

Extended k-nearest neighbours based on evidence theory

Hui Wang

School of Computing and Mathematics
University of Ulster, Northern Ireland, UK
h.wang@ulster.ac.uk

David Bell

School of Computer Science
Queen's University of Belfast, Northern Ireland, UK
da.bell@qub.ac.uk

Abstract

An evidence theoretic classification method is proposed in this paper. In order to classify a pattern we consider its neighbours, which are taken as parts of a single source of evidence to support the class membership of the pattern. A single mass function or *basic belief assignment* is then derived, and the belief function and the pignistic (“betting rates”) probability function can be calculated. Then the (posterior) conditional pignistic probability function is calculated and used to decide the class label for the pattern.

It is shown that such a classifier extends the standard majority voting based k-nearest neighbour classifier, and it is an approximation to the optimal Bayes classifier.

In experiments this classifier performed as well as or better than the voting and distance weighted k-nearest neighbours classifiers with best k , and its performance became stable when the number of neighbours considered is greater than 4.

Keywords: Dempster-Shafer theory, mass function, basic belief assignment, probability, classification, k-nearest neighbours.

1 Introduction

The Dempster-Shafer theory of evidence [1] is now widely accepted as a rich and flexible framework for representing and reasoning with imperfect information. Situations

of weak knowledge and heterogeneous information sources can be easily modelled, making them quite suitable for many application domains such as medical diagnosis and pattern recognition [2].

Denoeux proposed an evidence theoretic k-nearest neighbour (kNN) method for classification based on Dempster-Shafer theory, which is also called the *TBM classifier* [3, 4], where TBM is short for *Transferable Belief Model* [5]. In this method, each neighbour of a pattern to be classified is considered as a piece of evidence to support certain propositions about the class membership of the pattern. Based on the evidence, basic beliefs are assigned to the subsets of all classes. Such *basic belief assignments* are obtained for each of the k nearest neighbours and aggregated using the Dempster Rule. In order to improve the classification accuracy of this method, a procedure is proposed [6] to determine the optimal or near-optimal parameter values from the data by minimising an error function. This method proves to be very competitive with the standard kNN methods.

With the TBM classifier k basic belief assignments are derived and then combined into a single basic belief assignment using Dempster Rule. It is well known, however, that combining basic belief assignments is computationally expensive [2, 7]. As remarked in [2], the demands in space and time of the combination operation are so high that the operation becomes impractical when the *frame of discernment* Ω has more than 15 to 20 elements.

Many methods have been designed to speed up the combination process [2, 8–12]. These solutions are approximations of the combined basic belief assignment. If we need an exact solution to the combination for large Ω , we have yet to explore new efficient methods [2]. For applications with large number of classes (e.g., text categorisation, where there are usually tens or even hundreds of categories) the TBM classifier may not be applicable without an efficient combination method.

In this paper we suggest an alternative method for evidence theoretic classification to avoid the need of combination and the problem of choosing the best k for kNN. Instead of k basic belief assignments as used in the TBM Classifier we construct a single basic belief assignment from the neighbours. A classification rule is designed based on the basic belief assignment. This rule is shown to be an extended kNN method and an approximation of the optimal Bayes classifier. Experimental results are presented to show the competence of this rule.

2 Related work

2.1 A review of k-nearest neighbour rule

k-Nearest neighbour rule [13], kNN, is well known in the pattern recognition literature. According to this rule, an unclassified pattern (sample, instance) is assigned to the class

represented by a majority of its k nearest neighbours. This rule is nowadays usually called *voting kNN rule*. Cover and Hart [14] have shown that, as the number N of patterns and k both tend to infinity in such a manner that $k/N \rightarrow 0$, the error rate of the kNN rule approaches the optimal Bayes error rate.

kNN is popular in pattern recognition community due mainly to its good performance and its simple-to-use feature.

Since the inception of kNN some variations have been proposed in order to improve its performance.

2.1.1 Distance weighted kNN rule

In voting kNN the k neighbours are implicitly assumed to have equal weight in decision, regardless of their distances to the pattern x to be classified. It is intuitively appealing to give different weights to the k neighbours based on their distances to x , with closer neighbours having greater weights.

Let d be a distance measure, and x_1, x_2, \dots, x_k be the k nearest neighbours of x arranged in increasing order of $d(x_i, x)$. So x_1 is the first nearest neighbour of x . [15] proposes to assign to the i th nearest neighbour x_i a weight w_i defined as

$$(1) \quad w_i = \begin{cases} \frac{d(x_k, x) - d(x_i, x)}{d(x_k, x) - d(x_1, x)}, & \text{if } d(x_k, x) \neq d(x_1, x) \\ 1, & \text{if } d(x_k, x) = d(x_1, x) \end{cases}$$

Pattern x is assigned to the class for which the weights of the representatives among the k nearest neighbours sum to the greatest value. This rule was shown by Dudani to yield lower error rates than those obtained using the voting kNN rule. However some other researchers reached less optimistic conclusions [16–18]. [3] provides an excellent and detailed review of distance weighted kNN.

2.1.2 Evidence theoretic kNN rule

The evidence theoretic k -nearest neighbour rule [3] is a pattern classification method based on the Dempster-Shafer theory of belief functions. In this approach, each neighbour 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 neighbours of the pattern under consideration and aggregated using the Dempster's rule of combination. In [6] Zouhal and Denoeux state "... in many situations, this method was found experimentally to yield lower error rates than other methods using the same information". They then proposed an optimisation procedure to 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.

2.2 Classifier combination

Researchers have long pursued the promise of harnessing multiple classifiers to synthesize a more accurate classification procedure via some combination of the outputs of the contributing classifiers to build a *multiple classifier system* (MCS) [19, 20]. The development of multiple classifier systems has received increasing attention recently [21–24].

It should be noted that, in general, there is no guarantee that a multiple classifier system will be more robust or more accurate than either the individual classifiers of which it is comprised or than the best single classifier that might otherwise be built [22].

2.2.1 Classifier combination strategies

Since the best combination of a set of classifiers depends on the application and on the classifiers to be combined, there is no single, best combination scheme nor any unequivocal relationship between the accuracy of a multiple classifier system and the individual constituent classifiers.

Here are some typical combination strategies:

- A classifier is composed from multiple distinct classifiers by selecting the best classifier to use in different situations or contexts. For example, we may perform analytical or empirical studies to identify the most accurate classifier in some setting, seeking to learn about accuracy over output scores or some combination of output scores and features considered in the analysis.
- Another procedure for combining classifiers considers **outputs** generated by the contributing classifiers. For example, in a voting analysis, a combination function considers the final decisions made by each classifier as votes that influence an overall decision about the best classification.
- In a finer grained approach to combining multiple classifiers, the **scores** generated by the contributing classifiers (e.g., confidence measures or estimates of the posterior probabilities of the output classes, as Bayes classifiers and properly trained feed-forward neural networks do) are taken as inputs to a combination function.
- More complex strategies are used in ensemble methods. They first solve a classification or regression problem by creating multiple learners that each attempt to solve the task independently, then use the procedure specified by the particular ensemble method for selecting or weighting the individual learners. Ensemble methods include such techniques as Bayesian averaging, bagging, boosting, stacking, cascade generalization, and hierarchical mixture of experts.

Much of the work on combining text classifiers has centred on the use of basic strategies for selecting the best classifier or for combining the output of multiple classifiers. As some examples, [25] used weighted linear combinations of system ranks or scores; [26] used linear combinations of probabilities or log odds scores; [27] used a linear combination of normalized scores; [28] used voting and classifier selection techniques; and [29] used category-averaged features to pick a (potentially different) classifier to use for each category. [20] presents a probabilistic combination procedure that uses the **context-sensitive reliabilities** of classifiers [30] — a set of features that provide a low-dimensional abstraction on the discriminatory context for learning about reliability, along with the **outputs** or **scores** of classifiers and the **domain-level features**.

3 Definition and notations

Let Ω be a finite set called *frame of discernment*. A *mass function* or *basic belief assignment* is a mapping $m : 2^\Omega \rightarrow [0, 1]$ such that

$$\begin{aligned} \sum_{X \subseteq \Omega} m(X) &= 1 \\ m(\emptyset) &= 0 \end{aligned}$$

The mass $m(X)$ measures the amount of belief that is exactly committed to X . $X \subseteq \Omega$ is called a *focal element* of m if $m(X) > 0$. Note that in TBM $m(\emptyset)$ need not be 0, but we assume so to simplify the presentation and this does not affect the validity of our results.

Given two mass functions m_1 and m_2 defined over the same Ω we can combine them using the *Dempster Rule of Combination*¹ as follows:

$$(2) \quad m_1 \oplus m_2(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)}.$$

The *pignistic probability function* [5, 31] associated with m is $BetP : 2^\Omega \rightarrow [0, 1]$ such that for any $E \subseteq \Omega$

$$(3) \quad BetP(E) = \sum_{X \subseteq \Omega} m(X) \frac{|E \cap X|}{|X|},$$

For $E_1, E_2 \subseteq \Omega$, we can define *conditional pignistic probability* $BetP(E_1|E_2)$ as follows, in a way similar to conditional (classical) probability:

$$BetP(E_1|E_2) = BetP(E_1 \cap E_2) / BetP(E_2).$$

¹This is called *normalised conjunctive rule of combination* in [4], where the *disjunctive rule of combination* is also defined, which is not further pursued in this paper.

This conditional pignistic probability is different in general from the pignistic probability computed from conditional belief functions [5]. A full discussion of this issue is beyond the scope of this paper ².

4 Classification based on a single mass function from neighbours

In the TBM classifier, k mass functions are derived from neighbours for classification. In this section we show how to derive a single mass function from neighbours and how such a mass function is used for classification.

4.1 Probabilistic modelling of classification

Classification is a process to assign patterns to predefined categories (classes) [3]. The patterns are described by two variables:

- \mathbf{x} : vector of d attributes or features. The i th attribute is x_i whose domain is denoted by $dom(x_i)$, and the domain of \mathbf{x} is denoted by $V \stackrel{\text{def}}{=} dom(x_1) \times \cdots \times dom(x_d)$. In other words V is a d dimensional space.
- y : class variable, whose domain is a finite set denoted by W .

A classifier is a mapping

$$f : V \rightarrow W$$

which generates a class value (label) for any new pattern described by feature vector \mathbf{x} .

Building a classifier from data is usually called *supervised learning*. The data for building the classifier, or *training dataset*, can be specified as

$$D = \{ \langle t_i, c_i \rangle : t_i \in V, c_i \in W, \text{ where } i = 1, 2, \dots, n \}$$

It is usually assumed that D is a sample drawn from an (unknown) distribution p over $V \times W$, and that new examples will be drawn from the same distribution.

4.2 Deriving mass function

We take the frame of discernment to be V , i.e., $\Omega \stackrel{\text{def}}{=} V$. We are to show how to derive a mass function from given dataset D .

Suppose we want to classify $t \in \Omega$. In order to classify t we consider h *neighbourhoods* of t : E_1, E_2, \dots, E_h . Each *neighbourhood* is a region in Ω (V) covering a set

²Interested readers are invited to look at the example in [5] about Peter, Paula and Mary.

of neighbours of t . For simplicity we take E_i as a set of the neighbours, so $E_i \in 2^\Omega$. We let $E = \bigcap_i E_i$, $|E_i| = k_i$, and $N = \sum_{i=1}^h k_i$ – the total number of neighbours considered. The neighbourhoods may overlap, therefore it is likely that $|E| < N$. Some neighbours are counted more than some others so they play more important roles. This is in spirit similar to the weighted kNN methods [15, 32]. Figure 1 shows an example of 3 neighbourhoods.

The neighbours provide a source of evidence supporting propositions concerning the class membership of t . In the standard kNN, one neighbourhood of k neighbours is considered and classification is done, for example, by a majority voting among the neighbours. In the TBM classifier one neighbourhood of k neighbours is considered. The neighbours are taken as separate sources of evidence and used to generate k mass functions, each of which represents the support by a single neighbour.

Here we consider h neighbourhoods. Each neighbourhood is taken as one *part* of a source of evidence, and all neighbourhoods – together as a source of evidence – are used to generate a single mass function representing partial supports by different neighbourhoods.

Consider $E_i \in 2^\Omega$ and $c \in W$. We are interested in the joint probability $P(E_i, c)$ – the probability that a randomly selected element x of Ω belongs to E_i and is in class c , i.e., $x \in E_i$ and $f(x) = c$. Having no specific knowledge about the distribution p we can apply *the principle of indifference* to approximate $P(E_i, c)$ by

$$\bar{P}(E_i, c) = |E_i^c|/|D|$$

where $E_i^c = \{x \in E_i : f(x) = c\}$.

Then we define a function $m[t]$, induced for t from the h neighbourhoods, as a mapping $m[t] : 2^\Omega \rightarrow [0, 1]$ such that, for $X \in 2^\Omega$ and $c \in W$,

$$m[t](X, c) = \begin{cases} \bar{P}(X, c)/K, & \text{if } X = E_i \text{ for some } i \\ 0, & \text{otherwise} \end{cases}$$

Here K is a normalising factor. It follows that $K = \sum_{i=1}^h \sum_{c \in W} \bar{P}(E_i, c)$. Note that by $m[t](X, c)$ we mean $m[t](X \cap \{x \in \Omega : f(x) = c\})$, which is similar to the interpretation of joint probability $P(X, c)$ (see, for example, [33]).

Clearly $m[t]$ is a mass function. In particular $\sum_{c \in W} \sum_{X \in 2^\Omega} m[t](X, c) = \sum_{c \in W} \sum_{i=1}^h m[t](E_i, c) = 1$.

4.3 Classification based on $m[t]$

We propose to classify new patterns through conditional pignistic probability. For this we specify the joint pignistic probability as $\text{Bet}\bar{P} : 2^\Omega \rightarrow [0, 1]$ such that, for $X \in 2^\Omega$

and $c \in W$,

$$\overline{BetP}(X, c) = \sum_{i=1}^h m[t](E_i, c) \times \frac{|X \cap E_i|}{|E_i|}$$

The following proposition confirms that this is indeed a probability function.

Proposition 1. \overline{BetP} is a probability function on Ω . That is,

1. For any $X \in 2^\Omega$ and $c \in W$, $\overline{BetP}(X, c) \geq 0$;
2. $\overline{BetP}(\Omega) = 1$;
3. For any $X_1, X_2 \in 2^\Omega$ and $c \in W$, $\overline{BetP}(X_1 \cup X_2, c) = \overline{BetP}(X_1, c) + \overline{BetP}(X_2, c)$ if $X_1 \cap X_2 = \emptyset$.

Proof. The first claim is true following the fact that $m[t](X, c) \geq 0$ for any $X \in 2^\Omega$.

The second claim is true since

$$\begin{aligned} \overline{BetP}(\Omega) &= \sum_{c \in W} \overline{BetP}(\Omega, c) \\ &= \sum_{c \in W} \sum_{i=1}^h m[t](E_i, c) \times \frac{|\Omega \cap E_i|}{|E_i|} \\ &= \sum_{c \in W} \sum_{i=1}^h m[t](E_i, c) \times \frac{|E_i|}{|E_i|} = \sum_{c \in W} \sum_{i=1}^h m[t](E_i, c) = 1 \end{aligned}$$

Now we consider the third claim. $E_i \cap (X_1 \cup X_2) = (E_i \cap X_1) \cup (E_i \cap X_2)$. If $X_1 \cap X_2 = \emptyset$ then $|E_i \cap (X_1 \cup X_2)| = |E_i \cap X_1| + |E_i \cap X_2|$. As a result we have

$$\begin{aligned} \overline{BetP}(X_1 \cup X_2, c) &= \sum_{i=1}^h m[t](E_i, c) \frac{|E_i \cap (X_1 \cup X_2)|}{|E_i|} \\ &= \sum_{i=1}^h m[t](E_i, c) \frac{|E_i \cap X_1| + |E_i \cap X_2|}{|E_i|} \\ &= \sum_{i=1}^h m[t](E_i, c) \frac{|E_i \cap X_1|}{|E_i|} + \sum_{i=1}^h m[t](E_i, c) \frac{|E_i \cap X_2|}{|E_i|} \\ &= \overline{BetP}(X_1, c) + \overline{BetP}(X_2, c) \end{aligned}$$

□

Note that E_i is a region covering some neighbours of t so we have $t \in E_i$. We can understand t as a singleton set, therefore $t \cap E_i^c = \{t\}$ and $|\{t\}| = 1$. Then we have the

following joint, marginal and conditional pignistic probabilities for $t \in \Omega$,

$$\begin{aligned}
\overline{BetP}(t, c) &= \sum_{i=1}^h m[t](E_i, c)/|E_i| \\
\overline{BetP}(t) &= \sum_{c \in W} \overline{BetP}(t, c) \\
\overline{BetP}(c|t) &= \overline{BetP}(t, c)/\overline{BetP}(t) \\
&= \frac{\sum_{i=1}^h m[t](E_i, c)/|E_i|}{\overline{BetP}(t)} = \frac{\sum_{i=1}^h \overline{P}(E_i, c)/|E_i|}{K \times \overline{BetP}(t)} \\
&= \frac{\sum_{i=1}^h |E_i^c|/|E_i|}{|D| \times K \times \overline{BetP}(t)} = \delta(t) \sum_{i=1}^h |E_i^c|/|E_i|
\end{aligned}$$

where $\delta(t)^{-1} = |D| \times K \times \overline{BetP}(t)$.

Classification then proceeds using the following rule:

Rule 1. For $x \in \Omega$, $f(x) = c_*$ if $\overline{BetP}(c_*|x) \geq \overline{BetP}(c|x) \quad \forall c \in W$.

If $h = 1$ and we use Rule 1 for classification, then $\overline{BetP}(c|t) = \delta(t)|\{x \in E_1 : f(x) = c\}|/|E_1|$ and we end up with a majority voting kNN. Therefore Rule 1 is an extended majority voting based kNN.

Note that we don't assume that classes are exactly known for the neighbours. Consider a neighbour s . It is possible that the class label $f(s)$ is either known or unknown. It turns out that both cases can be dealt with by our kNN method. If $f(s)$ is known, the contribution of s in classifying t is accounted for as above. If $f(s)$ is unknown, s is not counted in any class in the calculation of $\overline{BetP}(c|t)$. Therefore s has no contribution to the classification of t . We argue that this agrees well with our intuition about the role of s if $f(s)$ is unknown.

5 An approximation of optimal classifiers

Here we discuss the relationship between our classifier and the Bayes optimal classifier from a theoretical perspective.

If the distribution p is known, then $P(X)$ is available for every $X \in 2^\Omega$. We can then define a mass function not specific to t as $m : 2^\Omega \rightarrow [0, 1]$ such that, for any $E \in 2^\Omega$ and $c \in W$,

$$(4) \quad m(E, c) = P(E, c)/K$$

where $K \stackrel{\text{def}}{=} \sum_{X \in 2^\Omega, c \in W} P(X, c)$ is a normalising factor. We then have $BetP(E, c)$ as in the previous section.

5.1 Optimal Bayes classifier

The *optimal (Bayes) classifier* $f^* : \Omega \rightarrow W$ is defined by

$$f^* : x \mapsto c_* \text{ such that } R(c_*|x) \leq R(c|x) \text{ for all } c \in W$$

with

$$R(c_*|x) = \sum_{c \in W} L(c_*, c)P(c|x)$$

and $L(c_*, c)$ is the loss incurred if the actual class is c_* but the assigned class is c . f^* minimizes the overall risk:

$$R(f) = \int R(f(x)|x)p(x)dx$$

We take a simplistic approach by setting

$$L(c_*, c) = \begin{cases} 0, & \text{if } c_* = c; \\ 1, & \text{otherwise.} \end{cases}$$

Thus the optimal classifier can be described by the following rule.

Rule 2. For $x \in \Omega$, $f^*(x) = c_*$ if $P(c_*|x) \geq P(c|x) \quad \forall c \in W$.

Similarly if the pignistic probability is known completely we have the following optimal pignistic classifier.

Rule 3. For $x \in \Omega$, $f^*(x) = c_*$ if $BetP(c_*|x) \geq BetP(c|x) \quad \forall c \in W$.

5.2 $BetP(c|t)$ vs $P(c|t)$

Assume that Ω is finite. Let $\binom{N}{n}$ be the combinatorial number representing the number of ways of picking n unordered outcomes from N possibilities. From combinatorics we know that $\binom{N}{n} = \frac{N!}{(N-n)!n!}$.

$$\text{Let } N = |\Omega|, K = \sum_{i=1}^N \binom{N-1}{i-1}, \alpha \stackrel{\text{def}}{=} \frac{1}{K} \sum_{i=1}^N \frac{\binom{N-2}{i-1}}{i}, \text{ and } \beta \stackrel{\text{def}}{=} \frac{1}{K} \sum_{i=1}^N \frac{\binom{N-2}{i-2}}{i}.$$

Theorem 1. $BetP(x) = \alpha p(x) + \beta$ for $x \in \Omega$.

Proof.

$$\begin{aligned}
BetP(x) &= \sum_{Y \subseteq \Omega} \frac{x \cap Y}{|Y|} m(Y) = \sum_{Y \subseteq \Omega, x \in Y} \frac{1}{|Y|} \frac{P(Y)}{K} \\
&= \frac{1}{K} \sum_{Y \subseteq \Omega, x \in Y} \frac{P(Y)}{|Y|} = \frac{1}{K} \sum_{Y \subseteq \Omega, x \in Y} \frac{\sum_{z \in Y} P(z)}{|Y|} \\
&= \frac{1}{K} \sum_{i=1}^N \frac{1}{i} \sum_{Y \subseteq \Omega, |Y|=i, x \in Y} \sum_{z \in Y} P(z) \\
&= \frac{1}{K} \sum_{i=1}^N \frac{1}{i} \left(\binom{N-1}{i-1} p(x) + \binom{N-2}{i-2} \sum_{z \neq x} P(z) \right) \\
&= \frac{1}{K} \sum_{i=1}^N \frac{1}{i} \left(\binom{N-1}{i-1} p(x) + \binom{N-2}{i-2} (1 - p(x)) \right) \\
&= \frac{1}{K} \sum_{i=1}^N \frac{1}{i} \left(\binom{N-2}{i-1} p(x) + \binom{N-2}{i-2} \right) = \alpha p(x) + \beta
\end{aligned}$$

The claim then follows. \square

Since both P and $BetP$ are probability functions they are both additive. Following Theorem 1 we then have:

Corollary 2.

$$BetP(E) = \alpha P(E) + \beta |E|$$

The following theorem establishes the connection between $BetP(c|t)$ and $P(c|t)$.

Theorem 2. *Let $t \in \Omega$, and $c_1, c_2 \in W$. Then $P(c_1|t) \leq P(c_2|t) \iff BetP(c_1|t) \leq BetP(c_2|t)$.*

Proof.

$$\begin{aligned}
BetP(c_1|t) \leq BetP(c_2|t) & \iff \\
\frac{BetP(t, c_1)}{BetP(t)} \leq \frac{BetP(t, c_2)}{BetP(t)} & \iff \\
BetP(t, c_1) \leq BetP(t, c_2) & \iff \\
\alpha P(t, c_1) + \beta \leq \alpha P(t, c_2) + \beta & \iff \\
P(t, c_1) \leq P(t, c_2) & \iff \\
\frac{P(t, c_1)}{P(t)} \leq \frac{P(t, c_2)}{P(t)} & \iff \\
P(c_1|t) \leq P(c_2|t) & \iff
\end{aligned}$$

\square

This theorem says that $P(c|t)$ and $BetP(c|t)$ are equivalent in classifying $t \in \Omega$. Therefore Rule 3 is as optimal as Rule 2.

5.3 Approximating the optimal classifier

We have discussed two optimal classifiers, Rules 2 and 3, and have shown their equivalence. In practice, however, the distribution is usually unknown and we can only approximate the optimal classifiers. We usually approximate the optimal Bayes classifier (Rule 2) by estimating the posterior probability $P(c|x)$. Estimation strategies include parametric or non-parametric estimation of the class-conditional densities $p(x|c)$ in combination with priors $P(c)$, and direct estimation of $P(c|x)$ [2].

The method presented in Section 4 is in fact an approximation of the optimal classifier (Rule 3) by estimating the posterior probability $BetP(c|x)$.

6 A classifier

Based on Rule 1 we can devise classifiers. The key issue is how neighbourhoods are interpreted and selected. For example we can interpret a neighbourhood as a set of nearest i neighbours (the neighbours are decided according Euclidean distance), denoted by iNN , and we consider neighbourhoods iNN for $i = 1, 2, \dots, h$. h can be a given value or it can be set to the number of data instances in the dataset. We can also interpret a neighbourhood as hyperrectangles (or hyperspheres or hypercubes) surrounding a data instance, and take all or part of the hyperrectangles (or hyperspheres or hypercubes) as the neighbourhoods needed to calculate $BetP(c|t)$.

Clearly there are other possible interpretations of neighbourhood and neighbourhood selection strategies. A theoretical analysis or a comprehensive experimental study is beyond the scope of this paper. This task will be the subject of future work.

In this paper we adopt the hypercube interpretation of neighbourhood along with a simple selection strategy. We assume that the attributes in the dataset are all numerical. For a positive integer d we partition every attribute into $d + 1$ equal-sized intervals. This effectively gives equal weights to all attributes. Consider an attribute A , and let $\mathbf{dom}(A)$ be its domain. We arrange the intervals in ascending order as $v_0, v_1, v_2, \dots, v_d$ such that for $x \in v_i, y \in v_j, x \leq y$ if $i \leq j$. Then every value of the attribute belongs to one and only one interval. For a value a of A , i.e., $a \in \mathbf{dom}(A)$, let v_i be the interval that a belongs to. For a non-negative integer $q \leq d$ we call the following extended interval the q th order interval of a , written as $v^q(a)$:

$$v_A^q(a) = \{x \in \mathbf{dom}(A) : x \leq \max v_{[i+q]_0^d}, x \geq \min v_{[i-q]_0^d}\}$$

where

$$[n]_0^d = \begin{cases} 0, & \text{if } n < 0 \\ d, & \text{if } n > d \\ n, & \text{otherwise.} \end{cases}$$

Clearly $v_A^0(a) = v_i$.

For a data vector (tuple) t , its q th order hypercube is

$$V^q(t) = \prod_A v_A^q(t(A))$$

where $t(A)$ is the projection of t to attribute A . Furthermore we let $\mathbf{cov}(V^q(t))$ be the coverage of $V^q(t)$, i.e., the number of data instances in the hypercube.

We take each $V^q(t)$ as a neighbourhood of t , and so we have $d + 1$ neighbourhoods for t : $V^0(t), V^1(t), \dots, V^d(t)$. Clearly $\mathbf{cov}(V^i(t)) \subseteq \mathbf{cov}(V^j(t))$ if $i \leq j$.

Our neighbourhood selection strategy is as follows: for a given h we consider h neighbourhoods i NN for $i = 1, 2, \dots, h$ where 1NN is a non-empty $V^q(t)$ with the smallest q , and 2NN is $V^{q+1}(t)$, and so on.

Classification is done according to Rule 1. We call this classification procedure as *ekNN*, or *extended kNN based on evidence theory*.

6.1 Discussion

The essence of the way in which $BetP(c|t)$ is estimated is averaging the conventional k-NN estimates of the posterior probabilities $\bar{p}_k(c|t)$ over different k-NN subsets. This coincides with some heuristics used in classifier combination. As discussed in [20], classifiers can be combined by a combination function whose inputs are the **scores** generated by the contributing classifiers. The scores can be, for example, confidence measures or estimates of the posterior probabilities of the output classes, as Bayes classifiers and properly trained feed-forward neural networks do.

The contribution of this paper in this respect is the following.

1. First of all, although such a heuristic is implied in some discussions a theoretical justification for the heuristic is still needed. kNN as a classification heuristic was first proposed in [13], but it didn't gain popularity until after Cover and Hart [14] have shown that, as the number N of patterns and k both tend to infinity in such a manner that $k/N \rightarrow 0$, the error rate of the kNN rule approaches the optimal Bayes error rate.

As discussed above, Rule 3 is equivalent to the optimal Bayes classification Rule 2, and Rule 1 is an approximation of Rule 3. We can therefore say that the principle behind rule 1 is theoretically justified.

2. Secondly, although such a classifier combination heuristic is implied in some discussions a comprehensive evaluation of a multiple classifier system, where kNN classifiers are combined by a linear function and the inputs are the posterior probabilities, is yet needed. This paper serves this purpose.
3. Thirdly, it has been accepted that there is no guarantee that the combined classifier is better than any single one of the component classifiers, or the more classifiers the better [34]. According to the proved relationship between the above classification rules, the more neighbourhoods we use the closer the estimated posterior probability is to the true posterior probability. We can therefore conclude that if we combine more kNN classifiers this way we end up with a combined classifier more robust, stable, and accurate than the component single classifiers.

7 Evaluation

The evaluation was done via experiments. The purpose of the evaluation is to show if and how the above classification procedure improves upon the standard kNN and the weighting kNN.

The data used in the experiments are public datasets from UC Irvine Machine Learning Repository ³. General information about these datasets is shown in Table 1.

In our experiments we set $d = 30$ (i.e., all attributes are partitioned into 30 equal-sized intervals), h to various values, and recorded the classification accuracy. As a comparison we implemented the voting based kNN classifier (referred to by kNN) and the distance weighted kNN classifier (referred to by wkNN) as discussed in Section 2.1.1 ⁴, and experimented with various values for k (from 1 to 10) and recorded the classification accuracy for each k values. To make the results comparable we used the same interpretation of neighbourhood as for ekNN, as discussed above. In other words the neighbourhood of t used in kNN and wkNN with k is $V^q(t)$ such that $\mathbf{cov}(V^q(t)) \geq k$ and q is the smallest such value. As a result, ekNN with $h = 1$ should be the same as kNN with $k = 1$, but not the wkNN with $k = 1$.

We also include the C4.5 and SVM (support vector machine) ⁵ results on the same set of data. However the purpose of this inclusion is not for comparison but as a ref-

³<http://www.ics.uci.edu/~mllearn/MLRepository.html>

⁴The weighting function used is Eq. 1

⁵Support vector machines are a recent development and are gaining popularity while decision trees are a classical type of classifiers and are still the most widely used machine learning/data mining technique [35]. C4.5 is a popular implementation of decision tree induction technique. For an overview of various machine learning/data mining techniques, the readers are invited to consult <http://www.kdnuggets.com/publications/surveys.html>.

erence. Note that ekNN assumes a simple neighbourhood interpretation and equal weighting for the attributes; furthermore, the implementations of ekNN, kNN and wkNN were not as optimised as the versions of C4.5 and SVM we used. Throughout the experiments the validation method is 5-fold cross validation.

The results for C4.5 and SVM are shown in Table 2, and the results for kNN, wkNN and ekNN are shown respectively in Tables 3, 5, and 4.

From the results we observe the following facts:

- On average ekNN performed better than kNN and wkNN on these datasets.
- For $h \geq 3$ ekNN performed as well or better than the voting kNN with best k values.
- For $h \geq 3$ the performance of ekNN did not change much with different values of h . Such a saturation property is useful since it relieves the designer of kNN system from the burden of searching for the optimal value of k .
- In the cases of kNN and wkNN, the best performance comes with varied k while in the case of ekNN the best performance saturates or stabilises as h goes higher. This observation leads us to conclude that making better use of the same set of neighbours can give rise to better performance.

8 Summary and conclusion

Based on the conceptual framework of Dempster-Shafer theory and TBM, a new non-parametric classifier has been proposed. To classify a pattern, this method considers several neighbourhoods, each of which is a set of neighbours. These neighbourhoods are taken as a single source evidence supporting propositions about the class membership of that pattern. This evidence is represented as a single mass function in order to quantify the uncertainty attached to the class membership of that pattern. This classifier is shown to be an extension of the standard voting based kNN.

This classifier is different from the TBM classifier in that it does not use the time consuming Dempster Rule of Combination to aggregate mass functions for classification.

On a theoretical side we discussed an optimal classifier based on pignistic probability. It was shown to be equivalent to the optimal Bayes classifier. Our kNN classifier (Rule 2) is an approximation of the optimal classifier.

In experiments using real world datasets this classifier outperformed on average the voting kNN and distance weighted kNN, and performed as well as or better than the voting kNN with *best* k values. Its performance became stable when the number of neighbourhoods considered is greater than 3. Such a property is useful since it relieves

the designer of a kNN system from the burden of searching for the optimal number of neighbours.

References

- [1] Shafer, G. (1976) *A mathematical theory of evidence*. Princeton University Press, Princeton, New Jersey.
- [2] Denoeux, T. and Yaghlane, A. B. (2002) Approximating the combination of belief functions using the fast möbius transform in a coarsened frame. *International Journal of Approximate Reasoning*, **31**, 77–101.
- [3] Denoeux, T. (1995) A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, **25**, 804–813.
- [4] Smets, P. (1998) The transferable belief model for quantified belief representation. In Gabbay, D. M. and Smets, P. (eds.), *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Vol. 1*, pp. 267–301. Kluwer, Dordrecht, The Netherlands.
- [5] Smets, P. and Kennes, R. (1994) The transferable belief model. *Artificial Intelligence*, **66**, 191–234.
- [6] Zouhal, L. M. and Denoeux, T. (1998) An evidence-theoretic k-nn rule with parameter optimization. *IEEE Transactions on Systems, Man and Cybernetics*, **28**, 263–271.
- [7] Wilson, N. (2000) Algorithms for dempster-shafer theory. In Gabbay, D. M. and Smets, P. (eds.), *Handbook of defeasible reasoning and uncertainty management*, pp. 421–475. Kluwer Academic Publishers, Boston.
- [8] Bauer, M. (1997) Approximation algorithms and decision making in the dempster-shafer theory of evidence – an empirical study. *International Journal of Approximate Reasoning*, **17**, 217–237.
- [9] Denoeux, T. (2001) Inner and outer approximation of belief structures using a hierarchical clustering approach. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, **9**, 437–460.
- [10] Harmanec, D. (1999) Faithful approximations of belief functions. In Laskey, K. B. and Prade, H. (eds.), *Uncertainty in Artificial Intelligence 15 (UAI99)*, Stockholm, Sweden.

- [11] Lowrance, J. D., Garvey, T. D., and Strat, T. M. (1986) A framework for evidential reasoning systems. In et al, T. K. (ed.), *Proc. AAAI86*, Philadelphia, August, pp. 896–903. AAAI.
- [12] Tessem, B. (1993) Approximations for efficient computation in the theory of evidence. *Artificial Intelligence*, **61**, 315–329.
- [13] Fix, E. and Hodges, J. L. (1951) Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report Technical Report 4. USAF School of Aviation Medicine, Randolph Field, TX.
- [14] Cover, T. M. and Hart, P. E. (1967) Nearest neighbour pattern classification. *IEEE Trans. Inform. Theory*, **IT-13**, 21–27.
- [15] Dudani, S. A. (1976) The distance-weighted k-nearest-neighbor rule. *IEEE Trans. Syst. Man Cyber.*, **6**, 325–327.
- [16] Baily, T. and Jain, A. K. (1978) A note on distance-weighted *k*-nearest neighbor rules. *IEEE Trans. Syst. Man Cyber.*, **8**, 311–313.
- [17] Dasarathy, B. V. (ed.) (1991) *Nearest Neighbor(NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, California.
- [18] Morin, R. L. and Raeside, D. E. (1981) A reappraisal of distance-weighted *k*-nearest neighbor classification for pattern recognition with missing data. *IEEE Trans. Syst. Man Cyber.*, **11**, 241–243.
- [19] Bennett, P., Dumais, S., and Horvitz, E. (2002) Probabilistic combination of text classifiers using reliability indicators: Models and results. *Proceedings of the 25th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR02)*, pp. 207–214.
- [20] Bennett, P., Dumais, S., and Horvitz, E. (2003) The combination of text classifiers using reliability indicators. Technical report. Microsoft. To appear in *Information Retrieval*, http://research.microsoft.com/~horvitz/tclass_combine.htm.
- [21] Kittler, J. and (editors), F. R. (2000) *Multiple Classifier Systems, Proc. of 1st International Workshop, MCS2000, Cagliari, Italy, 21-23 June 2000*. Lecture Notes in Computer Science, vol 1857, Springer-Verlag, Berlin.
- [22] Kittler, J. and (editors), F. R. (2001) *Multiple Classifier Systems, Proc. of 2nd International Workshop, MCS2001, Cambridge, UK, 2-4 July 2001*. Lecture Notes in Computer Science, vol 2096, Springer-Verlag, Berlin.

- [23] Kittler, J., Hatel, J., Duin, R., and Matas, J. (1998) On combining classifiers. *IEEE PAMI*, **20**, 226–239.
- [24] Xu, L., Krzyzak, A., and Suen, C. (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. on Systems, Man, and Cybernetics*, **22**, 418–435.
- [25] Larkey, L. and Croft, W. (1996) Combining classifiers in text categorization. *SIGIR'96, Proceedings of the 19th Annual International ACM Conference on Research and Development in Information Retrieval*, pp. 289–297.
- [26] Hull, D., Pedersen, J., and Schuetze, H. (1996) Method combination for document filtering. *SIGIR'96, Proceedings of the 19th Annual International ACM Conference on Research and Development in Information Retrieval*, pp. 279–287.
- [27] Yang, Y., Ault, T., and T. T. P. (2000) Combining multiple learning strategies for effective cross validation. *ICML'00, Proceedings of the 17th International Conference on Machine Learning*, pp. 1167–1182.
- [28] Li, Y. and Jain, A. (1998) Classification of text documents. *The Computer Journal*, **41**, 537–546.
- [29] Lam, W. and Lai, K. (2001) A meta-learning approach for text categorization. *SIGIR'01, Proceedings of the 24th Annual International ACM Conference on Research and Development in Information Retrieval*, pp. 303–309.
- [30] Toyama, K. and Horvitz, E. (2000) Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. *ACCV'00, Proceedings of the 4th Asian Conference on Computer Vision*.
- [31] Smets, P. (1990) The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 447–458.
- [32] Macleod, J. E., Luk, A., and Titterton, D. M. (1987) A re-examination of the distance weighted k-nearest neighbor classification rule. *IEEE Trans. Syst. Man Cyber.*, **17**, 689–696.
- [33] Hand, D., Mannila, H., and Smyth, P. (2001) *Principles of Data Mining*. The MIT Press.
- [34] Buxton, B. F., Langdon, W. B., and Barrett, S. J. (October 2001) Data fusion by intelligent classifier combination. *Measurement and Control*, **34**, 229–234.
- [35] Piatetsky-Shapiro, G. Poll: What data mining techniques you use regularly? <http://www.kdnuggets.com/news/2003/n22/1i.html>.

[36] Chang, C.-C. and Lin, C.-J. Libsvm – a library for support vector machines. Technical report. National Taiwan University. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>.

Dataset	#Attribute	#Example	#Class
Anneal	38	798	6
Australian	14	690	2
Auto	25	205	6
Diabetes	8	768	2
Glass	9	214	3
Heart	13	270	2
Hepatitis	19	155	2
Horse-Colic	22	368	2
Ionosphere	34	351	2
Iris	4	150	3
Sonar	60	208	2
Vehicle	18	846	4
Wine	13	178	3

Table 1: General information about the datasets.

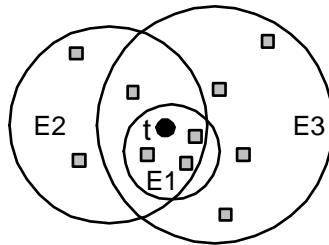


Figure 1: Three neighbourhoods of a pattern t – the black dot. The region delineated by each circle represents one neighbourhood. Each grey coloured rectangle is a neighbour of t . Neighbourhood E_1 covers three neighbours, E_2 covers six, and E_3 covers eight. The total number of distinct neighbours is 10.

Dataset	SVM	C4.5
Anneal	91.32	89.80
Australian	85.29	85.20
Auto	68.29	68.70
Diabetes	77.65	72.90
Glass	81.90	63.90
Heart	83.33	77.10
Hepatitis	80.65	80.70
Horse-Colic	83.01	80.90
Ionosphere	87.14	84.50
Iris	94.00	94.00
Sonar	72.20	69.40
Vehicle	76.80	67.90
Wine	94.29	91.00

Table 2: Classification accuracy by SVM and C4.5 as a reference. The C4.5 is the module in the SPSS-Clementine package, and the default parameters were used. SVM is an implementation by Chih-Chung Chang and Chih-Jen Lin [36]. The command line used was “-s 0 -t 0 -c 1 -v 5”, which means the type of SVM is C-SVC (C-support vector classification), the kernel function is linear, and the cost upper bound is 1.

Data	Min		Max		Avg. Accu	Samples				
	Accu.	k	Accu.	k		$k = 1$	$k = 3$	$k = 5$	$k = 7$	$k = 9$
anneal	77.88	10	83.83	1	80.01	83.83	81.33	79.20	78.63	78.51
australian	83.04	2	85.48	10	84.62	83.99	84.71	84.61	84.61	85.41
auto	42.44	7	59.02	2	48.63	58.05	52.68	48.29	42.44	43.41
colic	79.64	7	82.89	2	80.84	81.80	81.27	79.91	79.64	80.74
diabetes	71.73	1	74.22	2	72.61	71.73	73.57	72.52	72.00	72.78
glass	76.18	1	79.46	8	78.52	76.18	78.53	78.98	78.99	79.46
heart	71.11	10	76.30	4	74.26	74.07	75.19	74.81	74.81	73.33
hepatitis	78.71	1	79.35	2	79.29	78.71	79.35	79.35	79.35	79.35
iono	76.93	10	84.63	1	79.10	84.63	78.93	78.36	78.64	77.50
iris	93.33	1	96.00	3	95.33	93.33	96.00	96.00	95.33	95.33
sonar	65.89	10	72.08	1	68.62	72.08	69.22	68.29	68.29	65.90
vehicle	51.71	9	55.32	2	53.32	55.02	54.90	53.07	52.54	51.71
wine	89.29	3	92.65	1	91.04	92.65	89.29	90.43	90.41	91.56
Average	73.68	6	78.56	3	75.86	77.39	76.54	75.68	75.05	75.00

Table 3: The classification accuracy of standard voting based kNN .

Data	Min		Max		Avg. Accu	Samples				
	Accu.	h	Accu.	h		$h = 1$	$h = 3$	$h = 5$	$h = 7$	$h = 9$
anneal	83.71	9	83.96	5	83.84	83.83	83.83	83.96	83.90	83.71
australian	83.77	2	85.29	4	84.88	83.99	84.57	85.29	85.22	85.14
auto	58.05	1	61.95	10	60.15	58.05	59.51	60.00	60.98	61.46
colic	81.80	1	82.63	6	82.44	81.80	82.35	82.35	82.63	82.63
diabetes	71.73	1	75.12	4	74.43	71.73	74.86	74.73	74.73	74.86
glass	76.18	1	78.52	2	78.00	76.18	78.05	78.05	78.05	78.52
heart	74.07	1	75.56	2	75.08	74.07	74.81	75.19	75.19	75.19
hepatitis	78.71	1	79.35	2	79.29	78.71	79.35	79.35	79.35	79.35
iono	84.63	1	85.77	2	85.48	84.63	85.77	85.77	85.48	85.48
iris	93.33	1	96.00	3	95.67	93.33	96.00	96.00	96.00	96.00
sonar	72.08	1	76.43	4	75.66	72.08	75.47	76.43	76.43	76.43
vehicle	55.02	1	56.91	4	56.57	55.02	56.80	56.80	56.68	56.74
wine	92.65	1	93.78	4	93.21	92.65	93.21	93.78	93.21	93.21
Average	77.36	2	79.33	4	78.82	77.39	78.81	79.05	79.07	79.13

Table 4: The classification accuracy of our extended kNN classifier – $ekNN$.

Data	Min		Max		Avg. Accu	Samples				
	Accu.	k	Accu.	k		$k = 1$	$k = 3$	$k = 5$	$k = 7$	$k = 9$
anneal	78	10	84.71	1	80.491	84.71	81.76	79.45	79.32	79.19
australian	83.62	2	86.09	4	85.354	85.65	85.8	85.63	85.19	85.77
auto	44.88	8	61.95	1	51.218	61.95	55.12	50.73	46.34	45.85
colic	79.92	8	83.15	2	81.108	81.26	81.81	81	80.46	80.47
diabetes	72.65	5	75.52	2	73.657	72.91	74.74	72.65	73.43	74.09
glass	76.66	1	79.46	9	78.666	76.66	78.53	78.99	78.99	79.46
heart	71.11	10	76.3	4	74.445	74.81	75.19	75.19	75.19	73.33
hepatitis	78.71	1	79.35	10	79.286	78.71	79.35	79.35	79.35	79.35
iono	76.65	10	85.77	1	79.615	85.77	79.78	79.22	79.22	77.51
iris	94.67	1	96	5	95.465	94.67	96	96	95.33	95.33
sonar	66.36	10	76.91	1	70.344	76.91	73.05	69.25	69.25	67.34
vehicle	51.77	9	56.97	1	53.883	56.97	55.67	53.72	52.72	51.77
wine	89.87	8	93.24	1	91.1	93.24	90.98	89.87	90.41	91
Average	74.22	6	79.65	3	76.51	78.79	77.52	76.23	75.78	75.42

Table 5: The classification accuracy of the distance weighted kNN – $wkNN$.

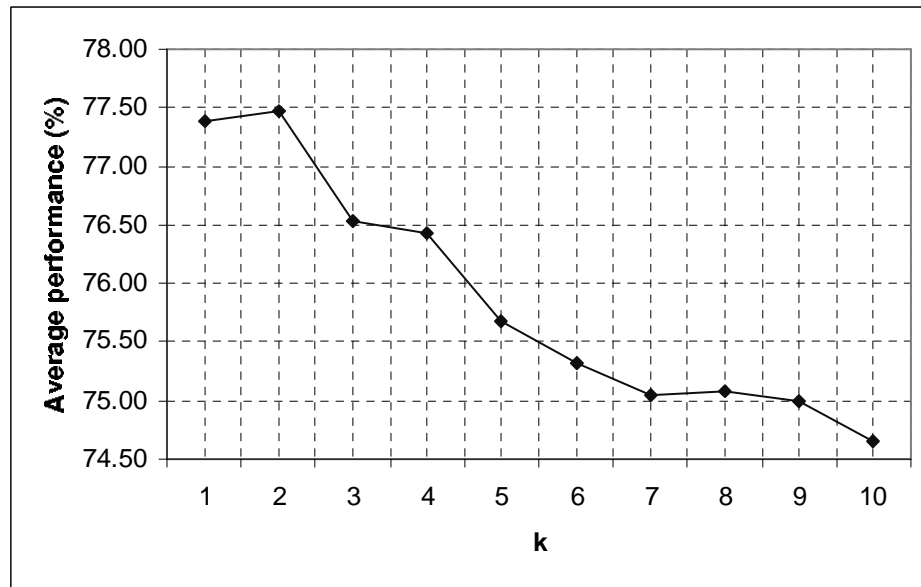


Figure 2: Average performance of kNN over all datasets as a function of k .

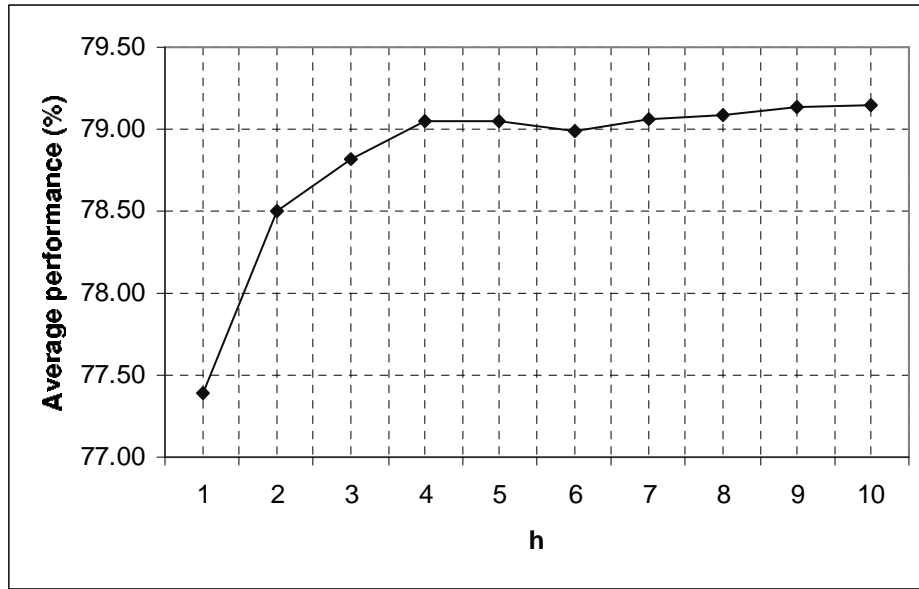


Figure 3: Average performance of $ekNN$ over all datasets as a function of k .

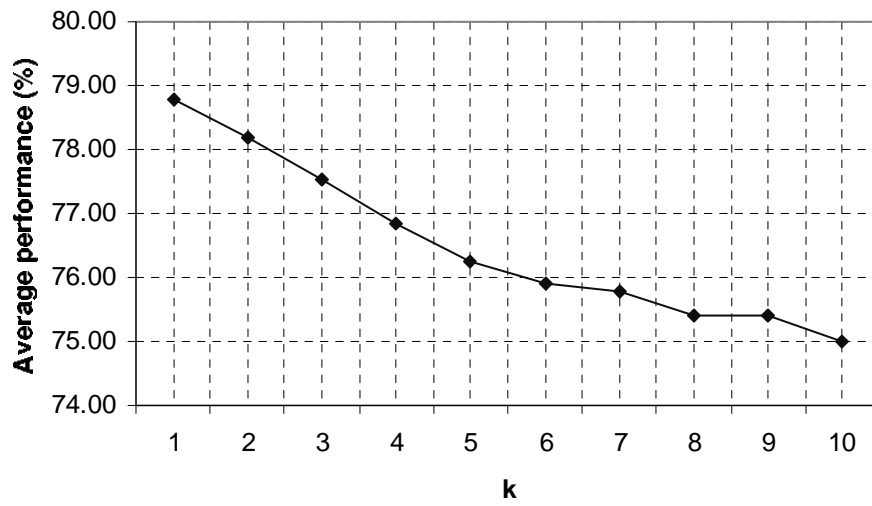


Figure 4: Average performance of $wkNN$ over all datasets as a function of k .