Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 26 (2005) 1761–1771

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Bayesian network classification using spline-approximated kernel density estimation

Yaniv Gurwicz, Boaz Lerner *

Pattern Analysis and Machine Learning Lab, Department of Electrical and Computer Engineering, Ben-Gurion University, P.O. Box 653, 84105 Beer-Sheva, Israel

Received 7 November 2004; received in revised form 7 November 2004
Available online 8 April 2005
Communicated by E. Backer

Abstract

The likelihood for patterns of continuous features needed for probabilistic inference in a Bayesian network classifier (BNC) may be computed by kernel density estimation (KDE), letting every pattern influence the shape of the probability density. Although usually leading to accurate estimation, the KDE suffers from computational cost making it unpractical in many real-world applications. We smooth the density using a spline thus requiring for the estimation only very few coefficients rather than the whole training set allowing rapid implementation of the BNC without sacrificing classifier accuracy. Experiments conducted over a several real-world databases reveal acceleration in computational speed, sometimes in several orders of magnitude, in favor of our method making the application of KDE to BNCs practical.

Keywords: Bayesian networks; Classification; Kernel density estimation; Naïve Bayesian classifier; Spline

1. Introduction

1.1. Density estimation for Bayesian network classifiers

A Bayesian network (BN) represents the joint probability distribution (density) $p(\boldsymbol{X})$ over a set of n domain variables $X = \{X_1, \ldots, X_n\}$ graphically (Pearl, 1988; Heckerman, 1995). An arc and a lack of an arc between two nodes in the graph demonstrate, respectively, dependency and independency between variables corresponding to these nodes (Fig. 1). A connection between X_i and its parents $\boldsymbol{Pa}_i$ in the graph is quantified probabilistically using the data. A node having no parents embodies the prior probability of the corresponding variable. By ordering the variables topologically, extracting the general factorization of this ordering (using

* Corresponding author.
E-mail addresses: yanivg@ee.bgu.ac.il (Y. Gurwicz), boaz@ee.bgu.ac.il (B. Lerner).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.12.008

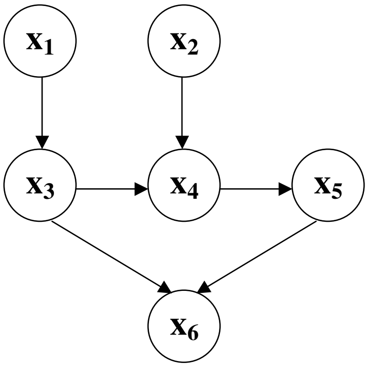

Fig. 1. A graph of an example Bayesian network. Arcs manifest dependencies between nodes representing variables.

the chain rule of probability) and applying the directed Markov property, we can decompose the joint probability distribution (density)

$$p(\boldsymbol{X}) = p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|\boldsymbol{Pa}_i). \tag{1}$$

The naïve Bayesian classifier (NBC) is a BN used for classification thus belonging to the Bayesian network classifier (BNC) family (John and Langley, 1995; Heckerman, 1995; Friedman et al., 1998; Lerner, 2004). It predicts a class C for a pattern $\boldsymbol{x}$ using Bayes' theorem

$$P(C|\boldsymbol{X} = \boldsymbol{x}) = \frac{p(\boldsymbol{X} = \boldsymbol{x}|C) \cdot P(C)}{p(\boldsymbol{X} = \boldsymbol{x})} \tag{2}$$

i.e., it infers the posterior probability that $\boldsymbol{x}$ belongs to C, $P(C|\boldsymbol{X} = \boldsymbol{x})$, by updating the prior probability for that class, $P(C)$, by the class-conditional probability density or likelihood for $\boldsymbol{x}$ to be generated from this class, $p(\boldsymbol{X} = \boldsymbol{x}|C)$, normalized by the unconditional density (evidence), $p(\boldsymbol{X} = x)$. The NBC represents a restrictive assumption of conditional independence between the variables (domain features) given the class allowing the decomposition and computation of the likelihood employing local probability densities

$$p(\boldsymbol{X}|C) = \prod_{i=1}^{n} p(X_i|C). \tag{3}$$

Estimating probability densities of variables accurately is a crucial task in many areas of machine learning (Silverman, 1986; Bishop, 1995). While estimating the probability distribution of a discrete feature is easily performed by computing the frequencies of its values in a given database, the probability density of a continuous feature taking any value in an interval cannot be estimated similarly thus requiring other, more complex methodologies. This is a major difficulty in the implementation of BNCs (John and Langley, 1995; Friedman et al., 1998; Elgammal et al., 2003; Lerner, 2004), and it requires either discretization of the variable into a collection of bins covering its range (Heckerman, 1995; Friedman et al., 1998; Yang and Webb, 2002; Malka and Lerner, 2004) or estimation, using parametric, non-parametric or semi-parametric methods (John and Langley, 1995; Lerner, 2004). Discretization is usually chosen for problems having small sample sizes that cannot guarantee accurate density estimation (Yang and Webb, 2002). Noticeably, prediction based on discretization is prone to errors due to lost of information. Generally, the accuracy discretization methods provide will peak for a specific range of bin sizes deteriorating as moving away from the center of this range (Malka and Lerner, 2004). A too small number of bins will smooth the estimated density and a too large number of bins will lead to the curse of dimensionality resulting in performance worsening in both cases. Besides, a too large number of bins will overload the calculation.

In parametric density estimation we assume a model describing the density and look for the optimal parameters for this model. For example, for a Gaussian model we ought estimating the data mean and variance. A single Gaussian estimation (SGE) is straightforward to implement and it bares almost no computational load to the NBC but its accuracy declines with the degree of deviation of the data from normality, which is expected in many real-world problems (John and Langley, 1995; Lerner, 2004). Extending parametric density estimation using Bayesian approaches (Heckerman, 1995), we update an a priori probability (e.g., Dirichlet prior) on the parameters using the likelihood for the data, thus combining prior and acquired knowledge jointly. However, when enough data is available (and the number of parameters is not too large) the likelihood in Bayesian estimation approaches quickly hammers the priors making these approaches somewhat redundant.

1.2. Non-parametric density estimation using kernels

Non-parametric methods of density estimation assume no model in hand generating the data but allow the data itself to determine the density. The most common non-parametric method is kernel density estimation (KDE) (Silverman, 1986) computing the density by a linear combination of S kernel functions K having width h that are allocated around each training data point x_t, $t = 1,\dots,S$. Based on these S points, the one-dimensional KDE $p_S(x)$ of $p(x)$ required in order to compute each of the class-conditional probability densities of the right hand side of (3) is

$$p_S(x) = \frac{1}{S \cdot h}\sum_{t=1}^{S} K\left(\frac{x - x_t}{h}\right), \tag{4}$$

$$\int_{-\infty}^{+\infty} K(u)\mathrm{d}u = 1 \text{ and } K(u) \geqslant 0 \quad \forall u \tag{5}$$

such that it is strongly pointwise consistent, i.e.,

$$\int |p_S(x) - p(x)|\mathrm{d}x \to 0 \quad \text{as } S \to \infty, \tag{6}$$

which means that in the limit, a posterior probability based on KDE (2) produces the Bayes' optimal classification error rate. A kernel commonly used in KDE is the standard Gaussian, which when used with a width of $h = 1/\sqrt{S}$ renders the KDE strongly pointwise consistent (John and Langley, 1995).

Fig. 2 demonstrates SGE and KDE in comparison to a histogram representation of the *Gamma-glutamyl transpeptidase* feature of the *liver-disorders* database of the UCI repository (Merz et al., 1997). The KDE tracks the histogram accurately while the SGE fails to reconstruct the histogram skewing toward the tail of the distribution. More evidence to the superiority of KDE to SGE for non-normal distributions in the context of the NBC can be found later in this paper and in John and Langley (1995) and Lerner (2004).

Although providing superior accuracy for the NBC, the KDE suffers from extensive

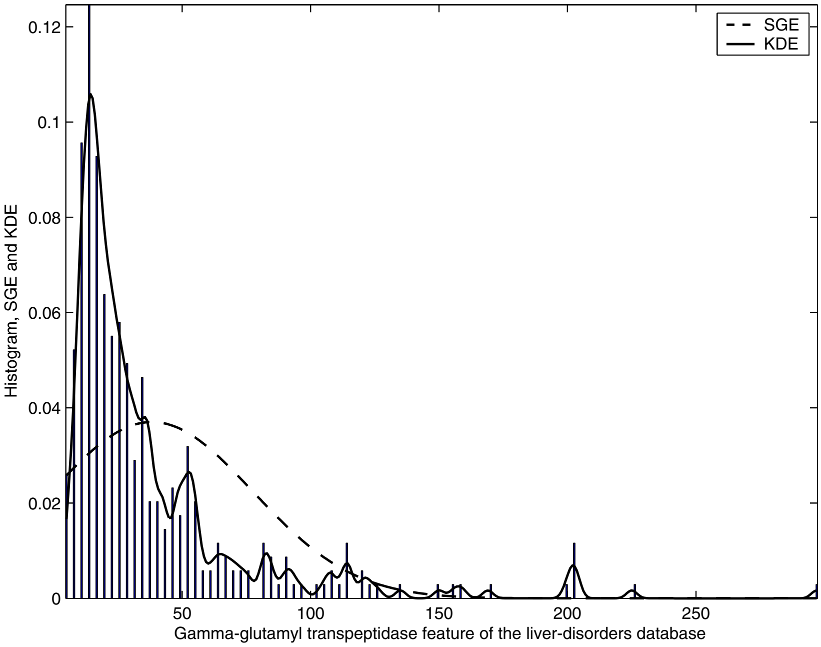

Fig. 2. SGE and KDE in comparison to a histogram representation of the *gamma-glutamyl transpeptidase* feature of the *liver-disorders* database.

computational cost limiting its implementation in real-world applications. Using KDE for the NBC, a class-conditional density for the ith variable, X_i, and kth class is computed for the mth test pattern, x_{im}^{tst}, using all training patterns x_{it}^{tr}

$$p(X_i = x_{im}^{tst}|C = k) = \frac{1}{N_{trk}} \sum_{t=1}^{N_{trk}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_{im}^{tst} - x_{it}^{tr})^2}{2\sigma^2}} \tag{7}$$

for a Gaussian kernel having a width σ around each of the N_{trk} training patterns of class k. Thus, the time complexity of estimating the likelihood employing KDE is $O(N_{ts} \cdot N_{tr} \cdot N_f \cdot N_d)$ for N_{ts} test patterns, N_{tr} training patterns, N_f features (variables) and N_d the number of calculations involved in computing a Gaussian, which for the common case $N_{tr} \gg N_c$ for N_c classes is much larger than $O(N_{ts} \cdot N_c \cdot N_f \cdot N_d)$ which is the complexity of SGE.

1.3. Related methodologies

To alleviate the complexity and enable fast implementation of non-parametric density estimation methods, a several approaches have been developed. Since usually many of the kernels are close to each other in feature space, binning (gridding) methods (Silverman, 1986; Jianqing and Marron, 1994; Gray and Moore, 2003) reduce the number of kernel evaluations by chopping each dimension into a number of intervals (bins), M, and representing all training points falling within an interval using a single kernel established employing all of these points. The problem is that M must be large to maintain precise estimation and the number of grid points increases as M^{N_f} (Gray and Moore, 2003). If however M is not large enough, the estimation may loose its accuracy. Silverman (1982, 1986) proposes an elaboration of binning using a fast Fourier transform performing discrete convolution to combine the grid counts and kernel weights. However, because a grid still underlies the method, it suffers from explosive scaling and error limitations (Gray and Moore, 2003). The fast Gauss transform (FGT) algorithm (Strain, 1991; Elgammal et al., 2003) expands the exponential of (7) using a Hermite series having a small number of terms around a small number of centers of 'boxes' clustering the training points. The fast multipole algorithm (FMA) (Greengard, 1988) relies on a spatial decomposition that separates the collection of patterns to regions. The effects of distant regions on test patterns are computed by the multipole expansion, and the effect of nearby regions is computed directly. Lambert et al. (1999) cast (7) using Taylor expansion to a specific order evaluating the approximation at a cost related to this order rather than the size of the training set. Hoti and Holmstrom (2004) transform the data using principal component analysis (PCA) to non-Gaussian and Gaussian data corresponding to the most and least significant PCA eigenvalues, respectively, and then apply density estimation only to the non-Gaussian part. This approach can relieve computational cost although the calculation of the non-Gaussian part of the data is still needed. Moore et al. (1997) suggest a tree in which each node summarizes the relevant statistics of all the data points below it in the tree. Using this multiresolution data structure saves the need to employ most of the training points increasing the speed of kernel regression.

Unfortunately, none of the approaches developed to alleviate KDE enabling fast implementation has ever been applied to BNs. Moreover, all of these methods aim at resolving the curse of dimensionality unnecessarily for the NBC decomposition (3). In this study, we propose a spline smoother to reduce the computational burden in KDE making probabilistic inference using the NBC feasible for real-world applications. Section 2 of the paper describes the spline smoother and its application to KDE for NBC. Section 3 outlines our experiments and their results for synthetic and real-world databases, while Section 4 concludes the paper.

2. Spline-approximated KDE for BNCs

Our approach differs from previous methods diminishing KDE computationally and relies on composing a spline from low-order polynomials each smoothes the density over a small interval resulting in the approximation of the whole density using very few coefficients.

2.1. The spline smoother

Splines have been used in many applications, such as medical (Wang and Amini, 2000), video segmentation (Precioso and Barlaud, 2002), image encoding and decoding (Wang et al., 2001) and moments of free-form surfaces (Soldea et al., 2002). Splines are smooth piecewise polynomial functions employed to approximate smooth functions locally (de Boor, 1978). The spline is used in a large interval for which a single approximation requires a polynomial of high degree that complicates the implementation and may overfit the data. Given the data $y(\delta_1),\ldots,y(\delta_P)$ with $a = \delta_1 < \cdots < \delta_j < \cdots < \delta_P = b$, we establish a piecewise interpolant f to y such that f agrees with low-degree polynomials $f_j(x)$ on sufficiently small intervals $[\delta_j, \delta_{j+1}]$, i.e.,

$$f(x) = f_j(x) \text{ for } \delta_j \leqslant x \leqslant \delta_{j+1}, \quad \forall j = 1,\ldots,P-1 \tag{8}$$

and the jth polynomial $f_j(x)$ coincides with y on the interval edges and its derivatives there satisfy some slope conditions set by the interpolation method being used. Using local polynomial coefficients a_{jl} derived from these slope conditions (de Boor, 1978), the polynomial of order N describing y within the jth interval is

$$f_j(x) = \sum_{l=1}^{N} (x-\delta_j)^{N-l} a_{jl}. \tag{9}$$

For example, a piecewise cubic function f agrees with y at $\delta_1,\ldots,\delta_P$, is continuous and has a continuous first derivative on $[a, b]$. It makes use of cubic polynomials ($N = 4$)

$$f_j(x) = (x-\delta_j)^3 a_{j1} + (x-\delta_j)^2 a_{j2} + (x-\delta_j) a_{j3} + a_{j4}. \tag{10}$$

Keeping some boundary conditions at $\delta_1, \ldots, \delta_P$ enables composition of these low order polynomials to a smooth piecewise polynomial function called a spline. Fig. 3 demonstrates such a composition of a spline from low-order polynomials. By approximating KDE using a spline instead of direct implementation we utilize those very few coefficients of the spline instead of the training set shaping KDE, thus eliminate the computational complexity of KDE facilitating classification using the NBC.

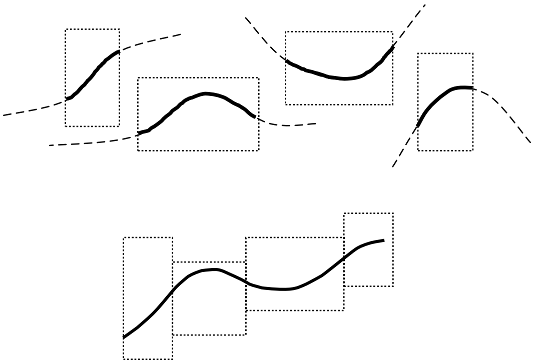

Fig. 3. A composition of a spline (bottom) from low-order polynomials (top) (Inspired by the Math Works MatLab documentation).

Fig. 4 shows an example in which a cubic spline smoother of KDE provides identical approximation to direct KDE for a section of the *weight percent of sodium in oxide* feature of the UCI Repository (Merz et al., 1997) *Glass* database (top). The figure also shows that the residual, i.e., the difference, between the two densities is negligible (bottom).

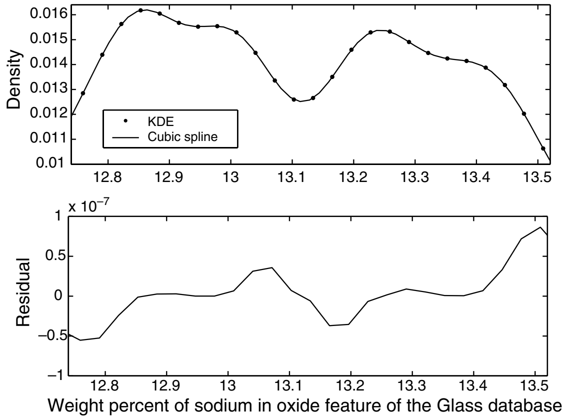

Fig. 4. Cubic spline smoother approximating KDE almost identicaly to direct KDE for a section of the *weight percent of sodium in oxide* feature of the *Glass* database (top), both having a negligible difference (Residual) (bottom).

2.2. Spline-approximated KDE

We suggest applying splines to KDE in order to ease probabilistic inference in NBCs. The spline smoother is applied during the test. After training, we compute for each of the $P-1$ consecutive intervals within the estimation range of each variable the N coefficients needed to approximate an Nth-order polynomial. We establish a $(P-1)\times N$ look-up-table (LUT) matrix, A, holding the a_{jl} coefficients, i.e., all the information needed for the estimation of this variable density. The value of N should be large enough to ensure satisfactory fitted curves, but not too large in order to avoid the curse-of-dimensionality and maintain the simple implementation using low order polynomials. During the test of the mth pattern represented by the ith variable, x_{im}^{tst}, we employ the N coefficients corresponding to the jth interval beginning at δ_j and coinciding with x_{im}^{tst} in order to evaluate the spline-based estimation for this test point

$$f_{ji}(x_{im}^{tst}) = \sum_{l=1}^{N}(x_{im}^{tst} - \delta_{ji})^{N-l} \cdot a_{jli}, \tag{11}$$

where a_{jli} is the lth spline coefficient of the jth interval of the ith variable.

Using spline-based KDE for the NBC, each class-conditional density of (3) for the ith variable and kth class is derived for the mth test pattern using (11) and N spline coefficients rather than using (7) and the whole training set. Thus, time complexity of estimating the likelihood employing spline-based approximation is $O(N_{ts}\cdot N_f\cdot N_c\cdot N_n)$ for N_{ts} test patterns, N_f features, N_c classes and N_n calculations involved in computing (11). Direct KDE has complexity of $O(N_{ts}\cdot N_f\cdot N_{tr}\cdot N_d)$ for N_{tr} training patterns and N_d calculations involved in a single Gaussian distribution in (7). Since N_d and N_n are of the same order the predominant difference in computational cost between the two estimation methods is attributed to the difference between N_{tr} and N_c where $N_{tr} \gg N_c$. Moreover for $N_d \sim N_n$, the complexity of spline-based KDE approximation is identical to that of SGE.

3. Experiments and results

3.1. Databases and methodology

We tested one synthetic and ten real-world databases with continuous features. The synthetic database has two classes and ten continuous features each having a several states sampled according to some a priori probability. Nine of the real-world databases are taken from the UCI repository, which is a well documented database (Merz et al., 1997). The remaining database is taken from a cytogenetic domain including more than 3000 patterns of four classes of signals represented using twelve features of size, shape, color and intensity (Lerner et al., 2001). In the experiments, we employed cross-validation (CV10) and hold-out (2/3 of the data for training) methodologies in databases having less and more than 3000 patterns, respectively. Patterns with missing values were deleted from the database. Table 1 summarizes important characteristics of the real-world databases. In addition, we chose for the

Table 1
Characteristics of the experimented real-world databases

Database	Number of classes	Number of features	Continuous/discrete features	Database size	Experiment methodology
Glass	7	9	9/0	214	CV10
Iris	3	4	4/0	150	CV10
Wine	3	13	13/0	178	CV10
Pima	2	8	8/0	768	CV10
Ionosphere	2	33	32/1	351	CV10
Letter	26	16	16/0	20,000	Hold-out
Adult	2	14	6/8	45,222	Hold-out
Liver disorders	2	6	6/0	345	CV10
Image	7	18	18/0	210	CV10
Cytogenetics	4	12	11/1	3144	Hold-out

KDE the standard Gaussian with a width of $h = 1/\sqrt{S}$ (Section 1.2).

3.2. Sensitivity to spline order

We investigated the influence of spline order on the estimation error and the NBC accuracy. For this purpose, we evaluated splines of orders $N = [1,4]$ approximating KDE of variables of each databases in comparison to direct KDE and SGE. Fig. 5 shows direct KDE and its 4th order spline approximation coinciding with each other for an example synthetic database feature. The figure also manifests the residuals between 1st and 4th order spline-based approximations and direct KDE. A similar experiment was performed with the *weight percent of sodium in oxide* feature of the UCI Repository *Glass* database. Fig. 6 reveals densities approximated by 1st and 4th order splines in comparison to direct KDE for a section of the density. The figure demonstrates the accuracy of the spline (especially cubic) approximating KDE. We also measured the mean squared error (MSE) between direct KDE and spline-based KDE approximation (i.e., the average residual),

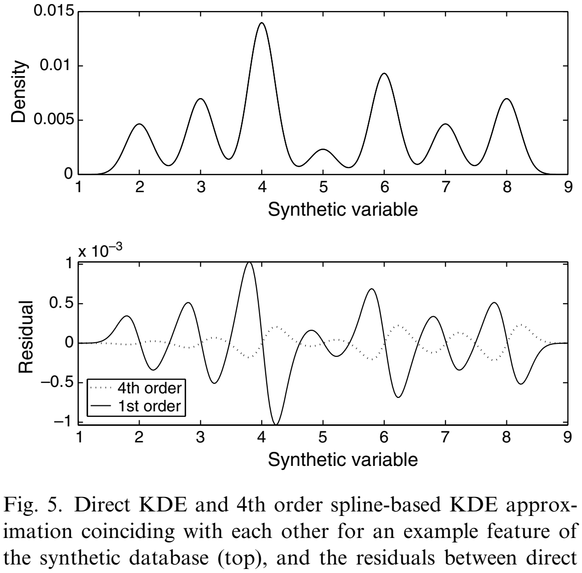

Fig. 5. Direct KDE and 4th order spline-based KDE approximation coinciding with each other for an example feature of the synthetic database (top), and the residuals between direct KDE and 1st and 4th order spline-based approximations for this feature (bottom).

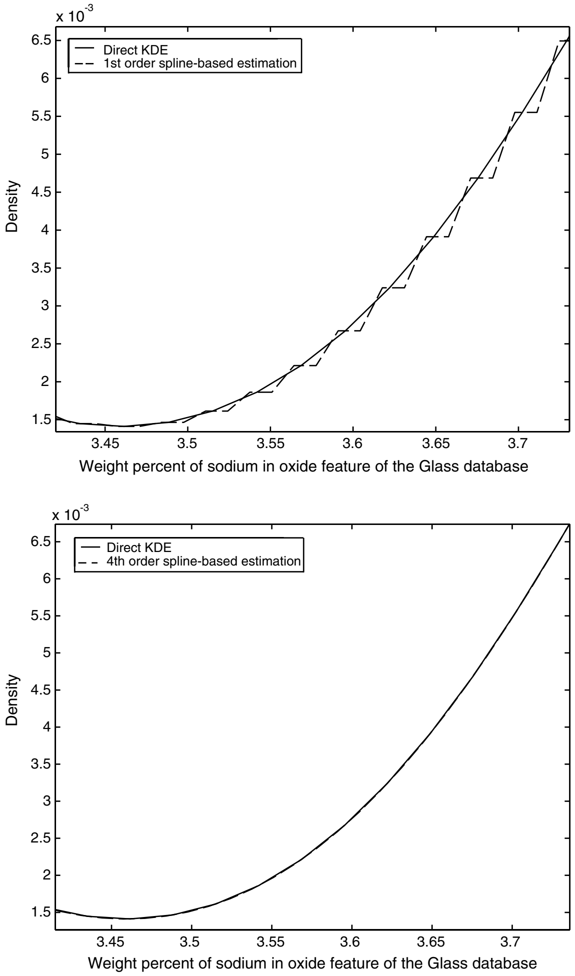

Fig. 6. 1st (top) and 4th (bottom) order spline-based KDE approximations for a section of the *weight percent of sodium in oxide* feature of the *Glass* database in comparison to direct KDE.

$$\mathrm{MSE} = \frac{1}{P}\sum_{j=1}^{P}[y(\delta_j) - f(\delta_j)]^2 \qquad (12)$$

for spline orders in [1,4]. As presented in Table 2, spline-based KDE approximation demonstrates a negligible MSE compared to direct KDE especially for orders greater than one.

Next, we conducted classification experiments on the real-world databases using the NBC employing SGE, KDE and spline-based KDE approximation. Table 3 demonstrates the

Table 2
The MSE between direct KDE and spline-based KDE approximation having orders in [1, 4] for the *weight percent of sodium in oxide* feature of the *Glass* database

Spline order	MSE ($\times 10^{-9}$)
1	146
2	10.6
3	9.99
4	9.15

Table 3
The NBC accuracy for different real-world databases when densities are based on 1st or 4th order spline KDE approximations in comparison to SGE[a]

Database	NBC classification accuracy (mean ± std) (%)		
	SGE	1st order spline	4th order spline
Glass	49.0 (±8.45)	40.7 (±7.37)	**65.5** (±11.19)
Iris	**96.0** (±4.42)	76.7 (±10.00)	95.3 (±5.21)
Wine	**97.7** (±2.76)	64.0 (±9.81)	95.5 (±4.17)
Pima	**76.0** (±4.89)	58.1 (±2.70)	69.4 (±3.67)
Ionosphere	82.9 (±3.42)	57.9 (±10.33)	**92.3** (±3.85)
Letter	65.5	73.3	**73.4**
Adult	82.6	79.5	**83.1**
Liver disorders	56.0 (±10.23)	50.4 (±7.24)	**64.3** (±5.66)
Image	62.9 (±8.73)	41.9 (±10.39)	**70.6** (±9.15)
Cytogenetics	67.5	41.0	**74.5**

[a] Accuracy based on 4th order spline KDE approximation is identical to that based on direct KDE. Bold font emphasizes the highest accuracy for a database.

superiority for most databases of 4th order spline in comparison to 1st order spline and SGE in approximating density for the NBC. Accuracy achieved using the 4th order spline is identical to that achieved using direct KDE. In three of the databases (Iris, Wine and Pima), feature distribution is close to normal thus SGE reaching asymptotic performance sooner than KDE (i.e., with a smaller sample size) yielding better accuracy than KDE and therefore better than the spline approximation. In those infrequent occasions of close to normal data distribution, the spline-based KDE approximation cannot ease the sample size sensitivity of KDE compared to SGE. However, in most real-world applications KDE and thus the suggested spline-based KDE approximation will outperform SGE leading to more accurate NBC performance.

3.3. Acceleration and sensitivity to sample size

We measured the acceleration (i.e., the ratio) in NBC run-time due to spline-based approximation with respect to direct KDE for increasing sample sizes. Table 4 shows the run-time (on an Intel P-II, 450 MHz processor with 192 MB RAM) using both techniques while classifying the synthetic database for sample sizes in the range [100–200 K] along with the corresponding accelerations. Fig. 7 demonstrates a sharper increase with sample size of the KDE run-time compared to that of the spline approximation as well as the acceleration achieved utilizing the latter. The change of slopes in both graphs is attributed to the switch of methodologies from CV to hold-out (Section 3.1), as each methodology employs different numbers of training and test patterns.

3.4. Sensitivity to dimensionality

Fig. 8 demonstrates the effect of increasing dimensionality on the NBC classification run-time when the classifier utilizes direct KDE in comparison to spline-based KDE approximation for 300 patterns of the synthetic database. The acceleration due to spline-based KDE approximation in comparison to direct KDE is constant for all dimensions (i.e., 54 for this database).

Table 4
The NBC run-time on the synthetic database for increasing sample sizes using direct KDE and 4th order spline-based KDE approximation, as well as the run-time acceleration achieved

Sample size	NBC run-time (s)		Run-time acceleration
	Direct KDE	Spline-based	
100	81	3.83	21
200	323	7.5	43
300	723	10.1	72
600	2899	19.8	146
1000	8070	36.9	219
2500	41,231	81	509
10,000	196,810	100	1968
50,000	4,897,232	513	9547
100,000	19,633,152	1041	18,863
200,000	77,459,336	2126	36,434

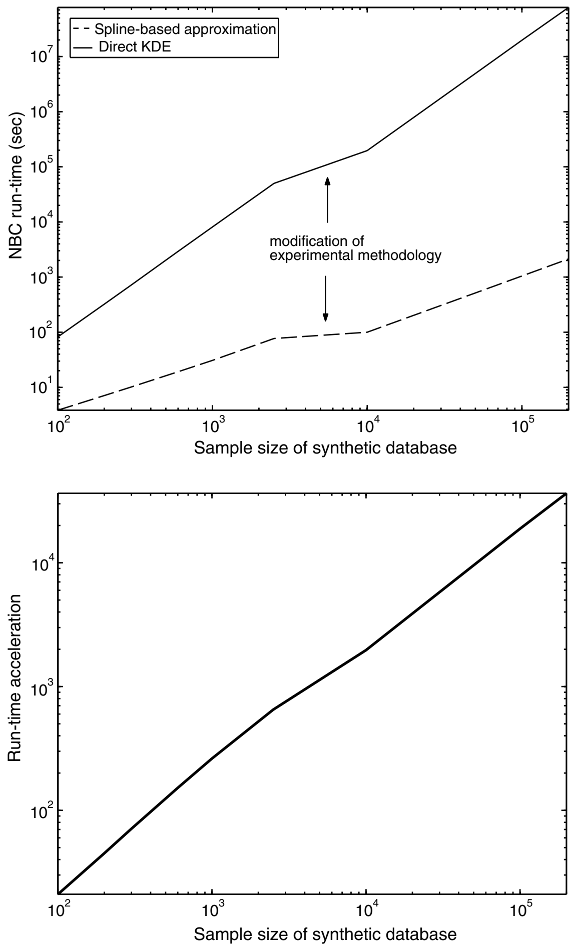

Fig. 7. The NBC run-time for KDE and 4th order spline-based KDE approximation (top), and accelerations due to the spline approximation (bottom) for increasing sample sizes of the synthetic database.

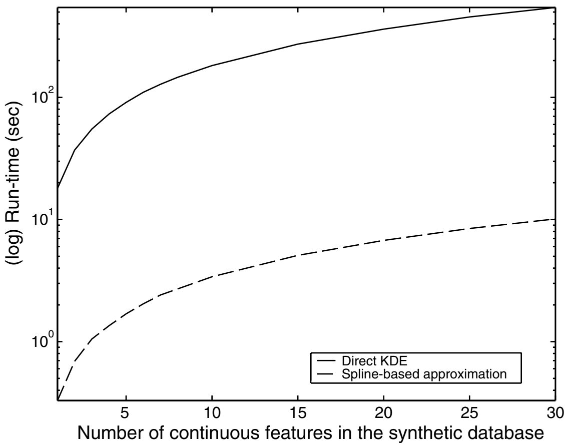

Fig. 8. The NBC classification run-time using direct KDE and spline-based KDE approximation for increasing dimensionality showing constant acceleration.

Table 5
The NBC run-times using a 4th order spline-based KDE approximation and direct KDE and the corresponding accelerations for a several real-world databases

Database	NBC run-time (s)		Run-time acceleration
	Direct KDE	Spline	
Glass	235	20	12
Iris	50	3.5	14
Wine	238	11.3	21
Pima	3103	19.3	161
Ionosphere	2120	40	53
Letter	841,510	4429	190
Adult	3,237,669	301	10,746
Liver disorders	530	7.3	73
Image	475	40	12
Cytogenetics	15,690	75	209

3.5. Accelerations for real-world databases

Experimenting with real-world databases of the UCI Repository and the cytogenetic domain, we compare in Table 5 run-times of the NBC employing direct KDE or a cubic spline KDE approximation as well as the corresponding acceleration achieved using the latter. For all databases we observe significant run-time acceleration spanning from 1 to 4 orders of magnitude, where large databases benefit the most pronounced acceleration. For example, classifying the *Adult* database having 45,222 patterns using direct KDE requires approximately 37 days compared to 5 min using the spline-based KDE approximation, leading to significant acceleration of more than 10^4. We note again that the NBC employing each of these two estimation methods achieves identical classification accuracy.

4. Discussion

Frequently, classification using BNCs within a domain having continuous variables requires density estimation. Non-parametric density estimation using kernels is accurate but computationally expensive since all training patterns participate in testing each unseen pattern, sometimes rendering the estimation impractical for real-world applications. We have presented a method based on a spline smoother approximating KDE that instead of using the training set utilizes the spline coefficients (only four in the case of a cubic spline), thus providing rapid evaluation of KDE. Moreover, spline-approximated KDE provides the KDE accuracy at the cost of SGE.

Classification experiments with an NBC on synthetic and real-world databases revealed increase with sample size of the acceleration achieved using the spline approximation compared to direct KDE. The experiments proved pronounced decrease of classification run-time sometimes by several orders of magnitude while preserving the predictive accuracy of the classifier, thereby making the suggested method practical for real-world applications. Although demonstrated for the NBC, the method is useful in reducing time complexity in other applications involving non-parametric density estimation. Finally, it is interesting to compare spline to other approximations of KDE.

Acknowledgement

This work was supported in part by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University, Beer-Sheva, Israel.

References

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

de Boor, C., 1978. A Practical Guide to Splines. Appl. Math. Sci., vol. 27. Springer-Verlag.

Elgammal, A., Duraiswami, R., Davis, L.S., 2003. Efficient kernel density estimation using the fast Gauss transform with applications to color modeling and tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25, 1499–1504.

Friedman, N., Goldszmidt, M., Lee, T.J., 1998. Bayesian network classification with continuous attributes: Getting the best of both discretization and parametric fitting. In: Proceedings of the 15th International Conference on Machine Learning, San Francisco, CA. Morgan Kaufmann, pp. 179–187.

Gray, G., Moore, A.W., 2003. Nonparametric density estimation: toward computational tractability. In: SIAM International Conference on Data Mining.

Greengard, L., 1988. The Rapid Evaluation of Potential Fields in Particle Systems. MIT Press, Cambridge, MA.

Heckerman, D., 1995. A tutorial on learning Bayesian networks. Microsoft Research Technical Report. MSR-TR-95-06.

Hoti, F., Holmstrom, L., 2004. A semi-parametric density estimation approach to pattern classification. Pattern Recognition Lett. 37, 409–419.

Jianqing, F., Marron, J.S., 1994. Fast implementation of nonparametric curve estimators. J. Comput. Graphical Statist. 3, 35–57.

John, G.H., Langley, P., 1995. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers, pp. 338–345.

Lambert, C.G., Harrington, S.E., Harvey, C.R., Glodjo, A., 1999. Efficient online nonparametric kernel density estimation. Algorithmica 25, 37–57.

Lerner, B., 2004. Bayesian fluorescence in-situ hybridization signal classification. Artificial Intelligence in Medicine 30, 301–316 (A special issue on Bayesian Models in Medicine).

Lerner, B., Clocksin, W.F., Dhanjal, S., Hult'en, M.A., Bishop, C.M., 2001. Feature representation and signal classification in fluorescence in-situ hybridization image analysis. IEEE Trans. Syst. Man Cybernet. A 31, 655–665.

Malka, R., Lerner, B., 2004. Classification of fluorescence in situ hybridization images using belief networks. Pattern Recognition Lett. 25, 1777–1785.

Merz, C., Murphy, P., Aha, D., 1997. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine. Available from: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Moore, A.W., Schneider, J., Deng, K., 1997. Efficient locally weighted polynomial regression predictions. In: Proceedings of the International Conference on Machine Learning, pp. 236–244.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan-Kaufman.

Precioso, F., Barlaud, M., 2002. B-spline active contour with handling of topology changes for fast video segmentation. J. Appl. Signal Process. 6, 555–560.

Silverman, B.W., 1982. Kernel density estimation using the fast Fourier transform. J. Roy. Statist. Soc. Ser. C: Appl. Statist. 33, 93–97.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. Chapman and Hall/CRC.

Soldea, O., Elber, G., Rivlin, E., 2002. Exact and efficient computation of moments of free-form surface and trivariate based geometry. Computer-Aided Des. 34, 529–539.

Strain, J., 1991. The fast Gauss transform with variable scales. SIAM J. Scientific Statist. Comput. 12, 1131–1139.

Wang, L.J., Hsieh, W.S., Truong, T.K., Reed, I.S., Cheng, T.C., 2001. A fast efficient computation of cubic-spline interpolation in image codec. IEEE Trans. Signal Process. 6, 1189–1197.

Wang, Y.P., Amini, A.A., 2000. Fast computation of tagged MRI motion fields with subspaces. In: Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 119–126.

Yang, Y., Webb, G.I., 2002. A comparative study of discretization methods for naïve Bayes classifiers. In: Proceedings of PKAW, The 2002 Pacific Rim Knowledge Acquisition Workshop, Tokyo, Japan, 159–173.