudys and colleagues on small sample effect [26,27] is also very relevant in this context. Raudys and Jain [26] used a simple isotropic Gaussian model to derive the minimum number of samples needed to achieve a given average misclassification error for a few different classifiers. Gaussian modeling makes the problem approachable, and we use a Gaussian model as well in this paper as a toy example. However, for more general cases, analytical results are very difficult to obtain. Our approach in this work is to link the small sample phenomena to the general bias/variance theory and to use cross-validation as a tool for quantitatively assessing the system's performances with real data sets.

Bayes fusion of classifiers in the context of computer vision has been studied by a number of authors. In [17,18], Kittler et al. consider different fusion strategies under a Bayesian theoretical framework, including sum, product, minimum, maximum, median, and majority voting. This work maintained that the sum rule may outperform the product rule due to its robustness to the noise of each single expert. In [32], Tax et al. demonstrated that the theoretically predicted relationship between the sum and product rule in [17,18] holds in the case of relatively high estimation errors. However, for small errors, the product rule outperforms the sum rule. In [1], Alkoot and Kittler further confirmed Tax's conclusions. Note also that Bayes fusion is an instance of Multiple Classifier Systems, which have received a good deal of attention recently (see for example [15]).

This paper is organized as follows. Section 2 covers the theory of Bayes fusion, highlighting the relationship between correlation and error rate in a simple Gaussian case (the bulk of the related computation is deferred to the Appendix). Section 3 introduces our formulation of the bias/variance theory for classification, and shows its application to a simple Gaussian case using Bayes and naive Bayes classifiers from finite size training samples. Section 4 describes our experimental set-ups and results. Section 5 has the conclusions.

2. Conditional independence and error rate

This section covers some important theoretical aspects of Bayes fusion. We assume here, that the statistical quantities of interest (probability density functions and posterior distributions) are known and study the performance of the Bayes fusion algorithm in such an ideal case. In Section 3 we will consider the more realistic case of estimation from finite-size samples.

Throughout this work, x indicates a generic feature to be classified. We will assume that a feature is generated by one class, y, in a pre-determined set. The (total) probability density function (pdf) of x is indicated by $p_x(x)$, while $p_{x|y}(x|y) = p_{x,y}(x,y)/P_y(y)$ is the conditional likelihood of x given the class y ($p_{x,y}(x,y)$ being the joint likelihood and $P_y(y)$ being the prior probability of class y). $P_{y|x}(y|x) = p_{x,y}(x,y)/p(x)$ is the posterior probability of class y given that the feature x was observed.

A classifier is a deterministic function that maps an input feature x onto one class index $\bar{y}$. The general form of the total expected loss (or total risk) R of a classifier is the following:

$$R = E_{x,y}[L(\bar{y}, y)] \tag{1}$$

where $E_{x,y}[\cdot]$ is the expectation with respect to the density $p_{x,y}(x,y)$. We will only consider the 0–1 loss function in this paper:

$$L(\cdot,\cdot) = 1 - \delta(\cdot,\cdot) \tag{2}$$

where $\delta(\cdot,\cdot)$ is 1 when its arguments coincide, 0 otherwise. In this case, the total risk corresponds to the probability of misclassification. A well-known result (immediately derived from (1)) states that the total risk is minimized by the maximizer $y^*(x)$ of the joint distribution $p_{x,y}(x,y)$. Thus, the minimum risk under 0–1 loss (called *Bayes rate*) is:

$$\begin{aligned} R_{\min} &= E_{x,y}[(1 - \delta(y^*, y))] \\ &= 1 - \sum_y \int p_{x,y}(x,y)\ \delta(y^*, y)\ \mathrm{d}x \\ &= 1 - \int \max_y\ p_{x,y}(x,y)\ \mathrm{d}x \end{aligned} \tag{3}$$

It is useful to represent classifiers in terms of "assignment regions". For a 2-class classifier, I_0 and I_1 represent the regions of the feature space that are assigned to class 0 and 1, respectively (I_1 being the complement of I_0.) Thus, the Bayes classifier defines:

$$I_0 = \left\{ x : \frac{p_{x|y}(x|0)P(0)}{p_{x|y}(x|1)P(1)} > 1 \right\} = \left\{ x : \frac{P_{y|x}(0|x)}{P_{y|x}(1|x)} > 1 \right\} \tag{4}$$

The naive Bayes classifier does not have access to the joint feature statistics, and can only use marginal information. Let us consider the simple 2-D case with $x = (x_1, x_2)$; our treatment extends easily to the N-D case. The naive Bayes classifier defines the following assignment regions [17]:

$$\begin{aligned} \hat{I}_0 &= \left\{ x : \frac{p_{x_1|y}(x_1|0)p_{x_2|y}(x_2|0)P_y(0)}{p_{x_1|y}(x_1|1)p_{x_2|y}(x_2|1)P_y(1)} > 1 \right\} \\ &= \left\{ x : \frac{P_{y|x_1}(0|x_1)P_{y|x_2}(0|x_2)P_y(1)}{P_{y|x_1}(1|x_1)P_{y|x_2}(1|x_2)P_y(0)} > 1 \right\} \end{aligned} \tag{5}$$

where $p_{x_1|y}(x_1|y)$ is the marginal conditional likelihood $\int_{-\infty}^{\infty} p_{x|y}(x|y)\ \mathrm{d}x_2$.

If the conditional densities $p(x|0)$ and $p(x|1)$ are separable, i.e. if $p_{x|y}(x|y) = p_{x_1|y}(x_1|y)p_{x_2|y}(x_2|y)$, then the naive Bayes classifier coincides with the regular Bayes classifier. This situation is called of *conditional independence*, because it means that the 1-D features x_1 and x_2 are statistically independent given that the class that generated them is known.

The conditional independence assumption, while not as strong as total independence, is unrealistic in most cases of interest. However, even when features are not conditionally independent, naive Bayes classification does not necessarily yield catastrophic results. This is seen by examining the form of the theoretical error rates $P_{\mathrm{B}}(E)$ and $P_{\mathrm{NB}}(E)$ in the Bayes and naive Bayes case, respectively. Their (non-negative) difference is equal to

$$\begin{aligned} \Delta P(E) &= P_{\mathrm{NB}}(E) - P_{\mathrm{B}}(E) \\ &= \int_{\hat{I}_0 \cap I_1} \left(p_{x,y}(x,1) - p_{x,y}(x,0) \right)\ \mathrm{d}x \\ &\quad + \int_{\hat{I}_1 \cap I_0} \left(p_{x,y}(x,0) - p_{x,y}(x,1) \right)\ \mathrm{d}x \end{aligned} \tag{6}$$

If the class priors are identical, then one can simplify the formula above as:

$$\Delta P(E) = \int_{(\hat{I}_0 \cap I_1) \cup (\hat{I}_1 \cap I_0)} |2P_{y|x}(1|x) - 1|\ p_x(x)\ \mathrm{d}x \tag{7}$$

Thus, the conditional independence assumption leads to noticeably higher error rate only if the density $p(x)$ places substantial mass in $(\hat{I}_0 \cap I_1)$ or in $(\hat{I}_1 \cap I_0)$, and when the posterior probabilities are far from 0.5 in those regions.

Intuitively, one may expect that $\Delta P(E)$ should be large when x_1 and x_2 are tightly correlated. Finding a general formula relating $\Delta P(E)$ to any particular measure of correlation is probably impossible. However, in some simple cases one can actually derive analytical solutions. For example, in the Appendix we consider the case in which $p_{x|y}(x|0)$ and $p_{x|y}(x|1)$ are both Gaussian distributions with the same covariance but different means, and with equal class priors. One can see that, all other conditions being the same, $\Delta P(E)$ is bounded from above by an increasing function of the correlation ρ between x_1 and x_2. This is shown in Fig. 1, which illustrates the error rate for the Bayes classifier and the upper bound for the error rate of the naive Bayes classifier as a function of the Mahalanobis distance between the two Gaussians and of each Gaussian's correlation. This result agrees nicely

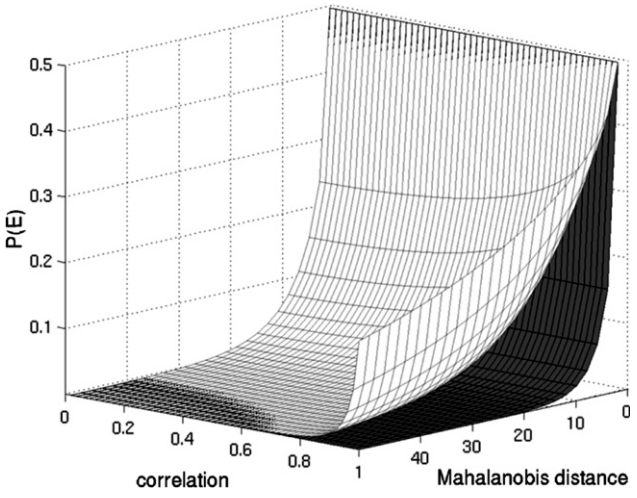


Fig. 1. The misclassification rate of the Bayes classifier (darker surface) and its upper bound for the naive Bayes classifier (lighter surface) for two equi-covariant 2-D Gaussians as a function of the Mahalanobis distance of their means and of the correlation ρ of each distribution.

with intuition; unfortunately, it cannot be generalized in a simple way to more complex cases.

Second order statistics are often insufficient for describing the statistical dependence of multiple features in the non-Gaussian case. In [29] the authors proposed to relate the misclassification rate with the mutual information of the feature vector which, in the 2-D Gaussian case, turns out to be a function of the correlation ρ. However, such a relationship can only be estimated empirically, and the computation of the mutual information is a difficult problem per se.

Indeed, as it turns out, these theoretical approaches are only meaningful when the distribution of the data can be estimated reliably, which implies that large training data sets are available. In practice, small sample effects may have a dominant role in the system's performance, as discussed in the next section.

3. Small sample analysis

A classifier is usually designed starting from a pool of training data. The size of the training sample may vary widely depending on the application. In some cases, there is plenty of data available, and the only constraint is the time required for training. In other cases, data may be scarce, perhaps due to the cost (including time) of acquiring it. Another important factor is the cost of manually labeling the data.

The finite sample size adds a new level of randomness to the analysis. Different training samples will determine different classification boundaries; the expected risk should therefore be averaged also over the distribution of training samples. A formal tool that enables to draw a qualitative assessment of the expected system's performance in these cases is the bias/variance theory, which is presented next.

3.1. The bias/variance theory for classification

The bias/variance theory was originally developed by Geman [12] for regression under quadratic loss. Different extensions of Geman's results to the classification case were proposed by several authors [3,8,10,19,33,35]. A main concern of this body of work was the derivation of a unified theory that would apply to both quadratic and 0–1 loss. In this paper, we make no attempt to theory unification; rather, we manipulate the formal identities proposed by Domingos [8] to obtain some simple and useful relationships. The bias/variance theory is very general, and can be used to explain the "peaking" phenomenon observed by Raudys and Jain [26] when an increasing number of features are added to a classifier.

All of the analysis in this section will concentrate on a single value x of the observed feature.[1] In order to keep the notation simple, we neglect to specify the dependence on x in all expressions within this section. We will assume that a classification algorithm is trained over a sample D. When presented with the feature x, this classifier will try to guess the "true" class that generated x; its output will be called $\bar{y}_D$. It is important to realize that $\bar{y}_D$ is a deterministic function of x; its randomness comes into play when considering different training samples D, and is induced by the distribution $P_D(D)$. When we need not emphasize its dependence on D, we will indicate the outcome of the classifier simply by $\bar{y}$. It should be clear that

$$P_{\bar{y}}(\bar{y}) = \sum_{D:\bar{y}_D=\bar{y}} P_D(D) \tag{8}$$

As in the previous section, we will denote by y^* the maximizer of the posterior probability distribution (or the "mode" of the posterior distribution). This is the Bayes classification of input x. Following Domingos [8], we will call *main prediction* (denoted by y_m) the mode of $P_{\bar{y}}(\bar{y})$. In other words, as different training samples are chosen, the feature x will be classified as y_m most of the times. Note that y_m depends only on the chosen algorithm and on the distribution of all possible training samples D. For a given feature x, the main prediction y_m may or may not coincide with the Bayes classification y^*. In the first case, the classifier is *unbiased* at x; otherwise, it is *biased*. Another quantity of interest is the *variance* (V), which describes how sensitive the classification is to the choice of the training data. More specifically, the variance is defined as

$$V = P_{\bar{y}}(\bar{y} \neq y_m) \tag{9}$$

As pointed out by Domingos, these quantities can be generalized to different choices of loss functions. For example, using a quadratic loss instead of the 0–1 loss gives the more familiar definition of variance as $E_{\bar{y}}[(\bar{y} - E(\bar{y}))^2]$. Finally, we will denote by *noise* (N) the misclassification rate of the Bayes classifier:

$$N = P(y \neq y^*) = 1 - \max_y P_y(y) \tag{10}$$

The reader is reminded that our notation neglects the dependence on x, and thus $P_y(y)$ is *not* the prior probability of class y, but rather the posterior probability $P_{y|x}(y|x)$.

As noted by several researchers, "complex" classifiers usually have less bias than "simpler" classifier, but their variance is higher. Thus, we may expect that the Bayes fusion classifier (which, in a sense, simplifies the statistical description of the features [10]) should have a lower variance than the Bayes classifier. Another factor influencing the variance is, of course, the size of the training sample: the variance decreases, in general, as the sample size increases [24,27].

The expected error rate, averaged over all classes *and* training samples, is

[1] Indeed, the difficulty of extending these analytical results to the average-case model [34] is a main shortcoming of the bias/variance theory.

$$P(E) = \sum_y \sum_{\bar{y}} P_{y,\bar{y}}(y,\bar{y})\ (1-\delta(y,\bar{y})) = \sum_y \sum_{D:\bar{y}_D \neq y} P_{D|y}(D|y) P_y(y) \quad (11)$$

Under the assumption that the training data are selected independent of the future data to be classified, we can assert that $P_{D|y}(D|y) = P_D(D)$ (see also [10]) and therefore

$$\sum_{D:\bar{y}_D \neq y} P_{D|y}(D|y) = P(\bar{y} \neq y) \quad (12)$$

and

$$P(E) = \sum_y P(\bar{y} \neq y) P_y(y) \quad (13)$$

Let us now assume that there are only two classes. Since the Bayes classification y^* does not change with D, we can write:

$$P(E) = P(y = y^*)\ P(\bar{y} \neq y^*) + P(y \neq y^*)\ P(\bar{y} = y^*) = \max_y P_y(y)\ P(\bar{y} \neq y^*) + \min_y P_y(y)\ P(\bar{y} = y^*) \quad (14)$$

Note that, in the case of multiple classes, we cannot obtain such a simple formula, unless

$$\sum_{y:y \neq y^*} P_y(y)\ P(\bar{y} \neq y) = P(y \neq y^*)\ P(\bar{y} = y^*) \quad (15)$$

which is not true in general.

We will now consider the biased and the unbiased cases separately. If the classifier is unbiased at x (that is, $y^* = y_m$), then, remembering our previous definitions and noting that $\min_y P_y(y) + \max_y P_y(y) = 1$, one easily sees that

$$P(E) = N + (1-2N)V \quad (16)$$

(note that the term $(1-2N)$ is always non–negative). In the biased case ($y* \neq y_m$), we have a different expression:

$$P(E) = N + (1-2N)(1-V) \quad (17)$$

In (16) and (17), the noise N is an unavoidable component of $P(E)$: the misclassification rate at x will never be smaller than the Bayes error. If the classifier is unbiased at x, then the error rate will increase with the variance. This was expected, and signifies that the error rate correlates with the size of the training sample (smaller samples means higher variance and therefore higher error rate). When the classifier is biased at x, though, the effect of variance is reversed. This phenomenon, first described by Friedman [10] using Gaussian approximations, can be explained by realizing that high variance means a higher opportunity for the classifier to generate classifications different from its (incorrect) mode.

It is instructive to compare Eqs. (16), (17) with the results obtained by Domingos [8]:

$$P(E) = \begin{cases} (2P(\bar{y} = y^*) - 1)N + V & \text{(when unbiased at x)} \\ (2P(\bar{y} = y^*) - 1)N - V & \text{(when biased at x)} \end{cases} \quad (18)$$

In order to reconcile the two expressions, we observe that if the classifier is unbiased at x, then $y^* = y_m$ and $P(\bar{y} = y^*) = P(\bar{y} = y_m) = 1 - V$. Otherwise, $P(\bar{y} = y^*) = P(\bar{y} \neq y_m) = V$. Substituting these identities for the corresponding terms in (18), one easily obtain (16) and (17). Although these two expressions for the misclassification rate at x are equivalent, one may argue that (16) and (17) are slightly more intuitive than (18), since it does not invoke the (only apparently) new quantity $P(\bar{y} = y^*)$. Indeed, analysis of Eqs. (16) and (17) allows us to draft a qualitative comparison between a complex classifier (C1) and a simpler classifier (C2). As mentioned earlier, the former is expected to be biased in a smaller region of the feature space than the latter, but its variance grows faster as the training sample size is reduced. For a given feature x, we can have four distinct cases:

(1) *Both C1 and C2 are unbiased*. In this case, C2 has a lower error rate for small samples, due to its smaller variance.
(2) *C1 is unbiased and C2 is biased*. For large samples, C1 has a lower error rate. As the sample size is reduced, the error rate of C1 increases while the error rate of C2 decreases, due to the increased variance.
(3) *C1 is biased and C2 is unbiased*. This is the reverse of the previous situation. Note, however, that this case is relatively unlikely, given our original assumption on the two classifiers.
(4) *Both C1 and C2 are biased*. C1 has lower error rate than C2 due to its higher variance.

It is thus conceivable that, if the set of feature points in which C1 is biased has low probability, the "simpler" classifier C2 may outperform the more complex classifier C1 for a small enough training sample. A simple yet enlightening example of this phenomenon is presented next.

3.2. A simple example

In order to highlight the effect of bias and variance on the error rate as a function of the training sample size, we present here a simple example involving Bayes fusion. Consider an experiment with two equi-probable classes and a 2-D feature $x = (x_1, x_2)$, where the conditional likelihoods are Gaussian with the same covariance matrix Σ but different means μ_0 and μ_1:

$$\mu_0 = (0,0)', \quad \mu_1 = (4,0)', \quad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \quad (19)$$

Equi-level curves of the two densities are shown in Fig. 2. According to (24), the error rate of the Bayes classifier in this case is $P_B(E) = 0.015$.

Our experiment proceeds as follows. A number, n, of data points are sampled from the first distribution, and the mean and covariance of the distribution are estimated from them. This operation is repeated for the second

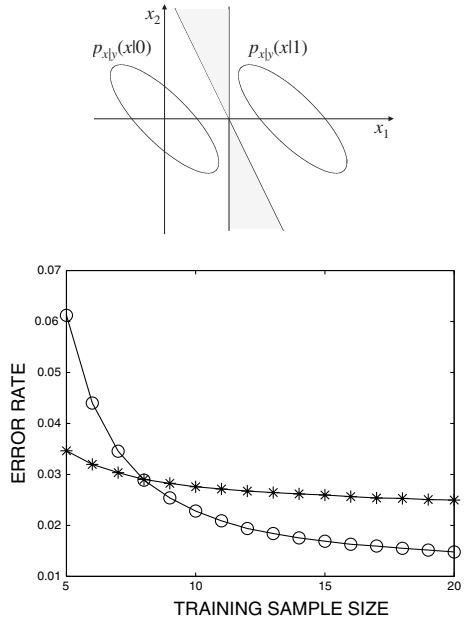


Fig. 2. (Top) Equi-level curves of two Gaussian densities used for the example of Section 3.2. The shaded area represents the region in which the naive Bayes classifier is biased. (Bottom) The error rate as a function of the training sample size for the Bayes classifier (circles) and for the naive Bayes classifier (stars).

distribution. Based on such computed values, and on the assumption that the distributions are Gaussian, a Bayes classifier[2] is designed (which in this case corresponds to the ML classifier). The error rate of this classifier, $P_B(E,n)$, is computed by Monte Carlo simulations (by feeding the classifier with a large number of test samples from the two distributions). Note that $P_B(E,n)$ is the expectation over $p_x(x)$ of the error (16) at each pixel. This error rate is larger than or equal to $P_B(E)$, due to the finite sample size. As the number of training samples n is reduced, $P_B(E,n)$ increases, due to the increase in the classifier's variance (see Fig. 2).

The same experiment is repeated but with the assumption that the conditional likelihoods are separable. As expected, the error rate for large samples is larger than in the previous case, a consequence of the fact that the naive Bayes classifier is biased for a large portion of the feature space (shown by the shaded area in Fig. 2). However, its error rate is less susceptible to the size of the training sample, due to its smaller variance. Indeed, a cross-over point between the two error curves is observed at $n = 8$. When the number of training samples per class is smaller than 8, the naive Bayes classifier outperforms the Bayes classifier – in spite of the fact that the conditional feature distributions are well-correlated ($\rho = 0.5$).

[2] Strictly speaking, a classifier designed based on a finite size training sample has a small likelihood of coinciding with the Bayes classifier. With some abuse of language, we will say that a classifier is Bayesian if it is generated by an algorithm that is asymptotically unbiased at each x.

4. Experiments

In order to validate the theoretical results of Section 3, we ran a number of cross-validation classification experiments over different data sets. Our main purpose was to evaluate how the empirical error rate correlates with the size of the training sample in the Bayes and Bayes fusion cases.

A number of practical decisions had to be taken at the time of experiment design, including the choice of the training and testing data sets, the visual features used for classification, the classification algorithms, and the evaluation criteria. All of these issues are discussed in the next subsections.

4.1. Image data sets

We have used three different image data sets for our experiments. Two of them contain images extracted from the sets used in [4] for experiments on texture classification. Selected image areas were hand-labeled and the classifiers were trained/tested only over those areas. The first data set, "Road", contains 18 outdoor images collected by a robotic vehicle. For this data set we considered three classes, corresponding to "soil", "tall vegetation", and "grass" (dry or green).

The second data set, "Rocks", is comprised of 11 images taken at JPL's "Mars Yard". These images contain a variety of different rocks and soil types. In [4], six different classes were identified for this data set. However, texture-based classification over such classes performed poorly. We therefore decided to use only two classes, "sand" and "rock", for which the correct classification rate using texture is more acceptable.

The third data set, "Trees", contains 20 images taken in UCSC's redwood's forest. Two classes were considered, namely, "tree trunk" and "soil cover". The colors of these two classes are very similar, since the soil was covered with mostly dry material. Hence, color was a much less reliable feature than texture in this case.

The original image files, together with the labeling used for our experiments, are publicly available at http://italia.cse.ucsc.edu/~manduchi/FusionDataSets. A representative image for each data set is shown in Fig. 3.

4.2. Visual features

We considered two visual features for our analysis: color and texture, which are co-located at each pixel. The perceived color is a function of the surface reflectivity and of the spectrum of the illuminant; its brightness is a function of the joint geometry of the illuminant and of the surface (and of the observer for non-Lambertian surfaces). Texture is somewhat independent of the spectrum of the illuminant, but depends heavily on the joint geometry of the surface and of the observer (and, to a lesser extent, of the illuminant). For one of the data sets ("Rocks"), color performed

Fig. 3. Representative images from the data sets "Road", "Rocks", and "Trees".

much better (in terms of misclassification rate) than texture, while the opposite is true for the data set "Trees". We used non-normalized (r, g, b) color features in our experiments (see [30] for a comparison of normalized and non-normalized colors in terms of misclassification rate). For texture, we used standard Gabor features [23]. Following [4], we filtered the images with a set of 8 (2 scales and 4 orientations) scaled and rotated version of a complex Gabor prototype kernel. We then extracted the magnitude of the output of each filter at each pixel, and smoothed the resulting texture field with a Gaussian kernel, with standard deviation proportional to the filter's scale. This provides us with a 8-dimensional feature vector for each pixel. Note that we did not enforce spatial coherence: each pixel was classified independently of the other pixels (although some degree of spatial correlation is implicit in the definition of texture). While spatial coherence is a powerful constraint, and usually leads to better classification rates, we felt that it would introduce one more level of complexity in our analysis, and make the results more difficult to interpret.

4.3. Classifier design

Bayesian classification requires an estimation of the class-conditional likelihood of the observations. We modeled such pdf's using mixtures of Gaussians (MoG), a standard procedure that has been used successfully in computer vision for color [16,22] and texture classification [4]. As for the class priors, we always assumed that they are uniform.

The MoG representation of the conditional likelihood $p_{x|y}(x|y)$ is

$$p_{x|y}(x|y) = \sum_{k=1}^{M(y)} \alpha_k G(x; \mu_k, \Sigma_k) \tag{20}$$

where $G(x; \mu_k, \sigma_k)$ is a Gaussian density with mean μ_k and covariance Σ_k, and α_k are positive constants such that $\sum_{k=1}^{M(y)} \alpha_k = 1$. In this work, all of the covariance matrices are constrained to be diagonal. This choice was made in order to reduce the overall number of free parameters, and therefore the variance of the estimate. Note that, even though the individual densities in the mixture are separable, the resulting conditional likelihood $p_{x|y}(x|y)$ is *not* separable in general.

It is instructive to compute the number of free parameters in our MoG models. Assume that there are N_C classes (where $N_C = 2$ for "Trees" and "Rocks", and $N_C = 3$ for "Road"), and let $x = (x_c, x_t)$ be the composite feature combining color and texture together (where x_c is the 3-dimensional color feature and x_t is the 8-dimensional texture feature.) In the MoG model for x there are $23\sum_{y=1}^{N_C} M_{ct}(y) - N_C$ parameters, where $M_{ct}(y)$ is the number of Gaussian modes used for class y. For the color and texture MoG models, the number of free parameters is $7\sum_{y=1}^{N_C} M_c(y) - N_C$ and $17\sum_{y=1}^{N_C} M_t(y) - N_C$, respectively. ML parameter estimation from the training data was carried out using the Expectation Maximization algorithm [6]. To reduce the risk of ending up in a local maximum of the likelihood surface, the procedure was initialized with values obtained from previous k-means clustering, as recommended in [31]. More precisely, k-means was run 20 times on the data starting from random values; the clustering that minimized the mean distance between the cluster centers and the feature vectors in the clusters was used for EM initialization.

One important issue with MoG models is the choice of the number of Gaussians modes to represent each conditional likelihood. Too few modes lead to inaccurate modeling, while too many modes may overfit the training data. We used Smyth's algorithm for model selection, which compares favorably against other algorithms such as BIC and vCV [31]. This algorithm is based on Monte Carlo cross-validation. The data is first randomly divided into two disjoint training and testing subsets; then, for each possible number of clusters within a pre-defined interval, the training subset is used to estimate the MoG parameters, and the resulting model is used to compute the log-likelihood of the testing data. This is repeated several times (in our work, 20–50 times). The number of modes is chosen by analyzing the log-likelihood (averaged over all iterations) for the different model dimensions. If the data is actually distributed according to a MoG model, the "correct" number of modes usually stands up as the maximizer of the average log-likelihood. However, in practice, a clear maximum cannot always be found within an interval of reasonably small model dimensions. Thus, rather than looking for the maximum, we decided to select the smallest dimension $M(y)$ such that the relative change in log-likelihood between $M(y)$ and $M(y) - 1$ is less than 2%.

The major practical difficulty in implementing such extensive cross-validation experiments lies in the amount of computing time required, which can limit the actual number of tests. In order to alleviate this problem, we implemented the fast cross-validation algorithm (BRACE) proposed by Moore and Lee [25]. The basic idea is to run the cross-validation experiments for different models (in this case, different number of Gaussian modes) in parallel. Models that, at a certain point in the experiments, are believed to have little likelihood of success, are eliminated from the "race". The underlying assumption is that unsuccessful models can be identified at early stages of the process by suitable statistical tests.

For each data set, we used this procedure to determine the dimension of the MoG for each class and for each selected portion of the training data used in the experiments. The results are reported in Figs. 5–7. Note that the number of modes selected never decreases as the training sample ratio increases. This can be justified intuitively based on the bias/variance considerations of Section 3. Smaller training sets lead to higher parameter estimation variance, hence simpler models (lower dimension) are preferred in these cases. A classification example for an image of the "Road" data set is shown in Fig. 4.

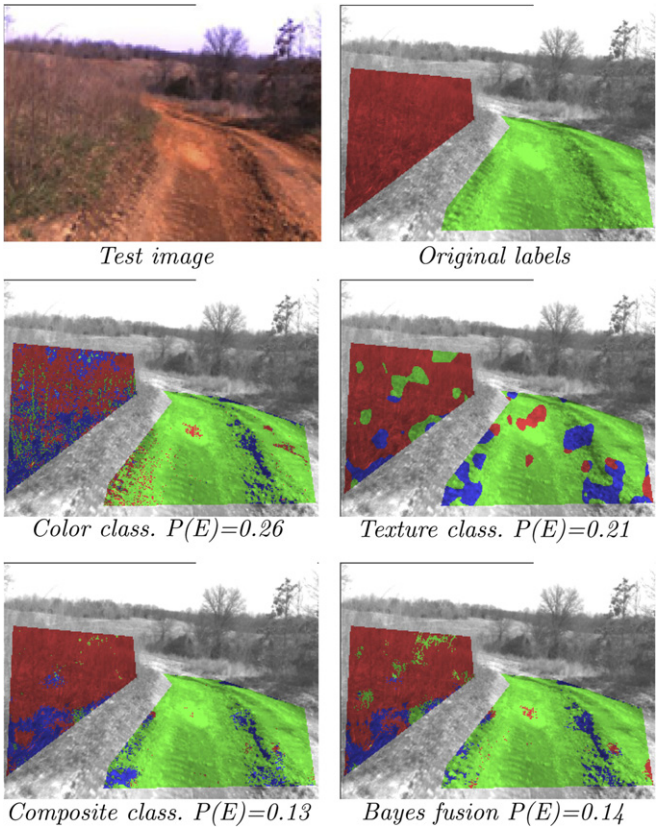
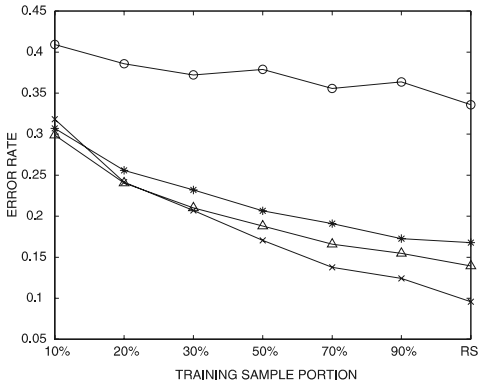


Fig. 4. A classification example from the "Road" data set. The system was trained over 25% of the images in the set. This image was not part of the training set. The classes are color-coded as follows: "soil", green;"tall vegetation", blue; "grass", red.

4.4. Performance metrics

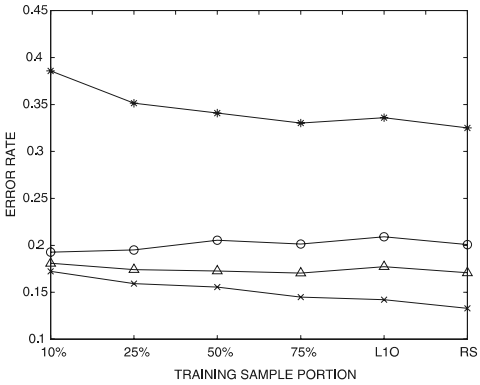
Our classifiers were trained and tested over each data set separately. The misclassification rate was estimated based on the confusion matrix computed by comparing the classification with the original hand-labeling. The confusion matrix at entry (i,j) contains the number of pixels hand-labeled as i and classified as j. In the case of several tests, we accumulate (add together) the corresponding confusion matrices. To account for the possibly different number of pixels across labels, we normalize each row of the cumulative confusion matrix so that it sums up to 1. Our estimate for the error rate is the mean of the entries in the diagonal of the normalized confusion matrix.

In order to study the dependence of the error rate of the two algorithms (Bayes classification and Bayes fusion of classifiers) on the size of the training set, we ran a number of different cross-validation experiments. First, we trained and tested the system over all the available data within each data set (i.e., the data that was hand-labeled). This approach (sometimes called the re-substitution estimate [2] and marked as RS in Figs. 5–7) is known to be biased (optimistic). Still, it gives useful indications about the system performance in the best-case scenario. We then used a K-fold cross-validation strategy [24]: for each data set with K images, we trained the system over K-1 images, and tested it on the remaining image. With some abuse of language, we call this a "leave-one-out" (L1O) procedure. In order to test over reduced training data sets, we used the following Monte Carlo cross-validation method. We randomly selected (without replacement) n% (with n varying between 10 and 90) training images out of each



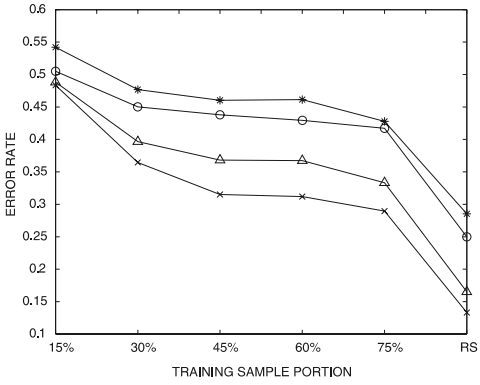
	10%	20%	30%	50%	70%	90%	RS
Color	2/2	2/2	3/3	3/3	4/3	4/3	4/3
Texture	1/1	1/2	2/2	5/2	6/2	7/2	7/2
Composite	2/2	2/2	5/3	6/4	6/5	8/5	10/6

Fig. 5. (Top) The misclassification rate for the data set "Trees" as a function of the training sample ratio using color (○), texture (✻), composite color/texture features (×), and Bayes fusion of color and texture (△). (Bottom) Number of Gaussian modes used for color, texture, and composite color/texture features (Class 0/Class 1) as a function of the proportion of data used for training.



	10%	25%	50%	75%	L1)	RS
Color	4/4	4/4	5/4	5/4	5/4	6/4
Texture	4/3	5/4	5/4	5/4	5/4	5/4
Composite	5/3	5/4	6/4	6/4	7/4	7/5

Fig. 6. (Top) The misclassification rate for the data set "Rocks" as a function of the training sample ratio using color (○), texture (✱), composite color/texture features (×), and Bayes fusion of color and texture (△). (Bottom) Number of Gaussian modes used for color, texture, and composite color/texture features (Class 0/Class 1) as a function of the proportion of data used for training.



	15%	30%	45%	60%	75%	RS
Color	2/2/2	2/2/2	2/2/2	2/2/3	2/3/3	2/4/4
Texture	2/2/2	2/2/2	2/3/2	2/4/3	2/4/3	2/4/4
Composite	3/2/2	4/2/2	4/2/3	4/3/4	4/3/4	5/5/6

Fig. 7. (Top) The misclassification rate for the data set "Road" as a function of the training sample ratio using color (○), texture (✱), composite color/texture features (×), and Bayes fusion of color and texture (△). (Bottom) Number of Gaussian modes used for color, texture, and composite color/texture features (Class 0/Class 1/Class 2) as a function of the proportion of data used for training.

data set, and used the remaining ones for testing. We then repeated the procedure 20 times (except for the re-substitution and the leave-one-out estimates, in which case the number of iterations is dictated by the size of the data set). The results are then averaged together. Note that we randomly selected whole images rather pixels from the cumulative training set. This seems like a more natural procedure in computer vision, where one typically accesses a whole image at a time, rather than just isolated pixels. However, due to the high correlation among neighboring features, this method may lead to somewhat pessimistic results, since the training samples for each label cannot be considered really independent.

The resulting error rates for the three data sets using color features, texture features, composite color–texture features, and Bayes fusion, are shown[3] in Figs. 5–7. As expected, the error rate for each feature, feature combination, and Bayes fusion, increases (in general) as the training data portion decreases, due to the increasing variance of the corresponding classifier. A first interesting observation is that, even when one of the two features gives bad results (e.g., color for the "Trees" data set and texture for the "Rocks" data set), combining features together (whether as a composite super–vector or by Bayes fusion) always decreases the error rate with respect to the better performing feature.

Comparing the performances of Bayes classification and of Bayes fusion, one observes that:

(1) When the proportion of the training data is large, the former always outperforms the latter;
(2) As the training data size is reduced, the error rates in the two cases become closer to each other. This is particularly apparent in the "Trees" and "Road" data sets (Figs. 5 and 7), where a cross-over point can be found, similarly to the case of Fig. 2.

5. Conclusions

We presented an analysis of the performances of Bayes fusion of classifiers as compared to Bayes classification. A theoretical analysis relating the error rates in the two cases for a simple Gaussian scenario has been provided. These results, however, apply only to the ideal case in which the statistical description of the data is perfectly known. For the more realistic case of finite size training samples, we introduced a new expression for the bias/variance dilemma, which casts light on the sometimes surprising small sample effects. Empirical results on three image data sets using color and texture features show that Bayes fusion has a higher error rate than Bayes classification for large training samples, but that the difference between the two becomes small (or negative) for smaller samples.

Acknowledgments

The authors thank Dr. Bruno Sansò for his help during the initial phase of this work. This work was supported by DARPA MARS through Subcontract 1235249 from JPL and by NSF through contract IIS-0222943.

[3] Note that in the third case ("Road") there are three classes, which explains why the misclassification rate may be as high as 50% or higher (but less than 66%).

Appendix A

In this Appendix we derive explicit formulas for $P_{\mathrm{B}}(E)$, the error rate of the Bayes classifier, and $P_{\mathrm{NB}}(E)$, the error rate of the naive Bayes classifier, for a simple case of classification into two classes with equi-covariant distributions and equal priors.

We will assume that the class-conditional likelihood for Class 0 is a Gaussian pdf, with zero mean and covariance $\Sigma = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{12} & \sigma_{22}^2 \end{pmatrix}$. The class-conditional likelihood for Class 1 is Gaussian, with mean $\mu = (\mu_1, \mu_2)'$ and covariance Σ. We will further assume that the two classes have the same prior probability.

We will begin with the computation of $P_{\mathrm{B}}(E)$, which in this case is a simple exercise [28]. The trick is to find a change of basis that makes our computation simple. Note that an invertible transformation in the feature space does not change the Bayes rate [30], and therefore we can compute $P_{\mathrm{B}}(E)$ in the transformed space.

Let

$$\Sigma = R' D_1^2 R \tag{21}$$

be the singular value decomposition of the positive definite matrix Σ, where D_1 is diagonal and R is a rotation matrix. After change of basis $\bar{x} = D_1^{-1} R x$, the Gaussian distributions are "standardized" to unit covariance. The mean of the Class 1 distribution transforms into $\bar{\mu} = D_1^{-1} R \mu$ (see Fig. 8). The Bayes classification boundary is the line $\bar{L}_B = \bar{\mu}/2 + a\bar{\xi}$, where

$$\bar{\xi} = T\bar{\mu} = T D_1^{-1} R \mu \tag{22}$$

is a vector orthogonal to $\bar{\mu}$ (T being the antisymmetric matrix operating a $\pi/2$ rotation: $T = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$). Computing the error rate $P_{\mathrm{B}}(E)$ is trivial after rotating the axes to bring the point $\bar{\mu}$ onto the horizontal axis (thus making the classification boundary a vertical line):

$$P_{\mathrm{B}}(E) = \int_{-\frac{\|\bar{\mu}\|}{2}}^{\infty} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{\frac{z^2}{2}}\, \mathrm{d}z \tag{23}$$

Now, $\|\bar{\mu}\|^2 = \bar{\mu}'\bar{\mu} = \mu'\Sigma^{-1}\mu$, and therefore (23) can be rewritten as

$$P_{\mathrm{B}}(E) = 1 - \Phi\left(\frac{\sqrt{\mu'\Sigma^{-1}\mu}}{2}\right) \tag{24}$$

where $\Phi(\cdot)$ is the error function. Thus, the Bayes rate depends only on the Mahalanobis distance between the means of the two Gaussians.

The classification boundary for the naive Bayes classifier, in the transformed space $\bar{x}$, is a line $\bar{L}_{\mathrm{NB}}$ intersecting the segment $[0, \bar{\mu}]$ halfway at an angle θ with respect to $\bar{L}_{\mathrm{B}}$. When $\theta \neq 0$, the error rate of this classifier is thus:

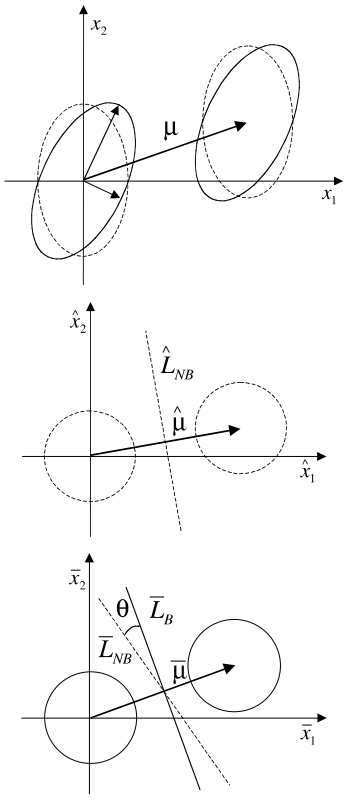


Fig. 8. (Top) Equi-level curves of the two Gaussian densities considered in the Appendix (continuous line) and of the two separable densities that are the outer product of the marginal densities (dashed line). (Center) The standardized version of the two separable densities in the $(\hat{x}_1, \hat{x}_2)$ space, together with the classification boundary line $\hat{L}_{\mathrm{NB}}$. (Bottom) the standardized version of the original densities in the $(\bar{x}_1, \bar{x}_2)$ space, together with the classification boundary lines $\bar{L}_{\mathrm{B}}$ and $\bar{L}_{\mathrm{NB}}$.

$$P_{\mathrm{NB}}(E) = 1 - \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{e}^{-\frac{z_1^2}{2}} \int_{-\infty}^{\frac{z_1 - \sqrt{\mu'\Sigma^{-1}\mu}/2}{|\tan\theta|}} \mathrm{e}^{-\frac{z_2^2}{2}}\, \mathrm{d}z_1\, \mathrm{d}z_2 \tag{25}$$

We now proceed to compute the angle θ. First, note that the naive Bayes classifier is designed "pretending" that each distribution is separable. More precisely, instead of the original densities, the outer products of the marginal densities are used. This is equivalent to assuming that each Gaussian has covariance $D_2^2 = \begin{pmatrix} \sigma_{11}^2 & 0 \\ 0 & \sigma_{22}^2 \end{pmatrix}$. Thus, in the transformed space of $\hat{x} = D_2^{-1} x$, the classification boundary is the line $\hat{L}_{\mathrm{NB}} = \hat{\mu}/2 + \alpha\hat{v}$, where $\hat{\mu} = D_2^{-1}\mu$ and $\hat{v} = T D_2^{-1}\mu$. Transformed into the space of $\bar{x}$, this line becomes $\bar{L}_{\mathrm{NB}} = \bar{\mu}/2 + \alpha\bar{v}$, where

$$\bar{v} = D_1^{-1} R D_2' D_2^{-1} \mu \tag{26}$$

The angle θ in Fig. 8 is thus the angle between $\bar{\xi}$ in (22) and $\bar{v}$ in (26):

$$\cos\,\theta = \frac{\bar{\xi}'\bar{v}}{\|\bar{\xi}\|\|\bar{v}\|} \tag{27}$$

Simple but tedious computation proves the following identities:

$$\bar{\xi}'\bar{v} = \mu' R' D_1^{-1} T' D_1^{-1} R D_2 T D_2^{-1} \mu = \sqrt{1-\rho^2}\left[\mu'\Sigma^{-1}\mu + \frac{2\sigma_{12}\mu_1\mu_2}{\det\Sigma}\right] \quad (28)$$

$$\|\bar{\xi}\| = \sqrt{\mu' R' D_1^{-1} T' T D_1^{-1} R \mu} = \sqrt{\mu'\Sigma^{-1}\mu} \quad (29)$$

$$\|\bar{v}\| = \sqrt{\mu' D_2^{-1} T' D_2 R' D_1^{-1} D_1^{-1} R D_2 T D_2^{-1}\mu} = \sqrt{\mu'\Sigma^{-1}\mu + \frac{4\sigma_{12}\mu_1\mu_2}{\det\Sigma}} \quad (30)$$

where ρ is the correlation betwen x_1 and x_2:

$$\rho = \frac{\sigma_{12}}{\sigma_1\sigma_2} \quad (31)$$

Hence,

$$\cos\theta = \sqrt{1-\rho^2}\,\frac{1+2K}{\sqrt{1+4K}} \quad (32)$$

with

$$K = \frac{\sigma_{12}\mu_1\mu_2}{\mu'\Sigma^{-1}\mu \det\Sigma} \quad (33)$$

A few observations can immediately be drawn from (32):

- When $\sigma_{12} = 0$ (i.e., if the original distributions are separable), then also $\rho = 0$ and therefore $\theta = 0$. In this case, as expected, the Bayes and the naive Bayes classifiers coincide.
- It is easy to see from (32) and (33) that $\cos^2\theta \geqslant 1-\rho^2$ and therefore

$$|\tan\theta| \leqslant \frac{|\rho|}{\sqrt{1-\rho^2}} \quad (34)$$

The above expression becomes an identity when $\mu_1 = 0$ or $\mu_2 = 0$, that is, when the mean of the second Gaussian lies on one of the axes. Plugging (34) into (25), one obtains an upper bound on the error rate of the naive Bayes classifier as a function of the correlation ρ and of the Mahalanobis distance $\mu'\Sigma^{-1}\mu$ (see Fig. 1).

References

[1] F.M. Alkoot, J. Kittler, Experimental evaluation of expert fusion strategies, Pattern Recognition Letters 20 (1999) 1361–1369.

[2] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth & Brooks, Pacific Grova, CA, 1984.

[3] L. Breiman, Bias, Variance and Arcing Classifiers, Tech. Report 460, Statistics Department, UC Berkeley, 1996.

[4] R. Castano, R. Manduchi, J. Fox, Classification experiments on real-world textures, Workshop on Empirical Evaluation in Computer Vision, Kauai, HI, 2001.

[5] W.S. Cooper, Some inconsistencies and misidentified modeling assumptions in probabilistic information retrieval, ACM Trans. on Information Systems 1 (1) (1995) 100–111.

[6] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society B 39 (1) (1977) 1–38.

[7] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero–one loss, Machine Learning 29(2/3)(1997)103–130.

[8] P. Domingos, A unified bias-variance decomposition for zero–one and squared loss, AAAI (2000) 564–569.

[10] J.H. Friedman, On bias, variance, 0/1-loss, and the curse-of-dimensionality, Data Mining and Knowledge Discovery 1 (1997) 55–77.

[11] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine Learning 29 (1997) 131–163.

[12] S.L. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural Computation 4 (1992) 1–58.

[15] International Workshop on Multiple Classifier Systems, <http://www.informatik.uni-trier.de/ley/db/conf/mcs/index.html>.

[16] M.J. Jones, J.M. Rehg, Statistical color models with application to skin detection, Cambridge Research Laboratory CRL 98/11.

[17] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Transactions on PAMI 20 (3) (1998).

[18] J. Kittler, A.A. Hojjatoleslami, A weighted combination of classifiers employing shared and distinct representations, Proceedings of CVPR, 1998, pp. 924–929.

[19] E.B. Kong, T.G. Dietterich, Error-correcting output coding corrects bias andvariance, Proceedings of the International Conference on Machine Learning, 1995, pp. 313–321.

[20] P. Langley, Induction of selective Bayesian classifiers, Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, Seattle, WA, 1994.

[21] D.D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval", *Proc. of ECML-98, 10th European Conference on Machine Learning*, 1998.

[22] R. Manduchi, A. Castano, A. Talukder, L. Matthies, Obstacle detection and terrain classification for autonomous off-road navigation, Autonomous Robots 18 (2005) 81–102.

[23] B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of image data, IEEE Transactions on PAMI 18 (8) (1996) 837–842.

[24] J.K. Martin, D.S. Hirschberg, Small sample statistics for classification error rates. I. Error rate measurements, Department of Information and Computer Science, UC Irvine, Tech. Report 96-21, 1996.

[25] A.W. Moore, M.S. Lee, Efficient algorithms for minimizing cross validation error, International Conference on Machine Learning (1994) 190–198.

[26] S. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Transactions on PAMI 13 (3) (1991) 252–264.

[27] S. Raudys, On dimensionality, sample size, and classification error of non–parametric linear classification algorithms, IEEE Transactions on PAMI 19 (6) (1997) 337–371.

[28] B.D. Ripley, Pattern recognition and neural networks, Cambridge University Press, London, 1996.

[29] X. Shi, R. Manduchi, A study on Bayes feature fusion for image classification, IEEE Workshop on Statistical Analysis in Computer Vision, Madison, Wisconsin, June 2003.

[30] X. Shi, R. Manduchi, Invariant operators, small samples, and the Bias/Variance dilemma, Proceedings of IEEE CVPR, Washington, DC, July 2004.

[31] P. Smyth, Clustering using Monte Carlo cross-validation, Proceedings of Knowledge Discovery and Data Mining, 1996, pp. 126–133.

[32] D. M. J. Tax, R.P.W. Duin, M. VanBreukelen, Comparison between product and mean classifier combination rules, Proceediongs of 1st International Workshop Statistical Techniques in Pattern Recognition, 1997, pp. 165–170.

[33] R. Tibshirani, Bias, variance and prediction error for classification rules, Techical Report, Department of Preventive Medicine and Biostatistics, University of Toronto, 1996.

[34] M. Tsuyuguchi, K. Uehara, Bias–Variance decomposition of zero–one loss in average-case model, Proceedings of AMAI 2002, 2002.

[35] D. Wolpert, On bias plus variance, Neural Computation 9 (1997) 1211–1243.