DOI: 10.2298/CSIS111211014T

Nearest Neighbor Voting in High Dimensional Data: Learning from Past Occurrences

Nenad Tomašev and Dunja Mladenić

Artificial Intelligence Laboratory, Jožef Stefan Institute and
Jožef Stefan International Postgraduate School
1000 Ljubljana, Slovenia
nenad.tomasev@ijs.si, dunja.mladenic@ijs.si

Abstract. *Hubness* is a recently described aspect of the *curse of dimensionality* inherent to nearest-neighbor methods. This paper describes a new approach for exploiting the hubness phenomenon in k-nearest neighbor classification. We argue that some of the neighbor occurrences carry more information than others, by the virtue of being less frequent events. This observation is related to the hubness phenomenon and we explore how it affects high-dimensional k-nearest neighbor classification. We propose a new algorithm, Hubness Information k-Nearest Neighbor (HIKNN), which introduces the k-occurrence informativeness into the hubness-aware k-nearest neighbor voting framework. The algorithm successfully overcomes some of the issues with the previous hubness-aware approaches, which is shown by performing an extensive evaluation on several types of high-dimensional data.

1. Introduction

Supervised learning (classification) is one of the most fundamental machine learning tasks, often encountered in various practical applications. It involves assigning a label to a new piece of input data, where the label is one out of several predefined categories. Many algorithmic approaches to performing automatic classification have been explored in the past. This includes, among others, Bayesian learning methods, support vector machines (SVM), decision trees and nearest neighbor methods [1].

The k-nearest neighbor algorithm is one of the simplest pattern classification algorithms. It is based on a notion that instances which are judged to be similar in the feature space often share common properties in other attributes, one of them being the instance label itself. The basic algorithm was first proposed in [2]. The label of a new instance is determined by a majority vote of its k-nearest neighbors (kNN) from the training set. This simple rule has some surprising properties which go in its favor. For instance, when there is no overlap between the classes, 1-nearest neighbor is asymptotically optimal [3] [4]. As for the kNN rule, it has been shown to be universally consistent under some strong assumptions, namely $k \to \infty$ and $k/n \to 0$ [5] [6].

Let $D = (x_1, y_1), (x_2, y_2), ..(x_n, y_n)$ be the data set, where each $x_i \in R^d$. The x_i are feature vectors which reside in some high-dimensional Euclidean space, and $y_i \in c_1, c_2, ..c_C$ are the labels. It can be shown that in the hypothetical case of an infinite data sample, the probability of a nearest neighbor of x_i having label c is asymptotically equal to the posterior class probability in point x_i, namely $p(c|x_i) = \lim_{n\to\infty} p(c|\text{NN}(x_i))$. Real-world data is usually very sparse, so the point probability estimates achieved by kNN in practice are much less reliable. However, this is merely one aspect of the well known *curse of dimensionality*.

Concentration of distances [7,8] is another phenomenon of interest, since all nearest-neighbor approaches require a similarity measure. In high-dimensional spaces, it is very difficult to distinguish between relevant and irrelevant points and the very concept of nearest neighbors becomes much less meaningful.

Hubness is a recently described aspect of the dimensionality curse, related specifically to nearest neighbor methods [9] [10]. The term is coined to reflect the emergence of *hubs*, very frequent nearest neighbors. As such, these points exhibit a substantial influence on the classification outcome. Two types of hubs can be distinguished: *good hubs* and *bad hubs*, based on the proportion of label matches/mismatches in their k-occurrences. The phenomenon of *hubness* will be explained in more detail in Section 2.2, and the previous approaches for exploiting hubness in kNN classification will be outlined in Section 2.3.

The issue of data dimensionality needs to be emphasized because most real world data sets are in fact high-dimensional, for example: textual documents, images, audio files, data streams, medical histories, etc.

1.1. Contributions

This paper aims at further clarifying the consequences of hubness in high dimensional kNN classification, by focusing on one specific aspect of the phenomenon - the difference in the information content of the individual k-occurrences. Here we summarize the main contributions of the paper:

- When there is hubness, some points occur much more frequently in k-neighbor sets. We claim that some occurrences hence become much less informative than others, and are consequently of much lower value for the kNN classification process.
- We propose a new hubness-aware approach to k-nearest neighbor classification, Hubness Information k-Nearest Neighbor (HIKNN). The algorithm exploits the notion of occurrence informativeness, which leads to a more robust voting scheme.
- We provide a thorough experimental evaluation for the approach, by testing it both on low-to-medium hubness data and also high-hubness data from two different domains: images and text. The experiments are discussed in Section 5, while Section 7 takes a deeper look into the class probabilities which the algorithm returns.

2. Related work

2.1. kNN classification

The k-nearest neighbor method is among the most influential approaches in machine learning, due to its simplicity and effectiveness. Many extensions to the basic method have been proposed, dealing with various different aspects - including attribute weighting [11], adaptive distances [12] [13], fuzzy labels [14] [15] [16], evidence-theoretic approaches [17], and many more. Some advanced algorithms have been proposed recently, including the large margin kNN classifier which learns the Mahalanobis distance matrices via semidefinite programming [18] [19].

2.2. Hubs, frequent nearest neighbors

The emergence of *hubs* as prominent points in k-nearest neighbor methods had first been noted in analyzing music collections [20] [21]. The researchers discovered some songs which were similar to many other songs (i.e. frequent neighbors). The conceptual similarity, however, did not reflect the expected perceptual similarity.

The phenomenon of *hubness* was further explored in [9] [22], where it was shown that hubness is a natural property of many inherently high-dimensional data sets. Not only do some very frequent points emerge, but the entire distribution of k-occurrences exhibits very high *skewness*. In other words, most points occur very rarely in k-neighbor sets, less often than what would otherwise have been expected. We refer to these rarely occurring points as *anti-hubs*. [23]

Denote by $N_k(x_i)$ the number of k-occurrences of x_i and by $N_{k,c}(x_i)$ the number of such occurrences in neighborhoods of elements from class c. The latter will also be referred to as the *class hubness* of instance x_i. A k-neighborhood of x_i is denoted by $D_k(x_i)$.

The skewness of the $N_k(x)$ distribution in high dimensional data can sometimes be very severe [22]. Let us illustrate this point by plotting the $N_k(x)$ distribution for one of the datasets which we used for the experiments, namely the Acquis data. This is shown in Figure 1, for $k = 5$. Such a drastic shift in the distribution shape must certainly be taken into account when designing kNN algorithms for high dimensional data.

Hubness-aware algorithms have recently been proposed for clustering [10], instance selection [24], outlier and anomaly detection [22] [25] and classification [9] [26] [27] [28], which we will discuss below.

2.3. Hubness-aware classification

Hubs, as frequent neighbors, can exhibit both *good* and *bad* influence on kNN classification, based on the number of label *matches* and *mismatches* in the respective k-occurrences.The number of good occurrences will be denoted by

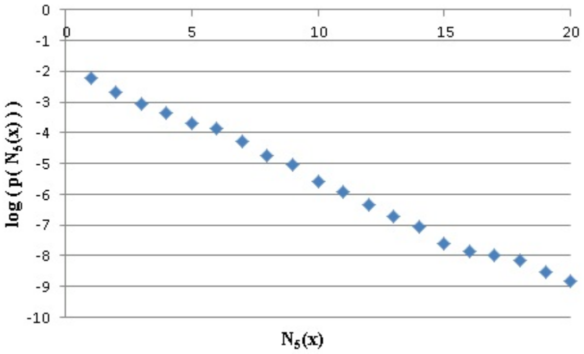


Fig. 1. The hubness distribution of the Acquis data is given for the 5-occurrence probabilities of $N_5(x) \in \{1..20\}$. We see that the distribution apparently forms a straight line on the logarithmic scale, so it is in fact exponential.

$GN_k(x_i)$ and the number of bad ones by $BN_k(x_i)$, so that $N_k(x_i) = GN_k(x_i) + BN_k(x_i)$.

All three previously proposed approaches deal with bad hubs in seemingly similar, but radically different ways. We will refer to these algorithms as hubness-weighted kNN (hw-kNN) [9], hubness fuzzy kNN (h-FNN) [26] and naive hubness Bayesian kNN (NHBNN) [27]. We discuss these ideas below, outlining their respective strengths and weaknesses.

hw-kNN

- **Idea:** When a point exhibits bad hubness, give its vote lesser weight. This has been achieved by calculating the standardized bad hubness as $h_B(x_i) = \frac{BN_k(x_i) - \mu_{BN_k}}{\sigma_{BN_k}}$, where μ_{BN_k} and σ_{BN_k} denote the mean and standard deviation of bad hubness, respectively. Each x_i is then assigned a voting weight of $w_i = e^{-h_B(x_i)}$.
- **Strengths:**
  - Reduces the influence of bad hubs
  - Very simple and easy to implement
- **Weaknesses:**
  - Each element still votes by its own label, which means that bad hubs still exhibit some detrimental influence
  - Some information is left unexploited, since class hubness is ignored
  - It is equivalent to kNN for $k = 1$

h-FNN

- **Idea:** Decompose bad hubness into fuzzy class-specific hubness-based votes as $u_c(x_i) \propto N_{k,c}(x_i)/N_k(x_i)$. This is only possible for points with

$N_k(x_i) > 0$ and only sensible for points with $N_k(x_i) > \theta$, where θ is some predefined threshold parameter. Anti-hubs are thus considered to be special cases. Their fuzzy votes are approximated by average class-to-class fuzzy votes. This algorithm is otherwise based on the fuzzy nearest neighbor (FNN) framework [14], with distance weighting included.

– **Strengths:**
  - Generalizes the hw-kNN approach by taking class hubness into account
  - Combines fuzzy votes with distance weighting

– **Weaknesses:**
  - No clear way of dealing with anti-hubs, approximations need to be used instead
  - Uses a threshold parameter θ for determining anti-hubs, which is difficult to set in practice. If learned automatically from the data, it can lead to over-fitting.

NHBNN

– **Idea:** Observe each k-occurrence as a random event and use the Naive Bayes rule to calculate the posterior class affiliation probabilities, as shown in Equation 1. The $x_{it}, t = \{1, 2..k\}$ represent the k nearest neighbors of x_i. As in h-FNN, anti-hubs are a special case and one needs to estimate their class hubness scores via local or global approximative approaches.

$$p(y_i = c|D_k(x_i)) \approx \frac{p(y_i = c)\prod_{t=1}^{k} p(x_{it} \in D_k(x_i)|y_i = c)}{\sum_{c \in C} p(y_i = c)\prod_{t=1}^{k} p(x_{it} \in D_k(x_i)|y_i = c)} \tag{1}$$

– **Strengths:**
  - Generalizes the hw-kNN approach by taking class hubness into account
  - Rephrasing the problem in Bayesian terms allows for further improvements and extensions based on the known ways for improving Bayesian classifiers

– **Weaknesses:**
  - Strong dependencies between occurrences in the same k-neighbor set greatly restrict the applicability of the approach in larger k-neighborhoods
  - Due to these dependencies, class affiliation probabilities tend to be close to 0 or 1 in many cases.
  - Additionally, both weaknesses of h-FNN hold for NHBNN as well

3. The motivation

3.1. Casting a vote: label vs class hubness

Before we delve into the specific ideas behind our proposed approach, the reasons for using the class hubness scores need to be further elucidated. For simplicity, let us begin by focusing on the 1-NN rule. It was already mentioned in the introduction that $p(c|x_i) = \lim_{n\to\infty} p(c|\text{NN}(x_i))$. If the data were not sparse and if there was no overlap between the classes and no noise, 1-NN would work really well. Of course, none of these conditions are met in real world data.

So, what happens is that nearest neighbors sometimes have different labels and this can already be seen on the training set. Observe an illustrative low-dimensional example displayed in Figure 2.

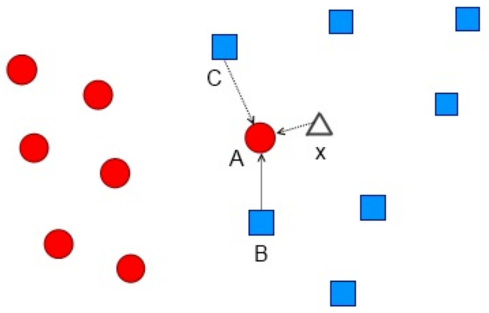


Fig. 2. Illustrative example of a binary classification case. The first class is given by the red circles, the second by the blue squares. A triangle represents an instance yet to be classified. An arrow is drawn from an instance to its nearest neighbor.

The point x is about to be classified. Let's say that the circles represent class 0 and squares represent class 1. According to the 1-NN rule, x would be assigned to class 0, since this is the label of NN$(x) = A$. But, we also have NN$(B) = A$ and NN$(C) = A$, and points B and C are of class 1. If we were to try approximating $p(y = c|A \in D_1(x)) \approx \frac{N_{1,c}(A)}{N_1(A)}$ for $c = 0, 1$, we would get $p(y = 0|A \in D_1(x)) = 0$ and $p(y = 1|A \in D_1(x)) = 1$. So, according to class hubness, x should be assigned to class 1, which seems more plausible when looking at the data.

Two-dimensional data does not exhibit hubness, so Figure 2 can only serve as a simplified model. A more general case is presented in Figure 3. Two examples are given, with class hubness scores shown on the right. In both examples, the label of x is 0 (the red circle).

In the first example, $N_{1,0}(x) = 3$ and $N_{1,1}(x) = 21$, which indicates high bad hubness. Therefore, if x is a neighbor to the point of interest, it is certainly beneficial to base the vote on the class hubness scores, instead of its label. It would reduce the probability of error.

In the second example, $N_{1,0}(x) = 3$ and $N_{1,1}(x) = 4$, which makes for a very small difference in class hubness scores. Even though $N_{1,1}(x) > N_{1,0}(x)$, the label of x is 0, so it is not entirely clear how x should vote. What needs to be evaluated is how much trust should be placed in the neighbor's label and how

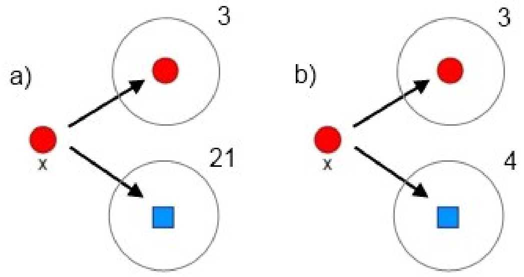


Fig. 3. A more general binary classification case. Class hubness is shown for point x towards both classes. Two examples are depicted, example 'a' where there is a big difference in previous k-occurrences, and example 'b' where there is nearly no observable difference.

much in the occurrence information. If there had been no previous occurrences of x on the training data (an anti-hub), there would be no choice but to use the label. On the other hand, for high hubness points we should probably rely more on their occurrence tendencies. It is precisely the points in between which need to be handled more carefully.

Anti-hubs While discussing the relevance of hubness for kNN classification, we must keep in mind that most points are in fact anti-hubs, when the inherent dimensionality of the data is high. This is illustrated in Figure 4, where the percentage of points exceeding certain k-occurrence thresholds is given. The Dexter data (from the UCI repository) exhibits some hubness, so that even for k as large as 10, there is still around 15% of instances that never occur as neighbors.

Both previously proposed class-hubness based approaches (h-FNN and NHBNN) have failed to provide an easy and consistent way of handling anti-hubs, which is probably their most pronounced weakness. In Section 4 we propose a new way of dealing with such low hubness points.

3.2. Informativeness

The basics What is the information content of an observed event? Intuitively, the more surprised we are about the outcome, the more information the outcome carries. We're all quite used to the sun coming up every morning and by observing this over and over again we don't gain any novel insights. If, however, the sun fails to appear on the sky someday, such a peculiar event would be much more informative, though unfortunate.

This is where information theory comes in. The event self-information is equal to the negative of the logarithm of its probability (i.e. the logarithm of the

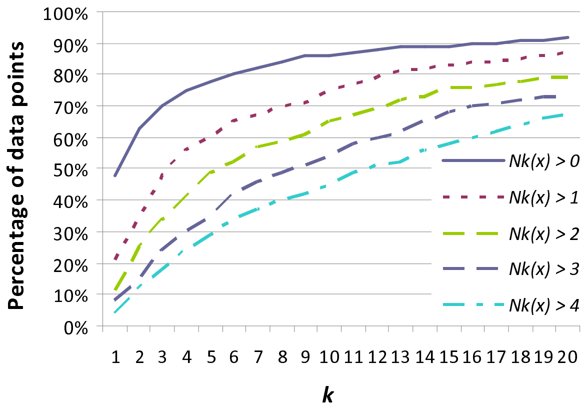


Fig. 4. Percentage of elements with hubness over a certain threshold, for $k = 1$ to $k = 20$ on Dexter data. Each line corresponds to one threshold.

inverse of the probability) [29]. It is often possible to estimate the event probabilities directly by observing the frequencies in previous occurrences, which is what we will be doing with the neighbor points.

Hubs Suppose that there is a data point $x_i \in D$ which appears in *all* k-neighbor sets of other $x_j \in D$. Assume then that we are trying to determine a label of a new data point x and that x_i also appears in this neighborhood, $D_k(x)$. This would not be surprising at all, since x_i appears in *all* other previously observed neighborhoods. Since such an occurrence *carries no information*, x_i should not be allowed to cast a vote. By going one step further, it is easy to see that less frequent occurrences may in fact be more informative and that such neighbors might be more local to the point of interest. This is exploited in our proposed approach.

Going back to the *always-a-neighbor* example, we can see that both the traditional kNN voting scheme and the fuzzy scheme proposed in the h-FNN algorithm fail to handle the extreme case properly. The fact is that whichever point x we observe, $x_i \in D_k(x)$, so there is no correlation between x_i being in $D_k(x)$ and the class affiliation of x. In case of the original kNN procedure, x_i would vote by its label, y_i. If, on the other hand, we were to vote by the class hubness induced fuzziness as in h-FNN, we would in fact be voting by class priors. This is, of course, the lesser evil, but it is still the wrong thing to do. Since there is no information that can be derived from the occurrence of x_i, its vote should be equal to zero.

This scenario does seem quite far-fetched. When reviewing the experimental results, though, it will become clear that such pathological cases are not only theoretically possible - they occasionally take place in real world data.

Anti-hubs Most high-dimensional points are anti-hubs and suppose that x_i is one such point that never occurs in k-neighborhoods on D, i.e. $N_k(x_i) = 0$. Let us say that we are trying to determine the label of a new point x and x_i is found among the neighbors, i.e. $x_i \in D_k(x)$. Such an occurrence would be highly informative. We could be fairly certain that the point x_i carries some important *local* information to the point of interest, since it is not a shared neighbor with many other points.

Of course, not all points are hubs and anti-hubs, as many points will fall somewhere between the two extremes. Any approach designed to handle the informativeness hubs and anti-hubs needs to be applicable to the entire spectrum of possible occurrence frequencies, so that these medium-hubness points are processed in an appropriate way.

It is quite surprising that these simple observations have before gone unnoticed. Previous kNN algorithms have not been taking occurrence informativeness explicitly into consideration.

This is very significant for high dimensional data, where hubs appear. The skewness in the $N_k(x)$ distribution induces the skewness in the distribution of self-information among individual neighbor occurrences. In the following Section we will propose an information-based voting procedure which exploits this fact.

4. The algorithm

Let now x_i be the point of interest, to which we wish to assign a label. Let x_{it}, $t = \{1, 2..k\}$ be its k nearest neighbors. We calculate the informativeness of the occurrences according to Equation 2. In all our calculations, we assume each data point to be its own 0^{th} nearest neighbor, thereby making all $N_k(x_i) \geq 1$. Not only does this give us some additional data, but since it makes all k-occurrence frequencies non-zero, we thereby avoid any pathological cases in our calculations.

$$\begin{aligned} p(x_{it} \in D_k(x_i)) &\approx \frac{N_k(x_{it})}{n} \\ I_{x_{it}} &= \log \frac{1}{p(x_{it} \in D_k(x_i))} \end{aligned} \tag{2}$$

We proceed by defining relative and absolute normalized informativeness. We will also refer to them as *surprise values*.

$$\alpha(x_{it}) = \frac{I_{x_{it}} - \min_{x_j \in D} I_{x_j}}{\log n - \min_{x_j \in D} I_{x_j}}, \quad \beta(x_{it}) = \frac{I_{x_{it}}}{\log n} \tag{3}$$

As we have been discussing, one of the things we wish to achieve is to combine the class information from neighbor labels and their previous occurrences. In order to do this, we need to make one more small observation. Namely, as the number of previous occurrences ($N_k(x_{it})$) increases, two things happen simultaneously. First of all, the informativeness of the current occurrence of x_{it} drops. Secondly, class hubness gives us a more accurate estimate of $p_k(y_i = c|x_{it} \in D_k(x_i))$. Therefore, when the hubness of a point is high, more information is contained in the class hubness scores. Also, when the hubness of a point is low, more information is contained in its label.

$$\bar{p}_k(y_i = c|x_{it} \in D_k(x_i)) = \frac{N_{k,c}(x_{it})}{N_k(x_{it})} = \bar{p}_{k,c}(x_{it})$$

$$p_k(y_i = c|x_{it}) \approx \begin{cases} \alpha(x_{it}) + (1 - \alpha(x_{it})) \cdot \bar{p}_{k,c}(x_{it}), & y_{it} = c \\ (1 - \alpha(x_{it})) \cdot \bar{p}_{k,c}(x_{it}), & y_{it} \neq c \end{cases} \tag{4}$$

The α factor controls how much information is contributed to the vote by the instance label and how much by its previous occurrences. If x_{it} never appeared in a k-neighbor set apart from its own, i.e. $N_k(x_{it}) = 1$, then it votes by its label. If, on the other hand, $N_k(x_{it}) = \max_{x_j \in D} N_k(x_j)$, then the vote is cast entirely according to the class hubness scores.

The fuzzy votes are based on the $p_k(y_i = c|x_{it})$, which are approximated according to Equation 4. These probabilities are then weighted by the absolute normalized informativeness $\beta(x_{it})$. This is shown in Equation 5.

$$u_c(x_i) \propto \sum_{t=1}^{k} \beta(x_{it}) \cdot d_w(x_{it}) \cdot p_k(y_i = c|x_{it}) \tag{5}$$

Additional distance weighting has been introduced for purposes of later comparison with the h-FNN algorithm [26], since it also employs distance weighting. It is not an essential part of the algorithm. We opted for the same distance weighting scheme used in h-FNN, which was in turn first proposed in FNN [14]. It is given in Equation 6.

$$d_w(x_{it}) = \frac{\|x_i - x_{it}\|^{-2}}{\sum_{t=1}^{k} (\|x_i - x_{it}\|^{-2})} \tag{6}$$

Equations 2, 3, 4 and 5 represent our proposed solution for exploiting the information contained in the past k-occurrences on the training data and we will refer to this new algorithm as Hubness Information k-Nearest Neighbor (HIKNN). It embodies some major improvements over the previous approaches:

- Unlike h-FNN and NHBNN, it is essentially parameter-free, one only needs to set the neighborhood size (k).
- Anti-hubs are no longer a special case. They are, however, handled appropriately via the information-based framework.
- Label information is combined with information from the previous k-occurrences, so that both sources of information are exploited for the voting.
- Total occurrence informativeness is taken into account

Algorithm 1 HIKNN: Training

```
**Input:** ($X$,$Y$,$k$)
training set $T = (X, Y) \subset R^{d\times 1}$
number of neighbors $k \in \{1, 2 \dots n-1\}$
**Train:**
kNeighbors = findNeighborSets($T$, $k$)
for all $(x_i, y_i) \in (X, Y)$ do
    $N_k(x_i) = 0$
    for all $c = 1 \dots C$ do
        count $N_{k,c}(x_i)$
        $N_k(x_i) += N_{k,c}(x_i)$
    end for
    calculate $\alpha(x_i)$ and $\beta(x_i)$ by Eq. 3
    for all $c = 1 \dots C$ do
        calculate $p_k(y = c|x_i)$ by Eq. 4
    end for
end for
```

The training phase of the algorithm is summarized in (1). The voting is simply done according to (5) and requires no further explanations.

The time complexity of HIKNN, as with all other hubness-based approaches, is asymptotically the same as constructing a kNN graph. Fast algorithms for constructing approximate kNN graphs exist, like the algorithm by [30]. This particular procedure runs in $\Theta(dn^{1+\tau})$ time, where $\tau \in (0, 1]$ is a parameter which is used to set a trade off between speed and graph construction accuracy.

5. Experiments

We compared our proposed HIKNN algorithm to the existing related algorithms: kNN, hw-kNN, h-FNN and NHBNN - on 32 classification problems. We had three test cases: low-to-medium hubness data of lower intrinsic dimensionality, high-hubness textual data and high-hubness image data. In all cases, 10-times 10-fold cross validation was performed. Corrected resampled t-test was used to check for statistical significance. All experiments were performed for $k = 5$, which is a standard choice. Default values described in the respective papers were used for the parameters in h-FNN and NHBNN. The detailed results are given in Table 3 and the basic properties of the datasets are discussed in Table 2.

5.1. The data

Low and medium hubness data Datasets from the well known UCI data repository (http://archive.ics.uci.edu/ml/datasets.html) are usually of low or medium hubness. Since such datasets are less interesting from the perspective of hubness-aware algorithms, we present here the results on a sample of 10 UCI datasets.

The datasets were picked so that they correctly reflect the results on the entire repository. The Manhattan distance was used for this data, as well as for the image data. All features were normalized prior to classification.

Text The Acquis aligned corpus data (http://langtech.jrc.it/JRC-Acquis.html) represents a set of more than 20000 documents in several different languages. In the experiments we only used the English documents. The data was preprocessed and represented as a bag-of-words (term frequencies). On top of this data, 14 different binary classification problems were examined. We used the cosine similarity.

Images We used several datasets in the experiments which were subsets taken from the ImageNet online repository (http://www.image-net.org/). These datasets were selected to match some of the ones used in [28]. All datasets are quantized feature representations. Representations iNet3-iNet7 are based on SIFT features and were also appended color information.

On the other hand, in case of iNet3Err100, iNet3Err150 and iNet3Err1000 - Haar wavelet features were used. These three representations have one interesting property. Due to an I/O error during feature extraction, 5 images were accidentally assigned empty representations (zero vectors). Normally, this would have probably gone unnoticed. In this case, however, the hubness of zero vectors increased drastically with the representation dimensionality. Since all 5 of these points were of the minority class, the classification results were affected greatly and this a prime example of how bad the bad hubness can get in high dimensional data.

5.2. The results

The results in Table 3 show that the hubness-aware algorithms clearly outperform the basic kNN algorithm. Also, HIKNN seems to be the overall best approach, with a clear edge on the textual and UCI data, while performing more or less equal as h-FNN and NHBNN on image datasets. The detailed comparison between the algorithms is shown in Table 1. By comparing both the total number of wins and also the number of wins between pairs of algorithms, we see that HIKNN is to be preferred to the second-best algorithm in the experiments, h-FNN - since it beats it quite convincingly 27(9) : 5(1) in direct comparison and 115(74) : 79(53) overall.

In further comparisons on the image data, we examined the entire range of k-values to see how the algorithms are influenced by neighborhood size. The results on iNet6 are shown in Figure 5. We see that an increase in k separates the algorithms and makes distinctions easier. HIKNN achieves the best results for $k > 5$, where the highest accuracies are achieved. It is not surprising that the accuracy gain over h-FNN increases with k, since the number of large hubs also increases - and the payoff from taking their informativeness into account becomes more substantial. Also, we see that NHBNN simply fails to work

Table 1. Pairwise comparison of classifiers: number of wins (with statistically significant ones in parenthesis)

	kNN	hw-kNN	h-FNN	NHBNN	HIKNN	Total
kNN	–	0 (0)	1 (0)	10 (8)	0 (0)	11 (8)
hw-kNN	32 (27)	–	14 (1)	18 (13)	1 (0)	65 (41)
h-FNN	31 (27)	17 (9)	–	25 (16)	5 (1)	79 (53)
NHBNN	22 (19)	14 (9)	4 (1)	–	3 (1)	43 (30)
HIKNN	**32 (31)**	**29 (14)**	**27 (9)**	**27 (20)**	–	**115 (74)**

when the dependencies between neighbors become too strong, in this case for $k > 15$. The accuracy graphs for the other datasets depict the same general tendencies.

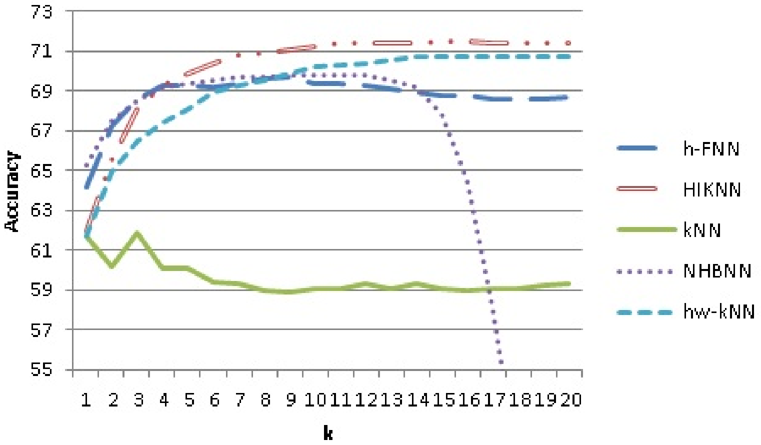


Fig. 5. Classifier accuracies over a range of neighborhood sizes $k \in 1..20$ on iNet6 dataset.

The results for the three iNet3Err representations require special attention. As mentioned in the data description, 5 points in the dataset ended up being zero vectors representing the minority class. We see how an increase in the representation dimensionality causes an amazing increase in bad hubness, which in turn completely disables the basic kNN classifier, as well as the hw-kNN approach. On this 3-category dataset kNN ends up being worse than zero-rule! Keep in mind that such a great drop in accuracy was caused by no more than 5 erroneous instances, out of 2731 total. In the end, 80% of the 5-occurrences were label mismatches. On the other hand, the algorithms based on class hubness: h-FNN, NHBNN and HIKNN - even though affected, retained a much more decent accuracy: 60% compared to the mere 21% by kNN. These five

Table 2. Overview of the datasets. Each dataset is described by its size, dimensionality, the number of categories, skewness of the N_5 distribution (S_{N_5}), proportion of bad 5-occurrences BN_5, as well as the maximal achieved number of occurrences on the dataset.

Data set	size	d	C	S_{N_5}	BN_5	$\max N_5$
dexter	300	20000	2	6.64	30.5%	219
diabetes	768	8	2	0.19	32.3%	14
glass	214	9	6	0.26	35.0%	13
ionosphere	351	34	2	2.06	12.5%	34
isolet1	1560	617	26	1.23	28.7%	30
page-blocks	5473	10	5	0.31	5.0%	16
segment	2310	19	7	0.33	5.3%	15
sonar	208	60	2	1.28	21.3%	22
vehicle	846	18	4	0.64	36.0%	14
vowel	990	10	11	0.60	9.7%	16
Acquis1	23412	254963	2	62.97	19.2%	4778
Acquis2	23412	254963	2	62.97	8.7%	4778
Acquis3	23412	254963	2	62.97	27.3%	4778
Acquis4	23412	254963	2	62.97	12.2%	4778
Acquis5	23412	254963	2	62.97	5.7%	4778
Acquis6	23412	254963	2	62.97	7.6%	4778
Acquis7	23412	254963	2	62.97	18.1%	4778
Acquis8	23412	254963	2	62.97	9.3%	4778
Acquis9	23412	254963	2	62.97	7.6%	4778
Acquis10	23412	254963	2	62.97	21.4%	4778
Acquis11	23412	254963	2	62.97	23.4%	4778
Acquis12	23412	254963	2	62.97	9.8%	4778
Acquis13	23412	254963	2	62.97	16.4%	4778
Acquis14	23412	254963	2	62.97	6.9%	4778
iNet3Err100	2731	100	3	20.56	10.2%	375
iNet3Err150	2731	150	3	25.1	34.8%	1280
iNet3Err1000	2731	1000	3	23.3	79.7%	2363
iNet3	2731	416	3	8.38	21.0%	213
iNet4	6054	416	4	7.69	40.3%	204
iNet5	6555	416	5	14.72	44.6%	469
iNet6	6010	416	6	8.42	43.4%	275
iNet7	10544	416	7	7.65	46.2%	268

Table 3. Overview of the experiments. Classification accuracy is given for kNN, hubness-weighted kNN (hw-kNN), hubness-based fuzzy nearest neighbor (h-FNN) and hubness information k-nearest neighbor (HIKNN). All experiments were performed for $k = 5$. The symbols •/○ denote statistically significant worse/better performance ($p < 0.05$) compared to HIKNN. The best result in each line is in bold.

Data set	kNN	hw-kNN	h-FNN	NHBNN	**HIKNN**
dexter	57.2 ± 7.0 •	67.7 ± 5.4	67.6 ± 4.9	**68.0 ± 4.9**	**68.0 ± 5.3**
diabetes	67.8 ± 3.7 •	75.6 ± 3.7	75.4 ± 3.2	73.9 ± 3.4	**75.8 ± 3.6**
glass	61.5 ± 7.3 •	65.8 ± 6.7	67.2 ± 7.0	59.1 ± 7.5 •	**67.9 ± 6.7**
ionosphere	80.8 ± 4.5 •	87.9 ± 3.6	90.3 ± 3.6 ○	**92.2 ± 3.2** ○	87.3 ± 3.8
isolet1	75.2 ± 2.5 •	82.5 ± 2.1 •	83.8 ± 1.8 •	83.0 ± 2.0 •	**86.8 ± 1.5**
page-blocks	95.1 ± 0.6 •	95.8 ± 0.6 •	96.0 ± 0.6	92.6 ± 0.6 •	**96.2 ± 0.6**
segment	87.6 ± 1.5 •	88.2 ± 1.3 •	88.8 ± 1.3 •	87.8 ± 1.3 •	**91.2 ± 1.1**
sonar	82.7 ± 5.5	83.4 ± 5.3	82.0 ± 5.8	81.1 ± 5.6 •	**85.3 ± 5.5**
vehicle	62.5 ± 3.8 •	65.9 ± 3.2	64.9 ± 3.6 •	63.7 ± 3.5 •	**67.2 ± 3.6**
vowel	87.8 ± 2.2 •	88.2 ± 1.9 •	91.0 ± 1.8 •	88.1 ± 2.2 •	**93.6 ± 1.6**
Acquis1	78.7 ± 1.0 •	87.5 ± 0.8 •	88.8 ± 0.7	88.4 ± 0.7 •	**89.4 ± 0.6**
Acquis2	92.4 ± 0.5 •	93.6 ± 0.5	93.3 ± 0.5	92.5 ± 0.5 •	**93.7 ± 0.5**
Acquis3	72.7 ± 0.9 •	78.7 ± 0.9	79.5 ± 0.9	78.9 ± 0.9	**79.6 ± 0.9**
Acquis4	89.8 ± 0.6 •	90.6 ± 0.6	90.5 ± 0.6	87.4 ± 0.7 •	**91.0 ± 0.5**
Acquis5	97.3 ± 0.3 •	97.6 ± 0.3	97.5 ± 0.3	95.1 ± 0.4 •	**97.7 ± 0.3**
Acquis6	93.6 ± 0.4 •	94.4 ± 0.5	94.0 ± 0.5 •	92.5 ± 0.5 •	**94.6 ± 0.5**
Acquis7	82.9 ± 0.8 •	86.3 ± 0.7 •	86.1 ± 0.6 •	85.7 ± 0.7 •	**87.0 ± 0.7**
Acquis8	92.3 ± 0.5 •	93.0 ± 0.5	93.1 ± 0.5	91.0 ± 0.5 •	**93.5 ± 0.5**
Acquis9	93.0 ± 0.5 •	**94.8 ± 0.4**	94.2 ± 0.5 •	93.4 ± 0.5 •	**94.8 ± 0.4**
Acquis10	83.1 ± 1.6 •	88.8 ± 0.7 •	88.7 ± 0.6 •	87.4 ± 0.7 •	**89.7 ± 0.5**
Acquis11	77.7 ± 0.9 •	81.8 ± 0.8	82.4 ± 0.6	81.9 ± 0.7	**82.5 ± 0.5**
Acquis12	91.9 ± 0.6 •	**92.8 ± 0.5**	92.6 ± 0.5	90.7 ± 0.6 •	**92.8 ± 0.5**
Acquis13	85.6 ± 0.7 •	87.5 ± 0.6	87.1 ± 0.7 •	85.2 ± 0.7 •	**88.0 ± 0.7**
Acquis14	94.2 ± 0.4 •	94.9 ± 0.4	94.6 ± 0.5	92.5 ± 0.5 •	**95.0 ± 0.5**
iNet3Err100	92.4 ± 0.9 •	93.6 ± 0.9 •	97.5 ± 0.9	97.5 ± 0.9	**97.6 ± 0.9**
iNet3Err150	80.0 ± 2.0 •	88.7 ± 2.0 •	94.6 ± 0.9	94.6 ± 0.9	**94.8 ± 0.9**
iNet3Err1000	21.2 ± 2.0 •	27.1 ± 11.2 •	59.5 ± 3.2	**59.6 ± 0.9**	**59.6 ± 3.2**
iNet3	72.0 ± 2.7 •	80.8 ± 2.3	**82.4 ± 2.2**	81.8 ± 2.3	82.2 ± 2.0
iNet4	56.2 ± 2.0 •	63.3 ± 1.9 •	**65.2 ± 1.7**	64.6 ± 1.9	64.7 ± 1.9
iNet5	46.6 ± 2.0 •	56.3 ± 1.7 •	**61.9 ± 1.7**	61.8 ± 1.9	60.8 ± 1.9
iNet6	60.1 ± 2.2 •	68.1 ± 1.6 •	69.3 ± 1.7	69.4 ± 1.7	**69.9 ± 1.9**
iNet7	43.4 ± 1.7 •	55.1 ± 1.5 •	**59.2 ± 1.5**	58.2 ± 1.5	56.9 ± 1.6
AVG	76.72	81.13	83.09	81.86	**83.60**

points occur in nearly all k-neighborhoods and this dataset shows how some pathological cases of very bad hubness also occasionally emerge in practical situations. Even if the erroneous points were not of the minority class, they would still have caused significant misclassification. Also, note that the major hub in the 1000-dimensional case appears in 86.5% of all k-neighbor sets. Its occurrence is, therefore, not very informative - and this further justifies the discussion presented in Section 3.2.

Bad hubness of the data is closely linked to the error of the kNN classification. The Pearson correlation coefficient comparing the kNN error with bad hubness percentages on the datasets in our experiments gives 0.94, which indicates strong positive correlation between the two quantities. HIKNN bases its votes on expectations derived from the previous k-occurrences, so it is encouraging that the correlation between the accuracy gain over kNN and bad hubness of the data is also very strong: 0.87 according to the Pearson coefficient.

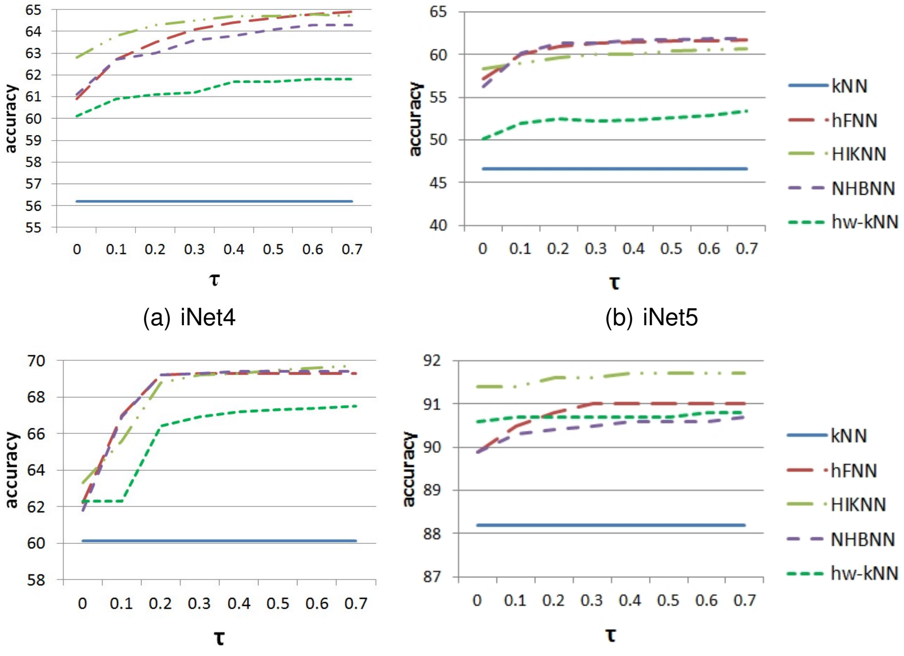
6. The approximate implementation

Computing all the k-neighbor sets on the training data in order to build an occurrence model could be overly time-consuming in large-scale data collections. Hubness-aware approaches would be applicable in large-scale scenarios only if it were possible to retain the previously observed improvements while working with some sort of approximate kNN sets.

Many approximate kNN algorithms have been proposed in the literature, either for speeding-up individual queries or constructing an entire kNN graph. It is the latter that is of interest for building an occurrence model. Many of these procedures had been proposed specifically for handling high-dimensional data, which is where hubness-aware classification has been shown to be useful.

In our experiments we focused on one such approach [30]. It is a divide and conquer method based on recursive Lanczos bisection. The time complexity of the procedure is $\Theta(dn^{1+\tau})$, where $\tau \in (0,1]$ reflects the quality of the approximation. There are two ways to implement the recursive division and we have chosen the GLUE method, as it has proven to be significantly faster than the OVERLAP method, though the quality of the resulting graph is only slightly inferior in comparison [30]. The question that we would like to answer is: *for which values of τ are we able to retain the improvements observed on actual kNN sets*?

Re-running all the experiments for all τ values would be beyond the scope of this paper. We did, however, examine the full spectrum of τ-values for the four datasets previously used in the experiments. We report the results for the iNet4, iNet5, iNet6 and Acquis1 datasets in Figure 6. The original Acquis data had too many features for our approximate kNN graph implementation to be able to handle it properly in reasonable time, so we considered a projection onto a 400-dimensional feature space. The data was projected via canonical correlation analysis procedure onto a common semantic space obtained by correlating the



(a) iNet4

(b) iNet5

(c) iNet6

(d) Acquis1 projection

Fig. 6. The accuracy of the hubness-aware approaches when the occurrence model is inferred from the approximate kNN graph generated by [30]. We see that there are significant improvements even for $\tau = 0$.

English and French aligned versions of documents from the dataset [31] [32]. It is one of the standard dimensionality reduction techniques used in text mining and its details are beyond the scope of this paper.

The results shown in Figure 6 are indeed very encouraging. They suggest that significant improvements over the kNN baseline are possible even when the graph is constructed in linear time (w.r.t. number of instances). Moreover, the quality level of $\tau = 0.2$ or $\tau = 0.3$ already seems good enough to capture most of the original occurrence information, as the resulting accuracies are quite close to the ones achieved in the original experiments.

The accuracy curves for different algorithms sometimes intersect. This can be seen for iNet5, iNet6 and Acquis1 in Figure 6. In general, the approximate results correspond rather well to the non-approximate results, but the correlation between the two can vary depending on the particular choice of τ.

In these initial findings HIKNN appears to be quite robust to the employed approximate kNN graph construction method for $\tau = 0$. This is a very nice property, as it allows for obtaining usable results in reasonable time. If better approximations are required, $\tau = 0.3$ should suffice.

A comparison between the results shown in Figure 6(d) and those previously summarized in Table 3 reveals that dimensionality reduction may sometimes significantly affect the classification process and improve the overall classification accuracy. Even though hubness is practically unavoidable in most high-dimensional data mining tasks, its severity does depend on the particular choice of feature representation and/or similarity measure. It is, therefore, not surprising that the dimensionality reduction of the Acquis data helped the kNN classifiers by reducing data hubness. The hubness was not entirely eliminated and this is why all the hubness-aware classification methods still managed to outperform the kNN baseline for all the τ values.

These initial experiments suggest that hubness-aware methods are applicable even to large datasets, as the scalable, approximate kNN graph construction methods are able to deliver good hubness estimates. More experiments are needed to reach the final verdict, on different types of high-dimensional data.

7. Estimating class probabilities

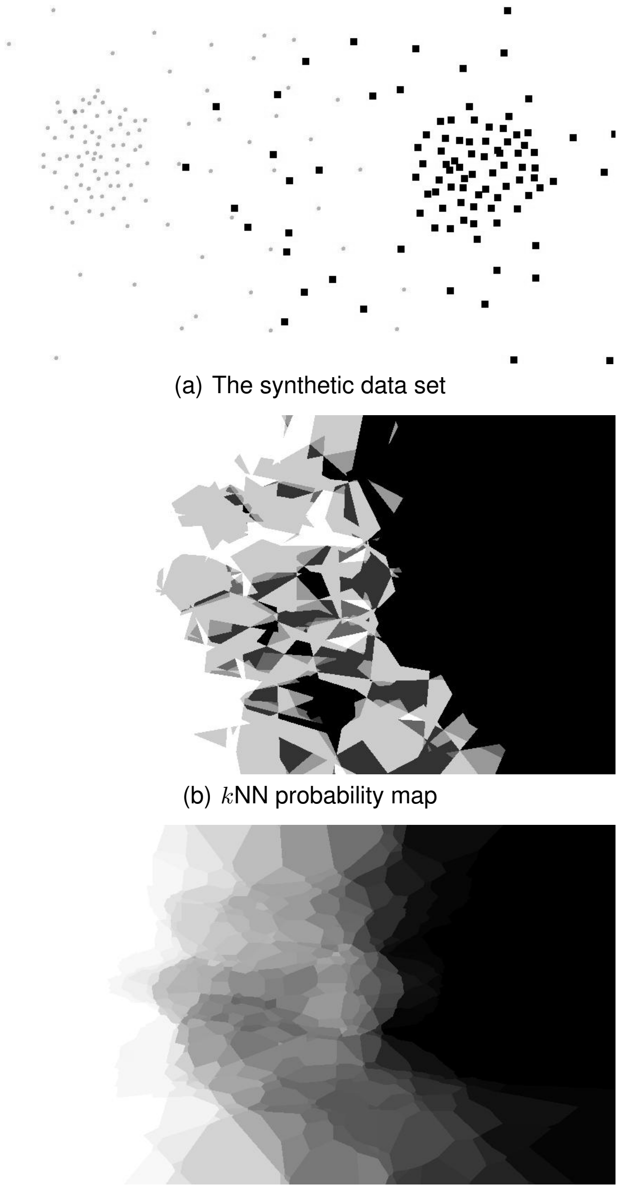
Most frequently, in classification, we are simply interested in assigning a label to a point of interest. What this label suggests is that we are entirely certain that a point belongs to a given class. However, this is just a special case of a more general problem. We would in fact like to be able to assign a 'fuzzy' label to each object, so that it belongs to several classes at the same time. This 'belonging' marks our confidence in any particular atomic label choice.

There are cases, however, when the classes overlap. This happens very frequently in real-world data. There exist points then, in these overlapping regions, that could belong to either of the neighboring categories. In such cases it is meaningless to assign a simple 'crisp' label to each point - what we would like to be able to do is to predict the actual class probability at each point, for every given class. This probability reflects the relative density of each class probability distribution at that point.

The HIKNN algorithm was made to be fuzzy and in the following experiments we wished to determine how well the predicted class probabilities reflect our intuition about the data. The basic kNN algorithm can also be used for point class probability estimates and it is a useful baseline for comparison.

In order to check if the predicted values make sense or not, we examined the algorithm output on synthetic 2D data. The fact that data has only 2 dimensions allows us to draw a *probability map*, where each pixel is 'classified' by the examined algorithms and assigned a probability of belonging to each class. We have generated several such datasets and here we discuss one of them. The dataset is simple, representing 2 categories with overlapping border regions. We have used HIKNN without the distance weighting. The resulting probability maps can be seen in Figure 7.

We see that the probability map generated by HIKNN looks much more natural in the overlapping region. The gradient between the classes should be more or less smooth if the model is able to generalize well. kNN produces a fractured



(a) The synthetic data set

(b) kNN probability map

(c) HIKNN probability map

Fig. 7. Probability maps inferred from kNN and HIKNN on synthetic data, for $k = 5$. Each pixel was classified by the algorithms and assigned a probability value of belonging to each of the two classes.

landscape, essentially over-fitting on the training data. These maps suggest that the votes based on previous occurrences may offer better estimates of the underlying class probabilities, which we intend to explore more thoroughly in our future work.

8. Conclusion

In this paper we presented a novel approach for handling high-dimensional data in k-NN classification, Hubness Information k-Nearest Neighbor (HIKNN). It is a hubness-aware approach which is based on evaluating the informativeness of individual neighbor occurrences. Rare neighbors (anti-hubs) are shown to carry valuable information which can be well exploited for classification.

The algorithm is parameter-free, unlike the previous class-hubness based hubness-aware classification algorithms. The danger of over-fitting is thereby greatly reduced.

The algorithm was compared to the three recently proposed hubness-aware approaches (hw-kNN, h-FNN, NHBNN), as well as the kNN baseline on 32 classification problems. Our proposed approach had an overall best performance in the experiments.

Since HIKNN modifies only the voting, it is easily extensible and could be combined with various sorts of metric learning or dynamic k-neighbor sets. We intend to explore these directions thoroughly in our future work.

Acknowledgments. This work was supported by the Slovenian Research Agency, the IST Programme of the EC under PASCAL2 (IST-NoE-216886) and the Serbian Ministry of Education and Science project no. OI174023.

References

1. I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann Publishers, 2005.
2. E. Fix and J. Hodges, “Discriminatory analysis, nonparametric discrimination: consistency properties,” USAF School of Aviation Medicine, Randolph Field, Texas, Tech. Rep., 1951.
3. T.M.Cover and P.E.Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. IT-13, no. 1, pp. 21–27, 1967.
4. L. Devroye, “On the inequality of cover and hart,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 75–78, 1981.
5. C.J.Stone, “Consistent nonparametric regression,” *Annals of Statistics*, vol. 5, pp. 595–645, 1977.
6. A. K. L. Devroye, L. Gyorfi and G. Lugosi, “On the strong universal consistency of nearest neighbor regression function estimates,” *Annals of Statistics*, vol. 22, pp. 1371–1385, 1994.
7. D. François, V. Wertz, and M. Verleysen, “The concentration of fractional distances,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 7, pp. 873–886, 2007.
8. C. C. Aggarwal, A. Hinneburg, and D. A. Keim, “On the surprising behavior of distance metrics in high dimensional spaces,” in *Proc. 8th Int. Conf. on Database Theory (ICDT)*, 2001, pp. 420–434.

9. M. Radovanović, A. Nanopoulos, and M. Ivanović, “Nearest neighbors in high-dimensional data: The emergence and influence of hubs,” in *Proc. 26th Int. Conf. on Machine Learning (ICML)*, 2009, pp. 865–872.
10. N. Tomašev, M. Radovanović, D. Mladenić, and M. Ivanović, “The role of hubness in clustering high-dimensional data.” in *PAKDD (1)’11*, 2011, pp. 183–195.
11. E.-H. Han, G. Karypis, and V. Kumar, “Text categorization using weight adjusted k-nearest neighbor classification,” in *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, D. Cheung, G. Williams, and Q. Li, Eds. Springer Berlin / Heidelberg, 2001, vol. 2035, pp. 53–65.
12. J. Wang, P. Neskovic, and L. N. Cooper, “Improving nearest neighbor rule with a simple adaptive distance measure,” *Pattern Recogn. Lett.*, vol. 28, pp. 207–213, January 2007.
13. Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles, “Iknn: Informative k-nearest neighbor pattern classification,” in *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*, ser. PKDD 2007. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 248–264.
14. J. E. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest-neighbor algorithm,” in *IEEE Transactions on Systems, Man and Cybernetics*, 1985, pp. 580–585.
15. R. Jensen and C. Cornelis, “A new approach to fuzzy-rough nearest neighbour classification,” in *Rough Sets and Current Trends in Computing*, ser. Lecture Notes in Computer Science, C.-C. Chan, J. Grzymala-Busse, and W. Ziarko, Eds. Springer Berlin / Heidelberg, 2008, vol. 5306, pp. 310–319.
16. W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and H. Dong, “An adaptive fuzzy knn text classifier,” in *Computational Science — ICCS 2006*, ser. Lecture Notes in Computer Science, V. Alexandrov, G. van Albada, P. Sloot, and J. Dongarra, Eds. Springer Berlin / Heidelberg, 2006, vol. 3993, pp. 216–223.
17. L. Wang, L. Khan, and B. Thuraisingham, “An effective evidence theory based k-nearest neighbor (knn) classification,” in *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 797–801. [Online]. Available: http://portal.acm.org/citation.cfm?id=1486927.1487026
18. K. Q. Weinberger, J. Blitzer, and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *In NIPS*. MIT Press, 2006.
19. M. R. Min, D. A. Stanley, Z. Yuan, A. J. Bonner, and Z. Zhang, “A deep non-linear feature mapping for large-margin knn classification,” in *ICDM*, 2009, pp. 357–366.
20. J. Aucouturier and F. Pachet, “Improving timbre similarity: How high is the sky?” *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, 2004.
21. J. Aucouturier, “Ten experiments on the modelling of polyphonic timbre,” Docteral dissertation, University of Paris 6, Tech. Rep., 2006.
22. M. Radovanović, A. Nanopoulos, and M. Ivanović, “Hubs in space: Popular nearest neighbors in high-dimensional data,” *Journal of Machine Learning Research*, vol. 11, pp. 2487–2531, 2011.
23. ——, “On the existence of obstinate results in vector space models,” in *Proc. 33rd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2010, pp. 186–193.
24. K. Buza, A. Nanopoulos, and L. Schmidt-Thieme, “Insight: efficient and effective instance selection for time-series classification,” in *Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part II*, ser. PAKDD’11. Springer-Verlag, 2011, pp. 149–160.
25. N. Tomašev and D. Mladenić, “Exploring the hubness-related properties of oceanographic sensor data,” in *Proceedings of the SiKDD conference*, 2011.

26. N. Tomašev, M. Radovanović, D. Mladenić, and M. Ivanović, “Hubness-based fuzzy measures for high dimensional k-nearest neighbor classification,” in *Machine Learning and Data Mining in Pattern Recognition, MLDM conference*, 2011.
27. ——, “A probabilistic approach to nearest neighbor classification: Naive hubness bayesian k-nearest neighbor,” in *Proceeding of the CIKM conference*, 2011.
28. N. Tomašev, R. Brehar, D. Mladenić, and S. Nedevschi, “The influence of hubness on nearest-neighbor methods in object recognition,” in *IEEE Conference on Intelligent Computer Communication and Processing*, 2011.
29. D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.
30. J. Chen, H. ren Fang, and Y. Saad, “Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection,” *Journal of Machine Learning Research*, vol. 10, pp. 1989–2012, 2009.
31. D. R. Hardoon, S. Szedmák, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.
32. H. Hotelling, “The most predictable criterion,” *Journal of Educational Psychology*, vol. 26, pp. 139–142, 1935.

Nenad Tomašev is a PhD student at the Artificial Intelligence Laboratory at J. Stefan Institute. He graduated in 2008 at the Department for Mathematics and Informatics at the University of Novi Sad, Serbia. His research topics include Machine Learning, Data Mining, Artificial Life and Stochastic Optimization. He has been actively participating in teaching at both Petnica Science Center and Višnjan Summer School.

Prof. Dr. Dunja Mladenić works as a researcher and a project manager at J. Stefan Institute, leading Artificial Intelligence Laboratory and teaching at J. Stefan International Postgraduate School. She is an expert on study and development of Machine Learning, Data/Text Mining, Semantic Technology techniques and their application on real-world problems. She is associated with the J. Stefan Institute since 1987. She got her MSc and PhD in Computer Science at University of Ljubljana in 1995 and 1998 respectively. She was a visiting researcher at School of Computer Science, Carnegie Mellon University, USA in 1996-1997 and in 2000-2001. Dunja Mladenić has experience in coordinating EU projects and acting on management board of several European research and development projects. She was a program co-chair of ECML 2007, a general chair of ECMLPKDD 2009. She is a co-inventor on a patent application with Cycorp Inc., USA.

Dunja Mladenić is the Slovenian representative in EC Enwise STRATA ETAN Expert Group ”Promoting women scientists from the Central and Eastern European countries and the Baltic States to produce gender equality in science in the wider Europe”. She serves as project evaluator of project proposals for EC programme on Information and Society Technology (IST). In 2001, she was evaluator of project proposals for National Science Foundation (NSF) initiative on Information Technology Research (ITR), NSF 00-126, USA. She has published papers in refereed journals and conferences, co-editor of several books, served in the program committee of international conferences and organized international events in the area of Text Mining, Link Analysis and Data Mining.

Received: December 11, 2011; Accepted: June 5, 2012.