A New Evidential K-Nearest Neighbor Rule based on Contextual Discounting with Partially Supervised learning

Thierry Denoeux, Orakanya Kanjanatarakul, Songsak Sriboonchitta

A New Evidential K-Nearest Neighbor Rule based on Contextual Discounting with Partially Supervised learning

Thierry Denœux[a,b,*], Orakanya Kanjanatarakul[c], Songsak Sriboonchitta[d]

[a] *Université de Technologie de Compiègne, CNRS, UMR 7253 Heudiasyc, Compiègne, France*
[b] *Shanghai University, UTSEUS, Shanghai, China*
[c] *Faculty of Management Sciences, Chiang Mai Rajabhat University, Thailand*
[d] *Faculty of Economics, Chiang Mai University, Thailand*

Abstract

The evidential K nearest neighbor classifier is based on discounting evidence from learning instances in a neighborhood of the pattern to be classified. To adapt the method to partially supervised data, we propose to replace the classical discounting operation by contextual discounting, a more complex operation based on as many discount rates as classes. The parameters of the method are tuned by maximizing the evidential likelihood, an extension of the likelihood function based on uncertain data. The resulting classifier is shown to outperform alternative methods in partially supervised learning tasks.

Keywords: Belief functions, Dempster-Shafer theory, classification, machine learning, soft labels, uncertain data.

1. Introduction

The evidential K-nearest neighbor (EKNN) classifier [6] is a distance-based classification algorithm based on the Dempster-Shafer (DS) theory of evidence [5, 30, 10]. Since its introduction in 1995, it has been used extensively (see, e.g., [3], [15], [31], [37]) and several variants have been developed [1], [18], [17], [20], [21], [22], [26], [36], [38]. The EKNN classifier is based on the following simple ideas: (1) each neighbor of the pattern x to be classified is considered as a piece of evidence about the class of x, represented by a DS mass function; (2) each mass function is *discounted* (weakened) based on its distance to x; and (3) the discounted mass functions induced by the K nearest neighbors of x are combined by Dempster's rule, the fundamental mechanism for pooling evidence in DS theory.

In [6], the parameters used to define the discount rate as a function of distance were fixed heuristically, and the method was shown to outperform other K-nearest neighbor rules. In [39], the authors showed that the performances of the method could be further improved by learning the parameters through minimizing the mean squared error (MSE) between

*Corresponding author.
Email address: thierry.denoeux@utc.fr (Thierry Denœux)

pignistic probabilities and class indicator variables. In [11], the EKNN rule was extended to the case where the class label of training patterns is only *partially known*, and described by a possibility distribution. However, the learning procedure defined in [39] cannot be straightforwardly extended to the partially labeled setting because (1) the discount rate defined in the procedure depends on the class of the neighboring pattern that is assumed to be known, and (2) combining arbitrary mass functions and computing pignistic probabilities has exponential complexity in the worst case.

In this paper[1], we revisit the EKNN classifier by exploiting some recent developments in the theory of belief functions: (1) The discounting operation is replaced by *contextual discounting* [25], allowing us to define one discount rate parameter per class even in the partially labeled case; and (2) instead of the MSE and pignistic probabilities, we propose to use the *conditional evidential likelihood* criterion [8, 29], which allows us to account for partial class labels in a natural way, and can be computed in linear time as a function of the number of classes.

The rest of this paper is organized as follows. Background definitions and results are first recalled in Section 2. The Contextual-Discounting Evidential K-NN (CD-EKNN) classifier is then introduced in Section 3, and experimental results are reported in Section 4. Section 5 concludes the paper.

2. Background

In this section, we provide a reminder of the main notions needed in the rest of the paper. Basic concepts of DS theory are first recalled in Section 2.1. The classical and contextual discounting operations are then reviewed in Section 2.2, and the notion of evidential likelihood criterion is briefly introduced in Section 2.3.

2.1. Basic concepts

Let Ω be a finite set. A *mass function* [30] is a mapping m from the power set of Ω, denoted as 2^Ω, to the interval $[0, 1]$, such that

$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{1}$$

and $m(\emptyset) = 0$. The subsets A of Ω such that $m(A) > 0$ are called the *focal sets* of m. Typically, Ω is a set of possible answers to some question, and $m(A)$ is interpreted as a share of a unit mass of belief allocated to the hypothesis that the truth is in A, and which cannot be allocated to any strict subset of A based on the available evidence. A mass function with only one focal set is said to be *logical*.

[1]This paper is a revised and extended version of a short paper presented at the BELIEF 2018 conference [19].

Given a mass function m, *belief* and *plausibility* functions are defined as follows:

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{2a}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \tag{2b}$$

for all $A \subseteq \Omega$. The quantity $Bel(A)$ can be interpreted as a degree of support in A, while $Pl(A)$ can be seen as a degree to which hypothesis A is consistent with the evidence [30]. The *contour function* $pl : \Omega \to [0, 1]$ is the restriction of the plausibility function Pl to singletons, i.e., $pl(\omega) = Pl(\{\omega\})$, for all $\omega \in \Omega$.

Two mass functions m_1 and m_2 representing independent items of evidence can be combined using *Dempster's rule* [30] as follows,

$$(m_1 \oplus m_2)(A) = \frac{1}{1 - \kappa} \sum_{B \cap C = A} m_1(B) m_2(C), \tag{3}$$

for all $A \subseteq \Omega$ such that $A \neq \emptyset$, where the quantity

$$\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \tag{4}$$

is called the *degree of conflict* between m_1 and m_2. The combined mass function $m_1 \oplus m_2$ is called the *orthogonal sum* of m_1 and m_2. Dempster's rule is commutative and associative. The contour function pl of $m = m_1 \oplus m_2$ is given by

$$pl(\omega) = \frac{pl_1(\omega) pl_2(\omega)}{1 - \kappa}, \tag{5}$$

for all $\omega \in \Omega$, where pl_1 and pl_2 are, respectively, the contour functions of m_1 and m_2.

Given a mass function m and a nonempty subset A of Ω such that $Pl(A) > 0$, the conditional mass function $m(\cdot|A)$ is defined as the orthogonal sum of m and the logical mass function with focal set A. Conversely, given a conditional mass function m_0 given A, its *conditional embedding* [32] is the (unconditional) mass function on Ω obtained by transferring each mass $m_0(C)$ to $C \cup \overline{A}$, for all $C \subseteq A$. Conditional embedding is a form of "deconditioning", i.e., it performs the inverse of conditioning.

Dempster's rule is essentially a conjunctive operation (it boils down to set intersection when combining two logical mass functions m_A and m_B with overlapping focal sets). A disjunctive counterpart of Dempster's rule is obtained by replacing intersection by union in the right-hand side of (3). The resulting operation, called the *disjunctive rule of combination* [12], is defined as

$$(m_1 \cup m_2)(A) = \sum_{B \cup C = A} m_1(B) m_2(C), \tag{6}$$

for all $A \subseteq \Omega$. The disjunctive rule is relevant for combining pieces of evidence, when we only know that at least one piece of evidence is reliable [33].

Given a mass function m, the associated *pignistic* probability distribution [35] is defined as

$$BetP(\omega) = \sum_{\{A \subseteq \Omega | \omega \in A\}} \frac{m(A)}{|A|}, \tag{7}$$

for all $\omega \in \Omega$. The pignistic mass-probability transformation (7) was advocated by Smets for decision-making [34],[35]. A recent review of methods for decision-making in the belief function framework can be found in [9].

2.2. Discounting

Let m be a mass function on $\Omega = \{\omega_1, \ldots, \omega_c\}$ and β a coefficient in $[0, 1]$. The *discounting* operation [30] with discount rate $1 - \beta$ transforms m into the following mass function:

$$^{\beta}m = \beta\, m + (1 - \beta)\, m_?, \tag{8}$$

where $m_?$ is the *vacuous* mass function defined by $m_?(\Omega) = 1$. Mass function $^{\beta}m$ is, thus, a mixture of m and $m_?$, and the discount rate is the weight of $m_?$ in the mixture. The contour function of $^{\beta}m$ is

$$^{\beta}pl(\omega_k) = 1 - \beta + \beta pl(\omega_k), \quad k = 1, \ldots, c, \tag{9}$$

where pl is the contour function of m.

The discounting operation can be justified as follows [35]. Assume that m is provided by a source that may be reliable (R) or not ($\neg R$). If the source is reliable, we adopt its opinion as ours, i.e., we set $m(\cdot|R) = m$. If it is not reliable, then it leaves us in a state of total ignorance, i.e., $m(\cdot|\neg R) = m_?$. Furthermore, assume that we have the following mass function on $\mathcal{R} = \{R, \neg R\}$: $m_{\mathcal{R}}(\{R\}) = \beta$ and $m_{\mathcal{R}}(\mathcal{R}) = 1 - \beta$, i.e., our degree of belief that the source is reliable is equal to β. Then, combining the conditional embedding of $m(\cdot|R)$ with $m_{\mathcal{R}}$ yields precisely $^{\beta}m$ in (8), after marginalizing on Ω.

In [25], the authors generalized the discounting operation using the notion of *contextual discounting*. In the corresponding refined model, $m(\cdot|R)$ and $m(\cdot|\neg R)$ are defined as before, but our beliefs about the reliability of the source are now defined given each state[2] in Ω, i.e., we have c conditional mass functions defined by $m_{\mathcal{R}}(\{R\}|\omega_k) = \beta_k$ and $m_{\mathcal{R}}(\mathcal{R}|\omega_k) = 1 - \beta_k$, for $k = 1, \ldots, c$. In this model, β_k is the degree of belief that the source of information is reliable, given that the true state is ω_k. Combining the conditional embeddings of $m(\cdot|R)$ and $m_{\mathcal{R}}(\cdot|\omega_k)$ for $k = 1, \ldots, c$ yields the following discounted mass function,

$$^{\boldsymbol{\beta}}m(A) = \sum_{B \subseteq A} m(B) \left(\prod_{\omega_k \in A \setminus B} (1 - \beta_k) \prod_{\omega_l \in \overline{A}} \beta_l \right) \tag{10}$$

[2]Actually, the contextual discounting operation can be defined in a more general setting, where the reliability of the source is defined given each element of a partition of Ω. Only the simplest case is considered here.

for all $A \subseteq \Omega$, where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_c)$, and a product of terms is equal to 1 if the index set is empty. It can also be shown [25] that ${}^{\boldsymbol{\beta}}m(A)$ is the result of the disjunctive combination of m with an unnormalized mass function m_0 defined as

$$m_0(C) = \prod_{\omega_k \in C} (1 - \beta_k) \prod_{\omega_\ell \in \overline{C}} \beta_\ell, \quad \forall C \subseteq \Omega \ .$$

The pignistic probability distribution associated to ${}^{\boldsymbol{\beta}}m$ does not have a simple expression but, as shown in [25], its contour function is

$${}^{\boldsymbol{\beta}}pl(\omega_k) = 1 - \beta_k + \beta_k pl(\omega_k), \quad k = 1, \ldots, c, \tag{11}$$

which has the same form as (9). Also, comparing Eqs (9) and (11), we can see that contextual discounting yields the same contour function as classical discounting when the coefficients β_k are equal, although the discounted mass functions are different.

Example 1. *As in [25], let us consider a simplified aerial target recognition problem, in which we have three classes: airplane ($\omega_1 \equiv a$), helicopter ($\omega_2 \equiv h$) and rocket ($\omega_3 \equiv r$). Let $\Omega = \{a, h, r\}$. Assume that a sensor has provided the following mass function for a given target: $m(\{a\}) = 0.5$, $m(\{r\}) = 0.5$, meaning that the sensor hesitates between classifying the target as an airplane or a rocket. The degree of belief β_1 that the sensor is reliable when the source is an airplane is equal to 0.6, whereas the sensor is known to be fully reliable ($\beta_2 = \beta_3 = 1$) when the target is a helicopter or a rocket. Mass function m_0 is then*

$$m_0(\emptyset) = 0.6, \quad m_0(\{a\}) = 0.4,$$

and the discounted mass function ${}^{\boldsymbol{\beta}}m = m \cup m_0$ is

$${}^{\boldsymbol{\beta}}m(\{a\}) = 0.5, \quad {}^{\boldsymbol{\beta}}m(\{r\}) = 0.3, \quad {}^{\boldsymbol{\beta}}m(\{a, r\}) = 0.2.$$

We can see that 40% of the mass initially assigned to $\{r\}$ has been transferred to $\{a, r\}$, which can be interpreted as follows [25]: "if the target is an airplane, then the source is not reliable, and we may erroneously declare it as a rocket; consequently, when the source reports a rocket, it may actually be a rocket or an airplane". □

2.3. Evidential likelihood

The notion of *evidential likelihood*, introduced in [8], extends the classical notion of likelihood to the case where statistical data are only partially observed and described by a belief function in sample space.

Let Y be a discrete random vector with finite sample space $\mathcal{Y}$ and probability mass function $p_Y(y; \theta)$ assumed to be known up to a parameter $\theta \in \Theta$. After a realization y of Y has been observed, the *likelihood function* is the mapping from Θ to $[0, 1]$ defined by

$$L(\theta) = p_Y(y; \theta), \quad \forall \theta \in \Theta. \tag{12}$$

Let us now assume that y is not observed precisely, but we collect some evidence about y that we represent by a mass function m on $\mathcal{Y}$. The likelihood function (12) can then be generalized [8] to

$$L_e(\theta) = \sum_{A \subseteq \mathcal{Y}} m(A) \sum_{y \in A} p_Y(y; \theta), \quad \forall \theta \in \Theta, \tag{13}$$

Function $L_e(\theta)$ defined by (13) is called the *evidential likelihood function* induced by the uncertain data m. Whenever mass function m in (13) is certain, i.e., when $m(\{y\}) = 1$, the evidential likelihood (13) coincides with the classical likelihood (12), which only depends on the pdf p_Y modeling the random data generating process.

By permuting the two summations in (13), we get another expression for $L_e(\theta)$ as

$$L_e(\theta) = \sum_{y \in \mathcal{Y}} p_Y(y; \theta) \sum_{A \ni y} m(A) = \sum_{y \in \mathcal{Y}} p_Y(y; \theta) pl(y), \tag{14}$$

where pl is the contour function associated to m. From the right-hand side of (14), we can see that $1 - L_e(\theta)$ equals the degree of conflict between the uncertain data m and the probability mass function $p(y; \theta)$. Maximizing $L_e(\theta)$ thus amounts to minimizing the conflict between the data and the model.

Equation (14) also reveals that $L_e(\theta)$ can, alternatively, be viewed as the expectation of $pl(Y)$ with respect to $p_Y(\cdot; \theta)$:

$$L_e(\theta) = \mathbb{E}_\theta[pl(Y)]. \tag{15}$$

In the special case where $Y = (Y_1, \ldots, Y_n)$ is an independent sample, and assuming that the contour function pl can be decomposed as

$$pl(y) = pl_1(y_1) \ldots pl_n(y_n), \tag{16}$$

a property called *cognitive independence* by Shafer [30], (15) simplifies to

$$L_e(\theta) = \prod_{i=1}^{n} \mathbb{E}_\theta[pl_i(Y_i)]. \tag{17}$$

3. Contextual-discounting Evidential K-NN classifier

In this section, we start by recalling the EKNN classifier in Section 3.1. The new variant based on contextual discounting is then introduced in Section 3.2, and partially supervised parameter optimization in this model is addressed in Section 3.3.

3.1. Evidential K-NN classifier

Consider a classification problem with c classes in $\Omega = \{\omega_1, \ldots, \omega_c\}$, and a learning set $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^n$ of n examples (x_i, y_i), where x_i is a p-dimensional feature vector describing example i, and $y_i \in \Omega$ is the class of that example. Let x be a new pattern to be classified, and $\mathcal{N}_K(x)$ the set of its K nearest neighbors in $\mathcal{L}$, according to some distance d (usually,

the Euclidean distance when the p features are numerical). In [6] and [39], it was assumed that each neighbor $x_j \in \mathcal{N}_K(x)$ induces a mass function $\widehat{m}_j$ defined as

$$\widehat{m}_j(\{\omega_k\}) = \beta_k(d_j) y_{jk}, \quad k = 1, \ldots, c \tag{18a}$$

$$\widehat{m}_j(\Omega) = 1 - \sum_{k=1}^{c} \beta_k(d_j) y_{jk}, \tag{18b}$$

where $y_{jk} = 1$ if $y_j = \omega_k$ and $y_{jk} = 0$ otherwise, $d_j = d(x, x_j)$ is the distance between x and x_j, and β_k is a decreasing function, usually taken as $\beta_k(d_j) = \alpha \exp(-\gamma_k d_j^2)$, where α is a coefficient in $[0, 1]$ and the γ_k's are strictly positive scale parameters. Mass function $\widehat{m}_j$ defined by (18) can be seen as a *discounted* version (using the classical discounting operation recalled in Section 2.2) of the certain mass function m_j such that $m_j(\{\omega_k\}) = y_{jk}$, $k = 1, \ldots, c$, with discount rate $1 - \beta_k(d_j)$. Its contour function is

$$\widehat{pl}_j(\omega_k) = 1 - \sum_{l=1}^{c} \beta_l(d_j) y_{jl} + \beta_k(d_j) y_{jk} = 1 - \sum_{l \neq k} \beta_l(d_j) y_{jl}, \tag{19}$$

for $k = 1, \ldots, c$.

By pooling mass functions $\widehat{m}_j$ induced by the K nearest neighbors of x using Dempster's rule (3), we get the combined mass function

$$\widehat{m} = \bigoplus_{x_j \in \mathcal{N}_K(x)} \widehat{m}_j, \tag{20}$$

which summarizes the evidence about the class of x based on its K nearest neighbors. The focal sets of $\widehat{m}$ are the singletons $\{\omega_k\}$, $k = 1, \ldots, c$, and Ω. The class with maximum pignistic probability or, equivalently, with maximum plausibility can then be selected [7].

In [39], it was proposed to leave parameter α fixed and to learn parameter vector $\gamma = (\gamma_1, \ldots, \gamma_c)$ by minimizing the following error function,

$$C(\gamma) = \sum_{i=1}^{n} \sum_{k=1}^{c} (\widehat{Betp}_i(\omega_k) - y_{ik})^2, \tag{21}$$

where $\widehat{Betp}_i$ is the pignistic probability distribution computed from mass function $\widehat{m}_i$ obtained from the K nearest neighbors of x_i:

$$\widehat{Betp}_i(\omega_k) = \widehat{m}_i(\{\omega_k\}) + \frac{\widehat{m}_i(\Omega)}{c}. \tag{22}$$

Because this classifier is based on c learnable parameters γ_k, $k = 1, \ldots, c$, it will be later referred to as the *γ_k-EKNN classifier*.

Extension to partially supervised data

In [6] and [11], it was proposed to apply the EKNN procedure to *partially labeled* data $\mathcal{L} = \{(x_i, m_i)\}_{i=1}^n$, where m_i is an arbitrary mass function (called a *soft label* [4, 29]) that represents partial knowledge about the class of example x_i. Such knowledge may be provided by an expert (for instance, in diagnosis problems), or by any other source of information about the class membership of training instances. The fully supervised case is recovered when all mass functions m_i are certain, while semi-supervised learning [2] is recovered when some m_i's are vacuous, and some are certain.

As noted in [6], since mass function $\widehat{m}_j$ defined by (18) is the discounted version of the certain mass function $m_j(\{y_j\}) = 1$, the same discounting operation can be applied whatever the form of m_j. More precisely, let x be a pattern to be classified, and x_j one of its K nearest neighbors. In [11], it was proposed to generalize (18) by discounting each neighbor mass function m_j with discount rate $1 - \beta(d_j) = 1 - \alpha \exp(-\gamma d_j^2)$ depending only on the distance d_j between x and x_j. The evidence from the j-th nearest neighbor is then represented by the following mass function:

$$\widehat{m}_j(A) = \beta(d_j) m_j(A), \quad \text{for all } A \subset \Omega \tag{23a}$$

$$\widehat{m}_j(\Omega) = 1 - \beta(d_j) + \beta(d_j) m_j(\Omega), \tag{23b}$$

and the contour function corresponding to $\widehat{m}_j$ in (23) is

$$\widehat{pl}_j(\omega_k) = 1 - \beta(d_j) + \beta(d_j) pl_{jk}, \tag{24}$$

where pl_{jk} is the plausibility that instance j belongs to class ω_k:

$$pl_{jk} = \sum_{\{A \subseteq \Omega | \omega_k \in A\}} m_j(A).$$

It can be checked that (24) becomes identical to (19) when $\beta_k(d_j) = \beta(d_j)$ and $pl_{jk} = y_{jk}$ for all k. The combined mass function $\widehat{m}$ is defined by (20) as in the fully supervised case. The rule defined by (23) has two parameters: α and γ. In [11], it was proposed to optimize these two parameters by minimizing an error function depending on the pignistic probability induced by $\widehat{m}$. This classifier will hereafter be referred to as the *(α, γ)-EKNN classifier*.

The (α, γ)-EKNN classifier yields good results in some cases, but it depends on only two parameters, against $c + 1$ for the γ_k-EKNN classifier. This lack of flexibility hampers the performance of the method for some datasets, as we will see in Section 4. Another issue with this extension of the EKNN procedure to partially labeled data is computational complexity. To compute the MSE criterion (21) or any error function depending on pignistic probabilities (as proposed in [11]), we need to compute the whole combined mass functions $\widehat{m}_i$, which, in the worst case of arbitrary mass functions m_i, has exponential complexity and may be problematic for classification tasks with a large number of classes. In contrast, the evidential likelihood recalled in Section 2.3 requires to compute only the contour function of the combined mass functions $\widehat{m}_i$, which can be done in time proportional to the number of classes.

In Section 3.2 below, we will solve these two issues by introducing a new version of the EKNN classifier suitable for partially supervised data, based on the combination of two ideas: using contextual discounting instead of classical discounting, and using the evidential likelihood instead of the MSE criterion.

3.2. Contextual discounting EKNN classifier

As the EKNN classifier recalled in the previous section is based on classical discounting, it can be readily generalized using the contextual discounting operation recalled in Section 2.2. In the general partially supervised case, the discounting operation (23) can be replaced by contextual discounting. Using (10), the evidence from the j-th neighbor with label m_j and situated at distance d_j then becomes

$$\widehat{m}_j(A) = \sum_{B \subseteq A} m_j(B) \left(\prod_{\omega_k \in A \setminus B} [1 - \beta_k(d_j)] \prod_{\omega_l \in \overline{A}} \beta_l(d_j) \right), \tag{25}$$

where the coefficients $\beta_k(d_j)$ can be defined as

$$\beta_k(d_j) = \alpha \exp(-\gamma_k d_j^2), \quad k = 1, \ldots, c. \tag{26}$$

This new rule, called *Contextual Discounting Evidential K-nearest neighbor* (CD-EKNN), has $c + 1$ learnable parameters: $\alpha \in [0, 1]$ and $\gamma_k \geq 0$, $k = 1, \ldots, c$.

Whereas the discounted mass function $\widehat{m}_j$ given by (25) has a complicated expression in general, its contour function can be obtained from (11) as

$$\widehat{pl}_j(\omega_k) = 1 - \beta_k(d_j) + \beta_k(d_j) pl_{jk}, \quad k = 1, \ldots, c, \tag{27}$$

where, as before, pl_{jk} is the plausibility that instance j belongs to class ω_k. The combined contour function after pooling the evidence of the K nearest neighbors can then be computed up to a multiplicative constant as

$$\widehat{pl}(\omega_k) \propto \prod_{x_j \in \mathcal{N}_K(x)} [1 - \beta_k(d_j) + \beta_k(d_j) pl_{jk}], \quad k = 1, \ldots, c. \tag{28}$$

We note that the term in the right-hand side of (28) can be computed in time proportional to the number K of neighbors and the number c of classes, after the K nearest neighbors of x have been found. The contour function is all we need to make decisions using the maximum-plausibility rule and, as we will see in the next section, to train the classifier by maximizing the evidential likelihood criterion.

Comparison with the γ_k-EKNN rule

In the case of fully supervised data, mass functions m_j are certain (they have only one focal set and it is a singleton). We then have $pl_{jk} = y_{jk} \in \{0, 1\}$, and (27) becomes

$$\widehat{pl}_j(\omega_k) = 1 - \beta_k(d_j) + \beta_k(d_j) y_{jk}, \quad k = 1, \ldots, c. \tag{29}$$

Comparing Eqs (19) and (29), we can see that the contour functions $\widehat{pl}_j$ computed by the γ_k-EKNN and CD-EKNN classifiers are different, except when $\beta_1 = \beta_2 = \ldots = \beta_c$, in which case Eq. (19) becomes

$$\widehat{pl}_j(\omega_k) = 1 - \beta(d_j) \sum_{l=1}^{c} y_{jl} + \beta(d_j) y_{jk} = 1 - \beta(d_j) + \beta(d_j) y_{jk},$$

which is identical to (29). Consequently, with fully supervised data, the two classifiers yield different decisions in general. To explain this difference, we need to understand the different interpretations of coefficients $\beta_k(d_j)$ in both rules. In the γ_k-EKNN model, $\beta_k(d_j)$ is the degree of belief that a neighbor situated at distance d_j and belonging to class ω_k provides reliable information. In contrast, in the CD-EKNN model, $\beta_k(d_j)$ is the degree of belief that a neighbor situated at distance d_j provides reliable information, given that the true class of the instance under consideration is ω_k.

To illustrate this difference between the two models, let us consider the three-class data shown in Figures 1 and 2, which display the contour lines in feature space of output masses computed by, respectively, the CD-EKNN and γ_k-EKNN classifiers with parameters $K = 10$, $\alpha = 0.5$ and $\gamma_1 = \gamma_2 = \gamma_3 = 1$. We can see that the γ_k-EKNN classifier assigns masses only to singletons and the whole frame Ω, whereas the CD-EKNN assigns masses to all non empty subsets of Ω.

As far as computational complexity is concerned, the γ_k-EKNN and CD-EKNN classifiers require exactly the same number of operations to make a decision: after the K nearest neighbors of input vector x have been found, computing the c combined plausibilities $\widehat{pl}(\omega_k)$ up to a multiplicative constant can be done in time proportional to K and c.

Comparison with the (α, γ)-EKNN rule

In the case of partially supervised data, the reference method is the (α, γ)-EKNN rule defined by Eqs (23)-(24). This method depends on only two parameters, whereas the CD-EKNN rule has $c + 1$ parameters. By comparing Eqs (24) and (27), we can see that the two methods yield the same contour functions (but not the same combined mass functions) when $\beta_1 = \beta_2 = \ldots = \beta_c$. In the general case, the two methods lead to different decisions, but have the same complexity when using the maximum-plausibility decision rule.

3.3. Learning

To learn the parameters $\theta = (\alpha, \gamma_1, \ldots, \gamma_c)$ of the CD-EKNN classifier defined in Section 3.2, we propose to maximize the *evidential likelihood* function recalled in Section 2.3. Before we introduce the evidential likelihood for this model, let us recall the expression of the classical likelihood in the case of fully supervised data $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^n$. Let $\widehat{pl}_i$ the contour function computed for instance i based on its K nearest neighbors using (28), $\widehat{pl}_{ik} = \widehat{pl}_i(\omega_k)$, and $\widehat{p}_{ik}$ the probability of class ω_k obtained from $\widehat{pl}_i$ after normalization:

$$\widehat{p}_{ik} = \frac{\widehat{pl}_{ik}}{\sum_{l=1}^{c} \widehat{pl}_{il}}. \tag{30}$$

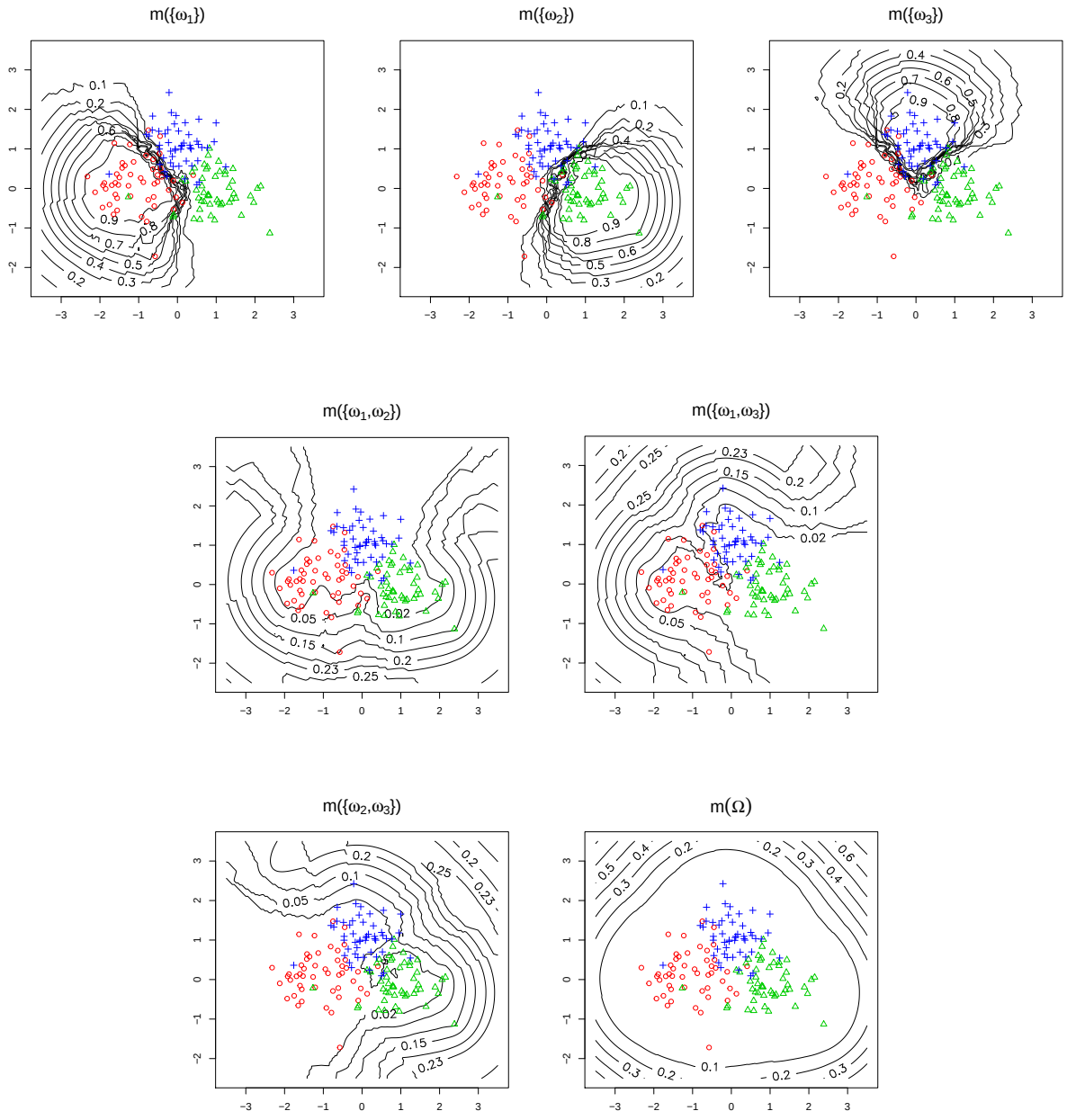


Figure 1: Contour plots of output masses computed by the CD-EKNN classifier for a three-class dataset.

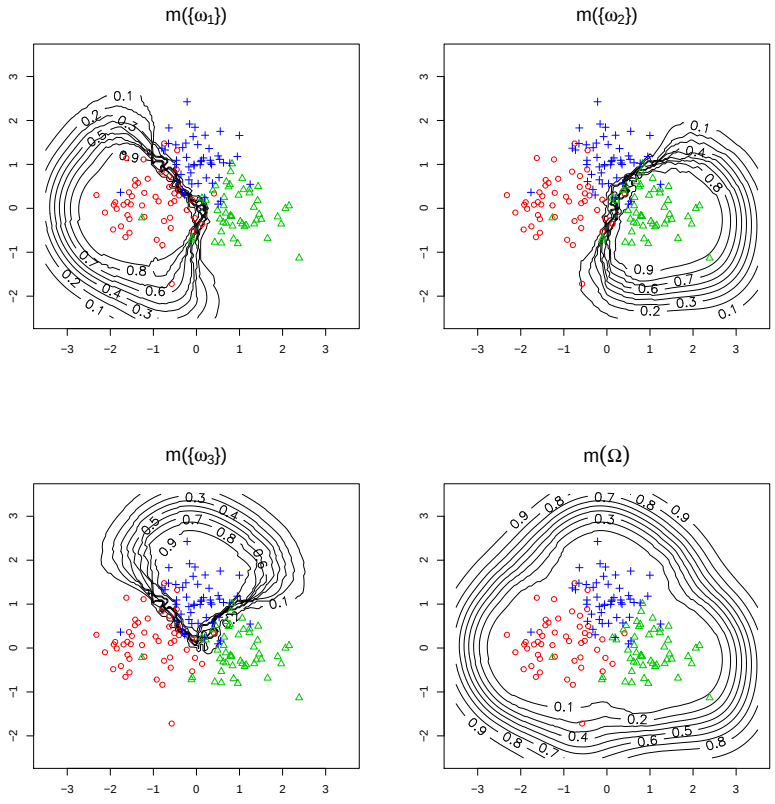


Figure 2: Contour plots of output masses computed by the γ_k-EKNN classifier for a three-class dataset..

Viewing $\widehat{p}_{ik}$ as a model of the conditional probability that instance i belongs to class ω_k given x_i, the conditional likelihood (given feature vectors $x_1, \ldots, x_n$) after observing the true class labels $y_1, \ldots, y_n$ is

$$L_c(\theta) = \prod_{i=1}^{n} \prod_{k=1}^{c} \widehat{p}_{ik}^{\,y_{ik}}. \tag{31}$$

This criterion can be used instead of the MSE criterion (21) in the case of fully supervised data. However, a distinctive advantage of the likelihood criterion as compared to MSE is that the former can be more easily extended to partially supervised learning, thanks to the notion of evidential likelihood recalled in Section 2.3.

Let us assume that the learning set is of the form $\mathcal{L} = \{(x_i, m_i)\}_{i=1}^{n}$, where m_i is a mass function that represents our partial knowledge of the class of x_i. The expected plausibility $\mathbb{E}_\theta[pl_i(Y_i)]$ in (17) can be written as

$$\mathbb{E}_\theta[pl_i(Y_i)] = \sum_{k=1}^{c} \widehat{p}_{ik} pl_{ik}.$$

The evidential likelihood (17) is then

$$L_e(\theta) = \prod_{i=1}^{n} \sum_{k=1}^{c} \widehat{p}_{ik} pl_{ik}, \tag{32}$$

and its logarithm is

$$\ell_e(\theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{c} \widehat{p}_{ik} pl_{ik} \right). \tag{33}$$

We can check that the evidential likelihood (32) boils down to the classical likelihood (31) when all mass functions m_i are certain, i.e., when $pl_i(\omega_k) = y_{ik}$ for all i and k.

The evidential log-likelihood $\ell_e(\theta)$ can be maximized using an iterative optimization procedure such as the Nelder-Mead or BFGS algorithms [13]. To handle the constraints $0 \leq \alpha \leq 1$ and $\gamma_k \geq 0$, we can reparameterize the problem by introducing new parameters $\xi \in \mathbb{R}$ and $\eta_k \in \mathbb{R}$ such that $\alpha = [1 + \exp(\xi)]^{-1}$ and $\gamma_k = \eta_k^2$. Pseudo-code for the calculation of the evidential likelihood is given in Algorithm 1 and the learning procedure is sketched in Algorithm 2. We note that the nearest neighbors are computed only once at the beginning of the procedure, and heuristics for the initialization of ξ and η_k, $k = 1, \ldots, c$ are given in Algorithm 2. The expression of the gradient of $\ell_e(\theta')$, with $\theta' = (\xi, \eta_1, \ldots, \eta_c)$, is given in Appendix A.

4. Numerical Experiments

In this section, we present some results with simulated and real datasets. In Section 4.1, we first consider the fully supervised case, in which the true class labels are provided to the learning algorithms. In Section 4.2, we then simulate label uncertainty by corrupting labels with noise and representing uncertainty using suitable mass functions.

Algorithm 1 Calculation of the evidential likelihood.

Require: K, $\theta' = (\xi, \eta_1, \ldots, \eta_c)$
Require: Matrix $D = (d_{ij})$ (of size $n \times n$) of distances between the n training vectors
Require: Matrix $I = (I_{ij})$ (of size $n \times K$), where I_{ij} is the index of the j-th neighbor of x_i in the learning set
Require: Matrix $\mathsf{PL} = (pl_{ik})$ (of size $n \times c$) of soft labels
  $\alpha := [1 + \exp(\xi)]^{-1}$
  for $k = 1$ **to** c **do**
    $\gamma_k := \eta_k^2$
  end for
  for $i = 1$ **to** n **do**
    for $k = 1$ **to** c **do**
      for $j = 1$ **to** K **do**
        $j' := I_{ij}$
        $\beta_{ijk} := \alpha \exp(-\gamma_k d_{ij'}^2)$
        $\widehat{pl}_{ijk} := 1 - \beta_{ijk} + \beta_{ijk} pl_{j'k}$
      end for
      $\widehat{pl}_{ik} := \prod_{j=1}^{K} \widehat{pl}_{ijk}$
    end for
    $\widehat{p}_{ik} := \widehat{pl}_{ik} / \sum_{l=1}^{c} \widehat{pl}_{il}$
  end for
  $\ell_e(\theta') := \sum_{i=1}^{n} \log \left(\sum_{k=1}^{c} \widehat{p}_{ik} pl_{ik} \right)$
Ensure: $\ell_e(\theta')$

Algorithm 2 Learning algorithm.

Require: Learning set $\mathcal{L} = \{x_i, m_i\}$, number K of neighbors
  Compute the distance matrix $D = (d_{ij})$
  for $i = 1$ **to** n **do**
    Find the K nearest neighbors of x_i in the learning set. Let I_{ij} denote the index of the j-th neighbor of x_i.
  end for
  $\xi_0 := 0$
  $\eta_{k0} := \left(\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j<i} d_{ij}^2 \right)^{-1/2}$, $k = 1, \ldots, c$
  $\theta'_0 := (\xi_0, \eta_{10}, \ldots, \eta_{c0})$
  Maximize $\ell_e(\theta')$ w.r.t. θ', starting from θ'_0
Ensure: $\widehat{\theta'} := \arg\max_{\theta'} \ell_e(\theta')$

4.1. Fully supervised data

In the experiments reported in this and the following section, we considered one simulated data distribution and five real data sets. The simulated data were generated from $c = 2$ Gaussian distributions with densities $\mathcal{N}(\mu_k, \sigma_k^2 I)$, where $\mu_1 = (0, 0, 0)^T$, $\mu_2 = (1, 0, 0)^T$, $\sigma_1^2 = 0.1I$, $\sigma_2^2 = 2I$, and I is the 3×3 identity matrix. Each simulated dataset had 50 vectors in each class. The real datasets[3] were the following:

- The Wine data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. There are $n = 178$ instances, $p = 13$ features corresponding to 13 constituents, and $c = 3$ classes corresponding to three types of wines.

- The Ionosphere dataset was collected by a radar system and consists of phased array of 16 high-frequency antennas with a total transmitted power of the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not. The Ionosphere dataset has $n = 351$ instances, $p = 34$ features and $c = 2$ classes.

- The Ecoli dataset contains data about protein localization sites in E. coli bacteria. We used only the quantitative attributes (2, 3, 6, 7, and 8) and the four most frequent classes: 'im', 'pp', 'imU' and 'cp', resulting in a dataset with n=307 objects, $p = 5$ attributes and $c = 4$ classes.

- The Sonar data were used by Gorman and Sejnowski [14] in a study of the classification of sonar signals using a neural network. The task is to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The dataset has $n = 204$ instances, $p = 60$ attributes and $c = 2$ classes.

- The Heart data [16] were collected as part of a study aiming to establish the intensity of ischemic heart disease risk factors in a high-incidence region in South Africa. There are $p = 8$ numeric attributes, and the response variable is the presence or absence of myocardial infarction (MI). There are 160 positive cases in this data, and a sample of 302 negative cases (controls).

For each dataset, we considered six classifiers:

1. The CD-EKNN rule with c scale parameters $\gamma_1, \ldots, \gamma_c$ trained with the likelihood criterion (32);
2. The CD-EKNN rule trained with the MSE criterion (21);
3. The γ_k-EKNN rule recalled in Section 3.1, trained with the MSE criterion (21);

[3]The Ionosphere, Sonar, Ecoli and Wine data can be retrieved from the UCI Machine Learning Repository at `http://archive.ics.uci.edu/ml`. The Heart data can be downloaded from `https://web.stanford.edu/~hastie/ElemStatLearn/`.

4. The γ_k-EKNN rule trained with the likelihood criterion (32);
5. The (α, γ)-EKNN rule based on classical discounting and the likelihood criterion (32);
6. The voting K-NN rule.

Figure 3 shows the contours of the negative log-likelihood function as a function of parameters γ_1 and γ_2 for the CD-EKNN and γ_k-EKNN classifiers, for two-class datasets Ionosphere (Figures 3a and 3b) and Sonar (Figures 3c and 3d). Parameter α was fixed at its maximum-likelihood value for the CD-EKNN classifier, and at 0.5 for the γ_k-EKNN classifier. As we can see, the optimum values of parameters γ_k are very different for the two classifiers. For instance, for the Ionosphere dataset, we have $(\widehat{\gamma}_1, \widehat{\gamma}_2) \approx (0, 0.11)$ for the CD-EKNN classifier, and $(\widehat{\gamma}_1, \widehat{\gamma}_2) \approx (0.15, 0)$ for the γ_k-EKNN classifier. This difference comes as no surprise given that these parameters have different interpretations in the two models, as discussed in Section 3.2.

The leave-one-out error rates achieved by the six considered classifiers are displayed in Figure 4 as functions of the number K of neighbors. For the simulated data (Figure 4a), the reported error rates are averages over 10 datasets. The error curves of classifiers trained with the MSE and likelihood criteria are drawn, respectively, as black and red lines. The main findings from these results are the following:

- The CD-EKNN and γ_k-EKNN classifiers basically have similar performances with fully supervised data regardless of the learning criterion. However, the γ_k-EKNN classifier yielded slightly higher error rates with the likelihood criterion for $K \geq 15$ on the simulated and Heart data (Figures 4a and 4f).

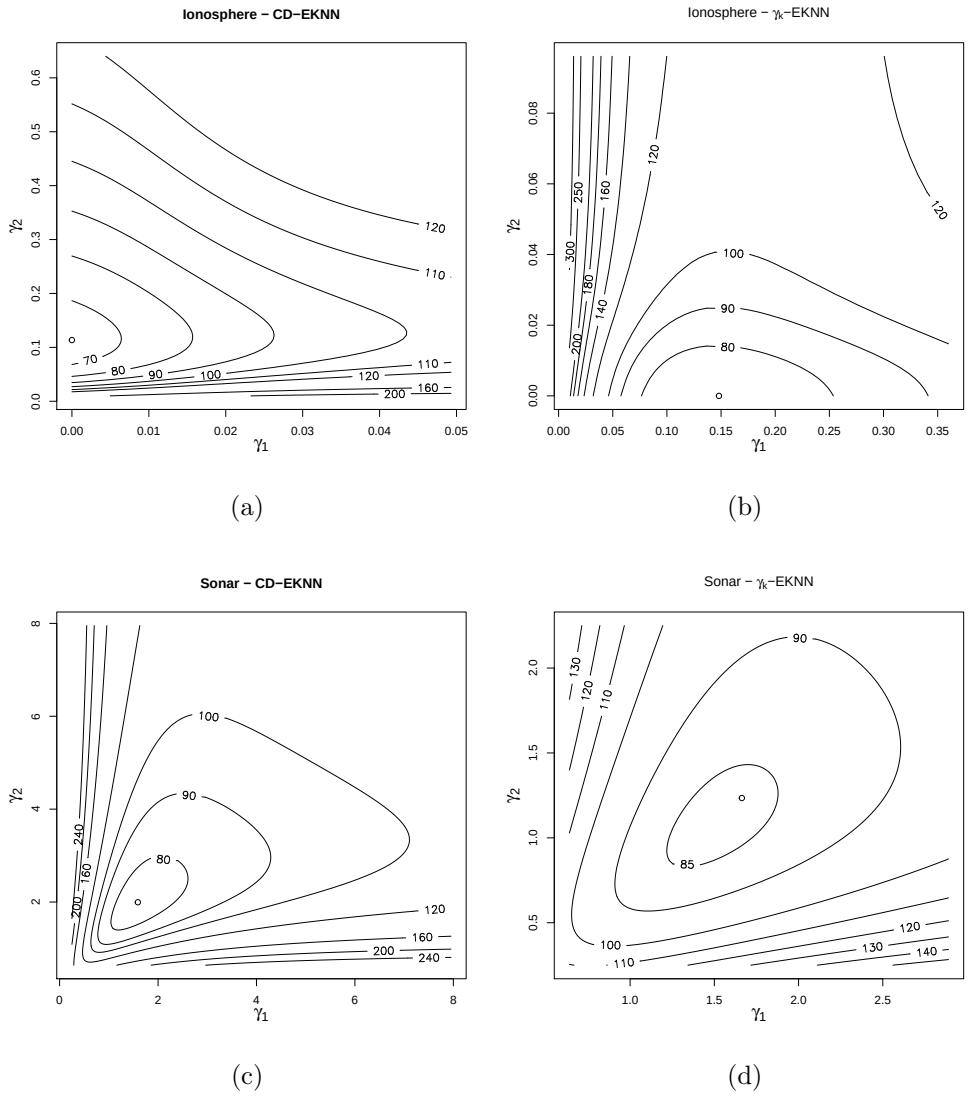
- The (α, γ)-EKNN classifier based on classical discounting performed significantly worse than the CD-EKNN and γ_k-EKNN classifiers, especially on the simulated, Ionosphere and Ecoli datasets (Figures 4a, 4c and 4d).

- The voting K-NN rule generally has the worst performances, except on the Heart data (Figure 4f). This result confirms previous findings [6] [39].

From this first experiment, we may conclude that the CD-EKNN rule performs neither better nor worse than the γ_k-EKNN rule, while both rules generally perform better than the (α, γ)-EKNN and voting K-NN rules. The CD-EKNN performs equally well when trained with the likelihood and MSE criteria, while the γ_k-EKNN rule seems to perform a little better with the original MSE criterion for large K. The case of partially supervised data will be addressed in the next section.

4.2. Partially supervised datasets

To study the performances of the CD-EKNN classifier with partially supervised data, we simulated soft labels for the six datasets used in the previous section, using the method described in [4] and [29]. For each instance i, a probability p_i was generated from a beta distribution with mean $\mu = 0.5$ and variance 0.04. Then, with probability p_i, the class label y_i of instance i was replaced by y'_i picked randomly from Ω. Otherwise, we set $y'_i = y_i$. The class information for instance i then consists of the "noisy" label y'_i and the probability p_i



(a) (b)

(c) (d)

Figure 3: Contours of the negative log-likelihood as a function of parameters γ_1 and γ_2 for the CD-EKNN (a)-(c) and γ_k-EKNN (b)-(d) classifiers, for the Ionosphere (a)-(b) and Sonar (c)-(d) datasets.

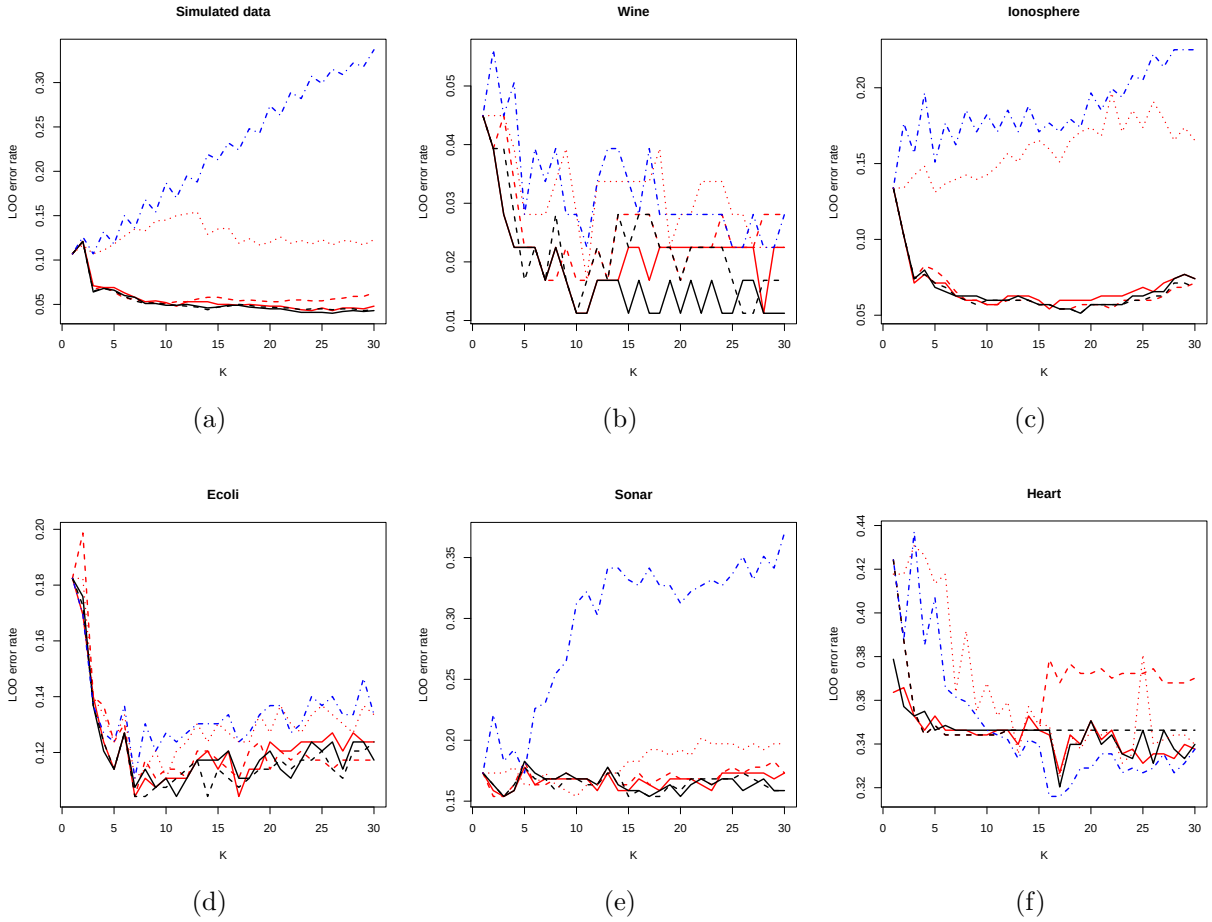


Figure 4: Leave-one-out error rates vs. number K of neighbors for fully supervised datasets. The methods are: the CD-EKNN classifier (solid lines), the (α, γ)-EKNN classifier (dashed lines), the γ_k-EKNN classifier (dotted lines) and the voting K-NN rule (dash-dotted lines). Black and red curves correspond, respectively, to classifiers trained with the MSE and likelihood criteria. This figure is better viewed in color.

that this label was generated at random. A labeling contour function pl_i was then defined as $pl_i(y_i') = 1$ and $pl_i(\omega) = p_i$ for all $\omega \neq y_i'$. This procedure ensures that the soft label pl_i is all the more uncertain that the label with maximum plausibility has the more chance of being incorrect. In particular, if $p_i = 1$, the noisy label is completely random and the soft label is vacuous (it verifies $pl(\omega_k) = 1$ for all k). If $p_i = 0$, then the noisy label y_i' is fully reliable (it is equal for sure to the true label), and the soft label is certain (it verifies $pl_i(y_i') = 1$ and $pl_i(\omega) = 0$ for all $\omega \neq y_i'$). This procedure simulates real situations in which labels are provided, e.g., by an expert or by some indirect method, and the uncertainty of the labeling process can be effectively quantified [4], [29].

Example 2. *For instance, assume that* $\Omega = \{\omega_1, \omega_2, \omega_3\}$ *and the true label is* $y_i = \omega_1$ *and* $p_i = 0.7$*. With probability 0.7,* ω_1 *is replaced by a label randomly picked in* Ω*. Assume that* ω_1 *is replaced by* ω_2*, so the noisy label is* $y_i' = \omega_2$*. The soft label is finally* $pl_i(\omega_1) = 0.7$*,* $pl_i(\omega_2) = 1$ *and* $pl_i(\omega_3) = 0.7$*.*

For each real dataset, we generated 10 learning sets with different randomly generated soft labels. We compared the performances of the following four classifiers:

1. The CD-EKNN rule trained with partial labels and the evidential likelihood criterion (32);
2. The (α, γ)-EKNN rule trained with partial labels and the evidential likelihood criterion (32);
3. The γ_k-EKNN rule trained with noisy labels y_i' and the MSE criterion (21);
4. The voting K-NN rule with noisy labels y_i'.

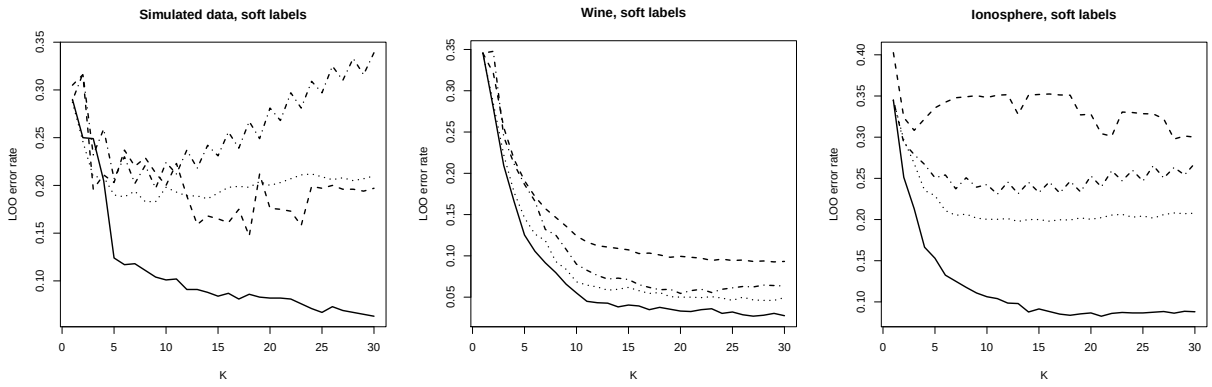
We note that noisy labels were used with the γ_k-EKNN and voting K-NN rules because these classifiers can only handle fully supervised data and there is no obvious way to use them with partially supervised data.

The results of this experiment are reported in Figure 5. We can see that there is considerable variability of the results across the datasets, but the main findings from these results can be summarized as follows:

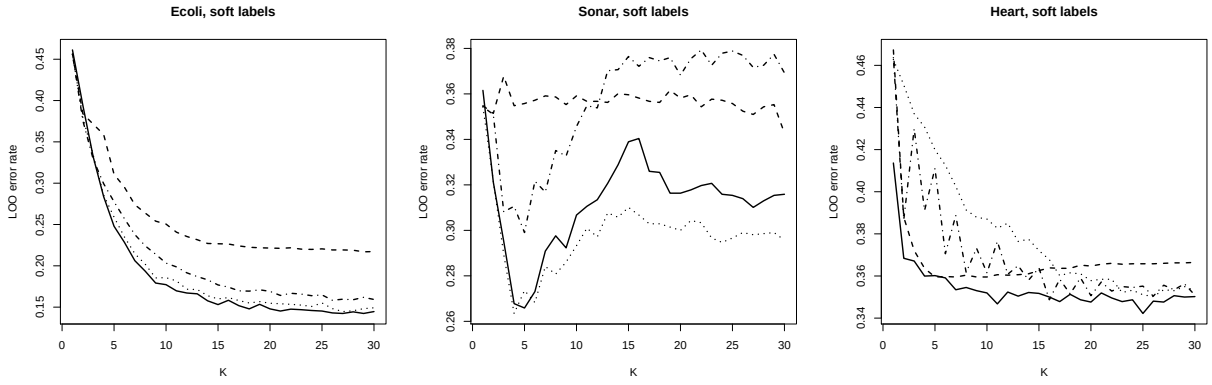
- The CD-EKNN classifier performs as well as the (α, γ)-EKNN rule on the Sonar data (Figure 5e), and strictly better on all other datasets. The performance gain is particularly large with the simulated and Ionosphere datasets (Figures 5a and 5c).

- The (α, γ)-EKNN rule performs poorly on some datasets, namely, the simulated, Ionosphere and Heart datasets (Figures 5a, 5c and 5f). This confirms the interest of using contextual discounting instead of classical discounting with the EKNN rule in the case of partially supervised data.

- The γ_k-EKNN and voting K-NN rules using noisy labels, generally perform poorly. This is not surprising, as they are not able to use the information about label uncertainty contained in soft labels. This result confirms similar findings reported in [4], [8] and [29] for parametric classifiers. The γ_k-EKNN rule proved even less robust that the simpler voting K-NN rule, which yielded lower error rates for the Wine, Ionosphere and Ecoli datasets (Figures 5b, 5c and 5d).



(a) (b) (c)



(d) (e) (f)

Figure 5: Leave-one-out error rates vs. number K of neighbors for partially supervised datasets. The methods are: the CD-EKNN classifier trained with the likelihood criterion (solid lines), the (α, γ)-EKNN classifier trained with the likelihood criterion (dashed lines), the γ_k-EKNN classifier trained with the MSE criterion and noisy labels (dotted lines) and the voting K-NN rule with noisy labels (dash-dotted lines).

5. Conclusions

The EKNN classifier introduced in [6] and perfected in [39] has proved very efficient for fully supervised classification. However, because it uses different discount rates for neighbors from different classes, the method cannot be readily extended to the partially supervised learning situation, in which we only have uncertain information about the class of learning instances. The simple approach outlined in [11] lacks flexibility as it relies on a single discount rate parameter; it also suffers from high computational complexity as it requires to combine the full mass functions.

In this paper, we have proposed a new variant of the EKNN classifier suitable for partially supervised classification, by replacing classical discounting with the contextual discounting operation introduced in [25]. The underlying model is based on the assumption that the reliability of the information from different neighbors depends on the class of the pattern to be classified. We also replaced the MSE criterion by the conditional evidential likelihood introduced in [8] and already used for partially supervised logistic regression in [29]. The resulting CD-EKNN classifier was shown to perform very well with partially supervised data, while performing as well as the original EKNN classifier with fully supervised data. Its computational complexity is identical to that of the original EKNN classifier, and it is not increased when learning from partially supervised data.

In contrast with the original EKNN classifier, which assigns masses only to singletons and the whole frame of discernment, the CD-EKNN classifier generates more general mass functions, as a result of applying the contextual discounting operation. In future work, it will be interesting to find out whether this richer information can be exploited for, e.g., classifier combination. Beyond contextual discounting, other mass correction mechanisms such as introduced in [27], [23]-[24] and [28] could also be investigated.

Acknowledgements

This research was supported by the Labex MS2T (reference ANR-11-IDEX-0004-02) and by the Centre of Excellence in Econometrics, Research Administration Center at Chiang Mai University.

Appendix A. Gradient calculation

Let d_{ij} denote the distance between x_i and x_j and let $\beta_{ijk} = \alpha \exp(-\gamma_k d_{ij}^2)$. The evidential log-likelihood (33) can be written as

$$\ell_e(\theta) = \sum_{i=1}^{n} \ell_{e,i}(\theta),$$

with

$$\ell_{e,i}(\theta) = \log\left(\sum_{k=1}^{c} \widehat{p}_{ik} pl_{ik}\right),$$

$$\widehat{p}_{ik} = \frac{\widehat{pl}_{ik}}{\sum_{l=1}^{c} \widehat{pl}_{il}},$$

and

$$\widehat{pl}_{ik} \propto \prod_{x_j \in \mathcal{N}_K(x_i)} \left[1 - \beta_{ijk}(1 - pl_{jk})\right].$$

Finally, we introduce new parameters ξ and η_k, $k = 1, \ldots, c$ such that $\alpha = [1 + \exp(\xi)]^{-1}$, $\gamma_k = \eta_k^2$ and $\theta' = (\xi, \eta_1, \ldots, \eta_c)$. With these notations, we have

$$\frac{\partial \ell_e(\theta')}{\partial \eta_k} = \sum_{i=1}^{n} \frac{\partial \ell_{e,i}(\theta')}{\partial \eta_k} \tag{A.1}$$

and

$$\frac{\partial \ell_e(\theta')}{\partial \xi} = \sum_{i=1}^{n} \frac{\partial \ell_{e,i}(\theta')}{\partial \xi}. \tag{A.2}$$

Calculation of $\frac{\partial \ell_{e,i}(\theta')}{\partial \eta_k}$. We have

$$\frac{\partial \ell_{e,i}(\theta')}{\partial \eta_k} = \frac{\partial \ell_{e,i}(\theta')}{\partial \gamma_k} \frac{\partial \gamma_k}{\partial \eta_k} = 2\eta_k \sum_{q=1}^{c} \frac{\partial \ell_{e,i}(\theta')}{\partial \widehat{p}_{iq}} \frac{\partial \widehat{p}_{iq}}{\partial \gamma_k}, \tag{A.3}$$

with

$$\frac{\partial \ell_{e,i}(\theta')}{\partial \widehat{p}_{iq}} = \frac{pl_{iq}}{\sum_{k=1}^{c} \widehat{p}_{ik} pl_{ik}} \tag{A.4}$$

and

$$\frac{\partial \widehat{p}_{iq}}{\partial \gamma_k} = \frac{\partial \widehat{p}_{iq}}{\partial \widehat{pl}_{ik}} \frac{\partial \widehat{pl}_{ik}}{\partial \gamma_k}. \tag{A.5}$$

Now,

$$\frac{\partial \widehat{p}_{iq}}{\partial \widehat{pl}_{ik}} = \begin{cases} \dfrac{\sum_{r=1}^{c} \widehat{pl}_{ir} - \widehat{pl}_{ik}}{\left(\sum_{r=1}^{c} \widehat{pl}_{ir}\right)^2} = \dfrac{1 - \widehat{p}_{ik}}{\sum_{r=1}^{c} \widehat{pl}_{ir}} & \text{if } q = k \\ \dfrac{-\widehat{pl}_{iq}}{\left(\sum_{r=1}^{c} \widehat{pl}_{ir}\right)^2} & \text{if } q \neq k, \end{cases} \tag{A.6}$$

and

$$\frac{\partial \widehat{pl}_{ik}}{\partial \gamma_k} = \sum_{j=1}^{K} \frac{\partial \widehat{pl}_{ik}}{\partial \beta_{ijk}} \frac{\partial \beta_{ijk}}{\partial \gamma_k}, \tag{A.7}$$

with

$$\frac{\partial \widehat{pl}_{ik}}{\partial \beta_{ijk}} = -(1 - pl_{jk}) \prod_{j' \neq j} \left[1 - \beta_{ij'k}(1 - pl_{j'k})\right] = \frac{-\widehat{pl}_{ik}(1 - pl_{jk})}{1 - \beta_{ijk}(1 - pl_{jk})}, \tag{A.8}$$

and

$$\frac{\partial \beta_{ijk}}{\partial \gamma_k} = -\alpha d_{ij}^2 \exp\left(-\gamma_k d_{ij}^2\right) = -\beta_{ijk} d_{ij}^2. \tag{A.9}$$

Calculation of $\frac{\partial \ell_{e,i}(\theta')}{\partial \xi}$. We have

$$\frac{\partial \ell_{e,i}(\theta')}{\partial \xi} = \frac{\partial \ell_{e,i}(\theta')}{\partial \alpha}\frac{\partial \alpha}{\partial \xi} = \alpha(1-\alpha)\sum_{q=1}^{c}\frac{\partial \ell_{e,i}(\theta')}{\partial \widehat{p}_{iq}}\frac{\partial \widehat{p}_{iq}}{\partial \alpha}, \tag{A.10}$$

where $\frac{\partial \ell_{e,i}(\theta')}{\partial \widehat{p}_{iq}}$ is given by (A.4) and

$$\frac{\partial \widehat{p}_{iq}}{\partial \alpha} = \sum_{k=1}^{c}\frac{\partial \widehat{p}_{iq}}{\partial \widehat{pl}_{ik}}\frac{\partial \widehat{pl}_{ik}}{\partial \alpha}. \tag{A.11}$$

Here, $\frac{\partial \widehat{p}_{iq}}{\partial \widehat{pl}_{ik}}$ is given by (A.6), and

$$\frac{\partial \widehat{pl}_{ik}}{\partial \alpha} = \sum_{j=1}^{K}\frac{\partial \widehat{pl}_{ik}}{\partial \beta_{ijk}}\frac{\partial \beta_{ijk}}{\partial \alpha}, \tag{A.12}$$

where $\frac{\partial \widehat{pl}_{ik}}{\partial \beta_{ijk}}$ is given by (A.8) and

$$\frac{\partial \beta_{ijk}}{\partial \alpha} = \exp(-\gamma_k d_{ij}^2) = \frac{\beta_{ijk}}{\alpha}. \tag{A.13}$$

The complete procedure is summarized in Algorithm 3.

Algorithm 3 Gradient calculation.

Require: $\{\eta_k, \gamma_k\}_{k=1}^{c}$, ξ, α, $\{pl_{ik}, \widehat{pl}_{ik}, \widehat{p}_{ik}\}_{i,k}^{n,c}$, $\{\beta_{ijk}\}_{i,j,k}^{n,K,c}$, $\{d_{ij}\}_{i,j}^{n,K}$

for $i = 1$ **to** n **do**
  for $q = 1$ **to** c **do**
    Compute $\frac{\partial \ell_e(\theta')}{\partial \widehat{p}_{iq}}$ using (A.4)
  end for
  for $k = 1$ **to** c **do**
    for $j = 1$ **to** K **do**
      Compute $\frac{\partial \widehat{pl}_{ik}}{\partial \beta_{ijk}}$, $\frac{\partial \beta_{ijk}}{\partial \gamma_k}$ and $\frac{\partial \beta_{ijk}}{\partial \alpha}$ using (A.8), (A.9) and (A.13)
    end for
    Compute $\frac{\partial \widehat{pl}_{ik}}{\partial \gamma_k}$ and $\frac{\partial \widehat{pl}_{ik}}{\partial \alpha}$ using (A.7) and (A.12)
    for $q = 1$ **to** c **do**
      Compute $\frac{\partial \widehat{p}_{iq}}{\partial \widehat{pl}_{ik}}$ using (A.6)
      Compute $\frac{\partial \widehat{p}_{iq}}{\partial \gamma_k}$ using (A.5)
    end for
    Compute $\frac{\partial \ell_{e,i}(\theta')}{\partial \eta_k}$ using (A.3)
  end for $\#k$ loop
  for $q = 1$ **to** c **do**
    Compute $\frac{\partial \widehat{p}_{iq}}{\partial \alpha}$ using (A.11)
  end for
  Compute $\frac{\partial \ell_{e,i}(\theta')}{\partial \xi}$ using (A.10)
end for $\#i$ loop
for $k = 1$ **to** c **do**
  Compute $\frac{\partial \ell_e(\theta')}{\partial \eta_k}$ using (A.1)
end for
Compute $\frac{\partial \ell_e(\theta')}{\partial \xi}$ using (A.2)

Ensure: Gradient $\frac{\partial \ell_e(\theta')}{\partial \theta'} = \left(\frac{\partial \ell_{e,i}(\theta')}{\partial \eta_1}, \ldots, \frac{\partial \ell_{e,i}(\theta')}{\partial \eta_K}, \frac{\partial \ell_{e,i}(\theta')}{\partial \xi} \right)^T$

References

[1] H. Altınçay. Ensembling evidential k-nearest neighbor classifiers through multi-modal perturbation. *Applied Soft Computing*, 7(3):1072–1083, 2007.

[2] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, Ma, 2006.

[3] X.-L. Chen, P.-H. Wang, Y.-S. Hao, and M. Zhao. Evidential KNN-based condition monitoring and early warning method with applications in power plant. *Neurocomputing*, 2018.

[4] E. Côme, L. Oukhellou, T. Denœux, and P. Aknin. Learning from partially supervised data using mixture models and belief functions. *Pattern Recognition*, 42(3):334–348, 2009.

[5] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[6] T. Denœux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. on Systems, Man and Cybernetics*, 25(05):804–813, 1995.

[7] T. Denœux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern Recognition*, 30(7):1095–1107, 1997.

[8] T. Denœux. Maximum likelihood estimation from uncertain data in the belief function framework. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):119–130, 2013.

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

[10] T. Denœux, D. Dubois, and H. Prade. Representations of uncertainty in artificial intelligence: Beyond probability and possibility. In P. Marquis, O. Papini, and H. Prade, editors, *A Guided Tour of Artificial Intelligence Research*, chapter 4. Springer Verlag, 2019.

[11] T. Denœux and L. M. Zouhal. Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, 122(3):47–62, 2001.

[12] D. Dubois and H. Prade. A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. *International Journal of General Systems*, 12(3):193–226, 1986.

[13] G. H. Givens and J. A. Hoeting. *Computational Statistics*. John Wiley & Sons, Hoboken, NJ, 2013.

[14] R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[15] N. Guettari, A. S. Capelle-Laizé, and P. Carré. Blind image steganalysis based on evidential k-nearest neighbors. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2742–2746, Sept 2016.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[17] L. Jiao, T. Denœux, and Q. Pan. Evidential editing k-nearest neighbor classifier. In S. Destercke and T. Denoeux, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 461–471, Cham, 2015. Springer International Publishing.

[18] L. Jiao, Q. Pan, X. Feng, and F. Yang. An evidential k-nearest neighbor classification method with weighted attributes. In *Proceedings of the 16th International Conference on Information Fusion*, pages 145–150, July 2013.

[19] O. Kanjanatarakul, S. Kuson, and T. Denœux. An evidential k-nearest neighbor classifier based on contextual discounting. In F. Cuzzolin, T. Denœux, S. Destercke, and A. Martin, editors, *Belief Functions: Theory and Applications: Fourth International Conference (BELIEF 2018)*, number 11069 in Lecture Notes in Artificial Intelligence, pages 155–162. Springer, Compiègne, France, Sept. 2018.

[20] C. Lian, S. Ruan, and T. Denœux. An evidential classifier based on feature selection and two-step classification strategy. *Pattern Recognition*, 48:2318–2327, 2015.

[21] C. Lian, S. Ruan, and T. Denœux. Dissimilarity metric learning in the belief function framework. *IEEE Transactions on Fuzzy Systems*, 24(6):1555–1564, 2016.

[22] Z.-G. Liu, Q. Pan, and J. Dezert. A new belief-based K-nearest neighbor classification method. *Pattern Recognition*, 46(3):834–844, 2013.

[23] D. Mercier, E. Lefèvre, and F. Delmotte. Belief functions contextual discounting and canonical decompositions. *International Journal of Approximate Reasoning*, 53(2):146–158, 2012.

[24] D. Mercier, F. Pichon, and E. Lefèvre. Corrigendum to "Belief functions contextual discounting and canonical decompositions" [International Journal of Approximate Reasoning 53 (2012) 146–158]. *International Journal of Approximate Reasoning*, 70:137 – 139, 2016.

[25] D. Mercier, B. Quost, and T. Denœux. Refined modeling of sensor reliability in the belief function framework using contextual discounting. *Information Fusion*, 9(2):246–258, 2008.

[26] N. K. Pal and S. Gosh. Some classification algorithms integrating Dempster-Shafer theory of evidence with the rank nearest neighbor rule. *IEEE Trans. on Systems, Man and Cybernetics – Part A*, 31(1):59–66, 2001.

[27] F. Pichon, T. Denœux, and D. Dubois. Relevance and truthfulness in information correction and fusion. *International Journal of Approximate Reasoning*, 53(2):159–175, 2012.

[28] F. Pichon, D. Mercier, E. Lefèvre, and F. Delmotte. Proposition and learning of some belief function contextual correction mechanisms. *International Journal of Approximate Reasoning*, 72:4–42, 2016.

[29] B. Quost, T. Denœux, and S. Li. Parametric classification with soft labels using the evidential EM algorithm: linear discriminant analysis versus logistic regression. *Advances in Data Analysis and Classification*, 11(4):659–690, Dec 2017.

[30] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, N.J., 1976.

[31] H. Shen and K.-C. Chou. Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types. *Biochemical and Biophysical Research Communications*, 334(1):288–292, 2005.

[32] P. Smets. Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9:1–35, 1993.

[33] P. Smets. The α-junctions: Combination operators applicable to belief functions. In D. M. Gabbay, R. Kruse, A. Nonnengart, and H. J. Ohlbach, editors, *Qualitative and Quantitative Practical Reasoning*, pages 131–153, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[34] P. Smets. Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38:133–147, 2005.

[35] P. Smets and R. Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66:191–243, 1994.

[36] Z.-G. Su, T. Denoeux, Y.-S. Hao, and M. Zhao. Evidential K-NN classification with enhanced performance via optimizing a class of parametric conjunctive t-rules. *Knowledge-Based Systems*, 142:7–16, 2018.

[37] Z.-G. Su and P.-H. Wang. Improved adaptive evidential k-NN rule and its application for monitoring level of coal powder filling in ball mill. *Journal of Process Control*, 19(10):1751–1762, 2009.

[38] A. Trabelsi, Z. Elouedi, and E. Lefevre. A novel k-NN approach for data with uncertain attribute values. In S. Benferhat, K. Tabia, and M. Ali, editors, *Advances in Artificial Intelligence: From Theory to Practice*, pages 160–170, Cham, 2017. Springer International Publishing.

[39] L. M. Zouhal and T. Denœux. An evidence-theoretic k-NN rule with parameter optimization. *IEEE Trans. on Systems, Man and Cybernetics C*, 28(2):263–271, 1998.