ISSN 1346-5597

National Institute of Informatics

NII Technical Report

# Adaptive Classification Using Shared-Neighbor Information

Michael E. Houle and Michael Nett

NII-2009-016E
Dec. 2009

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

# Adaptive Classification Using Shared-Neighbor Information

Michael E. Houle
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan
*meh@nii.ac.jp*

Michael Nett
RWTH Aachen University
Templergraben 55
52056 Aachen, Germany
*michael.nett@rwth-aachen.de*

December 22, 2009

**Abstract**

Nearest-neighbor approaches for classification have long been recognized for their potential in achieving low error rates, despite their perceived lack of scalability. Recent advances in the efficient computation of approximate $k$-nearest neighborhoods have made the nearest-neighbor approaches more affordable in practice. However, their effectiveness is still limited due to their sensitivity to noise and to the choice of neighborhood size $k$. In this paper, we propose a general-purpose method for nearest-neighbor classification that seeks to compensate for the effects of noise through the determination of natural clusters in the vicinity of the test item. The classification model, based on elements of the relevant-set correlation (RSC) model for clustering, also allows for the automatic determination of an appropriate value of $k$ for each test item. We also provide experimental results that demonstrate the competitiveness of our approach with that of other popular classification methods.

## 1 Introduction

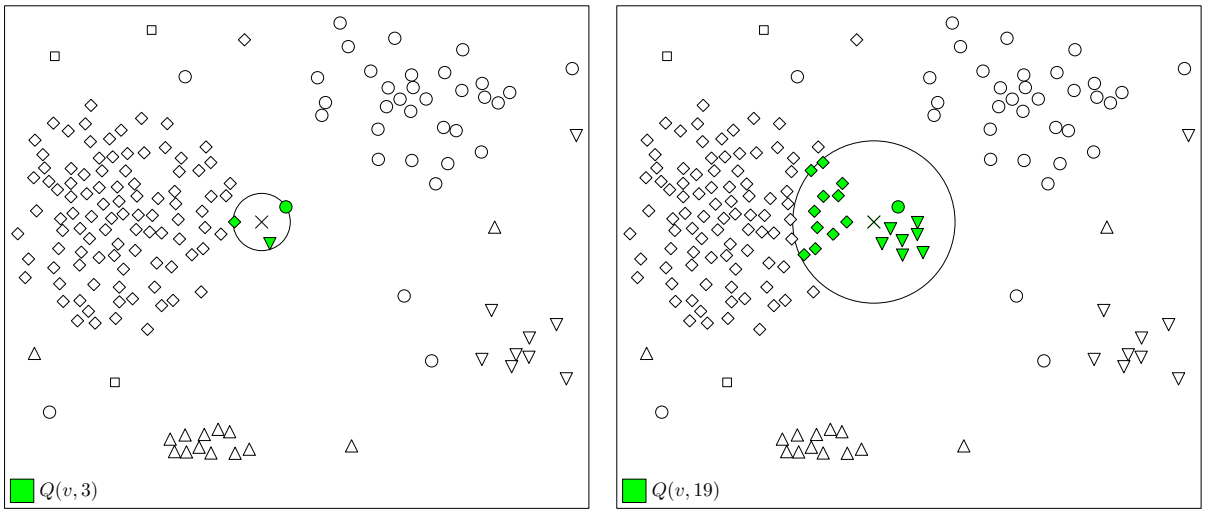
An important step in the data mining and knowledge discovery process involves the categorization of data objects. Medical diagnosis, credit and insurance risk assessment, trend analysis and many other tasks often require the construction of a model that is able to efficiently predict class labels for items drawn from very large data repositories.

Many approaches to supervised learning exist. Among the most well-known are those of *decision tree induction* [9, 14], *Bayesian classification* [9], *Support vector machines* (SVM) [17, 1] as well as hybrid methods [20, 7].

One of the earliest known methods for supervised learning, and perhaps the simplest, is that of *$k$-nearest-neighbor* ($k$-NN) classification. In contrast to the aforementioned approaches, which all attempt to infer decision rules by which the class labels of test items can be predicted, the $k$-NN strategy uses the $k$ most similar examples from the training set to suggest a class label for a test item. Similarity is typically measured according to a distance function defined on the data domain.

The easiest way of generating a class prediction is by a simple majority vote over the class labels of the $k$ examples [9], although more complex schemes have also been proposed: distance-weighted and rank-weighted methods, for example, reward or penalize classes according to the distance or rank of their training items to the test item.

Nearest-neighbor classification is generally regarded as a method capable of producing competitively-low error rates in practice. It has been shown to be 'asymptotically optimal', in that the probability of error of the nearest neighbor rule is bounded above by twice the




(a) If $k$ is too small, the classifier is susceptible to noise. Majority voting is assumed, together with the Euclidean distance.

(b) When $k$ is too large, small but relevant example subsets can be outvoted by larger but less relevant example subsets.

Figure 1: Possible shortcomings of a fixed choice of $k$.

Bayes minimum probability of error, as the training set size tends to infinity [6, 15]. In this sense, an infinite sample set can be regarded as containing half the classification information in the nearest neighbor.

Despite their advantages in quality, for large training set sizes and large feature set sizes, practitioners tend to favor other methods over $k$-NN classification due to the perceived high cost of similarity search. Evidence suggests that when the representational dimension of feature vectors is high (greater than 20 or so), an exact similarity search accesses an unacceptably-high proportion of the data elements, unless the underlying data distribution has special properties, such as a low fractal dimension or low intrinsic dimension [5]. The effect is particularly disastrous for multimedia data, where image representations often reach into the hundreds of dimensions, and text vectors typically span thousands to even millions of keyword dimensions – although only hundreds of these keywords may actually appear together in any document. However, recently developed techniques for approximate similarity search, such as locality-sensitive hashing (LSH) [12] and the SASH search index [11], are often able to provide $k$-NN classifiers with very accurate approximate neighbor lists several orders of magnitude faster than sequential search.

Another disadvantage of $k$-nearest neighbor classification concerns the choice of parameter $k$. If chosen too small, the neighborhood of the test item could be dominated by noise that can affect the prediction (see Figure 1a). If, on the other hand, $k$ is chosen too large, a small but relevant subset of examples in the neighborhood could be overruled in the voting by a larger but less relevant subset (see Figure 1b). Generally speaking, there is no fixed value of $k$ that prevents these two effects from manifesting themselves during the testing phase. Overcoming this problem thus requires additional information concerning the local distribution of data in the vicinity of the test item.

One way of gathering additional information on the training set distribution is through the use of unsupervised learning techniques such as clustering. The use of clustering to support classification is not new: pre-clustering has been employed for simplification of the training set, by replacing training examples with cluster representatives [13, 16]. However, a global clustering

is not likely to provide information that can resolve the difficulties surrounding the choice of neighborhood size $k$ for every test item. Still, the use of clustering techniques during the testing phase, applied in the vicinity of each test item, has the potential of providing information that can support the determination of an appropriate set of local training examples.

Other ways to utilize information on the local distribution of training items for nearest neighbor classification have been proposed in the past. Methods based on statistical confidence [18] can be used to dynamically choose a value of $k$ for applications where confidence is crucial. *Locally adaptive metrics* [7] aim to increase the expressiveness of neighborhoods by constricting them along more relevant dimensions, while elongating them along less relevant ones.

In this paper, we present a scheme for 'adaptive' nearest neighbor classification that uses unsupervised techniques for flexibly determining the size and composition of example sets from among the training items in the vicinity of the test item. The three main contributions of the paper are:

1. A new measure of cohesion of the training set items in the vicinity of the test item. The *mutual relevance measure* assesses the proportion of training items in the set of $k$-nearest neighbors that would include the test item as one of their own $k$-nearest neighbors. We argue that such examples are more likely to provide accurate class recommendations than training items that do not have the test item in their own vicinity. The measure is used in the determination of a value of $k$ appropriate for a given test item.

2. We adapt to local variations in the distribution of the training set through a selection of training items from a given $k$-nearest neighborhood. Members of the neighborhood set that are deemed to be poorly-associated with the remainder of the set are replaced by items from outside the set having a stronger level of association. The strength of the association is assessed according to a neighborhood correlation criterion proposed under the RSC model for data clustering [10].

3. We employ a modified majority voting scheme that compensates for extremes of variation in the class sizes. The vote for each class is normalized according to the expected proportion of class members within the neighborhood.

The remainder of the paper is organized as follows. Sections 3, 4 and 5 explain the aforementioned techniques in greater detail. Section 6.1 details the implementation of the proposed techniques while Section 6 presents experimental results and compares the performance of our approach with other well-known classification methods.

## 2 Preliminaries

Let $S$ be a data set of size $m$ drawn from some domain $\mathcal{U}$. Let RANK be a ranking function that for each item $v \in S$ induces a unique ordering $Q(v) = (v_1, v_2, \ldots, v_{m-1})$ of the items of $S \setminus \{v\}$, where $i < j$ implies that $v_i$ is deemed more 'relevant' or 'similar' to $v$ than $v_j$. Given any non-empty subset $T \subseteq S$ of size $n$, and any choice of $v \in S$, the ranking function also induces a collection of *relevant sets* for $v$ with respect to $T$, defined as follows: for any choice of $1 \leq k \leq n$, the top-$k$ relevant set $Q_T(v, k)$ consists of the first $k$ members of $T$ appearing in the list $Q(v)$. If the ranking function RANK is consistent with the ordering produced using some distance function $\text{DIST} : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}_{\geq 0}$, the notions of relevant sets and $k$-nearest neighborhoods coincide. In this paper, we will not explicitly require that RANK be based upon some distance function.

The notion of relevancy, while it is not symmetric, does admit an inverse relationship: if $w$ is a member of the top-$k$ relevant set of $v$, then $v$ can also be viewed as relevant to $w$. For every item $v \in S$, let $Q_T^{-1}(v,k)$ contain all items $w \in T$ for which $v$ is among their top-$k$ relevant items with respect to $T \cup \{v\} \setminus \{w\}$:

$$Q_T^{-1}(v,k) = \{w \in T \mid v \in Q_{T\cup\{v\}\setminus\{w\}}(w,k)\}.$$

In order to distinguish between these complementary forms of relevance information, we will refer to $Q_T(v,k)$ as the top-$k$ *forward* relevant set of $v$ with respect to $T$, and $Q_T^{-1}(v,k)$ as its corresponding top-$k$ *reverse* relevant set. It should be noted that whereas the size of $Q_T(v,k)$ is always equal to $k$, the same is not necessarily true of $Q_T^{-1}(v,k)$. Henceforth, when the set $T$ is understood, we will make use of the simplified notation $Q(v,k)$ and $Q^{-1}(v,k)$ for forward and reverse relevant sets, respectively.

## 3 Mutual Relevancy

Let us consider the situation in which test item $v \in S \setminus T$ is to be classified by means of a voting scheme on the top-$k$ relevant set $Q(v,k)$, where the value $k$ is as yet to be determined. If the relevance information provided by the underlying ranking function is of sufficiently high quality, we would expect that the smallest cluster of training set items containing $v$ would serve as the most natural candidate examples for predicting the class of $v$. One of the characteristics of a well-formed, compact cluster is that it exhibits a high degree of internal association, and is well-differentiated from other clusters. In our situation, where the only information assumed available is relevance information, the degree of internal association is reflected in the extent to which the relevant sets of its items agree.

Consider now the effect of choosing $k$ either too small, or too large, relative to the size of the smallest natural cluster of $T \cup \{v\}$ to which $v$ belongs. If $k$ is too small, many training examples in the relevant set $Q(v,k)$ can be expected to contain other members of the cluster in preference to $v$. If $k$ greatly exceeds the cluster size, then $Q(v,k)$ would contain many examples from outside the cluster, relatively few of which would in turn contain $v$ in their top-$k$ relevant sets. Instead, if the cluster is well-differentiated from the remainder of the training set, a choice of $k$ approximately equal to the cluster size would be most likely to result in $v$ being recognized as a neighbor of many members of $Q(v,k)$.

With this motivation, we propose the following *mutual relevance measure* as a guide to the determination of an appropriate value of $k$:

$$\textsc{m}(v,k) = \frac{\left|Q(v,k) \cap Q^{-1}(v,k)\right|}{k},$$

where $1 \leq k < n$, and $v \in S \setminus T$. Note that the measure achieves its maximum value 1 only when each neighbor $w \in Q(v,k)$ contains $v$ in its own relevant set $Q_{T\cup\{v\}\setminus\{w\}}(w,k)$.

At first glance, it may seem sufficient to use the mutual relevance measure to determine a neighborhood size for classification, by simply letting $k$ vary over a sufficiently large range and reporting the value for which $\textsc{m}(v,k)$ attains its maximum. However, the measure is clearly biased in terms of $k$. Whenever $k \geq \lceil \frac{n}{2} \rceil$, the forward and reverse relevant sets $Q(v,k)$ and $Q^{-1}(v,k)$ are guaranteed to intersect. As $k$ approaches $n$, these sets converge in membership, implying that $\textsc{m}(v,k)$ tends to 1.

In order to correct for the bias with respect to $k$, we employ a form of normalization first introduced in the context of the RSC model for data clustering [10], which also makes exclusive

use of relevant set information in the assessment of the quality of cluster candidates. As with RSC, we consider the hypothetical situation in which every relevant set $Q(v,k)$ is selected uniformly at random from among the members of $T$. In other words, we assume a hypothesis of *randomness*, in which the ranking function provides zero information. Even in this situation, the expected numbers of intersections between $Q(v,k)$ and $Q^{-1}(v,k)$, and the variances of these numbers, are not zero. These expectations and variances can be used to generate standard scores (also known as Z-scores [9]), which count the number of standard deviations the observed value of $\mathrm{M}(v,k)$ exceeds the value it would attain if the relevance sets contained zero information. These normalized, unitless Z-scores constitute a measure of the statistical significance of the mutual relevance relationship, and can be compared meaningfully across different values of $k$.

For any test item $v \in S \setminus T$, let $\underline{Q}(v,k)$ be a set of $k$ items selected uniformly at random from $T$, under the assumption of randomness. Let $X_w \in \{0,1\}$ be a binary random variable attaining the value 1 if and only if item $w \in T$ is contained in the forward relevant set $\underline{Q}(v,k)$. Similarly, let $Y_w \in \{0,1\}$ be a binary random variable attaining the value 1 if and only if item $v \in T$ is contained in the forward relevant set $Q_{T\cup\{v\}\setminus\{w\}}(w,k)$. Note that the random selection of the forward relevant sets based at training examples immediately determines the membership of all reverse relevant sets. Finally, let $\underline{\mathrm{M}}(v,k)$ denote the value of $\mathrm{M}(v,k)$ attained when the relevant sets are selected randomly. Then

$$\underline{\mathrm{M}}(v,k) = \frac{1}{k}\sum_{w\in T} X_w Y_w.$$

The probabilities of $X_w = 1$ and $Y_w = 1$ are each $p = \frac{k}{n}$, the proportion of training items in the relevant sets. Therefore the probability that an item $w \in T$ contributes to the mutual relevance measure is $p^2$. Given a fixed choice of $k$, we can determine the expectation and variance of the measure as follows: the expectation is

$$\mathsf{E}[\underline{\mathrm{M}}(k)] = \frac{1}{k}\sum_{w\in T} \mathsf{E}[X_w Y_w] \;=\; p$$

and the variance is

$$\mathsf{Var}[\underline{\mathrm{M}}(k)] = \frac{1}{k^2}\sum_{w\in T} \mathsf{Var}\left[X_w Y_w\right] \;=\; \frac{1}{n}(1-p^2)$$

due to the independence of the selection of relevant sets.

At this point we define the *normalized mutual relevance measure* as the standard score of the mutual relevance measure when measured against the randomness assumption:

$$\mathrm{M}^*(v,k) = \frac{\mathrm{M}(v,k) - \mathsf{E}[\underline{\mathrm{M}}(k)]}{\sqrt{\mathsf{Var}[\underline{\mathrm{M}}(k)]}} = \sqrt{n}\cdot\frac{\mathrm{M}(v,k) - k/n}{\sqrt{1-(k/n)^2}}.$$

It can be shown that $\mathrm{M}^*$ satisfies

$$\mathrm{M}^*(v,k) \le \sqrt{n}\cdot\sqrt{\frac{n-k}{n+k}}$$

for all $1 \le k < n$; this bound tends to 0 as $k \to n$, in accordance with our intuition. Also, we note that the use of $\mathrm{M}^*$ does not require any explicit assumptions on the distribution of the training set items themselves.

To adaptively determine a neighborhood size, we simply use the value $k_{\max}$ of $k$ maximizing $\mathrm{M}^*(v,k)$ — that is, the value of $v$ for which $\mathrm{M}^*(v,k)$ is most significant. In practice, for the

sake of efficiency, the range of values searched can be restricted to some large constant upper limit $K < n$. $k_{\max}$ can be determined algorithmically by a straightforward search for a global maximum, provided that sufficiently-large relevant sets have been precomputed.

**Algorithm 1** ADAPTIVE_K*(*item $v \in S \setminus T$*)*

1. *Set* $max = -\infty$ *and* $size = -1$.
2. *For* $k \in \{2, \ldots, K\}$ *do*
   (a) *Set* $score = \mathrm{M}^*(v, k)$.
   (b) *If* $score > max$ *set* $max = score$ *and* $size = k$.
3. *Return* $size$.

# 4 Neighborhood Reshaping

Suppose we have already decided upon the number $k$ of training examples that we wish to consider for the classification of a test item $v$. Ordinarily, one would make use of the set $Q(v, k)$ returned by the ranking function. In some situations, however, these top-$k$ relevant sets might not conform with the distribution of the smallest cluster of training items to which $v$ can be joined. This can result in increased numbers of noise items, or items from other natural clusters, appearing in the relevant set $Q(v, k)$. When a distance function is used for ranking, these problems generally arise when the cluster shape deviates markedly from spherical.

Instead of conducting a straightforward majority vote over the class labels present in $A = Q(v, k)$, we propose a method for analyzing and enhancing the quality of $A$ as a potential cluster in the vicinity of $v$, with respect to the training set. The method employs a cluster reshaping operation first proposed in the context of the RSC clustering model [10]. The RSC model is able to assess the quality of cluster candidates, the degree of association between pairs of cluster candidates, and the degree of association between clusters and data items, all according to the statistical significance of a form of correlation among pairs of relevant sets and/or candidate cluster sets. The RSC significance measures can be used to evaluate the relative importance of cluster candidates of various sizes, avoiding the problems of bias found with other shared-neighbor methods that use fixed neighborhood sizes. Here, we will present only those aspects of the RSC model that are relevant to our scenario, adapted to the context of classification. More detailed explanations and complete derivations can be found in [10].

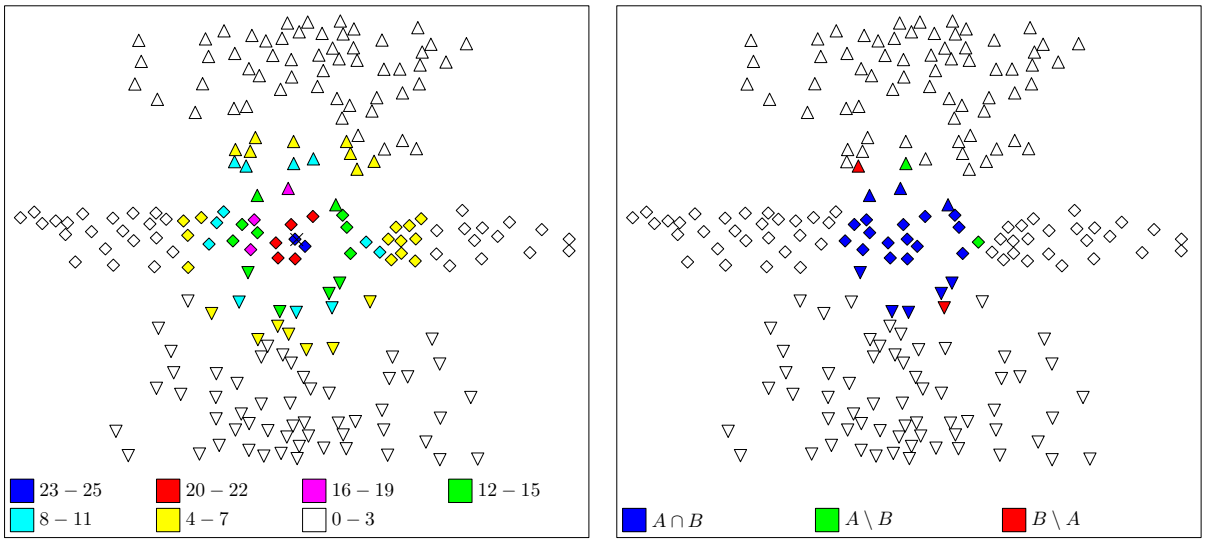
Consider two arbitrary subsets $A$ and $B$ of a data set (in our case, the training set $T$). The similarity of $A$ and $B$ can be determined strictly according to their memberships alone, using a form of Pearson correlation [9] on the zero-one set membership vectors of $A$ and $B$. The *inter-set correlation* between $A$ and $B$ is given by

$$\mathrm{R}(A, B) = \frac{n}{\sqrt{(n - |A|)(n - |B|)}} \left( \frac{|A \cap B|}{\sqrt{|A| \cdot |B|}} - \frac{\sqrt{|A| \cdot |B|}}{n} \right).$$

The degree of internal cohesion of a set $A$ is assessed according to the (first order) *intra-set correlation measure*, defined as the average correlation between $A$ and the relevant sets of size $|A|$ based at its members:

$$\mathrm{SR}(A) = \frac{1}{|A|} \sum_{v \in A} \mathrm{R}(A, Q(v, |A|)).$$

A value of 1 indicates perfect agreement among the items of $A$, whereas small values indicate a lack of cohesion.




(a) Item ranked according to their partial significance with respect to the top-25 items $A$. The values shown are of $|A \cap Q(v, |A|)|$.

(b) Reshaping the set $A$ of the 25 most relevant items yields a set $B$ which better matches the patterns observed in $A$.

Figure 2: Using partial significances to reshape neighborhoods.

To make comparisons of intra-set correlation values for sets of different sizes meaningful, we must first eliminate bias due to the size of $A$. This bias exists due to the nature of the Pearson correlation, in that a given value of the correlation is more significant if it is achieved for a larger set than for one that is smaller. As in the previous section, RSC normalizes the correlation measure by generating standard scores under the assumption of randomness of relevance information. The statistical significance of the intra-set correlation is derived as the standard score

$$Z(A) = \sqrt{|A|(n-1)} \cdot \text{SR}(A).$$

Not all elements of $A$ contribute equally to the above formulation of significance. In particular, some elements may have relevant sets which strongly correlated with $A$, and others may have relevant sets for which the correlation is relatively small. The RSC model provides a mechanism by which items of $A$ with small contributions to the significance $Z(A)$ can be replaced by items outside $A$ that are more strongly correlated with $A$.

The contribution to $\text{SR}(A)$ attributable to $v \in A$ is given by

$$\text{SR}(A) = \sum_{v \in A} t(v|A), \quad \text{where} \quad t(v|A) = \frac{1}{|A|} \text{R}(A, Q(v, |A|)).$$

The significance values themselves can be expressed as the sum of partial significances:

$$Z(A) = \frac{1}{\sqrt{|A|}} \sum_{v \in A} Z(v|A), \quad \text{where} \quad Z(v|A) = \sqrt{n-1} \cdot \text{R}(A, Q(v, |A|)).$$

Since the training set size $n$ does not depend on $A$, comparisons involving $Z(v|A)$ can be performed using $\text{R}(A, Q(v, |A|))$ alone.

Partial significances can be directly used to rank the items of $A$ according to their level of association with $A$, much like the items of a relevant set are ranked with respect to an individual

query item (see Figure 2a). The ranking can be extended to all items of $T$, as the definitions of partial significance is meaningful regardless of whether $v$ is actually a member of $A$. Under the RSC model the intra-set correlation of a set $B$ as the representative for the concept underlying a set $A$ is given by:

$$\text{SR}(B|A) = \frac{1}{|B|} \sum_{v \in B} \text{R}(A, Q(v, |A|)),$$

which in turn yields the following significance formula:

$$Z(B|A) = \frac{1}{\sqrt{|B|}} \sum_{v \in B} Z(v|A).$$

This latter equation suggests that the set $B$ consisting of the $|A|$ items with the highest partial significances with respect to $A$ achieves the highest quality $Z(B|A)$. This implies that we can 'reshape' a given set $A$ by replacing poorly-associated items of $A$ with well-associated items of $T \setminus A$, to yield a set $B$ with more significant internal association (see Figure 2b).

Applying the RSC significance formula for $Z(B|A)$ to the problem of selecting training examples for $k$-NN classification, we implemented the following algorithm. Since $k$ is assumed to be fixed here, ordering examples according to their significance scores $Z(x|A)$ turns out to be equivalent to ordering them according to the overlap sizes $|A \cap Q(x, |A|)|$. Another possible variant (not implemented in this study) would allow reshaping over a range of values of $k$; in this case, the maximizing of the actual significance scores $Z(B|A)$ is indicated. For efficiency, the algorithm evaluates only those training items $x$ for which $|A \cap Q(x, |A|)|$ is positive.

**Algorithm 2** RESHAPE*(*subset $A$ of $T$*)*

1. *For each $w \in Q(v, |A|)$, and for each $x \in Q^{-1}(w, |A|)$, set score$(x) = |A \cap Q(x, |A|)|$. Let $X$ be the set of training items for which a score has been set.*
2. *Sort the items of $X$ to obtain the list $(x_1, \ldots, x_u)$, such that score$(x_i) \geq$ score$(x_j)$ for all $1 \leq i < j \leq u$.*
3. *Return $\{x_1, \ldots, x_{|A|}\}$.*

# 5 Voting

Given a list of the top-$k$ training items deemed most relevant to the test item, our voting mechanism determines a prefix of the list eligible to participate in the voting.

Let $A = A_k = \{v_1, v_2, \ldots, v_k\}$ be a set of training items that serves as input for our voting algorithm, in increasing order of relevance. In order to allow the subsets $A_i = \{v_1, \ldots, v_i\}$ to be compared across different values of $i \in \{1, \ldots, k\}$, we again generate and compare standard scores, this time for the observed occurrences of a class with respect to some fixed set. One is likely to observe different classes in different regions and, since the classes are usually not of the same size, normalizing against the assumption that class labels are distributed uniformly at random does not provide a valid point of reference. Instead, in order to account for the classes' locality and their varying sizes, we exclude those classes that do not have at least one instance present in the set $A$. Let $T(A)$ represent the set of all training set items whose class label is shared by a member of $A$. For the remainder, we normalize against the assumption that the class labels are taken from items selected uniformly at random without replacement from $T$.

It is valid to measure against the assumption that a situation is the outcome of some random process only when the number of random variables involved is sufficiently large. Since a class

$C_i$ is represented in $T(A)$ with multiplicity $|C_i|$ or not at all, we expect $|T(A)| \gg |A|$ for most practical scenarios. If however the classes are small, this relationship may not hold and the voting may be prone to error. Whenever this situation is detected, a secondary voting method (such as simple majority voting) should be used.

Let $\mathrm{o}_{i,j}(A)$ be the number of observed items of class $C_j$ among the top-$i$ relevant items from $A$:

$$\mathrm{o}_{i,j}(A) = |\{v \in A_i \mid f(v) = C_j\}|\,.$$

The random vector $(\underline{\mathrm{o}}_{i,1}(A), \ldots, \underline{\mathrm{o}}_{i,m}(A))^T$ is hypergeometrically distributed. Thus the expected number of occurrences of a class $C_j$ in $A_i$ is

$$\mathsf{E}[\underline{\mathrm{o}}_{i,j}] = i\frac{|C_i|}{|T(A)|}$$

and has variance

$$\mathsf{Var}[\underline{\mathrm{o}}_{i,j}] = i\left(1 - \frac{|C_j|}{|T(A)|}\right)\frac{|T(A)| - i}{|T(A)| - 1} \cdot \frac{|C_j|}{|T(A)|}.$$

For the top-$i$ most relevant items of $A$ and a class $C_j$ we define the significance of $\mathrm{o}_{i,j}(A)$ as:

$$\mathrm{o}^*_{i,j}(A) = \frac{\mathrm{o}_{i,j}(A) - \mathsf{E}[\underline{\mathrm{o}}_{i,j}]}{\sqrt{\mathsf{Var}[\underline{\mathrm{o}}_{i,j}]}}.$$

Thus our method is described by the following algorithm:

**Algorithm 3** VOTE*(*subset $A$ of $T$*)*

1. *For $i \in \{1, \ldots, |A|\}$ and $j \in \{1, \ldots, m\}$ do*
   (a) *If $C_j$ has at least one member in $A_i$, set $m_{i,j} = \mathrm{o}^*_{i,j}(A)$.*
   (b) *Otherwise, set $m_{i,j} = -\infty$.*
2. *Set $max = -\infty$ and $winner = -1$.*
3. *For $i \in \{1, \ldots, |A|\}$ and $j \in \{1, \ldots, m\}$ do*
   (a) *If $m_{i,j} > max$ set $max = m_{i,j}$ and $winner = j$.*
4. *Return $C_{winner}$.*

# 6 Evaluation

## 6.1 Implementation

In order to improve the execution times the of our proposed algorithm, A$k$NN (Adaptive $k$-NN), we implement the techniques outlined in the three previous sections together with the following heuristic modifications:

- To avoid the $\mathcal{O}(|T|)$ cost of sequential search, we apply the SASH search index using default parameter settings [11] to obtain approximate $k$-nearest neighbor lists in time $\mathcal{O}(\log |T|)$.
- Forward and reverse relevant set sizes are restricted to a maximum of $\hat{k} = 20$.
- The reshaping does not consider all training items as candidates, but only training items from the forward relevant set $Q(v, \min\{|T|, 5\hat{k}\})$.

- The relevant sets $Q(v, \hat{k})$ are precomputed and cached for all $v \in T$.
- The intersections $|Q(v,k) \cap Q^{-1}(v,k)|$ were computed using distance comparisons, avoiding the explicit computation of reverse relevant sets.

Note that at training time, only one nearest neighbor query is required, namely that of $Q(v, \min\{|T|, 5\hat{k}\})$.

### 6.2 Data sets

We evaluated and tested our method, together with several competing strategies, on a variety of data sets. Table 1 describes the sizes, numbers of classes, type of features that are present in the sets used. We also state the proportion to which the approximate relevant sets, returned by the SASH index, resemble the exact relevant sets in average and standard deviation. Except where otherwise indicated, all sets in the table were drawn from the *UCI Machine Learning Repository* [2].

For the majority of the sets, two preprocessing steps were applied. Features unrelated to the task of classification, such as dates or timestamps, were deleted. Also, whenever no separate training set was supplied, we used uniform sampling to extract 80% of the items to serve as training items. Categorical features were handled by transformation to continuous coordinates, or interpreted as numerical values. Unknown or missing values were substituted by zero.

For the multiclassed *Reuters* newswire article data set, we used Porter stemming and constructed feature vectors with TF-IDF weighting. The classes were simplified by substituting general classes (such as the 'economy' subclass `ECAT`) for its subclasses (such as E1 and E2), leaving 57 distinct classes. Next, any item having more than one class label was eliminated from consideration, resulting in a data set of total size 554,651.

The Amsterdam Library of Object Images (ALOI) data set [8] was represented by dense 641-dimensional feature vectors based on color and texture histograms, the details of the preparation of which can be found in [3].

### 6.3 Methods

We now list the methods compared in our evaluation. Unless indicated otherwise, the implementations were in C++, using 3Gb of main memory.

- Our adaptive $k$-NN method, A$k$NN. We use cross-fold validation on random samples to determine which combination of adaptive techniques (dynamic determination of $k$, neighborhood reshaping and/or normalized voting) promises the highest accuracy on each data set. Whenever we do not use a dynamic choice of $k$ a value of $k = 5$ is chosen a-priori.
- The traditional $k$-nearest neighbor method ($k$-NN) was implemented in C++, using majority voting. Tied votes are broken in favor of the class whose closest item is nearest to the test item.
- $k$-nearest neighbor classification with votes weighted by the distance to the test item ($k$-NN-$\delta$). A distance of $\delta$ is accounted for with a weight of $1/\delta$.
- Support vector machines with radial basis function kernels (SVM) [4].
- Naïve bayes classifier (Bayes), implemented in Java. We use the WEKA implementation [19].
- A decision tree induction algorithm (J48), also implemented in Java. Again, the WEKA implementation is used [19].

| Data Set | Type | Size | Num Features | Num Classes | Accuracy |
|---|---|---|---|---|---|
| ALOI | numerical | 110,250 | 641 | 1,000 | 92.8%±20.9% |
| | Image histograms. Classes represent objects visible in the images. | | | | |
| Chess | numerical | 28,056 | 6 | 17 | 99.9%±0.3% |
| | Chess endgame database (King & Rook versus King). Classes represent the number of min-max optimal moves until checkmate. | | | | |
| Dorothea | numerical | 1,150 | 6,061 | 2 | 99.2%±8.7% |
| | Pharmaceutical data used in the NIPS 2003 feature selection contest. Classes consist of active and inactive compounds. | | | | |
| Gisette | numerical | 7,000 | 5,000 | 2 | 77.9%±29.6% |
| | Digit recognition data. Classes correspond to the digits 4 and 9. Also part of the NIPS 2003 contest. | | | | |
| Internet Ads | numerical | 3,279 | 1,558 | 2 | 99.2%±5.7% |
| | Images from Internet advertisements. Features contain image dimensions and phrases occurring in the URL of the image. | | | | |
| Isolet | numerical | 7,797 | 617 | 26 | 94.5%±17.2% |
| | Records of spoken letters. Features include spectral coefficients, contour and sonorant features. | | | | |
| Ozone Level | numerical | 5,070 | 70 | 2 | 97.6%±13.8% |
| | Ozone level prediction. Features contain wind and solar radiation measurements. Classes determine whether an alarming threshold is exceeded. | | | | |
| Pen Digits | numerical | 10,992 | 16 | 10 | 99.9%±3.0% |
| | Strokes of handwritten digits. Features represent the starting and ending points of the strokes. | | | | |
| Poker Hand | categorical | 1,025,010 | 10 | 10 | 97.6%±6.5% |
| | Hands of five poker cards. Classes represent the game value of the hand. 25,010 training examples are given. | | | | |
| Reuters | numerical | 554,651 | 57 | 320,648 | 46.7%±41.0% |
| | Text documents. Classes represent the genre of the document. | | | | |
| Spambase | numerical | 4,601 | 57 | 2 | 99.4%±6.0% |
| | E-Mail spam. Feature include the frequency of certain keywords. Classified as 'regular mail' or 'spam'. | | | | |
| Statlog Shuttle | numerical | 58,000 | 9 | 7 | 99.5%±4.8% |
| | Space shuttle flight data. | | | | |
| Vowel | numerical | 990 | 11 | 11 | 100.0%±0% |
| | Records of spoken vowels. | | | | |
| Wine | numerical | 178 | 13 | 3 | 100.0%±0% |
| | Chemical characteristics of wine. Classes represent cultivars of the wine samples. | | | | |

Table 1: Data sets used in the experimentation.

| Data Set | A$k$NN | $T_t$ | $T_c$ | SVM | $T_t$ | $T_c$ | Bayes |
|---|---|---|---|---|---|---|---|
| ALOI | 98.68% | 10679 | 1 | 44.15% | 31666 | 0.39 | N/A |
| Chess | 77.58% | 48 | 0.015 | 73.10% | 53 | 0.003 | 31.04% |
| Dorothea | 90.29% | 59 | 0.622 | 90.29% | 2 | $\approx 0$ | 81.43% |
| Gisette | 50.00% | 5653 | 10.7 | 50.00% | 801 | 0.298 | 48.40% |
| Internet Ads | 95.43% | 11 | 0.017 | 91.79% | 1 | 0.00015 | 96.80% |
| ISOLET | 91.14% | 600 | 1.3 | 95.77% | 38 | 0.010 | 86.58% |
| Ozone 1 | 97.24% | 13 | 0.080 | 97.24% | 3 | $\approx 0$ | 74.02% |
| Ozone 8 | 93.10% | 13 | 0.080 | 93.10% | 3 | $\approx 0$ | 69.63% |
| Pen Digits | 97.71% | 15 | 0.018 | 13.47% | 51 | 0.03 | 82.13% |
| Poker Hand | 55.25% | 62 | 0.032 | 58.62% | 411 | 0.0004 | 50.12% |
| Reuters | 85.69% | 2772 | 0.058 | 89.98% | 91425 | 0.226 | N/A |
| Spambase | 83.50% | 6 | 0.018 | 86.65% | 4 | 0.001 | 77.09% |
| Shuttle | 99.88% | 84 | 0.025 | 97.97% | 4924 | 0.006 | 92.21% |
| Vowel | 59.52% | $\approx 0$ | 0.008 | 62.55% | $\approx 0$ | $\approx 0$ | 46.10% |
| Wine | 66.67% | $\approx 0$ | $\approx 0$ | 41.67% | $\approx 0$ | $\approx 0$ | 94.44% |

| Data Set | 1-NN | 3-NN | 10-NN | 3-NN-$\delta$ | 10-NN-$\delta$ | J48 |
|---|---|---|---|---|---|---|
| ALOI | 98.58% | 97.57% | 94.10% | 98.17% | 96.18% | 89.78% |
| Chess | 54.00% | 62.95% | 77.50% | 63.52% | 81.49% | 80.56% |
| Dorothea | 84.86% | 90.29% | 90.29% | 87.71% | 90.00% | 84.86% |
| Gisette | 47.60% | 50.00% | 50.00% | 47.70% | 47.20% | 47.10% |
| Internet Ads | 96.19% | 85.98% | 85.98% | 96.34% | 95.27% | 97.25% |
| ISOLET | 88.58% | 87.30% | 88.26% | 90.89% | 92.37% | 83.45% |
| Ozone Level 1 | 95.08% | 97.24% | 97.24% | 95.87% | 97.24% | 96.65% |
| Ozone Level 8 | 90.13% | 93.10% | 93.10% | 91.72% | 92.90% | 93.29% |
| Pen Digits | 97.83% | 88.77% | 88.45% | 97.88% | 97.60% | 92.05% |
| Poker Hand | 51.14% | 53.86% | 57.04% | 53.57% | 56.92% | 56.66% |
| Reuters | 46.54% | 43.33% | 41.15% | 44.58% | 41.81% | N/A |
| Spambase | 83.50% | 60.59% | 60.59% | 85.23% | 84.04% | 91.96% |
| Shuttle | 99.89% | 99.83% | 99.78% | 99.87% | 99.85% | 99.95% |
| Vowel | 54.77% | 47.62% | 54.76% | 53.68% | 60.39% | 39.39% |
| Wine | 77.77% | 58.33% | 66.67% | 72.22% | 61.11% | 97.22% |

Table 2: Evaluation of the methods on different data sets. Times are reported in seconds. Some results are not available due to the excessive memory or time required for computation.

## 7 Results and Conclusion

In this paper we proposed a new nearest neighbor approach to classification that employs local clustering techniques, and evaluated it for a wide range of classification tasks (see Table 2). The results of the experimentation indicate that our proposed method is generally adaptable to variations in numbers and sizes of classes, and variations in the local distribution of data. The classification performance is usually superior to that of standard majority-voting $k$-NN classification for a single (fixed) value of $k$.

The classification accuracy of our methods compare favorably against those of support vector machines, particularly when the number of classes is high. With respect to computation cost, support vector machines are considerably faster for small training sets, but are outperformed by our method for data sets whose sizes or numbers of classes are large.

In the training phase our algorithm spends time on constructing the SASH and calculating the query cache. While the latter scales roughly $\mathcal{O}(|T| \log |T|)$, the construction of the SASH involves many distance computations which scale linearly with the dimension of the data. Most of the computation time spent in the prediction time involves the cross-fold validation that is used to determine the most effective combination of the adaptive techniques to use.

Experimentation reveals that the computationally expensive cross-fold validation approach achieves the best possible performance. However, evidence suggests that using a dynamic choice of $k$ together with majority voting is still competitive. If the user wishes to avoid the cost of cross-fold validation testing, our experimentation suggests the following rules of thumb:

1. Dynamic determination of $k$ should always be used.
2. Reshaping performs well when $k$ is allowed to vary, but is not likely to perform well for a fixed choice of $k$ or on training sets that are too sparse or too small.
3. Normalized voting is not likely to perform well on sets with strongly varying class sizes, and especially not for dual-class scenarios.

## References


[1] Aizerman, M., Braverman, E., Rozonoer, L.: Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control 25, 821–837 (1964)

[2] Asuncion, A., Newman, D.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[3] Boujemaa, N., Fauqueur, J., Ferecatu, M., Fleuret, F., Gouet, V., Le Saux, B.: Ikona: Interactive generic and specific image retrieval. In: Proc. Intern. Workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR) (2001)

[4] Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[5] Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.R.: Searching in metric spaces. ACM Computing Surveys 33(3), 273–321 (2001)

[6] Cover, T., Hart, P.: Nearest neighbor pattern classification. Information Theory, IEEE Transactions on 13(1), 21–27 (1967)

[7] Domeniconi, C., Peng, J., Gunopulos, D.: Locally adaptive metric nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 1281–1285 (2002)

[8] Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.: The amsterdam library of object images. Int. J. Comput. Vision 61, 103–112 (2005)

[9] Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005)

[10] Houle, M.E.: The relevant-set correlation model for data clustering. Stat. Anal. Data Min. 1(3), 157–176 (2008)

[11] Houle, M.E., Sakuma, J.: Fast approximate similarity search in extremely high-dimensional data sets. In: ICDE '05: Proceedings of the 21st International Conference on Data Engineering. pp. 619–630. IEEE Computer Society, Washington, DC, USA (2005)

[12] Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: STOC 1998: Proc. 30th ACM Symp. on Theory of Computing. pp. 604–613 (1998)

[13] Nguyen, H.T., Smeulders, A.: Active learning using pre-clustering. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning. p. 79. ACM, New York, NY, USA (2004)

[14] Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1(1), 81–106 (1986)

[15] Stone, C.J.: Consistent parametric regression. Annals of Statistics 5(4), 595–645 (1977)

[16] Suresh, Viswanath, P.: Rough-fuzzy weighted k-nearest leader classifier for large data sets. Pattern Recognition 42(9), 1719–1731 (September 2009), http://dx.doi.org/10.1016/j.patcog.2008.11.021

[17] Vapnik, V.N.: The Nature of Statistical Learning Theory (Information Science and Statistics). Springer (November 1999)

[18] Wang, J., Neskovic, P., Cooper, L.N.: Neighborhood size selection in the $k$-nearest-neighbor rule using statistical confidence. Pattern Recognition 39(3), 417 – 423 (2006)

[19] Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques with Java implementations. ACM SIGMOD Record 31(1), 76–77 (2002)

[20] Xie, Z., Hsu, W., Liu, Z., Lee, M.L.: SNNB: A selective neighborhood based naïve bayes for lazy learning. In: Proc. 6th Pacific-Asian Conf. Knowledge Discovery and Data Mining (PAKDD). pp. 104–114. Springer (2002)