Robust Discrete Bayesian Classifier Under Covariate and Label Noise

Wenlong Chen, Cyprien Gilet, Benjamin Quost, Sébastien Destercke

Robust Discrete Bayesian Classifier under Covariate and Label Noise

Wenlong Chen[1 [0009−0008−9909−0355]], Cyprien Gilet[1 [0000−0002−0617−0436]], Benjamin Quost[1 [0000−0002−0456−9953]], and Sébastien Destercke[2 [0000−0003−2026−468X]]

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex
surname.name@hds.utc.fr

Abstract. In this paper, we focus on the Discrete Bayesian Classifier (DBC), which discretizes the input space into regions where class probabilities are estimated. We investigate fuzzy partitioning as an alternative to the hard partitioning classically used to discretize the space. We show that our approach not only boosts the DBC's performance and resilience to noise, but also mitigates the loss of information due to discretization. The benefits of soft partitioning are demonstrated experimentally on several synthetic and real datasets.

Keywords: Bayesian classifier · Discretization · Robustness.

1 Introduction

In this paper, we address the problem of learning a classifier from data. Our goal is to train a model to accurately classify any new instance. We are particularly interested in the case where the data are corrupted by noise. More precisely, we study how softening/randomizing a discretized classifier can improve robustness to label or feature noise. We distinguish label noise, which corresponds to mistakes in the training labels, from feature (or attribute) noise [10], referred in the title as covariate noise, which might corrupt training as well as test features.

A wide range of techniques have been proposed to determine a classifier with theoretical performance guarantees. In order to well approximate the Bayes decision rule when processing numeric or mixed features in a high dimensional feature space, a relevant approach consists in partitioning the feature space so as to determine a discrete, nonparametric version of the Bayes classifier (DBC) [8,4,6,12,11]. In a nutshell, this approach discretizes the input space into regions (also called discrete profiles), into which the Bayes classifier is analytically determined by estimating the class frequencies; test instances are classified based on the estimates in the region to which they belong. Discrete profiles can correspond to the regions associated with the leaves of a tree when using supervised decision trees [5,21,13], or to the Voronoi cells derived from a K-means partitioning [18].

Most discretization methods are based on hard partitioning [24]: a test instance is mapped to a unique discrete profile, all the instances assigned to this

profile are assumed to share similar features and are associated to the same predicted class. Discretization has been shown to mitigate the impact of noise or outliers [15]. Yet, the hard partitioning may result in very similar instances in different profiles being assigned different outputs by the subsequent "hard DBC", and in the estimates for a given profile being unaffected by instances close but outside of this profile, thus potentially missing important information. The hard DBC may therefore still suffer from label and covariate noise.

Randomization, as well as softening strategies, have been shown to regularize classifiers while retaining their main features. For instance, some research [22] investigated the use of soft partitioning together with the naive Bayesian classifier with promising results, for a specific classifier with strong assumptions, together with a specific (0/1) loss function. Label smoothing, i.e., replacing hard labels with probabilistic ones, is commonly used in deep learning to reduce overfitting: introducing a small amount of uncertainty during training prevents the model from becoming overly confident, thus enhancing its generalization performance. Label smoothing can also improve robustness to label noise [17].

The goal of this paper is to investigate the use of soft clustering with the DBC, so as to benefit from the theoretical guarantees of this latter while improving classification performances by computing smoother decision boundaries. We first recall the basics of the DBC and discuss its limitations. Through detailed mathematical modeling and algorithms, we explain how soft partitioning can enhance its performances. Building on this, we propose the soft probabilistic DBC (SPDBC), which allows each data point to belong to different classes with a certain probability, thus enabling the model to handle overlapping or noisy data more flexibly. In contrast with [22], our present approach allows to consider any kind of loss/cost function.

Experiments on synthetic and standard real datasets, and in particular with a controlled level of noise, demonstrate how the SPDBC improves classification accuracy in noisy environments while maintaining stability. The experimental results confirm the effectiveness of our proposal in dealing with complex data structures and showcase its potential for practical applications.

The paper is organized as follows. Section 2 provides reminders on the discrete Bayesian classifier, talks about how to make DBC probabilistic(PDBC) and presents some criticisms of this classifier. Section 3 proceeds with our approach to deal with the limitation of DBC, resulting in the so-called soft probabilistic discrete Bayesian classifier (SPDBC). Section 4 reports some experiments, which notably stress the robustness of the SPDBC compared to the DBC when facing noisy data. Section 5 provides some conclusions to this preliminary work, mentioning several possible future directions.

2 Discrete Bayesian Classifier

2.1 Setting

We aim to compute a function $\delta : \mathcal{X} \to \mathcal{Y}$ able to provide, for any instance in the input space $x \in \mathcal{X}$, an estimate (or "guess") $\delta(x)$ of its (unknown) ac-

tual class $Y \in \mathcal{Y} = \{1, \ldots, K\}$. For this purpose, we leverage a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ composed of instances $x_i \in \mathcal{X}$ associated with labels $y_i \in \mathcal{Y}$, assumed to be observations of the actual classes of the instances.

We consider here asymmetrical decision problems, i.e. where the decision costs may not be the same (notably across errors). We assume to have access to a matrix $L = (L_{kl})$, whose general term $L_{kl} \geq 0$ quantifies the cost incurred from predicting $\delta(x) = l$ when the ground truth (i.e., actual class) is k. While it is reasonable to assume $L_{kk} = 0$ for all $k \in \mathcal{Y}$, we may have $L_{kl} \neq L_{lk}$ for some $k \neq l$: for instance, erroneously raising an alarm regarding the condition of a patient may be considered as less harmful (and therefore more acceptable) than failing to detect an actual condition of this same patient.

Bayes' decision strategy. Bayesian decision theory [2,9,20] provides a theoretical solution to this learning problem. Under the assumption that data are generated according to a joint probability $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$, it establishes that the decision strategy minimizing the *expected risk* (or misclassification loss) should be based on the posterior probabilities of the classes and on the misclassification costs:

$$\delta^B(x) = \arg \min_{k=1,\ldots,K} R_k(\delta|x), \quad \text{with} \quad R_l(\delta|x) = \sum_{k=1}^{K} L_{kl}\, \mathbb{P}\left(Y = k|X = x\right). \quad (1)$$

Generative models typically derive the posterior probabilities $\mathbb{P}\left(Y = k|X = x\right)$ from the class-conditional distributions $\mathbb{P}\left(X = x|Y = k\right)$ and the prior probabilities $\pi_k = \mathbb{P}\left(Y = k\right)$ using Bayes' rule:

$$\mathbb{P}\left(Y = k|X = x\right) = \frac{\pi_k \mathbb{P}\left(X = x|Y = k\right)}{\mathbb{P}\left(X = x\right)}, \quad \mathbb{P}\left(X = x\right) = \sum_{l} \pi_l \mathbb{P}\left(X = x|Y = l\right).$$

Therefore, a wide range of approaches aim at estimating the prior probabilities π_k and the conditional distributions $\mathbb{P}\left(X = x|Y = k\right)$ so as to implement the Bayes classifier [15,14,1], for instance via maximum likelihood (ML). While π_k can be estimated using the class frequencies in the training set, $\mathbb{P}\left(X = x|Y = k\right)$ usually requires additional assumptions. Thus, many strategies postulate a (semi-)parametric model for $\mathbb{P}\left(X = x|Y = k\right)$ and focus on estimating the parameters of the distribution, for instance as in discriminant analysis [16].

Discrete Bayesian classifier. When the distributional assumption is not satisfied, the resulting classifier may be a biased estimate of the actual Bayes classifier [15], and thus be far from optimal even if the training sample is large. An alternative then consists in using a nonparametric approach. First, the input space $\mathcal{X}$ is partitioned into T regions or *profiles* $\{\phi_1, \ldots, \phi_T\} = \mathcal{P}$. We introduce the mapping $\Phi : \mathcal{X} \to \mathcal{P}$, which maps any instance x to a profile $\Phi(x)$; we may interchangeably write $\Phi(x) = \phi_t$ or $x \in \phi_t$ whenever the instance x falls into the region ϕ_t (or, put another way, when x corresponds to profile ϕ_t).

Then, the discrete Bayesian classifier (DBC) amounts to estimate the class-conditional distributions $\mathbb{P}\left(X = x|Y = k\right)$ by the fractions of input samples from

a given class falling into the various profiles:

$$\widehat{p}_{kt} = \widehat{\mathbb{P}}\left(\Phi(X) = \phi_t \mid Y = k\right) := \frac{1}{n_k} \sum_{i \in \mathcal{I}_k} \mathbb{1}_{\left\{\Phi(x_i) = \phi_t\right\}}, \tag{2}$$

with $\mathcal{I}_k = \{i \in \{1, \ldots, n\} : Y_i = k\}$ the set of indexes corresponding to instances from class k, $n_k = |\mathcal{I}_k|$ the cardinal of this set and $\mathbb{1}_{\{\}}$ is the indicator function:

$$\mathbb{1}_{\left\{\Phi(x) = \phi_t\right\}} = \begin{cases} 1 & \text{if } x \text{ is assigned to cluster } \phi_t \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Two approximations are made here: one is due to the input space being discretized into profiles, and thus to the class-conditional density $\mathbb{P}\left(X = x | Y = k\right)$ being replaced with the probability of the profile $\mathbb{P}\left(\Phi(X) = \phi_t \mid Y = k\right)$; the second one results from this latter probability being estimated with the relative frequency of the profile ϕ_t in the instances of class k in the training set. Figure 1 displays a 2D example, where the input space is discretized using K-means.

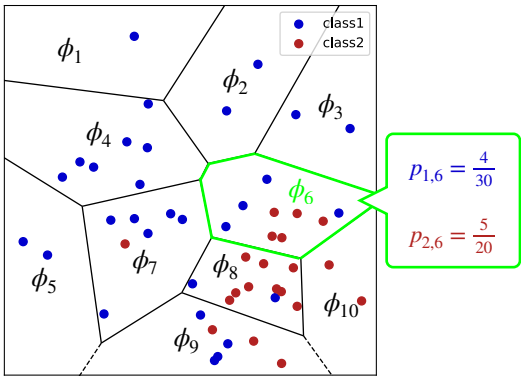


Fig. 1: Bivariate classification problem addressed using the DBC; class 1 (blue) counts $n_1 = 30$ training instances, class 2 (red) $n_2 = 20$ training instances.

The deterministic DBC consists in classifying any instance x into the class which minimizes the expected risk:

$$\delta_\pi^B(x) = \arg\min_{l \in \mathcal{Y}} f_l(\pi, x), \tag{4a}$$

$$f_l(\pi, x) := \sum_{k \in \mathcal{Y}} L_{kl} \sum_{t=1}^{T} \frac{\pi_k \, \hat{p}_{kt}}{\sum_j \pi_j \, \hat{p}_{jt}} \mathbb{1}_{\left\{\Phi(x) = \phi_t\right\}}, \quad \forall l \in \mathcal{Y}. \tag{4b}$$

Notice the similarity of Eq. (4b) with Eq. (1), where the posterior probabilities $\mathbb{P}\left(Y = k | x\right)$ have been replaced with their profile-based estimates

$$\mathbb{P}\left(Y = k | \Phi(x)\right) := \sum_{t=1}^{T} \frac{\pi_k \, \hat{p}_{kt}}{\sum_j \pi_j \, \hat{p}_{jt}} \mathbb{1}_{\left\{\Phi(x) = \phi_t\right\}}. \tag{5}$$

We note that since f_l is a risk associated with a specific class, theoretically the class with smaller f_l is more likely to be less risky as a prediction. We propose to map f_l into a probability of predicting class l, written $\mathbb{P}\left(\delta_\pi^B(x) = l\right)$, so as to derive a probabilistic counterpart to the deterministic DBC presented above. For this purpose, Definition 1 proposes a relative compensation probability assignment technique, that allows each point to belong to different classes with a certain estimated probability reflecting the associated risk. We first calculate a compensatory score λ_l for each class, such that if a class's score f_l is relatively lower, λ_l will be comparatively higher, and vice versa. We then normalize these λ_l values across all classes to calculate the final probabilities, so as to ensure that these probabilities sum up to 1.

Definition 1. *For any initial feature point $x \in \mathcal{X}$, the estimated probability for the discrete Bayes classifier δ_π^B to assign a class $l \in \mathcal{Y}$ is given by*

$$\begin{aligned} \mathbb{P}\left(\delta_\pi^B(x) = l\right) &= \frac{\lambda_l(\pi, x)}{\sum_{k\in\mathcal{Y}} \lambda_k(\pi, x)} \\ \textit{where} \quad \lambda_l(\pi, x) &= \sum_{k\in\mathcal{Y}} f_k(\pi, x) - f_l(\pi, x). \end{aligned} \tag{6}$$

This method naturally balances differences in scores between classes, reducing the undue influence of any single class due to scale discrepancies in scoring. Note that whenever the 0/1 loss function is considered, these probabilities boil down to the profile-based posterior probability estimates defined by Eq. (5).

Example 1. Assume a test instance x in Fig. 1 such that $\Phi(x) = \phi_6$, together with a loss matrix L satisfying $L_{11} = L_{22} = 0$, $L_{21} = 3$, $L_{12} = 2$; we can calculate

$$\begin{aligned} f_1(\pi, x) &= L_{21}\frac{\pi_2\hat{p}_{2,6}}{\pi_1\hat{p}_{1,6} + \pi_2\hat{p}_{2,6}} = 3 \times \frac{\frac{20}{50} \times \frac{5}{20}}{\frac{30}{50} \times \frac{4}{30} + \frac{20}{50} \times \frac{5}{20}} = \frac{5}{3}, \\ f_2(\pi, x) &= L_{12}\frac{\pi_1\hat{p}_{1,6}}{\pi_1\hat{p}_{1,6} + \pi_2\hat{p}_{2,6}} = 2 \times \frac{\frac{30}{50} \times \frac{4}{30}}{\frac{30}{50} \times \frac{4}{30} + \frac{20}{50} \times \frac{5}{20}} = \frac{8}{9}. \end{aligned}$$

Thus, λ_1 and λ_2 are

$$\begin{aligned} \lambda_1(\pi, x) &= (f_1(\pi, x) + f_2(\pi, x)) - f_1(\pi, x) = \frac{5}{3} + \frac{8}{9} - \frac{5}{3} = \frac{8}{9}, \\ \lambda_2(\pi, x) &= (f_1(\pi, x) + f_2(\pi, x)) - f_2(\pi, x) = \frac{5}{3} + \frac{8}{9} - \frac{8}{9} = \frac{5}{3}. \end{aligned}$$

Finally, we obtain

$$\mathbb{P}\left(\delta_\pi^B(x) = 1\right) = \frac{\lambda_1(\pi, x)}{\lambda_1(\pi, x) + \lambda_2(\pi, x)} = \frac{\frac{8}{9}}{\frac{8}{9} + \frac{5}{3}} = \frac{8}{23}, \quad \mathbb{P}\left(\delta_\pi^B(x) = 2\right) = \frac{15}{23}.$$

Proposition 1. *The decision rule of the DBC in Eq.* (4a) *is equivalent to picking the class with highest probability $\mathbb{P}\left(\delta_\pi^B(x) = l\right)$ defined in Eq.* (6)*:*

$$\delta_\pi^B : x \mapsto \underset{l\in\mathcal{Y}}{\operatorname{argmax}}\, \mathbb{P}\left(\delta_\pi^B(x) = l\right). \tag{7}$$

Proof. Let $f_l(\pi, x)$ be the minimum value among the set $\{f_k(\pi, x) : k \in \mathcal{Y}\}$; then,

$$\begin{aligned} & & \forall k \in \mathcal{Y}, \quad & f_l(\pi,x) \leq f_k(\pi,x) \\ \Leftrightarrow \quad & \forall k \in \mathcal{Y}, \quad \forall j \neq l, \quad & f_k(\pi,x) - f_j(\pi,x) \leq f_k(\pi,x) - f_l(\pi,x) \\ & \Leftrightarrow \quad \forall j \neq l, \quad & \underbrace{\sum_{k\in\mathcal{Y}} f_k(\pi,x) - f_j(\pi,x)}_{\lambda_j(\pi,x)} \leq \underbrace{\sum_{k\in\mathcal{Y}} f_k(\pi,x) - f_l(\pi,x)}_{\lambda_l(\pi,x)}. \end{aligned}$$

In other words,

$$f_l(\pi,x) = \min_{k\in\mathcal{Y}} f_k(\pi,x) \quad \Leftrightarrow \quad \lambda_l(\pi,x) = \max_{k\in\mathcal{Y}} \lambda_k(\pi,x).$$

Moreover, given the definition of $\mathbb{P}\left(\delta_\pi^B(x) = l\right)$ in Eq. (6),

$$\lambda_l(\pi,x) = \max_{k\in\mathcal{Y}} \lambda_k(\pi,x) \quad \Leftrightarrow \quad \mathbb{P}\left(\delta_\pi^B(x) = l\right) = \max_{k\in\mathcal{Y}} \mathbb{P}\left(\delta_\pi^B(x) = k\right).$$

To conclude, we have

$$\operatorname*{argmin}_{l\in\mathcal{Y}} f_l(\pi,x) \;=\; \operatorname*{argmax}_{l\in\mathcal{Y}} \mathbb{P}\left(\delta_\pi^B(x) = l\right),$$

and thus $\delta_\pi^B(x) = \arg\min_{l\in\mathcal{Y}} \; f_l(\pi,x)$ and $\delta_\pi^B = \arg\max_{l\in\mathcal{Y}} \; \mathbb{P}\left(\delta_\pi^B(x) = l\right)$ are equivalent. □

This stresses that our probabilized classifier is consistent with Bayes' decision strategy, as it will assign the highest probability to Bayes' optimal decision.

2.2 Impact of discretization

Advantages. As discussed in the Introduction, discretization makes the resulting classifier less sensitive to noise or outliers [15], two issues which may significantly affect parameter estimates of continuous distributions, and therefore impede approximating the actual Bayes' classifier. Last, for large-scale data sets, discretization can significantly reduce the overall computational complexity.

Limitations. As can be seen from Eq. (4b), the value of $f_l(\pi, x)$ is the same for all instances in the same profile. This results in the decision boundary of DBC being solely determined by the boundaries of the discrete profiles. Therefore, the partitioning algorithm may have a significant impact on the performances of the classifier. In particular, two instances close to each other in the input space may nevertheless be associated with different decisions. Fig. 2 displays the profiles obtained for a synthetic 2D dataset: as can be seen, some of them (like the leftmost profile in red) are associated with mixed subsets of training instances, in which case associating the profile with a single class will result in errors.

Besides, linking the instances to the profiles using a "hard" mapping (determined by the indicator function) results in instances close to x in the input

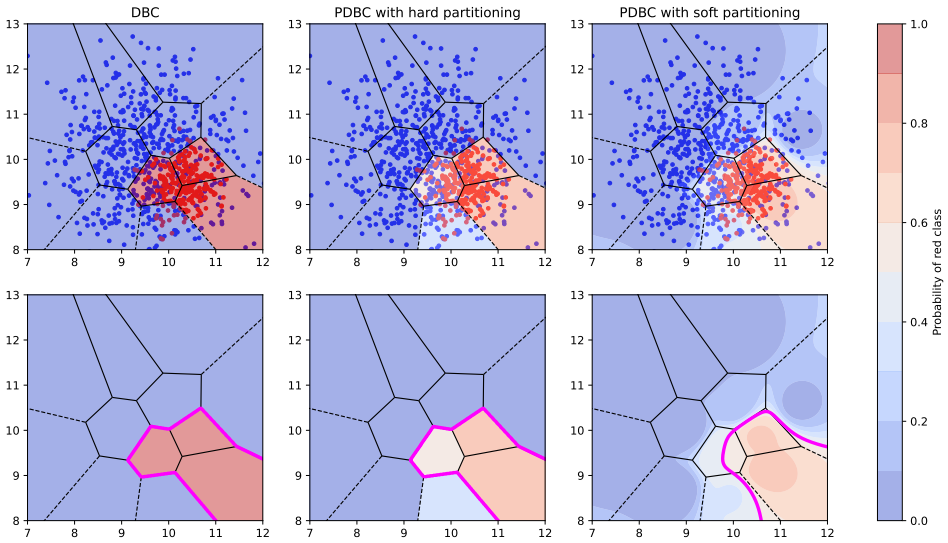


Fig. 2: An example for three different decision regions of DBC, the purple line is the decision boundary

space being associated to different profiles, hence having no influence on the class-conditional risks computed via Eq. (4b). This results in the decision boundary of DBC being a combination of multiple local optimal boundaries between profiles, determined entirely by the instances in these profiles.

These two considerations motivate determining the profile using soft clustering, so that instances can belong to several profiles with different degrees, resulting in DBC decisions being based on more global information.

3 PDBC with Soft Partitioning

We discuss here how soft profile memberships can be leveraged to improve the performances of the DBC. The main idea is to replace the indicator function $\mathbb{1}_{\{\Phi(x)=\phi_t\}}$ in Equation (4a) by a probability $\mathbb{P}(\Phi(x)=\phi_t)$, so as to regularise/soften the boundaries of the clusters. This means that any clustering methods producing probabilities such as Gaussian mixture [23] or other approaches producing probabilities [25] can be relevant.

In our case, we use the fuzzy C-means (FCM) algorithm [3] for this purpose. The choice of FCM is motivated by several reasons. This robust and versatile approach comes with a geometrical interpretation; it provides cluster representatives (the cluster centers), to which profiles can be associated, and based on which cluster memberships can be derived (thus, even among the points that would be associated with a given profile, cluster memberships can vary).

We first introduce the FCM algorithm, and then discuss how it can be combined with PDBC to improve the robustness of the decision boundaries.

3.1 Reminder on the Fuzzy C-means partitioning

Fuzzy C-means (FCM) is a typical fuzzy partitioning method [3] akin to the K-means algorithm. In this algorithm, any instance x is associated with a set of coefficients modelling its degrees of membership to each of the C clusters: in the following, $u_t(x)$ stands for the membership of instance x to the t-th cluster.

Centroids. In FCM, a cluster is characterized by its centroid c_t, defined as the average of all instances weighted by their membership degrees to the cluster:

$$c_t = \frac{\sum_x u_t(x)^m x}{\sum_x u_t(x)^m},$$

with m the hyper-parameter that controls the level of fuzziness of the cluster (the higher m, the fuzzier the cluster).

Algorithm. The FCM algorithm attempts to partition the set of n instances at hand $\{x_1, \ldots, x_n\}$ into a collection of T fuzzy clusters $\{\phi_1, \ldots, \phi_T\}$ as follows. Given a finite set of data, the algorithm returns a list of T cluster centers $C = \{c_1, ...c_T\}$ and a partition matrix $U = u_{it} \in [0,1], i = 1, ..., n, t = 1, ..., T$, where each element u_{it} tells the degree to which element x_i belongs to cluster ϕ_t. The FCM aims to minimize an objective function:

$$J(U, C) = \sum_{i=1}^{n} \sum_{t=1}^{T} u_{it}^m ||x_i - c_t||^2, \tag{8a}$$

$$\text{with } u_{it} = \frac{1}{\sum_{j=1}^{T} (\frac{||x_i - c_t||}{||x_i - c_j||})^{\frac{2}{m-1}}}. \tag{8b}$$

When m is low (close to 1), the cluster memberships become close to binary (0 or 1): FCM then behaves like the K-means algorithm, producing (almost) hard partitions. When m is high, the cluster memberships become fuzzier. The membership degrees of a data point tend to be distributed across different clusters. In this scenario, the cluster boundaries become softer, better capturing the fuzziness and uncertainty in the data.

3.2 Soft PDBC

According to [19], the membership degree $u_t(x)$ of x to the cluster ϕ_t, defined in Eq. (8b), can be interpreted as a posterior probability of x belonging to a ϕ_t given some assumptions, as it satisfies

$$0 \leq u_t(x) \leq 1, \quad \sum_{t=1}^{K} u_t(x) = 1. \tag{9}$$

In the following, given a collection of T fuzzy clusters $\{\phi_1, \ldots, \phi_T\}$ and their associated cluster centers $\{c_1, ...c_T\}$, we define, for all $x \in \mathcal{X}$ and $t \in \{1, \ldots, T\}$,

$$\mathbb{P}\left(\Phi(x) = \phi_t\right) \;=\; u_t(x) \;=\; \frac{1}{\sum_{j=1}^{T} \left(\frac{||x-c_t||}{||x-c_j||}\right)^{\frac{2}{m-1}}}. \tag{10}$$

We propose to substitute these estimated posterior probabilities to the indicator functions in our PDMC model.

Definition 2. *The estimated probability that an instance has the discrete feature profile ϕ_t given that its actual class label is $y = k$, is given by*

$$\hat{p}_{kt} = \frac{1}{n_k} \sum_{i \in \mathcal{I}_k} \mathbb{P}\left(\Phi(x_i) = \phi_t\right) \tag{11}$$

with $\mathcal{I}_k$ and n_k defined as in Section 2.1 and $\mathbb{P}\left(\Phi(x_i) = \phi_t\right)$ as in Eq. (10)*.*

Definition 3. *Under Definition 2, the risk f_l of predicting class $l \in \mathcal{Y}$ given a feature instance $x \in \mathcal{X}$ initially defined by Eq.* (4b) *becomes*

$$f_l(\pi, x) := \sum_{k \in \mathcal{Y}} L_{kl} \sum_{t=1}^{T} \frac{\pi_k \, \hat{p}_{kt}}{\sum_j \pi_j \, \hat{p}_{jt}} \, \mathbb{P}\left(\Phi(x) = \phi_t\right). \tag{12}$$

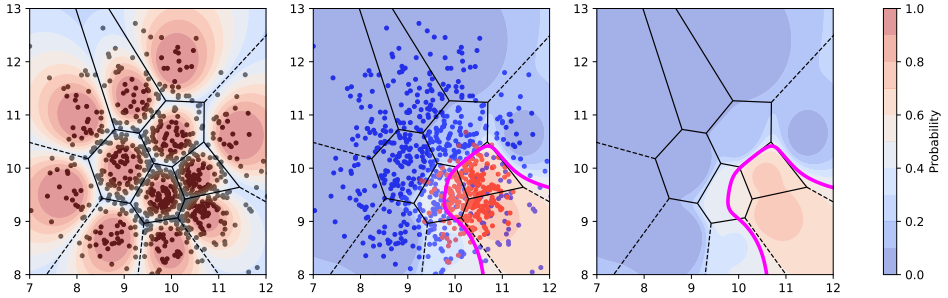


Fig. 3: Soft probabilistic DBC obtained using FCM: (left) probability that each position belongs to the current profile, (center and right) probability that each position in the feature space belongs to the red class.

Each data point may thus belong to various profiles with specific probabilities, allowing the model to consider the neighboring information in different profiles, and providing a softer decision boundary which is not constrained by the profile boundaries, as can be seen in Fig. 2. This will mitigate the impact of a crude partitioning, and arguably result in better generalization performances.

4 Experiments

The purpose of our experimental study is to validate the robustness and accuracy of our proposed model when dealing with noise. We use eight datasets from the UCI Machine Learning Repository: Iris, Breast Cancer, Diabetes, Heart Disease, Raisin, Zoo, Glass, Energy. For each dataset, we conduct two experiments. The code is available in our `Github repository: Menamot/SUM-experiments`.

Label Noise. First, we add label noise to the training set on Y with levels $s \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$, meaning that a fraction s of training labels are switched at random according to a uniform distribution on $\mathcal{Y} \setminus y_i$. We then compute the accuracy on the test set for each noise level, via 20 times 5-fold cross validation. The purpose of this experiment is to study whether the model can learn accurate decision boundaries from imperfectly-labeled training instances, which may occur whenever the data are manually labeled.

Covariate Noise. In a second step, we add noise to the features X_j in the test set according to a Gaussian distribution:

$$X_j^{\text{noise}} = X_j + \epsilon_j \quad \text{with} \quad \epsilon_j \sim \mathcal{N}\left(0, \sigma_j^2\right), \quad \sigma_j = \text{std}(\text{X}_\text{j}) \times s, \tag{13}$$

and with $s \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. This experiment aims at testing the robustness of the model to feature noise; whereas training data can be collected carefully, using high-quality sensors or by skilled operators, it is often not the case for test data, which are generally processed without additional treatment.

Dataset	**nb. samples**	**nb. classes**	**class frequencies**	**nb. features**
Iris	150	3	[0.33, 0.33, 0.33]	4
Breast Cancer	569	2	[0.37, 0.62]	30
Diabetes	768	2	[0.65, 0.35]	8
Heart Disease	303	2	[0.54, 0.46]	13
Raisin	900	2	[0.5, 0.5]	7
Zoo	101	7	[0.41, 0.20, 0.05, 0.13, 0.04, 0.08, 0.10]	16
Glass	214	6	[0.33, 0.36, 0.08, 0.06, 0.04, 0.14]	9
Energy	358	6	[0.31, 0.17, 0.20, 0.13, 0.13, 0.05]	34

Table 1: Datasets used in the experiments.

Partitioning methods. We use K-means as hard clustering approach, and Fuzzy C-means for soft partitioning. In order to remove the influence of the two partitioning methods on the profile position, we set the cluster centers in FCM to be the same as for K-means, which means that the profiles of the two clusters will be exactly the same, except that C-means allows data to belong to different profiles with specific degrees of membership [3], calculated using Eq. (10). In this way,

we can determine how replacing hard profile memberships with soft ones affects the results of the DBC. Note that since the cluster centers are defined as the K-means centers, the FCM algorithm only needs to calculate these membership degrees: the runtime of the two models is therefore almost identical to that of K-means. We select hyper-parameters using 10×10 cross validation.

Results. Figures 4 and 5 show that our model is indeed more robust than the original DBC. As discussed in Sections 2.2 and 3.2, we can see that our model exhibits better performances than the original DBC, even when the data are not corrupted by noise. This clearly demonstrates how our soft clustering-based approach improves the robustness and accuracy compared to the traditional hard clustering-based DBC, especially in presence of both label and covariate noise.

As *label noise* increases, soft clustering consistently outperforms hard clustering for all datasets. For instance, in the Iris and Breast Cancer datasets, the performance of soft clustering remains more stable, with accuracy decreasing less significantly compared to hard clustering when noise levels reach 0.25. This highlights the ability of soft clustering to mitigate the effects of noisy labels, resulting in smoother decision boundaries and improved robustness in classification.

Similarly, when *covariate noise* is introduced (with varying levels of standard deviation), soft clustering maintains a higher accuracy across the datasets. Even in the worst-case scenarios, such as with the Raisin and Zoo datasets, the performance of SPDBC is on par with the standard DBC—we speculate that this is due to the boundaries between the classes being already clear enough, thereby allowing K-means to effectively separate instances from different classes. In such cases, refining the decision boundaries such as with the SPDBC no longer results in a significant increase in accuracy.

Overall, these experimental results validate the effectiveness of soft probabilistic discrete Bayesian classifiers in noisy environments. Soft clustering improves the robustness of the model, resulting in a higher classification accuracy even in challenging conditions, such as when data are noisy or uncertain.

5 Conclusion

In this paper, we explored the capabilities of discrete Bayesian classification and its adaptation using soft partitioning techniques to address challenges posed by imperfect datasets. Our proposed approach, called soft probabilistic discrete Bayes classifier, allies the efficiency of discrete Bayesian classification with the flexibility of probabilistic profiles, thereby enhancing the classifier's effectiveness in practical scenarios.

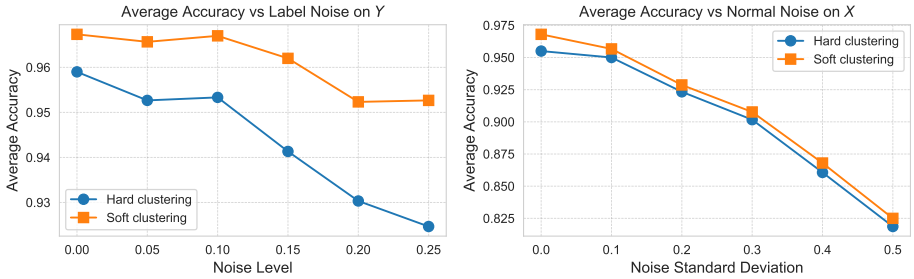
Our experimental results confirm that our approach not only improves the classification accuracy in noisy environments, but also maintains stability across different data distributions, making it a valuable tool for applications where robustness to noise is crucial. Additionally, the probabilistic approach of soft partitioning within the discrete Bayesian framework helps achieving more regular decision boundaries, which are essential for complex class structures.

Future work may explore further enhancements to the SPDBC model. We may for instance use advanced machine learning algorithms that can dynamically adjust the partitioning granularity based on data complexity and distribution shifts. We may also use neural networks to apply our soft probabilistic minimax approach to advanced image classification. Another interesting line of research would be to robustify this softening strategy, for instance by considering clustering methods delivering not a single probability for each instance but a set of probabilities, such as evidential clustering approaches [7].

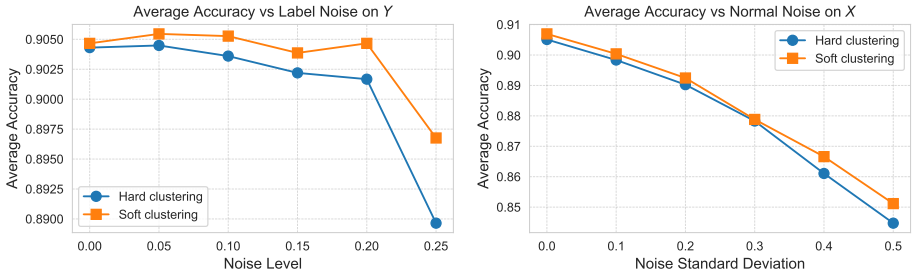
References

1. Ahmad, A., Quegan, S.: Analysis of maximum likelihood classification on multispectral data. Applied Mathematical Sciences **6**(129), 6425–6436 (2012)
2. Berger, J.O.: Statistical decision theory and Bayesian analysis; 2nd ed. Springer Series in Statistics, Springer, New York (1985)
3. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. Computers & geosciences **10**(2-3), 191–203 (1984)
4. Braga-Neto, U., Dougherty, E.R.: Exact performance of error estimators for discrete classifiers. Pattern Recognition **38**(11), 1799–1814 (2005)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman and Hall/CRC, 1st edn. (1984)
6. Dalton, L.A., Dougherty, E.R.: Bayesian minimum mean-square error estimation for classification error - part i: Definition and the Bayesian mmse error estimator for discrete classification. IEEE Transactions on Signal Processing **59**, 115–129 (2011)
7. Denœux, T., Masson, M.H.: Evclus: evidential clustering of proximity data. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **34**(1), 95–109 (2004)
8. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer-Verlag New York, 2nd edn. (1996)
9. Ferguson, T.: Mathematical Statistics : A Decision Theoretic Approach. Academic Press (1967)
10. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE transactions on neural networks and learning systems **25**(5), 845–869 (2013)
11. Gilet, C.: Discrete minimax classifier for personalized diagnosis in medicine. PhD Thesis, Université Côte d'Azur (2021), `https://tel.archives-ouvertes.fr/tel-03553934`
12. Gilet, C., Barbosa, S., Fillatre, L.: Discrete box-constrained minimax classifier for uncertain and imbalanced class proportions. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(6), 2923–2937 (2020)
13. Gilet, C., Guyomard, M., Barbosa, S., Fillatre, L.: Adjusting Decision Trees for Uncertain Class Proportions. In: Workshop on Uncertainty in Machine Learning at ECML/PKDD 2020 (2020), `https://sites.google.com/view/wuml-2020/program`
14. Grossman, D., Domingos, P.: Learning bayesian network classifiers by maximizing conditional likelihood. In: Proceedings of the twenty-first international conference on Machine learning. p. 46 (2004)
15. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. arXiv preprint arXiv:1302.4964 (2013)

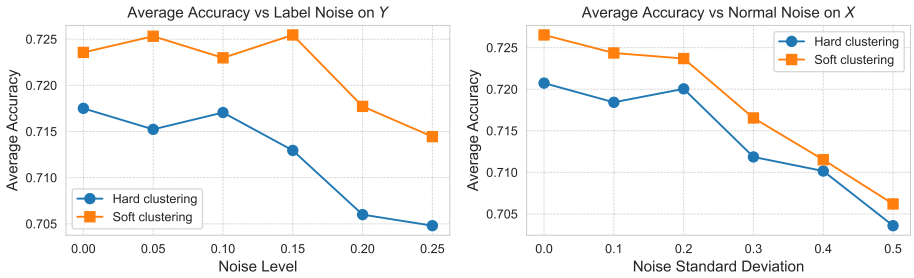
16. Klecka, W.R.: Discriminant analysis. Sage (1980)
17. Lukasik, M., Bhojanapalli, S., Menon, A., Kumar, S.: Does label smoothing mitigate label noise? In: International Conference on Machine Learning. pp. 6448–6458. PMLR (2020)
18. MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability pp. 281–297 (1967)
19. Mencar, C., Castiello, C.: A bayesian interpretation of fuzzy c-means. In: Conference of the European Society for Fuzzy Logic and Technology. pp. 443–454. Springer (2023)
20. Rish, I., et al.: An empirical study of the naive bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence. vol. 3, pp. 41–46. Seattle, WA, USA; (2001)
21. Scott, C., Nowak, R.D.: Minimax-optimal classification with dyadic decision trees. IEEE Transactions on Information Theory **52**(4) (2006)
22. Tang, Y., Pan, W., Li, H., Xu, Y.: Fuzzy naive bayes classifier based on fuzzy clustering. In: IEEE International Conference on Systems, Man and Cybernetics. vol. 5, pp. 6–pp. IEEE (2002)
23. Yang, M.S., Lai, C.Y., Lin, C.Y.: A robust em clustering algorithm for gaussian mixture models. Pattern Recognition **45**(11), 3950–3961 (2012)
24. Yang, Y., Webb, G.I.: A comparative study of discretization methods for naive-bayes classifiers. In: Proceedings of PKAW. vol. 2002 (2002)
25. Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 1, pp. 294–301. IEEE (2005)



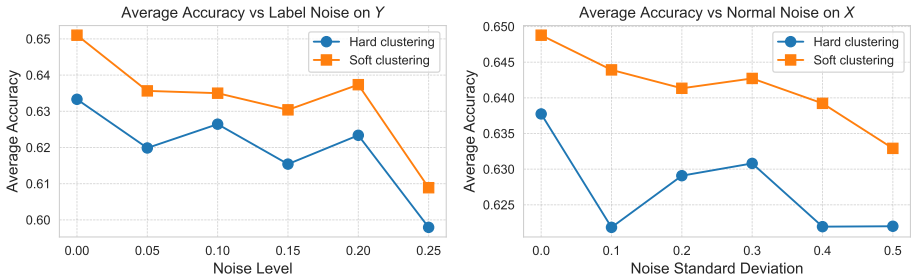
(a) Results, Iris dataset



(b) Results, Breast Cancer dataset

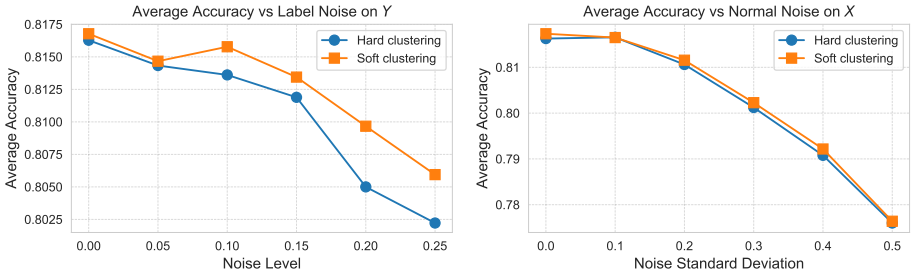


(c) Results, Diabetes dataset

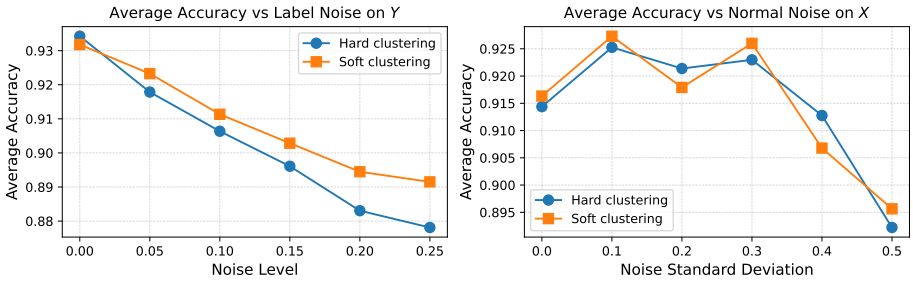


(d) Results, Heart Disease dataset

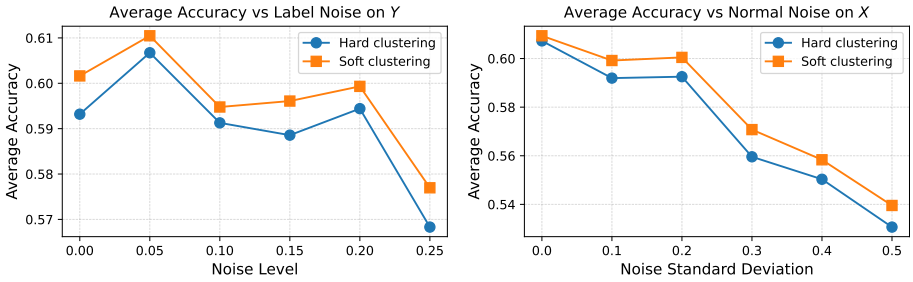
Fig. 4: Experimental results on different data sets: from top to bottom, Iris, Breast Cancer, Diabetes and Heart Disease.



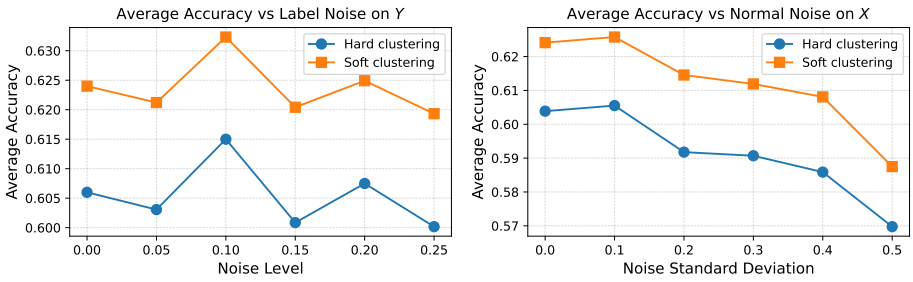
(a) Results, Raisin dataset



(b) Results, Zoo dataset



(c) Results, Glass dataset



(d) Results, Energy dataset

Fig. 5: Experimental results on different data sets: from top to bottom, Raisin, Zoo, Glass, and Energy.