Automatic Image Classification Using Pictorial Features

R. Schettini°, A. Valsasna°, C. Brambilla*, M. De Ponti§
°ITIM, *IAMI, Consiglio Nazionale delle Ricerche,
Via Ampere 56, 20131 Milano, Italy
§STMicroelectronics TPA Group, Printer Division,
Via Olivetti 2, 20041 Agrate Brianza, Italy

Abstract

The effective classification of image contents allows us to adopt those strategies that can best satisfy the increasing demand for quality, speed and ease of use in imaging applications. We present here the results of our experimentation using Cart trees for the classification of images indexed by low-level pictorial features, such as color, texture, and shape. Our study addressed the high-level problem of distinguishing photographs, graphics and texts for an application in the context of cross-media color reproduction. The results obtained to date are very good in terms of accuracy, and also demonstrate the strength of the approach in providing information that can be used to reduce the dimensions of the feature space.

Introduction

Content-based image classification has emerged as an important area in multimedia computing due to the rapid development of digital imaging, storage, and networking technologies.[1] The effective classification of image contents allows us to adopt those strategies that can meet the increasing demand for quality, speed and ease of use in imaging applications. We report here on our experience in the use of Cart trees for the classification of images indexed by low-level perceptual features such as color, texture, and shape, addressed to the high-level problem of distinguishing among photographs, graphics and texts.

The tree approach to classification provides a clear characterization of the conditions that determine when a case belongs to a certain class. We took this approach because we believe that, in problems such as ours in which the data structure presents a high level of complexity (large dimensions, mixture of data types and non-homogeneity), a good understanding of the predictive structure of the data is as important a criterion for good classification as accuracy.

Cart Classifiers

Cart classifiers are tree classifiers structured according to the Cart approach. The basic reference for this is the text by Breiman et al.,[2] which has had a seminal influence both in bringing tree methodology to the attention of the scientific community, and in stimulating the development of new strategies and algorithms. More concise descriptions can be found in [3,4], while an evaluation of the procedure's performance on several databases, together with a comparison with different approaches, can be found in [5]. Many references to the great variety of applications in which Cart classifiers are currently used are given in [6].

Generally speaking, Cart classifiers are trees constructed by recursively partitioning the predictor space, each split being formed by conditions related to the predictor values. The process is binary: the predictor space and each subset of it, are split exactly in two. In tree terminology the subsets are called nodes: the predictor space is the root node, terminal subsets are terminal nodes, and so on. The construction process is based on training sets of cases whose class $j\in\{1, \ldots, J\}$ is known. In our problem the predictors are the features indexing the images (the features used are listed in the following section), and the training sets are composed of images whose semantic class is known. Once a tree has been constructed, a class is assigned to each of the terminal nodes , and it is this that makes the tree a classifier: when a new case is processed by the tree, its predicted class is the class associated with to the terminal node into which the case finally moves on the basis of its predictor values.

The class assigned to each terminal node t is the one that minimizes the estimated expected misclassification cost within the node, which is given by

$$r(t) = \min_i \sum_j c(i \mid j) p(j \mid t), \qquad (1)$$

where $c(i|j)$ is the cost of misclassifying a class j case as a class i case, and $p(j|t)$ is the estimated probability of the class j in node t.

The performance of a tree is evaluated in terms of its overall misclassification probability, or misclassification cost, which, if T denotes the tree, is estimated by

$$R(T)=\sum_{t\text{ terminal node}} r(t)\cdot p(t), \qquad (2)$$

where *p(t)* is the estimated probability of a case being assigned to node *t*.

The critical problems of the splitting process are essentially two: how to identify candidate splits, and how to define the goodness of the splits. Candidate splits are generated by a set of admissible questions regarding the values of the predictors. These questions differ according to the nature of the predictors themselves. In the case of a category predictor, for example, all splits that assign the values of the predictor to two different groups are considered candidates. At each step of the process, all the predictors are searched one by one, and the best split, in the sense defined below, is found for each predictor. These best splits are then compared, and the best is again selected.

The idea central to the goodness of splits is that of selecting the splits so that the data in the descendant nodes are purer than the data in the original ones. To do so, different functions of impurity of the nodes, *i(t),* are introduced, and the decrease in value of the chosen function produced by a split is taken as a measure of the goodness of the split itself. For a node *t* and its descendant nodes t_l and t_r, this is

$$\Delta i(s,t)=i(t)-p_l i(t_l)-p_r i(t_r), \qquad (3)$$

where p_l and p_r are the proportion of the cases of *t* falling in t_l and t_r respectively, according to the split *s*.

The most commonly used function of node impurity is the Gini diversity index

$$i(t)=\sum_{i\neq j} p(i\,|\,t)p(j\,|\,t)=1-\sum_{j} p^2(j\,|\,t), \qquad (4)$$

which can be interpreted in terms of variances of Bernoulli variables. If, for each class *j,* we consider the random variable Y_j , which is 1(success) if a case of *t* belongs to class *j* and 0 (failure) otherwise, it can be modeled as a Bernoulli variable with probability of success *p(j|t),* and in this case the quantity

$$1-\sum_{j} p^2(j\,|\,t) \qquad (5)$$

is the sum of the variances of such variables.

The goodness of a split can also be evaluated by the reduction in deviance[7] produced by the split. For a node *t*, the deviance is defined as

$$D(t)=-2\sum_{j} n_{tj}\log p(j\,|\,t), \qquad (6)$$

where n_{tj} is the frequency of class j cases in node *t.* The underlying idea is that n_{tj} cases of the training set belonging to a node *t* constitute a random sample from the multinomial distribution specified by *p(j|t). D(t)* is proportional to the entropy function of the variable class within the node. Generally speaking the deviance is a function which quantifies the discrepancy of a fit from the data.[8]

Since the process goes on until some stopping rule is satisfied, the trees can be very big and overfit the data. One of the major innovations of Cart methodology is the possibility of performing a pruning process based on the idea of finding a trade-off between the complexity and the accuracy of the trees. For a tree *T,* the pruning process generates a sequence $\{T_l\}_{l\in\{1,\ldots,L\}}$ of subtrees decreasing in size, each of which is the best, in its size range, according to a cost-complexity measure defined as

$$R_\alpha(T)=R(T)+\alpha\,|\,T\,|, \qquad (7)$$

where |*T*| is the number of terminal nodes, and $\alpha(\geq 0)$ is a unit cost of complexity per terminal node. The subtrees are evaluated in terms of their overall misclassification probability, or misclassification cost, on the basis of test sets, or by means of cross-validation and the best subtree is then selected. Choosing the tree to be used for classification in this way reduces the strong dependence of the classification itself on the training data, and provides a more parsimonious classifier.

A very useful feature of Cart methodology, which helps to identify masking effects among the predictors and makes it easy to deal with missing data, are surrogate splits. Broadly speaking, a surrogate split is the split that most accurately matches the action of another split. If *s* is the split of a node t, and s_m is any split of the same node based on the m-th predictor, the surrogate split of s, indicated as s_m^*, is defined as

$$s_m^* = \arg\big(\max_{s_m} p(s,s_m)\big), \qquad (8)$$

where $p(s,s_m)$ is the estimated probability that s_m predicts *s* correctly. This is given by

$$p(s,s_m)=p_{ll}(s,s_m)+p_{rr}(s,s_m), \qquad (9)$$

where $p_{ll}(s, s_m)$ is the estimated probability that both *s* and s_m assign a case of *t* to the left descendant node t_l and $p_{rr}(s, s_m)$ is the probability that both *s* and s_m send a case of *t* to the right descendant node t_r.

Not all surrogate splits are useful; their goodness can be evaluated by the association function

$$\lambda(s,s_m^*)=\frac{\min(p_l,p_r)-\big(1-p(s,s_m^*)\big)}{\min(p_l,p_r)} \qquad (10)$$

which measures the relative reduction in error obtained by using s_m^* to predict the split *s* as compared with the $\max(p_l,p_r)$ rule prediction. If λ is negative, s_m^* is of no help in predicting *s* and is therefore disregarded. The splits which are good surrogates of the best splits in terms of the association function, are predictors which could be used advantageously in the classification problem studied, even though they may not appear in the tree structure.

The overall importance of the m-th predictor in the whole classification problem is measured by the function

$$M_m = \sum_{t\ node} \Delta I\left(s_m^*, t\right), \qquad (11)$$

where $\Delta I(s,t)=I(t)- I(t_l)- I(t_r)$, with $I(t)=i(t)p(t)$.

When the problem is missing data, surrogate splits are useful both when values are missing in the training set cases, and in the new cases to be classified. In the latter situation, if a case is missing a value of a predictor used to derive an optimal split s, it can still proceed along the tree according to the surrogate split of s that, measured by function λ, best matches s.

To conclude, a further advantage of trees classifiers is their robustness with regard to outliers, which are usually isolated in small nodes.

Image Description Using Pictorial Features

The features we used to index the images constitute a general purpose library of low-level pictorial features which can be calculated on the global image and/or on sub-images obtained by dividing the original image in different ways.

These features are:

- the color histogram in the hue-saturation-value color space (HSV) quantized in 64 colors;
- the Color Coherence Vectors (CCV) in the HSV color space quantized in 64 colors[9];
- the spatial-chromatic histogram and the histogram of the transitions of the color regions identified by the process of quantization in 11 colors (red, orange, yellow, green, blue, purple, pink, brown, black, gray and white)[10,11];
- the moments of inertia, i.e. the mean, variance, skewness and kurtosis, of the distribution of the colors in terms of hue, saturation and luminance[12];
- the percentage of "colored" and of "not-colored" pixels of the image;
- the statistical information on image edges extracted by Canny's algorithm: i) the percentages of low, medium, and high contrast edge pixels in the image; ii) the parametric thresholds on the gradient strength corresponding to medium and high contrast edges; iii) the number of connected regions identified by closed high contrast contours; iv) the percentage of medium contrast edge pixels connected to high contrast edges; v) the histogram of edge directions extracted by Canny's edge detector[13];
- the mean and variance of the absolute values of the coefficients of the sub-images of the first three levels of the multi-resolution Daubechies wavelet transform of the luminance image[14];
- the estimate of texture features based on the Neighborhood GrayTone Difference Matrix (NGTDM), i.e. coarseness, contrast, busyness, complexity, and strength[15,16];
- the spatial composition of the color regions identified by a process of quantization in 11 colors: i) fragmentation (the number of color regions); ii) distribution of the color regions with respect to the center of the image; iii) distribution of the color regions with respect to the x axis, and with respect to the y axis[10];
- a skin region detector trained on a large amount of labeled skin data, e.g.[17].

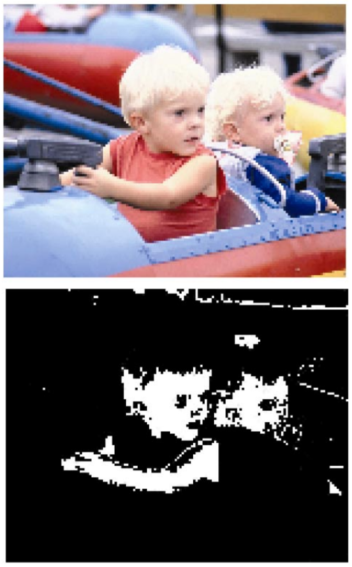

Figure 1. Skin regions detection

The total number of features is rather high, since the histograms used have large dimensions. However the widely differing natures of the indices limit the risk of having different images correspond to very close points in the feature space. Moreover, while all the features must be computed for the images in the training sets, only the features actually used by the classifier need to be computed for images in the test sets, and for new images processed by the classifier.

Results

The problem we addressed is that of distinguishing among photographs, graphics and texts.[18] The training set used to date consists of about 4000 photographs, 7000 graphic works, and 1500 texts. The images differ in size (ranging from 150x150 pixels to 1500x1500 pixels), resolution, and tonal depth. The photograph class contains photographs of indoor and outdoor scenes, landscapes, people, animals, and objects. The graphics class contains clipart, business graphics, and photo-realistic graphics. The text class contains digitalized handwritten texts, as well as scanned or computer generated texts in colour and in black and white, in various fonts.

Our analysis was always based on the assumptions of equal prior probability for each class and equal costs of misclassification among classes. Of course, the misclassification of photographs might, for example, be more costly than the misclassification of texts or graphics, depending upon the application concerned. To measure the impurity of the nodes we used the Gini index. To estimate the overall probability of misclassification of the trees derived from the pruning process, we used both cross-validation and test sets. Since, when the costs of the different misclassifications are considered equal, the overall probability of misclassification of a tree is equivalent to the overall cost of misclassification, we refer here only to the misclassification probability.

We first performed a straightforward 3-class classification.[19] The results were very good in the classification of photographs and texts, and not quite as good for graphics: averaging the different trials, which were performed by changing the test sets and the subdivisions of the training set required by the cross-validation, the estimate of the overall probability of misclassification obtained was about 5% for photographs, 10% for graphics, and 7% for texts. It is interesting to note that there was always one terminal node that incorporated almost 85% of the photographs of the training set, and another that incorporated about 85% of the texts of the training set. This means that most of the photographs and the texts can be characterized by the paths reaching those nodes. Even more interesting is the fact these paths involve no more than 1% of the all features. The graphic images were, instead, spread out among several terminal nodes.

Looking in detail at the misclassified images, we found that many of the photographs misclassified as graphics were photographs of objects on a large and uniformly colored background, typical of business graphics such as pies, histograms. The photographs misclassified as texts included, among others, photographs of playing cards. Most of the graphics misclassified as photographs were illustrations. The texts misclassified as graphics were those with only a few colored words in large font. For further verification we processed the photographs of a large database (about 26000): this gave the same level of misclassification, and confirmed the typology of the photographs misclassified.

We decided to experiment the strategy of performing a hierarchical 2-class classification as well. More specifically, we first classified a new image as either text, or photograph/graphics; then, if the image was not classified as text, it was subsequently classified in the photograph class or in the graphic class (of course, we could have followed a different order, the choice depending, once again, on the application involved). This strategy also gave very good results. On the average, the estimate of the overall probability of misclassification of the 2-class tree classifiers constructed for the first step of classification was about 5% for the text class, and 3% for the joint class of photographs and graphics. The estimate of the overall probability of misclassification of the 2-class tree classifiers constructed for the second step was about 5% for the photograph class and 7% for the graphic class. As before, we found most of the texts of the training set in a single node, and the photographs in another, both of which can be reached with paths involving very few features. When we processed the same 26000 photograph database, we obtained a level of misclassification of 0.6% in the first step (i.e. 0.6% of the photographs were misclassified as texts), and of 5% in the second step.

We must now decide which of the two approaches is to be preferred in practice. In the hierarchical approach the classifier used in the second step is constructed to handle a more specialized task, and this seems to be an advantage. On the other hand, there is the disadvantage that an image misclassified in the first step, is excluded from subsequent comparison. This drawback can be limited by taking into account the estimate of the probability of misclassification attached to each terminal node of the classifier. For some of the terminal nodes with few images, this estimate is in fact quite high (in some cases even more than 20%), and it is obvious that conclusions drawn from these nodes are much less reliable. We are considering also admitting to the second step of the classification process those images which would at present be excluded if, in the first step, they fell into a node with an estimate of probability of misclassification larger than a given threshold. It might be advantageous to consider the estimate of the misclassification probability associated with each terminal node in direct 3-class classification too, even if the classifier performs on the whole very well.

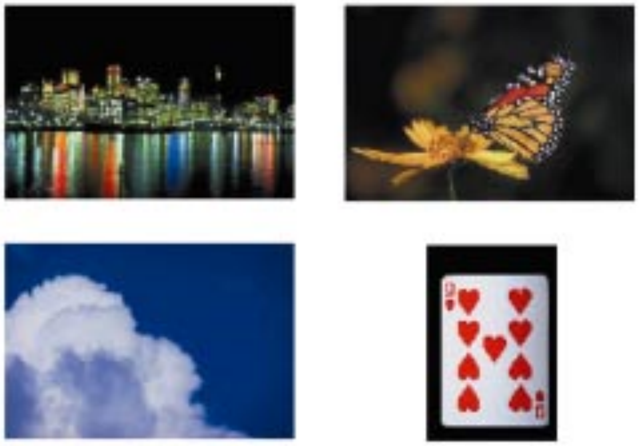

Figure 2. Some misclassified photographs

Future Work

We are currently constructing a very large database of graphics and texts in order to reach the same level of confidence in our classification model and results as those obtained for photographs. To reduce misclassification of photographs we plan to index sub-images, in particular the central part of the images. To increase the stability of the trees we plan to use bagging, or some other P&C (perturbing and combining) method.[20] These methods generate multiple versions of the predictors by perturbing the training sets, and then combine these multiple versions into a more stable single predictor. Finally we intend to deal with the fact that the classifier can not reject images that do not belong to any of the predefined classes.

Acknowledgement

Our investigation was performed as a part of a ST Microelectronics' research contract. The authors thank ST Microelectronics for the permission to present this paper.

References

1. O. Aigrain, H. Zhang, D. Petkovic, Content-based representation and retrieval of visual media: a state of the art review, Multimedia Tools and Applications, 3, pg. 179 (1996).
2. L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth and Brooks/Cole, 1984.
3. B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, 1996.
4. D.J. Hand, Construction and Assessment of Classification Rules, Wiley and Sons, 1997.
5. D. Michie, D.J. Spiegelhalter, C.C. Taylor (eds.), Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.
6. D. Steinberg, P. Colla, CART, Salford Systems, 1997.
7. J. M. Chambers, T.J. Hastie (eds.), Statistical Models in S, Chapman and Hall, 1992.
8. P. McCullagh, J.A. Nelder, Generalized Linear Models, Chapman and Hall, 1989.
9. G. Pass, R. Zabih and J. Miller, Comparing Images Using Color Coherence Vectors, Paper presented at the Fourth ACM Multimedia 96 Conference (1996).
10. P. Ciocca, R. Schettini, A relevance feedback mechanism for content-based retrieval, Information Processing and Management, **35**, 605-632 (1999).
11. Gagliardi, R. Schettini, A method for the automatic indexing of color images for effective image retrieval. The New Review of Hypermedia and Multimedia, **3**, 201-224 (1997).
12. M.A Stricker, M. Orengo, Similarity of Color Images. SPIE Storage and Retrieval for Image and Video Databases III Conference, (1995)
13. J. Canny, A computational approach to edge detection, IEEE Trans. On Pattern Analysis and Machine Intelligence, IEEE-**8,** 679-698 (1986).
14. F. Idris, S. Panchanathan, Storage and retrieval of compressed images using wavelet vector quantization, Journal of Visual Languages and Computing, **8**, 289-301 (1997).
15. M. Amadasun, R. King, Textural features corresponding to textural properties, IEEE Transaction on System, Man and Cybernetics, **19**(5), 1264-1274 (1989).
16. H. Tamura, S. Mori, T. Yamawaki, Textural features corresponding to visual perception, IEEE Transaction on System, Man and Cybernetics, **8**, 460-473 (1978).
17. Y. Miyake, H. Saitoh, H. Yaguchi, and N. TsukadaFacial pattern detection and color correction from television picture for newspaper printing, Journal of Imaging Technology, **16**, 165-169 (1990).
18. R. Schettini, C. Brambilla, G. Ciocca, M. De Ponti, Color image classification using tree classifiers, Proc. VII Color Imaging Conference: Scottsdale (Arizona), 269-272 (1999).
19. R. Shettini, C. Brambilla, A. Valsasna, M. De Ponti, Content-based image classification, Proc. Internet Imaging Conference, Proceedings of SPIE 3964 (G.B. Beretta, R. Schettini eds.), 28-33 (2000).
20. L. Breiman, Bagging predictors, Machine learning, **26,** 123-140 (1996).

Biography

Raimondo Schettini has been associated with the Italian National Research Council (CNR) since 1987. With 1994 he moved to the Institute of Multimedia Information Technologies, where he is currently in charge of the Image and color Analysis Lab. He has published more than ninety refereed papers on image processing, analysis and reproduction, and on image content-based indexing and retrieval. Since 1997 he also teaches a course on multimedia design at the School of Industrial Design of the Polytechnic of Milan. He is member of the CIE TC 8/3 and general co-chair of the Internet Imaging Conference.