Incremental Supervised Classification for the MTE distribution: A Preliminary Study*

Ildikó Flesch
Institute for Information and
Computer Sciences
Radboud University Nijmegen
Toernooiveld 1
6525ED Nijmegen, The Netherlands
ildiko@cs.ru.nl

Antonio Fernández and Antonio Salmerón
Dept. of Statistics and
Applied Mathematics
University of Almería
Carrera de Sacramento s/n
04120 Almería
{afalvarez,antonio.salmeron}@ual.es

Abstract

In this paper we propose an incremental method for building classifiers in domains with very large amounts of data or for data streams. The method is based on the use of mixtures of truncated exponentials, so that continuous and discrete variables can be handled simultaneously.

1 Introduction

In the last years, Bayesian networks [1, 8] have become a popular tool to solve *classification problems*, where the goal is to obtain a model able to assign a class label to an individual described in terms of an observed set of random variables [4], also called *features*. Classification is said to be *supervised* when the training data includes the value of the class for each data item, and *unsupervised* otherwise. In this work we are concerned with supervised classification.

One of the most successful classification models based on Bayesian networks is the so-called *naive Bayes* [2], where the features are assumed to be independent given the class variable.

*This work has been supported by the Spanish Ministry of Education and Science under project TIN2004-06204-C03-01 and by Junta de Andalucía under project P05-TIC-00276.

The use of Bayesian networks has also been extended to *regression* problems, formulated similarly to classification, with the difference that the class variable is continuous [3, 7, 12].

In probabilistic networks, incremental algorithms [4, 5, 6] are introduced to cope with many problem domains that involve large amounts of data. Exploiting the nature of incremental learning algorithms, we are able to learn both the structure and probability distribution of Bayesian networks stepwise, where at each step we update our present knowledge with new knowledge obtained by learning from new data. Note that two different sets of data according to the same probabilistic problem may consists of different knowledge about the probabilistic model.

The aim of this paper is to introduce incremental models for classification and regression, where discrete and continuous variables can appear either as class or as feature. We rely on the MTE (Mixtures of Truncated Exponentials) model [11], which appropriately fits the Bayesian network framework.

The rest of the paper is organised as follows. We give the basic definitions in section 2. The incremental learning problem is considered in section 3, where we describe our proposal. Section 5 is devoted to the experimental analysis and the paper ends with conclusions in section 6.

2 Preliminaries

A *Bayesian network* is a graphical representation of a probabilistic problem, formally defined as a pair $\mathcal{B} = (G, P)$, where P is the joint probability distribution on the set of random variables X_V, and G is an acyclic directed graph representing the dependence and independence relations among this set of random variables X_V satisfying the condition that each graphically represented marginal or conditional independence is valid also in the joint probability distribution [13].

The MTE model [11] is formally defined as follows:

Definition 1 *(**MTE potential**) Let X_V be a set of random variables. Let $Y = \{Y_1, Y_2, \ldots, Y_d\}$ and $Z = \{Z_1, Z_2, \ldots, Z_c\}$ be the set of discrete and continuous random variables, respectively, with $V = d + c$. We say that a function $f : \Omega_{X_V} \rightarrow \mathcal{R}_0^+$ is a* Mixture of Truncated Exponentials *potential (MTE potential) if one of the next conditions holds:*

- *$Y = \varnothing$ and f can be written as:*

$$f(x) = f(z) = a_0 + \sum_{i=1}^{m} a_i \exp\{\sum_{j=1}^{c} b_i^{(j)} x_j\}$$

  for all $z \in \Omega_z$, where a_i, $i = 0, \ldots, m$ and $b_i^{(j)}$, $i = 1, \ldots, c$ are real numbers;

- *$Y = \varnothing$ and there is a partition $D_1, D_2, \ldots, D_k$ of Ω_z into hypercubes such that f is defined as*

$$f(x) = f(z) = f_i(z) \text{ if } z \in D_i \ ,$$

  where each f_i, $i = 1, \ldots, k$ can be written in the form of equation 1.

- *$Y \neq \varnothing$ and for each fixed value $y \in \Omega_Y$, $f_y(z) = f(y, z)$ can be defined in the second condition.*

An MTE potential f is an MTE density if it integrates up to 1.

3 The theory of incremental supervised classification

3.1 The k-step incremental classification

Incremental learning algorithms are applied when (i) we may obtain new data for our problem domain to learn, (ii) the size of the given data is too large in a computational sense. In incremental processes we learn our model in steps, where at each step we only consider the new set of data. One step may mean either a time step implying that at certain time period we obtain new data for updating the representation of the system or we separate the data into disjoint subsets and, then, each step indicates one subset of the data to be learned. This is established in the following definition.

Definition 2 *(**k-step incremental approach**) If an incremental approach learning a probabilistic model is updated in k steps using k data sets, it is called a* k-step incremental approach.

Note that in a k-step incremental approach, k sets of data are used, which may be related to either time steps or separations of the entire data, or both.

Definition 3 *(**k-step incremental classification**) If the classification procedure is learned by a k-step incremental approach, it is called a* k-step incremental classification.

In this paper we focus on supervised classification and its incremental version, i.e. *k-step incremental supervised classification.*

Recall that Bayesian networks consist of two parts: (i) a graphical representation and (ii) a joint probability distribution. Therefore, since they both are related to the k-step incremental supervised classification, we need to discuss both parts in detail.

Regarding the graphical representation, we propose that a k-step incremental supervised classification model consists of k classification models. Here, the i-th classification model with $i \leq k$ represents the graphical representation of the learned model related to the i-th set of data. However, given a data set to

classify, classification models learned from different sets of data may assign a different class label to this data. Therefore, we also need a node that represents the class label of the entire k-step incremental supervised classification model, explaining the introduction of the so-called *main classifier*, denoted by Cl_M. The main classifier node plays the following role in this representation: it decides which class label has to be assigned to the data depending on the class labels at each classification model learned in the incremental approach. There are many possible structures to represent the relation between the main classifier and the incrementally learned classification models. In this paper, considering the fact that the main classifier depends on each learned classification model and therefore on each class label, we analyse the following two structures: (i) the main classifier acts as the common child of the k class variables of the k models, and (ii) the main classifier is the common parent of the k class variables.

To define the specific Bayesian network that represents the incremental supervised classification model, we also need to discuss the corresponding joint probability distribution. This will be done subsequently. Recall that in our acyclic directed graph, to represent each new knowledge in an exact way, the learned classification graph was inserted into the graph. Therefore, some classifiers may represent a relation of two or more random variables as a dependence, whereas other classifiers define this relation as an independence. To represent this precisely, at each new learned data set, we label the set of random variables related to the new data set by its step number, and insert it into the combined set of random variables. Moreover, we also need to add a random variable that represents the main classifier of our approach, already described in the graphical representation. For the formal definition, we need the following notation. Let for the k-step incremental classification X_{V_i} be the set of random variable of the i-th classification model and related to the data at the i-th step. Furthermore, the main classifier random variable is defined as X_{Cl_M}. Now, we can define the entire set of random variables and the related probability distribution as $X_V = X_{V_1} \cup X_{V_2} \cup \ldots \cup X_{V_k} \cup X_{\mathrm{Cl}_M}$, and P denotes the joint probability distribution related to X_V.

3.2 Model 1: The main classifier acting as a *child*

The class label defined by the main classifier depends on the k class labels in the graphical representation. This dependence can be represented in a way, where the main classifier acts as the parent of the k classifiers. To express this formally, we need the following notations. Let for the k-step incremental classification $G_i = (V_i, A_i)$ be the graph of the i-th classification model that is learned of the data at the i-th step. A *k-step incremental supervised classification graph* $G = (V, A)$ can be constructed in this way:

- the set of vertices is equal to the union of the vertices of the k classification models and the main classifier, formally, $V = V_1 \cup V_2 \cup \ldots \cup V_k \cup \mathrm{Cl}_M$, where V_i denotes the set of vertices of the ith classifier, $1 \leq i \leq k$ and Cl_M denotes the main classifier node;
- the set of arcs is equal to the union of the arcs of the k classification models and the arcs connecting the k classifiers with the main classification node all directing to the main classification node, formally $A = A_1 \cup A_2 \cup \ldots \cup A_k \cup \{(A_1, \mathrm{Cl}_M), (A_2, \mathrm{Cl}_M), \ldots, (A_k, \mathrm{Cl}_M)\}$.

According to the definition above, the k classification models remain marginally independent, which is consistent with the fact that they are learned from different sets of data. However, if the classification label of the main classifier is known, the classifier vertices of the classification models become dependent on each other, since the main classifier depends on the values of the k class labels all representing the learned knowledge of the k data sets.

The joint distribution P for a k-step incremental supervised classifier for model 1, defined for a set of random variables X_V, factorises as

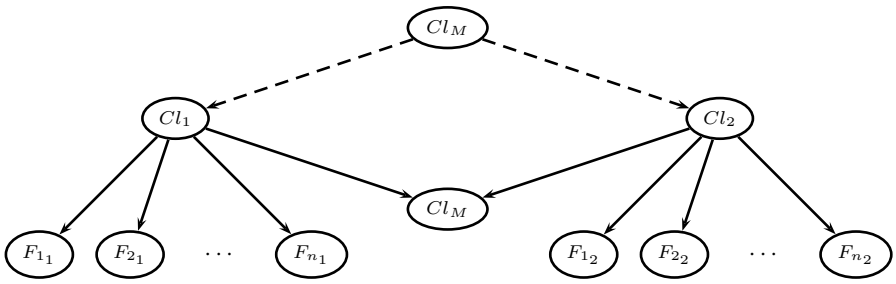

Figure 1: Model 1 of the 2-step incremental classification model for naive Bayesian classification. Model 2 is the same except for Cl_M, which is as indicated by the dashed lines.

$$P(X_V) = P(X_{V_1}, X_{V_2}, \ldots, X_{V_k}, X_{\mathrm{Cl}_M}) = P(\mathrm{Cl}_M \mid \mathrm{Cl}_1, \mathrm{Cl}_2, \ldots, \mathrm{Cl}_k) \prod_{i=1}^{k} P(X_{V_i}) \ .$$

Note that if we want to reason in model 1, we need to learn the parameters $P(\mathrm{Cl}_M \mid \mathrm{Cl}_1, \mathrm{Cl}_2, \ldots, \mathrm{Cl}_k)$. The way of doing it is explained in Section 4.

3.3 Model 2: The main classifier acting as a *parent*

In this section, the model for incremental supervised classification is discussed, when the main classifier has been chosen as the parent node of the k classifiers. In the graphical representation of model 2, the class variable of each one of the k models are independent of each other given the main class variable. This assumption is compensated with the reduction in the number of parameters that have to be learnt from the data.

The joint distribution P for the case of model 2 factorises as

$$P(X_V) = P(X_{V_1}, X_{V_2}, \ldots, X_{V_k}, X_{\mathrm{Cl}_M}) = P(\mathrm{Cl}_M) \prod_{i=1}^{k} P(X_{V_i}) \ .$$

We would like to emphasize that according to model 1 which is the correct model to represent incremental approach to supervised classification, model 2 is introduced in this paper due to the following reasons: (i) it is easier to learn parameter $P(\mathrm{Cl}_M)$ for model 2 than $P(\mathrm{Cl}_M \mid \mathrm{Cl}_1, \mathrm{Cl}_2, \ldots, \mathrm{Cl}_k)$ for model 1, since learning $P(\mathrm{Cl}_M)$ requires less data, and (ii) we think it is interesting to compare the behaviours of model 1 and model 2 with each other.

3.4 Incremental naive Bayesian classification models

For the sake of simplicity, we have assumed in this work that each one of the k classification models is a naive Bayes. Let us denote the feature variables as $F = \{F_1, F_2, \ldots, F_n\}$. The entire set of random variables is $X_V = \mathrm{Cl} \cup F$.

Naive Bayesian classification models have a straightforward graphical structure, hence their name, because they make a very strong independence assumption, namely, the feature variable are conditionally independent on each other. This assumption in the graphical representation can be expressed as $F_i \perp \perp_G F_j \mid \mathrm{Cl}, i \neq j$, whereas probabilistically, $P(\mathrm{Cl}, F_1, F_2, \ldots, F_n) = P(\mathrm{Cl}) \prod_{i=1}^{n} P(F_i \mid \mathrm{Cl})$.

Naive Bayesian classification models satisfying these independence assumptions have the advantage that by their graphical structure probabilistic reasoning is computationally easy to execute. To start, firstly the

cause-effect parameters $P(F_i \mid \mathrm{Cl})$ with $i \in \{1, 2, \ldots, n\}$ are learned. Secondly, the classification process computes probability $P(\mathrm{Cl} \mid F_1, F_2, \ldots, F_n)$, and assigns to a data the class label that has the highest a-posteriori probability.

The (in)dependence relations of naive Bayesian models have to be included in the incremental models. The graphical representation of a 2-step incremental naive Bayesian classification for model 1 and model 2 are shown in figure 1.

4 The learning and classification algorithms

In this section, we provide the learning and propagation algorithms of k-step incremental supervised classification models.

The pseudo code of the learning algorithm is given in Algorithm 1, for which, according to the definition of the k-step incremental classification, there are k data sets as inputs. Note that this learning algorithm consists of four main parts. In the first part, the incremental Bayesian classification model is initialised by setting it equal to the Bayesian classification model learned from the first data set, and inserting the main classifier into the model augmented with a dependence relation to the classifier in the first Bayesian classification model. In the second part, we learn the Bayesian networks related to the remaining $k-1$ sets of data and insert these learned models into the incremental model. Subsequently, in the third part, we build the database of the incremental search. This database is necessary, since it is used as the training set for the parameters of the main classifier which parameters are unknown yet. A sample of this database is constructed as follows. We take a sample of an input data set, then compute each classifier value in the k models having given the feature values of this sample, whereas we set the main classifier equal to the value of the classifier in the sample. Doing so, we know that the value of the main classifier is correct. Furthermore, if there is some lack of information in any of the k learned models, the assigned class la-

Algorithm 1: Incremental learning algorithm

Input: Sets of data $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$
Output: The incremental model

Learn Bayesian network $\mathcal{B}_1$ of dataset $\mathcal{D}_1$;
Initialise the incremental Bayesian network $\mathcal{B}$ as $\mathcal{B} = \mathcal{B}_1$;
Insert the main classifier Cl_M into $\mathcal{B}$;
Insert an arc between Cl_M and the classification vertex Cl_1 with direction depending on the choice of the model;
for $i = 2$ *to* k **do**
    Learn Bayesian classifier $\mathcal{B}_i$ of dataset $\mathcal{D}_i$;
    Rename the set of random variables $\{F_1, F_2, \ldots, F_n, \mathrm{Cl}\}$ in Bayesian network $\mathcal{B}_i$ into $\{F_{1_i}, F_{2_i}, \ldots, F_{n_i}, \mathrm{Cl_i}\}$;
    Insert $\mathcal{B}_i$ into $\mathcal{B}$;
    Insert an arc between Cl_M and Cl_i with direction depending on the choice of the model;
end
Initialise dataset $\mathcal{D}$ with elements $\{F_{1_1}, \ldots, F_{n_1}, \mathrm{Cl}_1, \ldots, F_{1_k}, \ldots, F_{n_k}, \mathrm{Cl}_k, \mathrm{Cl}_M\}$;
foreach *sample* d_i *in* $\mathcal{D}_i$, $1 \leq i \leq k$, *insert a new sample into database* $\mathcal{D}$ *filling the elements as follows* **do**
    forall *feature* $F_{j_i} \in d_i$, $1 \leq j \leq n$ **do**
        for $m = 1$ *to* k **do**
            $F_{j_m} = F_{j_i}$;
        end
    end
    if $(i = m)$ **then**
        $\mathrm{Cl}_m = \mathrm{Cl}_i$, $\mathrm{Cl}_i \in d_i$;
    else
        propagate value $\mathrm{Cl}_m = \mathrm{cl}_m$ in $\mathcal{B}_m$ given evidence $\{F_{1_m} = F_{1_i}, F_{2_m} = F_{2_i}, \ldots, F_{n_m} = F_{n_i}\}$;
    end
    set $\mathrm{Cl}_M = \mathrm{Cl}_i$;
end
Learn the parameters for Cl_M from $\mathcal{D}$;

bel for the feature variables of this sample can differ from the correct value. This difference provides the opportunity to learn the parameters for the main classifier that is done in the fourth part of the algorithm.

Algorithm 2: Incremental classification algorithm

Input: The incremental classifier $\mathcal{B}$ with the set of random variables $X_V = \{F_{1_1}, \ldots, F_{n_1}, \mathrm{Cl}_1, \ldots, F_{1_k}, \ldots, F_{n_k}, \mathrm{Cl}_k, \mathrm{Cl}_M\}$, and the set of observations $\{F_1, \ldots, F_n\}$.

Output: The assigned class label $\mathrm{Cl}_M = \mathrm{cl}_M$ for the set of observations in the input.

```
for i = 1 to k do
    for j = 1 to n do
        set F_{j_i} = F_j,
        F_j ∈ {F_1, ..., F_n};
    end
end
Compute Cl_M = cl_M by propagating evidence {F_{1_1}, ..., F_{n_1}, ..., F_{1_k}, ..., F_{n_k}} in Bayesian network B.
```

The classification algorithm is given in Algorithm 2. Here, the set of observed random variables is equal to the set of feature variables of a model. In the algorithm, these feature variables are filled in as evidence for the related feature variables at each classifier, and, subsequently, we assign a class label to the main classifier applying a propagation algorithm.

We would like to emphasize that the algorithms introduced in this section are not restricted to naive Bayesian classifiers but they are also applicable to other classifier models.

5 Experimental results

In this section, we discuss the results of our experiments that are carried out on sets of random variables that are either only discrete (see tables 1 to 4) or only continuous (see tables 5 to 6). We would like to note that the experiments are only executed for 2-step incremental naive Bayesian classifiers. In both cases we have used toy networks and databases sampled from them. We have considered two settings: *not divided knowledge* and *divided knowledge*. Not divided knowledge means that two train databases are sampled at random, and therefore follow the same model, whilst divided knowledge means that the two databases are sampled with some restrictions, in order to force them to contain distinct information. We believe that the later approach imitates the situation in which the data comes from a continuous stream, where the underlying distribution may change, or even if the model is the same, the amount of data required to properly recovering it is huge. The results reported in tables 1 to 4 for the discrete case, show the classification accuracy for the two initial classifiers, the classifier obtained by merging the two databases, and the incremental classifier. It can be seen that the incremental classifier is never worse than the individual ones, and often it is even competitive with the global classifier. The results reported in tables 5 to 6 for the continuous class framework, also show that the incremental regression model behaves intermediately between the two initial models and the global one, as it was intuitively to be expected. The accuracy is measured in terms of the root mean squared error. The columns mean and median indicate whether we use the median or the mean of the posterior distribution of the class variable to predict [12].

6 Conclusion

In this paper we have proposed a method for constructing incremental classification models able to deal with discrete and continuous variables. The preliminary results shown in section 5 show that the proposed models behave reasonably well with the toy examples. We think that the incremental approach is specially interesting for the case of the MTE distribution, where it is not possible (at least given the state-of-the-art) to keep the suffi-

nr. db1	nr. db2	acc 1	acc. 2	acc. all	acc. Incr
250	250	**86.6**	83.3	**86.6**	**86.6**
200	200	80	**83.3**	**83.3**	**83.3**
100	100	83.3	83.3	**86.6**	83.3
50	50	**83.3**	80	**83.3**	**83.3**
25	25	73.3	76.6	**83.3**	76.6
20	20	70	73.3	**80**	73.3
15	15	63.3	76.6	**83.3**	76.6
10	10	63.3	70	**83.3**	70

Table 1: Model 1: all binary, 10 ran var, class states 2, test 30, not divided knowledge

nr. db1	nr. db2	acc 1	acc. 2	acc. Incr
262	237	50	**53.3**	50
200	200	50	**53.3**	50
100	100	50	**53.3**	50
50	50	50	**56.6**	**56.6**
25	25	50	**53.3**	**53.3**
20	20	50	**56.6**	**56.6**
15	15	50	**53.3**	**53.3**
10	10	50	**56.6**	**56.6**

Table 2: Model 1: all binary, 10 ran var, class states 2, test 30, divided knowledge

nr. db1	nr. db2	acc 1	acc. 2	acc. all	acc. Incr
250	250	80	**82**	78	**82**
200	200	78	82	80	**84**
100	100	**82**	78	80	78
50	50	**82**	80	78	**82**
25	25	74	74	**82**	80
20	20	70	74	**82**	76
15	15	68	74	**86**	78
10	10	54	68	70	**76**

Table 3: Model 1: features binary, class var not bin, 10 ran var, class states 4, test 30, not divided knowledge

nr. db1	nr. db2	acc 1	acc. 2	acc. Incr
241	237	52	38	**62**
200	200	54	38	**64**
100	100	56	40	**66**
50	50	62	60	**70**
25	25	54	57.9	**66**
20	20	56	60	**68**
15	15	54	57.9	**66**
10	10	46	48	**56**

Table 4: Model 1: features binary, class var not bin, 10 ran var, class states 4, test 30, divided knowledge

	nr. of samples	mean	median
db 1	45	0.1455	0.1518
db 2	28	0.1388	**0.1444**
All	73	0.1460	0.1498
Incr	73	**0.1380**	0.1463

Table 5: Model 2: not divided knowledge

	nr. of samples	mean	median
db 1	45	0.1784	0.1740
db 2	28	0.1396	0.1476
All	73	**0.1252**	**0.1249**
Incr	73	0.1380	0.1463

(a)

	nr. of samples	mean	median
db 1	45	0.1331	0.1398
db 2	28	0.1388	0.1444
All	73	0.1412	0.1457
Incr	73	**0.1328**	**0.1356**

(b)

Table 6: Model 2, divided knowledge (a) and Model 1, not divided knowledge (b).

cient statistics necessary to estimate the parameters, and therefore, the only possibility to update an MTE model so far was to re-learn from scratch. Our plan is to continue with the theoretical and experimental analysis of the proposed models, as well as the extension of them to handle deterministic relationships among the class variables and the main class.

References

[1] E. Castillo, J.M. Gutiérrez, and A.S Hadi. Expert Systems and Probabilistic Network Models. Springer, 1997.

[2] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. Wiley, 2001.

[3] E. Frank, L. Trigg, G. Holmes, and I.H. Witten. Technical note: Naive Bayes for regression. *Machine Learning*, 41:5–25, 2000.

[4] N. Friedman, D. Geiger and M. Goldszmidt. Bayesian Network Classifiers Machine Learning, 29, pp. 131–163, 1997.

[5] N. Friedman and M. Goldszmidt. Sequential Update of Bayesian Network Structure In: Proc 13th UAI, 1997.

[6] N. Friedman, K. Murphy and S. Russell. Learning the structure of dynamic probabilistic networks. In: Proc 14th UAI, pp. 139–147, 1998.

[7] J.A. Gámez and A. Salmerón. Predicción del valor genético en ovejas de raza manchega usando técnicas de aprendizaje automático. In *Actas de las VI Jornadas de Transferencia de Tecnología en Inteligencia Artificial*, pages 71–80. Paraninfo, 2005.

[8] F.V. Jensen. Bayesian Networks and Decision Graphs. Springer, New York, 2001.

[9] S.L. Lauritzen. Graphical models. Clarendon Press, Oxford, 1996.

[10] R. G. Cowell and A. Philip Dawid and S. L. Lauritzen and D. J. Spiegelhalter. Probabilistic Networks and Expert Systems Springer-Verlag New York, ISBN = 0-387-98767-3, 1999.

[11] S. Moral, R. Rumi, A. Salmerón Mixture of truncated exponetials in hybrid Bayesian networks. *Lecture Notes in Artificial Intelligence* 2143:135–143, 2001.

[12] M. Morales, C. Rodríguez, and A. Salmerón. Selective naive Bayes predictor with mixtures of truncated exponentials. In *Proceedings of the International Conference on Mathematical and Statistical Modeling*, 2006.

[13] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kauffman, San Francisco, CA, 1988.