A Bayesian classifier for symbol recognition

Sabine Barrat, Salvatore Tabbone, Patrick Nourrissier

A Bayesian classifier for symbol recognition

S. Barrat[1], S. Tabbone[1] and P. Nourrissier[2]

[1] LORIA-UMR 7503, University of Nancy 2,
BP 239, 54506 Vandoeuvre-lès-Nancy, France
{barrat,tabbone}@loria.fr
[2] Netlor Concept, 14 boulevard du 21ème Régiment Aviation, 54000 Nancy, France
patrick@netlor.fr

Abstract. We present in this paper an original adaptation of Bayesian networks to symbol recognition problem. More precisely, a descriptor combination method, which enables to improve significantly the recognition rate compared to the recognition rates obtained by each descriptor, is presented. In this perspective, we use a simple Bayesian classifier, called naive Bayes. In fact, probabilistic graphical models, more specifically Bayesian networks, are a simple and intuitive way of probability distribution representation. In order to solve the dimensionality problem, we use a variable selection method. Experimental results, obtained in a supervised learning context and tested on GREC symbol database, are very promising.

Keywords: Symbol recognition, data analysis, probabilistic graphical models, Bayesian networks, variable selection.

1 Introduction

Classification is a basic task in data analysis and pattern recognition. This task requires a classifier, i.e a function that assigns a class label to instances described by a set of features. The induction of classifiers from data sets of labelled data (we speak about supervised learning) is a central problem in machine learning. In fact, in numerous applications, the aim is to assign a feature vector $f = \{f_1, f_2, ..., f_n\}$ to a class c_i among k classes, designed by the vector $c = \{c_1, c_2, ..., c_k\}$. Some approaches to this problem are based on various functional representations such as decision trees, neural networks, decision graphs [1], associated to decision rules. Probabilistic approaches also play a central role in classification. A way to reach the previous goal, by using probabilities, is to compute the conditional probability distribution $P(c_i|f)$, $\forall i \in \{1, 2, ..., k\}$ and assign the instance f to the class c_i for which this probability is maximal. We could formulate and solve complicated probabilistic models purely by algebric manipulations. However, we shall find it highly advantageous to improve the analysis using diagrammatic representations of probability distributions, called probabilistic graphical models. In fact they provide a simple way to visualize the structure and the properties (including conditional independence properties) of a probabilistic model, but they especially enable to perform complex computations like inference and learning, using graphical manipulations.

Thus we propose, in this paper, an original method of descriptor combination applied to symbol classification. We have adapted probabilistic graphical model theory to symbol recognition problem. The originality of our approach also relies on the use of the variable selection method LASSO [2], proposed to overcome the dimensionality problem related to the size of feature vectors and the inherent network complexity.

2 Bayesian networks as classifiers

2.1 Definitions and notations

We remind that our aim is to assign a feature vector $f = \{f_1, f_2, ..., f_n\}$ to a class c_i among k classes. Recent works in supervised learning [3–5] have shown that the naive Bayesian classifier has a surprisingly good performance. This classifier learns from training data (labelled data) the conditional probability of each feature F_j, $\forall j \in \{1, 2, ...n, \}$ given the class label c_i, $\forall i \in \{1, 2, ..., k\}$. Classification is then done by applying Bayes rule to compute the probability of c_i given the particular instance of $F_1, F_2, ..., F_n$, $\forall i \in \{1, 2, ..., k\}$ and then predicting the class with the highest posterior probability. This computation is rendered feasible by making a strong independence assumption: all the features F_j are conditionally independent given the value of the class C. Thus we obtain this formula:

$$P(c_i|f_1, f_2, ..., f_n) = \frac{P(f_1, f_2, ..., f_n, c_i)}{P(f_1, f_2, ..., f_n)} = \frac{P(f_1, f_2, ..., f_n|c_i) \times P(c_i)}{P(f_1, f_2, ..., f_n)}$$

where

$$P(f_1, f_2, ..., f_n) = \sum_{i=1}^{k} P(f_1, f_2, ..., f_n, c_i) = \sum_{i=1}^{k} P(f_1, f_2, ..., f_n|c_i) \times P(c_i)$$

In order to represent and manipulate independence assertions, we need an appropriate language and efficient machinery. Both are provided by Bayesian networks [6]. These networks are directed acyclic graphs that provide an efficient and effective representation of the joint probability distribution over a set of random variables. Each vertex in the graph represents a random variable, and edges represent dependence relations between the variables. More precisely, the network encodes the following conditional independence statement: each variable is independent of its nondescendants in the graph given the state of its parents. These independencies are then exploited to reduce the number of parameters needed to characterize a probability distribution, and to efficiently compute posterior probabilities given an evidence. If we represent a naive Bayesian classifier as a Bayesian network, it has the simple structure depicted in Fig. 1.

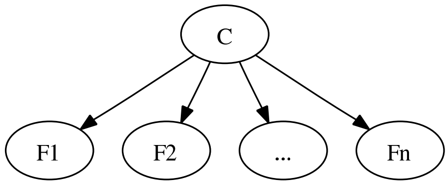

Fig. 1. Naive Bayes

This network encodes the main assumption behind the naive Bayesian classifier, namely, that each feature (each leaf in the graph) is independent from the rest of the features, given the state of the class variable (the root of the graph).

2.2 Parameter learning

The parameters of our model, i.e the probabilities $P(c_i|f_1, f_2, ..., f_n)$ and $P(c_i)$, $\forall i \in \{1, 2, ..., k\}$, are obtained by Maximum Likelihood Parameter Estimation method, because it is one of the most robust parameter estimation techniques, from a statistical point of view. The idea of this method is to obtain the most likely values of the parameters, for a given distribution, that will best describe the data. But when some instances are not observed in the training set, this method will evaluate the corresponding parameters by null values. If the resulted network is used to infer on these instances, the network will predict that these instances have a null probability. Thus this method can generate overfitting. To overcome this problem, we introduce additional pseudo counts at every instance in order to ensure that they are all 'virtually' represented in the training set. It doesn't mean that we have to construct a new training set, but that we will use Dirichlet priors. Using maximum likelihood estimator, with Dirichlet priors, implies that every instance, even if it is not represented in the training set, will have a not null probability. This method is explained more precisely in [7].

2.3 Inference

A query image, f_j, that we want to classify, characterized by its features $f_{j_1}, f_{j_2}, ..., f_{j_n}$, can be considered as an evidence which would be represented by:

$$P(f_j) = P(f_{j_1}, f_{j_2}, ..., f_{j_n}) = 1$$

when the network will be evaluated. Then the inference process consists in computing posterior probability distributions of one or several other subsets of nodes. In the case of classification, we infer the class node. According to our Bayesian network topology, the inference process propagates the values from the image feature level, represented by the n nodes $F_1, F_2, ..., F_n$, to the class node C. Currently we count several methods of exact or approximate inference techniques. Although the exact inference is generally a NP-hard problem [8], it is still efficient for some classes of Bayesian networks. Moreover the complexity in time of exact inference methods is early computable. When the result exceeds a reasonable threshold, we will prefer to use an approximate method [9, 10]. In our problem, the size of the network is not large, because we reduce the feature dimensionality with a variable selection method (see subsection 3.2), so we can use an exact inference method. One of the most robust methods of exact inference is the JLO algorithm, called junction tree algorithm too, developped in [11, 12]. This algorithm transforms the network into a junction tree: the variables are merged into clusters in order to delete all loops in the network and to make the clusters as small as possible. After converting the network to a tree, a message passing algorithm [13] is applied to the network. In this technique, each node is associated to a processor, which can send some messages to its neighbours, in an asynchronous way, until it reachs a stability. Then the probabilities of each node are updated in function of the evidence instance. After the belief propagation, we know the posterior probability $P(c_i|f_{j_1}, f_{j_2}, ..., f_{j_n}), \forall i \in \{1, 2, ..., k\}$. The query f_j is assigned to the class c_i which maximizes this probability.

3 Symbol recognition

This section explains how we have adapted the theoretical method before-mentioned to the symbol recognition problem. We present the used features and the method we have had to use in order to overcome large dimensionality problem.

3.1 Features

We have to classify black and white images, so we have choosen these three shape descriptors: Generic Fourier Descriptor (GFD), Zernike descriptor and the $\mathcal{R}$-signature $1D$. We briefly present each descriptor below.

Generic Fourier Descriptor Generic Fourier Descriptor is based on Fourier transform [14]. It is a pixel descriptor. The rotation invariance is achieved by using the modified polar Fourier transform (MPFT) proposed by Dengsheng Zhang and Guojun Lu in [14]. Then scaling invariance is achieved after normalization.

Zernike descriptor Zernike descriptor [15] is a pixel descriptor based on Zernike moments. Zernike moments of a given shape are calculated as correlation values of the shape with Zernike basis functions, in that all the pixels of the shape, independently of their position, contribute with the same weight to the Zernike moments. These moments are rotation invariant. To make the Zernike moments of the shape descriptor invariant also to translation and scaling, a given shape is scaling and translation normalized, by obtaining the smallest circle centered at the center of mass, covering all the shape pixels. Then the obtained circle is adjusted to match the radius of Zernike moment basis functions. The Zernike shape descriptor consists of low order magnitudes of Zernike moments.

$\mathcal{R}$-signature $1D$ The $\mathcal{R}$-signature $1D$ [16] uses Radon transform to represent an image. The Radon transform is the projection of an image in a particular plan. This projection has interesting properties which make it a good descriptor. According to these geometrical properties, a $1D$ signature of the transform is created. This signature checks the properties of invariance to some geometrical transformations, such as the translation and the scaling (after normalization). The rotation invariance is achieved by a cyclic permutation of the signature, or directly from its Fourier transform.

3.2 Dimensionality reduction

Once the n, $n \in \{1, 2, 3\}$ descriptors we want to combine are computed on each image, we dispose of n signatures per image. The concatenation of these signatures provides us a new feature vector per image. Each vector component is considered as a discrete random variable and corresponds to a node from the feature level of the network. The large dimensions of the initial signatures imply a large dimension for the feature vector and too many variables in the network compared to the size of training set: in fact existing algorithms on Bayesian networks are efficient until around 1000 variables, only if we have quite a few training data. To overcome this problem, we have to use a dimensionality reduction method, which enables to extract from the feature vectors, just the most relevant and discriminating features, with a minimal information loss. We have choosen the regression method LASSO for its stability and implementation facility, but especially because this method enables to select variables (when some regression coefficients are null) and is competitive in complex situations, contrary to the Principal Component Analysis (PCA) or the traditional methods like subset selection or ridge regression, for example.

Least Absolute Shrinkage and Selection Operator: LASSO The LASSO is an approach of variable selection and coefficient shrinkage introduced by R. Tibshirani [2]. This method shrinks the regression coefficients by imposing a penalty on their size. The LASSO coefficients minimize a penalized residual sum of squares:

$$\beta^{lasso} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

subject to

$$\sum_{j=1}^{p} |\beta_j| \leq s \tag{1}$$

The LASSO uses a $L1$ penalty: $\sum_{j=1}^{p} |\beta_j|$. This constraint implies that for smaller values of s ($s \geq 0$), some of the coefficients β will be null. So choosing s is like choosing the number of predictors in a regression model. Then the selection criterion is simple: we will select the variables corresponding to the coefficients different from zero.

The computation of the lasso solutions is a quadratic programming problem, and can be tackled by standard numerical analysis algorithms. One of the most adapted is the Least Angle Regression (LAR) procedure. In fact it exploits the special structure of the lasso problem, and provides an efficient way to compute the solutions simultaneously for all values of s. This algorithm is presented in detail in [17]. We have applied this method to our training data: in this case y represents our class variable C, and $x_j = \{x_{j_1}, x_{j_2}, ..., x_{j_p}\}$ the p features of the instance j.

3.3 From continuous to discrete data

The descriptors before-mentioned provide us signatures of non integer values. So, once we have selected variables, we dispose of a reduced set of continuous variables. But the naive Bayesian classifier requires discrete variables. So it is necessary to discretize our variables. We have used the method presented in [18], because it enables an important data reduction, by passing from continuous to discrete data, without any loss of information. Moreover it is easy to implement. Concretely this method consists in approximating probability laws by histograms. These histograms must optimally approximate, in the sense of the maximum likelihood and a mean squares cost, the unknown law of a random process with a single n-sample. In order to obtain the bin number defining the histogram and the distribution of these bins, the information criterion of Akaike (AIC), habitually used for model order selection, has been generalized to this problem. More precisely, an m-bins histogram is built from the sample. Then, the idea to reduce the histogram size is to merge the two adjacent bins, for which the difference between the AIC before merging and after merging is maximal. This process is repeated until this difference would be negative.

We apply this method for all variables on the training set. We dispose of a n-sample per variable and the method provides us one histogram per variable. Then considerate one histogram: each bin corresponds to a possible discrete value for the variable corresponding to this histogram, and we can obtain the probability of each possible value thanks to the number of elements in each bin. Thus we can estimate the probability distribution for each variable.

4 Experimental results

We have used the symbols of GREC database [19] to test our method. We have selected a subset of 50 models and generated a database of 3600 symbols, composed of 72 different images per model. Each image corresponds to a model, with some degradations and linear transformations (rotation and scaling) of different intensities.

On this database we have defined some learning tests: we have defined various training and test sets of different sizes. We have tested the method by using the signatures issued from one, two or three descriptors and executing all possible combinations. The tables below present the recognition rates obtained by the different tests. The variable selection method LASSO has enabled us to select 13 variables among 225 initial variables for GFD, 15 among 34 for Zernike and 13 among 180 for the $\mathcal{R}$-signature. Concerning the combinations, we have obtained 4 variables issued from GFD and 6 issued from Zernike for the combination of GFD and Zernike, 7 variables issued from GFD and 7 from the $\mathcal{R}$-signature for the combination of Zernike and the $\mathcal{R}$-signature, and 6 from Zernike and 6 form the $\mathcal{R}$-signature for the combination of Zernike and the $\mathcal{R}$-signature. Finally, for the combination of the 3 descriptors, we have 2 variables from GFD and the $\mathcal{R}$-signature and 3 from Zernike. The table 1 shows the means of the recognition rates obtained with some random selections of the same numbers of variables than the ones obtained with LASSO. The table 2 presents the results obtained after the automatic selection, with LASSO method, of a subset of relevant variables. In the both tables, G means that the GFD descriptor is used, Z that the Zernike descriptor is used and R for the $\mathcal{R}$-signature. Finally the '+' operator indicates that we combine the descriptors.

The experimental results show the interest of our method: the LASSO method enables us to choose a suitable number of variables for each descriptor, for our Bayesian classifier. Moreover, in the great majority of the examples, selecting the variables with the LASSO is more efficient than a random selection of the same number of variables. We also observe that, after applying the LASSO method, whatever the recognition rates obtained by one descriptor, we improve these rates by combining 2 or 3 descriptors. In the same way, whatever the recognition rates obtained by combining 2 descriptors, we improve these recognition rates by combining 3 descriptors.

Besides, even if we obtain a high recognition rate with Zernike descriptor, this rate will not decrease when we combine this descriptor with one or two other descriptors, whatever these descriptors and even if the added descriptors have a low rate (it is the case with the $\mathcal{R}$-signature), i.e the bad behaviour of a descriptor doesn't impede the other descriptor behaviours.

Specifications		G	Z	R	G+Z	G+R	Z+R	G+Z+R
training	test							
ims 1-54	ims 55-72	87.2	90.9	70.4	92.4	89.7	93.3	94.7
ims 19-72	ims 1-18	81.8	86.4	61.2	85.3	85.1	89.8	90.1
ims 1-36	ims 37-72	83	88.5	63	86.9	86.5	92	90.9
ims 37-72	ims 1-36	81.2	89.2	63.2	86.4	86.3	92.5	90.8
ims 1-18	ims 19-72	80.6	83.8	54.2	84.4	83.2	90.9	88.2
ims 55-72	ims 1-54	64.3	79.4	49.5	76.8	70.9	85.2	82.7

Table 1. Recognition rates without automatic variable selection

Specifications		G	Z	R	G+Z	G+R	Z+R	G+Z+R
training	test							
ims 1-54	ims 55-72	88.2	92.3	60.2	93.8	94.7	95.7	97.1
ims 19-72	ims 1-18	83.3	88.2	50	89.2	87.8	91.3	94
ims 1-36	ims 37-72	83.8	89.8	50.8	90	86.7	92.9	95.4
ims 37-72	ims 1-36	81	90.5	48.5	91.4	86	93.2	93.8
ims 1-18	ims 19-72	75.8	86.9	44.2	89.3	81.2	92.2	90.6
ims 55-72	ims 1-54	66.2	79.1	36.9	81.3	73.4	84.6	85.4

Table 2. Recognition rates with the variable selection method LASSO

5 Conclusion and future work

In this paper we have proposed an original adaptation of Bayesian theory. Moreover, the LASSO method has been proposed to solve the dimensionality problem of feature vectors. The experimental results are promising and show that our method enables to improve the recognition rate by combining descriptors.

In our future work, we want to integrate a relevance feedback approach. More precisely, it is a matter of taking into account some informations given by the user, on some queries, in order to modify the network parameters after inferred the class of a query image. Besides we want to use the information given by keywords associated to a subset of training data. It's possible because Bayesian networks enable to manage, in a same network, different kinds of information (in this case different media), and thanks to their ability to manage incomplete data.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
2. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) **58** (1996) 267–288
3. Domingos, P., Pazzani, M.J.: Beyond independence: Conditions for the optimality of the simple bayesian classifier. In: International Conference on Machine Learning. (1996) 105–112
4. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29**(2-3) (1997) 131–163
5. Friedman, J.H.: On bias, variance, 0/1loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery **1**(1) (1997) 55–77
6. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)
7. Robert, C.: A decision-Theoretic Motivation. Springer-Verlag (1997)
8. Cooper, G.: The computational complexity of probabilistic inference using bayesian belief networks. Artificial Intelligent **42**(2-3) (1990) 393–405
9. Jaakkola, T., Jordan, M.I.: Variational probabilistic inference and the QMR-DT network. Journal of Artificial Intelligence Research **10** (1999) 291–322
10. Jordan, M.I., Ghahramani, Z., Jaakkola, T., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning **37**(2) (1999) 183–233

11. F.Jensen, Lauritzen, S., Olesen, K.: Bayesian updating in recursive graphical models by local computations. Computational Statistical Quarterly **4** (1990) 269–282
12. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Readings in uncertain reasoning (1990) 415–448
13. Kim, J., Pearl, J.: A computational model for combined causal and diagnostic reasoning in inference systems. In: IJCAI-83. (1983) 190–193
14. Zhang, D., Lu, G.: Shape-based image retrieval using General Fourier Descriptor. Signal Processing : Image Communication **17** (2002)
15. Kim, H.K., Kim, J.D., Sim, D.G., Oh, D.I.: A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval. IEEE International Conference On Multimedia And Expo **1** (2000) 307–310
16. Tabbone, S., Wendling, L.: Technical Symbols Recognition Using the Two-dimensional Radon Transform. In: ICPR'02. Volume 2. (2002) 200–203
17. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression (2004)
18. Colot, O., Olivier, C., Courtellemont, P., El-Matouat, A., de Brucq, D.: Information criteria and abrupt changes in probability laws. Signal Processing VII : Theories and Applications **1** (2004) 387–391
19. Lladós, J., Kwon, Y.B., eds.: Graphics Recognition, Recent Advances and Perspectives, 5th InternationalWorkshop, GREC 2003, Barcelona, Spain, July 30-31, 2003, Revised Selected Papers. In Lladós, J., Kwon, Y.B., eds.: GREC. Volume 3088 of Lecture Notes in Computer Science., Springer (2004)