A Bayesian network for combining descriptors: application to symbol recognition

Sabine Barrat, Salvatore Tabbone

IJDAR
DOI 10.1007/s10032-009-0103-y

FULL PAPER

A Bayesian network for combining descriptors: application to symbol recognition

Sabine Barrat · Salvatore Tabbone

Received: 25 March 2009 / Revised: 23 September 2009 / Accepted: 3 November 2009

Abstract In this paper, we propose a descriptor combination method, which enables to improve significantly the recognition rate compared to the recognition rates obtained by each descriptor. This approach is based on a probabilistic graphical model. This model also enables to handle both discrete and continuous-valued variables. In fact, in order to improve the recognition rate, we have combined two kinds of features: discrete features (corresponding to shape measures) and continuous features (corresponding to shape descriptors). In order to solve the dimensionality problem due to the large dimension of visual features, we have adapted a variable selection method. Experimental results, obtained in a supervised learning context, on noisy and occluded symbols, show the feasibility of the approach.

Keywords Symbol recognition · Descriptor combination · Variable selection · Probabilistic graphical models · Bayesian networks

1 Introduction

Pattern recognition applications have to face the problem of describing a large number of different objects for recognition. A recognition system should be robust to variability (geometric transformations, noise, occlusions, ...) and to scalability, when a large number of classes and images should be recognized. Symbol recognition is a field within pattern recognition for which a lot of efforts have already been made [22,37,39,44]. Symbol recognition is usually decomposed into two steps: symbol description and classification [1,6,12]. In order to describe symbols, a lot of different shape descriptors have been proposed (see surveys [15,33,40]) but one descriptor is usually not enough to describe all kinds of shapes properly and therefore to give satisfactory shape recognition rates. One solution is to combine several descriptors in a classification task [38] or to use several classifiers and to combine their outputs [32,34]. 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 training sets (sets of labeled data) is a central problem in machine learning: it is a problem of supervised learning. In fact, in numerous applications, the aim is to assign a feature vector $f = \{f_1, f_2, \ldots, f_n\}$ to a class c_i among k classes, designed by a vector $c = \{c_1, c_2, \ldots, c_k\}$. Some approaches to this problem are based on various functional representations such as decision trees, neural networks, decision graphs [2,23,29,42], associated with decision rules.

Probabilistic approaches also play a central role in classification [21,43,24]. A way to reach the previous goal, by using probabilities, is to compute the conditional probability distribution $P(c_i|f), \forall i \in \{1, 2, \ldots, k\}$ and assign the instance f to the class c_i for which this probability is maximal. In order to represent probability distributions over a large set of variables, we introduce several conditional independence assumptions that will help to reduce the complexity of the model and provide a tractable model. Within the framework of the graphical models [16], a class of models called Bayesian networks allows an efficient representation of any probability distribution that can be factorized according to a set of independence assumptions. This factorization will help to reduce the computational complexity of the model. Moreover, this framework comes with many algorithms for performing inference (i.e., the computation of posteriors)

S. Barrat (✉) · S. Tabbone
LORIA-UMR 7503, University of Nancy 2,
BP 239, 54506 Vandoeuvre-lès-Nancy, France
e-mail: barrat@loria.fr

S. Tabbone
e-mail: tabbone@loria.fr

Published online: 26 November 2009

and learning (factorization of parameters fitting, computation of probability distributions …). We propose, in this paper, an original method of descriptor combination applied to symbol classification. Thus, we have adapted the probabilistic graphical model theory to the symbol recognition problem. In this model, continuous and discrete variables are combined. Continuous variables correspond to shape descriptors, and the discrete ones correspond to shape measures. Thanks to this combination, the proposed classifier is more robust to deformations and to the size of database, when the number of symbols increases. The originality of our approach also relies on the use of a variable selection method [35], to overcome the dimensionality problem related to the size of feature vectors and the inherent network complexity.

The organization of the paper is as follows. In Sect. 2, the main properties of a Bayesian network-based classifier are introduced and lead to the presentation of our probabilistic model for symbol recognition. The visual features used to represent the symbols are described in Sect. 3. The feature selection algorithm which allows us to increase the recognition rate by focusing only on the main features while reducing the dimensionality problem is also explained in Sect. 4. Our method is evaluated on a database of noisy and occluded symbols (Sect. 5). Finally, Sect. 6 brings conclusions and opens new perspectives to our work.

2 Representation and classification of images

2.1 Context and objectives

Our work is focused on symbol recognition by combining descriptors. Given an image database, where each image contains one symbol, we try to recognize the "perfect" symbol (the model) represented in each image. In fact, the symbols contained in the images are not perfect: they can be noisy, deformed and can have occlusions. This recognition problem can be view as a classification problem: our aim is to assign each image to the class corresponding to the perfect symbol (the model) of this image. However, no perfect symbol is available. Therefore, we cannot just minimize a distance between each image of the database and each perfect symbol. On the other hand, this classification task can be resolved by using a supervised learning method, from a subset of the database where the class label (the perfect symbol) is known for each image.

Moreover, in order to describe all kinds of shapes properly, even deformed or noisy shapes, and thus increase the recognition rate, our proposition is to combine several shape descriptors. Now, shape descriptors can provide vectors of continuous or discrete values:

let f_j be a query image characterized by a set of features F composed of:

- m continuous visual features, denoted $v_1, \ldots, v_m$,
- n discrete visual features, denoted $DF\ 1, \ldots, DF\ n$.

The chosen visual features are issued from 3 shape descriptors and 3 shape measures. Shape descriptors provide vectors of continuous values, and each shape measure provides a single discrete value.

Consequently, it seems appropriate to propose a classifier that enables to manage both discrete and continuous features. Although most classification methods handle only discrete data and thus require a pre-processing step of discretization in order to transform each continuous-valued variable into a discrete one, few classification methods can handle both discrete and continuous-valued variables. It is the case of Support Vector Machines [4], Random Forests [3], and Bayesian classifiers [13]. Support Vector Machines (SVM) and Random Forests (RF) are well known for their ability to handle high-dimensional data. On the contrary, Bayesian classifiers are sensitive to the dimensionality of the data, but they often perform well in many domains. Therefore, we have chosen to construct a Bayesian classifier for its ability to combine discrete and continuous-valued variables. Moreover, we show that this Bayesian classifier, associated with a variable selection method, is competitive with SVM, even on high-dimensional data.

2.2 Bayesian classifiers

Let I be a new image designed by a particular instance $f = \{f_1, \ldots, f_n\}$ of the feature vector $F = \{F_1, \ldots, F_n\}$. Our aim is to assign I to a class c_i among k classes. Each c_i is a particular instance of the variable C. The Naïve Bayes (NB) is a simple probabilistic classification algorithm that often performs well in many domains. This classifier encodes a distribution $P_{NB}(F_1, \ldots, F_n, C)$, from a given training set (composed of labeled data). The resulting probabilistic model can be used to classify the new instance I. In fact, the Bayes rule is applied to compute the probability of c_i given the particular instance f. Then the classifier based on NB returns the label $c_i, i \in \{1, \ldots, k\}$, that maximizes the posterior probability $P_i = P_{NB}(c_i | f_1, \ldots, f_n)$, where:

$$P_i = \frac{P_{NB}(f_1, \ldots, f_n | c_i) \times P_{NB}(c_i)}{P_{NB}(f_1, \ldots, f_n)}$$

and $P_{NB}(f_1, \ldots, f_n) = \sum_{j=1}^{k} P_{NB}(f_1, \ldots, f_n | c_j) \times P_{NB}(c_j)$.

However, we are interested in the probability distributions of discrete and continuous features and their conditional dependence relations. Let us consider each component of continuous vectors (issued from shape descriptors) as a continuous random variable and the discrete values (provided by shape measures) as discrete variables. This model is too big to be represented as a unique joint probability distribu-

tion. Therefore, it is required to introduce some sparse and structural *a priori* knowledge: the Naïve Bayes has to be extended to take into account continuous and discrete variables. In this perspective, the probabilistic graphical models, and especially Bayesian networks, are a good way to solve this kind of problem. In fact, within Bayesian networks, the joint probability distribution is replaced by a sparse representation only among the variables directly influencing one another. Interactions among indirectly related variables are then computed by propagating inference through a graph of these direct connections. Consequently, Bayesian networks are a simple way to represent a joint probability distribution over a set of random variables, to visualize the conditional properties and to compute complex operations like probability learning and inference, according to graph-based computations.

2.3 Bayesian networks

2.3.1 Definitions

Formally, a Bayesian network for a set of random variables V (continuous or/and discrete) is a pair $B = \langle G, \Theta \rangle$. The first component, G, is a directed acyclic graph whose vertices correspond to random variables $V_1, \ldots, V_n$, and whose edges represent direct dependencies between variables. The graph G encodes independence assumptions: each variable V_i is independent of its non-descendants given its parents in G. The second component of the pair, Θ, represents the set of parameters that quantifies the network. It contains a parameter $\theta_{v_i|Pa(v_i)} = P_B(v_i|Pa(v_i))$ for each possible value v_i of V_i, and $Pa(v_i)$ of $Pa(V_i)$, where $Pa(V_i)$ denotes the set of parents of V_i in G. That is, the Bayesian network, in its initial state, contains the initial *a priori* probabilities of each node of the network: $P_B(v_i|Pa(v_i))$. Thanks to the conditional independence assumption of each variable given its parents, the joint probability distribution $P_B(V_1, \ldots, V_n)$ can be reduced to this formula:

$$P_B(V_1, \ldots, V_n) = \prod_{i=1}^{n} P_B(V_i|Pa(V_i)) = \prod_{i=1}^{n} \theta_{v_i|Pa(v_i)}$$

The framework of Bayesian networks comes with many algorithms for performing inference (i.e., the computation of posteriors probabilities) and learning (factorization of parameters fitting, computation of probability distributions,…). The algorithms we used in this work are briefly described below.

2.3.2 Parameter learning

Only one has a description of a model, knowing the structure of the graph and probabilistic forms for each variables, one wants to estimate the numerical values of each parameter. Let assume we have either discrete or continuous variables (or a mix of them), and, for the simple case, a set of data describing many possible cases for each variables. The data set can either be complete or have missing data. In each case, a different solution will be used. In the case, where the data set has no missing values, an approach is to consider the parameters having the highest probabilities to generate the most similar data set if the Bayesian network was used to draw random values according to the probability distribution it describes (hence the name "generative model"). This method is known as the Maximum Likelihood. Let call $\mathcal{D}$ the data set, then $P(d|M)$ is the probability of a data $d \in \mathcal{D}$ to be generated by the model M and is called the likelihood of M given d. Therefore, the likelihood of M given the full data set $\mathcal{D}$ is:

$$L(M|\mathcal{D}) = P(\mathcal{D}|M) = \prod_{d \in \mathbb{D}} P(d|M)$$

For the sake of computational simplicity (or to help deriving an analytic form), the log-likelihood is often used:

$$L(M|\mathcal{D}) = \sum_{d \in \mathcal{D}} \log_2 P(d|M)$$

Therefore, the principle of maximum likelihood prefers to choose parameters with the highest likelihood:

$$\hat{\theta} = \text{argmax}_\theta L(M_\theta|\mathcal{D})$$

In general, a maximum likelihood is obtained by counting the frequencies over the total count. Whenever you do not have enough data for each case (missing data) one of the most popular algorithm is the Expectation-Maximization (EM) algorithm. The general purpose of this algorithm is explained in detail in [7]. During the Expectation phase, the data set is locally completed, then a Maximization step is performed to find the current maximum likelihood estimate (as seen above) using the completed data. In a Bayesian network, the first step of the EM algorithm can be easily done using a map *a posteriori* algorithm, i.e., computing the most probable values of the missing data variables given other known variables. The second step is then executed and can either be done with an optimization algorithm if no analytical form of the maximum likelihood is known, or with the previous approach. These two steps are repeated until convergence. The algorithm is initialized with random probability distribution parameters.

The EM algorithm is also used to learn the parameters of Gaussian distributions, which are considered as missing data.

2.3.3 Inference

An inference algorithm is necessary to compute the posterior probability distributions of unobserved nodes. According to

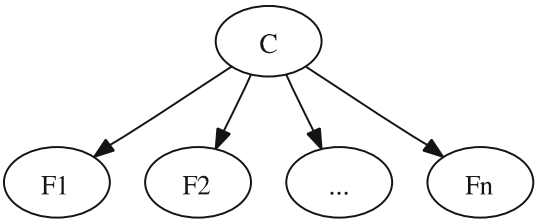

Fig. 1 Naïve Bayes

the Bayesian network topology, the inference process propagates the values from the leaf level to the inferred node. Many algorithms can be used [17]. The most popular is the message passing algorithm [20]. In this technique, each node is associated with a processor, which can send some messages to its neighbors, in an asynchronous way, until it reaches stability.

2.3.4 Bayesian network classifiers

Bayesian networks can be used as classifiers. For example, the Naïve Bayes can be represented by the structure in Fig. 1, where:

- C refers to the class variable,
- $F_1, \ldots, F_n$ are the feature variables.

The Naïve Bayes is a simple and efficient model, but it requires discrete variables. Since, we have to manage continuous (values provided by shape descriptors) and discrete (values provided by shape measures) variables, this model has to be extended to take into account continuous and discrete variables.

2.4 A Gaussian-Mixtures and Bernoulli Mixture model

A Bayesian network classifier, which handles both discrete and continuous-valued variables, is proposed. We present a hierarchical probabilistic model, the Gaussian-Mixtures and Bernoulli Mixture model, in order to classify large databases of symbols. In fact, the observation of some peaks on the different histograms of the vector components provided by shape descriptors has led us to consider that the continuous visual features can be estimated by mixtures of Gaussian densities. The discrete variables have a Bernoulli distribution. In fact, these variables can take two values: 1 if the corresponding shape measure provides a value smaller than 0.5, or else 2. Finally, the proposed model is inspired by the Naïve Bayes. Indeed, the class variable is connected to each other.

Now let F be the training set composed of m instances $f_j = \{f_{j_1}, \ldots, f_{j_n}\}, \forall j \in \{1, \ldots, m\}, \forall i \in \{1, \ldots, n\}$, where n is the dimension of the signatures provided by the concatenation of the feature vectors issued from the computation of all the descriptors for each image on the training set. Each instance $f_j, \forall j \in \{1, \ldots, m\}$ is then characterized by n continuous variables. A supervised classification is considered; then, F instances are divided into k classes $c_1, \ldots, c_k$. Let $G_1, \ldots, G_g$ be g groups whose each has a Gaussian density with a mean $\mu_l, \forall l \in \{1, \ldots, g\}$ and a covariance matrix $\sum_l$. Besides, let $\pi_1, \ldots, \pi_g$ be the proportions of the different groups, $\theta_l = (\mu_l, \sum_l)$ be the parameter of each Gaussian and $\Phi = (\pi_1, \ldots, \pi_g, \theta_1, \ldots, \theta_g)$ the global mixture parameter. The probability density of F conditionally to the class $c_i, \forall i \in \{1, \ldots, k\}$ can be defined by

$$P_i(f, \Phi) = \sum_{l=1}^{g} \pi_l \, p(f, \theta_l)$$

where $p(f, \theta_l)$ is the multivariate Gaussian defined by the parameter θ_l.

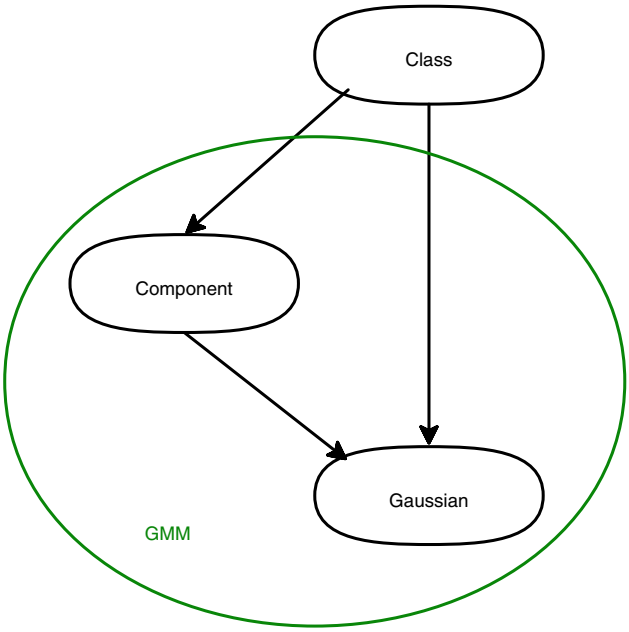

Fig. 2 A Probabilistic graphical model as GMMs

We have one Gaussian Mixture model per class, which can be represented by the probabilistic graphical model in Fig. 2, where:

- The "Class" node is a discrete node, which can take k values corresponding to the pre-defined classes $c_1, \ldots, c_k$.
- The "Component" node is a discrete node which corresponds to the components (i.e., the groups $G_1, \ldots, G_g$) of the mixtures. This variable can take g values, i.e., the number of Gaussians used to compute the mixtures. It is an hidden variable which represents the weight of each group (i.e., the $\pi_l, \forall l \in \{1, \ldots, g\}$).
- The "Gaussian" node is a continuous variable which represents each Gaussian $G_l, \forall \in \{l = 1, \ldots, g\}$ with its own parameter $(\theta_l = (\mu_l, \sum_l))$. It corresponds to the set of feature vectors in each class.

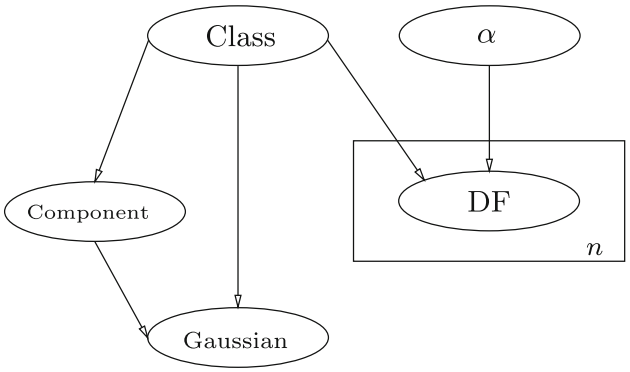

Fig. 3 The Gaussian-mixtures and Bernoulli mixture model

– Finally, the edges represent the effect of the class on each Gaussian parameter and its associated weight. The green circle does not belong to the graphical model: it is just a way to show the relation between the proposed probabilistic graphical model and GMMs: we have one GMM (encircled in green), composed of Gaussians and their associated weight, per class.

The model can be completed by the discrete variables, denoted $DF\ 1, \ldots, DF\ n$, where n is the number of shape measures, and DF_i represents the value of each shape measure. Dirichlet priors [27] have been used for the probability estimation of the variables $DF\ 1, \ldots, DF\ n$. That is we introduce additional pseudo counts at every instance in order to ensure that they are all "virtually" represented in the training set. Therefore every instance, even if it is not represented in the training set, will have a not null probability. Like the continuous variables, the discrete variables corresponding to the discrete measures are included in the graphical model by connecting them to the class variable.

Now our classifier can be depicted by the Fig. 3. The hidden variable "α" shows that a Dirichlet prior is used. The box around the variable DF denotes n repetitions of DF for each shape measure.

This Bayesian classifier means that continuous and shape features, representing images, are assumed to have been generated conditional on the same class. Therefore, the resulting Bernoulli and Gaussian mixture parameters should correspond: concretely if an image, represented by continuous visual descriptors, has an high probability under a certain class, then its discrete shape measures should have an high probability under the same class.

In order to classify a query image f_j, the class node C is inferred thanks to the massage passing algorithm. This image, characterized by its continuous shape features $v_{j_1}, \ldots, v_{j_m}$ and its discrete shape features $DF\ 1_j, \ldots, DF\ k_j$ is considered as an "evidence" represented by:

$$P(f_j) = P(v_{j_1}, \ldots, v_{j_m}, DF\ 1_j, \ldots, DF\ n_j) = 1$$

when the network is evaluated. Thanks to the inference algorithm, the probabilities of each node are updated in function of this evidence. After the belief propagation, we know, $\forall i \in \{1, \ldots, k\}$, the posterior probability:

$$P(c_i|f_j) = P(c_i|v_{j_1}, \ldots, v_{j_m}, DF\ 1_j, \ldots, DF\ n_j)$$

The query f_j is assigned to the class c_i which maximizes this probability.

3 Symbol description

This section explains how we have adapted the theoretical method before mentioned to a symbol recognition problem. We present the visual features we used. The set of chosen features is composed of 3 different off-the-shelf shape descriptors and 3 shape measures. The choice of these features is not really important because, the aim of this paper is to show that combining shape features improves the symbol classification, whatever the used features. The distinction between shape descriptors and shape measures is determined by the size of the features: we consider single value features as shape measures and feature vectors as shape descriptors. Moreover, shape measures are discretized with a discretization threshold fixed at 0.5. This discretization has sense with this kind of features, because each of them is composed of a single value normalized between 0 and 1. Thus, a shape measure has an intrinsic meaning. Finally, this discretization enables to consider shape measures as discrete variables and thus to show the interest of discrete and continuous features combination for shape recognition.

In this perspective, we have chosen three pixel shape descriptors: the Generic Fourier Descriptor (GFD), the Zernike descriptor, and the $\mathcal{R}$-signature $1D$ and three shape measures: compactness, rectangularity and ellipticity. We briefly present each descriptor and measures below.

3.1 Shape descriptors

Generic Fourier descriptor: Generic Fourier descriptor is based on Fourier transform [41]. The rotation invariance is achieved by using the modified polar Fourier transform (MPFT), and the scaling invariance is achieved after normalization.

Zernike descriptor: Zernike descriptor [14] is a 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 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 $1D$ [30] 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. 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 Shape measures

Compactness: The compactness measure C represents the ratio of the shape area to the area of a circle (the most compact shape) having the same perimeter:

$$C = \frac{4\pi A}{P^2}$$

where P is the perimeter and A, the area.
This measure is invariant to translation, rotation, and scaling.

Rectangularity: The rectangularity degree [28] R is equal to the ratio of the shape area to the area of its minimal bounding box:

$$R = \frac{A}{L * l}$$

where A is the shape area and L (respectively l) is the length (respectively the width) of the minimal bounding box.

Ellipticity: The ellipticity degree ϵ is obtained from the ratio of the major axis to the minor axis [31]:

$$\epsilon = 1 - \frac{b}{a}$$

where a is the major axis and b the minor axis. This measure is invariant to rotation, translation, and homothety.

4 Dimensionality reduction

Only the n, $n \in \{1, 2, 3\}$ continuous 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. The large dimensions of initial visual signatures and their concatenation imply a dimensionality problem. In fact, a too large feature dimension increases the computation time and causes a wrong Gaussian mixture learning, because of the Small Sample Size (SSS) problem: there is a disproportion between the training set size and the feature vector dimension. To overcome this problem, we have used a dimensionality reduction method. A lot of methods have been proposed in the literature to reduce the dimension of vectors [8,25]. Among dimensionality reduction methods, we consider especially feature selection methods, because they enable to reduce dimension by selecting a subset of initial features, on the contrary to the methods which reduce dimension by providing new variables issued from initial variable combinations [10]. Methods of variable selection are then more suitable to our problem, because our aim is to reduce the number of features, in order to reduce the size of our Bayesian network and our method complexity. The most popular methods of variable selection are heuristics based on sequential runs, which consist in iteratively adding or removing variables [9]. In these approaches, it is possible to begin with an empty set of variables and to add variables to the variables which are already selected (it is the Sequential Forward Selection (SFS) [26]), or to begin with the set of all variables et to remove variables in this set (it is the Sequential Backward Selection (SBS)). These methods are known for their simplicity and their rapidity. However, they are known for their instability too. Moreover, since they do not explore all possible subsets of variables and they do not enable to come back during the process, they are not optimal.
Thus, we have chosen a feature selection method since it enables to extract from the feature vectors, just the most relevant and discriminating features, with a minimal information loss. The regression method LASSO (Least Absolute Shrinkage and Selection Operator) [35] has been used for its stability and implementation efficiency. Moreover, the LASSO method especially enables to select variables and takes into account the class variable values to select a subset of variables. In Sect. 5, we compare the results obtained on our database, by the LASSO and the SFS method which stay one of the most popular.
The principle of LASSO is to shrink the regression coefficients by imposing a penalty on their size (we speak of shrinkage methods too). These 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 $\sum_{j=1}^{p} |\beta_j| \leq s$.

The linear form of the LASSO has been applied in a pre-processing stage, totally independent of our Bayesian classifier, on our visual features. For each training set, y_i

represents the sum of the mean vector features of the class c_i, and $x_j = \{x_{j_1}, x_{j_2}, \ldots, x_{j_p}\}$ the p features of the instance j. Then, just the subset of the selected variables is used in our model.

The LASSO uses a $L1$ penalty: $\sum_{j=1}^{p} |\beta_j|$. This constraint implies that for small 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. Therefore, the variables corresponding to the coefficients different from zero are selected.

The LASSO solutions have been computed by the Least Angle Regression (LAR) procedure [11]. This algorithm exploits the special structure of the LASSO problem and provides an efficient way to compute the solutions simultaneously for all values of s.

Thus, we can distinguish two independent phases in our method: first, the LASSO is used to select the most relevant visual features. Secondly, our Gaussian-Mixtures and Bernoulli Mixture model is used to classify new images represented by the pre-selected continuous visual features and the three discrete shape measures.

5 Experimental results

We have used the symbols of GREC database [36] for our tests (see Fig. 4), especially created for the symbol recognition contest GREC'2005.

This database is mainly defined from two application domains, architecture and electronic, because these symbols are most largely used by graphic recognition teams and represent a great number of different forms. We have 50 different symbol models for which we have applied some noises based on Kanungo [18] model. These noises are similar to noise obtained when a document is scanned, printed, or photocopied. Moreover, we have applied to these symbols some rotations of different degrees and different zooms, in order to obtain a database of 3, 600 images, constituted of 72 different images per model.

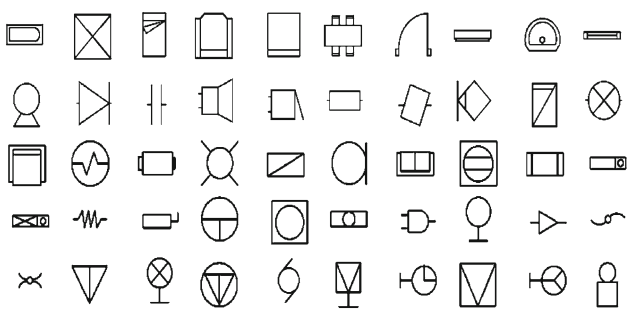

Fig. 4 GREC symbol database

Table 1 Mean numbers of variables in function of variable selection method

Number of variables from	GFD	Zernike	$\mathcal{R}$-signature $1D$
Without selection	225	34	180
SFS	141	34	48
LASSO	13	15	13

We have evaluated our method by performing a cross-validation by using 75% of the database for the training and the remaining 25% for the tests. The tests are repeated four times in order to use each database instance for the training and the tests. The recognition rate is obtained by taking the mean recognition rate of the 4 tests.

Since, we want to improve the recognition rate by combining descriptors and selecting variables, we limit ourselves to the experiments comparing:

- the classification after variable selection with the LASSO vs. the classification without automatic variable selection and with a well-known variable selection method,
- the classification by combining 2 or 3 continuous descriptors vs. the classification with only one continuous descriptor,
- the classification by combining discrete and continuous features vs. the classification with continuous descriptors only,
- the classification by combining 3 continuous descriptors with our method vs. two state-of-art classifiers.

First of all, we can remark that the reducing the size of the descriptors improves the recognition rate for all the classifiers. Moreover, the variable selection with the LASSO method has enabled us to significantly reduce the number of variables. Table 1 shows the mean number of variables selected for each descriptor with the LASSO method compared to the well-known SFS method [26]. We can see that the LASSO method enables to select fewer variables than the SFS method (see Table 1). Table 2 shows the mean recognition rates in function of different variable selection methods, by combining the 3 available descriptors with our Gaussian-Mixtures and Bernoulli Mixture model (GM-B) and two state-of-art classifiers: a classical SVM classifier [5] and the fuzzy k-nearest neighbor (FKNN) [19]. The recognition rates for these three classifiers without variable selection and after a variable subset selection with the SFS and LASSO methods, or with some random selections, are compared. For the random selection, the number of variables chosen randomly is set to the one obtained with the LASSO. The FKNN has been computed with $k = 1$ and with $k = m$ where m is the mean number of images per class in the training set. The results in Table 2 show that the variable selection with

Table 2 Mean recognition rates (in %) for SVM classifier, FKNN and GM-B in function of variable selection method

Variable selection method	SVM classifier	FKNN $k = 1$	FKNN $k = m$	GM-B
Without selection	87.6	89.9	88.6	89.8
Random selection	90.8	93.7	91.5	93.3
SFS	94.1	97.2	95.3	96.7
LASSO	95.7	98.8	96.2	100

Table 3 Mean recognition rates (in %) of GM-B after variable selection with the LASSO

G	Z	R	G+Z	G+R	Z+R	G+Z+R
99	100	46.1	100	99.3	100	100

the LASSO method improves the recognition rate by 8.7% on average compared to the classification without variable selection, by 5.3% on average with random selection, and by 1.8% on average compared to the SFS selection. Thus the LASSO method is more robust, experimentally, on this database, than the SFS method. In fact, the shrinkage methods like the LASSO are well known to be more stable than iterative methods, to select variables among a large set of variables but with few examples. Thus, the variables selected with the LASSO method have been used for our following experiments.

Let us consider Table 3. The notation G (respectively Z and R) means that the GFD descriptor (respectively the Zernike descriptor and the $\mathcal{R}$-signature $1D$) has been used. Finally, the "+" operator indicates that we have combined the descriptors. The recognition rates confirm that combining 2 or 3 descriptors performs always better than any of them alone. In fact, we observe that the combination of 2 descriptors increases the recognition rate by 18% on average compared to the use of only one descriptor. Besides we can notice that the combination of the 3 descriptors is better, by 18.3% on average, to use just one of them. Moreover, 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 $1D$). The bad behavior of a descriptor does not impede the other descriptor behaviors.

Finally, the last line of the Table 2 shows the effectiveness of our approach compared to the SVM classifier and the FKNN. The results have been obtained by combining the 3 descriptors and after using the variable selection method LASSO. It appears that the proposed GM-B results are always better than the ones of SVM and FKNN.

The initial database of 3,600 instances has been extended to a database of 5,400 instances by randomly generating occlusions on the half image set of each class from the initial database. In fact, we can meet this kind of degradation when we have to segment a graphical document where symbols are embedded into the graphic or are partially occluded by text for example. The generated occlusions are from different sizes, and their locations in the images have been chosen randomly. Now we have a larger and more distorted database, composed of 108 instances per class. For example, the Fig. 5 presents 4 symbol models (first column) and 5 occluded images disturbed by some noises derived from these models. Our classifier has been applied after variable subset selection with the LASSO. This time, our method has been evaluated by performing 3 cross-validations whose each proportion of the training set is 25,50 and 75% of the database, the remaining, respectively, 75,50, and 25% are hold for testing set. In each case, the tests are repeated 10 times in order that each database instance would be used for the training and the test. For each training set size, the recognition rate is obtained by taking the mean recognition rate of the 10 tests. On this database the LASSO method has enabled to select quite the same number of variables than with the initial database: 12 variables on average from GFD features, 13 from Zernike features and 13 from $\mathcal{R}$-signature $1D$ features. Let us consider Table 4. The used notations are the same as the ones previously used in Table 3. Moreover, the notation DF means that the three discrete shape measures have been used. The recognition rates show the descriptor combination interest. Indeed, even if the classification is less efficient on this database than on the initial one, the results show the combination of continuous descriptors improves the recognition rate. Moreover, the addition of the 3 discrete shape measures outperforms these results. In fact, the integration of the discrete measures improves the recognition rate by 3.8% on average compared to the recognition rate obtained by the combination of the 3 continuous descriptors. Finally, Table 5 shows that the proposed GM-B classifier performs always better than the SVM and the FKNN classifiers. In the same way, Table 6 shows the maximal and minimal values and the mean and the standard deviation of the recognition rates obtained by the 3 compared classifiers, during the 10 tests for a training on 50% of the database. The standard deviation is small whatever the classifier and shows a low variability of recognition rate following the different training and testing sets.

Table 7 shows CPU times of the SVM classifier, the FKNN, and the proposed GM-B classifier, for training and

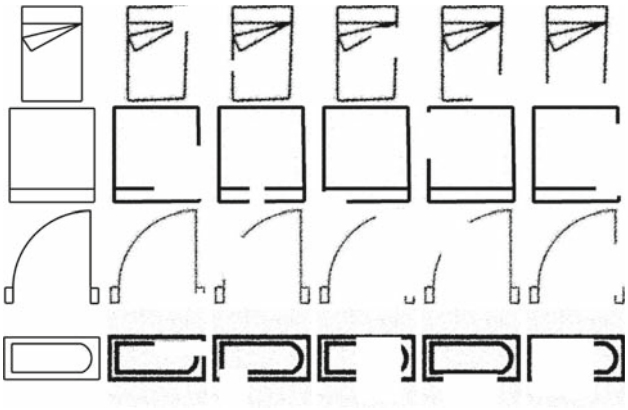

Fig. 5 Examples of noisy and occluded symbols for different models

test stages, with the same experimental conditions like in the Table 5. All the experiments have been performed with a processor Intel Core 2 Duo 2.40 GHz, 2 Go RAM, Windows OS. The three classifiers have been run with Matlab©. If we consider only the test stages (training has been made off line for the SVM and the GM-B classifier), the SVM classifier is faster than the two others. The CPU time is higher for the GM-B model because it depends on the number of Gaussians (in this case, 2) and the pre-defined precision of the EM algorithm. However, the processing time remains weak since it takes less than 0, 03 s per image. Following the discussion in Sect. 4, we can remark that without variable selection the CPU time raises drastically for the GM-B classifier.

6 Conclusion and future works

In this paper, we have proposed an original adaptation of the Bayesian theory to combine descriptors. We have shown that a Bayesian network has good properties for symbol recognition. In the proposed model, the bad behavior of a descriptor does not impede the behavior of the others. Moreover, we can take into account different types of descriptors. Indeed, we have combined discrete shape measures with continuous shape descriptors. This combination provides a classifier more robust to variability and scalability. Moreover, we have adapted the LASSO method, which solves our dimensionality problem and thus decreases our method complexity and especially increases the recognition rate. The experimental results are very promising and show the efficiency of our method.

In our future works, we want to use our approach in the case of very complex symbols like electrical wiring diagrams. In this case, to recover the maximum amount of information, it is useful to add more descriptors in our combination framework. Even if the LASSO has shown good results, it does not take into account correlation between variables and after selection, variables are always correlated. However, it is well

Table 4 Mean recognition rates (in %) of GM-B after variable selection with the LASSO—database including occluded symbols

Training part (%)	G	Z	R	G+Z	G+R	Z+R	G+Z+R	G+Z+R+DF
25	70.4	79	39.3	85.5	75.5	82.2	93.3	96.8
50	71	80.7	40.2	87.6	76.3	83.4	93.7	98.6
75	75.7	85.1	41.2	89.4	79.1	87.6	96.2	99.2

Table 5 Mean recognition rates (in %), by combining continous and discrete features (G+Z+R+DF), for SVM classifier, FKNN and GM-B after variable selection with the LASSO—database including occluded symbols

Training part (%)	SVM classifier	FKNN $k = 1$	FKNN $k = m$	GM-B
25	89.2	91.9	91.7	96.8
50	91	95.2	93	98.6
75	92.5	97.1	94.7	99.2

Table 6 Statistical measures (in %) for SVM classifier, FKNN and GM-B recognition rates, after variable selection with the LASSO, by combining continous and discrete features (G+Z+R+DF)—database including occluded symbols (training set = 50% of the database)

Measure	SVM classifier	FKNN $k = 1$	FKNN $k = m$	GM-B
Min	90.4	94.8	92.9	98.5
Max	91.7	95.4	93.03	98.65
Mean	91	95.2	93	98.6
Standard deviation	0.4	0.17	0.04	0.07

Table 7 CPU times (in seconds), for SVM classifier, FKNN and GM-B

Training part	SVM classifier		FKNN $k = 1$	FKNN $k = m$	GM-B with variable selection		GM-B without variable selection	
	Training (%)	Test			Training	Test	Training	Test
25	4	5	40	41	58	78	2726	25608
50	10	6	56	58	117	52	5696	17291
75	19	4	42	45	168	24	8110	8453

The CPU times are given for all the test images

known that the descriptors are often partially redundant since they address the same task. In this case, a more appropriate reduction method should be investigated.

Moreover, it can be interesting to annotate some symbols and add the information given by possible keywords associated with a subset of training data. Our motivation is based on the property of Bayesian networks to enable to manage, in a same network, different kinds of information (in this case different media), and to their ability to handle missing values.

References

1. Barbu, E., Chatelain, C., Adam, S., Heroux, P., Trupin, E.: A simple one class classifier with rejection strategy: application to symbol classification. In: 7th International Workshop, GREC 2007, Curitiba, Brazil, Sept 20–21 (2007)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
3. Breiman, L.: Random forests. In: Machine Learning, pp. 5–32 (2001)
4. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(2), 121–167 (1998)
5. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines, (2001)
6. Coustaty, M., Guillas, S., Visani, M., Bertet, K., Ogier, J.M.: On the joint use of a structural signature and a galois lattice classifier for symbol recognition. In: Graphic Recognition, volume 5046 of Lecture Notes in Computer Science, pp. 61–70. (2008)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J R. Stat. Soc. Ser. B Methodol. **39**(1), 1–38 (1977)
8. Denoeux, T., Masson, M.H.: Dimensionality reduction and visualization of interval and fuzzy data: a survey. In: Proceedings of the 56th Session of the International Statistical Institute (ISI é607), Lisboa, Portugal, Aug (2007)
9. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall, Englewood Cliffs NJ (1982)
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. 2nd edn. Wiley, London (2001)
11. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. **32**, 407–499 (2004)
12. Fornés, A., Escalera, S., Lladós, J., Sánchez, G., Mas, J.: Hand drawn symbol recognition by blurred shape model descriptor and a multiclass classifier. In: Graphic Recognition, volume 5046 of Lecture Notes in Computer Science, pp. 29–39. (2008)
13. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29**(2–3), 131–163 (1997)
14. 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. In: IEEE International Conference on Multimedia And Expo vol. 1, pp. 307–310. (2000)
15. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 4–37 (2000)
16. Jordan, M.I.: Graphical models. Mach. Learn. **19**, (2003)
17. Jordan, M.I. (ed.): Learning in Graphical Models. MIT Press, Cambridge (1999)
18. Kanungo, T., Haralick, R., Baird, H., Stuezle, W., Madigan, D.: A statistical, nonparametric methodology for document degradation model validation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1209–1223 (2000)
19. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybern. **15**(4), 580–585 (1985)
20. Kim, J.H., Pearl, J.: A computational model for combined causal and diagnostic reasoning in inference systems. In: IJCAI-83, pp. 190–193. (1983)
21. Kitamoto, A., Takagi, M.: Image classification using probabilistic models that reflect the internal structure of mixels. Pattern Anal. Appl. **2**(1), 31–43 (1999)
22. Lladós, J., Martí, E., Villanueva, J.J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. IEEE Trans. Pattern Anal. Mach. Intell. **23**(10), 1137–1143 (2001)
23. Nielsen, J.D., Rumí, R., Salmerón, A.: Supervised classification using probabilistic decision graphs. Comput. Stat. Data Anal. **53**(4), 1299–1311 (2009)
24. Paek, S., Chang, S.: A knowledge engineering approach for image classification based on probabilistic reasoning systems. In: IEEE International Conference on Multimedia and Expo, pp. 1133–1136. (2000)
25. Piccardi, M., Gunes, H., Otoom, A.F.: Maximum-likelihood dimensionality reduction in gaussian mixture models with an application to object classification. In: ICPR'08, pp. 1–4. (2008)
26. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognit.Lett. **15**(11), 1119–1125 (1994)
27. Robert, C.: A Decision-Theoretic Motivation. Springer, Berlin (1997)
28. Rosin, P. L.: Measuring rectangularity. Mach. Vis. Appl. **11**(4), 191–196 (1999)
29. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE Trans. Syst. Man Cybern. **21**(3), 660–674 (1991)
30. Tabbone, S., Wendling, L.: Technical symbols recognition using the two-dimensional radon transform. In: ICPR'02, vol. 2, pp. 200–203. Aug (2002)
31. Teague, M.R.: Image analysis via the general theory of moments. J. Opt. Soc. Am. **70**(8), 920–930 (1979)
32. Ramos-Terrades, O., Valveny, E., Tabbone, S.: On the combination of ridgelets descriptors for symbol recognition. In: Graphic

Recognition, volume 5046 of Lecture Notes in Computer Science, pp. 40–50. (2008)
33. Ramos-Terrades, O., Tabbone, S., Valveny, E.: A review of shape descriptors for document analysis. Int. Conf. Doc. Anal. Recognit. **1**, 227–231 (2007)
34. Ramos-Terrades, O., Valveny, E., Tabbone, S.: Optimal classifier fusion in a non-bayesian probabilistic framework. IEEE Trans. Pattern Anal. Mach. Intell. **31**(9), 1630–1644 (2009)
35. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B Methodol **58**(1), 267–288 (1996)
36. Valveny, E., Dosch, P.: Symbol recognition contest : a synthesis. In: Graphic Recognition, volume 3088 of Lecture Notes in Computer Science pp. 368–385. (2004)
37. Valveny, E., Dosch, P., Fornés, A., Escalera, S.: Graphics recognition. Recent advances and new opportunities: In: 7th International Workshop, GREC 2007, Curitiba, Brazil, September 20–21, 2007. Selected Papers, chapter Report on the 3rd Contest on Symbol Recognition, pp. 321–328. Springer, Berlin (2008)
38. Wendling, L., Rendek, J.: Symbol recognition using a 2-class hierarchical model of choquet integrals. In: CDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition, pp. 634–638. IEEE Computer Society, Washington, DC (2007)
39. Yang, S.: Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor. IEEE Trans. Pattern Anal. Mach. Intell. **27**(2), 278–281 (2005)
40. Zhang, D.S., Lu, G.: Review of shape representation and description techniques. Pattern Recognit. **37**(1), 1–19 (2004)
41. Zhang, D.S., Lu, G.: Shape-based image retrieval using general fourier descriptor. Signal Proces. Image Commun. **17**(10), 825–848 (2002)
42. Zhang, G.P.: Neural networks for classification: a survey. IEEE Trans. Syst. Man Cybern. **30**(4), 451–462 (2000)
43. Zhang, M., Jia, Y.: Probabilistic classification based image regions labeling. In: ICIG '04: Proceedings of the 3rd International Conference on Image and Graphics, pp. 100–103. IEEE Computer Society, Washington, DC (2004)
44. Zhang, W., Wenyin, L., Zhang, K.: Symbol recognition with kernel density matching. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2020–2024 (2006)