Computer Science and Information Systems 0(0):1–6 DOI: N/A

Correcting the Hub Occurrence Prediction Bias in Many Dimensions

Nenad Tomašev[1], Krisztian Buza[2], and Dunja Mladenić[1]

[1] Institute Jožef Stefan, Jamova 39
1000 Ljubljana, Slovenia
nenad.tomasev@gmail.com, dunja.mladenic@ijs.si
[2] Institute of Genomic Medicine and Rare Disorders, Tömő utca 25-29.
1083 Budapest, Hungary
chrisbuza@yahoo.com

Abstract. Data reduction is a common pre-processing step for k-nearest neighbor classification (kNN). The existing prototype selection methods implement different criteria for selecting relevant points to use in classification, which constitutes a selection bias. This study examines the nature of the instance selection bias in intrinsically high-dimensional data. In high-dimensional feature spaces, hubs are known to emerge as centers of influence in kNN classification. These points dominate most kNN sets and are often detrimental to classification performance. Our experiments reveal that different instance selection strategies bias the predictions of the behavior of hub-points in high-dimensional data in different ways. We propose to introduce an intermediate un-biasing step when training the neighbor occurrence models and we demonstrate promising improvements in various hubness-aware classification methods, on a wide selection of high-dimensional synthetic and real-world datasets.

Keywords: instance selection, data reduction, classification, bias, k-nearest neighbor, hubness, curse of dimensionality

1. Introduction

The k-nearest neighbor (kNN) classification method [11] is a widely used nonparametric data mining technique. The label in the point of interest is determined by a majority vote of its nearest neighbors. This deceivingly simple procedure exhibits some beneficial asymptotic properties. As the sample size approaches infinity, the nearest neighbor classifier (k=1) error rate is guaranteed to be no worse than twice the Bayes error rate, which is the optimal error rate for a given distribution [8].

Various classifiers have been developed over the years, yet kNN is still very frequently used in many practical applications. It is considered the state-of-the-art in time series classification, especially when paired with the dynamic time warping distance [46]. It is also a good basis for developing methods for learning under class imbalance, due to its high specificity and low generalization bias [3]. Some modifications of the basic approach have been proposed for object recognition [2], medical image segmentation [45], tag recommendation [16] and document classification. Results of kNN classification are easily interpretable, which is relevant for many types of expert systems.

Despite its popularity, kNN suffers from some serious drawbacks. Most importantly, there are issues with scalability, due to its high storage requirements and relatively slow classification response. Its high specificity bias, which is useful in imbalanced data classification, also makes it more prone to noise and erroneous/mislabeled data.

One of the most promising research directions in addressing these issues is data reduction. Reducing the size of the training set speeds up subsequent classification and reduces storage requirements, while possibly eliminating outliers and noisy examples. There are two types of data reduction algorithms, one where the prototypes are generated through some internal models and the other where they are selected from among the existing data points. In this paper, we will focus on the latter. We will discuss several different approaches to instance selection in Section 2.2.

High-dimensional data pose additional challenges for k-nearest neighbor methods. Concentration of distances [13] affects kNN classification in severely negative ways, as it becomes more difficult to distinguish between relevant and irrelevant points. As distances converge, everything starts to looks the same. The notion of nearest neighbors becomes far less meaningful [10], though it is usually still possible to differentiate between different categories.

Hubness is another consequence of high intrinsic data dimensionality that affects kNN methods [32]. The distribution of influence in k-nearest neighbor classification becomes highly asymmetric and skewed to the right. A small number of hubs emerges and dominates most k-nearest neighbor sets. Consequently, most remaining points occur rarely or never as neighbors. Hubs tend to link and co-occur in frequent neighbor pairs [39]. The presence of hubs is usually detrimental to classification, especially in presence of class imbalance [41]. Detrimental hub points are referred to as *bad hubs*.

Hubness has first been reported in music retrieval and recommendation systems [1], where it is still an important issue [12][15]. Other data domains where hubness was described include textual data, images [37] and time series [33]. Hubness is discussed in more detail in Section 2.1.

A detailed study of the influence of hubness on instance selection for kNN classification is currently lacking. Most recent instance selection surveys have failed to take data dimensionality into account and did not consider the implications of hub selections [27][14]. There has been some recent progress in terms of designing instance selection methods that take data hubness into account [4][9]. These hubness-based instance selection methods are included in our analysis.

1.1. Contributions

The main contribution of this paper is a new way of combining instance selection with k-NN classification. The proposed approach takes prototype hubness into account, estimated in an unbiased way on the training data. This information is then passed on to hubness-aware kNN classifiers.

As hubs are the centers of influence in intrinsically high-dimensional data and greatly determine the outcome of the k-nearest neighbor classification process, we have examined how different instance selection methods handle hub-points and how they affect the overall hubness in the data space. Multiple comparisons were performed on several different kNN-based data reduction techniques.

The main hypothesis that is examined in this paper is that the instance selection bias induces a bias in the hubness estimates derived from the selected data and that this has a detrimental effect on kNN classification. Our experimental results on data from different domains support this hypothesis.

Furthermore, we suggest that it might be possible to overcome this issue by using the recently proposed hubness-aware k-occurrence models [38][40][42]. We have shown that this coupling of instance selection with classification requires the selection methods to output the unbiased *prototype hubness estimates* recalculated on the training data after the initial instance selection phase, alongside with the selected subset. As many existing instance selection methods already calculate kNN graphs on the training data prior to selection, we argue that such recalculations can be done very efficiently in practice.

2. Related work

2.1. Hub points and their influence on classification

Let $D = (x_1, y_1), (x_2, y_2) \dots (x_N, y_N)$ be a set of labeled data points drawn i.i.d. from a joint distribution $p(x, y) = p(x) \cdot p(y|x)$ over $X \times Y$, where X is the feature space and Y the finite label space, $|Y| = C$.

Denote by $D_k(x_i) = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}) \dots (x_{ik}, y_{ik})\}$ the k-neighborhood of x_i. We will say that any $x \in D_k(x_i)$ is a neighbor of x_i and x_i is a reverse neighbor of any $x \in D_k(x_i)$. An occurrence of an element in some $D_k(x_i)$ will be referred to as k-occurrence. The number of k-occurrences of a point x will be denoted by $N_k(x)$. We will sometimes refer to $N_k(x)$ as the *hubness* of x.[3] We will say that a k-occurrence is *good* if the neighbor label matches the label in the point of interest, i.e. $x_{ij} \in D_k(x_i)$ is a good occurrence of x_{ij} if $y_{ij} = y_i$. Similarly, label mismatches define the *bad occurrences* of a neighbor point. The total occurrence count can thus be decomposed into the good an bad occurrence sums as $N_k(x_i) = GN_k(x_i) + BN_k(x_i)$, where GN_k and BN_k represent the good and bad hubness, respectively.

The bad occurrences cause misclassification in kNN methods, therefore a high bad hubness rate is usually a good indicator of the complexity of the classification task. In non-binary classification problems, it is sometimes useful not only to consider bad hubness as a single quantity, but rather to analyze all the class-specific occurrences separately. We will denote by $N_{k,c}(x_i)$ the number of k-occurrences of x_i in neighborhoods of examples that belong to class c.

In high dimensional data, the distribution of $N_k(x)$ becomes highly asymmetric, in a sense that it is skewed to the right. *Skewness*[4] of the k-occurrence distribution is defined as $g_1(N_k(x)) = \frac{m_3(N_k(x))}{m_2^{3/2}(N_k(x))} = \frac{1/n \sum_{i=1}^{N}(N_k(x_i)-k)^3}{(1/n \sum_{i=1}^{N}(N_k(x_i)-k)^2)^{3/2}}$. High positive skewness which is encountered in intrinsically high-dimensional data indicates that the distribution tail is longer on the right distribution side. This is illustrated in Figure 1. In very

[3] The word *hubness* is otherwise used to denote the neighbor occurrence distribution skewness when used in front of a data set or subset. When used in front of a single point x, it denotes the degree to which that point is a hub, which is measured by the point occurrence count, $N_k(x)$.

[4] Skewness of a probability distribution is its 3rd standardized moment and is frequently used in statistical analysis. The 4th moment is kurtosis, which quantifies the steepness of a distribution, and we will not consider it here.

high-dimensional data, the occurrence distribution actually approaches the power law. The example in Figure 1 shows the skewness in Gaussian data, but the phenomenon holds in general and is present in many real-world data sets.

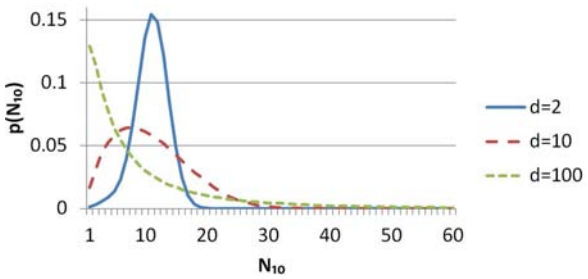


Fig. 1. The change in the distribution shape of 10-occurrences (N_{10}) in i.i.d. Gaussian data with increasing dimensionality when using the Euclidean distance. The graph was obtained by averaging over 50 randomly generated data sets. Hub-points exist also with $N_{10} > 60$, so the graph displays only a restriction of the actual data occurrence distribution.

Formally, we will say that **hubs** are points $x_h \in D$ such that $N_k(x_h) > k + 2 \cdot \sigma_{N_k(x)}$. In other words, their occurrence frequency exceeds the mean (k) by more than twice the standard deviation. We will denote the set of all hubs in D by H_k^D.

Even though the idea of simply removing bad hubs from the data might seem appealing, the problem is not so simple. The removal of hubs creates empty positions in the neighbor lists of their reverse nearest neighbors and these positions are then subsequently filled by other points when the kNN sets are re-calculated. This usually spawns more hubs and in turn, bad hubs as well.

Hubness-aware classification Several hubness-aware k-nearest neighbor methods for high-dimensional data classification have recently been proposed [42][40][38]. They are based on learning k-neighbor occurrence models on the training data, by calculating $N_{k,c}(x_i)$ for all $x \in D$. The idea is that previous occurrences of a neighbor point carry potentially useful information and that this information can be more valuable for predicting the label in the point of interest than the label of the neighbor point itself.

As a trivial example, consider a hub point x_M that belongs to class c_1. Furthermore, assume x_M is mislabeled, so that $y_M = c_2$ and that x_M belongs to a class interior of c_1. Most likely, there will be a label mismatch in all of x_M-s k-occurrences. Some of those mismatches might even induce misclassification. However, if previous occurrences of that point were known, it is possible to consider an occurrence of x_M as partial evidence towards having label $y = c_1$ in the query point, instead of $y = c_2$. Since hubs occur very frequently and most neighbor occurrences in high-dimensional data are hub occurrences, there are often plenty of past occurrences to learn from.

Instance selection can may cause detrimental hub points to emerge. Hubness-aware classification should therefore be considered for usage in conjunction with instance selection, instead of traditional k-nearest neighbor approaches. This idea is discussed in more detail in Section 3 and an experimental evaluation is presented in Section 5.

Several hubness-aware k-nearest neighbor classification approaches based on learning from neighbor occurrence models on the training data have recently been proposed. It is possible to incorporate hubness-based weights during voting, as in hw-kNN [34]. Class-conditional neighbor occurrences can be used to derive fuzzy votes for the fuzzy k-nearest neighbor voting framework, as proposed in h-FNN [42]. This has been extended to include neighbor occurrence self-information in vote weighting in HIKNN [38]. Alternatively, the Naive Hubness-Bayesian k-nearest neighbor (NHBNN) [40] presents a Bayesian re-interpretation of the k-nearest neighbor rule where neighbor occurrences are treated as random events. Class affiliation probabilities in NHBNN are determined as per Equation 1, where N is the data size, $n_c = |\{x_i : y_i = c\}|$ is the size of class c and λ is a smoothing parameter.

$$p(y_i = c|D_k(x_i)) \propto p(y_i = c) \prod_{t=1}^{k} p(x_{it} \in D_k(x)|y = c) = \frac{n_c}{N} \prod_{t=1}^{k} \frac{N_{k,c}(x_{it}) + 1 + \lambda}{n_c \cdot (k+1) + \lambda N} \tag{1}$$

An evaluation of the feasibility of using these hubness-aware classification approaches in conjunction with instance selection is discussed in Section 5.3.

2.2. Data Reduction

Prototype selection for k-nearest neighbor classification is a frequently used data preprocessing technique and many methods have been proposed over the years [14][23]. *Edition* methods try to eliminate noisy instances and *wrapper* methods try to preserve classifier accuracy by removing superfluous examples. Many methods are hybrid, as they try to achieve both goals, to a degree. This division reflects some fundamental differences in prototype selection strategies, as the edition methods seek to remove the border points, while the wrappers usually perform condensation by keeping precisely such points which are close to decision boundaries [14]. According to what is reported in the literature, good results can be obtained either by keeping or rejecting the border points and either by keeping or rejecting the central points. There is no unified approach and it is clear that the best strategy is data-dependant.

Regardless of the border point selection/rejection strategies, the methods which seek to safely reduce the data size often in fact aim at maximizing the *coverage* of points by their selected k-nearest prototypes [4]. Set coverage is an NP-complete problem. The prototype selection problem was shown to be equivalent to the set coverage problem, suggesting that one should apply approximate and heuristic methods.

We have considered several well known selection strategies, as well as a few very recent ones. Random sampling will be used as a baseline. Any complex, time-consuming method ought to perform at least as well as random sampling if it were to justify its use. Additionally, random sampling is unbiased, which fits the purpose of our comparisons quite well.

The other approaches we considered in this study are ENN [43], CNN [30], GCNN [7], RT3 [44], AL1 [9] and INSIGHT [4].

ENN: One of the first proposed approaches was the *edited nearest neighbor* (ENN) [43]. It keeps the examples which are correctly classified by the kNN rule on the training

data, k usually being set to 3 or 5. In high-dimensional data, there is no guarantee that bad hubs will be misclassified on the training data. Therefore, ENN might select points that would cause severe misclassification.

CNN: The *condensed nearest neighbor* (CNN) [30] method is an incremental procedure which retains at each step an instance if it is misclassified by the current prototype set. As outliers are often misclassified, this procedure retains most of the noise in the data and its reduction rate is not very high.

GCNN: A generalized CNN approach applies a strong absorption rule [7]. GCNN retains more examples than CNN, usually leading to a better accuracy. As with both ENN and CNN, there is no guarantee that the selected points would exhibit good hubness.

RT3: Another classic instance pruning technique is the RT3 rule presented in [44]. In the first pass, noisy instances are removed by a rule similar to ENN. The remaining points are sorted by the distance to their *nearest enemy* and then iteratively removed if their removal does not increase misclassification in the set of their reverse nearest neighbors. RT3 achieves very good data reduction. However, as it uses ENN-like noise filtering approach, it can lead to suboptimal selection sets, as some good hubs might be filtered out in the first pass.

AL1: Unlike the above outlined methods, AL1 [9] is a selection rule based on reverse-neighbor sets. A point x_i is retained if it is a reverse neighbor to at least one other point, assuming that x_i had not previously been covered by an already selected point.

INSIGHT: A hubness-aware selection strategy for time series classification was recently proposed [4]. INSIGHT takes into account the good and bad k-occurrences of each instance, and then chooses a previously specified number of instances as prototypes.

There are many other approaches as well, that are beyond the scope of the study that is presented in this paper. Genetic algorithms are a common approach [20]. Influence of prototype selection on future query quality has been examined in [48]. Special selection techniques have been applied to learning under class imbalance [31]. Hybrid selection methods have also been considered [5].

The instance selection methods evaluated in this paper all base their selection criteria on information obtained by analyzing k-neighbor sets. This allows us to implement our proposed approach (Section 3) with minimal / negligible overhead in terms of time-complexity. This is not an unreasonable assumption in practice, as many instance selection methods are tailored precisely for kNN classification.

3. The proposed approach

3.1. General outline

During instance selection, the original training set D is decomposed into two disjoint subsets, the set of selected and rejected examples, S and R respectively. We will use α to denote the selection rate $\alpha = \frac{|S|}{|D|}$. Traditionally, only S is used in the subsequent classifier training, while R is disregarded completely. What we essentially propose is to use $D = S \cup R$ for prototype occurrence modeling, i.e. hubness-aware classifier training, while only considering the prototypes $x \in S$ as potential neighbors. There is a way to do this with minimal overhead, in those selection methods which rely on k-nearest neighbor sets.

The reason for this lies not only in the fact that $D = S \cup R$ is a larger set and hence might lead to better estimates for classifier models, but rather that instance selection methods induce a bias when constructing S and data distribution there differs from the original data distribution. Therefore, calculating a kNN graph on S only would not be guaranteed to lead to a good neighbor occurrence model, as it would not necessarily be able to predict how prototype points $x \in S$ would occur on the test data, due to a difference in data distributions.

The proposed selection process is outlined in Figure 2, where the instance selection phase is extended by including the unbiased prototype hubness estimation, followed by hubness-aware k-nearest neighbor classification.

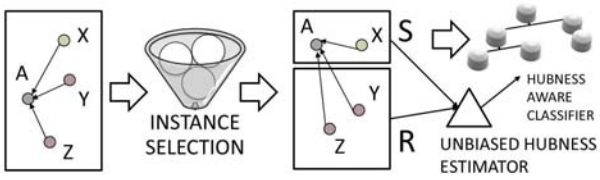


Fig. 2. The modified instance selection pipeline. An unbiased prototype occurrence profile estimator is included between the instance selector and a hubness-aware classifier. It ought to provide more reliable hubness estimates to the hubness-aware occurrence models. In the example we see that point A is a neighbor to three other points (X,Y,Z), but only one of them gets selected. Hence, some occurrence information is irretrievably lost.

Let the unbiased **prototype hubness** training estimate for a given selected set S be the relative neighbor occurrence frequency of $x \in S$ when only $x \in S$ are considered as potential neighbors to points in $x \in S \cup R$. For each instance $x \in S \cup R$, we calculate its nearest neighbors from S. Note that these prototype occurrence frequencies might differ significantly from their frequencies prior to instance selection. The rejected points $x_i \in R$ are put in a tabu-list and are not considered as potential neighbors.

Let $x_i \in S$ be a prototype point. Denote by $N_k^P(x_i)$, $N_{k,c}^P(x_i)$, $GN_k^P(x_i)$ and $BN_k^P(x_i)$ the unbiased hubness quantities: prototype occurrence frequency, prototype class hubness, prototype good hubness and prototype bad hubness, respectively.

Since all $x \in D$ are required for the unbiased prototype hubness training estimate, hubness-aware classifiers would not be able to perform these calculations internally if only S is provided to them after instance selection. This is why the proposed framework includes an intermediate step where the selection methods output the calculated unbiased prototype occurrence frequencies in a separate object.

In order to measure the extent of the selection bias, it is necessary to compare the unbiased prototype hubness training estimate to its biased counterpart. Let the **prototype pseudo-hubness** be the *biased* estimate inferred only from S. We will use $N_k^S(x_i)$, $N_{k,c}^S(x_i)$, $GN_k^S(x_i)$ and $BN_k^S(x_i)$ to denote the pseudo-hubness, class-specific pseudo-hubness, pseudo-good hubness and pseudo-bad hubness of $x_i \in S$, respectively.

The only case in which the pseudo-hubness quantities are themselves unbiased is when the instance selection is entirely random. However, even though random sampling is unbiased, there remains an issue of reliability of the restricted prototype estimates,

as they are inferred from a smaller sub-sample. The *standard error* of a probability estimate p is $\sqrt{\frac{p(1-p)}{n}}$, where n is the number of observations it is derived from. When estimating the class-specific neighbor occurrence profiles, $p(y = c|x_i \in D_k(x))$ is required, and there the number of observations is actually $n = N_k(x_i)$. Therefore, the expected error is proportional to the reciprocal of the square root of point hubness. However, $\sum_{x_i \in S} N_k^S(x_i) = k|S|$ and $\sum_{x_i \in S} N_k^P(x_i) = k|D|$. Therefore, $E(N_k^S(x_i)) = k$, while $E(N_k^P(x_i)) = k\frac{|D|}{|S|}$. Even in random sampling, $N_k^P(x_i)$ prototype hubness scores would be able to deliver better estimates by a factor of $\sqrt{\frac{|D|}{|S|}}$.

3.2. Scalability

Many kNN instance selection methods build an entire kNN graph on the training data during the instance selection phase. In order to calculate all the $N_k^P(x_i)$ and $N_{k,c}^P(x_i)$, these neighbor lists need to be modified so that they only contain members of S, the selected prototypes. This is easily achieved. First, all $x \in R$ are removed from the neighbor sets, which are then shifted to the left. The remaining positions in each $D_k(x)$ are then filled by considering all $\{x : x \in S \setminus D_k(x)\}$. This is illustrated in Figure 3.

As instance selection methods try to select relevant points, we would expect many of the occurrences in the original kNN graph to originate from the selected prototypes. In the worst case, calculating the entire prototype-restricted kNN graph is still $\frac{|D|}{|S|}$ times faster than calculating the training kNN graph, which doesn't increase the overall complexity.

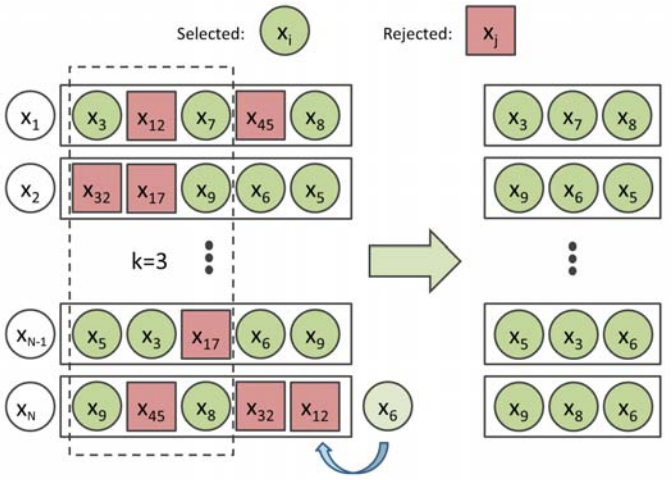


Fig. 3. The existing k-nearest neighbor lists on the training set $D = S \bigcup R$ are easily modified to obtain the unbiased prototype hubness estimates. The rejected examples are removed from the neighbor sets and the remaining neighbors are shifted to the left. It is possible to use different neighborhood sizes for instance selection and classification, which would significantly reduce the number of remaining calculations. In some cases, partial nearest neighbor queries might be needed to fill in the last few remaining positions.

Most existing hubness-aware classifiers [40][38][42] do not require additional training once provided with all the $N_{k,c}^{P}(x_i)$ values, or perform additional calculations that are linear in data size. This means that replacing basic kNN with its hubness-aware extensions does not increase the time complexity of the classification pipeline.

Instead of calculating a complete kNN graph exactly, which is not feasible for big data where there are millions of examples, it is possible to rely on fast approximate methods that produce fairly accurate approximations in reasonable time [6][29].

4. Data

We have compared the selected instance pruning methods and evaluated our proposed approach on several data domains. The benchmark contains quantized image representations, time series and class imbalanced high-dimensional Gaussian mixtures. An overview of important hubness-related properties of the data is given in Table 1. Manhattan distance was used for image representations, Euclidean on the Gaussian mixtures and dynamic time warping (DTW) on time series. Image and Gaussian data exhibits substantial hubness. The analyzed time series are of low-to-medium hubness.

Table 1. Overview of the datasets. Each dataset is described by its size, dimensionality, the number of categories, skewness of the N_k distribution (S_{N_k}), proportion of bad k-occurrences BN_k, the number of hubs ($|H_k^D|$), as well as the degree of the major hub. The neighborhood size of $k = 10$ was used in all experiments.

Data set	size	d	C	$S_{N_{10}}$	BN_{10}	$\lvert H_{10}^{D}\rvert$	$\max N_{10}$
iNet3	2731	416	3	4.61	26.1%	76	750
iNet4	6054	416	4	10.77	48.1%	137	906
iNet5	6555	416	5	7.42	50.3%	170	1635
iNet6	6010	416	6	4.32	56.9%	245	1834
iNet7	10544	416	7	5.56	55.0%	343	1638
GM_1	10785	100	20	4.40	41.4%	439	272
GM_2	8849	100	20	5.12	45.6%	319	274
GM_3	8102	100	20	5.35	40.0%	315	323
GM_4	11189	100	20	5.97	45.0%	509	338
GM_5	9859	100	20	5.32	49.2%	361	306
GM_6	10276	100	20	9.19	42.9%	291	500
GM_7	12572	100	20	6.80	45.3%	434	420
GM_8	8636	100	20	8.33	48.5%	256	517
GM_9	9989	100	20	5.26	53.0%	375	289
GM_{10}	9330	100	20	6.12	45.4%	320	357
50words	905	270	50	0.66	36.2%	38	33
Adiac	781	176	37	0.36	51.8%	20	28
Cricket X	780	300	12	0.38	33.1%	22	28
Cricket Y	780	300	12	0.46	34.9%	28	30
Cricket Z	780	300	12	0.37	33.2%	27	27
ECGFiveDays	884	136	2	0.00	3.6%	18	25
Haptics	463	1092	5	0.85	60.9%	20	35
InlineSkate	650	1882	7	0.42	52.3%	24	28
ItalyPowerDemand	1096	24	2	0.83	5.1%	46	46
MedicalImages	1141	99	10	0.35	31.6%	33	26

4.1. High hubness test data

The analyzed image datasets were taken as subsets of the ImageNet online repository[5], processed as quantized and normalized SIFT feature representations [37][24], enriched by color histogram information [47]. They exhibit high overall hubness, as well as high bad hubness.

The Gaussian mixture data was generated with a specific intent to pose great difficulties for k-nearest neighbor methods. Let μ_c and σ_c be the d-dimensional mean and standard deviation vectors of a hyper-spherical Gaussian class $c \in 1..C$ on a synthetic Gaussian mixture data set. The covariance matrices of the generated classes were diagonal for simplicity, i.e. the attributes were independent and the i-th entry in σ_c signifies the independent dispersion of that synthetic feature. For the first class, the mean vector was set to zeroes and the standard deviation vector was generated randomly. Each subsequent class c was randomly paired with one prior Gaussian class, which we will denote by $\bar{c}$, so that some overlap between the two was assured. For each dimension $i \in 1..d$ independently, μ_c was set to $\mu_c \approx \mu_{\bar{c}} \pm \beta \cdot \sigma_{\bar{c}}$ with equal probability, where $\beta = 0.75$. Additionally, dispersion was updated by the following rule: $\sigma_c = \gamma \cdot \sigma_{\bar{c}} + (\gamma - \beta) \cdot Z \cdot \sigma_{\bar{c}}$, where $\gamma = 1.5$ and Z is a uniform random variable defined on $[0, 1]$. Each class was set to be either a minority class or a majority class and the class sizes ranged from 20 to 1000, each being randomly determined either in the upper $[700, 1000]$ or the lower $[20, 170]$ interval of the range. All 10 compared synthetic datasets were set to be 100-dimensional and to contain 20 different classes.

Most datasets in Table 1 are quite challenging for kNN classification even prior to instance selection, especially image and Gaussian data that have about 50% of label mismatches in k-neighbor sets.

The degree of the major hub shows us how some individual points permeate surprisingly many k-neighbor sets. For example, in iNet6, the major hub appears in 30.5% of all query results. It often induces label mismatches and misclassification.

4.2. Low hubness test data

Instance selection methods are potentially quite useful for time series classification, due to a high time complexity of calculating all dynamic time warping (DTW) distances between pairs of time series. Performing instance selection reduces the number of distance calculations in future queries, which leads to a significant speed-up and improves overall scalability. DTW can be interpreted as an edit distance [22] and DTW calculation process can be viewed as a process of transforming one time series into another via cost-sensitive elongations and replacements. DTW calculates the transformation with minimal cost and the cost represents the distance between the time series. DTW can be calculated via dynamic programming [35]. It is possible to speed up the calculations by restricting possible index differences between compared time series subsequences to a fixed 'warping window' [18]. Recent research suggests that setting the warping window width to a relatively small value such as 5% of the time series length, does not negatively affect the classification accuracy and might even lead to some improvements [35]. We have used this approach in our experiments.

[5] http://www.image-net.org/

In order to evaluate our approach on time-series data, we used publicly available real-world datasets from the UCR repository[6]. Here we report the results on 10 representative datasets, namely: 50words [36], Adiac [17], Cricket X [21], Cricket Y, Cricket Z, ECGFiveDays, Haptics [25] InlineSkate [26], ItalyPowerDemand [19], MedicalImages. Similar trends exist in other cases as well.

The examined time series datasets do not exhibit substantial hubness and there are no major hubs in the data. This is a consequence of the fact that the intrinsic dimensionality of time series data is usually much lower than its embedding dimensionality due to correlations between subsequent measurements and signals. Therefore, negative aspects of the dimensionality curse are not as pronounced.

5. Experiments

The evaluation of classification under instance selection was performed as 10-times 10-fold cross validation. Statistical significance was tested using the corrected re-sampled t-test to compensate for dependencies between the runs. A standard neighborhood size of $k = 10$ was used in all experiments.

The selection rate for INSIGHT and baseline random sub-sampling was set to $\alpha = 0.1$. We are interested mainly in instance selection methods that can significantly reduce the data size. The GCNN rule requires a strong absorption parameter. The default value that was proposed in the original paper did not perform well in the high-dimensional case, as it was selecting almost the entire datasets. A value of 0.1 was used instead, after performing a series of initial trials, resulting in a more reasonable reduction rate.

Section 5.1 presents the evaluation of the influence of instance selection on data hubness, as well as an examination of hub selection bias of different selection strategies. The difference between biased and unbiased prototype hubness training estimates is shown in Section 5.2 and the benefits of the proposed hubness-aware instance selection framework are confirmed by an experimental evaluation of kNN classification performance under instance selection in Section 5.3.

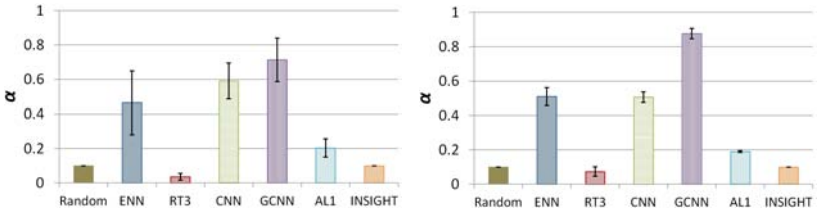
5.1. Hubs and Instance Selection

In intrinsically high-dimensional data, hubs arise as most influential points in k-nearest neighbor classification. In case they are selected as prototypes, the selection will tend to preserve the original distribution of influence. Failing to select major data hubs can potentially induce substantial changes in kNN structure with unpredictable consequences that might be beneficial or detrimental.

The overall selection rate for the examined methods on high-hubness data is shown in Figure 4. Random sub-sampling and INSIGHT have been pre-set to fixed selection rates of $\alpha = 0.1$. RT3 selects very small prototype subsets. On the other hand, GCNN displays a low reduction rate and retains most points. ENN and CNN achieve selection rates of about 50% on this data.

While some hubs are retained by the examined selection strategies, many hubs also get rejected in the process, as shown in Figure 5. The highest proportion of hubs get selected

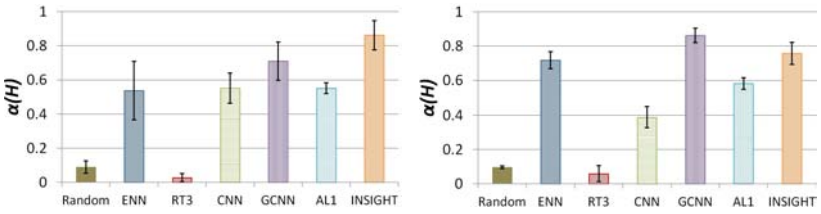
[6] http://www.cs.ucr.edu/~eamonn/time_series_data/



(a) iNet (b) Gaussian Mixtures

Fig. 4. Average selection rate α of the examined instance selection methods.

in INSIGHT and GCNN, while Random and RT3 select the fewest among the original hubs.



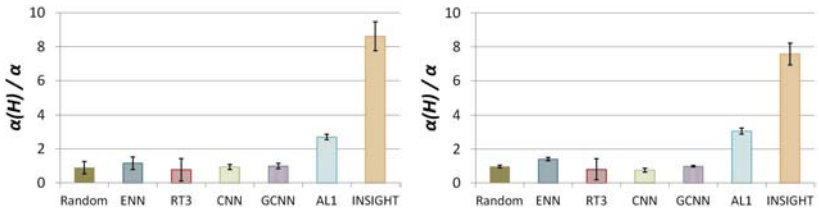
(a) iNet (b) Gaussian Mixtures

Fig. 5. Average hub selection rate $\alpha(H)$ of different instance selection methods. A higher rate implies a preservation of the distribution of influence.

Figure 6 shows the hub selection rates normalized by the overall selection rates. Values greater than 1 indicate a positive preference for selecting hub points. INSIGHT and AL1 select a very high proportion of hub points, as they are based on analyzing reverse neighbor sets. ENN is the only remaining method which achieves a hub selection rate significantly higher than random, on Gaussian mixtures, about 1.4. On the examined Gaussian data, the hub selection rate of CNN is even significantly lower than random.

The fact that many of the original hubs are frequently not being selected by the examined strategies suggests that applying such instance selection prior to kNN classification is expected to have a significant impact on classification performance, either by increasing or decreasing the original classification accuracy.

Dependency on Neighborhood Size Selecting the optimal neighborhood size in k-nearest neighbor methods is a complex issue. It is important to consider neighborhoods that are large enough to compensate for noise but also small enough not to breach the locality assumption across the data space. It is possible to use cross-validation [28] or other techniques to determine good candidate k-values on training data. As an in-depth analysis of the neighborhood size selection problem is beyond the scope of this study, most experiments in Section 5.2 and Section 5.3 were performed for a fixed, pre-determined



(a) iNet (b) Gaussian Mixtures

Fig. 6. Averaged normalized hub selection rate $\alpha(H)$ of different instance selection methods. A number close to 1 implies that the hub selection rate does not differ from that of random subsampling.

neighborhood size. Here we will briefly examine the consequences of varying neighborhood sizes.

As Figure 7 shows, the examined instance selection methods do not appear to be very sensitive to the choice of k, in terms of hub selection rates which remain stable in the tested intervals. The fluctuations in ENN most probably stem from tie resolution in kNN classification and the tie resolution strategy. As for AL1, its hub selection rate decreases monotonously with k, as more and more hubs get covered by other hubs and are therefore rejected by the algorithm. Overall, similar k values tend to produce similar results and the relative ordering of the methods with respect to hub selection remains invariant.

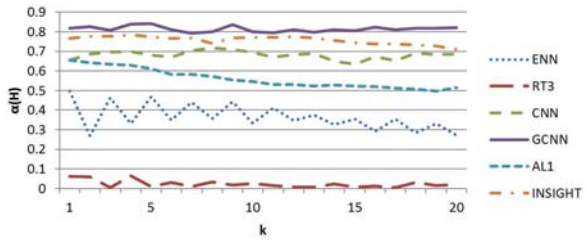


Fig. 7. The stability of hub selection rates of different instance selection methods under changing neighborhood sizes, calculated on the iNet6 dataset.

Prototype occurrence skewness The skewness of the prototype k-occurrence distribution can differ substantially from the hubness of the data prior to instance selection. As Figure 8 shows, different selection methods induce different degrees of prototype hubness. CNN and GCNN induce the highest prototype set hubness among the compared approaches across different data domains. Only among time series data does random subsampling induce a higher skewness of the prototype k-occurrence distribution. INSIGHT achieves the lowest prototype occurrence skewness, as it rejects anti-hubs and orphans, which reduces the hubness of the data.

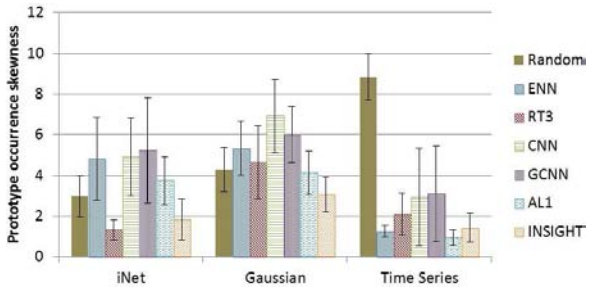


Fig. 8. Average unbiased skewness in the prototype occurrence distributions, SN_k^P, given for different instance selection methods.

Some approaches are inconsistent in the way in which they change data hubness, like ENN and RT3. On certain datasets they increase it while on others they decrease the overall data hubness.

5.2. Biased and Unbiased Hubness Estimates

As instance selection methods incorporate a selection bias, calculating the hubness of the selected prototypes within the selected subset alone ($N_k^S(x_i)$, $N_{k,c}^S(x_i)$, $GN_k^S(x_i)$,$BN_k^S(x_i)$) yields biased pseudo-hubness estimates. This fact was not given much thought prior to hubness-aware kNN methods, as other kNN classifiers do not use these quantities explicitly in classification. Here we examine the consequences of using the biased estimates calculated only on the prototype set S.

Figure 9 shows high regularity in estimating label mismatch percentages on image data. In ENN, RT3 and INSIGHT there is a consistent underestimation of the actual bad influence of the selected prototypes. In CNN and GCNN there seems to be a consistent overestimation of bad hubness. Underestimating bad hubness might be potentially much more dangerous than overestimating it, as it might cause the models to favor certain hub points that might actually turn out to be bad hubs. This might increase the misclassification rate.

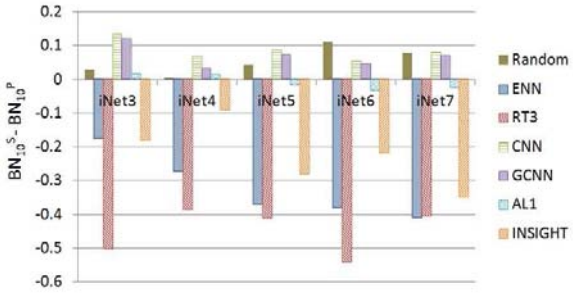


Fig. 9. The difference between the pseudo-bad hubness estimated on the set of selected instances S and the actual prototype bad hubness estimated on the entire training set.

Figure 10 gives a more detailed insight into the consequences of the selection bias, as it shows the average point-wise error in estimating future bad occurrence probabilities. The lowest average bad hubness estimation error is achieved by CNN and GCNN. RT3 displays a very high bad hubness estimation error rate and even INSIGHT and ENN exhibit non-negligible estimation error rates on several datasets. These occurrence estimates were calculated for $k = 10$. In subsequent classification, these errors accumulate when individual votes are factored in the final decision.

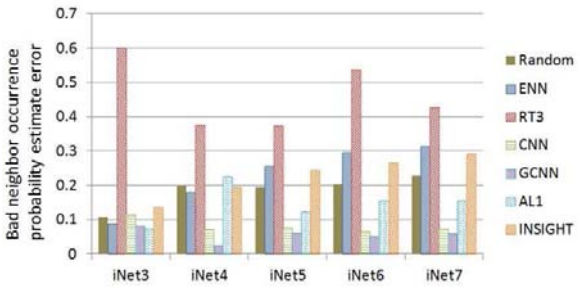


Fig. 10. The average absolute difference in estimating the bad 10-occurrence probabilities of individual prototype points on ImageNet data, where $Err_{AVG}^{p(BN_{10}^S)} = E_{\{x:N_{10}^S(x)>0 \vee N_{10}^P(x)>0\}}(|\frac{BN_{10}^S(x)}{N_{10}^S(x)} - \frac{BN_{10}^P(x)}{N_{10}^P(x)}|)$.

Even though the absolute and relative $BN_k^S(x)$ differ notably from $BN_k^P(x)$, prototype neighbor points mostly retain their general class hubness tendencies. There is a high average correlation between $N_{k,c}^S(x)$ and $N_{k,c}^P(x)$ for $c \in C$, as can be seen in Figure 11. The correlation is only low in case of RT3.

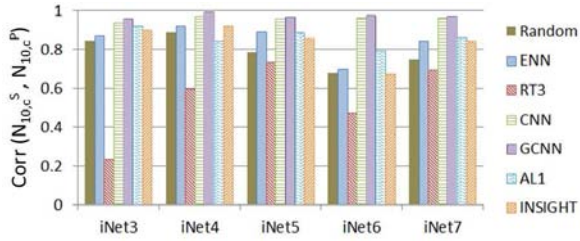


Fig. 11. Average Pearson correlation between class hubness tendencies of prototype neighbor points for the compared selection methods on ImageNet data.

Unlike bad hubness, no regularity can be seen in underestimating or overestimating the skewness of the prototype occurrence distribution itself, as shown in Figure 12. Therefore, SN_k^S is not a viable substitute for SN_k^P.

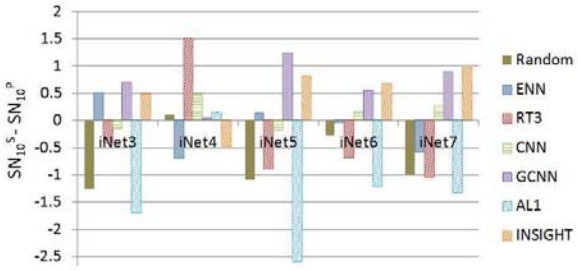


Fig. 12. The difference between the pseudo-hubness estimated on S and the prototype occurrence skewness estimated on the entire training set. There is no apparent regularity, which means that very little can be discerned from observing pseudo-hubness of prototypes on a single dataset, as one can not even know with certainty whether the estimate exceeds the actual data hubness or underestimates it instead.

5.3. Classification Experiments

We have tested the biased and unbiased hubness estimates within several different hubness-aware classifiers and occurrence models: hw-kNN [34], h-FNN [42], NHBNN [40] and HIKNN [38]. The relative classifier ranks based on the achieved accuracy with no instance selection are shown in Figure 13 and a comparison of classifier ranks in case of biased and unbiased neighbor occurrence estimates for various instance selection methods is given in Figure 14. Improvements can be seen in the overall average accuracy as well.

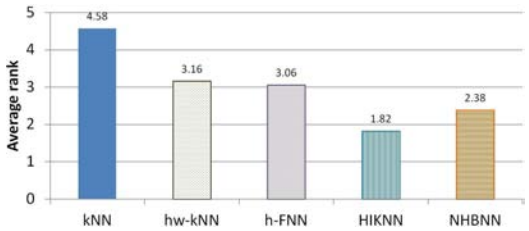
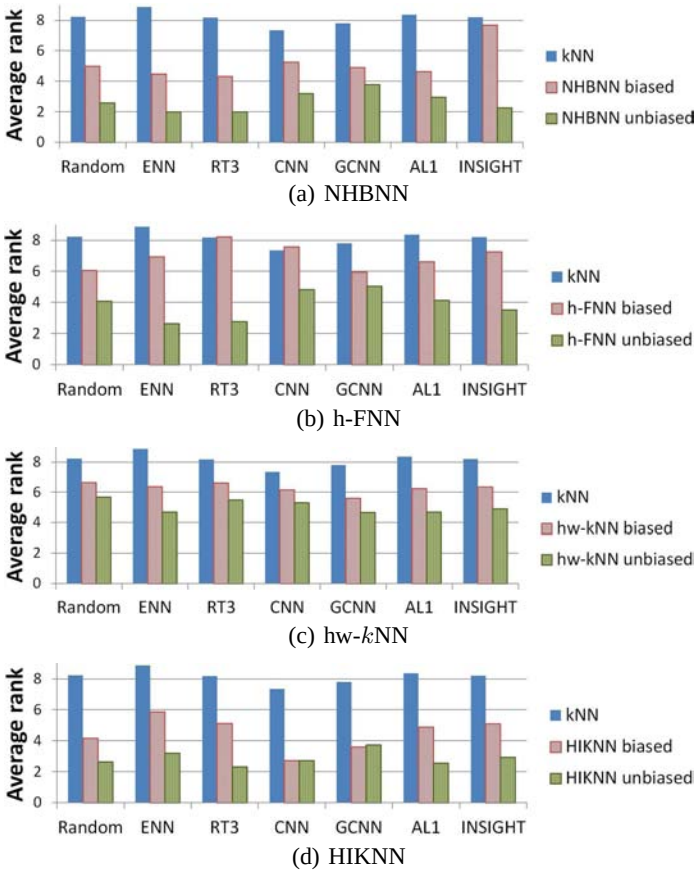


Fig. 13. Average relative classifier ranks of kNN, hw-kNN, HIKNN, h-FNN and NHBNN with no instance selection on the tested datasets. Lower ranks correspond to better performance.

Using the proposed unbiased hubness estimation framework significantly improves the performance of all examined hubness-aware classification methods. Most improvements are present for selection strategies with a lower selection rate. This is natural, since a smaller sample allows for a higher bias in learning neighbor occurrence models and also provides less data to estimate the class-conditional neighbor occurrence frequencies from. These results confirm our initial hypothesis, that the instance selection bias reflects negatively on neighbor occurrence models in hubness-aware k-nearest neighbor classifiers and that using an unbiased estimate leads to better results.

Correcting the instance selection bias in neighbor occurrence estimation yields smallest improvements in case of hw-kNN, as it does not differentiate class-conditional neigh-



(a) NHBNN

(b) h-FNN

(c) hw-kNN

(d) HIKNN

Fig. 14. Average relative classifier ranks over all tested datasets for all tested instance selection methods and classification algorithms, in case of biased and unbiased neighbor occurrence estimation.

bor occurrence frequencies, rather relying on aggregate concepts of good and bad hubness.

The two instance selection strategies that favor hubs, AL1 and INSIGHT, perform approximately as well as random sub-sampling under the investigated selection rates. This suggests that they might not be selecting the most appropriate hub-points. AL1 simply tries to select a very small number of points that maximize coverage, so it is possible that many of the selected hub-points are in fact bad hubs and are causing misclassification.

6. Conclusions and Future Work

This paper examines the role of hubs in instance selection for kNN classification and proposes a new framework for coupling instance selection with hubness-aware k-nearest neighbor classification for classifying intrinsically high-dimensional data.

Several standard kNN-based selection strategies have been examined: random subsampling, ENN, RT3, CNN, GCNN, AL1 and INSIGHT. The initial analysis has shown that different selection strategies exhibit different hub selection rates and preferences. GCNN and INSIGHT select most hubs on average. ENN, AL1 and INSIGHT exhibit a general preference for selecting hubs as prototypes. Consequently, different instance selection methods affect the selected prototype hubness in different ways.

Each selection strategy embodies a certain bias regarding the points it retains as prototypes. This can lead to underestimation or overestimation of the potential future bad influence of some selected hub points and reduce the subsequent classification performance. We have proposed to use an unbiased hubness estimate in conjunction with the hubness-aware classification models as a way to overcome this deficiency.

We have examined the classification accuracy and rank of HIKNN, NHBNN, hw-kNN, h-FNN and the baseline kNN under biased and unbiased hubness estimates for all 7 compared instance selection methods, on 25 real-world and synthetic datasets. An extensive experimental evaluation has shown promising improvements when the unbiased hubness estimate is used. This confirms our initial hypothesis, that the instance selection bias reflects negatively on classifiers that build neighbor occurrence models from the selected prototype set.

In future work, we intend to use these initial discoveries in order to design new and better instance selection methods for intrinsically high-dimensional data.

Acknowledgements

Research partially performed within the framework of the grant of the Hungarian Scientific Research Fund - OTKA 111710 PD. This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

1. Aucouturier, J., Pachet, F.: Improving timbre similarity: How high is the sky? Journal of Negative Results in Speech and Audio Sciences 1 (2004)
2. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR'08. pp. –1–1 (2008)
3. van den Bosch, A., Weijters, T., Herik, H.J.V.D., Daelemans, W.: When small disjuncts abound, try lazy learning: A case study (1997)
4. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: 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. pp. 149–160. PAKDD'11, Springer-Verlag, Berlin, Heidelberg (2011)
5. Caises, Y., González, A., Leyva, E., Pérez, R.: Combining instance selection methods based on data characterization: An approach to increase their effectiveness. Inf. Sciences 181(20), 4780–4798 (Oct 2011)
6. Chen, J., ren Fang, H., Saad, Y.: Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection. Journal of Machine Learning Research 10, 1989–2012 (2009)
7. Chou, C.H., Kuo, B.H., Chang, F.: The generalized condensed nearest neighbor rule as a data reduction method. In: Proceedings of ICPR. pp. 556–559. IEEE Computer Society, Washington, USA (2006)

8. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory IT-13(1), 21–27 (1967)
9. Dai, B.R., Hsu, S.M.: An instance selection algorithm based on reverse nearest neighbor. In: Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part I. pp. 1–12. PAKDD'11, Springer-Verlag, Berlin, Heidelberg (2011)
10. Durrant, R.J., Kabán, A.: When is 'nearest neighbour' meaningful: A converse theorem and implications. Journal of Complexity 25(4), 385–397 (2009)
11. Fix, E., Hodges, J.: Discriminatory analysis, nonparametric discrimination: consistency properties. Tech. rep., USAF School of Aviation Medicine, Randolph Field (1951)
12. Flexer, A., Gasser, M., Schnitzer, D.: Limitations of interactive music recommendation based on audio content. In: Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound. pp. 13:1–13:7. AM '10, ACM, New York, NY, USA (2010)
13. François, D., Wertz, V., Verleysen, M.: The concentration of fractional distances. IEEE Transactions on Knowledge and Data Engineering 19(7), 873–886 (2007)
14. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell. 34(3), 417–435 (Mar 2012)
15. Gasser M., Flexer A., S.D.: Hubs and orphans - an explorative approach. In: Proceedings of the 7th Sound and Music Computing Conference. SMC'10 (2010)
16. Gemmell, J., Schimoler, T., Ramezani, M., Mobasher, B.: Adapting K-Nearest Neighbor for Tag Recommendation in Folksonomies. In: Anand, S.S., Mobasher, B., Kobsa, A., Jannach, D., Anand, S.S., Mobasher, B., Kobsa, A., Jannach, D. (eds.) ITWP. CEUR Workshop Proceedings, vol. 528. CEUR-WS.org (2009)
17. Jalba, A., Wilkinson, M., Roerdink, J., Bayer, M., Juggins, S.: Automatic diatom identification using contour analysis by morphological curvature scale spaces. Machine Vision and Applications 16, 217–228 (2005)
18. Keogh, E., Ratanamahatana, C.: Exact Indexing of Dynamic Time Warping. Knowledge and Information Systems 7(3), 358–386 (2005)
19. Keogh, E., Wei, L., Xi, X., Lonardi, S., Shieh, J., Sirowy, S.: Intelligent icons: Integrating lite-weight data mining and visualization into gui operating systems. In: Data Mining, 2006. ICDM '06. Sixth International Conference on. pp. 912 –916 (dec 2006)
20. Kim, K.j.: Artificial neural networks with evolutionary instance selection for financial forecasting. Expert Syst. Appl. 30(3), 519–526 (Apr 2006)
21. Ko, M.H., West, G., Venkatesh, S., Kumar, M.: Using dynamic time warping for online temporal fusion in multisensor systems. Information Fusion 9(3), 370 – 388 (2008), special Issue on Distributed Sensor Networks
22. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In: Soviet Physics Doklady. vol. 10, pp. 707–710 (1966)
23. Liu, H.: Instance Selection and Construction for Data Mining. Springer-Verlag, Berlin, Heidelberg (2010)
24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91 (Nov 2004)
25. Malek, B., Orozco, M., Saddik, A.E.: Novel shoulder-surfing resistant haptic-based graphical password. In: Proceedings of EuroHaptics06 (2006)
26. Mörchen, F., Ultsch, A., Hoos, O.: Discovering interpretable muscle activation patterns with the temporal data mining method. In: Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) Knowledge Discovery in Databases: PKDD 2004, Lecture Notes in Computer Science, vol. 3202, pp. 512–514. Springer Berlin Heidelberg (2004)
27. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Kittler, J.: A review of instance selection methods. Artif. Intell. Rev. 34(2), 133–143 (Aug 2010)
28. Paik, M., Yang, Y.: Combining nearest neighbor classifiers versus cross-validation selection. In: Statistical Applications in Genetics and Molecular Biology 3 (1) (2004) article 12 (2004)

29. Paulevé, L., Jégou, H., Amsaleg, L.: Locality sensitive hashing: A comparison of hash function types and querying mechanisms. Pattern Recogn. Lett. 31(11), 1348–1358 (2010)
30. PE, H.: The condensed nearest neighbor rule. IEEE Trans Inf Theory 14, 515–516
31. Pérez-Rodríguez, J., De Haro-García, A., Garcá-Pedrajas, N.: Instance selection for class imbalanced problems by means of selecting instances more than once. In: Proceedings of the 14th international conference on Advances in artificial intelligence: spanish association for artificial intelligence. pp. 104–113. CAEPIA'11, Springer-Verlag, Berlin, Heidelberg (2011)
32. Radovanović, M., Nanopoulos, A., Ivanović, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research 11, 2487–2531 (2011)
33. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Time-series classification in many intrinsic dimensions. In: SDM'10. pp. 677–688 (2010)
34. Radovanović, M., Nanopoulos, A., Ivanović, M.: Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In: Proc. 26th Int. Conf. on Machine Learning (ICML). pp. 865–872 (2009)
35. Ratanamahatana, C., Keogh, E.: Everything You Know about Dynamic Time Warping is Wrong. In: SIGKDD International Workshop on Mining Temporal and Sequential Data (2004)
36. Rath, T., Manmatha, R.: Word Image Matching using Dynamic Time Warping. In: Proceedings of the IEEE Computer Society Conference on CVPR Conference. vol. 2, pp. 512–521. IEEE (2003)
37. Tomašev, N., Brehar, R., Mladenić, D., Nedevschi, S.: The influence of hubness on nearest-neighbor methods in object recognition. In: Proceedings of the 7th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). pp. 367–374. IEEE, New York, NY, USA (2011)
38. Tomašev, N., Mladenić, D.: Nearest neighbor voting in high dimensional data: Learning from past occurrences. Computer Science and Information Systems 9(2) (2012)
39. Tomašev, N., Mladenić, D.: Hub co-occurrence modeling for robust high-dimensional knn classification. In: Proceedings of the ECML conference. pp. 643–659. Springer-Verlag, Berlin, Germany (2013)
40. Tomašev, N., Radovanović, M., Mladenić, D., Ivanović, M.: A probabilistic approach to nearest neighbor classification: Naive hubness bayesian k-nearest neighbor. In: Proceeding of the Conference on Information and Knowledge Management. pp. 2173–2176. ACM, New York, NY, USA (2011)
41. Tomašev, N., Mladenić, D.: Class imbalance and the curse of minority hubs. Knowledge-Based Systems 53(0), 157 – 172 (2013)
42. Tomašev, N., Radovanović, M., Mladenić, D., Ivanović, M.: Hubness-based fuzzy measures for high-dimensional k-nearest neighbor classification. In: Proceedings of the Machine Learning and Data Mining Conference. pp. 16–30. Springer-Verlag, Berlin, Germany (2011)
43. Wilson, D.R.: Asymptotic properties of nearest neighbor rules using edited data. Institute of Electrical and Electronic Engineers Transactions on Systems, Man and Cybernetics 2, 408–421 (1972)
44. Wilson, D.R., Martinez, T.R.: Instance pruning techniques. In: Proceedings of the ICML conference. pp. 404–411. Morgan Kaufmann (1997)
45. Yan, H., Mao, J., II, Y.Z., Chen, B.: Magnetic resonance image segmentation using optimized nearest neighbor classifiers. In: ICIP (3)'94. pp. 49–52 (1994)
46. Yu, D., Yu, X., Hu, Q., Liu, J., Wu, A.: Dynamic time warping constraint learning for large margin nearest neighbor classification. Inf. Sci. 181(13), 2787–2796 (Jul 2011)
47. Zhang, Z., Zhang, R.: Multimedia Data Mining: a Systematic Introduction to Concepts and Theory. Chapman and Hall (2008)
48. Zhu, X., Wu, X.: Scalable representative instance selection and ranking. In: Proceedings of the 18th International Conference on Pattern Recognition - Volume 03. pp. 352–355. ICPR '06, IEEE Computer Society, Washington, DC, USA (2006)