Fusing Segmentation and Classification from Multiple Features

Roberto Manduchi

University of California, Santa Cruz
Santa Cruz, CA 95064

Abstract. This paper presents a strategy for combining the results of image classification and image segmentation. The visual features used for classification and segmentation may be different in general. Fusion is performed in a Maximum Likelihood framework using the Expectation Maximization algorithm. Preliminary results show that segmentation may effectively contribute to increase the quality of classification.

1 Introduction

In several computer vision problems, the analyst has access to different types of observables (let's call them "features") for the same image. These features often correspond to very different physical causes. For example, the color of a pixel depends on the combination of the reflectance of the surface, the spectrum of the illuminant, and the illumination geometry. The texture around a pixel depends on local albedo and/or geometrical variations. The optical flow at one point depends on the 3D motion of the imaged surface relative to the camera. All of these different observables should be combined to infer information about the scene.

Two fundamental low–level tasks of image analysis are segmentation/grouping and classification, as summarized below.

1. *Segmentation/grouping*: The aim is to identify perceptually homogeneous regions in the image. Such regions don't have pre-defined *labels*: they do not necessarily correspond to semantic categories known in advance. Segmentation is typically performed by first establishing a suitable similarity metric, sometimes derived by generative models. The segmentation problem can then be recast as an optimization task (energy or cost minimization [17], Maximum Likelihood or Maximum a posteriori [16]). This approach is inherently global, in that the decision at any pixel requires examination of all other points in the image.
2. *Classification (labeling)*: In this case, a number of classes is known in advance, together with their statistical description or at least with a number of labeled training samples. The goal is to assign each image point or region to just one class. Classification may use image features defined at the pixel level (such as

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 601–608, 2004.

color, motion field or texture[1]), or at a higher abstraction level (e.g., shape or spatial distribution). Note that even in the case of pixel–based classification, global reasoning is often invoked to enforce spatial coherence priors.

Whereas these two families of algorithms are traditionally used in different contexts, we show in this paper that segmentation, an unsupervised process that is blind to the semantic categories defined by the user, can actually be used to improve the classification process. This is particularly useful in two cases of practical importance. The first case is when the visual features used to segment the image cannot be used for classification. For example, gradient–based techniques such as snakes can effectively isolate an image region, but classification based on the boundary contour alone may be difficult. Another example is the use of optical flow to identify areas corresponding to different motion models. Once these regions have been segmented out, further reasoning, possibly using different features such as color, texture and shape, should be used for region labeling.

The second case of interest is when there is shortage of training data. Due to the "curse of dimensionality", high–dimensional features (e.g., texture) require much larger training data sets than low–dimensional features (e.g., color). Whereas learning a classifier may be unpractical in such cases, clustering only requires a notion of distance in feature space, which can be defined without reliance on training data.

Intuitively, segmentation should provide some sort of prior information to the classifier. If two image points belong to the same segment, it is reasonable to expect (although by no means necessary) that they also belong to the same class. A naive application of this intuition could lead to a procedure that assigns all points in the same segment to the class that is best represented in the segment. This "hard" fusion policy, however, would be unsatisfactory in general; a softer strategy that takes into consideration the degree of confidence in both classification and segmentation would be much more desirable. Indeed, our algorithm requires that the result of these two operation is expressed either in terms of a class– (or segment–) posterior distribution or, equivalently, in terms of class– (or segment–) conditional likelihoods. These type of information is normally available from the classifier. For what concerns the segmenter, some algorithms do produce soft cluster assignment (e.g., Expectation Maximization), while others produce hard (binary) assignments (e.g. k–means, graph cutting, snakes). It is always possible, though, to artificially "soften up" the result of hard segmenter, by creating at each point a distribution over the set of segments, peaking at the segment assigned to that point.

The intuition behind our approach is simple. The segmenter identifies areas that are homogeneous, according to one considered feature. We hypothesize that there is a correlation between these segments and the semantic classes that we actually interested in. The result of the classifier (based on a different feature)

[1] Strictly speaking, texture is an attribute of a region, not of a single pixel. However, one may define a texture field, by assigning to each pixel the texture of a small region centered in that pixel.

will be biased toward using any evidence of such correlation. The correlation between clusters and classes, however, is unknown, and must be estimated from the image being analyzed. This "chicken and egg" problem is solved using the elegant formalism of the Expectation Maximization algorithm.

The main hypothesis used by our approach is the conditional independence of the two features for each class/cluster combination. This is a generalization of class–conditional feature independence, often assumed in decision fusion. When this property is satisfied, it is well known that the posterior class distribution given the two observed features factorizes into the product of the two marginal posterior class distributions (divided by the class prior). Classifiers that compute the product of these two posterior distributions are often called "naive Bayes". Since the conditional assumption is at the core of our algorithm, we discuss its relevance and shortcomings in Section 2. Our iterative solution to the class/segment fusion is introduced in Section 3, where we also show an example of application. Section 4 has the conclusions.

2 Conditional Independence and Bayes Fusion

We introduce here the notation that will be used throughout this article. Let f_1 and f_2 be two different local feature vectors. For example, f_1 could be the (r,g,b) color at a pixel, and f_2 the texture descriptor at the same pixel. Consider a set of N classes $\{c_j\}$. The class–conditional likelihood of feature f_i given class c_j is represented by $p_i(f_i|c_j)$. P_j represents the prior probability of class c_j, while $P_i(c_j|f_i)$ is represents the class–posterior probability distribution for a given feature f_j. Bayes' rule can thus be expressed as $P_i(c_j|f_i) = p_i(f_i|c_j)P(c_j)/p(f_i)$, where $p_i(f_i)$ is the total likelihood of feature f_i. The mode of the posterior probability yields the Bayes classification at the chosen pixel.

The fusion problem arises when we have independent information about class assignment from the two features f_1 and f_2. We would like to infer $P_{1,2}(c_j|f_1, f_2)$ from $P_1(c_j, f_1)$ and $P_2(c_j, f_2)$ (or, equivalently, $p_{1,2}(f_1, f_2|c_j)$ from $p_1(f_1|c_j)$ and $p_2(f_2|c_j)$). Unfortunately, it is impossible, in general,to infer the joint density $p_{1,2}(f_1, f_2|c_j)$ from its marginals. A popular simplifying assumption is the class–conditional independence of the two features, that is:

$$p_{1,2}(f_1, f_2|c_j) = p_1(f_1|c_j)p_2(f_2|c_j) \tag{1}$$

for each choice of class c_j. This assumption is easily transformed into an equivalent condition on the posterior distributions:

$$P_{1,2}(c_j|f_1, f_2) = P_1(c_j|f_1)P_2(c_j|f_2)/P(c_j) \tag{2}$$

Equation (2) determines a *Bayes fusion* classifier, more commonly known as a *naive Bayes* system [11, 3]. These two terms will be liberally interchanged in this work.

How acceptable is the conditional independence assumption? Although it is a weaker condition than total independence, conditional independence can be

grossly violated in practice. The real question, however, is how much this (generally wrong) assumption affects the classification performances. For example, it would be interesting to compare the misclassification rate of the Bayes fusion classifier with the (necessarily lower) Bayes rate (that is, the misclassification rate of the Bayes classifier which uses the real joint posterior $P_{1,2}(c_j|f_1, f_2)$ [16]). An analytical expression for such quantities cannot be found in general, although Shi and Manduchi [18] computed an upper bound for the difference of these two misclassification rates as a function of the correlation between the two features in a simple equivariant Gaussian case.

It is well known that, in practical applications, Bayes fusion (or naive) classifiers perform rather well, despite the possible inaccuracy of the approximation in (1) [11, 3]. Experimental studies include [8, 9]. Friedman [4] justifies the sometimes surprisingly good results achieved by naive Bayes classifiers in light of the *bias/variance dilemma*. The bias/variance theory, first introduced by Geman for the regression problem [5], links the expected quadratic estimation error to the randomness in the choice of the training data set and the complexity of the algorithm. More precisely, the *bias* represents the difference between the estimates, averaged over all possible choices of training samples, and the optimal (in L_2 sense) estimate (i.e., the conditional expectation). The *variance* is the actual variance of estimation, again computed using the distribution over the training samples. In general, complex regression algorithms have low bias but high variance (i.e. they may overfit the data), while this behavior is reversed for simpler algorithms. Since the squared bias and the variance contribute as additive terms to the overall estimation error, it is seen that lower complexity algorithms may outperform more complex algorithms when only a limited amount of training data is available (see also [14, 15, 13]).

A similar situation occurs in the case of classification, although the definition of bias and variance is somewhat different here. Friedman [4] first showed that even in this case, variance with respect to the choice of training sample has an important role in the quality of the result (that is, the misclassification rate). Further work in the field includes [10, 1, 2, 19, 20]. In spite of their obvious bias (consequent to the approximation in (1)), naive Bayes systems are described by a "simple" posterior distribution, and it is reasonable to assume that they are less sensitive to the choice of the training data [4]. Shi and Manduchi [18] confirmed this hypothesis, by showing experimentally that the difference between the misclassification rate of a Bayes fusion classifier and the Bayes rate decreases as fewer and fewer data are used for training.

3 Bayes Fusion of Segmentation and Classification

In this section we tackle the main objective of this contribution, namely the fusion of a classifier with a segmenter. As we mentioned in the Introduction, we will assume that segmentation is expressed by either a posterior distribution $P_k(s_k|f_i)$ over the set of segments $\{s_k\}$, or a conditional likelihood $p_k(f_i|s_k)$. Let's assume that f_1 is the feature used for classification, and f_2 is the feature

used for segmentation. We would like to be able to use the segmentation using f_2 to assist the classification over the set of classes $\{c_j\}$. Formally, our problem an be formulated as follows:

Given $P_1(c_j|f_1)$ and $P_2(s_k|f_2)$, estimate $P_{1,2}(c_j|f_1, f_2)$.

We could also consider a parallel problem, but defined using the conditional likelihoods:

Given $p_1(f_1|c_j)$ and $p_2(f_2|s_k)$, estimate $P_{1,2}(c_j|f_1, f_2)$.

This formulation makes our fusion problem similar to the case of Section 2, with one important difference: now the two marginal posterior distributions are defined over different sets, $\{c_j\}$ and $\{s_k\}$, that are semantically different (and have different cardinality in general). In order to attempt a solution to this problem, we first extend the notion of conditional independence to the case of conditional likelihoods defined over the cartesian product of the features and the cartesian product of the class/segment sets:

$$p_{1,2}(f_1, f_2|c_j, s_k) = p_1(f_1|c_j)p_2(f_2|s_k) \tag{3}$$

The same cautionary disclaimer about the validity and consequences of the conditional independence approximation, discussed in Section 2, applies to this case as well. Given this assumption, we can use Bayes' rule to write the joint posterior distribution given the two features as follows:

$$P_{1,2}(c_j, s_k|f_1, f_2) = \frac{p_1(f_1|c_j)p_2(f_2|s_k)P_{1,2}(c_j, s_k)}{\sum_{\bar{j},\bar{k}} p_1(f_1|c_{\bar{j}})p_2(f_2|s_{\bar{k}})P_{1,2}(c_{\bar{j}}, s_{\bar{k}})} \tag{4}$$

The posterior distribution $P_{1,2}(c_j|f_1, f_2)$ can then be obtained by marginalizing $P_{1,2}(c_j, s_k|f_1, f_2)$ in (4):

$$P_{1,2}(c_j|f_1, f_2) = \sum_k P_{1,2}(c_j, s_k|f_1, f_2)$$

The only unknown quantity in (4) is the joint prior distribution $P_{1,2}(c_j, s_k)$. In fact, this distribution is the key to understanding our fusion strategy. One easily proves that if the priors are separable, that is, if $P_{1,2}(c_j, s_k) = P_1(c_j)P_2(s_k)$, then segmentation does not contribute to the fusion process. Indeed, in this case $P_{1,2}(c_j, s_k|f_1, f_2)$ factorizes into $P_1(c_j|f_1)P_2(s_k|f_2)$, and marginalization over s_j simply yields $P_1(c_j, s_k|f_1, f_2) = P_1(c_j|f_1)$.

The more interesting cases are when $P_{1,2}(c_j, s_k)$ is not separable, that is, when knowledge about which segment the point belongs to gives us some prior information about the class. Since we don't know $P_{1,2}(c_j, s_k)$, we should try to extract it from the data. We will first consider the case the conditional likelihoods $p_1(f_1|c_j)$ and $p_2(f_2|s_k)$ (and therefore $p_{1,2}(f_1, f_2|c_j, s_k)$ from (3)) are known, or that a reasonable assumption about their values can be made. We can then use

a Maximum Likelihood criterion, and search for the joint priors that maximize the likelihood of the data $p_{1,2}(f_1, f_2)$ according to our model, where

$$p_{1,2}(f_1, f_2) = \sum_{j,k} p_{1,2}(f_1, f_2|c_j, s_k) P_{1,2}(c_j, s_k)$$

A classic solution is given by the Expectation Maximization algorithm, based on the following iterations:

1. For each pixel x in the image, use the current values for $P_{1,2}(c_j, s_k)$ to estimate an updated posterior distribution $P_{1,2}(c_j, s_k|f_1(x), f_2(x))$ as by (4);
2. For each class c_j and segment s_k, average the posterior probabilities $P_{1,2}(c_j, s_k|f_1(x), f_2(x))$ over the image to obtain the updated prior distribution $P_{1,2}(c_j, s_k)$.

At each step, the total likelihood $p_{1,2}(f_1, f_2)$ increases or stays the same, and therefore this procedure is guaranteed to converge to a (possibly local) maximum.

If the conditional likelihood of one feature (or both) is not known, but the class– or segment–conditional probability is known, then some modifications are in order. This could be the case when a hard segmenter is artificially transformed into a soft segmenter by creating a simple posterior distribution at each pixel, as mentioned earlier in the Introduction. For example, for a pixel x with feature $f_2(x)$ that was assigned to segment s_k, we could hypothesize a posterior distribution[2]

$$\begin{array}{r} P_2(s_r|f_2) = 1 - \epsilon \ , \ r = k \\ \epsilon/(N_s - 1) \ , \ r \neq k \end{array}$$

where N_s is the number of segments, and ϵ is a (small) positive constant. By averaging the values of $P_2(s_k|f_2)$ over the image, one can estimate the prior $P_2(s_k)$. One could then set an artificial total likelihood $p_2(f_2)$ that is constant over all features in the image. At this point, the conditional likelihoods can be computed using Bayes' rule.

As an application example, consider the image of Figure 1. In this case, color was used for classification, while texture was used for segmentation. A poorly trained color classifier produced the unsatisfactory labeling of Figure 1 (b). The three classes are: obsidian (the blue-ish rock, c_1); basalt (the red rock, c_2); and sand (c_3.) Texture–based unsupervised segmentation into two regions s_1 and s_2 (using Gabor features) yielded the results shown in Figure 1 (c). Texture was unable to separate the two rocks, but did a good job at separating both rocks from the sand. The fused classification (into the original three classes) is shown in Figure 1 (d). It is seen that the quality of classification has improved through fusion, although a small region surrounding the blue rock has been misclassified. Table 1 shoes the joint prior distribution $P_{1,2}(c_j, s_k)$, which cannot be factorized into the product of the two marginal priors.

[2] This distribution may be inconsistent if two points with exactly the same feature f_2 belong to different segments. This has not proven to be a problem in practice.

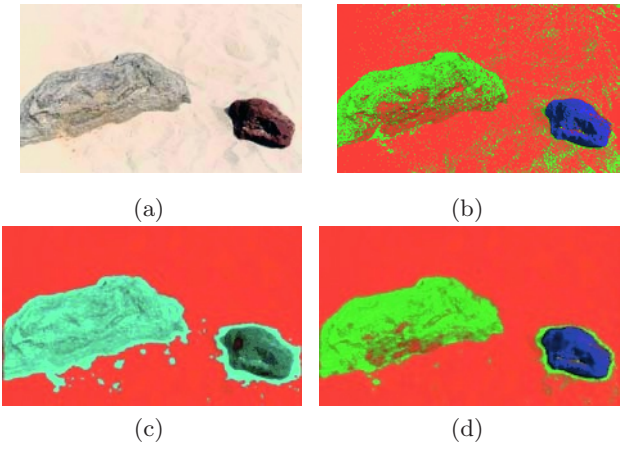

Fig. 1. (a): Original image. (b): Supervised color–based classification (green: obsidian; blue: basalt; red: sand.) (c) Unsupervised texture–based segmentation. (d) Fusion of segmentation and classification.

Table 1. The prior distribution $P_{1,2}(c_j, s_k)$ for the example of Figure 1. Note that $P_{1,2}(c_j, s_k)$ is not separable.

	s_1	s_2
c_1	0.120	0.262
c_2	0.001	0.045
c_3	0.564	0.008

4 Conclusions

Our fusion technique merges information from classification and segmentation (with each point of the image characterized by an assignment distribution.) In some sense, this corresponds to looking at the segmentation as a kind of classification itself, with classes that don't have a logical correspondence with those used by the classifier.

A more intriguing form of hybrid fusion would also consider cases where only partial segmentation is available, such as an edge segment partially separating two regions. This edge segment may provide useful information to the classifier (the regions at the two sides of the edge are likely to contain points from two different classes). Unless one enforces contour closure, however, this information cannot be directly exploit using the framework discussed in this paper, and more research is needed for this type of problems.

References

1. L. Breiman, "Bias, variance and arcing classifiers", Tech. Report 460, Statistics Department, UC Berkeley, 1996.
2. P. Domingos, "A unified Bias-Variance decomposition for zero-one and squared loss", *AAAI*/IAAI, 564–569, 2000.

3. 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.
4. J.H. Friedman, "On bias, variance, 0/1–loss, and the curse-of-dimensionality", *Data Mining and Knowledge Discovery*, 1, 55-77 (1997).
5. S.L. Geman, E. Bienenstock and R. Doursat, "Neural networks and bias/variance dilemma", *Neural Computation* 4: 1-58, 1992.
6. T. Heskes, "Bias/Variance decomposition for likelihood–based estimators",*Neural Computation* 10, 1425–1433 (1998).
7. N.R. Howe and D.P Huttenlocher, "Integrating Color, Texture, and Geometry for Image Retrieval", *Proc. Computer Vision and Patter Recognition*, 2000, 239–247.
8. J. Kittler, M. Hatef. R. Duin and J. Matas, "On combining classifiers", *IEEE Trans. PAMI* 20(3), March 1998.
9. J. Kittler and A.A. Hojjatoleslami, "A weighted combination of classifiers employing shared and distinct representations", *Proc. CVPR*, 924–929 (1998).
10. E.B. Kong and T.G. Dietterich, "Error–correcting output coding corrects bias and variance", *Proc. Intl. Conf. Machine Learning*, 313–21, 1995.
11. Pat Langley, "Induction of selective Bayesian classifiers", *Proc. of the 10th Conference on Uncertainty in Artificial Intelligence*, Seattle, WA (1994).
12. R. Manduchi, "Bayesian Fusion of Color and Texture Segmentations", *7th IEEE International Conference on Computer Vision*, Kerkyra, September 1999, 956–962.
13. 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).
14. S. Raudys and A.K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners", *IEEE Trans. PAMI*, 13(3):252–64, 1991.
15. S. Raudys, "On dimensionality, sample size, and classification error of nonparametric linear classification aglgorithms", *IEEE Trans. PAMI*, 19(6):337–71, 1997
16. B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
17. J. Shi, J. Malik, "Normalized Cuts and Image Segmentation", *Proc. Computer Vision and Patter Recognition*, 1997, 731–737.
18. X. Shi and R. Manduchi, "A Study on Bayes Feature Fusion for Image Classification", *IEEE Workshop on Statistical Algorithms for COmputer Vision*, 2003.
19. R. Tibshirani, "Bias, variance and predicition error for classification rules", Tech. Report, Dept. pf Prev. Medicine and Biosatistics, Univ, of Toronto, 1996.
20. D. Wolpert, "On bias plus variance", *Neural Computation* 9, 1211–1243 (1997).