Learning mixtures of polynomials from data using B-spline interpolation

Pedro L. López-Cruz, Concha Bielza and Pedro Larrañaga
Computational Intelligence Group, Departamento de Inteligencia Artificial
Facultad de Informática, Universidad Politécnica de Madrid, Spain
pedro.lcruz@upm.es, {mcbielza,pedro.larranaga}@fi.upm.es

Abstract

Hybrid Bayesian networks efficiently encode a joint probability distribution over a set of continuous and discrete variables. Several approaches have been recently proposed for working with hybrid Bayesian networks, e.g., mixtures of truncated basis functions, mixtures of truncated exponentials or mixtures of polynomials (MoPs). We present a method for learning MoP approximations of probability densities from data using a linear combination of B-splines. Maximum likelihood estimators of the mixing coefficients of the linear combination are computed, and model selection is performed using a penalized likelihood criterion, i.e., the BIC score. Artificial examples are used to analyze the behavior of the method according to different criteria, like the quality of the approximations and the number of pieces in the MoP. Also, we study the use of the proposed method as a non-parametric density estimation technique in naive Bayes (NB) classifiers. Results on real datasets show that the non-parametric NB classifier using MoPs is comparable to the kernel density-based NB and better than Gaussian or discrete NB classifiers.

1 Introduction

Problems defined in hybrid domains with both continuous and discrete variables are frequently found in different fields of science. A Bayesian network (Pearl, 1988) is a kind of probabilistic graphical model which encodes a factorization of the joint probability distribution over a set of random variables. Hybrid Bayesian networks for domains with continuous and discrete variables pose a number of challenges regarding the representation of conditional probability distributions, inference, learning from data, etc.

Langseth et al. (2012) have recently proposed mixtures of truncated basis functions (MoTBFs) as a framework for representing hybrid Bayesian networks. MoTBFs generalize mixtures of truncated exponentials (MTEs) (Moral et al., 2001) and mixtures of polynomials (MoPs) (Shenoy and West, 2011). MoTBFs, MTEs and MoPs are closed under multiplication, addition and integration. Therefore, exact probabilistic inference can be performed using the Shenoy-Shafer (1990) architecture.

Different methods have been proposed for approximating MTEs from data by using least squares (Rumí et al., 2006) or maximum likelihood (ML) estimation (Langseth et al., 2010). Recently, Langseth et al. (2012) propose methods for estimating MoTBFs by minimizing the Kullback-Leibler divergence.

In this paper, we present a method for learning MoPs directly from data using B-spline interpolation (Zong, 2006). Given a dataset, this method can be used for finding a MoP approximation of the probability density which generated the data. Our proposal ensures that the MoP is a valid density, i.e., it is continuous, non-negative and integrates to one. Previous proposals for learning MoPs assume that the mathematical expression of the generating parametric density is known (Shenoy and West, 2011) or that the true densities of the Chebyshev points are available (Shenoy, 2012). On the contrary, the proposed method only uses

the dataset without assuming any prior knowledge. First, the probability density is approximated using the B-spline interpolation method by Zong (2006), which provides ML estimators of the mixing coefficients of the linear combination of B-splines from the data. Second, the approximated B-splines are developed into a MoP function. Third, penalized likelihood scores such as the BIC score are used for performing model selection in a principled way, avoiding overfitting and models with high complexity.

The remainder of the paper is organized as follows. Section 2 reviews MoPs and the methods found in the literature for learning them. Section 3 details the proposed method for learning MoP approximations of probability densities from data. Section 4 shows the use of the proposed methods for non-parametric density estimation in naive Bayes classifiers. Sections 5 and 6 include the experimental evaluation of the proposed methods. Finally, Section 7 ends with conclusions and future work.

2 Mixtures of polynomials

Let X be a one-dimensional random variable with probability density $f_X(x)$. A MoP approximation of $f_X(x)$ over a closed domain $\Omega_X = [\omega_1, \omega_2] \subset \mathbb{R}$ is a L-piece d-degree piecewise function of the form (Shenoy and West, 2011)

$$\varphi_X(x) = \begin{cases} p_i(x) & \text{for } x \in A_i,\ i = 1, \dots, L \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $p_i(x)$ is a polynomial function $b_{0i} + b_{1i}x + b_{2i}x^2 + \cdots + b_{di}x^d$, $\{b_{0i}, \dots, b_{di}\}$ are constants and $A_1, \dots, A_L$ are disjoint intervals in Ω_X which do not depend on x with $\Omega_X = \cup_{i=1}^{L} A_i$, $A_i \cap A_j = \emptyset, i \neq j$.

MoPs are closed under multiplication, integration, differentiation and addition. Therefore, exact inference can be performed with the Shenoy-Shafer algorithm. Previous works used the Taylor series expansion (TSE) (Shenoy and West, 2011) or the Lagrange interpolating polynomial (LIP) (Shenoy, 2012) for estimating $p_i(x)$. The mathematical expression of the probability density $f_X(x)$ needs to be known for computing the TSE. However, real data might not fit any known parametric density, so TSE cannot be used in practice. Similarly, Shenoy (2012) proposes estimating $p_i(x)$ as the LIP over the Chebyshev points defined in A_i. However, the true probability densities of the Chebyshev points in each A_i need to be known or estimated beforehand.

3 Learning MoPs using B-spline interpolation

B-splines or basis splines (Schoenberg, 1946) are polynomial curves which form a basis for the space of piecewise polynomial functions over a closed domain $\Omega_X = [\omega_1, \omega_2]$ (Faux and Pratt, 1979). Zong (2006) proposed a method for finding B-spline approximations of probability density functions from data. He found a B-spline approximation of the density $f_X(x)$ as a linear combination of $M = L + r - 1$ B-splines

$$\varphi_X(x\ ;\ \boldsymbol{\alpha}) = \sum_{j=1}^{M} \alpha_j B_j^r(x), \tag{2}$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_M)$ are the mixing coefficients and $B_j^r(x), j = 1, \dots, M$ are B-splines with order r (degree $d = r - 1$).

Given a non-decreasing knot sequence of real numbers $\boldsymbol{\delta} = (a_0, a_1, \dots, a_L)$, $a_{i-1} < a_i$, the jth B-spline $B_j^r(x)$ with order r is written as

$$B_j^r(x) = (a_j - a_{j-r})H(x - a_{j-r}) \cdot \sum_{t=0}^{r} \frac{(a_{j-r+t} - x)^{r-1} H(a_{j-r+t} - x)}{w'_{j-r}(a_{j-r+t})}, \tag{3}$$

where $w'_{j-r}(x)$ is the first derivative of $w_{j-r}(x) = \prod_{u=0}^{r}(x - a_{j-r+u})$ and $H(x)$ is the Heaviside function

$$H(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0. \end{cases}$$

A B-spline $B_j^r(x)$ can be written as a MoP function (Equation (1)) with L pieces, where each piece $p_i(x)$ is defined as the expansion of Equation (3) in the interval $A_i = [a_{i-1}, a_i), i = 1, \dots, L$. To define a MoP using B-spline interpolation, four elements need to be specified:

the order r, the number of knots/pieces L, the knot sequence $\boldsymbol{\delta}$ and the mixing coefficients $\boldsymbol{\alpha}$. We used uniform B-splines so the knots in the sequence $\boldsymbol{\delta}$ are equally spaced and yield intervals A_i with equal width: $a_i - a_{i-1} = \frac{\omega_2 - \omega_1}{L}$. MoPs are closed under multiplication and addition. Therefore, the linear combination of M B-splines with order r (Equation (2)) yields a MoP function with L pieces, where each piece $p_i(x)$ is a polynomial with order r defined in the interval A_i: $p_i(x) = \sum_{j=1}^{M} \alpha_j B_j^r(x), \forall x \in A_i = [a_{i-1}, a_i)$.

B-splines have a number of interesting properties for learning MoP approximations of probability densities, e.g., $B_j^r(x)$ is right side continuous, differentiable, positive in (a_j, a_{j+r+1}) and zero outside.

Given a dataset $\mathcal{D} = \{x_1, \ldots, x_N\}$ with N observations of variable X, Zong (2006) derived the following iterative formula for finding the ML estimators of the mixing coefficients, $\widehat{\boldsymbol{\alpha}}$, in Equation (2):

$$\widehat{\alpha}_j^{(q)} = \frac{1}{N c_j} \sum_{x \in \mathcal{D}} \frac{\widehat{\alpha}_j^{(q-1)} B_j^r(x)}{\varphi_X\left(x; \widehat{\boldsymbol{\alpha}}^{(q-1)}\right)}, j = 1, \ldots, M, \tag{4}$$

where q is the iteration number in the optimization process and

$$c_j = \int_{a_{j-r}}^{a_j} B_j^r(x) dx = \frac{a_j - a_{j-r}}{r}.$$

Zong (2006) showed that Equation (4) yields the only maximum of the log-likelihood of $\mathcal{D}$ given the approximation (Equation (2)), subject to the constraints $\sum_{j=1}^{M} \alpha_j c_j = 1$ and $\alpha_j \geq 0, j = 1, \ldots, M$. These constraints ensure that $\varphi_X(x; \widehat{\boldsymbol{\alpha}})$ is a valid probability density, i.e., it is non-negative and integrates to one. The initial values $\widehat{\alpha}_j^{(0)}$ are set to $1/\sum_{j=1}^{M} c_j$. Equation (4) iterates until $\left|\frac{\ell^{(q)} - \ell^{(q-1)}}{\ell^{(q)}}\right| < \epsilon$, where $\ell^{(q)}$ is the log-likelihood of $\mathcal{D}$ given $\varphi_X\left(x; \widehat{\boldsymbol{\alpha}}^{(q)}\right)$ at iteration q of the optimization process. We used $\epsilon = 10^{-6}$ in our experiments. The computational complexity of this optimization process is $O(MNq_{\max})$, where $q_{\max}$ is the number of iterations of Equation (4) performed until the algorithm converges.

Algorithm 1 summarizes the whole process for obtaining a MoP approximation of a probability density function using a dataset. Algorithm 1 needs the number of pieces L be specified a priori. Since the ML estimators of the mixing coefficients, $\widehat{\boldsymbol{\alpha}}$, are computed in Equation (4), we can use a penalized likelihood score to perform model selection and find L. Here, we used the Bayesian information criterion (BIC) and selected the MoP with the highest BIC score:

$$BIC(\varphi_X(x), \mathcal{D}) = \ell(\mathcal{D}|\varphi_X(x)) - \frac{(M-1)\log N}{2}. \tag{5}$$

Algorithm 1. *Learning a MoP approximation of a probability density from data*

Inputs: A dataset $\mathcal{D}$ with N observations, the number of pieces (L) and the order of the polynomials (r).

Outputs: A L-piece $(r-1)$-degree MoP approximation $\varphi_X(x; \widehat{\boldsymbol{\alpha}})$ of the probability density underlying the dataset $\mathcal{D}$.

Steps:

1. *Compute the domain of the approximation $\Omega_X = [\omega_1, \omega_2]$ where $\omega_1 = \min_{\mathcal{D}}(X)$ and $\omega_2 = \max_{\mathcal{D}}(X)$.*
2. *Compute the knot sequence $\boldsymbol{\delta} = (a_0, a_1, \ldots, a_L)$ and define the intervals $A_i = [a_{i-1}, a_i), i = 1, \ldots, L$.*
3. *Build the $M = L + r - 1$ B-splines $B_j^r(x)$ by applying Equation (3).*
4. *Compute the ML estimators of the mixing coefficients, $\widehat{\boldsymbol{\alpha}}$, by applying Equation (4).*
5. *Compute the polynomials $p_i(x)$ as the linear combination of the B-splines defined for each interval A_i, and build the MoP.*
6. *Normalize the MoP by dividing the coefficients of the polynomials $p_i(x)$ by $\int_{\Omega_X} \varphi_X(x) dx$.*

Algorithm 1 can be easily extended for finding MoP approximations of multivariate probability

densities from data using multivariate B-spline approximations as proposed by Zong (2006). Then, the conditional density of a variable X given its continuous parents $\mathbf{Y}$ can be evaluated by dividing the multivariate MoP approximations of the joint densities $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})$ and $\varphi_{\mathbf{Y}}(\mathbf{y})$. However, obtaining MoP approximations of these joint densities is more challenging due to the higher number of parameters and the increasing number of instances needed to estimate them.

4 Non-parametric naive Bayes classifiers using MoPs

In this section, we show how to use the proposed method as a non-parametric density estimation technique in naive Bayes (NB) classifiers. We consider a supervised classification problem with a discrete class variable C with values in $\Omega_C = \{1,\ldots,K\}$ and a vector of n continuous predictive variables $\mathbf{X} = (X_1,\ldots,X_n)$ with $\Omega_{X_v} \subset \mathbb{R}, v = 1,\ldots,n$. The NB classifier (Minsky, 1961) models the probability of the class labels as a categorical distribution $p_C(c), c \in \Omega_C$. The predictive variables are assumed to be conditionally independent given the class. Here, we model the conditional densities of every predictive variable X_v given the class $C = c$ with a MoP $\varphi_{X_v|c}(x_v)$. Algorithm 2 details the process for learning a NB classifier from data using MoPs.

Algorithm 2. *Learning NB classifiers with MoP approximations of the conditional density functions*

Inputs: A dataset $\mathcal{D} = \{(\mathbf{x}_z, c_z)\}, z = 1,\ldots,N$, *where* $\mathbf{x}_z = (x_{z1},\ldots,x_{zn})$, *the order* r *of the polynomials and the maximum number of pieces* L_{max} *for each MoP.*

Outputs: The estimated probabilities $p_C(c)$ *and* $\varphi_{X_v|c}(x_v)$.

1. *For each class value* $c \in \Omega_C = \{1,\ldots,K\}$
   (a) *Estimate* $p_C(c)$
   (b) *For each variable* $X_v \in \mathbf{X}$
      i. *Find* $\mathcal{D}_{v|c} = \{x_{zv} \in \mathcal{D} | c_z = c\}$.
      ii. *For each* $L \in \{1,\ldots,L_{max}\}$:
         A. *Find a MoP* $\varphi_{X_v|c}(x_v)$ *from* $\mathcal{D}_{v|c}$ *with* L *pieces (Algorithm 1).*
         B. *Compute* $BIC(\varphi_{X_v|c}(x_v), \mathcal{D}_{v|c})$ *in Equation* (5).
      iii. *Select the MoP with the highest BIC score.*

Once the probability distributions have been estimated with Algorithm 2, a new instance $\mathbf{x}$ is classified by applying the maximum a posteriori rule: $c^* = \arg\max_{c\in\Omega_C} p_C(c) \prod_{v=1}^{n} \varphi_{X_v|c}(x_v)$.

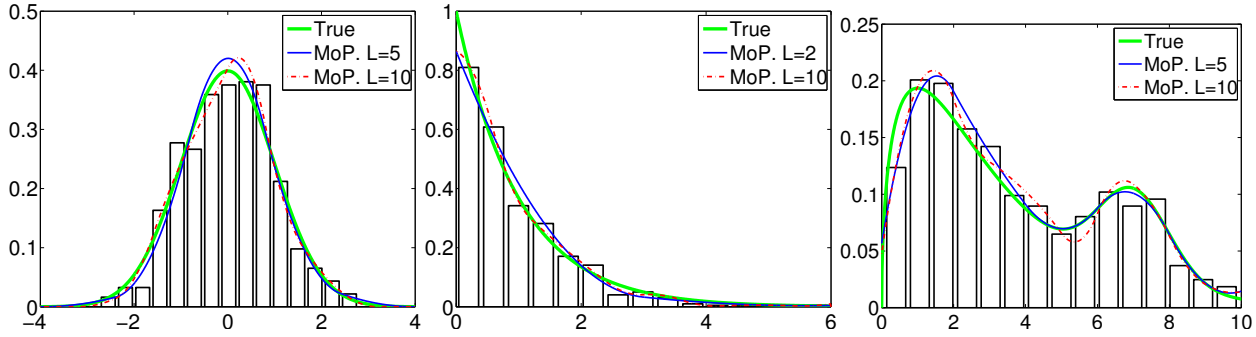
5 Experiments with MoP approximations

We analyzed the behavior of Algorithm 1 for building MoP approximations of probability densities from data using artificial examples. Figure 1 shows the MoPs obtained with 500 observations sampled from a Gaussian, an exponential and a mixture model. MoPs with order $r = 3$ and $L \in \{1,\ldots,10\}$ pieces were obtained using Algorithm 1. The MoPs obtained using the BIC score had fewer pieces than the MoPs with the highest log-likelihood. Equation (6) shows the MoP with the highest BIC score for the finite mixture distribution ($L = 5$ in Figure 1(c)) as an example:

$$\varphi_X(x) = \begin{cases} 0.0567 + 0.1924x - 0.0627x^2 & 0 \le x < 2 \\ 0.3246 - 0.0756x + 0.0043x^2 & 2 \le x < 4 \\ 0.5716 - 0.1990x + 0.0197x^2 & 4 \le x < 6 \\ -1.0265 + 0.3336x - 0.0247x^2 & 6 \le x < 8 \\ 1.6972 - 0.3473x + 0.0179x^2 & 8 \le x \le 10 \end{cases} \tag{6}$$

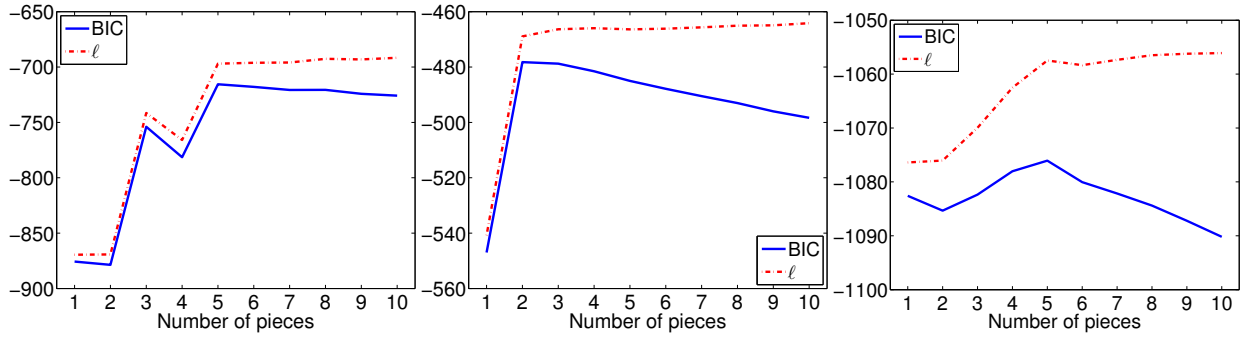
It is easy to check that the MoP in Equation (6) is continuous for $x \in \{2,4,6,8\}$ and $\int_0^{10} \varphi_X(x)dx = 1$.

We studied the influence of the number of pieces L in the MoP approximations of the three densities in Figure 1. Figure 2 shows the log-likelihood and the BIC score of the MoPs with different values of $L \in \{1,\ldots,10\}$. In general, the log-likelihood of the MoPs increased with the number of pieces L. However, some values of L yielded lower log-likelihood values than MoPs



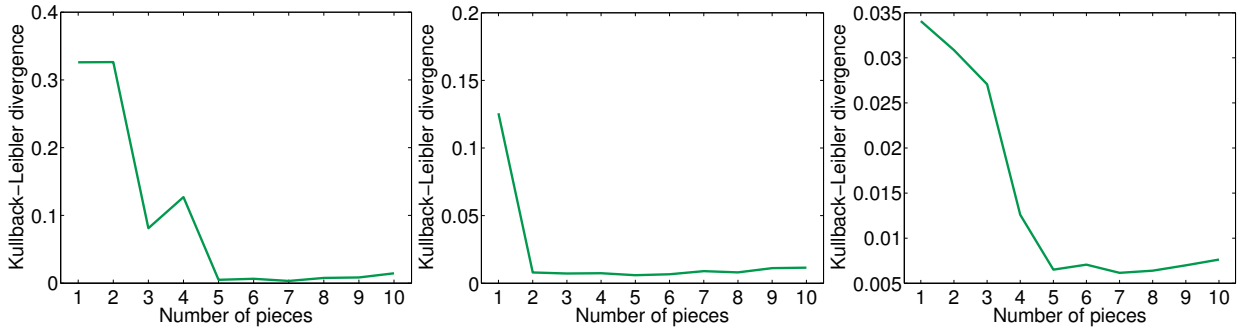
(a) $\mathcal{N}(0,1)$ (b) $Exp(1)$ (c) $0.8\chi^2(3) + 0.2\mathcal{N}(7,1)$

Figure 1: MoP approximations of a sample with 500 data from (a) a Gaussian, (b) an exponential and (c) a mixture of a χ^2 and a Gaussian distributions. The figure shows the true density (light solid line) and the MoP approximations with the highest BIC score (dark solid line) and the highest log-likelihood (dashed line). The histogram of the sample is also shown.



(a) $\mathcal{N}(0,1)$ (b) $Exp(1)$ (c) $0.8\chi^2(3) + 0.2\mathcal{N}(7,1)$

Figure 2: BIC score (solid) and log-likelihood (dashed) of the MoP approximations of the distributions in Figure 1 for different numbers of pieces $L \in \{1, \ldots, 10\}$.



(a) $\mathcal{N}(0,1)$ (b) $Exp(1)$ (c) $0.8\chi^2(3) + 0.2\mathcal{N}(7,1)$

Figure 3: Kullback-Leibler divergence of the MoP approximations and the true distributions in Figure 1 for different numbers of pieces $L \in \{1, \ldots, 10\}$.

with fewer pieces, e.g., the MoP with $L = 4$ pieces in Figure 2(a) had a lower log-likelihood than the MoP with $L = 3$ pieces. The domain of the MoP Ω_X was divided into L intervals with equal width. Therefore, we conclude that a bad partition of the domain Ω_X can yield worse MoP approximations even if a higher number of pieces is used. This highlights the importance of choosing the cut points of the intervals A_i which partition Ω_X. We can also see that the changes in the log-likelihood of the approximations become less prominent as we increase L.

We observed that choosing the MoP with the highest log-likelihood yielded approximations with a high number of pieces which overfitted the data, e.g., Figures 1(b) and 1(c) show small oscillations in the MoPs with the highest log-likelihood ($L = 10$). Figure 3 shows the Kullback-Leibler (KL) divergence of the MoPs and the true distributions for different L. We can see that the number of pieces selected using the BIC score yielded good approximations with a low KL divergence. Increasing L did not yield important reductions in the KL divergence. In fact, the KL divergence increased for high values of L, confirming that overfitting can occur in MoPs with many pieces.

6 Experiments with NB classifiers

We evaluated the proposed Algorithm 2 for learning a non-parametric NB classifier using MoPs. We set the order of the polynomials to $r = 3$, and the maximum number of pieces for a MoP to $L_{max} = 8$. We retrieved 14 datasets from the UCI[1] and KEEL[2] repositories and compared the proposed NB classifier (NBMoPBIC) with other NB classifiers implemented in Weka which use: Gaussian densities (NBGauss), kernel-based densities (NBKernel), Fayyad and Irani's (1993) discretization (NBFI) and equal-frequency discretization with 5 bins (NBEF5) or 10 bins (NBEF10).

Table 1 shows the mean accuracy achieved by each classifier in each dataset estimated using a stratified 10-fold cross-validation. NBMoPBIC achieved the best accuracy in six datasets, whereas NBKernel yielded the best accuracy in five datasets. The average ranking of the algorithms was NBKernel $\succ$ NBMoPBIC $\succ$ NBEF10 $\succ$ NBFI $\succ$ NBEF5 $\succ$ NBGauss. The null hypothesis of equal performance of all algorithms was rejected at a significance level $\alpha = 0.05$ using Friedman's test (p-value = 0.0237) and Iman-Davenport test (p-value = 0.0182).

Table 2 shows the number of datasets in which the first of the two algorithms in the tested null hypothesis (H_0) won, tied or lost against the second algorithm. NBMoPBIC yielded better results in more datasets than the other algorithms, with the exception of NBKernel. NBKernel achieved better results in more datasets than the other algorithms. NBGauss lost in more datasets than the other algorithms.

Table 2 also includes the results of the statistical tests for finding significant differences between the algorithms. The binomial test checks whether or not the ratio of wins versus losses is significant. No significant differences between the number of wins and losses were found at a significance level $\alpha = 0.05$. Considering $\alpha = 0.1$, NBMoPBIC significantly outperformed NBGauss, whereas NBKernel significantly outperformed NBEF5. The Bergmann-Hommel post-hoc test (García and Herrera, 2008) checks all the pairwise comparisons between algorithms in all datasets. We did not find statistically significant differences between any pair of algorithms. This test ensures that the rejected null hypotheses are compatible, and this restriction makes it more difficult to find significant results when many algorithms are compared. Therefore, we also applied the Wilcoxon rank-sum non-parametric test, which compares each pair of algorithms independently taking into account all the datasets. According to this test, NBMoPBIC and NBKernel significantly outperformed NBGauss and NBEF5. Additionally, NBKernel outperformed NBFI. NBEF10 was the best performing discretization algorithm. No significant differences were found between NBMoPBIC, NBKernel and NBEF10. However, NBMoPBIC and NBKernel won in more datasets than NBEF10.

[1] Available at `http://archive.ics.uci.edu/ml/`
[2] Available at `http://www.keel.es/`

Table 1: Mean accuracy of the classifiers estimated using a stratified 10-fold cross-validation. The best result for each dataset is highlighted with boldface letters.

	NBMoPBIC	NBGauss	NBKernel	NBFI	NBEF5	NBEF10
appendicitis	**0.8582**	0.8482	**0.8582**	0.8391	0.8200	0.8500
fourclass	0.8793	0.7541	**0.8839**	0.7818	0.7691	0.8341
glass2	**0.9485**	0.9113	0.9208	0.9251	0.9069	0.9346
haberman	0.7295	0.7453	0.7452	0.7224	0.7388	**0.7585**
ion	**0.9229**	0.8117	0.9200	0.8916	0.8859	0.8887
iris	**0.9600**	**0.9600**	**0.9600**	0.9333	0.9333	0.9400
liver	0.6453	0.5512	**0.6832**	0.5775	0.6394	0.6129
newthyroid	0.9487	**0.9632**	0.9630	0.9489	0.9541	0.9587
phoneme	**0.7914**	0.7600	0.7840	0.7720	0.7709	0.7707
svmguide1	0.9432	0.9313	0.9590	**0.9642**	0.9601	0.9625
vehicle	0.5992	0.4633	0.6134	0.6122	0.5863	**0.6323**
waveform	**0.8106**	0.8088	0.8070	0.8078	0.8082	0.8064
wdbc	0.9456	0.9331	**0.9490**	0.9455	0.9350	0.9473
wine	0.9778	0.9778	0.9778	**0.9833**	**0.9833**	0.9667

Table 2: Statistical comparison of the NB classifiers. The table shows the number of datasets in which the first algorithm in the tested null hypothesis (H_0) wins, ties or loses against the second algorithm. The p-values of the binomial, Bergmann-Hommel and Wilcoxon rank-sum tests are reported. Statistically significant results at a significance level $\alpha = 0.05$ are highlighted in boldface.

H_0	W / T / L	p_{Binomial}	$p_{\text{Berg-Hom}}$	p_{Wilcoxon}
NBMoPBIC = NBKernel	4 / 2 / 8	0.3877	1.0000	0.1294
NBMoPBIC = NBEF10	8 / 0 / 6	0.7905	1.0000	0.3910
NBMoPBIC = NBFI	10 / 0 / 4	0.1796	0.7423	0.1040
NBMoPBIC = NBEF5	10 / 0 / 4	0.1796	0.1792	**0.0353**
NBMoPBIC = NBGauss	9 / 3 / 2	0.0654	0.1760	**0.0244**
NBKernel = NBEF10	10 / 0 / 4	0.1796	1.0000	0.1726
NBKernel = NBFI	10 / 0 / 4	0.1796	0.4316	**0.0203**
NBKernel = NBEF5	11 / 0 / 3	0.0574	0.1000	**0.0017**
NBKernel = NBGauss	9 / 2 / 3	0.1460	0.0821	**0.0068**
NBEF10 = NBFI	9 / 0 / 5	0.4240	1.0000	**0.0494**
NBEF10 = NBEF5	10 / 0 / 4	0.1796	0.7423	**0.0494**
NBEF10 = NBGauss	10 / 0 / 4	0.1796	0.7423	**0.0295**
NBFI = NBEF5	8 / 2 / 4	0.3877	1.0000	0.3013
NBFI = NBGauss	9 / 0 / 5	0.4240	1.0000	0.1531
NBEF5 = NBGauss	8 / 0 / 6	0.7905	1.0000	0.2676

7 Conclusion

We have presented a method for learning MoP approximations of probability densities from data using a linear combination of B-splines. The ML estimators of the mixing coefficients of the linear combination were found and the BIC score was used for model selection. This provided a principled way for finding the number of pieces in a MoP, which yielded accurate approximations and avoided overfitting.

The use of MoPs as a non-parametric density estimation technique for naive Bayes classifiers was also studied. NB with MoPs outperformed

Gaussian NB and discrete NB with EF5 discretization. NB with MoPs was comparable to kernel density-based NB and discrete NB with EF10 discretization. MoPs offer some advantages over kernels as non-parametric density estimators. First, MoPs provide an explicit model of the generating probability density. Second, MoPs are more efficient than kernels regarding storage and classification time because MoPs do not need to save and analyze the complete dataset to evaluate the density of a value. On the contrary, training time is higher for MoPs because parameter estimation is involved, although Equation (4) converges in few iterations (Zong, 2006).

Future work includes the extension of Algorithm 1 so that the intervals A_i do not have the same width (non-uniform B-splines). Finding the best knot sequence given a dataset is expected to reduce the number of pieces necessary to find an accurate MoP approximation of the underlying probability density. Heuristic or optimization techniques, e.g., simulated annealing or differential evolution, could be used to find estimators of the knots. Here, we only considered MoPs with order $r = 3$, but higher orders need to be investigated in the future. Also, extensions to more complex Bayesian classifiers which do not assume conditional independence of the predictive variables given the class will be considered, e.g., tree-augmented naive Bayes, k-dependence Bayesian classifiers, etc. Finally, the performance of the proposed method will be compared with other non-parametric techniques, e.g., MoTBFs, MTEs, etc.

Acknowledgments

This work has been supported by the Spanish Economy and Competitiveness Ministry, Cajal Blue Brain (C080020-09), TIN2010-20900-C04-04 and Consolider Ingenio 2010-CSD2007-00018 projects. PLLC is supported by the Spanish Education Ministry (FPU AP2009-1772).

References

I.D. Faux and M.J. Pratt. 1979. *Computational Geometry for Design and Manufacture*. Wiley.

U. M. Fayyad and K. B. Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the 13th IJCAI*, pages 1022–1027. Morgan Kaufmann.

S. García and F. Herrera. 2008. An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694.

H. Langseth, T. D. Nielsen, R. Rumí, and A. Salmerón. 2010. Parameter estimation and model selection for mixtures of truncated exponentials. *International Journal of Approximate Reasoning*, 51:485–498.

H. Langseth, T. D. Nielsen, R. Rumí, and A. Salmerón. 2012. Mixtures of truncated basis functions. *International Journal of Approximate Reasoning*, 53:212–227.

M. Minsky. 1961. Steps toward artificial intelligence. *Proceedings of the Institute of Radio Engineers*, 49:8–30.

S. Moral, R. Rumí, and A. Salmerón. 2001. Mixtures of truncated exponentials in hybrid Bayesian networks. In *Proc. of the 6th ECSQARU. LNAI 2143*, pages 145–167. Springer-Verlag.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

R. Rumí, A. Salmerón, and S. Moral. 2006. Estimating mixtures of truncated exponentials in hybrid Bayesian network. *Test*, 15(2):397–421.

I. J. Schoenberg. 1946. Contributions to the problem of approximation of equidistant data by analytic functions. Part A: On the problem of smoothing of graduation. A first class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4:45–99.

P. P. Shenoy and G. Shafer. 1990. Axioms for probability and belief functions propagation. In *Proc. of the 4th UAI*, pages 169–198. North-Holland.

P. P. Shenoy and J. C. West. 2011. Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52(5):641–657.

P. P. Shenoy. 2012. Two issues in using mixtures of polynomials for inference in hybrid Bayesian networks. *International Journal of Approximate Reasoning*, 53(5):847–866.

Z. Zong. 2006. *Information-Theoretic Methods for Estimating Complicated Probability Distributions*. Elsevier.