Study of the paper 'On the Optimality of the Simple Bayesian Classifier under Zero-One Loss'

Authors :

ZOTTO Nicola
ABECIDAN Rony

Introduction

In this report, we review a paper titled "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss" [1] by Domigos and Michael Pazzani of the University of California published for the "Machine Learning" international forum in 1997. In the paper, the authors verify both theoretically and empirically that the "naive" Bayesian classifier does not require attribute independence to be optimal under zero-one loss and that it can perform quite well in practice on data sets with strong attribute dependence. Additionally, the authors review how to best extend the simple Bayesian classifier and propose necessary conditions to verify that the Bayesian classifier is an optimal predictor under zero-one loss even when the independence assumption is violated. We are going to first, recall the definition of the "naive" Bayesian classifier and it's main advantages and drawbacks. We then resume the conclusions of the article studied on the optimality of this classifier when the "naive" assumption is violated and on the conditions for this optimality. Finally, we propose an extension of the paper by implementing and studying some points evoked in the article and we propose at the end some strategies that may help the naive Bayes classifier to achieve a better performance on any dataset. All our experiments are in the repo : `https://github.com/RonyAbecidan/NBC_01loss`

1 About the Naive Bayesian Classifier

In this section we will present general concepts used in the paper. We recall the general principle of supervised learning, the definition of a Naive Bayes Classifier and of the zero-one loss.

1.1 The optimal classifier under the zero-one loss

Let $\mathcal{X}$ be a subset of $\mathbb{R}^D$ with D a certain integer.

Let $\mathcal{Y}$ be a finite set.

Let us suppose that there exists a function f^* called a **classifier** which associate to each point of $\mathcal{X}$, a point of $\mathcal{Y}$ [1]. We can define a "classification problem" as follows:

$$\boxed{\text{Find the best approximation of the classifier } f^*}$$

In order to find this approximation, we use a dataset composed of couples (x, y) such that $y = f^*(x)$. We call x an **observation** or an **example**, y the **target** and our dataset $\mathcal{D}$ can be defined as:

$$\boxed{\mathcal{D} = \{(x_i, y_i)_{1 \leq i \leq n}, x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}}$$

The process of using this dataset to infer some classifier f that approximate f^* is called **supervised learning**.

In practice, the quality of a learned classifier is evaluated using a **loss function**. This function quantifies how much we "suffer" by choosing our classifier f to predict the target of a certain observation compared to using the **true** classifier f^*.

One of the most commonly used loss functions for classification problems is the **zero-one loss function**, defined as follows for $x \in \mathcal{X}$:

$$\boxed{l_{0-1}(f(x), f^*(x)) = \mathbb{1}_{f(x) \neq f^*(x)}}$$

This loss function quantifies the *misclassification rate* of a given classifier. And is generally recognized as the natural way to penalize a classifier.

In the **Bayesian framework** one suppose that observations are drawn from a probability distribution $p(x, y)$ that the classifier is trying to capture. In this context, the risk suffered is quantified by the expectation of the loss function under this unknown distribution.

In the case of the 0-1 loss, this risk is easily computed:

$$\begin{aligned}\mathbb{E}_{(x,y=f^*(x))\sim p}(l_{0-1}(f(x), f^*(x)) &= \mathbb{E}_{(x,y)\sim p}(\mathbb{1}_{f(x)\neq y}) \\ &= \mathbb{P}_{(x,y)\sim p}(f(x) \neq y)\end{aligned}$$

[1] corresponding to what we can call a category

In the end, we are interested in the probability that our classifier f will mislabeled a random observation $x \in \mathcal{X}$ drawn by the probability distribution p. We can consider a classifier $\hat{f}$ to be optimal under the 0-1 loss if it minimizes this probability among the set of all possible classifier. Minimizing this probability equates to maximizing the probability of success. And so, the **optimal classifier** under the 0-1 loss is obtained by the relation:

$$\boxed{\hat{f}(x) = \arg\max_y p(x, y) = \arg\max_y p(x)\, p(y|x) = \arg\max_y p(y|x)}$$

This is called **Bayes rule** or **Bayes theorem**.

If we have k **classes** or **categories** in $\mathcal{Y}$, we call $\mathcal{C}_k$ the set of all the points $x \in \mathcal{X}$ that belong to the class k. Using this notation we retrieve the following definition of the Bayes classifier from the paper:

$$\boxed{\hat{f}(x) = \arg\max_k p(\mathcal{C}_k|x)}$$

From this definition we can say that obtaining the optimal classifier boils down to being able to estimate as closely as possible the **posterior probabilities** $p(\mathcal{C}_k|x)$. In practice, there exists a large number of model which are designed to compute this approximation. The paper studied focuses on a specific model called "naive" Bayesian classifier.

In practice, the same observations from a dataset could be assigned to different classes. This happens when some attributes used to describe the observations are not completely discriminant for the final prediction. That's why, there exists a generalization of the 0-1 loss function defined as follows:

$$\boxed{l_{0-1}(f(x), f^*(x) = k) = 1 - p(\mathcal{C}_k|x)}$$

1.2 A little remark about the optimality under the 0-1 loss

The paper studies different properties of the **zero-one loss** used for classification tasks and, compare it to the famous **mean-squared error loss** used for probability estimation.

The authors have pointed out the different optimality measures provided by these different loss functions. Indeed, the **mean-squared error** measures to what extent a classifier succeed in capturing the posterior probabilities $p(\mathcal{C}_k|x)$ while, the **zero-one** loss measures how far the attribution of the classes is correct, despite of the uncorrectness on the posterior probabilities.

Actually, using the 0-1 loss, we only "suffer" when the class predicted for an observation is not the true one which maximizes the posterior probabilities $p(\mathcal{C}_k|x)$.

This allows some classifiers to be optimal under the 0-1 loss even if, the predicted probabilities can be different from the real values.

1.3 Some definitions

We will call the **Bayes rate** for an observation, the lowest zero-one loss achievable by any classifier on that observation.

We will say that a classifier is **locally optimal** for a given observation if and only if its zero-loss on that precise observation equates to the Bayes rate.

We will consider a classifier to be **globally optimal** or **optimal** for **a data set** in the case where this classifier is locally optimal for every observation composing the data set considered.

And finally, a classifier will be qualified as **globally optimal** or **optimal** for **a problem** iff it is globally optimal for any possible set of observations of that problem.

1.4 The 'naive' assumption

In order to access to the posterior probabilities $p(\mathcal{C}_k|x)$, a common approach is to consider **generative models**. Meaning models such that the likelihoods $p(x|\mathcal{C}_k)$ and the priors $\mathbb{P}(\mathcal{C}_k)$ are explicitly learned in the training process, and later used in order to perform predictions.

If one assumes that the attributes of the observations are independent given the class, the following equation is obtained:

$$p(x|\mathcal{C}_k) = \prod_{i=1}^{D} p(x_i|\mathcal{C}_k) \tag{1}$$

This assumption is called the "naive" assumption since, in practice, it will almost systematically be broken. Bayesian classifiers using this assumption are called **naive Bayesian classifiers**.

This type of classifier makes a decision using the probabilities:

$$p(\mathcal{C}_k|x)\ \alpha\ \mathbb{P}(\mathcal{C}_k)p(x|\mathcal{C}_k) = \mathbb{P}(\mathcal{C}_k)\prod_{i=1}^{D} p(x_i|\mathcal{C}_k) \tag{2}$$

Additionally, when this assumption holds, the Bayesian classifier can be shown to be optimal under the 0-1 loss through a direct application of Bayes' theorem.

1.5 Advantages and Drawbacks of the Naive Bayesian Classifier

The naive Bayesian classifier is a widely known classifier that presents concrete advantages with respect to other classifiers.

One such advantage is that there are few parameters to learn in this model regardless of the size of the dataset and of the number of features of each observation. This is easily illustrated by considering the case in which we work only with observations made of binary features that follow Bernoulli distributions conditioned to the classes. If we call c the number of classes and a the number of attributes, we need to learn $O(ca)$ parameters. This is considerably less than the number of probabilities to compute if we want the exact joint distribution that samples the observations conditioned to the classes, which is $O(c2^a)$.

Moreover, this model is often efficient in practice and enable to work with features of different natures (numeric, categorical, binary). On top of that, it is considered interpretable since we know exactly how the data is generated. For this reason, the naive Bayesian has been, and is the subject of a lot of research.

However, the naive Bayesian classifier is not a perfect classifier. And there are some important drawbacks that must be discussed.

First, the naive assumption is, as implied by it's name, rarely applicable in practice. This is largely considered to negatively impact the accuracy of this classifier. For this reason, the naive Bayesian classifier is often used as a baseline to compare other models. With the expected result that, since the naive assumption is violated, the classifier will perform poorly.

One other important thing to consider is that this model is very sensitive to events that aren't sufficiently represented in the dataset used for the training of the model or so called *rare events*. For example, let us consider a binary feature p such that the proportion of observations x in $\mathcal{C}_k$ for which x_p is achieved is very small in our dataset, only 10^{-8} for instance. This could be rewritten with the following equation :

$$p(x_p = 1|\mathcal{C}_k) = \frac{n_{p,k}}{N_k} = 10^{-8}$$

with $n_{p,k}$ the number of observations in $\mathcal{C}_k$ for which x_p is achieved and, N_k the number of observations in $\mathcal{C}_k$

In this case, for any observation x such that $x_p = 1$, this very small proportion will be present in the computation of $p(x|\mathcal{C}_k)$ according to Equation (1) and as a result will reduce this probability to a very small number. Because of this, the information provided by the probabilities $(p(x_i|\mathcal{C}_k))_{i\neq p}$ is lost.

The article studied provides a way to mitigate the effect of this drawback using a prior : The idea is to add a small correction into all the probabilities $p(x_i|\mathcal{C}_k)$ called the **Laplace correction**. For instance, the article proposes to consider the correction factor $f = \frac{1}{n}$ with n the number of observations in the dataset such that the probabilities we consider become:

$$p(x_i|\mathcal{C}_k) = \frac{n_{i,k} + f}{N_k + fn_i}$$

with n_i the number of possible values for the feature i (2 for a binary feature for instance).

This approach can be criticized when working with a rather large dataset, since this correction factor will be practically unnoticeable. We believe that a sound alternative could be to perform the correction with respect to the number of attributes rather than the size of the dataset. We perform a numerical application to illustrate our point:

Let there be a dataset composed of 1000 observations, each characterized by 10 features such that a certain binary feature p is not represented for a certain class k, and that there are 100 observations belonging to the class k in the dataset.

By applying the naive assumption directly, we obtain $p(x_p|\mathcal{C}_k) = 0$

With the Laplace correction used in the article, we obtain $p(x_p|\mathcal{C}_k) = \frac{\frac{1}{1000}}{100+\frac{2}{1000}} \sim 10^{-5}$

And using the correction proposed above, $p(x_p|\mathcal{C}_k) = \frac{\frac{1}{10}}{100+\frac{2}{10}} \sim 10^{-3}$

In this example, the Laplace correction used in the article doesn't solve completely the problem. The probability of the rare event is still very small and potentially small enough to hide a lot of information from $p(x|\mathcal{C}_k)$. By using our proposed correction we found a small probability as well, but perhaps a more acceptable one. In our opinion using this correction still discloses that the event considered is rare while preserving more information for $p(x|\mathcal{C}_k)$.

At the end, the choice of the correction should result from a prior knowledge of the task at hand. If the 'rare' events are really discriminant, representing them with a very low probability can be relevant. And on the contrary, if the 'rare' events aren't intuitively useful for the final predictions we can choose to represent them with still low probability but enough high to not hide too many information from the others events.

1.6 What happen when the naive assumption is broken ?

In the paper studied, the authors point out that the naive Bayesian classifier could perform better than expected when the naive assumption is violated.

We liked the first example they use to illustrate that fact[2] and we are going to extend it a bit in order to emphasize on the final surprising result.

We consider a situation in which we have for instance $k+1$ features F_1, F_2,...F_{k+1} following a Bernoulli law and a binary target T such that :

$$\begin{aligned} F_1 = F_2 = ... = F_k \\ F_1 \perp F_{k+1} \\ \mathbb{P}(T=0) = \mathbb{P}(T=1) = 0.5 \end{aligned}$$

In the following we will note $p(T=t|x) = p(t|x)$

In that case, the naive assumption is obviously not respected since F_1, F_2,...F_{k-1} and F_k are totally dependent.

Now, for one observation x, if we call $p_x = p(1|x_1)$ and $q_x = p(1|x_{k+1})$, the optimal prediction for x under the zero-one loss is obtained by considering only the independent features F_1 and F_{k+1} and lies on the following quantity:

$$\begin{aligned} p(1|x) - p(0|x) \; \alpha \; & p(x|1) - p(x|0) \;\; (A) \\ & = p(x_1|1)p(x_{k+1}|1) - p(x_1|0)p(x_{k+1}|0) \\ & \; \alpha \; p(1|x_1)p(1|x_{k+1}) - p(0|x_1)p(0|x_{k+1}) \;\; (B) \\ & = p_x q_x - (1-p_x)(1-q_x) := \mathcal{Q}_1(x) \end{aligned}$$

(A) : Naive assumption + The sign of $p(1|x) - p(0|x)$ is independent of $p(x) \geq 0$, so there is no need to consider this probability in the computation of $\mathcal{Q}_1$.

(B) : The sign of $p(x_1|1)p(x_{k+1}|1) - p(x_1|0)p(x_{k+1}|0)$ is independent of $p(x_1)p(x_{k+1}) \geq 0$ and $p(0) = p(1)$ by assumption.

In that case, if $\mathcal{Q}_1(x) > 0$, the optimal classifier will assign to x the class 1, and 0 otherwise.

Now, if we consider our naive Bayesian classifier using all the features, the prediction for x lies on the following quantity :

$$\begin{aligned} p(1|x) - p(0|x) \; \alpha \; & p(x|1) - p(x|0) \\ & = p(x_1|1)^k p(x_{k+1}|1) - p(x_1|0)^k p(x_{k+1}|0) \\ & \; \alpha \; p(1|x_1)^k p(1|x_{k+1}) - p(0|x_1)^k p(0|x_{k+1}) \\ & = p_x^k q_x - (1-p_x)^k (1-q_x) := \mathcal{Q}_k(x) \end{aligned}$$

(Same simplifications as before)

In that case, if $\mathcal{Q}_k(x) > 0$, the naive Bayesian classifier will assign to x the class 1, and 0 otherwise. So, in practice, the positive labelling happens when $q_x > \frac{(1-p_x)^k}{p_x^k + (1-p_x)^k}$.

Now, it's interesting to see to what extent our Bayesian classifier will perform "badly" compared to the optimal classifier under the zero-one loss. Intuitively, the more equal attributes we consider, the more the naive assumption is broken and the worse the classifier should perform.

We plotted the decision boundary of the optimal classifier against the decision boundaries of the naive Bayesian classifiers using a certain number k of totally dependent features and obtained the following figure:

[2] section 4: *An example of optimality without independence*

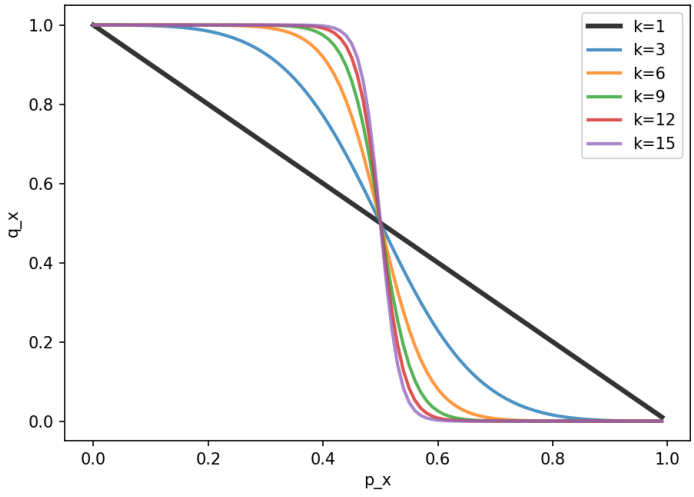

Figure 1: Performance of the Naive Bayesian Classifier using equal attributes for the prediction against the optimal Classifier under the 0-1 loss

The black line represents the optimal decision boundary. We can observe from this graph that, even in the case where we consider several equal attributes, the region in which the Naive Bayesian classifier give an opposite result to the optimal one is quite narrow.
In fact, in the situation in which the number of equal attributes tends to infinity, we can show that the area of this region is precisely $\frac{1}{4}$. Thus, even in a case in which the naive assumption is completely unsatisfied, the Naive Bayesian Classifier give the same predictions as the optimal classifier under the 0-1 loss in a rather vast region of the square $[0,1]$x$[0,1]$.

Proof :
Let x be an observation that we want to classify

We know that the decision boundary has for equation : $q_x = \frac{(1-p_x)^k}{p_x^k+(1-p_x)^k}$

Now, **if** $p_x < 0.5$, then $p_x < 1 - p_x$ and so,

$$\frac{(1-p_x)^k}{p_x^k+(1-p_x)^k} = \frac{1}{(\frac{p_x}{1-p_x})^k+1} \to 1 \text{ when } k \to \infty$$

Similarly, when $p_x < 0.5$, $1 - p_x < p_x$

$$\frac{(1-p_x)^k}{p_x^k+(1-p_x)^k} = \frac{(\frac{1-p_x}{p_x})^k}{1+(\frac{1-p_x}{p_x})^k} \to 0 \text{ when } k \to \infty$$

And, if $p_x = 0.5$, $1 - p_x = p_x$

$$\frac{(1-p_x)^k}{p_x^k+(1-p_x)^k} = \frac{1}{2}$$

Finally when $k \to \infty$, the area of the problematic region is $2 * \frac{(\frac{1}{2}*\frac{1}{2})}{2} = (\frac{1}{2} * \frac{1}{2}) = \frac{1}{4}$
□

We see that the more equal attributes we consider, and the more the final decision for x will lies only on $p_x = p(1|x_1)$. This is not surprising since we gave more and more importance to that feature which is repeated infinitely. Hence, we have seen here that even when the features are strongly correlated, the naive Bayesian classifier could achieve as satisfying predictions as the optimal classifier under the 0-1 loss in many cases. This is surprising since the naive assumption seems at first, to be the key of the efficiency of the Bayesian classifier and yet, even without it, it is still performant. In the end, this perception is maybe naive itself and the author of the paper have showed that this assumption is not the most important one for the prediction.

2 Necessary and sufficient conditions for the optimality of the naive Bayesian classifier

In the paper, the authors first establish a review of the existing literature on the naive Bayesian classifier to motivate their initial claim that it may be an optimal predictor on problems involving strongly dependent attributes. Most notably they cite John and Langley [2] who showed that the naive Bayesian classifier limited performances in many domains was due to unwarranted Gaussian assumptions for continuous attributes rather to a particular problem of the classifier. Additionally, through a large scale empirical study on twenty-eight data sets from the UCI repository (Merz, Murphy Aha, 1997), they compare the naive Bayesian classifier to other state-of-the-art classifiers. They have discovered that the Bayesian classifier performed better than one would expect, even when the independence assumption is violated. This observation leads them to searching for a new theoretical understanding of the Bayesian classifier's optimality.

2.1 The fundamental theorem of the article studied

The most important theorem showed in this article is given below

For any classification problem between 2 classes '+' and '-', the Bayesian classifier is locally optimal under the zero-one loss in half the volume of the possible values of (p(x),r(x),s(x)) where:

$$p(x) = p(+|x)$$
$$r(x) = \mathbb{P}(+)\prod_{i=1}^{D} p(x_i|+)$$
$$s(x) = \mathbb{P}(-)\prod_{i=1}^{D} p(x_i|-)$$

This result shows that in the general two-class context, whatever the naive assumption is respected or not, at the end, the naive Bayesian classifier still remains performant in many cases. In other words, this classifier can be optimal under others conditions than the naive one. This is a really satisfying result.

Nevertheless, the optimality under the mean squared error is ensured only in the case where the naive assumption holds since as we said before, this loss function is sensible to the ability of the classifier to well estimate the posterior probabilities.

In practice, this theorem arises from the following one :

For a two-class classification problem, the naive Bayesian classifier is locally optimal under the zero-one loss for an observation x iff

$$(p(x) \geq \frac{1}{2} \wedge r(x) \geq s(x)) \vee (p(x) \leq \frac{1}{2} \wedge r(x) \leq s(x))$$

Proof :

In the two class context, one observation x can belong to the class '+' or the class '-'. Furthermore, we know that the Bayesian classifier is locally optimal under the 0-1 loss when this loss is minimal, i.e. when the correct class for x is predicted.

In the case where $p(x) = \mathbb{P}(+|x) \geq \frac{1}{2}$, the most probable class is '+' and this prediction is correctly provided by the Bayesian classifier when $r(x) \geq s(x)$.

In the same way, in the case where $p(x) = \mathbb{P}(+|x) \leq \frac{1}{2}$, the most probable class is '-' and this prediction is correctly provided by the Bayesian classifier when $r(x) \leq s(x)$.

□

The paper doesn't mention it, but considering the previous proof this theorem could be extended in the k-classes context:

For any classification problem between k classes, we note :

$$p_k(x) = p(\mathcal{C}_k|x)$$
$$r_k(x) = P(\mathcal{C}_k)\prod_{i=1}^{D} p(x_i|\mathcal{C}_k)$$

In that context, The naive Bayesian classifier is locally optimal under the zero-one loss for an observation x if :

$$\bigvee_{i=1}^{k} (p_i(x) = \max_j p_j(x)) \wedge (r_i(x) = \max_j r_j(x))$$

Now, as the authors explained, this condition can't be verified in practice since it implies that we have the knowledge of the true probabilities $p_k(x)$. By the way, even in this case, as they say, the computation will be infeasible due to the necessity to consider a very large number of possibilities (which we are precisely trying to avoid using a naive Bayesian classifier).

2.2 The limited capacity of the NBC for information storage :

Even if we have seen previously that the naive Bayes classifier could be optimal under the 0-1 loss even when the naive assumption is violated. This optimality is however limited by some conditions.

One of them is the limited capacity of this classifier for information storage.

Let d be the number of numbers representable in a certain machine, a be the number of attributes of an observation x, v the maximal number of values taken by an attribute and c the number of classes.

Hence, in that case, the authors of the article have provided the following theorem :

The Bayesian classifier can't be globally optimal for more than $d^{c(av+1)}$ problems.

This imply that the naive Bayesian classifier is not optimal when we deal with a very wide class of problems. In practice, this remark is true for any classifier since the problem of the limited information storage is shared by all of them.

Proof:

In order to make a prediction for an observation x, the naive Bayes classifier needs to approximate the probabilities $\mathbb{P}(\mathcal{C}_k)$ and $p(x|\mathcal{C}_k)$.

Hence, at the end, there is at most : $\underbrace{c}_{\mathbb{P}(\mathcal{C}_k)} + \underbrace{c * (av)}_{p(x|\mathcal{C}_k)} = c(av+1)$ probabilities to estimate.

But, we can represent each of these probabilities only with d different values. Thus, the Bayesian classifier can't distinguish between more than $d^{c(av+1)}$ problems.

□

Thanks to the previous proof, we can retrieve the capacity of the naive Bayesian classifier for information storage. We see that, at the end, we will need to store $c * (av+1) = O(a)$ estimates in order to make a decision for any new observation.

In general, this capacity could be easily exceeded in large dimension. Indeed, considering that the number of possible observations is $O(v^a)$, if we work with many attributes which can take many values, the capacity will be completely exceeded.

In contrast to the NBC, there are others classifiers such as decision trees which have a memory size proportional to the sample size n. In that case, in theory, they should converge to the optimal when $n \to \infty$. However, as it is explained in the article studied, in practice we work with training set of finite size and we don't have all the observations possible since the goal is to learn and not to memorize. Hence, there is also a risk to exceed their capacity when the number of attributes a become too large. This could be seen as a result of the **Curse of dimensionality**.

2.3 Behaviour with nominal attributes

In the case we work with nominal attributes, we estimate the probabilities using Bernoulli or multinomial law. Due to that, the decision boundaries will be hyper-planes. One way to see it could be simply to observe the log-likelihood of the Bernoulli law $\mathcal{B}(\theta)$:

$$l_\theta(x) = \log(\theta^x(1-\theta)^{1-x}) = x\log(\theta) + (1-x)\log(1-\theta)$$

We see that we obtain the equation of an hyperplane in the set of possible observations x.

Due to that, the Naive Bayes classifier can't be optimal for problems with nominal attributes involving classes which can't be separated using hyperplanes as decision boundaries. Nevertheless, if we work with numeric attributes, using a normal law for estimates their distribution could lead to quadratic decision boundaries like in the QDA.

2.4 Optimality for simple set of rules

Finally, the authors showed that the Bayesian classifier is optimal for learning conjunction of literals as well as disjunctions of literals so, very simple rules. It is important to note that in both cases the independence assumption is systematically violated.

Indeed, if we consider for instance a simple conjunction involving only 2 literals : $X_1 \wedge X_2$. This conjunction is True (1) when the two literals are true and False (0) otherwise. In that precise case,

$$\begin{aligned}
\mathbb{P}((0,1)|0) &= \frac{1}{3} \\
\mathbb{P}(X_1 = 0|0) &= \frac{2}{3} \\
\mathbb{P}(X_2 = 1|0) &= \frac{1}{3} \\
&\text{So,} \\
\mathbb{P}((0,1)|0) &\neq \mathbb{P}(X_1 = 0|0)\mathbb{P}(X_2 = 1|0)
\end{aligned}$$

Nevertheless, we consider that this result is very poor since:

- The set of rules considered are very simple. For instance, for a conjunction made of k literals, $2^k - 1$ observations are labeled 0 and only one is labeled 1. In such situation there is maybe no need to use any algorithm for the prediction. Indeed, if k is low (<5 for

instance), the number of observations possible is also low and there is no need to learn anything. On the contrary, when k is big, predicting 0 always is clearly a paying strategy. In practice, encountering such a particular situation is not very probable.

- Their demonstration lies on the fact that all the observations are represented in the dataset (knowledge of all the truth table) and are present only once (they are equally likely). This seems very unlikely to meet such 'perfect' conditions.

In order to test a bit more how the NBC can perform using some set of rules, we have done a little experiment.

We have first considered a situation in which we are trying to learn a conjunction of 10 literals. This was done as a baseline, in order to see that, in practice, the NBC is indeed optimal under the 0-1 loss for this type of problem in the context described in the article. We have compared its performance to others classifiers : LDA, 3-NN, and a decision tree and we have observed as expected that the NBC was optimal under the 0-1 loss. The decision tree was also optimal and the two others classifier didn't succeeded in capturing the only assignment to the value 1 for the observation $\underbrace{(1, 1.., 1)}_{10}$.

Then, we have considered a slightly more complex set of rules using the following disjunction of conjunctions :

$$(x_1 \wedge x_2 \wedge x_3) \vee (x_4 \wedge x_5) \vee (x_6 \wedge x_7 \wedge x_8 \wedge x_9) \vee x_{10}$$

This time, the NBC wasn't optimal under the 0-1 loss and was outperformed by the decision tree and the 3-NN. The optimal classifier under the 0-1 loss was the decision tree without surprise here. Indeed, a decision tree is constructed precisely for the search of a set of rules of whatever nature. In fact, more globally, a decision tree should be optimal under the 0-1 loss for any decision which is taken under a precise set of rules.

At the end, the new set of rules chosen was still not as complex as we may find in the nature and thus, the NBC risks to not be appropriate in general when we are searching the best prediction possible.

Nevertheless, even if a decision tree can perform better than the NBC, its computational cost can be very important when we work with a large training set compared to the one of the NBC in the same context.

So, we have seen at the end that the naive Bayesian classifier can be optimal under the zero-one loss when the independence assumption is broken in very simple cases.

2.5 Efficiency of the NBC on small datasets

When we consider a classifier, we are generally interested into two kinds of errors that it could make according to a problem. The **bias** is the error due to too simple assumptions compared to the complexity of the model and the **variance** is the error due to the sensitivity of the noise present in the training set.

In practice, as said before, the 0-1 loss measures how far the attribution of the classes is correct, despite of the incorrectness on the posterior probabilities. Hence, whatever the 'poor' hypothesis we make for the prediction of the probabilities, as long as we correctly label the observations, this loss will not be affected at the end. In that case, the 0-1 loss is mainly sensitive to the variance rather than the bias.

In general, due to the naive assumption, the NBC present a large bias and a low variance. On the contrary, due to the complexity of the model behind a decision tree, this classifier present a large variance and a low bias.

Using big datasets, the model have to learn a lot of information and need to be enough complex for a good understanding of the data. That's why in practice, decision trees outperformed the NBC on big datasets. On the contrary, as observed by the author of the article, the NBC can outperform efficient classifier such as decision trees when we work with small data sets because it is enough 'complex' for the understanding of a little amount of data.

We have effectively observed that result in an experiment using various datasets from several sources given in our repo. The naive Bayes classifier was used against others famous classifiers : the LDA, the 3-NN and the decision tree and we splitted systematically our dataset into a training and a test set (70%-30%) using a random seed of 0.

You can see below the accuracies obtained on the test sets for each dataset using all the classifiers:

Dataset	Zoo	Iris	Wine	Glasses	Breast cancer	Pokemon	Banknote	Mushroom
Size	101	150	178	214	569	800	1372	8124
Data Type	Categorical	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous	Categorical
NBC	93.5%	100%	94.4%	46.2%	92.4%	92.9%	83.3%,	80.8%
LDA	96.8%	97.8%	98.1%	63.1%	97.1%	92.5%	97.1%,	94.7%
3-NN	93.5%	97.8%	70.4%	64.6%	91.9%	95%	99.8%,	99.9%
Decision Tree	96.8%	97.8%	94.4%	64.6%	91.2%	95.8%	97.6%	100%

Globally,the performance of the naive Bayes classifier is often among the best for the rather small datasets (n<600) and rather among of the lowest for the others. This seems consistent with the analysis of the paper and, we can also see here that the decision tree seems to be more reliable than the NBC when we use rather big datasets (n>600).

Moreover, in pratice, we have observed many times that using a Gaussian naive Bayes classifier with a dataset made of categorical attributes leads surprisingly often to better accuracies. Nevertheless, choosing a normal law for approximating a categorical attribute isn't very interpretable.

3 Extensions for the Bayesian classifier

For real world application, the paper mentions multiple attempt of extends of the simple Bayesian classifier in order to increase its efficiency. Some propositions of the paper are given below :

- Deleting one of two highly correlated attributes (Langley and Sage [5]). This could be done using a correlation matrix or a statistical test as the Z-score.
- Iteratively joining dependent attribute values using statistical testing to determine whether two attributes are significantly dependent (Kononenko [3]).
- Iteratively joining two attributes such that the estimated accuracy is minimized. This is performed by computing the leave-one-out average cross-entropy for each possible pairs of attributes joined and joining the pair that yields the best accuracy (Pazzani [6]).
- Using a tree like architecture called "recursive Bayesian classifier" to overcome the inability of the naive Bayesian classifier to solve non-linear problems (Langley [4]).

The authors specifically address two of these proposed solutions in an empirical study: Joining the two most dependent attribute values and joining the attribute pair that best improves the estimated accuracy. This study is performed having gained new insight on the theoretical optimality of the naive Bayesian classifier where the independence assumption does not hold. The performance of the extended Bayesian classifiers using these methods were compared using two artificial problems: exclusive OR and parity, each time with either two or six relevant attributes. As well as various UCI data sets, to compare their effectiveness on "natural problems".

The results from this empirical study confirm that, since the dependence of attributes isn't a cause of error of the naive Bayesian classifier, detecting and removing dependent attributes isn't an effective extension of the Bayesian classifier.

4 How to be as optimal as possible under the 0-1 loss with the NBC ?

We have tried to think about some strategies enabling the NBC to obtain better prediction for unknown observations for any dataset. We have implemented them in the notebook present in our Github repo.

Our ideas are the following ones :

- Use a random forest algorithm in order to detect the most discriminant features for the decision and then, only use them for the NBC.
- Random NBC : Create several NBC trained with bootstraped sample of the original one selecting features to use randomly for each and then, make the predictions using a majority vote.

Random forest for eliminating features :

For this experiment, we have used the same datasets as before and we have cutted them into a training and a test set (70%-30%) using a random seed of 0.

We have trained the NBC using all the features at first and then, we trained it only using the k most discriminant features according to the random forest algorithm. k was chosen so that we obtain the best results with the smallest number of features.

You can find below the accuracies on the test set according to the two training phase.

Dataset	Zoo	Wine	Glasses	Breast cancer	Pokemon	Banknote	Mushroom
Size	101	178	214	569	800	1372	8124
Data Type	Categorical	Continuous	Continuous	Continuous	Continuous	Continuous	Categorical
k - features percentage	$10 - 62.5\%$	$6 - 46.15\%$	$2 - 22.2\%$	$6 - 20\%$	$2 - 28.6\%$	$2 - 50\%$	$3 - 13.6\%$
using all the features	93.5%	94.4%	46.2%	92.4%	92.9%	83.3%	80.8%
using the k most discriminant features	93.5%	96.3%	58.46%	93.6%	92.08%	87.1%	86.5%

The **'features percentage'** corresponds to the percentage of features we used in our dataset for the final decision.

As you can see, in the bulk of the cases, we have obtained much better results using this feature selection strategy. Nevertheless, this is not based on the correlation between the features, and so, the k ones which are selected by the random forest classifier could be correlated in principle. Thus, we again see that the NBC can be reliable even when the naive assumption isn't really verified.

Results using the Random NBC :

For this experiment, we have implemented a variant of the random forest algorithm using NBC classifiers. The pseudo-code is the following :

Algorithm 1 Random Naive Bayesian Classifier

1. For $t = 1$ to T :

(a) Draw a bootstrap sample of size N from the training data.

(b) Train a Naive Bayes Classifier C_t to the bootstrapped data, by recursively repeating the following steps :

Select at random p features for the training

Train the current classifier C_t with the N observations using only the p selected features

2. Output the prediction

To make a prediction at a new point

Let $\hat{C}_t(x)$ be the class prediction of the t-th random bayesian classifier

Then $\hat{C}^T_{\mathrm{rb}}(x) =$ majority vote $\left\{\hat{C}_t(x)\right\}_1^T$.

For evaluating this new strategy, we have implemented a leave-one-out cross validation for estimating the generalization accuracies of the models considered. We have evaluated a NBC alone against a Random NBC using the cross validation for all the datasets studied in this report.

The results obtained are presented below:

Dataset	Zoo	Wine	Glasses	Breast cancer	Pokemon	Banknote	Mushroom
Size	101	178	214	569	800	1372	8124
Data Type	Categorical	Continuous	Continuous	Continuous	Continuous	Continuous	Categorical
NBC	89.1%	97.8%	47.5%	93.8%	93.0%	84.0%	92.1%
Random NBC	83.2%	98.3%	50.5%	93.3%	93.4%	72%	84.8%

We have chosen to use 20 NBC for all the datasets, except the biggest one for which we use only 10 NBC due to time constraints

In general, the Random Bayesian classifier seems to improve the prediction of unknown observations on rather small datasets (n<=800). We think, in accordance with the authors, that using this algorithm on small datasets with observations in large dimensions can be efficient. We already see here that the NBC is efficient on the zoo and wine datasets which are the ones which correspond the most to the previous description.

5 Conclusion

The article studied showed that the naive Bayesian classifier does not necessarily require attributes independence to be optimal under the zero-one loss and, that it can be a sound choice when working with small datasets. We have verified these claims through an analysis of both the theoretical and experimental results of the paper, as well as our own experiments and research. We built upon these results with our own empirical study of the naive Bayesian classifier by focusing on the influence of the dataset's size on the accuracy and proposed, an extension for this classifier inspired by the random forest algorithm, *Random NBC*. We believe a more detailed analysis of the random naive Bayesian classifier could lead to the development of an efficient extension to this model. We were notably unable to complete an analysis of the influence of the dependence of the attributes on the predictions, nor were we able to perfectly replicate the methodology used in the studied article due to time constraints. None the less, we agree with the authors that further study of the conditions for optimality of the naive Bayesian classifier is necessary as it's an attractive model for real world applications due to it's simplicity, and interpretability.

References

[1] Pedro Domingos and Michael Pazzani. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss". en. In: (), p. 28.

[2] George H John and Pat Langley. "Estimating Continuous Distributions in Bayesian Classi ers". en. In: (), p. 8.

[3] Igor Kononenko. "Semi-naïve Bayesian classifier". In: vol. 58. Apr. 2006, pp. 206–219. DOI: 10.1007/BFb0017015.

[4] Pat Langley. "Induction of recursive Bayesian classifiers". en. In: *Machine Learning: ECML-93*. Ed. by J. Siekmann et al. Vol. 667. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 153–164. ISBN: 978-3-540-56602-1 978-3-540-47597-2. DOI: 10.1007/3-540-56602-3_134. URL: http://link.springer.com/10.1007/3-540-56602-3_134 (visited on 04/11/2020).

[5] Pat Langley and Stephanie Sage. "Induction of Selective Bayesian Classi ers". en. In: (), p. 8.

[6] Michael J Pazzani. "Searching for Dependencies in Bayesian Classi ers". en. In: (), p. 10.