Decision-making from Partial Test Instances by Active Feature Querying

Benjamin Quost BENJAMIN.QUOST@HDS.UTC.FR
UMR UTC-CNRS 7253 Heudiasyc, Université de Technologie de Compiègne, France

Abstract

We consider a classification problem in which test instances are not available as complete feature vectors, but must rather be uncovered by repeated queries to an oracle. We have a limited budget of queries: the problem is then to find the best features to ask the oracle for. We consider here a strategy where features are uncovered one by one, so as to maximize the separation between the classes. Once an instance has been uncovered, the distribution of the remaining instances is updated according to the observation. Experiments on synthetic and real data show that our strategy remains reasonably accurate when a decision must be made based on a limited amount of observed features. We briefly discuss the case of imprecise answers, and list out the many problems arising in this case.

Keywords: Partially supervised decision making; robust decision making; active learning.

1. Introduction

In a typical supervised classification setting, a classifier is trained to identify the class of instances drawn from a population of reference. Classically, instances are interpreted as the realization of a random vector $X \in \mathbb{R}^p$, whose elements are the features describing the instance. In order to classify any new (test) instance, Bayesian decision theory [5, 14] advocates estimating the posterior probabilities of the classes, for which two main approaches can then be deployed. Discriminative approaches aim at estimating these probabilities directly, for instance by assuming a parameterized model which has to be estimated from training data; generative approaches rather estimate the class-conditional distributions of the feature vector and the class frequencies from the training data [13]. Then, for a new instance x, the posterior probabilities are deduced using Bayes rule.

In both cases, it is obvious that making a decision requires the test instance to be perfectly known. In some applications, however, instances may only come partially observed or unobserved. For instance, in medical diagnosis, nothing is known about a patient before a diagnosis is performed: uncovering some features may require performing time-consuming, invasive or potentially hazardous tests, which should therefore be avoided if possible.

In this work, test instances are assumed to be unobserved: the missing features must be uncovered in order to make a decision. This problem differs from imputation since values are uncovered and not estimated. It is related to *active learning* [6, 11, 3, 15, 16], which however occurs during training, and generally aims at uncovering information regarding the class variable so as to improve the classification model being trained. Note that in the two last references [15, 16], the problem of uncovering missing feature values was addressed, and the proposal is close in spirit to the strategy presented here. In [10], the problem of processing partial test instances was addressed. The missing features were assumed to be always the same; two models were available (one based on the features always available, the other on all features), and the issue was then to identify which instances should be completed and which may be left incomplete.

Here, we investigate an iterative strategy where the feature values of a test instance are retrieved progressively, with the purpose of increasing evidence pointing towards one of the possible classes, by exploiting the model inferred in the training step. An oracle is assumed to provide this information, in the form of answers regarding the actual value taken by features. We mainly consider the case of precise answers provided with respect to a single feature — i.e., each answer consists in a single realization for one of the variables describing the test instance. We nevertheless discuss the case of imprecise answers (sets of values being then provided as answers) and point out the main issues caused by such partial pieces of information.

The article is organized as follows. Section 2 briefly recalls the setting and provides some notations. In Section 3, we describe an approach where features are uncovered based on their expected tendency to determine the class information. Section 4 presents some experiments on synthetic and real data. Section 5 briefly discusses the case of imprecise answers, and the implications on the strategy proposed. Eventually, Section 6 concludes and presents the many directions in which future work may be conducted.

2. Setting

2.1. Classical Decision Making

A test instance x has to be classified into one of $K \geq 2$ classes $\Omega = \{\omega_1, \ldots, \omega_K\}$; it is assumed to be the realization of a random vector $X \in \mathscr{X} = \mathbb{R}^p$, the (unknown) class information being encoded by a random variable $Z \in \Omega$.

We will place ourselves in the case of a well-known generative model: the prior probabilities $\pi_k = \Pr(Z = \omega_k)$ and class-conditional densities $f_k = f_{X|Z=\omega_k}$ are estimated using training data.

When a new (test) instance x is observed, Bayesian decision theory recommends that the class $\omega^\star$ with highest posterior probability be chosen:

$$\omega^\star = \arg\max_{k=1,\ldots,K} \Pr(\omega_k|x), \tag{1}$$

with

$$\Pr(\omega_k|x) = \frac{\pi_k f_k(x)}{\sum_\ell \pi_\ell f_\ell(x)}. \tag{2}$$

In other terms, making a decision amounts to determining the class which dominates the others in terms of posterior probability, which we will write $\omega_k \succ_p \omega_\ell$:

$$\omega_k \succ_p \omega_\ell \Leftrightarrow \Pr(\omega_k|x) > \Pr(\omega_\ell|x) \Leftrightarrow \frac{f_k(x)}{f_\ell(x)} > \frac{\pi_\ell}{\pi_k}. \tag{3}$$

2.2. Decision Making from Partial Instances

The problem addressed in this work is that the test instances x_i are only partially observed. For each instance, we decompose the feature vector into two parts: $x = (x_O, x_M)$, with $O \subseteq \{1,\ldots,p\}$ is the subset of indices of the observed variables, and $M \subseteq \{1,\ldots,p\}$ stands for the unobserved ones (obviously, O and M form a partition of $\{1,\ldots,p\}$). Whenever $M \neq \emptyset$, the conditional densities $f_k(x)$ cannot be computed any more. Then, should a decision be made using the observed vector x_O alone, the missing variables can be marginalized out:

$$f_k(x_O) = \int_{\mathcal{X}_M} f_k(x) dx_M. \tag{4}$$

Obviously, the observed feature vector x_O may not be sufficient in order to make an accurate decision. In this case, one may consider uncovering (some of) its missing part x_M, in order to increase the amount of information based on which the decision is to be made.

Rather than choosing the features to be uncovered in a single shot, we propose to do it progressively, exploiting the information retrieved in each step for making the next choices. For this purpose, we repeatedly query the oracle for missing values, and we update the information regarding the class-conditional densities for the feature vector and therefore the posterior probabilities of the classes.

3. Iterative Feature Querying

3.1. Global Strategy

Iterative approach We consider a test instance with missing features which needs to be classified. An oracle is able to provide the missing values. We propose to uncover features iteratively. At each step $t = 0, 1, 2, \ldots$, we consider the sets of observed features O_t and missing features M_t (again, forming a partition of $\{1,\ldots,p\}$). The process, roughly described in Algorithm 1, consists in picking (sets of) variables Q_t, so as to transfer them from M_t into O_{t+1}.

Algorithm 1: Iterative querying process

Input: model; test instance with observed ($O_0 = \emptyset$) and missing ($M_0 = \{1,\ldots,p\}$) features
Output: Updated sets of features O_t and M_t

```
t ← 0;
while querying for new features is still possible do
    t ← t + 1;
    identify a new part x_{Q_t} of x to be uncovered;
    update the sets of features: O_t ← O_{t−1} ∪ Q_t,
      M_t ← M_{t−1} \ Q_t;
    update the available information over the
      remaining features in M_t using x_{Q_t};
end
return sets of features O_t and M_t, observed vector x_{O_t}
```

Assume no information regarding the instance to be classified is available: $O_0 = \emptyset$ and $M_0 = \{1,\ldots,p\}$. Should a decision be made, the class with highest prior probability would be chosen. Making the best decision would instead require uncovering all p variables, i.e. $Q_1 = \{1,\ldots,p\}$; and choosing a class according to Equation (3). Imagine that this is not possible, due to resource constraints: only a subset of variables with indices Q_1 can be uncovered in a first step: $O_1 \leftarrow Q_1$ and $M_1 \leftarrow \{1,\ldots,p\} \setminus Q_1$. Then, either new variables can be queried for; or a decision can be made based on the available information.

Decision making Should the processed be stopped at some step t and a decision be made based on O_t, the missing variables can be marginalized out:

$$\begin{aligned}\Pr(\omega_k|x_{O_t}) &= \frac{\Pr(\omega_k, x_{O_t})}{\Pr(x_{O_t})} \\ &= \frac{f_k(x_{Q_t}|x_{O_{t-1}}) \Pr(\omega_k|x_{O_{t-1}})}{\sum_\ell f_\ell(x_{Q_t}|x_{O_{t-1}}) \Pr(\omega_\ell|x_{O_{t-1}})}.\end{aligned} \tag{5}$$

Note that this latter expression is that of Equation (2) where prior probabilities π_k were replaced with posterior probabilities $\Pr(\omega_k|O_{t-1})$. Alternatively, we could check whether $\omega_k \succ_p \omega_\ell$, for all $\ell \neq k$, as in Equation (3):

$$\omega_k \succ_p \omega_\ell \quad \Leftrightarrow \quad \frac{f_k(x_{Q_t}|x_{O_{t-1}})}{f_\ell(x_{Q_t}|x_{O_{t-1}})} > \frac{\Pr(\omega_\ell|x_{O_{t-1}})}{\Pr(\omega_k|x_{O_{t-1}})}.$$

Feature choice strategy The problem of choosing features to query for is vast: how many features should be uncovered, which ones, in which order. The main issue is obviously that of the choice criterion, that is, a measure of

informativeness of each feature with respect to the task at hand (here, classification).

We propose here to query for the single feature X_{q_t} [1] which appears to be the most informative, repeatedly if resources allow for it. Uncovering subsets of features has been studied in a similar setting [4]; it will not be addressed here. It can nevertheless be seen as a direct generalization of the approach presented in this article, with two notable caveats: (a) evaluating the informativeness criterion over all possible subsets of features is combinatorial; and (b) uncovering larger pieces of information at once limits the interest of exploiting the information x_{Q_t} retrieved at a given step in order to choose the next query Q_{t+1}.

3.2. Informativeness Criterion

Intuitively, in the iterative querying strategy described above, the variables to query for should be chosen according to the information it will supposedly bring in order to choose between the classes in presence. Obviously, a variable is informative if it weighs heavily in predicting the output variable Z. Therefore, having previously observed the values in x_O, we propose to choose the variable X_q [2] which has the greatest influence on Z, by minimizing the *conditional entropy* $H(Z|X_q,x_O)$ [12]:

$$H(Z|X_q,X_O=x_O) = -\sum_{k=1}^{K}\int_{\mathscr{X}_q}\Pr(\omega_k,x_q|x_O)\log\frac{\Pr(\omega_k,x_q|x_O)}{f_{X_q|X_O=x_O}(x_q)}dx_q. \quad (6)$$

Note that this conditional entropy, which indicates to which extent the outcome of the class variable Z is determined by the random variable $X_q|x_O$, can be re-expressed as the *expected differential entropy* of $Z|x_q,x_O$, which turns out to be the (discrete) entropy of the class posterior probability distribution $\Pr(Z|x_q,x_O)$:

$$H(Z|X_q,x_O) = \underbrace{-\int_{\mathscr{X}_q}\sum_k \Pr(\omega_k|x_q,x_O)\log\Pr(\omega_k|x_q,x_O)df_{X_q|x_O}(x_q)}_{\mathbb{E}_{X_q|x_O}\left[H(Z|x_q,x_O)\right]}. \quad (7)$$

Thus, this strategy can be seen as picking the feature which is expected to maximize the "unevenness" of the class posterior probability distribution $\Pr(Z|x_O,x_q)$, i.e. so that probability mass is distributed as much as possible towards a single class. Remark that minimizing the entropy amounts to maximizing an expected Kullback-Leibler divergence between the class-conditional densities and the mixture density for X_q:

$$H(Z|X_q,X_O=x_O) = H(Z) - \underbrace{\sum_k \pi_k\int_{\mathscr{X}_q} f_k(x_q|x_O)\log\frac{f_k(x_q|x_O)}{f(x_q|x_O)}dx_q}_{\mathbb{E}_Z\left[\mathrm{D_{KL}}\left(\Pr(X_q|Z,x_O)||\Pr(X_q|x_O)\right)\right]}. \quad (8)$$

In other terms, the querying strategy can be seen as choosing the variable which maximizes the information gain induced by using the marginal distribution $\Pr(X_q|x_O)$ instead of the class-conditional distribution $\Pr(X_q|Z,x_O)$, averaged with respect to the distribution of Z.

3.3. Entropy Calculation Issues

A model of the joint distribution $\Pr(X_q,Z|x_O)$ is required for any $q\in\{1,\ldots,p\}$, in order to determine the most informative query. Such a model can be derived in a generative setting: for instance, in the Gaussian case, Property 1 makes it possible to easily update the class-conditional distributions according to the features retrieved. This is not for discriminative models (since $\Pr(Z|X)$ is directly modelled).

However, even in the former case, $H(Z|X_q,x_O)$ cannot generally be computed exactly. Note that Equation (6) can be decomposed into

$$H(Z|X_q,x_O) = \underbrace{-\sum_k \pi_k\int_{\mathscr{X}_q} f_k(x_q|x_O)\log f_k(x_q|x_O)dx_q}_{H(X_q|Z,x_O)} \underbrace{-\sum_k \pi_k\log\pi_k}_{H(Z)} + \underbrace{\int_{\mathscr{X}_q} f(x_q|x_O)\log f(x_q|x_O)dx_q}_{-H(X_q|x_O)}; \quad (9)$$

the existence of a closed form for $H(X_q|Z,x_O)$ depends on $f_k(\cdot|x_O)$; and no closed form exists for $H(X_q|x_O)$, since $f_{X_q}(\cdot|x_O)$ is a mixture of distributions.

It will always remain possible to compute an approximation to Equation (7), or to the right-hand terms in Equation (9) for which a closed form does not exist, via Monte-Carlo strategies. For instance, in the former case, we have

$$H(Z|X_q,x_O) \simeq \frac{1}{T}\sum_{t=1}^{T} H(Z|x_O,x_q^{(t)}),$$

where the T instances $x_q^{(1)},\ldots,x_q^{(T)}$ are sampled according to $f_{X_q|x_O}(\cdot) = \sum_k \pi_k f_k(\cdot|x_O)$. Note that experimentally, the estimates obtained seem more accurate when approximating Equation (7) than Equation (9).

3.4. Conditional Distribution Updating

In Algorithm 1, the last step of an iteration consists in updating the "available information" regarding the remaining missing variables, according to the value which has just been retrieved.

1. We deliberately use a lowercase letter whenever the query will be made regarding a single unknown variable.
2. We drop here the subscript referring to the iteration in which the query is made, for the sake of simplicity.

In the case where the answer provided by the oracle is a precise value x_q for the queried variable X_q, a trivial strategy would consist in simply transfering the variable just observed from the missing features M_t to the observed ones O_t, and picking the next variable to be queried regardless of the piece of information x_q just acquired. Note that this strategy, where conditional entropies are estimated separately (variable-wise), can be seen as a "naive" procedure, since it amounts to neglect the interactions between the variables in the querying strategy, and to choosing the queries based on the training data uniquely.

Indeed, the piece of uncovered information is likely to modify our knowledge of the class-conditional distributions of the remaining missing variables, especially in the case of high correlations. Therefore, we may exploit it further by updating these class-conditional distributions, i.e. by conditioning with respect to the new value x_q. As a result, the querying strategy depends on the test instance being processed. Note that this step can be difficult to proceed with, depending on the distributions considered. However, in some particular cases, we may specify a closed form for the updated class-conditional distributions. This is for instance the case with the (multivariate) Gaussian pdf, which can be updated according to a new value as described by Property 1 (see Appendix A).

3.5. Simple Example

We illustrate the strategy described above on a simple example. The (known) class-conditional distributions are Gaussian. The example makes use of two properties of Gaussian distributions, recalled in Appendix A.

Example 1 (Gaussian case) *Assume a population of instances $x_i \in \mathbb{R}^3$ distributed in $K = 2$ classes. The distribution in each class is Gaussian, with $\pi_1 = 0.5 = \pi_2$, and class-conditional expectations and covariance matrices*

$$\mu_1 = \begin{pmatrix} 1.5 \\ -0.5 \\ -0.5 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} -1.5 \\ 0.5 \\ 0.5 \end{pmatrix},$$

$$\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & 0.75 & 0.25 \\ 0.75 & 1 & 0 \\ 0.25 & 0 & 1 \end{pmatrix}.$$

The entropy is approximated as described in Section 3.3. Note that a closed form for the entropy exists in the Gaussian case (see Property 2): thus, both the first and third terms in Equation (9) may be computed exactly, leaving only the second one for approximation. However, as mentioned in Section 3.3, it turned out that the approximations were more accurate based on Equation (7).

The variable to be queried first is X_1. Intuitively, it is the most discriminative one, since it maximizes the unevenness of the posterior probability distribution (see Figure 1, where the variation of posterior probabilities along the X_1 axis is sharper). This intuition is confirmed by the expected entropies estimated here:

$$H(Z|X_1) \simeq 0.166, \quad H(Z|X_2) = H(Z|X_3) \simeq 0.581.$$

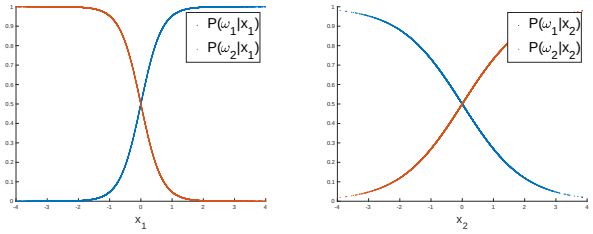

Figure 1: Posterior distributions $\Pr(Z = \omega_k | X_j = x_j)$ for $j = 1$ (left), and $j \in \{2,3\}$ (right).

Assume that the realization for X_1 is $x_1 = 0.5$. Should a decision be made, the class posterior probabilities would lead to choose class ω_1: indeed, $\Pr(\omega_1|X_1 = 0.5) \simeq 0.818$. Missing values can however be further uncovered. For this purpose, we first have to update the distributions of the missing variables. Let $X_{23|1}$ stand for the random vector obtained by conditioning $(X_2, X_3)^T$ on $(X_1 = x_1)$. Using Property 1, we have that

$$X_{23|1} \underset{\omega_k}{\sim} \mathcal{N}\left(\mu_{k,23|1}, \Sigma_{k,23|1}\right),$$

with

$$\mu_{1,23|1} = \begin{pmatrix} -1.25 \\ -0.75 \end{pmatrix}, \quad \mu_{2,23|1} = \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$

$$\Sigma_{1,23|1} = \Sigma_{2,23|1} = \begin{pmatrix} 0.4375 & -0.1875 \\ -0.1875 & 0.9375 \end{pmatrix}.$$

The marginal conditional distributions for $X_2|X_1 = x_1$ and $X_3|X_1 = x_1$, can then be deduced, and the posterior probabilities consequently updated (see Figure 2). As it turns out, variable $X_2|x_1$ is now the most discriminative one:

$$H(Z|X_2, x_1) \simeq 0.0225, \quad H(Z|X_3, x_1) \simeq 0.3637.$$

Assume that the value uncovered for X_2 is $x_2 = 2$. When it comes to decision making, the posterior probabilities obtained would now lead to choose class ω_2, with $\Pr(\omega_1|X_1 = 0.5, X_2 = 2) \simeq 0$; if variable X_3 had been uncovered instead, giving $x_3 = 0.5$ class ω_1 would have been chosen ($\Pr(\omega_1|X_1 = 0.5, X_2 = 2) \simeq 0.69$). The correct decision, based on the full feature vector, is class ω_2 ($\Pr(\omega_1|x) \simeq 0$).

4. Experiments

4.1. Synthetic Data

We first report experiments realized on synthetic data. We generated $n = 1000$ data according to a Gaussian mixture

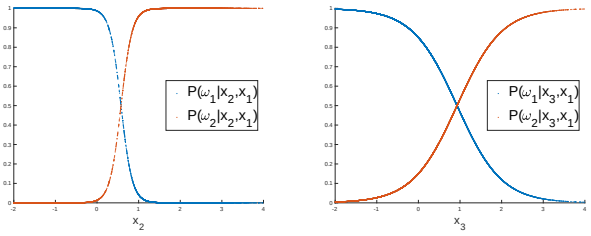

Figure 2: Posterior distributions $\Pr(Z=\omega_k|X_j=x_j,x_1)$ for $j=2$ (left) and $j=3$ (right).

with the following parameters: $\pi_1=0.6,\pi_2=0.4$, and class-conditional expectations and covariance matrices

$$\mu_1=\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix},\quad \mu_2=\begin{pmatrix}-1\\-1\\-1\\-1\\-1\end{pmatrix};$$

$$\Sigma_1=\begin{pmatrix}1 & 0.25 & 0 & 0 & 0\\ 0.25 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0.75 & 0.25\\ 0 & 0 & 0.75 & 1 & 0.5\\ 0 & 0 & 0.25 & 0.5 & 1\end{pmatrix},$$

$$\Sigma_2=\begin{pmatrix}1 & 0.25 & 0.5 & 0 & 0\\ 0.25 & 1 & 0.75 & 0 & 0\\ 0.5 & 0.75 & 1 & 0.25 & 0\\ 0 & 0 & 0.25 & 1 & 0.5\\ 0 & 0 & 0 & 0.5 & 1\end{pmatrix}.$$

We randomly split the data into a training set (60% of the data) and a test set (40%), and we trained a quadratic discriminant analysis model, i.e. we estimated a multivariate Gaussian mixture model on the labeled data. Then, four strategies were compared for processing each test instance:

1. instances were classified using all features (baseline);
2. a fixed amount of features were uncovered at random;
3. a fixed amount of features were uncovered based on the estimated entropies $H(Z|X_j)$, computed separately for all natural features X_j ($j=1,\ldots,p$), without updating the distributions;
4. a fixed amount of features were uncovered via an iterative procedure where the class-conditional distributions are updated sequentially, according to the uncovered values.

We let the amount n_f of uncovered features vary: we set $n_f=1,\ldots,p-1$. We generated 10^4 samples in order to estimate the conditional entropy, by approximating Equation (7). We repeated the procedure described above $T=25$ times, computing each time the error rate.

Figure 3 displays the error rates thus obtained, along with 95% confidence intervals error rates. Obviously, when $n_f=1$, both strategies 3 and 4 perform similarly (slight differences being sometimes observed, due to the approximation of the conditional entropy). Overall, the sequential strategy performs significantly better than random selection, and better than the basic entropy procedure (the difference with this latter being significant for $n_f=2$ and $n_f=3$. For all three methods, the accuracy gets closer to that of the baseline as n_f increases (and is obviously identical when $n_f=5$), due to the total amount of uncovered information growing with n_f.

Here, we also provide information with respect to the queries made (based on the data generated during one repetition of the procedure). Using the naive entropy approach, we observed that $q_1=3$ — note that actually, all variables are equally informative, but approximations in the entropy calculation lead to choose this variable. We estimated the frequencies of each sequence of queries computed with the sequential strategy, differentiating according to the actual class of the instances. The three most frequent sequences are displayed in Table 1 (class 1) and 2 (class 2).

The following remarks can be made. First of all, the two classes have a frequent query in common — we may imagine that it corresponds to instances near the classification boundary. Second, the remaining queries differ, for each class, by the two last variables queried for: they can thus be deemed as typical from the class. This insight is confirmed by the fact that the two most frequent queries for instances in class 1 *were never computed* for instances in class 2 (the two most frequent queries for class 2 having however been computed, for 10 and 8 instances respectively).

This confirms that according to the data, there may exist "typical sequences" for instances from a particular class. This opens the way to strategies where elements of sequences can be suggested, e.g. according to the similarity of the test instance being evaluated to instances with known optimal sequences, in order to alleviate the computational cost of the sequential strategy presented above.

Table 1: Top three most frequent sequences, class ω_1

Sequence	nb.	frequency
(3, 1, 2, 4, 5)	33	14.04%
(3, 1, 2, 5, 4)	29	12.34%
(3, 4, 2, 5, 1)	25	10.64%

4.2. Real Data

We present here results obtained on four real datasets of the UCI Machine Learning repository [8]. For each dataset, we used the procedure described in Section 4.1. The data were

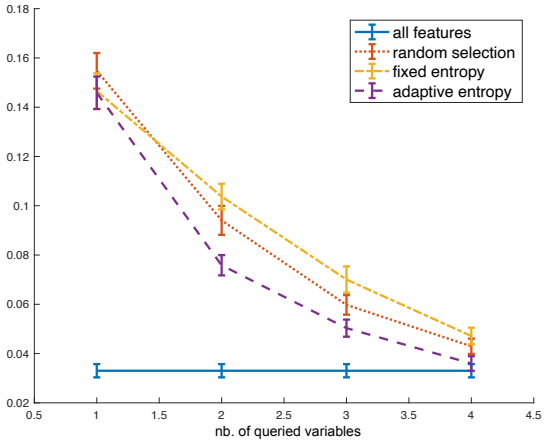

Figure 3: Error curves as a function of the number of uncovered variables, synthetic (Gaussian) data.

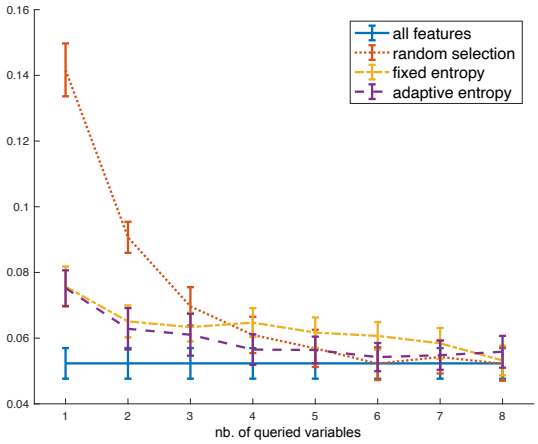

Figure 4: Error curves, breast cancer.

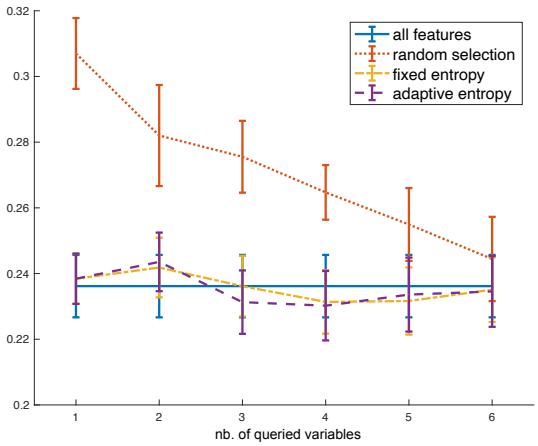

Figure 5: Error curves, pima data.

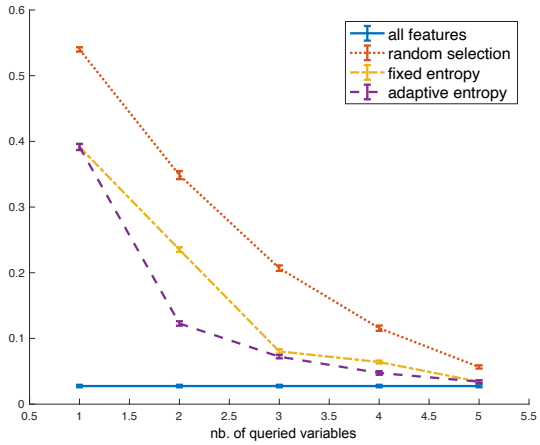

Figure 6: Error curves, optdigits data.

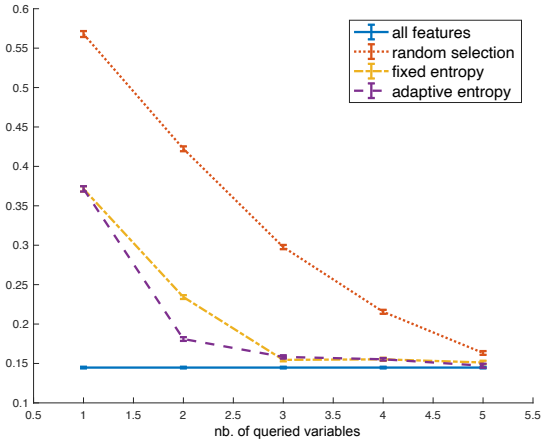

Figure 7: Error curves, satimage data.

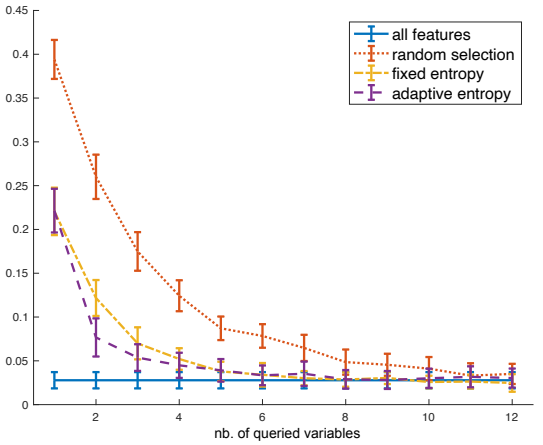

Figure 8: Error curves, wine data.

Table 2: Top three most frequent sequences, class ω_2

Sequence	nb.	frequency
(3, 2, 4, 5, 1)	46	28.40%
(3, 2, 4, 1, 5)	26	16.05%
(3, 4, 2, 5, 1)	22	13.58%

centered and scaled, and in order to avoid numerical issues when estimating the covariance matrices, we performed a PCA in order to keep $p = 6$ descriptive variables for the optdigits and satimage data. In the case of the optdigits data, we focused on five classes (corresponding to the digits '0', '1', '3', '7', '8') in order to keep the computational cost reasonable (remember for each test instance, we compute the average error rate for each possible amount of retrieved features, which is quadratic in the number of features).

Figures 4 to 7 display the classification error as a function of the number of uncovered variables. These curves confirm the insight given by the experiments on the synthetic data, in that selecting the variables according to their expected "discriminative power" seems a good strategy, especially if the number of uncovered variables is low. It also mitigates the enthusiasm for the sequential procedure. Indeed, this procedure does not perform systematically better than the naive entropy approach; when it does, the difference is rarely significant (it is actually the case for the optdigits, satimage and wine datasets, with $n_f = 2$).

We may provide an explanation to this fact. Our strategy for updating the class-conditional distributions and for estimating the conditional entropy relies here heavily on the assumption that the data are Gaussian. Should this distributional assumption be erroneous, the updated distributions might further differ from the actual distributions of the missing variables; besides, the estimated entropy values might be improper. This would therefore lead to making queries that are actually not optimal with respect to the actual distribution of the data. Should this latter insight be confirmed, using robust strategies based on sets of distributions might be an appropriate solution.

5. Imprecise Answers

5.1. Problem

The strategy presented above assumes answers to be precise, in the form of a value x_q. This assumption is reasonable in some cases, such as for instance in the context of a medical examination as mentioned in the introduction. In other settings, however, the answer may be a piece of imprecise or uncertain knowledge with respect to the queried variable.

We will discuss here the case where the expert provides imprecise but certain answers, in the form of intervals $R_q \subset \mathscr{X}_q$ in which the value of the queried variable lies (or is considered to). This kind of information actually corresponds to epistemic uncertainty [9], due to the expert being unable to answer with precision to the query. This is sometimes due to the elicitation process: for instance, the expert may be asked whether a given variable is greater or smaller than a given threshold.

5.2. Incorporating Imprecise Answers

An imprecise answer must be taken into account at two levels: computing the next best query, and making a decision.

Naive entropy strategy The naive strategy can still be used (since the distributions used to choose the queries are not affected by the answers). Should a decision be made, the variable can be marginalized out according to this truncated distribution — which amounts to replace the class-conditional densities in Equation (5) by their integral over R_q, that is by the class-conditional probabilities $\Pr(R_q|\omega_k, x_{O_{t-1}})$.

In such a case of ill-known feature values $x_q \in R_q$, the criteria used in the precise case can be replaced by a cautious counterpart. The imprecise answer defines a credal set for the posterior probabilities: for any class $\omega_k \in \Omega$,

$$\mathscr{P}_k = \left\{\Pr(\omega_k|x_q, x_O) \text{ such that } x_q \in R_q\right\}.$$

Once this credal set is specified, further queries can be made by computing pessimistic entropy values (in a cautious and therefore robust perspective), typically using a maximum of entropy criterion [1, 2]. Alternatively, a decision can be made, by using cautious strategies [17] such as interval dominance.

Sequential strategy In this case, the distribution of an imprecisely observed variable can be updated by truncating the density. The variable can be queried for again, in the hope for a more precise answer allowing to further decrease the expected entropy for the posterior probability distribution over the classes.

A critical consequence of the realization x_q being partially identified is that updating accordingly the class-conditional distributions of the missing variables becomes difficult. Indeed, in each class ω_k, we only know that the conditional distribution of the missing variables $X_M|R_q, x_O$ lies in the set of conditional pdfs

$$\mathscr{F}_{X_M|\omega_k,R_q x_O} = \left\{f_{X_M|Z=\omega_k,x_q,x_O} \text{ such that } x_q \in R_q\right\}.$$

Note that using Bayes rule yields, for any variable $X_j \in X_M$,

$$f_{X_j|\omega_k,x_O,R_q}(x_j) = \frac{\Pr(R_q|\omega_k, x_O, x_j) f_{X_j|\omega_k,x_O}(x_j)}{\Pr(R_q|\omega_k, x_O)}. \quad (10)$$

Although the denominator and the right-hand term in the numerator can be easily computed, $\Pr(R_q|\omega_k, x_O, x_j)$ depends on x_j. Therefore, characterizing this distribution, or

sampling from it in order to compute the conditional entropy, is a difficult problem, even in the simple case of Gaussian class-conditional distributions. Note that recent work regarding the propagation of uncertainty using copula dependence [18] may provide a starting point to address this issue in the general case.

6. Conclusion and Perspectives

Summary In this article, we addressed the case of supervised classification, where test instances to be classified are not observed, or only partially, and where an oracle can be interrogated with respect to the missing values in order to increase the amount of information upon which the decision will be made. In order to make these choices, we propose to use a conditional entropy criterion, which indicates the variable which is expected to determine the class variable the most. We propose two strategies for making successive choices this purpose. A naive approach consists in computing the conditional entropies separately, i.e. based on the training data uniquely; a sequential approach consists in updating the class-conditional distributions of the missing variables according to the pieces of information uncovered in the process, thus depending on the test instances.

Both strategies are evaluated on several datasets, under the assumption that the class-conditional distributions are multivariate Gaussians (which corresponds to the well-known quadratic discriminant analysis classifier). They are compared to classical QDA where all features are used for making decisions, and a strategy where features are uncovered at random for each test instance. The results obtained show the interest of choosing variables based on their extent to influence the class information. Although the sequential procedure seems to be able to provide better choices than the naive one, it may also be highly sensitive to the distributional assumptions being (reasonably) satisfied.

We then briefly discuss the issue of imprecise answers to the queries made. Then, the oracle provides sets of possible values for the queried variable, rather than a single (precise and certain) value. Although the naive approach can still be used in this case, using both the precise-probabilistic approach presented or a robust variant, the sequential approach becomes much more difficult to implement, due to the class-conditional distributions being ill-specified.

Future work The future directions of this preliminary work are many, and point towards concepts and tools developed within the imprecise-probabilistic framework. The first direction would be to further investigate the case of imprecise answers, for instance by deriving sets of posterior probabilities over the classes induced by the intervals provided, and consequently using a robust strategy to choose between the queries, for instance by using extensions of the entropy [1, 2], and for making decisions [17, 7]. We may also study the case of imprecise and uncertain answers; in this case, the same issue of specifying the class-conditional distributions (and updating them) will be key.

The results obtained on real data also ask the question of robustness with respect to the distributional assumption. In this case, sets of class-conditional distributions may also be used, this time to account for the fact that the actual distributions might differ from the estimated ones. This is likely to be critical, especially in the sequential approach where the sensitivity to the distributional assumption being violated can be expected to increase, possibly dramatically, with the successive conditionings being made.

Another important issue to be addressed is that of choosing the amount of features to be uncovered. In the experiments realized, we let the amount of uncovered features vary up to the maximum possible, therefore showing that a few carefully chosen instances make it possible to attain a good classification accuracy. It might be interesting to be able to quantify the (expected) amount of remaining (i.e., missing) information. However, although the total conditional entropy can be decomposed using a chain rule, this decomposition does not hold when the class-conditional distributions are altered by conditioning with respect to the successive uncovered values. This issue of choosing the number of queries may also be further developed with queries having a cost (possibly depending on the feature, such as in the medical examination example), which calls for being able to evaluate the cost-benefit ratio of a query.

Several other directions seem to be of interest. We mentioned in the experiments that some sequences of queries seem to be frequently observed. We may investigate how such typical sequences could be inferred, for instance using a separate (validation) set of instances. We may also consider other criteria for choosing the queries — this direction being strongly related to that of robustness.

Appendix A. Properties of Gaussian Random Vectors

Property 1 (Conditioning in Gaussian vectors) *Let $X \sim \mathcal{N}(\mu, \Sigma)$ be a Gaussian random vector with expectation μ and covariance matrix Σ. Assume that X can be partitioned into two subvectors X_A and X_B, where A and B indicate the indices of the corresponding variables:*

$$X = \begin{pmatrix} X_A \\ X_B \end{pmatrix}, \quad \mu = \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix}.$$

Then, the distribution of X_A conditional on $X_B = x_B$ is multivariate Gaussian:

$$X_A | X_B = x_B \sim \mathcal{N}(\mu_{A|B}, \Sigma_{A|B}), \tag{11}$$

with

$$\mu_{A|B} = \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B), \quad \Sigma_{A|B} = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}.$$

Property 2 (Entropy of a Gaussian random vector) *Let $X \sim \mathcal{N}(\mu, \Sigma)$ be a Gaussian random in $\mathcal{X} = \mathbb{R}^p$ vector with expectation μ and covariance matrix Σ. Its differential entropy can be computed as follows:*

$$H(X) = \frac{p}{2}\log(2\pi) + \frac{p}{2} + \frac{1}{2}\log(\det\Sigma).$$

References

[1] J. Abellán and S. Moral. Maximum of entropy for credal sets. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 11(5):587–598, 2005.

[2] J. Abellán and S. Moral. Upper entropy of credal sets. applications to credal classification. *International Journal of Approximate Reasoning*, 39(2-3):235–255, 2005.

[3] Alessandro Antonucci, Giorgio Corani, and Sandra Gabaglio. Active learning by the naive credal classifier. In *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models*, pages 3–10, 2012.

[4] Nawal Benabbou and Patrice Perny. Adaptive elicitation of preferences under uncertainty in sequential decision making problems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4566–4572, 2017.

[5] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985.

[6] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15:201–221, 1994.

[7] Thierry Denœux. Decision-making with belief functions: a review. *International Journal of Approximate Reasoning*, 109:87–110, 2019.

[8] D. Dua and C. Graff. UCI Machine Learning repository, 2019. URL http://archive.ics.uci.edu/ml.

[9] S. Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering and System Safety*, 54(2-3):217–223, 1996.

[10] Pallika Kanani and Prem Melville. Prediction-time active feature-value acquisition for customer targeting. In *Advances in Neural Information Processing Systems*, 2008.

[11] Edwin Lughofer. Hybrid active learning for reducing the annotation effort of operators in classification systems. *Pattern Recognition*, 45(2):884–896, 2012.

[12] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[13] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In *NIPS'01: Proceedings of the 14th International Conference on Neural Information Processing Systems*, pages 841–848, 2001.

[14] Christian P. Robert. *The Bayesian choice : from decision-theoretic foundations to computational implementation*. Springer, 2007.

[15] Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8: 1625–1657, 2007.

[16] Maytal Saar-Tsechansky, Prem Melville, and Foster Provost. Active feature-value acquisition. *Management Science*, 55(4):664–684, 2009.

[17] M.C.M. Troffaes. Decision making under uncertainty using imprecise probabilities. *Int. J. of Approximate Reasoning*, 45:17–29, 2007.

[18] Jiaxin Zhang and Michael Shields. On the quantification and efficient propagation of imprecise probabilities with copula dependence. *International Journal of Approximate Reasoning*, 122:24–46, 2020.