An Evidence-Theoretic $k$-NN Rule with Parameter Optimization

Lalla Meriem Zouhal and Thierry Deneux

Abstract—This paper presents a learning procedure for optimizing the parameters in the evidence-theoretic $k$-nearest neighbor rule, a pattern classification method based on the Dempster–Shafer theory of belief functions. In this approach, each neighbor of a pattern to be classified is considered as an item of evidence supporting certain hypotheses concerning the class membership of that pattern. Based on this evidence, basic belief masses are assigned to each subset of the set of classes. Such masses are obtained for each of the $k$-nearest neighbors of the pattern under consideration and aggregated using the Dempster’s rule of combination. In many situations, this method was found experimentally to yield lower error rates than other methods using the same information. However, the problem of tuning the parameters of the classification rule was so far unresolved. In this paper, we determine optimal or near-optimal parameter values from the data by minimizing an error function. This refinement of the original method is shown experimentally to result in substantial improvement of classification accuracy.

Index Terms—Evidence theory, learning systems, parameter estimation, pattern classification, uncertainty.

I. INTRODUCTION

In the classical approach to statistical pattern recognition, the entities to be classified are assumed to be selected by some form of random experiment. The feature vector describing each entity is then a random vector with a well-defined—though unknown—probability-density function—depending on the pattern category. Based on these densities and on the prior probability of each class, posterior probabilities can be defined and the optimal Bayes decision rule can then be theoretically used for classifying an arbitrary pattern with minimal expected risk. Since the class-conditional densities and prior probabilities are usually unknown, they need to be estimated from the data. A lot of methods have been proposed for building consistent estimators of the posterior probabilities under various assumptions. However, for finite sample size, the resulting estimates generally do not provide a faithful representation of the fundamental uncertainty pertaining to the class of a pattern to be classified. For example, if only a relatively small number of training vectors is available and a new pattern is encountered that is very dissimilar to all of the previous patterns, the uncertainty is quite high, and this situation of near-ignorance is not reflected by the outputs of a conventional parametric or nonparametric statistical classifier, whose principle fundamentally relies on asymptotic assumptions. This problem is particularly acute in situations in which decisions need to be made based on weak information, such as commonly encountered in system diagnosis applications.

As an attempt to provide an answer to the above problem, it was recently suggested that the pattern classification problem be reformulated by considering the following question. Given a training set of finite size containing feature vectors with known (or partially known) classification and a suitable distance measure, how is the uncertainty pertaining to the class of a new pattern characterized? In a recent paper [2], an answer to this question was proposed based on the Dempster–Shafer theory of evidence [12]. This approach considers each neighbor of a pattern to be classified as an item of evidence supporting certain hypotheses concerning the class membership of that pattern. Based on this evidence, basic belief masses are assigned to each subset of the set of classes. Such masses are obtained for each of the $k$-nearest neighbors of the pattern under consideration and aggregated using the Dempster’s rule of combination. Given a finite set of actions and losses associated to each action and each class, decisions can then be made by using some generalization of the Bayes decision theory.

In many situations, this method was found experimentally to yield lower error rates than other methods based on the same information. However, the problem of optimizing the parameters involved in the classification rule was so far unresolved. In this paper, we determine optimal or near-optimal parameter values from the data by minimizing a certain error function. This refinement of the original method is shown experimentally to result in substantial improvement of classification accuracy.

The paper is organized as follows. First, the evidence-theoretic $k$-NN rule is recalled in Section II. The basic concepts of the Dempster–Shafer theory are assumed to be known; refer to [12] and [14] for detailed presentations and to [2] for a short introduction. Section III describes the learning procedure as well as an approximation to it, allowing it to near-optimize the error function very efficiently. Simulation results are then presented in Section IV, and Section V concludes the paper.

II. EVIDENCE-THEORETIC $k$-NN RULE

We consider the problem of classifying entities into $M$ categories or classes. The set of classes is denoted by $\Omega = \{1, 2, \ldots, M\}$,
The available information is assumed to consist of a training set \( T = \{ (\mathbf{x}^{(1)}, \omega^{(1)}), \ldots, (\mathbf{x}^{(N)}, \omega^{(N)}) \} \) of \( N \) \( n \)-dimensional patterns \( \mathbf{x}^{(i)} \), \( i = 1, \ldots, N \) and their corresponding class labels \( \omega^{(i)} \), \( i = 1, \ldots, N \), taking values in \( \Omega \). The similarity between patterns is assumed to be correctly measured by a certain distance function \( d(\cdot, \cdot) \).

Let \( \mathbf{x} \) be a new vector to be classified on the basis of the information contained in \( T \). Each pair \( (\mathbf{x}, \omega) \) constitutes a distinct item of evidence regarding the class membership of \( \mathbf{x} \). If \( \mathbf{x} \) is “close” to \( \mathbf{x}^{(i)} \), according to the relevant metric \( d \), we will be inclined to believe that both vectors belong to the same class. On the contrary, if \( \mathbf{x} \) is very far, the consideration of \( \mathbf{x}^{(i)} \) will leave us in a situation of almost complete ignorance concerning the class of \( \mathbf{x} \). Consequently, this item of evidence may be postulated to induce a basic belief assignment (BBA) \( m(\{ \omega \}) \) over \( \Omega \) defined by

\[
\begin{align*}
    m(\{ \omega \}) & = \alpha \phi_q(d) \\
    m(\Omega \setminus \{ \omega \}) & = 1 - \alpha \phi_q(d) \\
    m(\emptyset) & = 0, \quad \forall A \in 2^\Omega \setminus \{ \Omega, \{ \omega \} \}
\end{align*}
\]

where \( d(\cdot, \cdot) = d(\mathbf{x}, \mathbf{x}^{(i)}) \), \( \omega \) is the class of \( \mathbf{x}^{(i)} \), \( \alpha \) is a parameter such that \( 0 < \alpha < 1 \), and \( \phi_q \) is a decreasing function verifying \( \phi_q(0) = 1 \) and \( \lim_{d \to \infty} \phi_q(d) = 0 \). Note that \( m(\mathbf{x}) \) reduces to the vacuous belief function \( m(\Omega | \mathbf{x}) = 1 \) when the distance between \( \mathbf{x} \) and \( \mathbf{x}^{(i)} \) tends to infinity, reflecting a state of total ignorance. When \( d \) denotes the Euclidean distance, a rational choice for \( \phi_q \) shown in [4], is

\[
\phi_q(d) = \exp(-\gamma q d^2)
\]

\( \gamma_q \) being a positive parameter associated to class \( \omega_q \).

As a result of considering each training pattern in turn, we obtain \( N \) BBA’s that can be combined using the Dempster’s rule of combination to form a resulting BBA \( m \), synthesizing our final belief regarding the class of \( \mathbf{x} \)

\[
m = m(\{ \mathbf{x}^{(1)} \}) \oplus \cdots \oplus m(\{ \mathbf{x}^{(N)} \}).
\]

Since those training patterns situated far from \( \mathbf{x} \) actually provide very little information, it is sufficient to consider the \( k \)-nearest neighbors of \( \mathbf{x} \) in this sum. An alternative definition of \( m \) is therefore

\[
m = m(\{ \mathbf{x}^{(i_1)} \}) \oplus \cdots \oplus m(\{ \mathbf{x}^{(i_k)} \})
\]

where \( I_k = \{ i_1, \ldots, i_k \} \) contains the indexes of the \( k \)-nearest neighbors of \( \mathbf{x} \) in \( T \).

Adopting this latter definition, \( m \) can be shown [2] to have the following expression:

\[
m(\{ \omega_q \}) = \frac{1}{K} \left( \prod_{i \in I_k} \left( 1 - \prod_{q \in \Omega} \left( 1 - \alpha \phi_q(d) \right) \right) \prod_{q \notin I_k} \left( 1 - \alpha \phi_q(d) \right) \right),
\]

\[
m(\Omega) = \frac{1}{K} \prod_{i=1}^{M} \left( 1 - \alpha \phi_q(d) \right)
\]

where \( I_k \) is the subset of \( I_k \), corresponding to those neighbors of \( \mathbf{x} \) belonging to class \( \omega_q \), and \( K \) is a normalizing factor. Hence, the focal elements of \( m \) are singletons and the whole frame \( \Omega \). Consequently, the credibility and the plausibility of each class \( \omega_q \) are, respectively, equal to

\[
\text{bel}(\{ \omega_q \}) = m(\{ \omega_q \})
\]

\[
\text{pl}(\{ \omega_q \}) = m(\{ \omega_q \}) + m(\Omega).
\]

The pignistic probability distribution, as defined by Smets [13], is given by

\[
\text{BetP}(\{ \omega_q \}) = \sum_{\{A \mid \omega_q \subseteq A \}} \frac{m(A)}{|A|}
\]

\[
= \frac{m(\{ \omega_q \}) + m(\Omega)}{M}
\]

for \( q = 1, \ldots, M \). Let us now assume that, based on this evidential corpus, a decision has to be made regarding the assignment of \( \mathbf{x} \) to a class, and let us denote by \( \alpha_q \) the action of assigning \( \mathbf{x} \) to class \( \omega_q \). Let us further assume that the loss incurred in case of wrong classification is equal to one, while the loss corresponding to correct classification is equal to zero.

Then, the lower and the upper expected losses [3] associated to action \( \alpha_q \) are, respectively, equal to

\[
R_u(\alpha_q | \mathbf{x}) = 1 - \text{pl}(\{ \omega_q \})
\]

\[
R_t(\alpha_q | \mathbf{x}) = 1 - \text{bel}(\{ \omega_q \}).
\]

The expected loss, relative to the pignistic distribution, is

\[
R_{\text{BetP}}(\alpha_q | \mathbf{x}) = 1 - \text{BetP}(\{ \omega_q \}).
\]

Given the particular form of \( m \), the three strategies minimizing \( R_u \), \( R_t \), and \( R_{\text{BetP}} \) lead to the same decision in that case: the pattern is assigned to the class with maximum belief assignment. Other decision strategies, including the possibility of pattern rejection as well as the existence of unknown classes, are studied in [3] and [5].

III. PARAMETER OPTIMIZATION

A. The Approach

In the above description of the evidence-theoretic \( k \)-NN rule, we left open the question of the choice of parameters \( \alpha \) and \( \gamma = (\gamma_1, \ldots, \gamma_M) \) appearing in (1) and (4). Whereas the value of \( \alpha \) proves in practice not to be too critical, the
tuning of the other parameters was found experimentally to have significant influence on classification accuracy. In [2], it was proposed to set $\alpha = 0, \gamma_q$ to the inverse of the mean distance between training patterns belonging to class $\omega_q$. Although this heuristic yields good results on average, the efficiency of the classification procedure can be improved if these parameters are determined as the values optimizing a performance criterion. Such a criterion can be defined as follows.

Let us consider a training pattern $\mathbf{x}^{(t)}$ belonging to class $\omega_q$. The class membership of $\mathbf{x}$ can be encoded as a vector $\mathbf{t}^{(t)} = (t_1^{(t)}, \cdots, t_M^{(t)})^t$ of $M$-binary indicator variables $t_j^{(t)}$ defined by $t_j^{(t)} = 1$, if $j = q$ and $t_j^{(t)} = 0$, otherwise. By considering the $k$-nearest neighbors of $\mathbf{x}^{(t)}$ in the training set, we obtain a “leave-one-out” BBA $m^{(t)}$, characterizing our belief concerning the class of $\mathbf{x}^{(t)}$ if this pattern was to be classified using other training patterns. Based on $m^{(t)}$, an output vector $\mathbf{P}^{(t)} = \left( \text{BetP}^{m^{(t)}}(\omega_1^{(t)}), \cdots, \text{BetP}^{m^{(t)}}(\omega_M^{(t)}) \right)^t$ of pignistic probabilities can be computed, $\text{BetP}^{m^{(t)}}$ being the pignistic probability distribution associated to $m^{(t)}$. Ideally, vector $\mathbf{P}^{(t)}$ should as “close” as possible to vector $\mathbf{t}^{(t)}$, closeness being defined, for example, according to the squared error $E(\mathbf{t}^{(t)})$

$$E(\mathbf{t}^{(t)}) = (\mathbf{P}^{(t)} - \mathbf{t}^{(t)})^t(\mathbf{P}^{(t)} - \mathbf{t}^{(t)}) = \sum_{q=1}^M (P_q^{(t)} - t_q^{(t)})^2.$$  

The mean squared error over the whole training set $T$ of size $N$ is finally equal to

$$E = \frac{1}{N} \sum_{t=1}^N E(\mathbf{t}^{(t)}).$$

Function $E$ can be used as a cost function for tuning the parameter vector $\gamma$. The analytical expression for the gradient of $E(\mathbf{t}^{(t)})$, with respect to $\gamma$, can be calculated, allowing the parameters $\gamma_q$ to be determined iteratively by a gradient search procedure (see Appendix I). Alternatively, the minimum of function $E$ can be approximated in one step for large $N$ by using the approach described in the sequel.

B. One-Step Procedure

For an arbitrary training pattern $\mathbf{x}^{(t)}$ and fixed parameters, vector $\mathbf{P}^{(t)}$ can be regarded as a function of the following two vectors:

1) vector $\mathbf{d} = (d_1^{(t)}, \cdots, d_N^{(t)})^t$ of squared distances between $\mathbf{x}^{(t)}$ and its $k$-nearest neighbors; and

2) vector $\mathbf{w}$ containing the class labels of these neighbors. For small $k$ and large $N$, $\mathbf{d}$ can be assumed to be close to zero, allowing each component $P_q^{(t)}$ to be approximated by the Taylor series expansion around zero, up to the first order

$$P_q^{(t)}(\mathbf{d}) \approx P_q^{(t)}(0) + \nabla \mathbf{d} P_q^{(t)}(0) \mathbf{d} = 0 \mathbf{d}^2.$$  

The first term in this expression can be readily obtained from (7) and (8) by setting $d_j^{(t)}$ to zero for all $j$, which leads to

$$P_q^{(t)}(0) = \frac{(1-\alpha)^k}{K} \left[ (1-\alpha)^{-k_0} - 1 + \frac{1}{M} \right],$$

where $k_q$ is the number of neighbors of $\mathbf{x}^{(t)}$ in class $\omega_q$ and

$$K = \frac{(1-\alpha)^k}{\sum_{q=1}^M (1-\alpha)^{-k_0} - k + 1}.$$

The computation of the first-order term

$$\nabla \mathbf{d} P_q^{(t)}(0) \mathbf{d} = \sum_{j=1}^n \frac{\partial P_q^{(t)}(0)}{\partial d_j^{(t)}} d_j^{(t)} d_j^{(t)}$$

is more involved (see Appendix II). This term can be shown to be of the form $A^{(t)} \gamma$, where $A^{(t)}$ is a square matrix of size $M$. As a consequence, both $E(\mathbf{t}^{(t)})$ and $E$ can be approximated by quadratic forms of $\gamma$, which allows the minimum to be approached directly by solving a system of linear equations.

Fig. 1 and 2 show the quality of this approximation in the case of two Gaussian classes with mean vectors $\mu_1 = (2, 0)^t$ and $\mu_2 = (-2, 0)^t$, respectively, and covariance matrices $\Sigma_1 = \Sigma_2 = \mathbf{I}$. The data set contained 500 samples of each class. Displayed are the mean squared error as a function of $\gamma$ (Fig. 1) and its quadratic approximation (Fig. 2), for $k = 5$. The minima of the two functions differ by less than 0.0007%, which proves the relevance of the approximation in that case. Note that the quality of the approximation depends on both $k$ and $N$.

IV. NUMERICAL EXPERIMENTS

The performances of the above methods were compared to those of the voting $k$-NN rule with randomly resolved ties, the distance-weighted $k$-NN rule [6], the fuzzy $k$-NN rule proposed by Keller [9], and the evidence-theoretic rule.
without parameter optimization [2]. Experiments were carried out on a set of standard artificial and real-world benchmark classification tasks. The main characteristics of the used data sets are summarized in Table I.

Data sets $B_1$ and $B_2$ were generated by using a method proposed in [7]. The data consist in three Gaussian classes in ten dimensions with diagonal covariance matrices $D_1$, $D_2$, and $D_3$. The $i$th diagonal element $D_{ii}$ of $D_i$ is defined as a function of two parameters $a$ and $b$

$$D_{2i}(a, b) = a + (b - a) \frac{i - 1}{n - 1}$$

$$D_{2i}(a, b) = a + (b - a) \frac{n - i}{n - 1}$$

$$D_{3i}(a, b) = a + (b - a) \min \left( i, \frac{n - i}{n/2} \right)$$

where $n = 10$ is the input dimension. The mean vectors and covariance matrices (with $n = 10$) for the three classes were

$\mu_1 = (0, \ldots, 0)$, $\mu_2 = (1, \ldots, 1)$, $\mu_3 = (1, -1, \ldots, 1, -1)$, $D_1(1, 10)$, $D_2(10, 1)$, $D_3(1, 10)$.

The ionosphere data set (Ion) 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 kW [11].

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 vehicle data set (Veh) was collected from silhouettes by the Hierarchical Image Processing System (HIPS) extension BINATTS. Four model vehicles were used for the experiment: bus, Chevrolet van, Saab 9000, and Opel Manta 400. The data were used to distinguish three-dimensional (3-D) objects within a two-dimensional (2-D) silhouette of the objects [11].

The sonar data were used by Gorman and Sejnowski in a study of the classification of sonar signals using a neural network [8]. The task is to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock.

Test error rates are represented as a function of $k$ in Figs. 3–12. The results for synthetic data are averages over six independent training sets. Test error rates obtained for the
values of $k$ yielding the best classification of training vectors are presented in Table II.

As can be seen from these results, the evidence-theoretic rule, with optimized $\gamma$, presented in this paper, always performed as well or better than the four other rules tested and significantly improved at the 90% confidence level over the evidence-theoretic rule with fixed $\gamma$ on data sets $B_1$ and $B_2$ when considering the best results obtained for $1 \leq k \leq 40$. However, the most distinctive feature of this rule seems to be its robustness, with respect to the number $k$ of neighbors taken in consideration (Figs. 3–12). By optimizing $\gamma$, the method learns to discard those neighbors whose distance to the pattern under consideration is too high. Practically, this property is of great importance since it relieves the designer of the system from the burden of searching for the optimal value of $k$. When the number of training patterns is large, the amount of computation may be further reduced by adopting the approximate one-step procedure for optimizing $\gamma$, which gives reasonably good results for small $k$ (Figs. 3, 5, 7, 9, and 11). However, the use of the exact procedure should be preferred for small- and medium-sized training sets.

### V. CONCLUDING REMARKS

A technique for optimizing the parameters in the evidence-theoretic $k$-NN rule has been presented. The classification rule obtained by this method has proved superior to the voting, distance-weighted, and fuzzy rules on a number of benchmark problems. A remarkable property achieved with this approach is the relative insensitivity of the results to the choice of $k$.

The method can be generalized in several ways. First of all, we can assume a more general metric than the Euclidean one considered so far and apply the principles described in this paper to search for the optimal metric [15]. For instance, let $\Sigma_q$ be a positive-definite diagonal matrix with diagonal elements...
The parameters $\gamma_{q,j}$ for $1 \leq q \leq M$ and $1 \leq j \leq n$ can then be optimized using exactly the same approach as described in this paper, which may in some cases result in further improvement of classification results. A more general form could even be assumed for $\sum q$, with the risk of a dramatic increase in the number of parameters for large input dimensions.

More fundamentally, the method can also be extended to handle the more general situation in which the class membership of training patterns is itself affected by uncertainty. For example, let us assume that the class of each training pattern $\mathbf{x}^{(i)}$ is only known to lie in a subset $A^{(i)}$ of $\Omega$ (such a situation may typically arise, e.g., in medical diagnosis problems, in which some records in a database are related to patients for which only a partial or uncertain diagnosis is available). A natural extension of (1)–(3) is then

$$m(A^{(i)}|\mathbf{x}^{(i)}) = \alpha \phi(d^{(i)})$$

$$m(\Omega |\mathbf{x}^{(i)}) = 1 - \alpha \phi(d^{(i)})$$

$$m(B |\mathbf{x}^{(i)}) = 0, \quad \forall B \in 2^\Omega \setminus \{\omega_i | A^{(i)}\}$$

with $\phi(d) = \exp(-\gamma d^2)$ and $\gamma$ being a positive parameter (note that we cannot define a separate parameter for each class in this case since the class of $\mathbf{x}^{(i)}$ is only partially known). The BBA’s defined in that way correspond to simple belief functions and can be combined in linear time, with respect to the number of classes. For optimizing $\gamma$, the error criterion defined in (15) has to be generalized in some way. With the same notations as in Section III-A, a possible expression for...
Fig. 11. Test error rates on sonar data, as a function of \( k \) for the ETF (x) and ETO \( k \)-NN rules: (\( \cdot \)) gradient method and (\( \cdot - \cdot \)) linearization method.

Fig. 12. Test error rates on sonar data, as a function of \( k \) for the voting (\( \cdot \)), ETO (\( \cdot \cdot \)), fuzzy (\( \cdot - \cdot \)), and distance-weighted (\( \cdot - \cdot \)) \( k \)-NN rules.

the error concerning pattern \( \mathbf{x}^{(t)} \) is

\[
E(\mathbf{x}^{(t)}) = \{\text{BetP}^{(t)}(A^{(t)}) - 1\}^2
\]

which reflects the fact that the pignistic probability of \( \mathbf{x}^{(t)} \) belonging to \( A^{(t)} \), given the other training patterns, should be as high as possible. The value of \( \gamma \) minimizing the mean error may then be determined by using an iterative search procedure. Experiments with this approach are currently under way and will be reported in future publications.

**APPENDIX I**

**COMPUTATION OF THE DERIVATIVES OF \( E \), w.r.t., \( \gamma_q \)**

Let \( \mathbf{x}^{(t)} \) be a training pattern and \( m^{(t)} \) be the BBA obtained by classifying \( \mathbf{x}^{(t)} \), using its \( k \)-nearest neighbors in the training set. Function \( m^{(t)} \) is computed according to (7) and (8)

\[
m^{(t)}(\{\omega_q\}) = \frac{1}{K} \left( 1 - \prod_{i \in I^{(t)}_{k,q}} \{1 - \alpha \phi_q(d^{(t,i)})\} \right) \cdot \prod_{i \in I^{(t)}_{k,q}} \prod_{j \in I^{(t)}_{k,r}} \{1 - \alpha \phi_r(d^{(t,j)})\},
\]

\[
\forall q \in \{1, \ldots, M\}
\]

\[
m^{(t)}(\Omega) = \frac{1}{K} \prod_{i = 1}^{M} \prod_{j \in I^{(t)}_{k,i}} \{1 - \alpha \phi_r(d^{(t,j)})\}
\]

where \( I^{(t)}_{k,i} \) denotes the set of indexes of the \( k \)-nearest neighbors of pattern \( \mathbf{x}^{(t)} \) in class \( \omega_q \), \( d^{(t,i)} \) is the distance between \( \mathbf{x}^{(t)} \) and \( \mathbf{x}^{(t,i)} \), and \( K \) is a normalizing factor. In the following, we shall assume that:

\[
\phi_q(d^{(t,i)}) = \exp(-\gamma d^{(t,i)^2})
\]

which will simply be denoted by \( \phi_q(d^{(t,i)}) \). To simplify the calculations, we further introduce the unnormalized orthogonal sum [13] \( m^{(t)'} \), defined as \( m^{(t)'}(A) = Km^{(t)}(A) \) for all \( A \subseteq \Omega \). We also denote as \( \overline{m^{(t)\prime}} \) the unnormalized orthogonal sum of the \( m(\{\omega_j\}) \) for all \( j \in I^{(t)}_k \), \( j \neq i \), that is, \( m^{(t)'} = \overline{m^{(t)\prime}} \wedge m(\{\omega_i\}) \), where \( \wedge \) denotes the unnormalized orthogonal sum operator. More precisely, we have

\[
m^{(t)'}(\{\omega_q\}) = m^{(t)}(\{\omega_q\}|\mathbf{x}^{(t)})(\overline{m^{(t)\prime}}(\{\omega_q\}) + \overline{m^{(t)\prime}}(\Omega)),
\]

\[
\forall q \in \{1, \ldots, M\}
\]

\[
m^{(t)'}(\Omega) = \overline{m^{(t)\prime}}(\Omega)m(\Omega)|\mathbf{x}^{(t)}).
\]

The error for pattern \( \mathbf{x}^{(t)} \) is

\[
E(\mathbf{x}^{(t)}) = \sum_{q = 1}^{M} (P_q^{(t)} - t_q^{(t)})^2
\]

where \( t_q^{(t)} \) is the class membership vector for pattern \( \mathbf{x}^{(t)} \) and \( P_q^{(t)} \) is the pignistic probability of class \( \omega_q \) computed from \( m^{(t)} \) as \( P_q^{(t)} = m^{(t)}(\{\omega_q\})/(m^{(t)}(\Omega))/M \).

The derivative of \( E(\mathbf{x}^{(t)}) \), with respect to each parameter \( \gamma_q \), can be computed as

\[
\frac{\partial E(\mathbf{x}^{(t)})}{\partial \gamma_q} = \sum_{i \in I^{(t)}_{k,q}} \frac{\partial E(\mathbf{x}^{(t)})}{\partial \phi_q^{(t)}} \frac{\partial \phi_q^{(t)}}{\partial \gamma_q}
\]

with

\[
\frac{\partial E(\mathbf{x}^{(t)})}{\partial \phi_q^{(t)}} = \sum_{r = 1}^{M} \frac{\partial E(\mathbf{x}^{(t)})}{\partial P_r}\frac{\partial P_r}{\partial \phi_q^{(t)}}
\]

\[
= \sum_{r = 1}^{M} 2(P_q^{(t)} - t_q^{(t)}) \frac{\partial m^{(t)}(\{\omega_r\})}{\partial \phi_q^{(t)}} + \frac{1}{M} \frac{\partial m^{(t)}(\Omega)}{\partial \phi_q^{(t)}}
\]
and
\[ \frac{\partial \phi^{(t)}(t)}{\partial \eta} = -\delta^{(t)} \cdot \phi_q^{(t)}(t). \]  

The derivatives in (29) can be computed as

\[ \frac{\partial m^{(t)}(\omega_k)}{\partial \eta_k} = \frac{\partial}{\partial \eta_k} \left[ \frac{m^{(t)}(\omega_k)}{K} \right] = \frac{1}{K^2} \left[ K m^{(t)}(\omega_k) \frac{\partial K}{\partial \eta_k} \right] - m^{(t)}(\omega_k) \frac{\partial K}{\partial \eta_k} \]  

\[ \frac{\partial m^{(t)}(\Omega)}{\partial \phi_q} = \frac{1}{K^2} \left[ K m^{(t)}(\Omega) \frac{\partial K}{\partial \phi_q} \right] - m^{(t)}(\Omega) \frac{\partial K}{\partial \phi_q} \]  

where \( \delta \) is the Kronecker symbol and

\[ \frac{\partial m^{(t)}(\Omega)}{\partial \phi_q} = -m^{(t)}(\Omega) \]  

which completes the calculation of the gradient of \( m(\omega_k) \), w.r.t. \( \eta_k \).

To account for the constraint \( m' \), we introduce new parameters \( \eta_q \), such that

\[ m'(\omega_k) = \frac{1}{K^2} \left[ K m'(\omega_k) \frac{\partial K}{\partial \eta_q} \right] - m'(\omega_k) \frac{\partial K}{\partial \eta_q} \]  

\[ \frac{\partial K}{\partial \phi_q} = \sum_{r=1}^{M} \frac{\partial m^{(t)}(\omega_k)}{\partial \phi_q} + \frac{\partial m^{(t)}(\Omega)}{\partial \phi_q}. \]  

Finally

\[ \frac{\partial m^{(t)}(\omega_k)}{\partial \phi_q} = (m'(\omega_k) + m'(\Omega)) \delta_{\phi_q} - \alpha m(\omega_k) \]  

where \( \delta \) is the Kronecker symbol and

\[ \frac{\partial m^{(t)}(\Omega)}{\partial \phi_q} = m^{(t)}(\Omega) \]  

which completes the calculation of the gradient of \( E(x^{(t)}) \), w.r.t. \( \gamma_q \).

To account for the constraint \( \gamma_q \geq 0 \), we introduce new parameters \( \mu_q \) \((q = 1, \ldots, M)\), such that

\[ \gamma_q = (\mu_q)^2 \]  

and we compute \( \{\partial E(x^{(t)})\}/\partial \mu_q \) as

\[ \frac{\partial E(x^{(t)})}{\partial \mu_q} = \frac{\partial E(x^{(t)})}{\partial \gamma_q} \frac{\partial \gamma_q}{\partial \mu_q} = 2 \mu_q \frac{\partial E(x^{(t)})}{\partial \gamma_q}. \]  

**APPENDIX II**

**LINEARIZATION**

We consider the expansion around \( d^2 = 0 \) of \( P_q^{(t)} \) by a Taylor series up to the first order

\[ P_q^{(t)}(d^2) = P_q^{(t)}(0) + (\nabla_d P_q^{(t)}(0))^t d^2 \]  

where \( P_q^{(t)}(0) \) is given by (18). In the following, we shall compute the first-order term in the above equation and deduce from that result a method for determining an approximation to the optimal parameter vector. To simplify the notation, the superscript \( (t) \) will be omitted from the following calculations.

As a result of the definition of the pignistic probability, we have

\[ \frac{\partial P_q}{\partial \phi_q} = \frac{\partial m(\omega_q)}{\partial \phi_q} + \frac{1}{M} \frac{\partial m(\Omega)}{\partial \phi_q}. \]  

The derivatives of \( m(\omega_q) \) and \( m(\Omega) \) can be more conveniently expressed as a function of the unnormalized BBA \( m' \)

\[ \frac{\partial m(\omega_q)}{\partial \phi_q} = \frac{1}{K^2} \left[ K m(\omega_q) \frac{\partial K}{\partial \phi_q} - m'(\omega_q) \frac{\partial K}{\partial \phi_q} \right] \]  

\[ \frac{\partial m(\Omega)}{\partial \phi_q} = \frac{1}{K^2} \left[ K m(\Omega) \frac{\partial K}{\partial \phi_q} - m'(\Omega) \frac{\partial K}{\partial \phi_q} \right] \]  

To compute \( \{\partial m'(\omega_q)/\partial \phi_q\} \) and \( \{\partial m'(\Omega)/\partial \phi_q\} \), we need to distinguish two cases.

**Case 1:** \( i \in I_{k,q} \). We then have

\[ \frac{\partial m'(\omega_q)}{\partial \phi_q} = -\alpha \gamma_q \exp(-\gamma_q d^2) \prod_{j \in I_{k,q}, j \neq i} \{1 - \alpha \exp(-\gamma_q d_{ij}^2)\} \prod_{j \in I_{k,q}, j \neq i} \{1 - \alpha \exp(-\gamma_q d_{ij}^2)\} \]  

\[ \frac{\partial m'(\Omega)}{\partial \phi_q} = \alpha \gamma_q \exp(-\gamma_q d^2) \prod_{r=1}^{M} \prod_{j \in I_{k,q}, j \neq i} \{1 - \alpha \exp(-\gamma_q d_{ij}^2)\}. \]  

Setting all distances to zero in the above equations, we have

\[ \frac{\partial m'(\omega_q)}{\partial \phi_q} \bigg|_{d^2=0} = -\alpha \gamma_q (1 - \alpha)^{k_q-1} \]  

\[ \frac{\partial m'(\Omega)}{\partial \phi_q} \bigg|_{d^2=0} = \alpha \gamma_q (1 - \alpha)^{k_q-1}. \]  

**Case 2:** \( i \in I_{k,q}, l \neq q \). We have

\[ \frac{\partial m'(\omega_q)}{\partial \phi_q} = \alpha q \exp(-\gamma_q d^2) \]  

\[ \cdot \left( \prod_{j \in I_{k,q}, j \neq q} \{1 - \alpha \exp(-\gamma_q d_{ij}^2)\} \right) \times \prod_{r \neq q} \prod_{j \in I_{k,q}, j \neq i} \{1 - \alpha \exp(-\gamma_q d_{ij}^2)\} \]  

\[ \frac{\partial m'(\Omega)}{\partial \phi_q} = \alpha q \exp(-\gamma_q d^2) \prod_{r=1}^{M} \prod_{i \neq q} \{1 - \alpha \exp(-\gamma_q d_{ij}^2)\}. \]  

Setting the distances to zero in the above equations, we have

\[ \frac{\partial m'(\omega_q)}{\partial \phi_q} \bigg|_{d^2=0} = \alpha q (1 - \alpha)^{k_q-1} \]  

\[ \frac{\partial m'(\Omega)}{\partial \phi_q} \bigg|_{d^2=0} = \alpha q (1 - \alpha)^{k_q-1}. \]  

where \( k_q = |I_{k,q}| \).
The derivatives of $K$ are simply obtained as follows:

$$\frac{\partial K}{\partial \delta^2} = \sum_{i=1}^{M} \frac{\partial m'(\omega_0^0)}{\partial \delta^2} + \frac{\partial m'(\Omega)}{\partial \delta^2}. \tag{51}$$

Hence

$$\frac{\partial K}{\partial \delta^2} = \alpha \gamma \sum_{i=1, r \neq q}^{M} (1 - (1 - \alpha)^k_h)(1 - \alpha)^{k-h-1}. \tag{52}$$

It follows from the preceding calculations that, for $i \in I_{k_h}$, the derivatives of $m'(\omega_q^0)$, $m'(\Omega)$, and $K$ for $d^2 = 0$ are proportional to $\gamma_r$. Since $m'(\omega_q^0)$, $m'(\Omega)$, and $K$ do not themselves depend on $\gamma$ for $d^2 = 0$, the derivative of $P_q$ is also proportional to $\gamma$. Hence, we have

$$\sum_{i \in I_{k_h}} \frac{\partial P_q}{\partial \delta^2} \bigg|_{d^2 = 0} = A_{q, r} \gamma_r, \tag{53}$$

for all $r \in \{1, \ldots, M\}$, $A_{q, r}$ being some constant not depending on $\gamma$. Consequently, we can write

$$P_q = P_q(0) + \sum_{r=1}^{M} A_{q, r} \gamma_r \tag{54}$$

and expressing this result in the matrix form

$$P \approx P(0) + A \gamma \tag{55}$$

where $A = (A_{i, j})$ is a square matrix of size $M$.

The above calculations have been performed for an arbitrary training pattern $x$. Reintroducing the pattern index $\ell$, we have

$$P(\ell) \approx P(\ell)(0) + A(\ell) \gamma. \tag{56}$$

Introducing these terms into the mean squared error, we have

$$E = \frac{1}{N} \sum_{\ell=1}^{N} (P(\ell) - t(\ell))^T (P(\ell) - t(\ell)) = \frac{1}{N} \sum_{\ell=1}^{N} (P(\ell)(0) - t(\ell))^T + A(\ell) \gamma)^T (P(\ell)(0) - t(\ell))^T + A(\ell) \gamma) \tag{57}$$

$$= \frac{1}{N} \sum_{\ell=1}^{N} (P(\ell)(0) - t(\ell))^T + 2\gamma^T A(\ell)^T A(\ell) \gamma. \tag{58}$$

The gradient $E$, with respect to $\gamma$, is therefore given by

$$\nabla \gamma E = \frac{1}{N} \left\{ \sum_{\ell=1}^{N} A(\ell)^T (P(\ell)(0) - t(\ell)) + \sum_{\ell=1}^{N} A(\ell)^T A(\ell) \right\}. \tag{59}$$

Minimizing $E$ under the constraint $\gamma \geq 0$ is a nonnegative least-squares problem that may be solved efficiently using, for instance, the algorithm described in [10, p. 161].

**REFERENCES**


Lalla Meriem Zouhal received the M.S. degree in electronics from the Faculté des Sciences Hassan II, Casablanca, Morocco, in 1990 and the DEA (Diplôme d’Etudes Approfondies) degree in system control and the Ph.D. degree from the Université de Technologie de Compiègne, Compiègne, France, in 1992 and 1997, respectively.

Her research interests concern pattern classification, Dempster–Shafer theory, and fuzzy logic.

Thierry Denœux received an engineering degree and the Ph.D. degree from the Ecole Nationale des Ponts et Chaussées, Paris, France, in 1985 and 1989, respectively.

He was with the Lyonnaise des Eaux water company from 1989 to 1992, where he was in charge of research projects concerning the application of neural networks to forecasting and diagnosis. He is now with the Université de Technologie de Compiègne, Compiègne, France, as an Assistant Professor. His research interests include artificial neural networks, statistical pattern recognition, uncertainty modeling, and data fusion.

Dr. Denœux received the “Habilitation à diriger des Recherches” from the Institut National Polytechnique de Lorraine, France, in 1996.