Efficient Learning of Naive Bayes Classifiers under Class-Conditional Classification Noise

François Denis FRANCOIS.DENIS@LIF.UNIV-MRS.FR
Christophe Nicolas Magnan CHRISTOPHE.MAGNAN@LIF.UNIV-MRS.FR
Liva Ralaivola LIVA.RALAIVOLA@LIF.UNIV-MRS.FR
LIF, UMR CNRS 6166, 39, rue F. Joliot Curie, 13453 Marseille Cedex 13, FRANCE

Abstract

We address the problem of efficiently learning Naive Bayes classifiers under class-conditional classification noise (CCCN). Naive Bayes classifiers rely on the hypothesis that the distributions associated to each class are product distributions. When data is subject to CCC-noise, these conditional distributions are themselves mixtures of product distributions. We give analytical formulas which makes it possible to identify them from data subject to CCCN. Then, we design a learning algorithm based on these formulas able to learn Naive Bayes classifiers under CCCN. We present results on artificial datasets and datasets extracted from the UCI repository database. These results show that CCCN can be efficiently and successfully handled.

1. Introduction

Naive Bayes classifiers are widely used in Machine Learning. Indeed, they can efficiently be learned, they provide simple generative models of the data and they achieve pretty good results in various classification tasks such as text classification. Naive Bayes classifiers rely on the hypothesis that the attributes of the description domain are independent conditionally to each class, i.e. conditional distributions are *product distributions*, but it has often been noticed that they keep achieving good performances even when these conditions are not met (Domingos & Pazzani, 1997). Nevertheless, Naive Bayes classifiers are not very robust to classification noise since independence of the attributes is not preserved. In this paper, we address the problem of efficiently learning binary Naive Bayes classifiers under *class-conditional classification noise* (CCCN), i.e. when the label l of any example is flipped to $1-l$ with a probability η_l which only depends on l. Eliminating class noise in datasets has been studied in several papers (see (Zhu et al., 2003) for a general approach and (Yang et al., 2003) for an approach dedicated to Naive Bayes classifiers: however, the model of noise the authors consider in the last reference is not comparable to the model we consider). When data is subject to CCC-noise, conditional distributions become mixtures of product distributions. Mixtures of product distributions are still fairly simple distributions which have been studied in several papers (Geiger et al., 2001; Whiley & Titterington, 2002; Freund & Mansour, 1999; Feldman et al., 2005). In particular, mixtures of product distributions can be identified from data under some mild hypotheses. However, these results are not very useful in order to learn Naive Bayes classifiers under CCC-noise: indeed, they make it possible to estimate the mixture coefficients by using each conditional distribution separately, providing estimators whose convergence rates are rather slow, while it should be possible to use them together to obtain better and faster estimates. In this paper, we aim at finding efficient estimates based on the available data in the CCCN learning framework.

We give analytical formulas which express the mixtures coefficients of the conditional distributions in function of the noisy conditional distributions. We use these formulas to design efficient estimators for the mixture coefficients. We also show how these formulas can be used to estimate the parameter $P(y=1)$ in an asymetrical semi-supervised learning framework, where the available data is made of unlabeled and positive examples (i.e. from one class). Next, we use these estimators to design an algorithm, `NB-CCCN` capable of learning a Naive Bayes classifier from labeled data subject to CCC-noise. We also design a learn-

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

ing algorithm NB-CCCN-EM which combines NB-CCCN and the EM method: NB-CCCN-EM starts by computing a Naive Bayes classifier by using NB-CCCN and then, uses the EM method to maximize the likelihood of the learning data. We carry out experiments on both artificial data generated from randomly drawn Naive Bayes classifiers and data from the UCI repository. We compare four learning algorithms: the classical Naive Bayes algorithm (NB), an algorithm (NB-UNL) which directly estimates the mixture coefficients from unlabeled data by using analytical formulas taken from (Geiger et al., 2001), NB-CCCN and NB-CCCN-EM. These experiments show that when CCC-noise is added to data, NB-UNL, NB-CCCN and NB-CCCN-EM succeed in handling the additional noise in the data, achieving performances which are close to their performances on non-noisy data. The two latter algorithms are far better than NB-UNL. Obviously, NB-CCCN-EM achieves better performance than NB-CCCN when the comparison criterion is the likelihood. This property entails that NB-CCCN-EM achieves better performance than NB-CCCN on classification tasks on artificial data drawn from noisy product distributions, since in that case, maximizing the likelihood is a good heuristic for classification. However, NB-CCCN achieves better performance than NB-CCCN-EM on real data.

A discussion on supervised learning under class-conditional classification noise is carried out in Section 3. We define the notion of *identifiability under class-conditional classification noise* and we relate it to the identifiability of mixtures of distributions. We give the analytical formulas which express the mixtures coefficients of the conditional distributions in function of the noisy conditional distributions in Section 4. We also describe in Section 4 the estimators of these coefficients and the algorithms NB-CCCN, NB-CCCN-EM and NB-UNL. Our experiments are described in Section 5.

2. Preliminaries

2.1. The Naive Bayes Classifier

Let $X = \prod_{i=1}^{m} X^i$ be a domain defined by m symbolic attributes. For all $x \in X$, let us denote by x^i the projection of x on X^i and let us denote by $Dom(x^i)$ the set of possible values of x^i. Let P be a probability distribution over X and let $Y = \{0, 1\}$ be the set of classes. Y is provided with conditional probability distributions $P(.|x)$ for all $x \in X$.When attributes are independent conditionally to each class, then $P(x|y) = \prod_{i=1}^{m} P(x^i|y)$ is a *product distribution* over$X$ $\forall\, y \in Y$.In such a case, the Bayes classifier is equal to the Naive Bayes classifier C_{NB} defined by $C_{NB}(x) = \underset{y \in Y}{argmax}\; P(y) \prod_{i=1}^{m} P(x^i|y)$.

Naive Bayes classifiers are completely specified by the following set of parameters: $p = P(y = 1)$, $P_+^i(k) = P(x^i = k|y = 1)$ and $P_-^i(k) = P(x^i = k|y = 0)$ where $1 \leq i \leq m$ and $k \in Dom(x^i)$. An instance of these parameters is called a *model* and is denoted by θ.

2.2. Identifying Mixture of Product Distributions

Let $\mathcal{P}$ be a set of distributions over X . We say that the 2-mixtures of elements of $\mathcal{P}$ are *identifiable* if for any $P_1, P_2, P_1', P_2' \in \mathcal{P}$ and any $\alpha, \alpha' \in [0, 1]$,
$\alpha P_1 + (1 - \alpha) P_2 = \alpha' P_1' + (1 - \alpha') P_2'$
$\Rightarrow \alpha' = \alpha, P_i' = P_i \;\forall i$ or $\alpha' = 1 - \alpha, P_1' = P_2, P_2' = P_1$.

A necessary and sufficient condition for identifiability of finite mixtures has been given in (Yakowitz & Spragins, 1968). Identifiability of finite mixtures of product distributions has been proved in (Geiger et al., 2001; Whiley & Titterington, 2002) (under mild conditions). Learning of product distributions has been studied in (Freund & Mansour, 1999) and more recently in (Feldman et al., 2005).

As we shall use it in the experiments, let us give without proof and explanations some details on the way mixture of two distributions on binary attributes are identified in (Geiger et al., 2001). These formulas hold when the number of attributes is at least three.

Let P be a mixture of two product distributions $P(.|y = 0)$ and $P(.|y = 1)$ over $X = \{0, 1\}^r$ where the mixture coefficient is $\alpha = P(y = 1)$. Let $z_{ij...r} = P(x^i = 1, x^j = 1, ..., x^r = 1)$, $p_i = P(x^i = 1|y = 1)$, $q_i = P(x^i = 1|y = 0)$, $\alpha = P(y = 1)$. Therefore $z_{ij...r} = \alpha p_i p_j ....p_r + (1 - \alpha) q_i q_j ...q_r$.

Let $s, x_1, ..., x_m, u_1, ..., u_m$ be the new coordinates after the following transformation: $\alpha = (s + 1)/2, p_i = x_i + (1 - s) u_i, q_i = x_i - (1 + s) u_i$. A second transformation on coordinates z is recursively defined as follows: $z_{ij} \leftarrow z_{ij} - z_i z_j$, $z_{ijr} \leftarrow z_{ijr} - z_{ij} z_r - z_{ir} z_j - z_{jr} z_i - z_i z_j z_r$ and so forth... Then, x, u, s can be computed as follows: $x_i = z_i$, $u_1 = \pm\sqrt{z_{12} z_{13} z_{23} + (z_{123})^2/4}/z_{23}$, $s = -z_{123}/(2 u_1 z_{23})$, $u_i = z_{1i}/(p_2(s) u_1)$ for $i > 1$ with $p_2(s) = 1 - s^2$. In Section 4.5, we propose an algorithm based on these formulas to compute Naive Bayes models from unlabeled data.

3. Supervised Statistical Learning under CCCN

Let X be a discrete domain, and let $Y = \{0, 1\}$. In supervised statistical learning, it is supposed that examples $(x_1, y_1), \dots, (x_l, y_l)$ are independently and identically distributed according to a probability distribution P over $X \times Y$. Then, the goal is to build a classifier $f : X \to Y$ which minimizes the functional risk $R(f) = P(y \neq f(x))$, i.e. which approximates the Bayes classifier f^* defined by $f^*(x) = ArgMax_y P(y|x)$. Here, we consider the case where the examples are submited to an additional *class conditional classification noise*. That, is, we suppose that the examples are independently drawn according to the probability distribution $P^{\overrightarrow{\eta}}$ defined by $P^{\overrightarrow{\eta}}(x,1) = (1-\eta^1)P(x,1) + \eta^0 P(x,0)$ and $P^{\overrightarrow{\eta}}(x,0) = \eta^1 P(x,1) + (1-\eta^0)P(x,0)$ where $\overrightarrow{\eta} = (\eta^0, \eta^1) \in [0,1]^2$. However, our goal remains the same as in the original problem: minimizing the risk relative to P. For any distribution Q on $X \times Y$ such that $Q(1) = \sum_{x \in X} Q(x,1) \in (0,1)$, let us denote by Q_+ (resp. Q_-) the distribution defined on X by $Q_+(x) = Q(x,1)/Q(1)$ (resp. $Q_-(x) = Q(x,0)/Q(0)$ where $Q(0) = 1 - Q(1)$).

Note that if we let $P'(x,y) = P(x, 1-y)$, $\eta'^0 = 1 - \eta^1$ and $\eta'^1 = 1 - \eta^0$, the distributions $P^{\overrightarrow{\eta}}$ and $P'^{\overrightarrow{\eta'}}$ are identical while the Bayes classifiers associated with P and P' are complementary. Hence, we shall suppose that $\eta^0 + \eta^1 \leq 1$ to raise ambiguity. Note also that when $\eta^0 + \eta^1 = 1$, $P_+^{\overrightarrow{\eta}}(x) = P_-^{\overrightarrow{\eta}}(x) = P(x)$ and therefore, nothing better can be done than predicting the labels randomly. So, we shall suppose from now that $\eta^0 + \eta^1 < 1$.

It may happen that Bayes classifiers are identical for the two distributions P and $P^{\overrightarrow{\eta}}$: $P^{\overrightarrow{\eta}}(1|x) \geq P^{\overrightarrow{\eta}}(0|x)$ $\Leftrightarrow (1-\eta^1)P(1|x) + \eta^0 P(0|x) \geq (1-\eta^0)P(0|x) + \eta^1 P(1|x)$ $\Leftrightarrow (1-2\eta^1)P(1|x) \geq (1-2\eta^0)P(0|x)$. When the classification noise is *uniform* (i.e. $\eta^0 = \eta^1$) and $< 1/2$, the distributions P and $P^{\overrightarrow{\eta}}$ define the same Bayes classifier. This is also the case when the problem is deterministic, i.e. $P(1|x) = 0$ or $P(0|x) = 0$ and $\eta^0, \eta^1 < 1/2$.

In all these cases, the strategy which consists in minimizing the empirical risk is as consistent for one distribution as for the other. But when the Bayes classifiers do not coincide, another strategy should be taken. Let us compute $R^{\overrightarrow{\eta}}(f) = P^{\overrightarrow{\eta}}(f(x) \neq y)$ for any classifier $f : X \to Y$. Let us denote $p_f = P(f(x) = 1)$ and $P_j(x,i) = P((x,i)|f(x) = j)$.

$$\begin{aligned} R^{\overrightarrow{\eta}}(f) &= P_0^{\overrightarrow{\eta}}(x,1) \cdot (1-p_f) + P_1^{\overrightarrow{\eta}}(x,0) \cdot p_f \\ &= [(1-\eta^1)P_0(x,1) + \eta^0 P_0(x,0)] \cdot (1-p_f) \\ &\quad + [(1-\eta^0)P_1(x,0) + \eta^1 P_1(x,1)] \cdot p_f \\ &= (1-p_f)[(1-\eta^0-\eta^1)P_0(x,1) + \eta^0] \\ &\quad + p_f[(1-\eta^0-\eta^1)P_1(x,0) + \eta^1] \\ &= (1-\eta^0-\eta^1)R(f) + \eta^1 \cdot p_f + \eta^0 \cdot (1-p_f). \end{aligned}$$

Therefore, we need to minimize

$$R(f) = \frac{R^{\overrightarrow{\eta}}(f) - \eta^1 p_f - \eta^0(1-p_f)}{1-\eta^0-\eta^1} \tag{1}$$

which does not boil down to minimizing $R^{\overrightarrow{\eta}}(f)$ and can be a difficult task since in general, we may not suppose that the noise rates are known.
Consider a simple example: let $X = \{a\}$, let P_1 be such that $P_1(0|a) = 1/3$, $\overrightarrow{\eta^1} = (0,0)$, P_2 be such that $P_2(0|a) = 2/3$ and $\overrightarrow{\eta_2} = (1/2, 0)$. We have $P_1^{\overrightarrow{\eta^1}} = P_2^{\overrightarrow{\eta_2}}$ while the Bayes classifiers associated with P_1 and P_2 are complementary. Therefore, the problem seems to be ill-posed when the Bayes classifiers are different for P and $P^{\overrightarrow{\eta}}$. However, when the underlying distribution P is known to belong to some restricted set of distributions $\mathcal{P}$, the problem may be feasible.

Definition 1. *Let $\mathcal{P}$ be a set of distributions over $X \times Y$. We say that $\mathcal{P}$ is* identifiable under class conditional classification noise *if for any $P \in \mathcal{P}$, any noise rates η^0 and η^1 satisfying $\eta^0 + \eta^1 < 1$, $P^{\overrightarrow{\eta}}$ determines P, i.e. $\forall P_1, P_2 \in \mathcal{P}, \forall \overrightarrow{\eta}_1 = (\eta_1^0, \eta_1^1), \overrightarrow{\eta}_2 = (\eta_2^0, \eta_2^1) \in [0,1]^2$ such that $\eta_1^0 + \eta_1^1 < 1$ and $\eta_2^0 + \eta_2^1 < 1$,*

$$P_1^{\overrightarrow{\eta}_1} = P_2^{\overrightarrow{\eta}_2} \Rightarrow P_1 = P_2 \text{ and } \overrightarrow{\eta}_1 = \overrightarrow{\eta}_2.$$

Let

$$p = P(y = 1) = \sum_{x \in X} P(x,1). \tag{2}$$

We have

$$\begin{cases} P_+^{\overrightarrow{\eta}}(x) = \alpha P_+(x) + (1-\alpha)P_-(x) \\ P_-^{\overrightarrow{\eta}}(x) = \beta P_+(x) + (1-\beta)P_-(x) \end{cases} \tag{3}$$

where

$$\alpha = \frac{p \cdot (1-\eta^1)}{p \cdot (1-\eta^1) + (1-p)\eta^0} \tag{4}$$

$$\beta = \frac{p \cdot \eta^1}{p \cdot \eta^1 + (1-p) \cdot (1-\eta^0)} \tag{5}$$

$P_+^{\overrightarrow{\eta}}(x)$ and $P_-^{\overrightarrow{\eta}}(x)$ are mixtures of the two distributions $P_+(x)$ and $P_-(x)$.

Lemma 1. *Let P be a probability distribution over $X \times Y$, let $\overrightarrow{\eta} = (\eta^0, \eta^1) \in [0,1]^2$ such that $\eta^0 + \eta^1 < 1$ and let p, α and β be defined by (2), (4) and (5). Then,*
$(\alpha = 0 \Leftrightarrow p = 0) \Rightarrow \beta = 0$
$(\beta = 1 \Leftrightarrow p = 1) \Rightarrow \alpha = 1$
$(\alpha = \beta) \Leftrightarrow (p = 0 \vee p = 1)$.

Proof. Straightforward. □

It can easily be derived from previous equations that

$$\eta^0 = \frac{(p-\beta)(1-\alpha)}{(1-p)(\alpha-\beta)} \text{ and } \eta^1 = \frac{\beta(\alpha-p)}{p(\alpha-\beta)}. \tag{6}$$

These relations show that even if α and β are known, the values of p, η^0 and η^1 are not determined yet: for any $p \in [min(\alpha,\beta), max(\alpha,\beta)]$ there exist some values of η^0 and η^1 which are consistent with the data. However, it is easy to show the following proposition.

Proposition 1. *Let $\mathcal{P}$ be a class of distributions over $X \times Y$ and let $\mathcal{Q} = \{P(\cdot|y)|y = 0$ or $y = 1, P \in \mathcal{P}\}$. If the 2-mixtures of $\mathcal{Q}$ are identifiable, then $\mathcal{P}$ is* identifiable under class conditional classification noise.

Proof. Let $P \in \mathcal{P}$ and η^0, η^1 be noise rates satisfying $\eta^0 + \eta^1 < 1$. There exist unique mixture coefficients such that $P_+^{\vec{\eta}}(x) = \alpha P_+(x) + (1-\alpha)P_-(x)$ and $P_-^{\vec{\eta}}(x) = \beta P_+(x) + (1-\beta)P_-(x)$. We have $P^{\vec{\eta}}(1) = (1-\eta^1)p + \eta^0(1-p) = \frac{\alpha(p-\beta)}{\alpha-\beta} + \frac{(1-\alpha)(p-\beta)}{\alpha-\beta} = \frac{p-\beta}{\alpha-\beta}$ and therefore

$$p = \beta + (\alpha-\beta)P^{\vec{\eta}}(1). \tag{7}$$

Then, equations (6) determine η^0 and η^1. □

4. Learning Mixtures of Product Distributions under CCCN

From previous section, the set of 2-mixtures of product distributions is identifiable from CCC-noise. That is, Naive Bayes classifiers can be learned from data subject to class conditional classification noise. But, estimating the mixture coefficients by using data from $P_+^{\vec{\eta}}$ and $P_-^{\vec{\eta}}$ separetely provides estimators whose convergence rates are very low. We show below that, by using data drawn according to $P^{\vec{\eta}}$, we obtain simple and efficient estimates of the mixture coefficients and of the parameters which depend on them.

4.1. Analytical Expressions for Mixture Coefficients

Let P_1 and P_2 be two product distributions over $X_1 \times X_2$, let x_1 and x_2 be the attributes corresponding to X_1 and X_2. For any distribution Q over $X_1 \times X_2$, any $i = 1, 2$ and any $c \in X_i$, let us denote $Q(x_i = c)$ by $Q^i(c)$. Let $Q_\alpha = \alpha P_1 + (1-\alpha)P_2$ and $Q_\beta = \beta P_1 + (1-\beta)P_2$ be two mixtures of P_1 and P_2. Suppose that $\alpha \neq \beta$. We can express P_1 and P_2 as linear combination of Q_α and Q_β:

$$\begin{cases} (\alpha-\beta)P_1 = (1-\beta)Q_\alpha - (1-\alpha)Q_\beta \\ (\alpha-\beta)P_2 = \alpha Q_\beta - \beta Q_\alpha \end{cases} \tag{8}$$

Let $(a,b) \in X_1 \times X_2$. We have $Q_\alpha(a,b) = \alpha P_1(a,b) + (1-\alpha)P_2(a,b) = \alpha P_1^1(a)P_1^2(b) + (1-\alpha)P_2^1(a)P_2^2(b)$ and then, by replacing P_1 and P_2 with the expressions provided by equations (8),
$(\alpha-\beta)^2 Q_\alpha(a,b) =$
$\alpha[(1-\beta)Q_\alpha^1(a) - (1-\alpha)Q_\beta^1(a)][(1-\beta)Q_\alpha^2(b) - (1-\alpha)Q_\beta^2(b)]$
$+(1-\alpha)[\alpha Q_\beta^1(a) - \beta Q_\alpha^1(a)][\alpha Q_\beta^2(b) - \beta Q_\alpha^2(b)]$.

After simplifications, we obtain

$$(\alpha-\beta)^2 D = \alpha(1-\alpha)C, \tag{9}$$

where $C = (Q_\alpha^1(a) - Q_\beta^1(a))(Q_\alpha^2(b) - Q_\beta^2(b))$ and $D = Q_\alpha(a,b) - Q_\alpha^1(a)Q_\alpha^2(b)$. Similarly, we have

$$(\alpha-\beta)^2 E = \beta(1-\beta)C, \tag{10}$$

where $E = Q_\beta(a,b) - Q_\beta^1(a)Q_\beta^2(b)$.

If $\beta = 1$ or $\beta = 0$, (9) can be used to directly compute α:

$$\alpha = \begin{cases} \frac{D}{D+C} & \text{if } \beta = 1 \\ \frac{C}{D+C} & \text{if } \beta = 0 \end{cases} \tag{11}$$

Suppose now that $\beta(1-\beta) \neq 0$.

From (9), we get $\alpha^2 = \frac{\alpha C - \beta D(\beta - 2\alpha)}{C+D}$. Replacing α^2 with this expression in (10), we obtain an expression of α as a function of β:

$$\alpha = \beta \cdot \frac{(1-\beta)(C+D) - \beta E}{E(1-2\beta)} \tag{12}$$

Now, replacing α with this expression in (10), we obtain

$$\beta \cdot (1-\beta) \cdot (\beta^2 - \beta + \lambda_\beta) = 0 \tag{13}$$

where $\lambda_\beta = \frac{CE}{(C+D+E)^2 - 4DE}$. Since $\beta(1-\beta) \neq 0$,

$$\beta \in \left\{ \frac{1+\sqrt{1-4\lambda_\beta}}{2}, \frac{1-\sqrt{1-4\lambda_\beta}}{2} \right\} \tag{14}$$

which provides the two admissible solutions (α_1, β_1) and (α_2, β_2) to the problem. Note that $\alpha_2 = 1 - \alpha_1$ and $\beta_2 = 1 - \beta_1$.

We have proved the following proposition:

Proposition 2. *Let $Q_\alpha = \alpha P_1 + (1-\alpha)P_2$ and $Q_\beta = \beta P_1 + (1-\beta)P_2$ be mixtures of the product distributions P_1 and P_2. Suppose that $\alpha \neq \beta$. Then, (11), (12) and (14) provide analytical expressions of the mixture coefficients α and β.*

4.2. Learning Bayes Classifiers from Positive and Unlabeled Data

A particular semi-supervised learning framework suppose that available samples are unlabeled or labeled according to some predefinite class, that may be called the positive class (see (DeComité et al., 1999; Denis et al., 2003; Li & Liu, 2003; Li & Liu, 2005)). That is, it is supposed that two sources of data provide sample according to the two following distributions over X: $P(x) = P(x, 0) + P(x, 1)$ and $P(x|1)$. In this framework, a critical parameter is $P(y = 1)$: often, it is supposed that it is given, as an additional piece of information on the problem. Proposition 2 shows that when Naive Bayes classifiers are used in this framework, the parameter $P(y = 1)$ can be estimated from data according to equation (13).

Corollary 1. *Let P be a distribution over $X \times Y$ such that P_+ and P_- are product distributions over X. Let x_1 and x_2 be two different attributes, let $a \in Dom(x_1), b \in Dom(x_2)$ and let us denote $P(x_1 = a, x_2 = b)$ by $P^{1,2}(a, b)$, $P(x_i = c, 0) + P(x_i = c, 1)$ by $P^i(c)$ and $P(x_i = c|1)$ by $P^i(c|1)$ for any $c \in Dom(x_i)$. Then,$P(y = 1) =$*

$$\frac{P^{1,2}(a,b) - P^1(a|1)P^2(b|1)}{P^{1,2}(a,b) + P^1(a)P^2(b) - P^1(a)P^2(b|1) - P^1(a|1)P^2(b)}. \tag{15}$$

Proof. Let $Q_\alpha(x) = P^{1,2}(x) = P^{1,2}(x|1)P(y = 1) + P^{1,2}(x|0)P(y = 0)$ and $Q_\beta(x) = P^{1,2}(x|1)$: Q_α is a mixture of the two product distributions $P^{1,2}(x|1)$ and $P^{1,2}(x|0)$ with $P(y = 1)$ as mixture coefficient. We have also $\beta = 1$. Formula 11 yields the formula stated in the corollary. □

A consistent estimator of $P(y = 1)$ can be derived from equation 15. From any samples S_{unl} and S_{pos} of unlabeled and positive data, consider equation 15 for all or some pair of attributes and all or some of their values: $\hat{P}(y = 1) =$

$$\frac{\sum \hat{P}^{i,j}(a,b) - \hat{P}^i(a|1)\hat{P}^j(b|1)}{\sum \hat{P}^{i,j}(a,b) + \hat{P}^i(a)\hat{P}^j(b) - \hat{P}^i(a)\hat{P}^j(b|1) - \hat{P}^i(a|1)\hat{P}^j(b)}.$$

where the sums are taken over all attributes i and j and values $a \in Dom(x_i)$ and $b \in Dom(x_j)$.

4.3. Learning Bayes Classifiers under Class Conditional Classification Noise

Equations (14) and (12) can be used to efficiently identify Naive Bayes classifiers under class conditional classification noise: let x_1 and x_2 be two attributes of X, let $X_1 = Dom(x_1)$ and $X_2 = Dom(x_2)$, let P_1 and P_2 be defined on $X_1 \times X_2$ by $P_1(a, b) = P_+(x_1 = a, x_2 = b)$, $P_2(a, b) = P_-(x_1 = a, x_2 = b)$, $Q_\alpha(a, b) = P_+^{\overrightarrow{\eta}}(x_1 = a, x_2 = b)$ and $Q_\beta = P_-^{\overrightarrow{\eta}}(x_1 = a, x_2 = b)$.

Two pairs (α_1, β_1) and (α_2, β_2) of admissible solutions are computed using equations (14) and (12); for each pair, p, η^0 and η^1 are computed using equations (7) and (6). Only one of these solutions satisfies $\eta^0 + \eta^1 < 1$.

Algorithm 1 NB-CCCN: learn a Naive Bayes classifier from data subject to CCC-noise

```
Input: $S_{lab}^{\overrightarrow{\eta}}$, a labeled dataset subject to CCCN
  1) Compute $\hat{\lambda}_\alpha$ and $\hat{\lambda}_\beta$ using (17) and (16).
  2) Compute values for $\alpha$ and $\beta$ by solving $\hat{\lambda}_\beta = \beta - \beta^2$ and $\hat{\lambda}_\alpha = \alpha - \alpha^2$.
  3) Select the unique admissible solution $(\hat{\alpha}, \hat{\beta})$.
  4) Compute a model $\hat{\theta}$ by using equations 8.
Output: $\hat{\theta}$, an estimate of the target model.
```

We now introduce a learning algorithm, NB-CCCN (algorithm 1), which learns Naive Bayes classifiers from labeled data subject to class-conditional classification noise. Let $S_{lab}^{\overrightarrow{\eta}}$ be a data set drawn according to $P^{\overrightarrow{\eta}}$. For any pair of attributes x_i and x_j and for any pair of elements $(a, b) \in Dom(x_i) \times Dom(x_j)$, let

$$\begin{aligned}\hat{C}_{i,j}^{a,b} &= (\widehat{P_+^{\overrightarrow{\eta}}}(x_i = a) - \widehat{P_-^{\overrightarrow{\eta}}}(x_i = a))\cdot \\ &\quad (\widehat{P_+^{\overrightarrow{\eta}}}(x_j = b) - \widehat{P_-^{\overrightarrow{\eta}}}(x_j = b)), \\ \hat{D}_{i,j}^{a,b} &= \widehat{P_+^{\overrightarrow{\eta}}}(a,b) - \widehat{P_+^{\overrightarrow{\eta}}}(x_i = a)\widehat{P_+^{\overrightarrow{\eta}}}(x_j = b), \\ \hat{E}_{i,j}^{a,b} &= \widehat{P_-^{\overrightarrow{\eta}}}(a,b) - \widehat{P_-^{\overrightarrow{\eta}}}(x_i = a)\widehat{P_-^{\overrightarrow{\eta}}}(x_j = b)\end{aligned}$$

where $\widehat{P_+^{\overrightarrow{\eta}}}$ and $\widehat{P_-^{\overrightarrow{\eta}}}$ are empirical estimates of $P_+^{\overrightarrow{\eta}}$ and $P_-^{\overrightarrow{\eta}}$ computed on $S_{lab}^{\overrightarrow{\eta}}$. An estimate of $\hat{\lambda}_\beta$ of $\lambda_\beta = \beta - \beta^2$ is computed by:

$$\hat{\lambda}_\beta = \frac{\sum \hat{C}_{i,j}^{a,b}\hat{E}_{i,j}^{a,b}}{\sum(\hat{C}_{i,j}^{a,b} + \hat{D}_{i,j}^{a,b} + \hat{E}_{i,j}^{a,b})^2 - 4\hat{D}_{i,j}^{a,b}\hat{E}_{i,j}^{a,b}} \tag{16}$$

where the sums are taken over all pairs (i, j) of attributes and all pair of values $(a, b) \in Dom(x_i) \times Dom(x_j)$. Similarly, an estimate of $\hat{\lambda}_\alpha$ of $\alpha - \alpha^2$ is computed by:

$$\hat{\lambda}_\alpha = \frac{\sum \hat{C}_{i,j}^{a,b}\hat{D}_{i,j}^{a,b}}{\sum(\hat{C}_{i,j}^{a,b} + \hat{D}_{i,j}^{a,b} + \hat{E}_{i,j}^{a,b})^2 - 4\hat{D}_{i,j}^{a,b}\hat{E}_{i,j}^{a,b}}. \tag{17}$$

Then, let β_1 and β_2 (resp. α_1 and α_2) be the two solutions of $\hat{\lambda}_\beta = \beta - \beta^2$ (resp. $\hat{\lambda}_\alpha = \alpha - \alpha^2$). Only one pair (α_i, β_j) is compatible with the hypotheses. A model is then computed by using equations (8).

4.4. Algorithm to Learn Naive Bayes Models under CCCN using E.M.

Given a sample $S_{lab}^{\overrightarrow{\eta}}$ composed of labeled examples subject to class-conditional classification noise, we could build a Naive Bayes classifier by using maximum likelihood estimates *if we could know which examples have been corrupted.* But unfortunately, this piece of

```
Algorithm 2 NB-CCCN-EM Learning Naive Bayes classifiers with CCC-noise using E.M.
Input: $S_{lab}^{\overrightarrow{\eta}}$, a labeled dataset subject to CCCN
  1) Run algorithm NB-CCCN, $\theta^0$= model infered by this algorithm
  2) $\forall (x', y') \in S_{lab}^{\overrightarrow{\eta}}$, compute $Pr(C(x,y)|\theta_k, \overrightarrow{\eta_k})$ using formulas (18)
  3) Compute a new model $\theta^{k+1}$ using formulas (18), (19), (20) and (21)
  4) Iterate to step 2 until stabilization
Output: $\hat{\theta}_{ML}$
```

```
Algorithm 3 NB-UNL: compute Naive Bayes models from unlabeled data
Input: $z$
  1) Estimate $u_k^+ = \frac{\sum_{\substack{1 \leq i,j \leq m \\ i \neq j \neq k}} \sqrt{z_{ki} z_{kj} z_{ij} + (z_{kij})^2/4}}{\sum_{1 \leq i,j \leq m, i \neq j \neq k} z_{ij}}$ $\forall k \in \{1, .., m\}$, $u_k^- = -u_k^+$
  2) Estimate $s^+ = -\frac{\sum_{1 \leq i,j,k \leq m, i \neq j \neq k} z_{ijk}}{\sum_{1 \leq i,j,k \leq m, i \neq j \neq k} 2u_i z_{jk}}$, $s^- = -s^+$
  3) Compute model $\theta^+$ (resp. $\theta^-$) from $u_1^+$ (resp. $u_1^-$) and $u_i^+$ or $u_i^-$ $(i > 1)$ according to the sign of $z_{1i}$ i.e. such that $sign(u_i) = sign(z_{1i}/(p_2(s)u_1^+))$ (resp. $sign(u_i) = sign(z_{1i}/(p_2(s)u_1^-)))$
Output: two models $\theta^+$ and $\theta^-$.
```

information is missing. E.M. is a standard method which can be used in such situations. Let θ_k be a Naive Bayes model for the data and let $\overrightarrow{\eta_k} = (\eta_k^0, \eta_k^1)$ be a noise model. For any example $(x, y) \in S_{lab}^{\overrightarrow{\eta}}$, we can compute the probability $Pr(C(x,y)|\theta_k, \overrightarrow{\eta_k})$ (denoted by $P_k(C(x,y))$) that (x, y) has been corrupted by noise in the model $\theta_k, \overrightarrow{\eta_k}$:

$$P_k(C(x,y)) = \frac{P(1-y|x,\theta_k)\eta_k^{1-y}}{P(1-y|x,\theta_k)\eta_k^{1-y} + P(y|x,\theta_k)(1-\eta_k^y)} \quad (18)$$

By using this formula, we can compute for any $z \in \{0, 1\}$ the probability that the label of the example were z before the noise step, and then compute new models $\theta_{k+1} = \{p_{k+1}^l = P_{k+1}(y = l), P_{k+1}^{ial} = P_{k+1}(x^i = a|y = l)\}$ and $\overrightarrow{\eta_{k+1}} = \{\eta_{k+1}^0, \eta_{k+1}^1\}$ by maximizing the likelihood of these new data. Knowing that $n = |S_{lab}^{\overrightarrow{\eta}}|$, $S_l^{\overrightarrow{\eta}} = \{(x,y) \in S_{lab}^{\overrightarrow{\eta}} | y = l\}$, probabilities p_{k+1}^l, P_{k+1}^{ial}, η_{k+1}^l are computed as follows:

$$n.p_{k+1}^l = \sum_{S_l^{\overrightarrow{\eta}}} (1 - P_k(C(x,l))) + \sum_{S_{1-l}^{\overrightarrow{\eta}}} P_k(C(x, 1-l)) \quad (19)$$

$$n.P_{k+1}^{ial} = \sum_{\substack{S_l^{\overrightarrow{\eta}} \\ x^i = a}} (1 - P_k(C(x,l))) + \sum_{\substack{S_{1-l}^{\overrightarrow{\eta}} \\ x^i = a}} P_k(C(x, 1-l)) \quad (20)$$

$$\eta_{k+1}^l = \frac{\sum_{S_{1-l}^{\overrightarrow{\eta}}} P_k(C(x, 1-l)}{\sum_{S_l^{\overrightarrow{\eta}}} (1 - P_k(C(x,l))) + \sum_{S_{1-l}^{\overrightarrow{\eta}}} P_k(C(x, 1-l))} \quad (21)$$

We note NB-CCCN-EM the corresponding algorithm.

4.5. Algorithm to Compute Naive Bayes Models from Unlabeled Data

In this section, we use formulas from Section 2.2 to compute Naive Bayes models parameters from unlabeled data. Note that the $z_{ij...r}$ can be estimated from data. Note also that two models can be computed; each of them depending on the sign of u_1. We deduce from these formulas the Algorithm NB-UNL. Experimental results on artificial data (Section 5.1) show that huge samples are necessary to provide accurate estimates of the parameters of the target models.

5. Experiments

We present now our experiments on artificial data and data from the UCI repository data.

5.1. Results on Artificial Data

5.1.1. Protocol

The target model $\theta^t = \{P(y = 1), P_+, P_-\}$ is randomly drawn, distributions P_+ and P_- being product distributions over $\{0, 1\}^{10}$. The learning datasets are generated with model θ^t. For each $n \in \{100, 200, ..., 2000\}$, 200 independent datasets of n examples are drawn. The results (Figures 1 and 2, Table 1) are averages computed on these 200 datasets. The class labels computed by θ^t are flipped with probability η^1 for examples $(x, 1)$ and η^0 for examples $(x, 0)$. Test sets S_{test} contain 1000 examples generated from θ^t. The classes of the test data are computed according to θ^t; they are not corrupted by any noise.

5.1.2. Accuracy of the Estimates

We first compare the accuracy of estimates provided by four algorithms: NB-CCCN, standard Naive Bayes algorithm (denoted by NB), NB-CCCN-EM and NB-UNL. The criterion for comparison is the Kullback-Leibler distance d_{kl} between the target distribution $P(.,.)$ and the predicted distribution $\hat{P}(.,.)$. Figure 1 shows the Kullback-Leibler distance between the inferred models and the target model as a function of $|S_{lab}|$.

These results show that NB-CCCN and NB-CCCN-EM provide accurate estimates of the target and converge faster than NB-UNL.

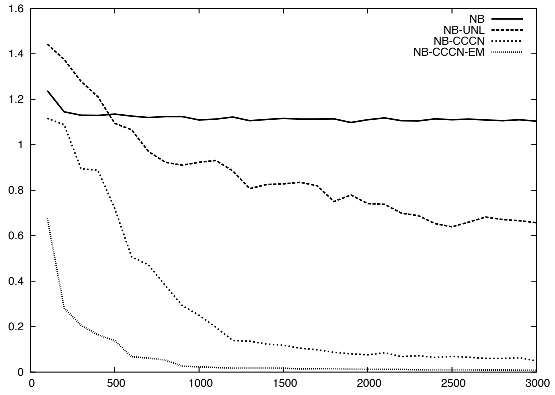

Figure 1. The Kullback-Leibler distance between the target model and the inferred one as a function of the size of the training sample. We set $\eta^0 = 0.2$ and $\eta^1 = 0.5$.

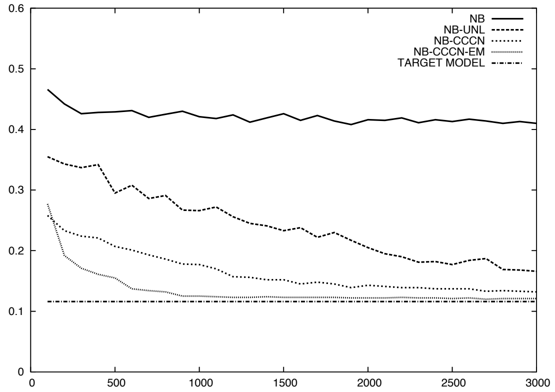

Figure 2. Error rates of the algorithms as a function of the size of the training sample. We set $\eta^0 = 0.2$ and $\eta^1 = 0.5$

We have carried out other experiments where EM is run on randomly drawn initial models: many runs are necessary to obtain a high likelihood while using the model inferred by NB-CCCN as the initial model makes it possible to run EM only once.

5.1.3. Prediction Rate Results

We now present the results obtained for classification tasks. The experimental protocol is described in Section 5.1.1. Two criteria are considered to compare the four algorithms: the prediction rate ($\hat{P}(f(x) = y)$) on test data (denoted by acc in table 1) and the classical F *score*, defined by $F = \frac{2 \cdot TP}{FP + 2 \cdot TP + FN}$; where TP is the number of positive examples correctly classified, FP the number of negative examples incorrectly classified and FN the number of misclassified positive examples. The results for both criteria are reported in Table 1 and Figure 2 shows the evolution of the error rate ($\hat{P}(f(x) \neq y)$) as a function of $|S_{lab}|$. These results show that NB-CCCN and NB-CCCN-EM converge towards the target very quickly in comparison to NB-UNL. Standard Naive Bayes algorithm obviously does not identify the target model.

Table 1. Results for experiments on artificial data, for each algorithm, we report the accuracy acc=$\hat{P}(f(x) = y)$ and the F-score F. The examples have 10 binary descriptive attributes. $\eta^0 = 0.2$, $\eta^1 = 0.5$. Best results are in boldface.

Algo.	$\|S_{lab}\|$	100	500	2000	3000
θ^t	*acc*	0.884	0.884	0.884	0.884
	F	0.925	0.925	0.925	0.925
NB	*acc*	0.534	0.571	0.584	0.590
	F	0.562	0.600	0.619	0.627
NB-CCCN	*acc*	**0.742**	0.793	0.857	0.868
	F	**0.842**	0.876	0.909	0.915
NB-CCCN-EM	*acc*	0.723	**0.845**	**0.878**	**0.879**
	F	0.810	**0.897**	**0.921**	**0.921**
NB-UNL	*acc*	0.645	0.705	0.795	0.834
	F	0.722	0.777	0.851	0.881

Table 2. Six UCI datasets used: $|S|$ is the size of the datasets, *NbAtt* the number of attributes, $|Dom(x^i)|$ the size of the attribute domains.

Dataset	$\|S\|$	*NbAtt*	$\|Dom(x^i)\|$
House Votes	433	16	2
Tic Tac Toe	958	9	3
Hepatitis	155	19	2-10
Breast Cancer	286	9	2-11
B. C. Wisc.	699	9	10
Bal. Scale	576	4	5

5.2. Results on UCI Repository Datasets

This section presents experiments on six datasets from the UCI repository (Merz & Murphy, 1998) and shows that the performances of NB-CCCN and NB-CCCN-EM on real data remain very good even when class-conditional classification noise is added to the data. We test our algorithms and Naive Bayes algorithm on datasets: House Votes, Tic Tac Toe, Hepatitis, Breast Cancer, Breast Cancer Wisconsin, and Balance Scale (see Table 2). In this last dataset, we have only used data whose class is "right" or "left".

As for the experimental protocol, we first run algorithms on the datasets without adding noise; secondly, we added noise to the learning data according to the noise parameters $\eta^0 = 0.2$ and $\eta^1 = 0.5$ without modifying classes of the test data and we relaunch the same algorithms on these noisy data (see Table 3 for results). We have used 10-fold cross-validation per experiment and the results are averaged over 10 experiments.

Table 3. Prediction rate (ac), log-likelihood (lk) and estimates of the noise rates obtained by the four algorithms for UCI datasets without noise and when noise is added to the training examples. MC = Majority class. (*) raw estimates are slightly negative. We set $\overrightarrow{\eta} = (0.20, 0.50)$

Dataset		MC	NB	NB-	NB-CC
				CCCN	CN-EM
H.Votes	ac	0.62	0.904	**0.916**	0.882
no noise	lk	-	-3134	-3035	**-2915**
	$\overrightarrow{\hat{\eta}}$	-	-	(.02,.08)	(.04,.20)
H.Votes	ac	0.38	0.866	**0.900**	0.873
$\overrightarrow{\eta}$ noise	lk	-	-4130	**-3037**	-3041
	$\overrightarrow{\hat{\eta}}$	-	-	(.33,.58)	(.20,.56)
T.T.T.	ac	0.65	**0.697**	0.682	**0.697**
no noise	lk	-	**-8726**	-8854	**-8726**
	$\overrightarrow{\hat{\eta}}$	-	-	(.09,.19)	(.00,.00)
T.T.T.	ac	0.35	0.562	**0.664**	0.587
$\overrightarrow{\eta}$ noise	lk	-	-8828	-8818	**-8815**
	$\overrightarrow{\hat{\eta}}$	-	-	(.24,.62)	(.21,.56)
Hepat.	ac	0.79	0.827	**0.850**	0.770
no noise	lk	-	-1982	-2416	**-1902**
	$\overrightarrow{\hat{\eta}}$	-	-	(.31,.03)	(.50,.03)
Hepat.	ac	0.21	0.590	**0.811**	0.758
$\overrightarrow{\eta}$ noise	lk	-	-2095	-2273	**-1946**
	$\overrightarrow{\hat{\eta}}$	-	-	(.25,.55)	(.29,.45)
Br.Can.	ac	0.70	0.730	**0.760**	0.718
no noise	lk	-	-2520	-2682	**-2448**
	$\overrightarrow{\hat{\eta}}$	-	-	(.06,.20)	(.13,.27)
Br.Can.	ac	0.30	0.581	**0.732**	0.722
$\overrightarrow{\eta}$ noise	lk	-	-2573	-2623	**-2479**
	$\overrightarrow{\hat{\eta}}$	-	-	(.19,.59)	(.33,.56)
Br.C.W.	ac	0.66	0.973	0.972	**0.975**
no noise	lk	-	-7244	-7790	**-7096**
	$\overrightarrow{\hat{\eta}}$	-	-	(.01,.12)	(.00,.05)
Br.C.W.	ac	0.34	0.964	0.967	**0.974**
$\overrightarrow{\eta}$ noise	lk	-	-9015	-7818	**-7395**
	$\overrightarrow{\hat{\eta}}$	-	-	(.02,.05)	(.22,.50)
B.Scale	ac	0.50	**0.994**	0.980	0.993
no noise	lk	-	-3485	**-3445**	-3484
	$\overrightarrow{\hat{\eta}}$	-	-	(.00,.00)*	(.00,.00)
B.Scale	ac	0.50	0.743	**0.847**	0.794
$\overrightarrow{\eta}$ noise	lk	-	**-3611**	-3710	**-3611**
	$\overrightarrow{\hat{\eta}}$	-	-	(.10,.52)	(.06,.36)

The results on House Votes, Hepatitis and Breast Cancer Wisconsin datasets clearly show that the noise added to the data has significantly been erased by `NB-CCCN` and `NB-CCCN-EM`, preserving a rather high classification accuracy. The results on Tic Tac Toe and Breast Cancer are close to those obtained by the majority class rule but Naive Bayes classifiers are unadapted to these datasets. For Balance Scale dataset, both `NB-CCCN` and `NB-CCCN-EM` are significantly less accurate when noise is added to the learning examples. Nevertheless, the results remain much better than the majority class rule.

6. Conclusion

We provide analytical formulas which can be used to learn Naive Bayes classifiers under class-conditional classification noise. The algorithms we design achieve good performances in classification on both artificial and real data. However, it would be interesting to precise the rate of convergence of our estimators and provide theoretical bounds. The experiments we have carried out suggest that CCC-noise can be erased from data while noisy test data cannot be used to attest the successful handling of noise. This observation must be related to Equation (1) which shows that minimizing the empirical risk on noisy data is not a consistent strategy when the noise rates are high. Future work should include the description of a consistent learning principle in the CCCN learning framework.

References

DeComité, F., Denis, F., Gilleron, R., & Letouzey, F. (1999). Positive and unlabeled examples help learning. *ALT 99, 10th In. Conf. on Algorithmic Learning Theory.*

Denis, F., Gilleron, R., Laurent, A., & Tommasi, M. (2003). Text classification and co-training from positive and unlabeled examples. *Proc. of the ICML 2003 workshop: The Continuum from Labeled to Unlabeled Data.*

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103–130.

Feldman, J., O'Donnell, R., & Servedio, R. A. (2005). Learning mixtures of product distributions over discrete domains. *Proceedings of FOCS 2005* (pp. 501–510).

Freund, Y., & Mansour, Y. (1999). Estimating a mixture of two product distributions. *Proceedings of COLT'99.*

Geiger, D., Heckerman, D., King, H., & Meek, C. (2001). Stratified exponential families: graphical models and model selection. *Annals of Statistics*, *29*, 505–529.

Li, X., & Liu, B. (2003). Learning to classify texts using positive and unlabeled data. *Proceedings of IJCAI 2003.*

Li, X., & Liu, B. (2005). Learning from positive and unlabeled examples with different data distributions. *Proceedings of ECML 2005* (pp. 218–229).

Merz, C., & Murphy, P. (1998). UCI repository of machine learning databases.

Whiley, M., & Titterington, D. (2002). *Model identifiability in naive bayesian networks* (Technical Report).

Yakowitz, S.J. & Spragins, J.D. (1968). On the identifiability of finite mixtures. *The Annals of Mat. St.*, *39.*

Yang, Y., Xia, Y., Chi, Y. & Muntz, R.R. (2003). *Learning naive bayes classifier from noisy data.* CSD-TR 030056.

Zhu, X., Wu, X., & Chen, Q. (2003). Eliminating class noise in large datasets. *ICML* (pp. 920–927).