A Study on Bayes Feature Fusion for Image Classification

X. Shi and R. Manduchi
Department of Computer Engineering
University of California, Santa Cruz
{jennifer,manduchi}@soe.ucsc.edu

Abstract

We consider here the problem of image classification when more than one visual feature are available. In these cases, Bayes fusion offers an attractive solution by combining the results of different classifiers (one classifier per feature). This is a general form of the so-called "naive Bayes" approach. Analyzing the performance of Bayes fusion with respect to a Bayesian classifier over the joint feature distribution, however, is tricky. On the one hand, 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. On the other hand, as noted by Friedman, the low variance associated with naive Bayes estimation may dramatically mitigate the effect of its bias. In this paper, we attempt to assess the trade-off between these two factors by means of experimental tests on two image data sets using color and texture features. Our results suggest that (1) the difference between the correct classification rates using Bayes fusion and using the joint feature distribution is a function of the conditional dependence of the features (measured in terms of mutual information), however: (2) for small training data size, Bayes fusion performs almost as well as the classifier on the joint distribution.

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: a set of local features (such as color, texture, velocity) are available for each pixel. Our work deals with the *statistical fusion* of such features, to obtain a classification which is hopefully more reliable than by using just one feature.

In principle, one may simply create a super–feature as the juxtaposition of all the available feature vectors. However, in many practical cases one may prefer to analyze the different features separately, and later fuse together the results of the classifications. One motivation for this second approach is that it allows one to use different classification strategy for different features. Each classifier may be optimized for the characteristics of the feature it operates on.

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 rule: the multiplicative rule on the posterior probabilities (or *Bayes fusion*). When the different features are single–dimensional, Bayes fusion of classifiers is usually called *naive Bayes* classification. It is well–known that the Bayes fusion of classifiers (or naive Bayes classification) coincides with Bayesian classification over the "composite" feature only if the individual features are conditionally independent for any given class.

In this work we study the performance of Bayes fusion for a case of interest in computer vision: the classification of image pixels based on color and texture features. Color and texture give us different (and somewhat complementary) pieces of information about the characteristics of a surface. The combination of color and texture is often used for image retrieval and remote sensing, and has potential for application in other fields such as robotics and biometrics.

Our work was originated by the question of how well (or badly) Bayes fusion of classifiers compares with Bayesian classification over the composite feature. As noted by Friedman [1], the bias associated with the conditional independence approximation may actually be offset by the low variance of estimation. In addition, we were interested in actually measuring the conditional dependence between features, and studying its influence on the classification error rate of Bayes fusion. The results of our experiments over two different image data sets suggest that (1) the conditional dependence (as measured by the mutual information between features) may have a role in the overall error rate, and that (2) when the size of the training data is small, Bayes fusion performs almost as well as Bayesian classification over the composite features.

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, and discussing issues related to the bias/variance dilemma. Section 3 describes our experimental set up and results. Section 4 has the conclusions.

1.1. Previous Work

The naive Bayes classifier has been studied in pattern recognition, machine learning, sensor fusion and information retrieval for several decades. As noted by Lewis [5], there have been mostly two research directions in the field of naive Bayes in both machine learning and information retrieval.

The first direction attempts to modify the feature set so that the resulting features are as independent as possible. Typical work includes [4], which uses a greedy algorithm to select a subset of features that are only weakly correlated. N. Friedman [6] proposed an algorithms 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 doesn't hold [7, 1, 8]. In particular, J. Friedman studies naive Bayes classifiers in the context of the bias/variance dilemma [9].

Bayes fusion of classifiers in the context of computer vision has been studied by Kittler [10, 11], who experimented with handwritten character recognition. Kittler showed, both theoretically and empirically, that the multiplicative rule of Bayes fusion is less sensitive to estimation errors than other rules.

Finally, note that naive Bayes and Bayes fusion system are instances of Multiple Classifier Systems, which have received a good deal of attention recently (see for example [12]).

2. Bayes Fusion - Theory

This section covers some important theoretical aspects of Bayes fusion. In Section 2.1 we assume that the statistical quantities of interest (density functions and posterior distributions) are known, and study the performance of the Bayes fusion algorithm in such an ideal case. In Section 2.2 we will consider the more realistic case of estimation from finite–size samples, and look at Bayes fusion in the context of the bias/variance dilemma.

Throughout this work, f will indicate a generic feature to be classified. We will assume that a feature is generated by one class in a pre–determined set of classes. The probability density function of f is indicated by $p(f)$, while $p(f|y) = p(f,y)/P(y)$ is the conditional likelihood of f given the class y ($P(y)$ being the prior probability of class y), and $P(y|f) = p(f,y)/p(f)$ is the posterior probability of class y given that the feature f was observed.

2.1. Conditional Independence and Error Rate

The error rate of a classifier over two classes $y = 0$ and $y = 1$ is

$$P(E) = \int_{f\in\mathcal{I}_0} p(f,1)\, df + \int_{f\in\mathcal{I}_1} p(f,0)\, df \qquad (1)$$

where $\mathcal{I}_0$ and its complement, $\mathcal{I}_1$ are the assignment regions (i.e., feature f is assigned to class 0 if $f \in \mathcal{I}_0$, to class 1 otherwise). The Bayesian classifier defines region $\mathcal{I}_0$ by

$$\mathcal{I}_0 = \{f : p(f,0) > p(f,1)\} \qquad (2)$$

If $f = (f_1, f_2)$, the Bayes fusion of classifiers uses the following assignment rule:

$$\hat{\mathcal{I}}_0 = \left\{ f : \frac{p_1(f_1,0)p_2(f_2,0)}{P(0)} > \frac{p_1(f_1,1)p_2(f_2,1)}{P(1)} \right\} \qquad (3)$$

where $p_1(f_1, y)$ is the marginal density $\int_{-\infty}^{\infty} p(f,y)df_2$. Note that (3) corresponds to assigning feature f to class y=0 if $\frac{P_1(0|f_1)P_2(0|f_2)}{P(0)} > \frac{P_1(1|f_1)P_2(1|f_2)}{P(1)}$. If f_1 and f_2 are single–dimensional, then Bayes fusion coincides with the naive Bayes classifier. For the sake of simplicity, we will assume that this is the case throughout this section.

It is well known that if the conditional densities $p(f|0)$ and $p(f|1)$ are separable, i.e. if $p(f|y) = p_1(f_1|y)p_2(f_2|y)$, then the naive Bayes classifier coincides with the regular Bayesian classifier. This situation is called of *conditional independence*, because it means that the 1-D features f_1 and f_2 are statistically independent for any given class.

The conditional independence assumption, while not as strong as total independence, is probably unrealistic in most cases of interest. However, even when features are not conditionally independent, naive Bayes classification does not necessarily yield catastrophic results. Approximating $p(f|i)$ with $p_1(f_1|i)p_2(f_2|i)$ leads to choosing sub–optimal assignment regions. The features f that contribute to the increased classification error rate are those in the regions $\hat{\mathcal{I}}_0 \cap \mathcal{I}_1$ and $\hat{\mathcal{I}}_1 \cap \mathcal{I}_0$. Indeed, if $P_{NB}(E)$ and $P_B(E)$ are the error rates of the naive Bayes and of the Bayesian classifier respectively, their (non–negative) difference is

$$\Delta P(E) = P_{NB}(E) - P_B(E) \qquad (4)$$

$$= \int_{\hat{\mathcal{I}}_0\cap\mathcal{I}_1} (p(f,1) - p(f,0))\, df + \int_{\hat{\mathcal{I}}_1\cap\mathcal{I}_0} (p(f,0) - p(f,1))\, df$$

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

$$\Delta P(E) = \int_{(\hat{\mathcal{I}}_0\cap\mathcal{I}_1)\cup(\hat{\mathcal{I}}_1\cap\mathcal{I}_0)} |2P(1|f) - 1|\, p(f)\, df$$

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

Intuitively, one may expect that $\Delta P(E)$ should be higher when f_1 and f_2 are tightly correlated. Finding a general formula relating $\Delta P(E)$ to any particular measure of correlation is probably impossible. However, for simple cases one can actually find analytical solutions. For example, in the Appendix we describe the case of $p(f,0)$ and $p(f,1)$ being Gaussian distributions with the same covariance but different means and 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 f_1 and f_2. This agrees nicely with our intuition; unfortunately, it cannot be generalized in a simple way to even slightly more complex cases. Still, this result strengthens our conjecture that there should be some relationship between statistical dependence and $\Delta P(E)$.

An important question is: What is a sensible metric for the "conditional dependence"? In the simple case discussed in the Appendix, second–order statistics were sufficient to bound the error rate. This is not surprising, given the Gaussian hypothesis of our case study. In more realistic cases one should take into account two main factors. First, the class–conditional densities may be (and usually are) non–Gaussian, and possibly multi–modal. Second, f_1 and f_2 may have different dimensionality. For example, in our experiments, f_1 is a three–dimensional color vector, and f_2 is a eight–dimensional texture vector.

We propose to use a more general measure of conditional independence, based on the mutual information between f_1 and f_2. For a given class y, the (conditional) mutual information is

$$MI(y) = E\left[\log \frac{p(f|y)}{p_1(f_1|y)p_2(f_2|y)}\right] \tag{5}$$

where the expectation is computed over the conditional density $p(f|y)$. Note that the mutual information is well defined also when f_1 and f_2 are vectors with different dimensionality. It is well known that the mutual information is non–negative (being a Kullback–Leibler distance) and that

$$MI(y) = H_1(y) + H_2(y) - H(y) \tag{6}$$

where $H_1(y)$, $H_2(y)$ and $H(y)$ are the class–conditional differential entropies of $p_1(f_1|y)$, $p_2(f_2|y)$ and $p(f|y)$ respectively. Note that the entropy of a N–dimensional Gaussian density with covariance Σ is

$$H = 0.5\left(N + N\log(2\pi) + \log\det\Sigma\right)$$

Thus, for the case considered in the Appendix, we may relate the mutual information to the correlation ρ as follows:

$$\rho(y) = \sqrt{1 - \exp(-2\,MI(y))} \tag{7}$$

For non–Gaussian variables, the quantity $\rho(y)$ as defined in (7) does not usually correspond to the correlation between f_1 and f_2. Still, we may use it as a convenient way to represent the mutual information (note that $0 \le \rho \le 1$ while the mutual information is not bounded from above). We will call $\rho(y)$ as in (7) the *generalized correlation* between two variables (or vectors) f_1 and f_2 given class y. In Section 3 we study the relationship between generalized correlation and $\Delta P(E)$ over two different data sets.

2.2. Bias and Variance

The previous section made a very strong assumption, that the conditional likelihoods (whether joint or marginal) are known. In practice, such functions are estimated from a finite sample T, and the estimates are random variables themselves. Such randomness plays an important role in the performances of the classifier and, according to Friedman [1], may explain why naive Bayes classifiers perform well in spite of their obvious bias.

A powerful characterization of the system performances in the case of small sample size is the bias/variance decomposition of the mean–squared estimation error [9]. For example, assume that in our two–class case, f is generated by class y. We can define the squared estimation error as $(y - P_T(1|f))^2$, where the suffix T represents the training sample set used for estimating the posterior probability. Averaging over repeatedly realized training samples T, one obtains the following well–known decomposition of the mean–squared estimation error [9]:

$$E_T\left[(y - P_T(1|f))^2\right] = E_\epsilon[\epsilon|f] + \tag{8}$$
$$+ \left(P(1|f) - \bar{P}(1|f)\right)^2 + E_T\left[\left(P_T(1|f) - \bar{P}(1|f)\right)^2\right]$$

where $\bar{P}(1|f) = E_T[P_T(1|f)]$ and $\epsilon = y - P(1|f)$. The first term of the right hand side of (8) is the irriducible prediction error, due to the randomness of the feature. The second term is the square of the *bias*. "Simple" estimators such as naive Bayes are typically highly biased. The third term is the *variance* of the estimator, and reflects the sensitivity of the estimate to the training sample T. A typical example of high variance is "overfitting", which happens when a system "memorizes" the training data rather than "generalizing" from it.

While equation (8) is rather intuitive and general, it does not give us a clear answer in terms of classification rate. To this purpose, we should replace the square cost function with the 0–1 loss function. Several authors have studied the bias/variance dilemma for this case [13, 14, 1]. We refer to the work of Friedman [1], which is particularly intuitive, and briefly summarize its main points in the following.

Friedman showed that, for a given feature f, generated by class y, and averaging over all traning samples T of same size, the probability of classification error is

$$P(E(f)) = P(y \neq \hat{y}(f)) = \tag{9}$$

$$= P(y_B(f) \neq y) + P(\hat{y}(f) \neq y_B(f))|2P(1|f) - 1|$$

where $y_B(f)$ is the Bayesian classification using the "real" posterior distribution, and $\hat{y}(f)$ is the classification based on the limited training data set. Equation (9) decomposes the probability of misclassification (for a given feature f) into the sum of two parts: the irriducible error rate using an "ideal" Bayesian classifier, and a function of the *boundary error* $P(\hat{y}(f) \neq y_B(f))$. The boundary error is the equivalent of the sum of squared bias and variance in (8). Friedman proposed the following approximation to the boundary error:

$$P(\hat{y}(f) \neq y_B(f)) \approx \Phi\left(\frac{bb}{\sqrt{(}\text{Var } P(1|f))}\right) \qquad (10)$$

where $\Phi(x)$ is the Gaussian cumulative distribution function, Var $P(1|f)$ is the variance of the estimator (corresponding to the last factor of the right–hand side of (8)), and bb is the *boundary bias*:

$$bb(f) = -\text{sign}(P(1|f) - 0.5)(\bar{P}(1|f) - 0.5) \qquad (11)$$

If $\bar{\mathcal{I}}_0$ and its complement, $\bar{\mathcal{I}}_1$, are the assignment regions induced by the posterior distribution $\bar{P}(1|f)$, then one easily sees that the boundary bias $bb(f)$ is positive if $f \in (\mathcal{I}_0 \cap \bar{\mathcal{I}}_1) \cup (\mathcal{I}_1 \cap \bar{\mathcal{I}}_0)$, negative otherwise. In other words, the sign of $bb(f)$ indicates whether a classifier based on $\bar{P}(1|f)$ (the posterior probability averaged over all training sets) gives the same answer as the Bayesian classifier, which is based on $P(1|f)$.

Equation (10) highlights two important (and somewhat counter–intuitive at first) facts. First, the bias (as defined in (8)) does not directly affect the error rate. Indeed, the only dependence on the posterior distribution $P(1|f)$ is through the sign of $(P(1|f) - 0.5)$. Second, for negative boundary bias, the error rate decreases with decreasing variance. However, it the boundary bias is positive, the error rate *increases* with decreasing variance.

Thus, regardless of the actual estimation bias, a classification procedure is successful if the boundary bias is predominantly negative, and the variance is small enough. This may explain why highly biased estimators, such as naive Bayes, can perform surprisingly well in spite of their high bias. Indeed, Friedman argues that "oversmoothing" procedures (such as naive Bayes) have normally negative boundary biases, and therefore the estimation variance is likely to dominate the classification performances. If the variance of the naive Bayes estimator is small enough, one may expect that its performances be close to that of the Bayesian classifier. In the next section, we verify this conjecture empirically, by looking at the performances of both classifiers as a function of the size of the training sets T.

3. Experiments

Our experiments had two main purposes: (1) To study the dependence of $\Delta P(E)$ (that is, the difference between the error rate of Bayes fusion of classifiers and Bayesian classification) on the mutual information between features, and: (2) To compare the sensitivity of the two classification strategies to the size of the training data.

Of course, a number of practical decisions needed to be taken for such experiments, including choosing the train/test data sets, the visual features, the classification algorithms, and the criteria for testing. All of these issues are discussed in the following.

3.1. Image Data Sets

We have selected two different image data sets for our experiments. These images were extracted from the sets used in [15] 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 images collected by a robotic vehicle in outdoor environments. For this data set we considered three classes, corresponding to, "soil" (class 1), "tall vegetation" (class 2) , and "grass" (dry or green – class 3).

The second data set, "Rock", contains 11 images taken at JPL's "Mars Yard". These images contain a variety of different rocks and soil types. In [15], six different classes were identified for this data set. However, texture–based classification over these classes performed rather poorly. Instead, we decided to use only two classes, "sand" (class 1) and "rock" (class 2), for which the correct classification rate using texture is more acceptable.

The original data, together with the label files, is available at http://italia.cse.ucsc.edu/~ manduchi/SACV03.

3.2. Visual Features

We considered two visual features: color and texture. This choice was very attractive for three main reasons. First. color and texture features are co–located at each pixel, which makes the fusion scheme pretty straightforward and allows us to test on a very large number of samples. Second, color and texture complement each other nicely. The perceived color is a function of the surface reflectivity and of the spectrum of the illuminant; its brightness is a a function of the joint geometry of the illuminant and of the surface (and of the observer for non-Lambertian surfaces). Texture is pretty much independent of the spectrum of the illuminant, but is heavily dependent on the joint geometry of the surface and of the observer (and, to a lesser extent, of the illuminant). Third, for the type of images we used, texture classification performs much worse than color, and it

was interesting to see whether fusing these two features together would lead to a better or worse classification than using color alone.

We have used non–normalized (r,g,b) color features. Our experiments have shown that, when enough training data is available, normalized colors typically give worse performances than non–normalized ones. For texture, we used standard Gabor features [16]. Following [15], 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 outputs for each pixel, and smoothed each resulting image with a Gaussian kernel, with standard deviation proportional to the standard deviation of the envelope of the corresponding Gabor kernel. This provides us with a 8–dimensional feature vector for each pixel.

3.3. Classifier Design

Bayesian classification was performed using a Mixture–of–Gaussian (MoG) model for the conditional likelihoods. MoG color classification has been used successfully in computer vision [17]. Experimental results described in [15] show that MoG texture classification outperforms another popular approach based on the distance of local histograms for the same type of images used in this work.

The MoG representation for the conditional likelihood $p(f|y)$ is

$$p(f|y) = \sum_{k=1}^{N(y)} \alpha_k G(f; \mu_k, \Sigma_k) \qquad (12)$$

where $G(f; \mu_k, \sigma_k)$ is a Gaussian density with mean μ_k and covariance Σ_k, and α_k are positive constants such that $\sum_{k=1}^{N(y)} \alpha_k = 1$. In this work, all 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 conditional likelihood $p(f|y)$ is *not* separable in general. Let $f = (f_1, f_2)$ be the composite feature, where f_1 is the 3–dimensional color feature and f_2 is the 8–dimensional texture feature. If there are N_c classes, then the number of free parameters in the composite model $p(f)$ is $12 \sum_{y=1}^{N_c} N(y) - N_c$. For the color and texture models, the numbers of free parameters are $4 \sum_{y=1}^{N_c} N_1(y) - N_c$ and $9 \sum_{y=1}^{N_c} N_2(y) - N_c$ respectively (where $N_i(y)$ is the number of modes used to model $p(f_i|y)$). ML parameter estimation from the training data was carried out using the Expectation Maximization algorithm. To reduce the risk of hitting local minima, the procedure was initialized with values obtained after running k–means clustering 20 times on the same data (starting from random values). The k–means clustering that minimizes the mean distance between the cluster centers and the feature vectors in the clusters is used for EM initialization.

Road	$N(1)$	$N(2)$	$N(3)$
Color	3	3	3
Texture	4	4	4
Composite	5	4	4

Rock	$N(1)$	$N(2)$
Color	2	3
Texture	3	3
Composite	3	3

Table 1: The number of modes for the MoG models.

One important problem 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 with other algorithms such as BIC and *v*CV [2]. This algorithm is based on cross-validation, randomly dividing the data set into two disjoint train and test subsets, and, for each possible value of the number of clusters, using the training subset to estimate the parameters and using such parameters to compute the log likelihood over the test subset. This is repeated several times (in our work around 20 to 50 times). The number of modes is chosen by analyzing the log likelihood (averaged over the number of iterations) for the different model dimensions. In "ideal" situations (i.e., when the data is actually distributed according to a MoG model), the "correct" number of modes usually stands up as the maximizer of the averag log–likelihood. However, in practice, it is often the case that a clear maximum cannot be found, at least for a reasonably small range of possible dimensions. Rather than looking for maxima, we chose dimensions corresponding to obvious changes in the log–likelihood (e.g., such that, if choosing one dimension less, there would be a substantial decrease in log–likelihood). Note that, as noted by Smyth himself, the procedure should not be implemented as a "black box", but its results should be interpreted by the user. The selected number of modes are reported in Table 1.

The MoG model also provides a simple and convenient way to estimate the mutual information. The differential entropies in (6) can be computed by Monte Carlo integration using the explicit form of $p(f|y)$ in (12). In particular, to compute the marginal entropies $H_i(y)$, we marginalized the conditional likelihood $p(f|y)$, rather than using the likelihoods computed over the features independently. Indeed, if the data is not distributed according to our model, it may well happen that the densities computed for the individual features do not coincide with the marginals of the density of the composite feature. This may give problems when

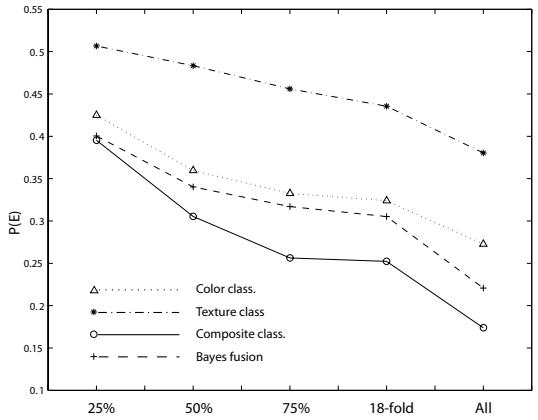

Figure 1: The misclassification rate $P(E)$ for all different experiments over the data set "Road". The generalized correlations for the three classes are: ρ(soil)=0.21; ρ(tall vegetation)=0.66; ρ(grass)=0.46.

combining together entropies in (6), such as yielding a negative value for the mutual information. These problems are avoided using the marginals from the same density $p(f|y)$.

Note that we did not enforce spatial coherence: each pixel is classified independently of the other pixels (although some degree of spatial correlation is implicit in the definition of texture feature). While spatial coherence is a powerful prior, 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.

3.4. Performance Metrics

Our classifiers were trained and tested over each data set separately. The probability of misclassification $P(E)$ 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 accumulated (summed) the confusion matrices. To account for possibly different number of pixels across labels, we normalized each row of the resulting matrix so that it sums to 1. Then, our estimate for $P(E)$ was the mean of the entries in the diagonal of the normalized confusion matrix.

In order to evaluate the sensitivity of the two algorithms (Bayesian classification and Bayes fusion of classifiers) to the size of the training set, we considered different 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 [3]) is known to be biased (optimistic). Still, it gives useful indications about the system

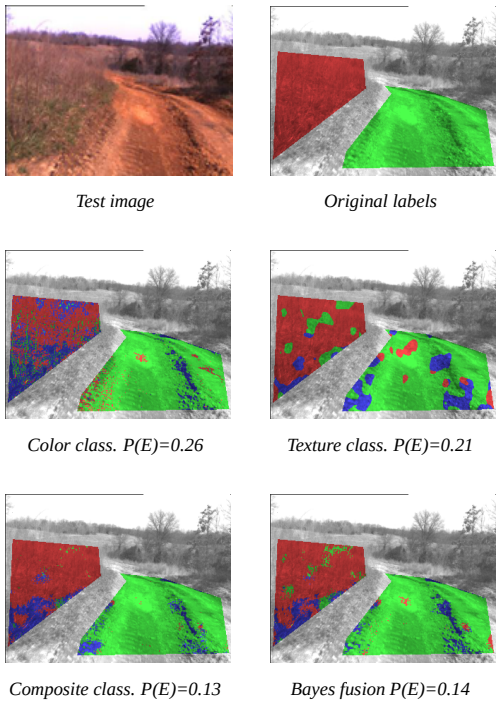

Figure 2: A classification example from the "Road" data set. The system was trained over 25% of the images in the set. This image was not part of the training set.

performance in the best–case scenario. We then used a K–fold cross–validation strategy [18]: for each data set with K images, we trained the system over K-1 images, and tested it on the remaining image. This leads to 18 tests for the "Road" data set and 11 tests for the "Rock" data set. In order to test over reduced training data sets, we used a variation of the bootstrapping method. We randomly selected (without replacement) $n\%$ (with n = 25, 50 and 75) training images out of each data set, and used the remaining ones for testing. We then repeated the procedure 20 times, averaging the results. Note that we randomly selected images from the image set rather than randomly select pixels from the whole training set; this seems like a more natural procedure in computer vision, where one typically can access a whole image rather than just isolated pixels. However, due to the high correlation among neighboring features, this method may lead to somewhat pessimistic results (the training samples for each label cannot really be considered independent).

The resulting values for $P(E)$ for the two data sets using color features, texture features, composite color–texture features, and Bayes fusion, are shown in Figures 1 and 4. Some classification examples (relative to one trial using 25% images in the data set for training) are shown in Figures 2, 3 and 5. In these figures, the colored areas represent the portion of the images that have been hand–labeled.

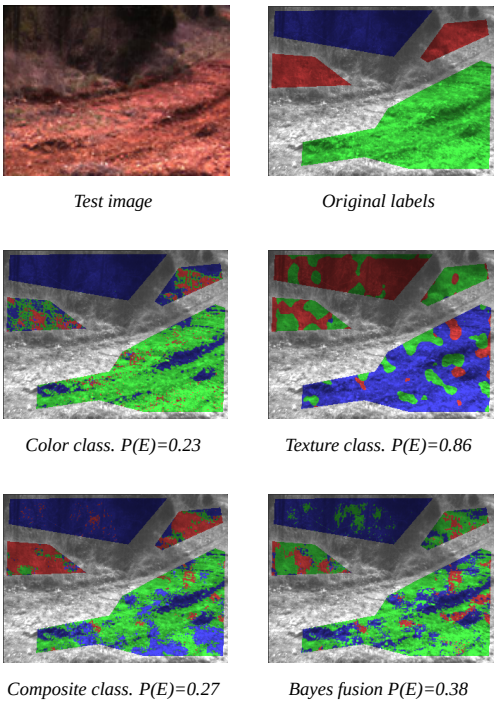

Figure 3: A classification example froom the "Road" data set. The system was trained over 25% of the images in the set. This image was not part of the training set.

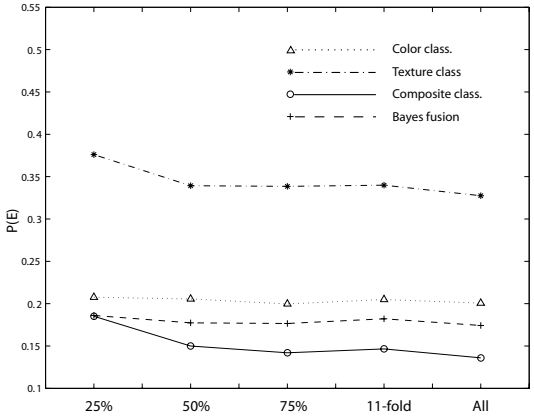

Figure 4: The misclassification rate $P(E)$ for all different experiments over the data set "Rock". The generalized correlations for the two classes are: ρ(sand)=0.06; ρ(rock)=0.15.

4. Discussion and Conclusions

It is interesting to compare the behavior of the error rate $P(E)$ in Figures 1 and 4. The error rate is higher for smaller training sets, which is to be expected given that smaller training sets correspond to higher estimation variance. It is clear that color features are much more reliable than texture features (note that for the "Rock" data set, 40% or more of the pixels get misclassified using texture). A reason for this poor performance is the relatively small numer of scales/orientations used here. In addition, these real–world images often prove very challenging for texture classification. In spite of such poor results when used alone, texture may help improve the already good performance of color. Note that this is true on average, and not necessarily on single images. For example, Figures 3 shows an extreme case with texture misclassifying most of the image ($P(E)$=0.86), in which case combining texture with color gives worse results than using color alone.

Comparing the classification using composite color–texture features and using Bayes fusion, one notes that, as expected, the former has lower error rate than the latter. However, the difference $\Delta P(E)$ between the two decreases when smaller training sets are used (and in fact the two strategies give the same error rate when the training data is formed by 25% of the images in the image set). This is a very interesting behavior, and suggests that, in spite of their large bias, naive Bayes system may generalize as well as "real" Bayesian classifiers for small training data size. Note that the number of free parameters is comparable in the two cases.

Finally, we can see that, for larger training set sizes, $\Delta P(E) \approx 0.05$ for the "Road" set, while $\Delta P(E) \approx 0.03$ for the "Rock" set. The values of the generalized correlation between color and texture for each class are (0.21, 0.66, 0.46) in the "Road" case, and (0.06, 0.15) in the "Rock" case. Thus, at least in our experiments, larger generalized correlations lead to higher $\Delta P(E)$, as conjectured in Section 2.1. Admittedly, more experiments over diverse image sets would be necessary to draw solid empirical evidence supporting our conjecture. Still, we believe that these results are encouraging, and they prompted us to pursue further ongoing research in this direction.

Acknowledgment

This work was supported by NSF under contract IIS–0222943. Bruno Sanso of the AMS Department at UCSC provided invaluable feedback during the initial phase of the work.

References

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

[2] P. Smyth, "Clustering using Monte Carlo cross-validation", *Proc. Knowledge Discovery and Data Mining*, 126–133 (1996)

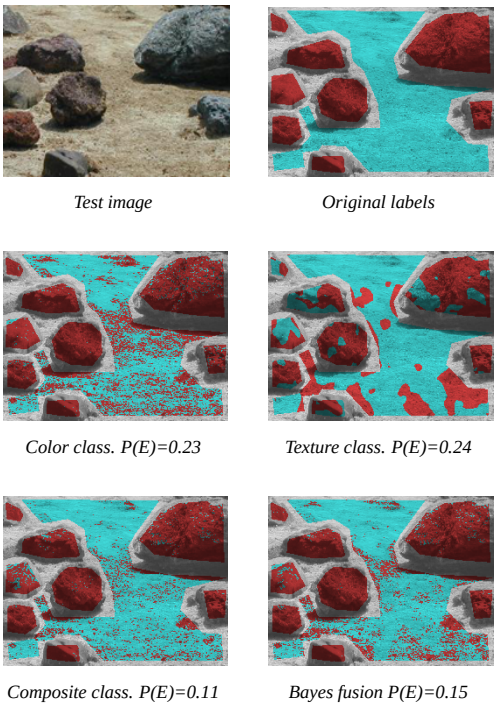

Figure 5: A classification example from the "Rock" data set. The system was trained over 25% of the images in the set. This image was not part of the training set.

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

[4] Pat Langley, "Induction of selective Bayesian classifiers", *Proc. of the 10th Conference on Uncertainty in Artificial Intelligence*, Seattle, WA (1994).

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

[6] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers", *Machine Learning* 29, November 1997.

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

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

[9] S.L. Geman, E. Bienenstock and R. Doursat, "Neural networks and bias/variance dilemma", *Neural Computation* 4: 1-58, 1992.

[10] J. Kittler, M. Hatef. R. Duin and J. Matas, "On combining classifiers", *IEEE Trans. PAMI* 20(3), March 1998.

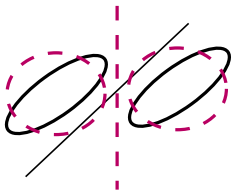

Figure 6: Equilevel curves of the pdf $p(f)$ considered in the Appendix (solid black line) and of its separable approximation $p_1(f_1)p_2(f_2)$ (dashed red line). The two straight lines represent the boundary of the assignment regions in the two cases.

[11] J. Kittler and A.A. Hojjatoleslami, "A weighted combination of classifiers employing shared and distinct representations", *Proc. CVPR*, 924–929 (1998).

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

[13] T. Heskes, "Bias/Variance decomposition for likelihood–based estimators", *Neural Computation* 10, 1425–1433 (1998).

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

[15] R. Castano, R. Manduchi, J. Fox, "Classification Experiments on Real-World Textures", *Workshop on Empirical Evaluation in Computer Vision*, Kauai, HI (2001).

[16] B.S. Manjunathand W. Y. Ma, "Texture features for browsing and retrieval of image data", *IEEE Trans. PAMI*, 18(8), pp. 837-842 (1996).

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

[18] J.K. Martin and D.S. Hirschberg, "Small sample statistics for classification error rates I: Error rate measurements", Dept. of Inf. and Comp. Sci., UC Irvine, Tech. Report 96–21 (1996).

[19] B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

Appendix

Assume that both conditional likelihoods $p(f|0)$ and $p(f|1)$ are Gaussian with the same covariance Σ but different means. Furthermore, assume that the prior probabilities for the classes are the same. It is well known [19] that the classification boundary in this case is a line, crossing the mid-point between the two mean vectors, and that $P_B(E)$ is only a function of the Mahalanobis distance between the two mean vectors:

$$P_B(E) = 1 - \Phi\left(0.5\,\sqrt{\Delta\mu'\Sigma^{-1}\Delta\mu}\right) \qquad (13)$$

where $\Phi(x)$ is the Gaussian cumulative distribution function: $\Phi(x) = (1/\sqrt{2\pi})\int_{-\infty}^{x}\exp(-y^2/2)$.

The naive Bayes classifier assumes that the two conditional likelihoods are separable, and that their conditional likelihoods are the outer products of the corresponding marginal densities, which are Gaussians with diagonal covariance. The classification boundary is thus still a line, but a different line than in the case of the Bayesian classifier. It is possible, after some tedious computation not reported here, to derive a form for the classification error rate $P_{NB}(E)$:

$$P_{NB}(E) = 1 - \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-\frac{{f_1}^2}{2}} \int_{f_1 - \frac{\sqrt{\Delta\mu'\Sigma^{-1}\Delta\mu}}{2\,\tan\theta}}^{\infty} e^{-\frac{{f_2}^2}{2}}\, df_1\, df_2 \qquad (14)$$

where

$$\cos\theta = \sqrt{1-\rho^2}\,\frac{1 + \frac{2\sigma_{12}\Delta\mu_1\Delta\mu_2}{\det(\Sigma)\Delta\mu'\Sigma^{-1}\Delta\mu}}{\sqrt{1 + \frac{4\sigma_{12}\Delta\mu_1\Delta\mu_2}{\det(\Sigma)\Delta\mu'\Sigma^{-1}\Delta\mu}}}$$

where $\Delta\mu_i$ is the i-th component of $\Delta\mu$, $\sigma_{12} = \mathrm{Cov}(f_1, f_2)$ is the off-diagonal element of Σ, and $\rho = \sigma_{12}/\sigma_1\sigma_2$ is the correlation between the two features.

While equation (14) may seem somewhat complex, one can make it more tractable as follows. First, one easily shows that $|\sin\theta| \leq \rho$. Second, the upper bound $\sin\theta = \rho$ is reached only when $\Delta\mu_1 = 0$ or $\Delta\mu_2 = 0$, that is, when the two Gaussians have means aligned along one cartesian axis. This is the situation that maximizes $P_{NB}(E)$ in (14), and thus can be used to compute an upper bound for $P_{NB}(E)$. Figure 7 shows the graph of the error rate $P(E)$ for the Bayes case (blue lines), and the upper bound for the error rate for the naive Bayesian case (red lines), as a function of the Mahalanobis distance $\sqrt{\Delta\mu'\Sigma^{-1}\Delta\mu}$ as well as of the correlation ρ. As expected, the error rate for the Bayesian case is independent of ρ.

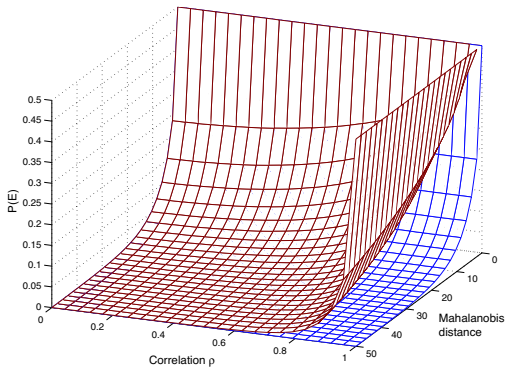

Figure 7: The misclassification rate $P(E)$ of the Bayesian classifier (blue) and of the naive Bayes classifier (red) for the case considered in the Appendix (see text).