This is the published version (version of record) of:


Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30018288

Copyright : ©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Using Choquet integrals for kNN approximation and classification

Gleb Beliakov and Simon James

Abstract—k-nearest neighbors (kNN) is a popular method for function approximation and classification. One drawback of this method is that the nearest neighbors can be all located on one side of the point in question \( x \). An alternative natural neighbors method is expensive for more than three variables. In this paper we propose the use of the discrete Choquet integral for combining the values of the nearest neighbors so that redundant information is canceled out. We design a fuzzy measure based on location of the nearest neighbors, which favors neighbors located all around \( x \).

I. INTRODUCTION

Function approximation is an important task in data analysis and supervised classification. A popular method for both approximation of real-valued functions and classification is the \( k \)-nearest neighbors method (kNN). It consists in calculating the distances from a point in question to all the data in the training set, and then aggregating the values of the \( k \) nearest data. The aggregation is typically based on using a (weighted) arithmetic mean or simply the count of how many of the \( k \) nearest neighbors belong to a given class. In the latter case the classification is performed by the “majority vote”.

One problem with this method is that the “votes” of the neighbors may be skewed, or correlated. Consider the data presented on Figs. 1, 2. All the nearest neighbors happen to be on one side of the point in question, whereas there are plenty of neighbors on the other side whose votes are not counted. This gave rise to the following observation: we want to include information provided by the neighbors which are close to the point in question but also distributed all around it. A method based on this principle is called the natural neighbor method [1]–[3].

However calculation of the natural neighbors, which is based on calculating the Voronoi diagrams, is rather complicated for more than two variables. To our knowledge, only implementations for the two- and three-variate cases are available [3]–[5]. The main reason is the rapidly increasing computational complexity (exponential in the number of variables), which makes the algorithms impractical.

In this paper we propose an alternative approach to canceling out contributions of the neighbors which are in some sense correlated, i.e., lie on the same side of the point in question \( x \). We propose to use an alternative aggregation method for combining information provided by the neighbors, which explicitly takes into account these dependencies. It is based on the discrete Choquet integral, which is a popular aggregation method precisely for such situations [6]–[9].

Choquet integrals are defined with respect to fuzzy measures, also called capacities. Discrete fuzzy measures are defined on a power set of \( N = \{1, \ldots, n\} \), \( 2^N \). Their values are interpreted as contributions to the value of the Choquet integral by not only the individual inputs but their groups (coalitions). Thus contribution of two (or more) inputs may be greater or smaller than the sum of individual contributions, depending on whether they have positive or negative “synergy”. The interactions within groups of inputs are conveniently represented by various interaction indices.

The next section gives the necessary background on aggregation functions, in particular Choquet integrals. In Section III we formulate the problem of determining the appropriate fuzzy measure based on the relative position of the abscissae of the training data and present a method of function approximation based on kNN and the Choquet integral. In Section IV we outline a classification method based on the same approach, but which is different to approximation due to the specifics of classification problems. Section V contains the conclusions and outlines future research.

II. PRELIMINARIES

A. k-nearest neighbors method

The method of \( k \)-nearest neighbors is a very popular tool for approximation of real-valued functions and classification, see e.g. [10]–[12]. It is based on calculating the distances between the reference data (it is often called training data, although no actual training in the kNN method takes place) and the point in question \( x \), at which either the value of a function or a class label is required.

Let us denote the training data by \( D = \{(x_i, y_i)\}, i = 1, \ldots, K, x_i \in \mathbb{R}^n, y_i \in \mathbb{R} \). For classification problems \( y_i \) will be in a discrete set of labels. Consider first the approximation problem. The data are assumed to be generated by an unknown function \( f \), such that \( f(x_i) = y_i + \varepsilon_i \), where \( \varepsilon_i \) denotes random errors (noisy data). For a point \( x \in \mathbb{R}^n \) we need to determine an approximate value \( y \approx f(x) \).

Calculate the pairwise distances \( d_i = ||x - x_i|| \) (in some norm), and sort \( D \) in the order of increasing \( d_i \). There are many works dedicated to the choice of such a norm, see, e.g. [10], [12], which is a very hard and context dependent problem. In this study we assume it is given. Then approximate \( f(x) \) by \( y = \sum_{i=1}^k w_i f(x_i) \), where the weights \( w_i \) are determined usually by some non-increasing function \( w_i = h(d_i) \), see [12]–[14]. In recent studies, Yager [15] proposes the use of Induced Ordered Weighted Averaging functions (Induced OWA) instead of the weighted mean to aggregate the values \( f(x_i) \) and an algorithm to learn \( w \) from
the data. The Choquet integral was used for the same purpose in [16].

![Image](118x410 to 247x516)

**Fig. 1.** An example (from the area of remote sensing, the data are taken by an airplane flying over a region in two directions) illustrating the inadequacy of the kNN method. The value at \( x \) is determined exclusively by the data represented by filled circles, i.e. is extrapolated and not interpolated.

![Image](252x307 to 256x307)

**Fig. 2.** An example illustrating a drawback of the kNN method for classification. The label at \( x \) is determined by the data represented by filled circles, whereas the classes are linearly separable (\( x \) may represent a missing datum on this figure).

For classification problems one proceeds in a similar way, but the label \( y(x) \) is determined by the “votes” of the \( k \) nearest neighbors \( x_i, i = 1, \ldots, k \). \( V o r_{vm} = \sum_{i \in m} w_i \delta_{im}, m = 1, \ldots, M \), where \( \delta_{im} = 1 \) iff \( i = m \) and \( M \) is the total number of classes. Typically the weights \( w_i = \frac{1}{k} \) or \( w_i = h(d_i) \) in the case of classification.

**B. Natural neighbors method**

Another popular method of multivariate approximation is the Natural Neighbor scheme by Sibson [1], [3], [17]. The idea of this method is to build an interpolant whose value at \( x \) would depend on a few data points close to \( x \) at the same time distributed all around \( x \). It favorably contrasts with the nearest neighbor methods in which only the distance from \( x \) matters.

In the natural neighbor scheme, the interpolant is a weighted average of the neighboring data values

\[ f(x) = \sum_{j=1}^{J} w_j(x) y_j, \]

where the weight \( w_j(x) \) is proportional to the volume of the part of Voronoi cell \( Vor(x_j) = \{ z : ||z - x_j|| \leq ||z - x_k||, k \neq j \}, \) which is cut by the Voronoi cell \( Vor(x) = \{ z : ||z - x|| \leq ||z - x_k|| \}, \) when \( x \) is added to the Voronoi diagram as one of the sites. Since Voronoi cell \( Vor(x) \) borders only a few neighboring Voronoi cells, only a few neighboring data points around \( x \) participate in calculation of \( f(x) \) (so called natural neighbors). More recently variations of Sibson’s method were developed, based on other rules for calculating weights \( w_j(x) \) [17], [18].

Sibson’s interpolant possesses many useful properties, but it is computationally expensive, as each \( x \) requires computation of a new Voronoi diagram having \( x \) as one of the sites. There are methods that allow an update of the Voronoi diagram when \( x \) is added to the list of sites in 2- and 3-variate cases, so that the whole Voronoi diagram needs not be built for every \( x \). Such methods are very competitive, but we are unaware of any extension for more than three variables.

**C. Aggregation functions**

Aggregation functions play an important role in several areas, including fuzzy logic, decision making, expert systems, risk analysis and image processing. Recent books [19]–[23] provide a comprehensive overview of aggregation functions and methods of their construction.

The purpose of aggregation functions is to combine several input values into a single output value, which in some sense represents all the inputs. Typically the inputs and outputs are real numbers from \([0, 1]\), although other choices are possible, e.g. discrete sets, intervals and linguistic labels. Notable examples are weighted means, medians, ordered weighted averaging (OWA) functions, discrete Choquet and Sugeno integrals, triangular norms and conorms, uninorms and nullnorms.

**Definition 1:** An aggregation function is a function of \( n > 1 \) arguments \( f : [0, 1]^n \rightarrow [0, 1] \), with the properties

(i) \( f(0, 0, \ldots, 0) = 0 \) and \( f(1, 1, \ldots, 1) = 1 \).

(ii) \( x \leq y \) implies \( f(x) \leq f(y) \) for all \( x, y \in [0, 1]^n \).

The vector inequality is understood componentwise. Aggregation functions may possess various properties, which often classify them into special classes. We are interested in averaging functions. An aggregation function \( f \) is called averaging if it is bounded (for all \( x \in [0, 1]^n \)) by

\[ \min_{i=1,...,n} x_i \leq f(x) \leq \max_{i=1,...,n} x_i = \max(x). \]

This condition is equivalent to idempotency: \( f(t, t, \ldots, t) = t \) for and any \( t \in [0, 1] \).

Weighted arithmetic means are the most common averaging aggregation functions. Discrete Choquet integrals generalize both the weighted arithmetic means and OWA functions. These functions are defined with respect to a fuzzy measure, and can take into account not only the relative weightings of the individual inputs, but also their groups (coalitions). A discrete fuzzy measure allows one to assign relative importances to all possible groups of criteria, and thus

implies that adding new elements to a coalition does not
whichever is more convenient. The conditions of mono-
by using its inverse, called 
That is, one can either use 
The M"obius transformation is helpful in expressing various
quantities, like the interaction indices discussed later, in a
more compact form. It also serves as an alternative repre-
sentation of a fuzzy measure, called M"obius representation.
That is, one can either use v or M to perform calculations,
whichever is more convenient. The conditions of mono-
tonicity of a fuzzy measure, and the boundary conditions
v(∅) = 0, v(N) = 1 are expressed, respectively, as
\[ \mathcal{M}(B) \geq 0, \text{ for all } A \subseteq N \text{ and } i \in A, \]
\[ \mathcal{M}(\emptyset) = 0 \text{ and } \sum_{A \subseteq N} \mathcal{M}(A) = 1. \]

Definition 4: The discrete Choquet integral with respect
to a fuzzy measure v is given by
\[ C_v(x) = \frac{1}{|B|} \sum_{B \subseteq A \subseteq N} \mathcal{M}(A) \min_{i \in A} x_i. \]

There are various types of fuzzy measures, like symmetric,
additive, decomposable, sub- and supermodular, possibility
and necessity, plausibility and belief, self-dual, balanced
and Sugeno fuzzy measures to name a few [6]. In this
contribution we are specifically interested in K-additive fuzzy
measures.

Definition 5: A fuzzy measure v is called K-additive (1 ≤
K ≤ n) if its M"obius transformation verifies
\[ \mathcal{M}(A) = 0 \]
for any subset A with more than K elements, |A| > K, and
there exists a subset B with k elements such that \( \mathcal{M}(B) \neq 0 \).

When dealing with multiple inputs, it is often the case
that these are not independent, and there is some interaction
(positive or negative) among the inputs. For instance, two
or more inputs may point essentially to the same concept. If
the inputs are combined by using, e.g., weighted means, their
scores will be double counted. To measure such concepts as
the importance of an input and interaction among the inputs
we will use the concepts of Shapley value, which measures
the importance of an input i in all possible coalitions, and
the interaction index, which measures the interaction of a
pair of inputs i, j in all possible coalitions [6], [24].

Definition 6: Let v be a fuzzy measure. The Shapley index
for every i ∈ N is
\[ \phi(i) = \sum_{A \subseteq N \setminus \{i\}} \frac{(n - |A| - 2)|A|!}{(n - 1)!} \times [v(A \cup \{i\}) - v(A \cup \{i\}) + v(A)]. \]
The Shapley value is the vector φ(v) = (φ(1), . . . , φ(n)). It
satisfies \( \sum_{i=1}^{n} \phi(i) = 1 \).

Definition 7: Let v be a fuzzy measure. The interaction
index for every pair i, j ∈ N is
\[ I_{ij} = \sum_{A \subseteq N \setminus \{i,j\}} \frac{(n - |A| - 2)|A|!}{(n - 1)!} \times [v(A \cup \{i,j\}) - v(A \cup \{i\}) + v(A \cup \{j\}) + v(A)]. \]
The interaction indices verify \( I_{ij} < 0 \) as soon as i, j are
positively correlated (negative synergy). Similarly \( I_{ij} > 0 \) for
negatively correlated inputs (positive synergy). \( I_{ij} \in [-1,1] \)
for any pair i, j.

Definition 8: Let v be a fuzzy measure. The interaction
index for every set A ⊆ N is
\[ I(A) = \sum_{B \subseteq A \subseteq N \setminus \emptyset} \frac{(n - |B| - |A|)!|B|!}{(n - |A| + 1)!} \sum_{C \subseteq A} (-1)^{|A\setminus B|} v(B \cup C). \]
M"obius transformation helps one to express the indices
mentioned above in a more compact form [6], [24], [25]
\[ \phi(i) = \sum_{B \subseteq A \subseteq N} \frac{1}{|B|} \mathcal{M}(B), \]
\[ I(A) = \sum_{B \subseteq A \subseteq N} \frac{1}{|B| - |A| + 1} \mathcal{M}(B). \]
Interaction indices also serve as an alternative representation
of fuzzy measures, beside the standard and M"obius repre-
sentations. One can recover other representations from the
interaction indices, in particular
\[ v(A) = \sum_{B \subseteq N \setminus A} \beta^A_B I(B), \]
where \( \beta^A_B = \frac{k!}{k^j} B_{k-j} \) and \( B_0 = 1, B_k = - \sum_{j=0}^{k-1} \frac{1}{k-j} B_{k-j} B_k, k > 0 \) are Bernoulli numbers [26].
The monotonicity property is also expressed in terms of interaction indices. In the general case such expression is given in [6], and for $\mathcal{K}$-additive fuzzy measures this condition is expressed below in Eq.(10).

A fundamental property of $\mathcal{K}$-additive fuzzy measures, which justifies their use in simplifying interactions between the criteria in multiple criteria decision making is the following [24].

Proposition 1: Let $v$ be a $\mathcal{K}$-additive fuzzy measure, $1 \leq \mathcal{K} \leq n$. Then

- $I(A) = 0$ for every $A \subseteq \mathcal{N}$ such that $|A| > \mathcal{K}$;
- $I(A) = M(A)$ for every $A \subseteq \mathcal{N}$ such that $|A| = \mathcal{K}$.

Thus $\mathcal{K}$-additive measures acquire an interesting interpretation. These are fuzzy measures that limit interaction among the criteria to groups of size at most $\mathcal{K}$. For instance, for 2-additive fuzzy measures, there are pairwise interactions among the criteria but no interactions in groups of 3 or more.

In this paper we will deal with 2-additive fuzzy measures.

This case we have the following expressions.

$$C_v(x) = \sum_{i=1}^{n} M(\{i\}) x_i + \sum_{\{i,j\} \subseteq \mathcal{N}} M(\{i,j\}) \min(x_i, x_j).$$

$$C_v(x) = \sum_{\{i,j\} \subseteq \mathcal{N}} \min(x_i, x_j) I_{ij} + \sum_{i=1}^{n} x_i \phi(i) - \frac{1}{2} \sum_{j \neq i} I_{ij},$$

$$v(\{i\}) = M(\{i\}) = \phi(i) - \frac{1}{2} \sum_{j \neq i} I_{ij},$$

III. KNN APPROXIMATION WITH CHOQUET INTEGRAL

The main idea of the proposed approach is to replace the weighted arithmetic mean used to combine the inputs $f(x_i)$ in the kNN method with a Choquet integral $C_v(f(x_1), \ldots, f(x_k))$. Previous studies [16] indicate that this is a fruitful approach which increases the accuracy of approximation and classification. In [16] the author has defined a global fuzzy measure based on the “diversity” of the set of training examples. In our approach we define separate fuzzy measures for each $x_i$ with the purpose to take into account pairwise interactions between the inputs, related to the relative orientation of the abscissae of the data. For this we need to construct a fuzzy measure $v$ based on the relative positions of the data.

We note that for function approximation, we need to combine the inputs not as $f(x) = C_v(f(x_1), \ldots, f(x_k))$ (where the output depends on the ordering of magnitudes of $f(x_1), \ldots, f(x_k)$), but with a construction, called induced Choquet integral [27], $f(x) = C_v(z, f(x_1), \ldots, f(x_k))$, in which it depends on the ordering of the components of an auxiliary vector $z$, called the order inducing variable. In our case, the inducing variable reflects the distances between $x$ and the data $x_1, \ldots, x_k$, e.g., $z = (-||x - x_1||, \ldots, -||x - x_k||)$.

**Definition 9:** The induced discrete Choquet integral with respect to a fuzzy measure $v$ and the inducing variable $z$ is given by

$$C_v(z, x) = \sum_{i=1}^{n} x_i [v(\{j|z_j \geq z(i)\}) - v(\{j|z_j \geq z(i+1)\})],$$

where $z, \phi = (z(1), z(2), \ldots, z(n))$ is a non-decreasing permutation of the input $z$.

We construct a 2-additive fuzzy measure based on the Shapley values $\phi(i), i = 1, \ldots, k$, which correspond to the weights of the arithmetic mean in the standard kNN, plus the approximated interaction indices $I_{ij}$. Consider two data $x_i$ and $x_j$ (among the $k$ nearest neighbors) and the point in question $x$. We wish to assign positive values to the interaction index $I_{ij}$ if both $x_i$ and $x_j$ are on the same side of $x$, and zero if they are on the opposite sides. Denote by $\alpha_{ij}$ the angle between the vectors $x_i - x$ and $x_j - x$. Then cosine of $\alpha_{ij}$ will be a reasonable quantity satisfying this requirement, which in addition is easily computed by

$$\cos(\alpha_{ij}) = \frac{(x_i - x) \cdot (x_j - x)}{||x_i - x|| ||x_j - x||},$$

where $\cdot$ is the standard scalar product. Thus we take for all $i, j \in \{1, \ldots, k\}$: $I_{ij} = \max\{\cos(\alpha_{ij}), 0\} \equiv \cos(\alpha_{ij})^+$.

We know that if the values of $\phi(i), i = 1, \ldots, k$ and of $I_{ij}, i, j = 1, \ldots, k$ are specified, there is a unique 2-additive measure $v$ (possibly non monotonic) [6], [24], p.429, with such values. To ensure monotonicity of $v$ we impose the constraints

$$\frac{1}{2} \left( \sum_{j \in \mathcal{N} \setminus \{i\}} I_{ij} - \sum_{j \not\in A} I_{ij} \right) \leq \phi(i),$$

for all $A \subseteq \mathcal{N} \setminus \{i\}, i = 1, \ldots, k$, where $\mathcal{N} = \{1, \ldots, k\}$ and $\phi(i)$ are the Shapley indices. The constraints are satisfied if and only if $v$ is a 2-additive fuzzy measure.

Of course our choice of $\phi(i)$ and $I_{ij}$ may be inconsistent with (10). Let us denote by $\tilde{\phi}(i)$ and $\tilde{I}_{ij}$ the values provided by kNN and $\cos(\alpha_{ij})^+$, which may be inconsistent with (10). We enforce the constraints by choosing $\phi(i)$ and $I_{ij}$ which minimize the expression

$$\sum_{i=1}^{k} (\tilde{\phi}(i) - \phi(i)) + \sum_{i \neq j} |\tilde{I}_{ij} - I_{ij}|$$

subject to (10) and $\sum_{i=1}^{k} \phi(i) = 1$.

Denote by $r_i = r_i^+ - r_i^- = \phi(i) - \tilde{\phi}(i), s_{ij} = s_{ij}^+ - s_{ij}^- = \tilde{I}_{ij} - I_{ij}$. We translate the above minimization problem into
a linear programming problem

\[
\begin{align*}
\min & \quad \sum_{i=1}^{k} r_i^+ + r_i^- + \sum_{j \neq i} s_{ij}^+ + s_{ij}^- \\
\text{s.t.} & \quad r_i^+ - r_i^- = \tilde{\phi}(i) - \phi(i), \\
& \quad s_{ij}^+ - s_{ij}^- = \tilde{I}_{ij} - I_{ij}, \\
& \quad \frac{1}{k} \left( \sum_{j \in N \setminus \{i\}, I_{ij} - \sum_{l \in \mathcal{A}} I_{lj} \right) \leq \phi(i), \\
& \quad \sum_{i=1}^{k} \phi(i) = 1, \\
& \quad r_i^+, r_i^-, s_{ij}^+, s_{ij}^- \geq 0, \\
& \quad \phi(i) \geq 0, -1 \leq I_{ij} \leq 1,
\end{align*}
\]

for all \( \mathcal{A} \subseteq N \setminus \{i\}, i=1, \ldots, k \).

By solving problem (11) for each \( x \) and its \( k \) nearest neighbors \( x_i \), we obtain the values \( \phi(i) \) and \( I_{ij} \) which best fit our initial guess \( \tilde{\phi}(i) \) and \( \tilde{I}_{ij} \) and satisfy (10).

The problem (11) however can be quite large (although not particularly difficult). It involves \( k \Omega(k-1) \) inequality constraints. Thus it is computationally very expensive to set up and solve such an optimization problem for every \( x \). Below we propose a simplified, very fast algorithm for ensuring monotonicity of the fuzzy measure and computing the value of the Choquet integral.

Note that the value of the induced Choquet integral with such a fuzzy measure \( v, C_{v,x} \), is required only for one inducing variable \( x \). By looking at Eq. (9), only \( n \) values of fuzzy measure \( v \) participate in its calculation, depending on the inducing variable, and one can write \( f(x) = C_{v,x} = \sum_{i=1}^{k} w_i f(x_i) \), where the coefficients \( w_i \) are found from (6) and (9). We only need to ensure that \( w_i \geq 0 \) and \( \sum w_i = 1 \), which is easily done.

We performed a number of numerical experiments for small dimensional problems to illustrate the usefulness of the new method, and report the following results. For the univariate case we generated 20 random data (the test function \( f_3(x) = \max(x \sin(\pi x), x^2) \)) and fitted the data using the standard kNN method and the proposed method based on the Choquet integral. The results are illustrated on Fig. 1.

For two and three dimensional cases we took test functions \( f_2(x) = \max(x_1^2 \sin(\pi x_2), x_1 x_2^2) \) (Fig. 5) and \( f_3(x) = f_3(f_2(x_2), x_3) \) on \([0,1]^2 \) and \([0,1]^3 \) respectively. In the two-variate case we generated a) 256 lined data (as on Figures 1 and 6-8), b) 121 data on a \( 11 \times 11 \) regular grid (Fig. 9-10) and c) 256 uniformly distributed random data. Figures 6-10 illustrate approximation in the two-variate case. Note the improvement obtained by the proposed method compared to the standard kNN.

In Tables I-III we present the RMSE and the maximal error of approximation (as the test data we used the values of test functions at a fine regular grid (of 500, 2500 and 10000 data points).

We note that the improvement by the new method is quite substantial in one- and two-dimensional cases, especially when the data have special structure (lined) (which was the
motivation for this paper). The reduction of the maximal error is quite significant in 2D. In all cases the performance of the new method is not sensitive to the choice of \( k \), as long as \( k \) is sufficiently large. In no case did the proposed method give results worse than the standard kNN.

### TABLE I

<table>
<thead>
<tr>
<th>training data</th>
<th>k</th>
<th>standard</th>
<th>proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 (random)</td>
<td>3</td>
<td>0.02114 (0.0004)</td>
<td>0.01644 (0.05262)</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>0.02350 (0.10581)</td>
<td>0.01183 (0.05257)</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.03135 (0.13946)</td>
<td>0.01021 (0.05228)</td>
</tr>
</tbody>
</table>

### TABLE II

<table>
<thead>
<tr>
<th>training data</th>
<th>k</th>
<th>standard</th>
<th>proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 (lined)</td>
<td>3</td>
<td>0.04245 (0.14851)</td>
<td>0.04230 (0.14850)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.04197 (0.13630)</td>
<td>0.04180 (0.135869)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.03077 (0.15603)</td>
<td>0.02571 (0.07984)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.03561 (0.21237)</td>
<td>0.01117 (0.03733)</td>
</tr>
<tr>
<td>256 (random)</td>
<td>3</td>
<td>0.02287 (0.13947)</td>
<td>0.02026 (0.13514)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.03078 (0.19781)</td>
<td>0.01703 (0.13483)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.04601 (0.24344)</td>
<td>0.01668 (0.13518)</td>
</tr>
<tr>
<td>121 (grid)</td>
<td>3</td>
<td>0.00938 (0.06695)</td>
<td>0.00919 (0.06528)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.02399 (0.13605)</td>
<td>0.00769 (0.06820)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.04821 (0.20245)</td>
<td>0.00762 (0.06832)</td>
</tr>
</tbody>
</table>

### TABLE III

<table>
<thead>
<tr>
<th>training data</th>
<th>k</th>
<th>standard</th>
<th>proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1536 (lined)</td>
<td>3</td>
<td>0.04837 (0.32283)</td>
<td>0.04839 (0.32290)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.04613 (0.29225)</td>
<td>0.04609 (0.29451)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.03962 (0.22121)</td>
<td>0.03910 (0.22215)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.02809 (0.30602)</td>
<td>0.02029 (0.15018)</td>
</tr>
</tbody>
</table>

IV. KNN CLASSIFICATION WITH CHOQUET INTEGRAL

We consider a multiclass classification problem. The approach is very similar to that of function approximation: we aim at canceling out the redundant votes of the neighbors that lie on the same side of \( x \). In traditional kNN classification, the votes are combined by a weighted arithmetic mean

\[
V(c_m) = \sum_{i=1}^{k} w_i \delta_{im}, \quad m = 1, \ldots, M, \quad \delta_{im} = 1 \text{ iff } i = m, \quad w_i = \frac{1}{k} \text{ and } M \text{ is the total number of classes.}
\]

We replace this aggregation function with the induced Choquet integral \( C_{v,m} \), where the fuzzy measure \( v \) is obtained as...
in Section III. The difference to the function approximation case is the integrand. Here the votes are computed by $V(x_m) = C_l(x_m)(\delta_m)$, with $\delta_m = (\delta_1, \delta_2, \ldots, \delta_M)$, and $m = 1, \ldots, M$.

We pointed out that solving Problem (11) could be numerically expensive, so again we are interested in a shortcut. Now we need to aggregate $M$ binary vectors $\delta_m$, and the simplification will be different. This constitutes one of the lines of our future research.

V. CONCLUSIONS

In this contribution we proposed a method of function approximation and classification based on the traditional kNN, but which favors the neighbors which are distributed all around $x$. The method is similar in its objective to the natural neighbor method, but uses a different technique based on fuzzy measures. The weighted arithmetic mean in the kNN method is replaced by the more general Choquet integral, and the fuzzy measure is chosen adaptively for each point $x$, so that neighbors distributed around $x$ are favored and those on the same side are penalized. We presented results of several numerical experiments in low dimension which illustrate the advantages of the new method. However it is not clear whether it will hold true for higher dimensions.

Our future research is along the following lines: a) to analyze the computational complexity of the proposed method and the specified shortcuts; b) to validate this method on a number of standard test problems in approximation and classification, and to benchmark it against the alternatives; c) to examine high dimensional problems; and d) to consider generalizations of the Choquet integral.

REFERENCES