Rapid Spline-based Kernel Density Estimation for Bayesian Networks

Yaniv Gurwicz and Boaz Lerner*
Pattern Analysis and Machine Learning Lab
Department of Electrical & Computer Engineering
Ben-Gurion University, Israel
yanivg@ee.bgu.ac.il; boaz@ee.bgu.ac.il

Abstract

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

1. Introduction

A Bayesian network (BN) represents the joint probability distribution $p(\boldsymbol{X})$ over a set of n domain variables $\boldsymbol{X}=\{X_1,\ldots,X_n\}$ graphically. A connection between variable (graph node) X_i and its parents $\boldsymbol{Pa_i}$ in the graph is quantified probabilistically using the data. By ordering the variables topologically, extracting the general factorization of this ordering and applying the directed Markov property we can decompose the joint probability distribution

$$p(C \mid \boldsymbol{X}) = \frac{p(\boldsymbol{X},C)}{p(\boldsymbol{X})} = \frac{p(\boldsymbol{X} \mid C)\cdot p(C)}{p(\boldsymbol{X})}. \quad (1)$$

The naïve Bayesian classifier (NBC) is a BN predicting a class C for a pattern $\mathbf{x}$ using Bayes' theorem

$$P(C \mid \boldsymbol{X}=\mathbf{x}) = \frac{p(\boldsymbol{X}=\mathbf{x} \mid C)\cdot P(C)}{p(\boldsymbol{X}=\mathbf{x})}, \quad (2)$$

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

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

This likelihood is obtained through the computation of each of the local densities, which for a continuous variable requires either discretization or estimation using parametric, non-parametric or semi-parametric methods. The most common non-parametric method is kernel density estimation (KDE) [1], allowing the data to determine the estimation without making any prior assumptions about the distribution. Although usually more accurate for the NBC than other density estimation methods [2, 3], the KDE suffers from extensive computational cost limiting its implementation for real-world applications.

Using KDE for the NBC, each class-conditional density for the ith variable and kth class is computed for the mth test pattern x_{im}^{tst} using all training patterns x_{it}^{tr}

$$p(X_i = x_{im}^{tst} \mid C=k) = \frac{1}{N_{trk}} \sum_{t=1}^{N_{trk}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_{im}^{tst}-x_{it}^{tr})^2}{2\sigma^2}} \quad (4)$$

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 $O(N_{ts}\cdot N_{tr}\cdot N_f)$ for N_{ts} test patterns, N_{tr} training patterns and N_f features (variables), which for the common case $N_{tr}>>N_c$ for N_c classes is much larger than $O(N_{ts}\cdot N_f\cdot N_c)$, which is the complexity of a parametric method assuming a single Gaussian for each class.

To alleviate the complexity and enable fast implementation of KDE, several approaches such as binning [4], FFT gridding [1] and FMA [5] have been developed, however none of them has never been applied

* Corresponding author

to the field of BNs. Moreover, all of these methods aim at resolving the curse of dimensionality unnecessarily for the NBC decomposition in (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 describes the spline smoother and spline-based KDE for the NBC. Section 3 outlines our experimental layout and results while Section 4 concludes the paper with a discussion.

2. Spline-based KDE

2.1. The Spline Smoother

Splines are smooth piecewise polynomial functions employed to approximate smooth functions locally [6]. 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 density $y(\delta_1),\ldots,y(\delta_p)$ at $a=\delta_1<\ldots<\delta_j<\ldots<\delta_p=b$, 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.,

$$f(x) = f_j(x) \text{ for } \delta_j \le x \le \delta_{j+1},\ \forall j = 1,\ldots,P-1 \tag{5}$$

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 [6], the polynomial of order N describing y within the jth interval is

$$f_j(x) = \sum_{l=1}^{N} (x-\delta_j)^{N-l} a_{jl}. \tag{6}$$

Fig. 1 shows an example in which a cubic spline smoother of KDE provides identical approximation to direct implementation of KDE for a section of the *weight percent of sodium in oxide* feature of the UCI Repository [7] *Glass* database (top). A negligible difference (residual) between the two densities is shown (bottom).

The use of a spline to generate high-order mathematical extensions of functions has been shown in several fields like moments of free-form surfaces, video segmentation, image encoding and decoding and medical applications. We apply the spline to interpolate and represent densities derived by KDE using low-order polynomials having few coefficients, thereby avoiding the computational complexity of the kernel method.

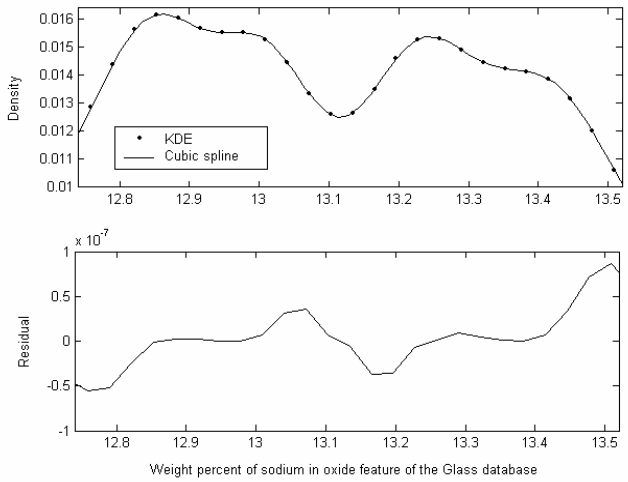

Figure 1: Cubic spline smoother approximating KDE almost identically to direct KDE for a section of the weight percent of sodium in oxide feature of the Glass database (top). Both densities are having a negligible difference (residual) (bottom).

2.2. Spline-based KDE for NBC

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 $(P\text{-}1)\times N$ look-up-table matrix, A, holding all the a_{jl} coefficients, i.e., all the information needed for the estimation of this variable. 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

$$f_j(x_{im}^{tst}) = \sum_{l=1}^{N} (x_{im}^{tst} - \delta_{ji})^{N-l} \cdot a_{jli} \tag{7}$$

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 (7) and N spline coefficients rather than using (4) 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 (7). 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 (4). 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 $N_{tr} >> N_c$.

3. Experimental Results

3.1. Experimental methodology

We have 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 selected according to some a priori probability. Nine of the real-world databases are taken from the UCI repository [7]. The remaining database is taken from a cytogenetic domain including more than 3,000 patterns of four classes of signals represented using twelve features of size, shape, color and intensity [3]. In the experiments, we employed cross validation (CV10) and hold-out (2/3 of the data for training) methodologies for databases having less and more than 3,000 patterns respectively.

3.2. Acceleration and sensitivity to sample size

We have 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. Figure 2 and Figure 3 show respectively the run-time using both techniques while classifying the synthetic database for sample sizes in the range [100,200K] as well as the corresponding accelerations. The figures demonstrate respectively, a sharper increase of run-time due to KDE compared to cubic spline-based KDE and acceleration increase with sample size.

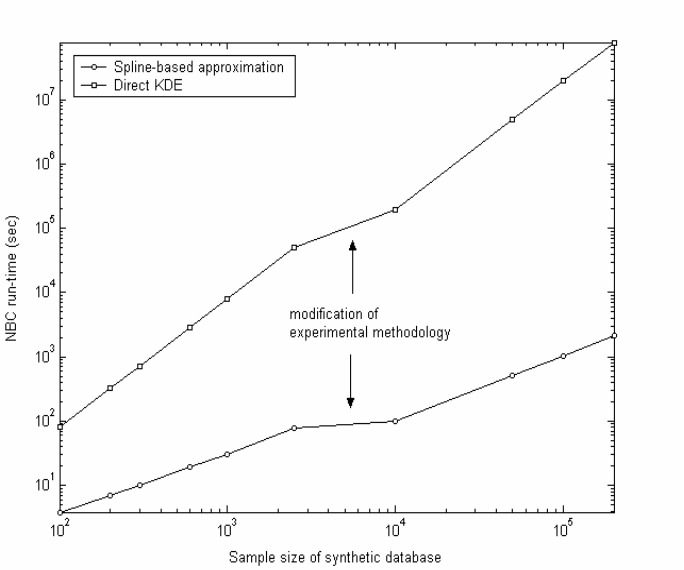

Figure 2: NBC run-time for KDE and 4th order spline-based KDE.

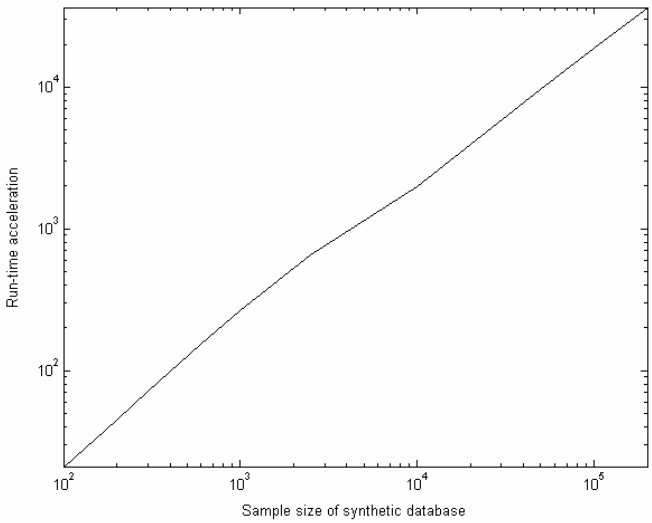

Figure 3: Acceleration due to the spline for increasing sample sizes of the synthetic database.

3.3. Acceleration for real-world databases

Experimenting with real-world databases of the UCI Repository and the cytogenetic domain, we compare in Table 1 run-times of the NBC employing direct KDE and a cubic spline-based KDE 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 on a standard PC compared to 5 minutes using the spline-based KDE, leading to significant acceleration of more than 10^4. It is important to mention that the NBC employing these two estimation methods achieves identical classification accuracy.

4. Discussion

Usually, classification using BNs 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 computation impractical for real-world applications. We presented a method based on a spline smoother for 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. 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 our 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.

Table 1: NBC run-time using a 4th order spline-based KDE and direct KDE and the corresponding acceleration for several real-world databases of the UCI Repository.

Dataset	NBC Run-Time (sec)		Run-Time Acceleration
	Direct	Spline	
Glass	235	20	12
Iris	50	3.5	14
Wine	238	11.3	21
Pima	3,103	19.3	161
Ionosphere	2,120	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

Acknowledgment

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

5. References

[1] B. W. Silverman. "Kernel Density Estimation Using the Fast Fourier Transform". *Journal of the Royal Statistical Society Series C: Applied Statistics*, Vol. 33, 1982.

[2] G. H. John and P. Langley. "Estimating Continuous Distributions in Bayesian Classifiers". *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995.

[3] B. Lerner. "Bayesian Fluorescence In-Situ Hybridization Signal Classification". *Artificial Intelligence in Medicine, special issue on Bayesian Models in Medicine,* Vol. 30, pp. 301-316, 2004.

[4] G. Gray and A.W. Moore, "Nonparametric Density Estimation: Toward Computational Tractability". *SIAM International Conference on Data Mining*, January 2003.

[5] L. Greaangard. "The Rapid Evaluation of potential Fields in Particle Systems". *MIT Press, Cambridge, MA*, 1988.

[6] de Boor, C. A practical guide to splines. Applied Mathematical Sciences 27, Springer-Verlag, 1978.

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