Noname manuscript No.
(will be inserted by the editor)

Hubness-aware kNN Classification of High-dimensional Data in Presence of Label Noise

Nenad Tomašev · Krisztian Buza

the date of receipt and acceptance should be inserted later

Abstract Learning with label noise is an important issue in classification, since it is not always possible to obtain reliable data labels. In this paper we explore and evaluate a new approach to learning with label noise in intrinsically high-dimensional data, based on using neighbor occurrence models for hubness-aware k-nearest neighbor classification. Hubness is an important aspect of the curse of dimensionality that has a negative effect on many types of similarity-based learning methods. As we will show, the emergence of hubs as centers of influence in high-dimensional data affects the learning process in presence of label noise. We evaluate the potential impact of hub-centered noise by defining a hubness-proportional random label noise model that is shown to induce a significantly higher kNN misclassification rate than the uniform random label noise. Real-world examples are discussed where hubness-correlated noise arises either naturally or as a consequence of an adversarial attack. Our experimental evaluation reveals that hubness-based fuzzy k-nearest neighbor classification and Naive Hubness-Bayesian k-nearest neighbor classification might be suitable for learning under label noise in intrinsically high-dimensional data, as they exhibit robustness to high levels of random label noise and hubness-proportional random label noise. The results demonstrate promising performance across several data domains.

Keywords: classification, label noise, k-nearest neighbor, high-dimensional data, hubness, neighbor occurrence models

Nenad Tomašev
Institute Jožef Stefan
Artificial Intelligence Laboratory
Jamova 39, 1000 Ljubljana, Slovenia
nenad.tomasev@gmail.com

Krisztian Buza
Institute of Genomic Medicine and Rare Disorders
Semmelweis University
Üllői út 26 2, 02-097 Budapest, Hungary
chrisbuza@yahoo.com

1 Introduction

Designing effective and robust supervised learning algorithms for classification in presence of label noise is an important practical issue, as obtaining reliable data labels is often expensive or simply infeasible due to data size in large-scale systems [18].

Classification noise can be random, feature-dependent or adversarial. Label flip probabilities can be either uniform and symmetric or depend on particular classes and class pairs. The simple *random classification noise* (RCN) model was first introduced in [2]. It is a model of how non-adversarial noise might affect the data. Given a training set $T = (X, Y)$ of labeled examples and a value $\eta \in (0, 1/2)$, $D_{\eta,T}$ denotes the distribution corresponding to T corrupted with random classification noise at rate η. A draw from $D_{\eta,T}$ is equivalent to a uniformly random draw from T where the label y of the selected (x, y) is randomly flipped with probability η.

The issue of unreliable and noisy labels can be approached in two ways: by trying to identify and correct/eliminate suspect data points or by incorporating noise into the learning model. Neither approach is trivial, as it is not always easy to distinguish mislabeled examples from the exceptions to general rules, atypical data points. When an instance lies far from its class interior and in proximity of instances from different classes, it can sometimes be mistaken for a mislabeled point [43]. Yet, atypical points sometimes hold valuable discriminative information, as they might help in defining proper class boundaries for classification. Additionally, many filtering approaches assume all of the data is available at the filtering stage and not prohibitively large [66].

Instead of filtering or explicit noise source modeling, it is also possible to design learning techniques that exhibit implicit robustness to high rates of label noise. In this paper, we will demonstrate that the recently proposed hubness-aware k-nearest neighbor classification methods [38][53][52][49] can be used for robust classification of intrinsically high-dimensional data under the assumption of label noise. This robustness is a consequence of the fact that, unlike in most kNN approaches, neighbor instances do not vote by their label at classification time. Instead, their vote is determined by their past occurrences on the training data.

Hubness [39] is a ubiquitous property of intrinsically high-dimensional data. With increasing dimensionality, the degree distribution of the kNN graph becomes increasingly skewed and hubs emerge as central and influential points among the data. The kNN graph itself assumes a scale-free-like topology. This has multiple consequences for similarity-based learning methods and k-nearest neighbor methods in particular. Additionally, it changes how random labeling noise affects the learning process. Errors in hub point labels can induce severe mislabeling while errors in orphan points or regular points have little influence on the classification accuracy in kNN methods.

In this paper, we introduce the concept of *hubness-proportional random label noise* as an adversarial noise model where the most influential points in the data are most likely to be corrupted. The probability of a label flip is set to be proportional to the neighbor occurrence frequency of the data point. Hubness-proportional random label noise models how a potentially successful malicious attempt can compromise the most relevant and influential neighbor points in order to disrupt kNN-based retrieval, recommendation or prediction systems.

To our knowledge, this paper is the first detailed study dedicated to examining the influence of hubness on kNN classification with uncertain data labels.

The paper is organized as follows: Section 2 summarizes the related work and the existing approaches for dealing with label noise. Consequences of hubness in intrinsically high-

dimensional data are discussed in Section 3. Neighbor occurrence models for hubness-aware kNN classification of high-dimensional data are described in detail in Section 4, followed by examples that demonstrate their potential for learning under label noise. Section 5 introduces the concept of hubness-proportional random label noise and gives practical examples of the susceptibility of kNN methods to label noise under high data hubness. The data used in the experiments is described in Section 6, followed by experimental results and summaries in Section 7. In Section 8, the main contributions of the paper are summarized and several directions for future work are proposed.

2 Related work

Label noise often occurs in large scale problems where labelling is crowdsourced to a large number of non-experts instead of having the domain experts carefully label each data instance, for instance via Amazon's Mechanical Turk. In such cases, it has been shown to be beneficial to obtain multiple labels for each data point or a carefully selected subsets of data points [24][42]. Evaluating the labeling accuracy of individual experts and non-experts can also be used in order to improve label quality by preferring certain labelers over others [15][60]. Modeling the concept evolution over time as the user's perception of the concepts that are being tagged by employing structured labeling has been shown to improve consistency and yield considerable improvements [30]. Unreliable labels can also result from automated information retrieval and tagging.

Data filtering for removing the mislabeled data points prior to model learning for classification is often used in practice. A simple approach is to rely on classification ensembles and to filter out those instances that are misclassified by the ensemble on the training data by taking a majority vote [9][58][65][47]. It is possible to detect data sub-samples that lead to high classification errors via cross-validation and to improve classification performance by relying on multiple data representations and discriminating subspaces [57]. Examples that lie in neighborhoods where a proportion of the dominating class is significantly lower than average are also suspect and their elimination can help with improving kNN classification accuracy [31]. Boolean rules inferred from the measurements can be used for detecting noisy data points [28]. Neural networks have been used for correcting the mislabeled examples in [63], by iteratively updating class affiliation probabilities based on the difference from the trained neural network output. Unlabeled examples can also be taken into account in filtering in a semi-supervised type of approach, raising the overall noise detection accuracy [22]. It is possible to formulate the noise removal task as an optimization problem, which might sometimes be preferable in comparison with the ensemble based filtering approaches [56].

Mutual information is a popular feature selection criterion and a robust estimation of mutual information via a probabilistic noise model was able to improve feature selection performance under label noise [17]. This was achieved by an adaptive hyper-sphere radius selection in nearest-neighbor entropy estimators. Certain feature extraction strategies have been successfully employed to improve classification accuracy in noisy medical data [35].

Presence of label noise in class-imbalanced learning tasks can be highly detrimental and it was shown to affect the learning process differently depending on whether mislabeling occurs in the minority or the majority class [23]. This is important as most noise removal strategies treat these two cases equally.

Non-uniform label noise sometimes arises due to systematic errors in data acquisition or the experimental design that produces the data in question [36][21]. The type of noise should be determined prior to deciding on the optimal noise handling strategy.

As individual labels are unreliable, it is possible to use multi-instance learning in order to aggregate instances and assign labels to groups of instances instead. This has been shown to be a promising approach [25].

While boosting methods may be popular in practice [11][26][65], recent research suggests that many types of boosting methods that can be interpreted as convex potential boosters are highly susceptible to random classification noise [32]. Branching program based boosters that do not fall into this framework can still achieve good learning accuracy on noisy data.

Designing classifiers that are able to implicitly handle noisy and mislabeled data points is another approach and one such classifier is the adaptive k-nearest neighbor classifier (AKNN) [59] that re-scales the distances of training points to the query, based on their proximity to the closest point of a different class. As labels in mislabeled points often do not match the labels among their neighbors, this approach disregards most mislabeled points as it adaptively increases their distance to the query. This approach will be our baseline for the experiments in Section 7. Deep learning algorithms can be extended to handle label noise by additional network layers for noise modeling [45]. Robust kernels can be learned from the data in order to improve the effectiveness of kernel-based methods under label noise [7] and robust SVM methods have also been considered [44][6].

The existing noise-handling strategies fail to take data hubness into account and do not attribute special attention to potential errors in the hub points, which might be an issue when learning from high-dimensional data. This problem was identified in [10], where it was noted that a surprising number of classification errors in time series kNN classification can be attributed to hub points.

3 Hubness in Intrinsically High-dimensional Data

Hubness is a consequence of high intrinsic data dimensionality related to the degree distribution of the kNN graph [39]. Hub points arise as centers of influence, as they occur very frequently as nearest neighbors. In fact, the entire neighbor occurrence frequency distribution becomes skewed and most points become *anti-hubs* or *orphans*, i.e. they occur rarely or never as neighbors to other points. Hubs often exhibit a detrimental influence by inducing many label mismatches in kNN sets and they can become semantic singularities in the data. These *bad hubs* can arise for many reasons and they are not necessarily erroneous data points. However, there is an increased chance for their emergence in presence of label noise.

Hubness has first been reported in music retrieval systems [4][3], where it is still an important and largely unresolved issue [16], despite some recent advances [41][19]. Hub songs were occurring exceedingly often in top-k result sets, even in cases when there was no apparent semantic connection to the queries.

Let $T = \{(x_1, y_1), (x_1, y_1) \dots (x_N, y_N)\}$ be a training 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. 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 $D_k(x_i)$ is referred to as k-occurrence. The number of k-occurrences of a point x is denoted by $N_k(x)$ and will sometimes be referred to as

the *hubness* of x[1]. A k-occurrence is considered *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 *bad occurrences* of neighbor points. Total occurrence counts consist of a sum of good and bad occurrences, as $N_k(x_i) = GN_k(x_i) + BN_k(x_i)$, where GN_k and BN_k represent good and bad hubness, respectively. It is possible to consider class-conditional occurrence sums as well and 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*[2] of the neighbor k-occurrence frequency distribution is defined as follows:

$$SN_k(x) = \frac{m_3(N_k(x))}{m_2^{3/2}(N_k(x))} = \frac{1/N \sum_{i=1}^{N}(N_k(x_i) - k)^3}{(1/N \sum_{i=1}^{N}(N_k(x_i) - k)^2)^{3/2}} \quad (1)$$

High positive skewness of the neighbor k-occurrence frequency in intrinsically high-dimensional data indicates that the distribution tail is longer on the right, as illustrated in Figure 1. In many dimensions, the k-occurrence frequency distribution approaches a power law.

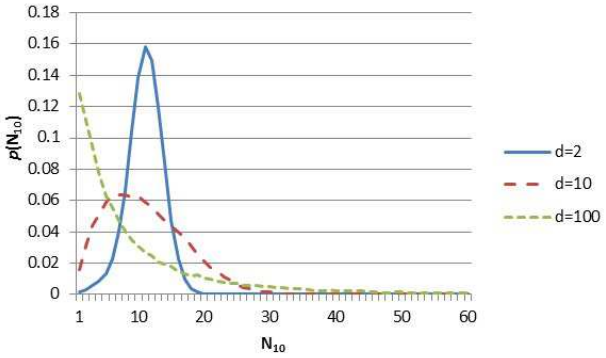


Fig. 1 The change in the distribution shape of 10-occurrences (N_{10}) in i.i.d. Gaussian data with increasing dimensionality when using the Euclidean distance. The graph was obtained by averaging over 50 randomly generated data sets. Hub-points exist also with $N_{10} > 60$, so the graph displays only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

Most data in practical applications is intrinsically high-dimensional and k-nearest neighbor hubs have been shown to arise in text, audio, images [48], collaborative filtering data [34] and time series [37].

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

[2] Skewness of a probability distribution is its 3rd standardized moment and is frequently used in statistical analysis.

4 Neighbor Occurrence Models

Section 4.1 describes the basic ideas behind the neighbor occurrence models in hubness-aware classifiers and Section 4.2 gives examples of how the use of neighbor occurrence models might improve classification performance under label noise.

4.1 Learning from Past Occurrences

In order to alleviate the negative influence of bad hubs in the data and allow for robust k-nearest neighbor classification under the assumption of hubness, several hubness-aware kNN methods have recently been proposed: hubness-weighted kNN (hw-kNN) [38], hubness-fuzzy kNN (h-FNN) [53], hubness-information kNN (HIKNN) [49], Naive Hubness-Bayesian kNN (NHBNN) [52] and Augmented Naive Hubness-Bayesian kNN (ANHBNN) [51]. All hubness-aware approaches are based on learning from past occurrences by means of building a neighbor occurrence model from the observations on the training data. These class-conditional neighbor occurrence probabilities are used to predict consequences of certain neighbor occurrences in future tests and to infer the class affiliation probabilities of future query points.

The weighting approach in hw-kNN is simple, yet quite effective in many cases. It is based on diminishing the effect of bad hubs on classification. Standardized bad hubness defined by $h_B(x_i) = \frac{BN_k(x_i) - \mu_{BN_k}}{\sigma_{BN_k}}$ is used to determine vote weights, where μ_{BN_k} and σ_{BN_k} denote mean bad hubness and its standard deviation. Each x_i is then assigned a voting weight of $w_i = e^{-h_B(x_i)}$. While this weighting reduces the contribution of bad hubs to the vote, there is still some unexploited information in the past neighbor occurrences that can be used for better class prediction.

Class-conditional occurrence profiles can be used to determine soft votes in a fuzzy k-nearest neighbor framework and this was a basis for hubness-fuzzy kNN (h-FNN).

$$u_c(x) = \frac{\sum_{i=1}^{k} u_{ci}\left(\|x - x_i\|^{-(2/(m-1))}\right)}{\sum_{i=1}^{k}\left(\|x - x_i\|^{-(2/(m-1))}\right)}, \tag{2}$$

Let x be a newly observed data instance for which we wish to perform classification. The degree of membership of x in each class c is then defined as in Equation 2.

$$u_{ci} = \begin{cases} p_k(y = c|x_i) \approx \frac{N_{k,c}(x_i)+\lambda}{N_k(x_i)+n_c\lambda}, & \text{if } N_k(x_i) > \theta, \\ \frac{\lambda+\sum_{(x,y)\in(X,Y)|y=y_i} N_{k,c}(x)}{n_c\lambda+\sum_{(x,y)\in(X,Y)|y=y_i} N_k(x)}, & \text{if } N_k(x_i) \leq \theta. \end{cases} \tag{3}$$

The θ parameter in Equation 3 represents the anti-hub cut-off point and can even be set to 0 by default. Distance weighting is optional and a default parameter value of $m = 2$ has been previously shown to perform well and is used in our experiments. The parameter value can be further optimized via cross-validation. This simple approach performs surprisingly well in many cases and will be the focus of our experimental comparisons in Section 7.

Fuzzy votes of neighbor points in h-FNN are derived from their past k-occurrence profiles. Their own labels are not directly taken into account. The fuzzy vote derived from the reverse-nearest neighbor set is in fact an estimate of the true class density distribution in the neighbor point. Since most neighbors are hubs and hubs occur on average in many k-nearest neighbor sets, these estimates can be quite robust to random label flips and noise. h-FNN has

been shown to perform substantially better than basic fuzzy k-nearest neighbor method [27] on high-dimensional data.

An extension of h-FNN in the form of HIKNN was later proposed, by including the original label information and giving preference to rarely occurring neighbor points. Neighbor occurrence self-information is defined as $I_{x_{it}} = \log \frac{1}{p(x_{it} \in D_k(x))}$, where $p(x_{it} \in D_k(x)) \approx \frac{N_k(x_{it})}{N}$ is the probability of the point occurring as a neighbor. Relative and absolute surprise factors $\alpha(x_{it}) = \frac{I_{x_{it}} - \min_{x_j \in D} I_{x_j}}{\log n - \min_{x_j \in D} I_{x_j}}$ and $\beta(x_{it}) = \frac{I_{x_{it}}}{\log N}$ can be derived from the neighbor occurrence self-information and are used for weighting neighbor votes and weighting the contributions of the label information and occurrence profile information in the final fuzzy votes. The distance weighting factor $d_w(x_{it})$ is optional and we have used the same weighting scheme as in h-FNN in our experiments.

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

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

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

Equations 4 and 5 represent the HIKNN voting framework, based on previously defined quantities. This form of assigning voting weights incorporates a bias towards more 'local' neighbor points, since hubs tend to be located closer to cluster centers in high-dimensional data. However, the increased specificity bias of the HIKNN learning approach makes it somewhat more prone to noise and mislabeling.

Naive Hubness-Bayesian k-nearest neighbor (NHBNN) [52] represents another approach to learning from past occurrences, which is based on a Naive Bayesian estimate of the class affiliation probabilities. Denote the data size by $N = |T|$ and the size of class c by $n_c = |\{x_i : y_i = c\}|$. The NHBNN rule is then given as follows, with a λ smoothing parameter.

$$p(y_i = c | D_k(x_i)) \propto p(y_i = c) \prod_{t=1}^{k} p(x_{it} \in D_k(x) | y = c) = \frac{n_c}{N} \prod_{t=1}^{k} \frac{N_{k,c}(x_{it}) + 1 + \lambda}{n_c \cdot (k+1) + \lambda N} \tag{6}$$

Naive Bayes rule is based on an independence assumption between the attributes and this assumption is severely compromised in NHBNN. Nevertheless, Naive Bayes is known to often be able to deliver good results in presence of strong functional correlations between the attributes [40] and NHBNN has been shown to perform well in class imbalanced classification tasks on high-dimensional data [50].

An important property of all hubness-aware kNN classification methods is that it is possible to use them in boosting. Neighbor occurrence models can be trained with instance weights. This is why in Section 7 we have avoided considering boosting approaches as competitive baselines, as hubness-aware classification methods can actually be used in conjunction with those noise-tolerant boosting strategies. A detailed examination of this idea is beyond the scope of this paper and will be addressed in future work.

4.2 Handling of Mislabeled Points in Hubness-aware Classification

Most k-nearest neighbor methods have a high sensitivity bias as they retain all the examples and do not generalize by building explicit models. Learning from past occurrences in form of building neighbor occurrence models increases the generalization capabilities of kNN classification, especially in case of h-FNN and NHBNN.

Consider a simple 2-dimensional example shown in Figure 2. If a point is mislabeled, h-FNN can learn that its past occurrences are inconsistent with its label and would prefer the occurrence information to the label information when making a classification decision. While this approach seems conceptually well suited for handling mislabeled data points, it is not only the mislabeled points that become bad hubs and exhibit a detrimental influence on kNN classification. Bad hubs are not uncommon in intrinsically high-dimensional data and this is why it might be a good idea to take the past occurrence information into account. Past occurrence evidence is derived from the label distribution of a potentially large number of reverse nearest neighbors in case of strong hub points, so it is also more robust to label noise.

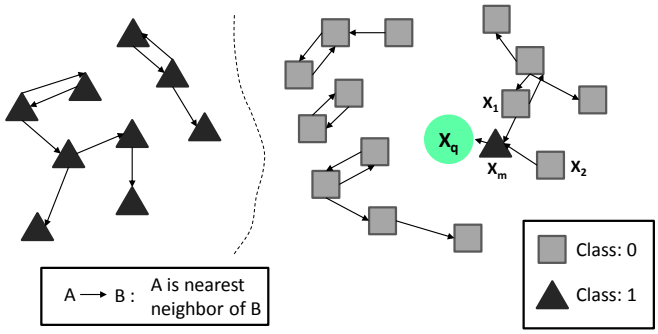


Fig. 2 An illustrative example of 1-NN classification in presence of incorrect data labels. Consider X_q as the query point. Its nearest neighbor $X_m = \text{NN}(X_q)$ seems to be mislabeled. The 1-NN rule would assign X_q to class 1 instead of class 0, due to the label of its nearest neighbor. This error might further propagate if X_q is retrieved in future classification queries. In this particular case, it is possible to sidestep the issue by using a larger neighborhood, though this is not always feasible in more complex data. However, we will demonstrate that it is possible to reach the correct classification decision even for $k = 1$, by applying hubness-aware classification. Namely, $X_m = \text{NN}(X_1)$ and $X_m = \text{NN}(X_2)$ and $Y_1 = Y_2 = 0$. Counting the occurrences gives $N_1(X_m) = 2$, $N_{1,0}(X_m) = 2$ and $N_{1,1}(X_m) = 0$. Consider the hubness-fuzzy k-nearest neighbor method, without the optional distance weighting. This gives $u_0(X_m) = \frac{2+\lambda}{2+2\cdot\lambda}$ and $u_1(X_m) = \frac{\lambda}{2+2\cdot\lambda}$. Assume $\lambda = 1$ here, for simplicity. This yields $u_0(X_m) = 0.75$ and $u_1(X_m) = 0.25$. As X_m is the only point to vote since $k = 1$, these are also the final h-FNN class affiliation probability estimates for X_q. Therefore, $p(Y_q = 0) = 0.75$ and $p(Y_q = 1) = 0.25$. These estimates lead to a correct X_q classification, despite its mislabeled nearest neighbor. Similar derivations hold for NHBNN or HIKNN. This example illustrates the motivation behind learning from past occurrences and hubness-aware classification. Due to the fact that hubs often turn out to be detrimental to classification [37] and that bad hubness is not uncommon in intrinsically high-dimensional data, it is not uncommon for a point to have bad hubs among its k-nearest neighbors. However, it is not always that easy to extract useful information from past hub occurrences, especially in highly non-homogenous high-entropy occurrence profiles, where the occurrence profile itself cannot clearly indicate how the vote should be placed. Different strategies for combining all this information for inferring the class affiliation in the point of interest yield different hubness-aware approaches.

As most neighbor k-occurrences in high-dimensional data are in fact hub occurrences, flipping a single label is much less likely to cause misclassification in hubness-aware classifiers. In fact, randomly flipping a label of a hub point might induce severe misclassification in the basic kNN method and many standard kNN methods. However, a noisy hub label does not exhibit a greater influence on hubness-aware classification with h-FNN or NHBNN than any other noisy label. This is due to the fact that the labels are not used directly in voting for classification.

Formally, assume we are observing a point (x_m, y_m) whose label has been randomly flipped. In kNN, this noisy information propagates to $N_k(x_m)$ k-nearest neighbor sets, all kNN sets where x_m occurs as a neighbor. In h-FNN, the only influence that y_m has on future classification is via the occurrence models of the neighbors of x_m, points $x_i \in D_k(x_m)$. In many intrinsically high-dimensional data sets, a large proportion of points are orphans that never occur as neighbors. Due to that fact, the expectation of $N_k(x)$ for those points that do occur in neighbor sets increases and $E(N_k(x_m)|\exists x : x_m \in D_k(x)) > k = |D_k(x_m)|$ if orphans are present in the data. It follows that non-orphan mislabeled points are expected to propagate the noisy information in more cases in kNN than h-FNN, assuming high data dimensionality. However, orphans play no role in kNN, but they are being taken into account when building the neighbor occurrence models in h-FNN and orphan points can also have noisy labels. The expected number of cases where a randomly mislabeled point would exhibit its influence is therefore the same, if we take both orphans and the occurring neighbor points into account.

However, while the expected number of error propagation cases might be the same, the expected effect is not. Class probability density estimation quality depends on the number of sample neighbor or reverse neighbor points. In general, 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. A single noisy labeled data point or a constant number of noisy labeled data points would have a more pronounced effect on those estimates that are derived from a smaller number of sample points, trivially. In kNN, the class probability estimates are derived from exactly k neighbors. In neighbor occurrence models, it was already mentioned that $E(N_k(x)|\exists x_i : x \in D_k(x_i)) > k$ in high-dimensional data in presence of orphan points. Therefore, a single mislabeling has a higher expected effect on kNN voting than on one fuzzy h-FNN vote. As h-FNN estimates the class affiliation probabilities from k such fuzzy votes, a much larger number of points is used in deriving the final estimate.

5 Hubness-proportional random label noise

In adversarial classification tasks like intrusion detection or spam filtering, malicious adversaries may manipulate data labels in order to affect the classification outcome. As hubs are the centers of influence in kNN classification, we postulate that most damage could potentially be done to kNN-based learning systems by targeting hub points specifically with label noise.

Uniform random label noise is, therefore, insufficient to properly estimate the pessimistic scenarios of potentially successful malicious hub label flips that might be targeted in kNN-based systems. It is also insufficient for estimating the worst-case robustness of such systems and can be used primarily for evaluating the average system behaviour in presence of label noise. In order to be able to better test the sensitivity of kNN-based systems to adversarial hub-targeted attacks or non-adversarial systematic hub-centered errors, it is preferable to use a non-uniform hub-preferential noise model.

Def. Let **hubness-proportional random label noise**[3] with the noise rate η be the random label noise that results from the following stochastic process:

Step 1: Randomly select a set S of distinct data points from T of size $|S| = \eta \cdot N$, according to the discrete probability distribution that defines the inclusion probability of $x_i \in T$ to S in a single draw as $p_{\text{select}}(x_i) = \frac{N_k(x_i)}{k \cdot N}$.
Step 2: Randomly flip the label of each $x_i \in S$, so that $p_{\text{flip}}(y_i \rightarrow c) = \frac{1}{C-1}$ for $c \neq y_i$.

Since orphan points have $N_k(x_i) = 0$, some smoothing might be necessary in practice in order to ensure positive selection/flip probabilities for all data points. If the k-neighbor set of each point is extended to include the point itself, then the issue is avoided and $p_{\text{select}}(x_i) = \frac{N_k(x_i)+1}{(k+1) \cdot N}$.

Enforcing the mislabeling rate by fixing the number of induced mislabeled points is required since it is impossible to simply define $p_{\text{flip}}(y_i) = \eta \cdot \frac{N_k(x_i)}{k}$ for independent label flips, due to the fact that it is possible to have $\frac{N_k(x_i)}{k} > \frac{1}{\eta}$, which would then result in $p_{\text{flip}}(y_i) > 1$.

Hypothesis: In intrinsically high-dimensional data, hubness-proportional random label noise affects kNN classification more severely than uniform random label noise and represents a challenging noise model that can be used to evaluate the limit-case robustness of kNN methods.

Nevertheless, it should be noted that the hubness-proportional random label noise is not the worst case scenario in itself. The worst case would be to consider having all of the top $\eta \cdot N$ most frequent neighbor points mislabeled, since this would maximize the number of compromised k-neighbor occurrences. Since such a worst case scenario is unlikely in the non-adversarial case and also difficult to achieve in the adversarial case unless complete information about all the data and all the system components is available, we prefer to rely on the proposed stochastic hubness-proportional random label noise model for evaluating the kNN-based system robustness instead.

It is possible to extend the proposed stochastic model by conditioning the label flips on the values of the original labels and considering the principal class-conditional gradients of misclassification in the data. The misclassification gradients can be deduced from the classification confusion matrices. The adversary could then flip the label of the target instance to the value that is most likely to cause misclassification in the class that is estimated as most common in the target instance's occurrence profile. While such advanced strategies are conceivable, they would require the class-conditional occurrence estimates that are non-trivial to obtain, so the following evaluation focuses on the simpler yet challenging hubness-proportional random label noise model instead.

In Section 5.2 we will discuss possible adversarial approaches for inducing hubness-correlated label noise in real-world data and show that the hubness-proportional random label noise model can be helpful in evaluating the robustness of the systems to such attacks.

The impact of hub-centered label flips on machine learning performance can be quite substantial, depending on the underlying data dimensionality and hubness. Section 5.1 gives a practical example of how things can go wrong even if no more than a few labels of the highly influential examples get compromised.

[3] Hubness-proportional label noise will also be referred to as $N_k(x)$-proportional label noise interchangeably throughout the paper.

5.1 Susceptibility of kNN to Hub-centered Noise: An Illustrative Example

In many types of networks, the presence of hubs can increase robustness to random noise [64]. However, this comes at a price. At the same time, the presence of hubs makes scale-free networks significantly more vulnerable to hub-centered inaccuracies. Small changes and low noise levels can sometimes substantially harm system performance. The example outlined in Figure 3 illustrates this problem [48].

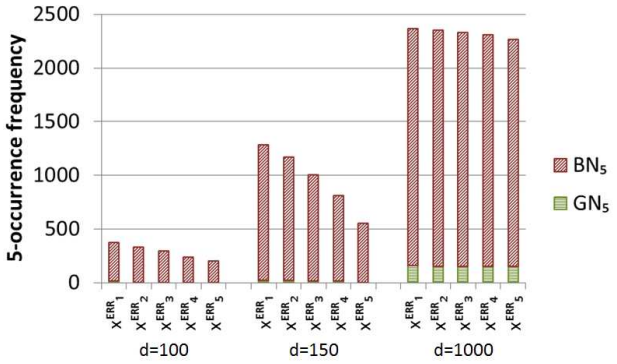


Fig. 3 The emergence of top 5 major hubs on the iNet3Err dataset [48]. Good and bad 5-occurrence frequencies are denoted by GN_5 and BN_5, respectively. Under the particular choice of feature representation (SIFT [33] bag of visual words) and metric (Manhattan), noisy feature vectors that resulted as errors in the feature extraction pipeline ended up becoming the major hubs in the data, as the size of the visual word vocabulary was increased. Their influence was highly detrimental, as most of their occurrences induced label mismatches.

Figure 3 shows the emergence of 5 major hubs on the iNet3Err quantized SIFT [33] representation and their detrimental influence. The data corresponds to a 3-class subset from the public ImageNet repository [14] (`http://www.image-net.org/`). The experiments were run to determine the optimal bag of visual words vocabulary size [48] and an increase in dimensionality resulted in a sudden and severe drop in object recognition performance. As a result, the basic 5-NN classifier performed worse than zero-rule for the 1000-dimensional case, as shown in Table 1. Subsequent analysis has determined the cause of this pathological behavior to lie in the emergence of several extremely bad pervasive hub images.

Table 1 Classification accuracy of kNN and four hubness-aware kNN algorithms (hw-kNN, NHBNN, h-FNN, HIKNN) on iNer3Err image data. Statistically significant improvements are denoted by $\circ$.

Data set	5-NN	hw-kNN	NHBNN	h-FNN	HIKNN
ImNet3Err	21.2 $\pm$ 2.1	27.1 $\pm$ 11.3	59.5 $\pm$ 3.2 $\circ$	59.5 $\pm$ 3.2 $\circ$	59.6 $\pm$ 3.2 $\circ$

In this particular case, the image hubs were erroneously represented by zero-vectors as a result of an I/O error in the feature extraction pipeline. The Manhattan distance from a zero vector to any given quantized image representation remains constant, regardless of the codebook size. The number of local image features that are being quantized is constant, so the L_1 norm of the quantized vector does not change. At the same time, the distances

between pairs of images increase on average with increasing dimensionality, as the weights are spread over a larger number of buckets. This causes the zero vectors to become major hubs in the data.

The particular error was easily corrected by re-running the feature extraction component for the images in question and updating the extraction code. It can also be argued that carefully designed data processing systems ought to perform run-time consistency checks to ensure valid data representation. However, this example clearly shows the potential danger that lies hidden in the hubness of the data. Only 5 detrimental hub images had rendered the kNN-based object recognition component effectively useless on a 2731 image dataset.

Even properly processed data sources might contain non-negligible amounts of external systematic and non-uniform label noise [36][21]. If such noise were to align itself in the data space with the central cluster regions that exhibit the highest overall hubness in the data [54], hub-centered noise could still arise.

Highly detrimental hub points can also naturally arise in the centers of borderline regions between different classes and need not be noisy instances. Ensuring a valid and correct data representation is not enough to prevent such pathological cases from ever occurring in practice. Additional data filters and instance selection components might be necessary for robust data pre-processing. Robust systems need to be able to handle not only the average levels of noise and data corruption but also the more extreme cases of hub-centered noise.

5.2 Simulating the Adversarial Hub-targeted Label Flips

Hub-centered label noise has a high pay-off in adversarial scenarios, where malicious intrusions might be difficult or costly and the adversaries might look for ways of maximizing the disruption to the target systems with minimal intervention. If the adversaries were to be able to predict the relevance of the known examples on unseen data, they would be able to target hub points specifically.

If given access to a sample of the data drawn from the same or similar distribution as the unseen data, it would be possible to evaluate the future occurrence frequencies of potential target examples, especially as estimating the exact frequencies is less important than estimating whether a given point is a potential hub target or not. As hubs occur very frequently in k-nearest neighbor sets, even a small sample would suffice for detecting most hub targets in a very straightforward way, assuming some knowledge of the underlying metrics and data representations.

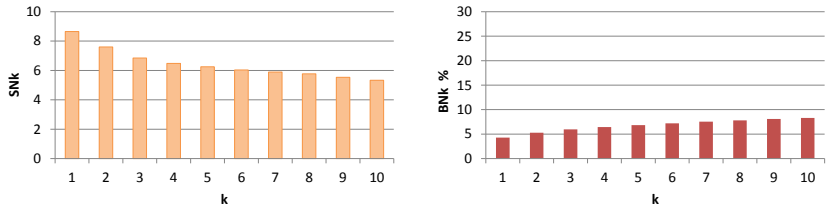
If the exact data representation or the distance measure are unknown, it is still possible to trivially determine target hubs if given access to the system itself, by running a series of queries on the system and keeping track of the returned items in the top-k result sets and their properties.

Even if the exact similarity measure is unknown and constantly querying the system is not feasible, there are some known properties of hubness that can be exploited for approximately detecting high-hubness targets. For instance, it is a known property of hubs that they tend to lie close to local cluster centers [54]. Clustering can, therefore, also be used for estimating point centrality and the centrality can be used for estimating point hubness, the same way that point hubness can be used for effective clustering of high-dimensional data.

In certain cases, indirect hubness-correlated attacks are possible based on some known properties of certain data types and metrics. Document relevance is preserved across languages, so it possible to approximate document hubness if given access to data translations [55]. In textual data in particular, document length can sometimes be correlated with

hubness and shorter (or longer) documents might have a tendency for becoming hubs, under different circumstances [37].

We will illustrate this point by simulating an adversarial attack on the labels of the SMS message spam data [12, 13, 1] from the UCI repository (`https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection`). A 4-gram representation was extracted and the cosine similarity was used after applying TF-IDF, which is standard in many text processing applications. N-grams were preferable to simple bag of words here, due to a high frequency of misspelled words and alternate spellings. The data contains 5574 messages including 4827 regular and 747 spam messages. This collection of short messages exhibits substantial hubness, which can be seen in Figure 4.



(a) Skewness (b) Bad hubness percentage

Fig. 4 Skewness and bad hubness in the SMS spam dataset, over a range of neighborhood sizes. The data was represented via 4-grams and the similarity was calculated as cosine similarity after applying TF-IDF. In all cases, the data exhibits substantial hubness. The classification task is not too difficult in absence of noise, given the low rate of label mismatches in kNN sets.

Let $x_i = \{f_i^j, j \in \{1 \dots v_{\text{size}}\}\}$ be the feature representation of message x_i. The f_i^j corresponds to the occurrence frequency of the j-th n-gram in the message. As the SMS messages are short, most f_i^j equal zero and the representation is sparse. The TF-IDF weight for the j-th n-gram in message x_i is defined as $w^j_{\text{tfidf}}(x_i) = f_i^j \cdot \log \frac{N}{|x_p : f_p^j > 0|}$. We will also assign a weight to the entire message by summing all the individual TF-IDF weights for the terms that occur in that message, as follows: $W_{\text{tfidf}}(x_i) = \sum_{j:f_i^j>0} w^j_{\text{tfidf}}(x_i)$.

In the SMS spam message dataset, hub messages seem to have the lowest average total weight, as shown in Figure 5.

We have simulated an adversarial label noise attack that exploits this fact. Instead of calculating or estimating target message hubness, the attack is based on targeting the messages of lowest weight. We have examined both the stochastic and the deterministic scenario. In the deterministic scenario, the adversary is able to flip the labels of $\eta \cdot N$ messages of lowest weight in the collection. In the more realistic, probabilistic case, the adversary is able to compromise the label of x_i with a probability $p_{\text{flip}}(y_i)$ that is proportional to the inverse of the total message TF-IDF weight: $p_{\text{flip}}(y_i) \propto \frac{1}{W_{\text{tfidf}}(x_i)}$. In the absence of the exact TF-IDF weights, the weights can be approximated from a sample or from other textual sources.

The experiments were run as repeated random subsampling. In each iteration, the data was randomly split so that 70% was taken as training and 30% as test data. Different noise models were applied to the training data in separate experiments. Test data was used to query the training data and the k most similar messages were selected for each test data point. The average bad hubness rate was calculate in each case. The summary of the experiments is given in Figure 6, for different noise rates.

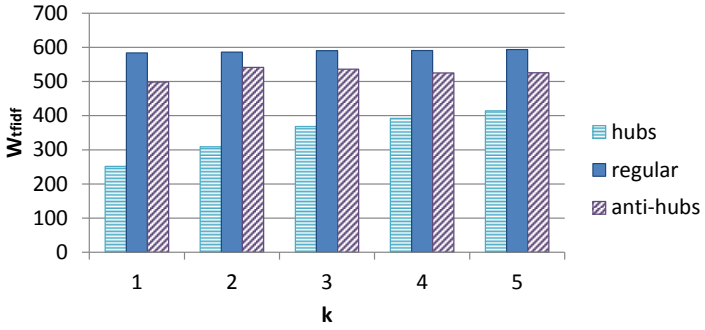


Fig. 5 The average message weight as a sum of the TFIDF weights of all its n-grams for hubs, regular messages and anti-hubs, over a range of neighborhood sizes. Hub messages have the lowest average length for small neighborhood sizes.

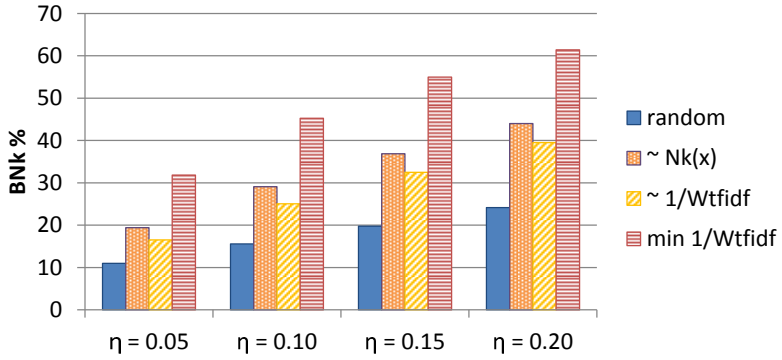


Fig. 6 The induced bad hubness percentages in the SMS spam dataset under different noise models and different noise rates. The random inverse TF-IDF weight noise model induces almost as much bad hubness in the data as the hubness-proportional random label noise. The deterministic case where the $\eta \cdot N$ shortest messages are selected and mislabeled induces a much higher bad hubness rate then any other examined noise model, as many hub messages get mislabeled.

The experiments have shown that the inverse message weight noise model produces a very similar bad hubness rate to the hubness-proportional random label noise. The deterministic alternative, where a number of messages of lowest weight is selected, produces an even more severe bad hubness in the data and causes significant misclassification.

These experiments clearly demonstrate that it is possible to exploit some known properties of hubs in standard feature representations to approximate random hubness-proportional label noise. Strictly speaking, the outcome is not an indirect hubness-proportional noise

model, as much as hub-targeted and hubness-correlated noise model. The distinction, though, is not that important from the practical viewpoint. Since it might be possible to carry out malicious attacks targeting hub examples in particular, hubness-proportional random label noise can be used to model the consequences of a successful attack and to evaluate the robustness of the system.

6 Data

The experiments were performed on several data domains. The benchmark consists of quantized image representations, high-dimensional Gaussian mixtures, UCI datasets[4], as well as UCR time series datasets[5]. Images and Gaussian data exhibited substantial hubness, while the selected UCI and UCR datasets exhibited low-to-medium hubness on average.

Image data used in the experiments consists of several high-hubness subsets of the ImageNet repository[6] that were previously used in hubness studies [53][48]. We have examined quantized 400-dimensional SIFT feature and 100-dimensional Haar wavelet representations.

The Gaussian mixture data was also used in previous experiments [50]. It was generated with a specific intent to pose difficulties for k-nearest neighbor classification. 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 set to be 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 initial class, the mean vector was set to zeroes and the standard deviation vector was generated randomly. Each subsequent class c was then randomly 'paired' with one prior Gaussian class, denoted by $\bar{c}$, so that some overlap between the two was assured. For each dimension $i \in 1..d$ independently, μ_c was then set to $\mu_c \approx \mu_{\bar{c}} \pm \beta \cdot \sigma_{\bar{c}}$ with equal probability, where $\beta = 0.75$. 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 $[0, 1]$. The assigned class sizes were randomly taken from a range between 20 and 1000. The 10 generated synthetic datasets were set to be 100-dimensional and to contain 20 different classes.

A set of 15 representative UCI datasets was selected for experiments, as follows: Arrhythmia, Ozone, Ecoli, Gisette, Glass, Haberman, Ionosphere, Mfeat-factors, Mfeat-fourier, Mfeat-karhunen, Iris, Segment, Sonar, Vehicle and Ovarian. While some of this data is high-dimensional, the selected datasets do not exhibit severe hubness, unlike the selected image data.

Similarly, we have selected a set of 15 representative UCR time series datasets, as follows: CricketX [29], CricketY, CricketZ, FacesUCR, MedicalImages, MALLAT, Motes, OliveOil, SonyAIBORobotSurface, SonyAIBORobotSurfaceII, SwedishLeaf, Symbols, SyntheticControl, Trace, TwoPatterns. kNN classification is often used in the time series domain, especially when used in conjunction with the dynamic time warping distance [62][61]. This approach is among the most popular ones and competitive with the state-of-the-art. Therefore, it is important to evaluate the robustness of kNN methods to label noise on time series data.

[4] https://archive.ics.uci.edu/ml/datasets.html

[5] www.cs.ucr.edu/ eamonn/time_series_data/

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

7 Experiments

All experiments were run as 10-times 10-fold cross-validation. Corrected re-sampled t-test was used for statistical significance comparisons [8].

Most experiments were performed in 3 main experimental setups: learning and classification with correct data labels; learning and classification under uniform random label noise rate $\eta = 0.3$ and learning and classification under hubness-proportional random label noise rate $\eta = 0.3$. The influence of varying the noise levels is discussed in the experiments in Section 7.2.

We have compared the performance of kNN, neighbor-weighted kNN (NWKNN) [46] and adaptive kNN (AKNN) [59] with the performance of hubness-aware classification methods, in particular hw-kNN, HIKNN, NHBNN and h-FNN. The neighbor-weighted kNN is an extension of the basic kNN method that incorporates class-conditional vote weighting for class-imbalanced data. The third baseline, AKNN, is the most competitive baseline approach, due to its noise-robust distance re-scaling strategy, as explained in Section 2.

Manhattan distance was used for comparing image representations, Euclidean on UCI data and Gaussian mixtures and dynamic time warping (DTW) on time series. Neighborhood size of $k = 5$ was used in most experiments, while Section 7.3 discusses the influence of varying neighborhood size.

The summary of the experiments on ImageNet datasets is given in Table 2, the summary of experiments on high-dimensional Gaussian mixtures in Table 3, the summary of experiments on UCI datasets in Table 4 and the summary of experiments on time series UCR data in Table 5. The experiments demonstrate a substantial difference in robustness between kNN and the tested hubness-aware approaches. For example, the average accuracy of kNN on ImageNet data drops from 79.8% to 70.5% and 43.8% when the correct labels are influenced with $\eta = 0.3$ random label noise and $\eta = 0.3$ hubness proportional random label noise, respectively. In the same circumstances, the average classification accuracy of h-FNN changes from 81.4% to 79.6% and then to 79.2%. The absolute accuracy drop of 36% in kNN corresponds to a drop of mere 2.2% in h-FNN.

Among the hubness-aware classification methods, h-FNN achieves the best results overall for the $\eta = 0.3$ random label noise level and $k = 5$, uniform or hubness-proportional. It is followed by NHBNN, AKNN and HIKNN, depending on the data domain and the setup. The worst among the tested hubness-aware approaches was the hubness-weighted kNN, which is not surprising, as it performs voting by label, unlike the other compared hubness-aware methods that base their votes on the neighbor occurrence models.

Uniform label noise affects the perceived class distribution in the data and it shifts it towards the uniform class distribution. This difference between the perceived class distribution on the training data and the actual class distribution on the test data implies that the class-conditional vote weighting in NWKNN will be negatively affected by the label noise. This is why the experiments indicate that NWKNN actually performs worse than the basic kNN in presence of label noise. AKNN performs best among the baselines.

The improvements offered by h-FNN over kNN are more pronounced in high-hubness data than in the examined low-to-medium hubness data. Nevertheless, h-FNN seems to achieve promising results in those cases as well.

The improvement rate may vary when using different neighborhood sizes, as discussed in Section 7.3.

7.1 Robustness to Hubness-proportional Label Noise

Hubness-proportional random label noise increases label mismatch percentages in k-nearest neighbor sets more than the uniform random label noise, as shown in Figure 7. The difference is more pronounced in high-hubness data, which also explains why there is a bigger difference between the performance of kNN and the performance of hubness-aware methods in those cases.

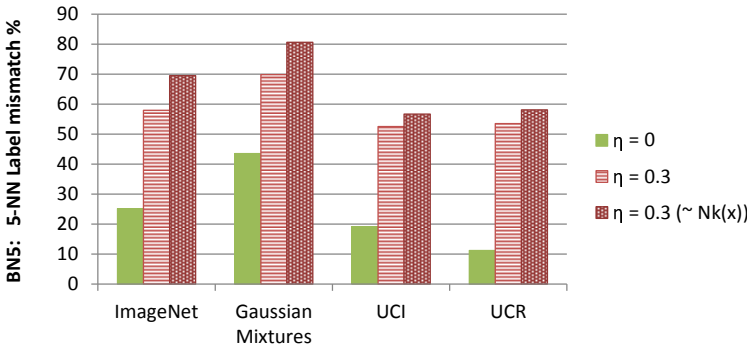


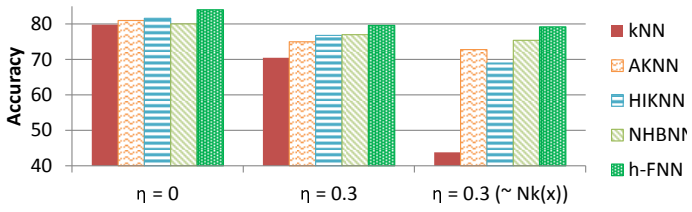
Fig. 7 The percentage of label mismatches in 5-NN sets as label noise is introduced in the data. The non-noisy case is denoted by $\eta = 0$, the uniform random label noise case with the 30% noise level by $\eta = 0.3$ and the $N_k(x)$-proportional random label noise case with the same noise level as $\eta = 0.3(\tilde{}N_k(x))$. The $N_k(x)$-proportional random label noise increases bad neighbor occurrence percentages more than the uniform label noise. The change is more pronounced in high-hubness data than in low-to-medium hubness data.

The average accuracy of kNN, AKNN, HIKNN, NHBNN and h-FNN for each data domain and each experimental setup separately is shown in Figure 8. AKNN exhibits a somewhat lower robustness than h-FNN and NHBNN to uniform random noise, though it is still comparable. However, under hubness-proportional random label noise the difference becomes apparent.

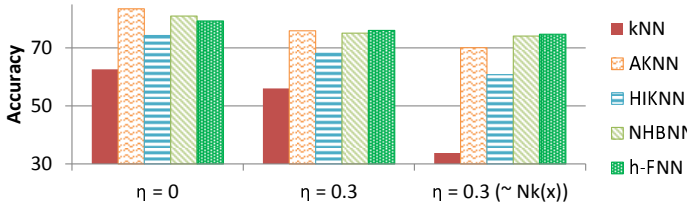
These results indicate that our initial hypothesis was correct and that hubness-proportional random label noise poses significant challenges for kNN classification. Furthermore, by using the neighbor occurrence models for hubness-aware kNN classification and the hubness-based fuzzy k-nearest neighbor approach in particular, it is possible to perform well even under high noise rates. The overall classification performance could be further improved by performing data filtering prior to classification, though this is a separate topic and beyond the scope of this study.

7.2 Influence of Varying Noise Levels

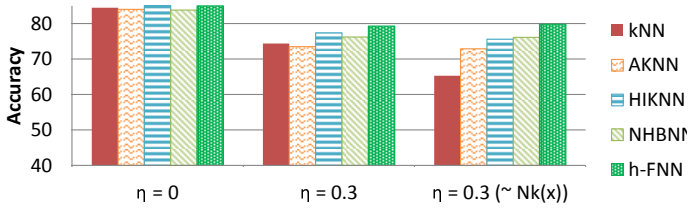
As most experiments were performed for the noise rate $\eta = 0.3$, a series of comparisons was run on multiple datasets for multiple increasing noise levels. A comparison of kNN, AKNN, HIKNN and h-FNN in terms of classification accuracy under uniform random label



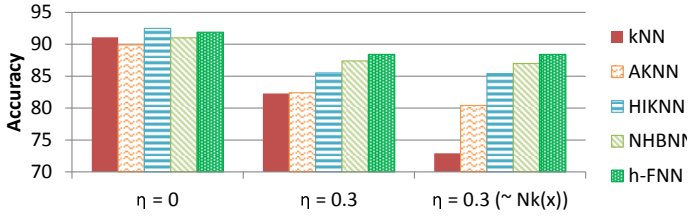
(a) ImageNet data



(b) Gaussian mixtures



(c) UCI data

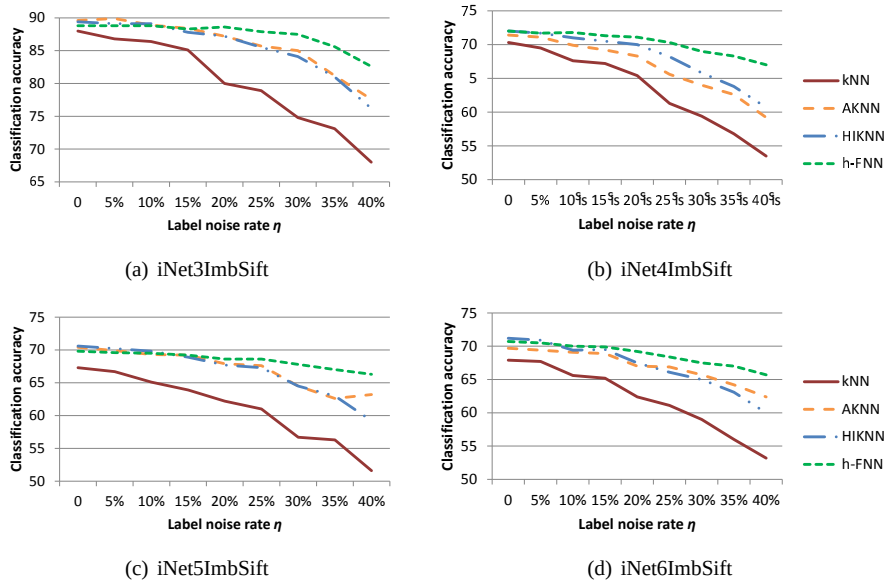


(d) UCR data

Fig. 8 Average classification accuracy of kNN, AKNN, HIKNN, NHBNN and h-FNN on different data domains. Comparisons are given for correct labels, $\eta = 0.3$ random label noise level, as well as $N_k(x)$-proportional $\eta = 0.3$ label noise. h-FNN exhibits the overall best robustness to label noise among the compared approaches. As for the baseline kNN, a steep decline in accuracy can be observed, especially in case of adversarial $N_k(x)$-proportional label noise, where it performs much worse than when noise is randomly distributed throughout the label space. No significant difference in performance of h-FNN can be seen between the two compared types of noise.

noise is shown in Figure 9, for iNet3ImbSift-iNet6ImbSift image datasets. The accuracy of

h-FNN appears to have the slowest deterioration rate with respect to label noise and the biggest improvements can be seen for the highest noise rates.

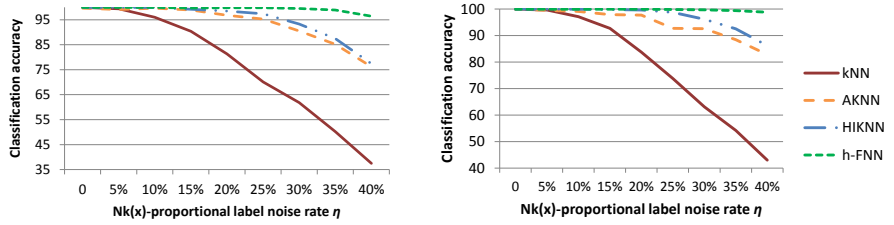


(a) iNet3ImbSift (b) iNet4ImbSift

(c) iNet5ImbSift (d) iNet6ImbSift

Fig. 9 Classification accuracy of kNN, AKNN, HIKNN and h-FNN over a range of increasing noise rates of uniform random label noise. In each case, the accuracy of h-FNN exhibits the slowest decline and it proves to be robust to high noise levels.

Similarly, a comparison between the accuracy of kNN, AKNN, HIKNN and h-FNN under increasing hubness-proportional random label noise is shown in Figure 10. h-FNN achieves a high robustness in this case and is apparently not noticeably affected by the change in the label noise distribution, unlike the other methods that show a steeper performance decline.



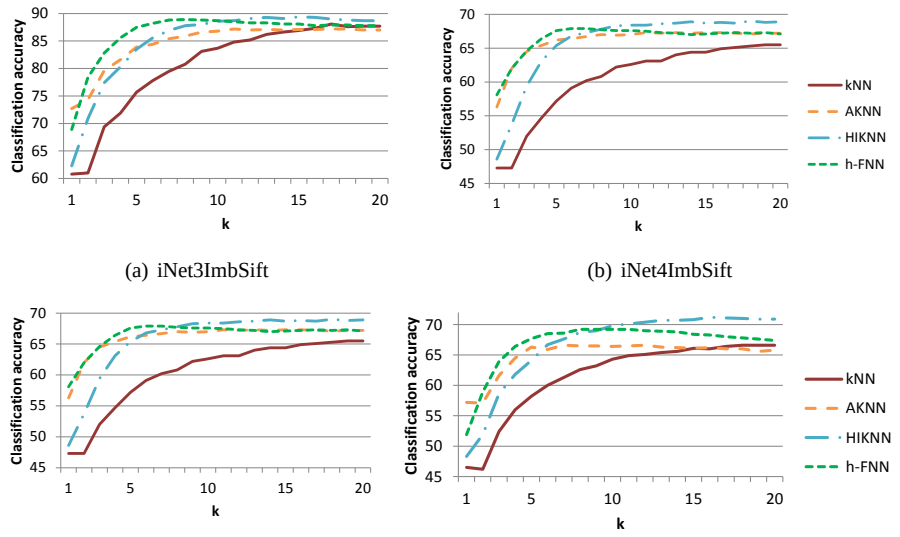
(a) iNet3Haar (b) iNet4Haar

Fig. 10 Classification accuracy of kNN, AKNN, HIKNN and h-FNN over a range of increasing $N_k(x)$-proportional noise rates. In each case, the accuracy of h-FNN seems to be least affected by the introduction of label noise.

These experiments show that the improvements that have been observed are consistent over various noise rates and noise models. Hubness-based fuzzy k-nearest neighbor classification achieves the best results among the compared approaches regardless of the noise rate or the noise distribution, though the improvements are most significant for higher noise levels, which is a beneficial property.

7.3 Influence of Varying Neighborhood Size

Choosing an optimal neighborhood size is a non-trivial problem in most kNN methods. A value of $k = 5$ was used in most experiments presented here, which is a common default choice. Larger values of k might sometimes be preferable in presence of noise. In order to evaluate the influence of neighborhood size on classification results, we have compared the classification accuracy of kNN, AKNN, HIKNN and h-FNN for a fixed random label noise rate of $\eta = 0.3$ over a range of increasing neighborhood sizes, as shown in Figure 11.



(a) iNet3ImbSift (b) iNet4ImbSift

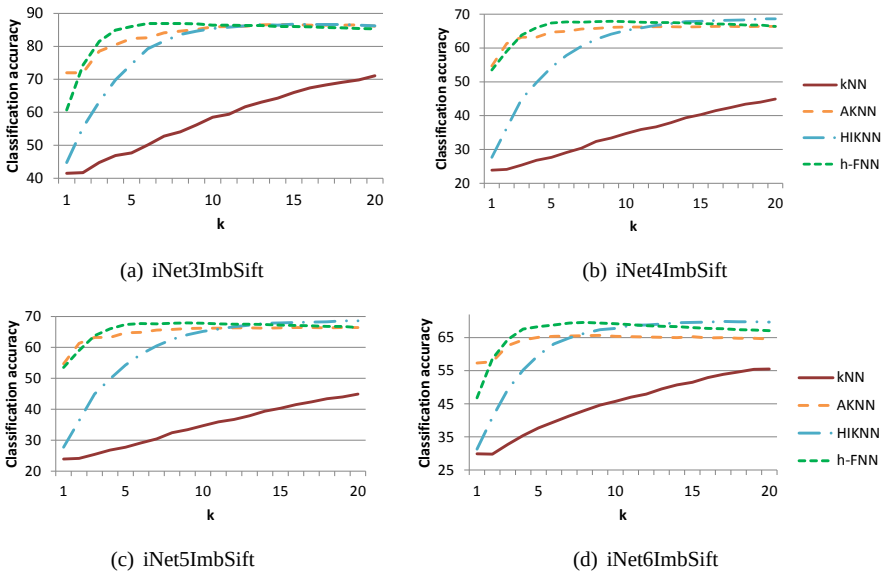
(c) iNet5ImbSift (d) iNet6ImbSift

Fig. 11 Classification accuracy of kNN, AKNN, HIKNN and h-FNN for a uniform random label noise rate of $\eta = 0.3$ over a range of increasing neighborhood sizes. Using larger neighborhoods increases the robustness of the compared classification methods to mislabeling. The best overall results are achieved by h-FNN and HIKNN.

According to Figure 11, an increase in neighborhood size improves kNN classification performance on these particular datasets, as it reduces the influence of noise. Not all remaining algorithms improve with increasing neighborhood size, as h-FNN and AKNN reach a plateau somewhere between $k = 5$ and $k = 10$. HIKNN continues to improve and outperforms h-FNN for large neighborhood sizes, while h-FNN remains dominant when smaller neighborhoods are used.

While the performance improvements over kNN vary, the highest accuracy achieved by the hubness-aware classifiers remains higher than the highest accuracy achieved by the kNN baseline, over the examined range of neighborhood sizes.

Figure 12 shows the same comparisons for hubness-proportional random label noise. Unlike in the uniform case, kNN performs much worse throughout the tested range and even for $k = 20$ and requires even larger neighborhoods to compensate for the same noise rate.



(a) iNet3ImbSift (b) iNet4ImbSift

(c) iNet5ImbSift (d) iNet6ImbSift

Fig. 12 Classification accuracy of kNN, AKNN, HIKNN and h-FNN for a hubness-proportional random label noise rate of $\eta = 0.3$ over a range of increasing neighborhood sizes. Using larger neighborhoods increases the robustness of the compared classification methods to mislabeling. The best overall results are achieved by h-FNN and HIKNN.

Using large neighborhoods is not always possible, especially in class imbalanced data with rare categories and small dispersed class clusters. Using larger neighborhoods in such cases would breach the locality assumption and might compromise the precision on smaller classes, thereby reducing the overall system effectiveness. This is why achieving good performance for smaller neighborhood sizes is a highly desirable property.

7.4 Hubness-proportional Random Label Noise and Other Types of Classifiers

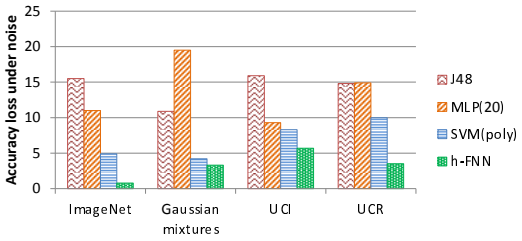
In addition to the experiments presented earlier, several standard non-kNN classifiers were compared under the given noise models, in order to see whether the hubness-proportional random label noise affects any of them in a different way than the uniform random label noise.

We have compared J48 decision trees, multi-layer perceptron (MLP) and SVM in 10-times 10-fold cross-validation on all previously examined datasets under all examined noise models for the noise rate of $\eta = 0.3$. We have used MLP in the following modes: MLP(5)

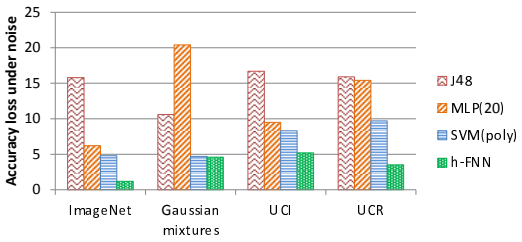
with 5 hidden nodes, MLP(20) with 20 hidden nodes and MLP(20,5) with two hidden layers of 20 and 5 hidden nodes, respectively. SVM was evaluated both with the polynomial kernel and the RBF kernel. The hyperparameters were determined based on a local search on subsets of the training data. WEKA implementations were used for the experiments (`http://www.cs.waikato.ac.nz/ml/weka/`). The experiment summaries are given in Table 6.

In most data domains, there were no substantial differences between algorithm performance under uniform random label noise and the hubness-proportional random label noise for these algorithms. The J48 implementation of decision trees seems to be highly susceptible to label noise in both noise models, unlike SVM.

The robustness of SVMs to hubness-proportional random label noise can be attributed to the fact that their classification performance relies mostly on the quality of support vectors and hubs in general are not always good support vectors in intrinsically high-dimensional data. The hubness ratio that is defined as the ratio between good and bad occurrence frequencies ($\frac{BN_k(x)}{GN_k(x)}$) was shown to be more relevant in past experiments on SVMs. The percentage of points with the hubness ratio close to 1 among the support vectors was shown to increase with increasing data dimensionality [20]. These examples lie closer to the separating hyperplane than other examples, on average.



(a) Uniform label noise



(b) Hubness-proportional label noise

Fig. 13 Absolute classification accuracy loss under the tested noise models, given for J48, MLP, SVM and h-FNN. The hubness-aware h-FNN classifier exhibits a higher overall robustness to label noise under the tested noise models for the noise rate of $\eta = 0.3$.

The performance of hubness-aware classifiers seems to be competitive when compared with the remaining non-kNN baselines. They clearly outperform the baselines on time series data where kNN is known to perform well. They seem only to be outperformed by SVM on the tested Gaussian mixtures which were specifically designed to be challenging for kNN classification, as discussed in the original paper [50].

The average absolute classification accuracy loss under noise for different approaches is shown in Figure 13. The hubness-aware h-FNN classifiers exhibits a higher robustness to both uniform and hubness-proportional random label noise for the tested noise level, compared to J48, MLP and SVM.

8 Conclusions and Future Work

High data dimensionality poses significant challenges for k-nearest neighbor classification. We have examined the influence of hubness as an aspect of the curse of dimensionality [5] on the problem of kNN classification with label noise. The emergence of hubs induces a change in the distribution of influence that affects the susceptibility of k-nearest neighbor methods to mislabeled training examples. Mislabeled hub points can potentially induce severe misclassification.

In order to evaluate the risk posed by unreliable hub labels, we have defined hubness-proportional random label noise, where the label flip probability η is modulated by the ratio of the neighbor occurrence frequency and the neighborhood size. The proposed noise model increases the probability of hub points being mislabeled. Our experiments reveal that the proposed noise model increases the average percentage of label mismatches in k-nearest neighbor sets and has a much greater impact on classifier performance than the uniform random label noise.

It was shown that the hubness-correlated label noise can arise either naturally from systematic errors in the data or in adversarial scenarios.

Hubness-proportional random label noise model can be used as an adversarial model that approximates a partially successful label flip attack that targets hub examples as most relevant points in a kNN-based system. We have demonstrated how certain properties of hubs can be used under certain standard representations and metrics to indirectly guess the hubness of data points in order to select hub targets for mislabeling. Our simulations on SMS message spam data indicate that the message totals of the TF-IDF weights can be used to pinpoint hubs in the data, as their weight is significantly lower than that of the regular and orphan messages. We have defined an inverse weight-proportional stochastic label noise model and were able to approximate the negative effects of hubness-proportional random label noise. Alternative adversarial hub-targeted scenarios were also discussed.

We have proposed to use the neighbor occurrence models for hubness-aware kNN classification of intrinsically high-dimensional data under label noise. Several recently proposed hubness-aware classifiers were compared to several kNN baselines in several different experimental setups: on correct data labels, on uniform label noise and on hubness-proportional label noise. Hubness-based fuzzy k-nearest neighbor classification (h-FNN) was determined to be most robust among the compared hubness-aware approaches across different experimental setups on multiple data domains, including quantized image representations, high-dimensional Gaussian mixtures, UCI data and UCR time series data.

The comparisons with SVM, J48 decision trees and the multi-layer perceptron (MLP) show that, while these non-kNN classification models are not in general susceptible to

hubness-proportional random label noise, hubness-aware classifiers are overall competitive under both examined noise models.

Hubness-based fuzzy k-nearest neighbor classification is implicitly robust to learning with label noise, due to the nature of the voting procedure and the way the hubness-based fuzzy votes are inferred. In future work we wish to explore combining the advantages of h-FNN and HIKNN with other explicit noise handling strategies, including but not limited to noise detection and removal. Strategies that properly filter hubs and ensure their label consistency should be seriously considered. Additionally, we wish to explore the options for combining noise-resilient boosting methods with hubness-aware classification.

Acknowledgements

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. Research partially performed within the framework of the grant of the Hungarian Scientific Research Fund (grant No. OTKA 111710).

References

1. Almeida, T.A., Hidalgo, J.M.G., Yamakami, A.: Contributions to the study of sms spam filtering: New collection and results. In: Proceedings of the 11th ACM Symposium on Document Engineering, DocEng '11, pp. 259–262. ACM, New York, NY, USA (2011)
2. Angluin, D., Laird, P.: Learning from noisy examples. Machine Learning **2**(4), 343–370 (1988)
3. Aucouturier, J.: Ten experiments on the modelling of polyphonic timbre. Doctoral dissertation. University of Paris (2006)
4. Aucouturier, J., Pachet, F.: Improving timbre similarity: How high is the sky? Journal of Negative Results in Speech and Audio Sciences **1** (2004)
5. Bellman, R.E.: Adaptive Control Processes - A Guided Tour. Princeton University Press, Princeton, New Jersey, U.S.A. (1961)
6. Biggio, B., Nelson, B., Laskov, P.: Support vector machines under adversarial label noise. In: ACML, pp. 97–112 (2011)
7. Bootkrajang, J., Kabán, A.: Learning kernel logistic regression in the presence of class label noise. Pattern Recognition (2014)
8. Bouckaert, R., Frank, E.: Evaluating the replicability of significance tests for comparing learning algorithms. In: Advances in Knowledge Discovery and Data Mining, pp. 3–12. Springer Berlin Heidelberg (2004)
9. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. Journal of Artificial Intelligence Research **11**, 131–167 (1999)
10. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: Time-series classification based on individualised error prediction. In: Proceedings of the 2010 13th IEEE International Conference on Computational Science and Engineering, pp. 48–54. IEEE Computer Society, Washington, DC, USA (2010)
11. Cantador, I., Dorronsoro, J.: Boosting parallel perceptrons for label noise reduction in classification problems. In: Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, pp. 586–593. Springer Berlin Heidelberg (2005)
12. Cormack, G.V., Gómez Hidalgo, J.M., Sánz, E.P.: Spam filtering for short messages. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07, pp. 313–320. ACM, New York, NY, USA (2007)
13. Cormack, G.V., Hidalgo, J.M.G., Sánz, E.P.: Feature engineering for mobile (sms) spam filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, pp. 871–872. ACM, New York, NY, USA (2007)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
15. Donmez, P., Carbonell, J.G., Schneider, J.: Efficiently learning the accuracy of labeling sources for selective sampling. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pp. 259–268. ACM, New York, NY, USA (2009)

16. 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. ACM, New York, NY, USA (2010)
17. Frenay, B., Doquire, G., Verleysen, M.: Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis **71**(0), 832 – 848 (2014)
18. Frenay, B., Verleysen, M.: Classification in the presence of label noise: a survey. Neural Networks and Learning Systems, IEEE Transactions on **PP**(99), 1–1 (2013)
19. Gasser M. Flexer A., S.D.: Hubs and orphans - an explorative approach. In: Proceedings of the 7th Sound and Music Computing Conference. ACM, New York, NY, USA (2010)
20. Georgios, K.: Investigating the Impact of Hubness on SVM Classifiers. University of the Aegean, Lesvos, Greece (2011)
21. Görnitz, N., Porbadnigk, A.K., Binder, A., Sannelli, C., Braun, M., Müller, K.R., Kloft, M.: Learning and evaluation in presence of non-iid label noise. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, pp. 293–302 (2014)
22. Guan, D., Yuan, W., Lee, Y.K., Lee, S.: Identifying mislabeled training data with the aid of unlabeled data. Applied Intelligence **35**(3), 345–358 (2011)
23. Hulse, J.V., Khoshgoftaar, T., Napolitano, A.: Skewed class distributions and mislabeled examples. In: Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on, pp. 477–482 (2007)
24. Ipeirotis, P., Provost, F., Sheng, V., Wang, J.: Repeated labeling using multiple noisy labelers. Data Mining and Knowledge Discovery **28**(2), 402–441 (2014)
25. Jin, R., Ghahramani, Z.: Learning with multiple labels (2003)
26. Karmaker, A., Kwek, S.: A boosting approach to remove class label noise. Int. J. Hybrid Intell. Syst. **3**(3), 169–177 (2006)
27. Keller, J.E., Gray, M.R., Givens, J.A.: A fuzzy k-nearest-neighbor algorithm. IEEE Transactions on Systems, Man and Cybernetics **15**(4), 580–585 (1985)
28. Khoshgoftaar, T.M., Seliya, N., Gao, K.: Detecting noisy instances with the rule-based classification model. Intelligent Data Analysis **9**(4), 347–364 (2005)
29. Ko, M.H., West, G., Venkatesh, S., Kumar, M.: Using dynamic time warping for online temporal fusion in multisensor systems. Information Fusion: Special Issue on Distributed Sensor Networks **9**(3), 370–388 (2008)
30. Kulesza, T., Amershi, S., Caruana, R., Fisher, D., Charles, D.: Structured labeling for facilitating concept evolution in machine learning. In: Proceedings of the 32nd annual ACM conference on Human factors in computing systems, pp. 3075–3084. ACM (2014)
31. Lallich, S., Muhlenbach, F., Zighed, D.: Improving classification by removing or relabeling mislabeled instances. In: Foundations of Intelligent Systems, pp. 5–15. Springer Berlin Heidelberg (2002)
32. Long, P., Servedio, R.: Random classification noise defeats all convex potential boosters. Machine Learning **78**(3), 287–304 (2010)
33. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
34. Nanopoulos, A., Radovanović, M., Ivanović, M.: How does high dimensionality affect collaborative filtering? In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 293–296. ACM, New York, NY, USA (2009)
35. Pechenizkiy, M., Tsymbal, A., Puuronen, S., Pechenizkiy, O.: Class noise and supervised learning in medical domains: The effect of feature extraction. In: CBMS, pp. 708–713. IEEE Computer Society (2006)
36. Porbadnigk, A., Gornitz, N., Sannelli, C., Binder, A., Braun, M., Kloft, M., Muller, K.R.: When brain and behavior disagree: Tackling systematic label noise in eeg data with machine learning. In: Brain-Computer Interface (BCI), 2014 International Winter Workshop on, pp. 1–4 (2014). DOI 10.1109/iww-BCI.2014.6782561
37. Radovanović, M.: Representations and Metrics in High-Dimensional Data Mining. Izdavačka knjižarnica Zorana Stojanovića, Novi Sad, Serbia (2011)
38. Radovanović, M., Nanopoulos, A., Ivanović, M.: Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In: Proceedings of the 26th International Conference on Machine Learning (ICML), pp. 865–872. Morgan Kaufmann, San Francisco, CA, USA (2009)
39. Radovanović, M., Nanopoulos, A., Ivanović, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research **11**, 2487–2531 (2010)
40. Rish, I.: An empirical study of the naive bayes classifier. In: Proceedings of the IJCAI Workshop on Empirical Methods in Artificial Intelligence. AAAI Press, Menlo Park, CA, USA (2001)
41. Schnitzer, D., Flexer, A., Schedl, M., Widmer, G.: Using mutual proximity to improve content-based audio similarity. In: ISMIR'11, pp. 79–84. International Society for Music Information Retrieval, Canada (2011)

42. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? improving data quality and data mining using multiple, noisy labelers. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pp. 614–622. ACM, New York, NY, USA (2008)
43. Srinivasan, A., Muggleton, S., Bain, M.: Distinguishing exceptions from noise in non-monotonic learning. In: Proceedings of the 2nd International Workshop on Inductive Logic Programming, pp. 97–107 (1992)
44. Stempfel, G., Ralaivola, L.: Learning svms from sloppily labeled data. In: Proceedings of the 19th International Conference on Artificial Neural Networks: Part I, ICANN '09, pp. 884–893. Springer-Verlag, Berlin, Heidelberg (2009)
45. Sukhbaatar, S., Fergus, R.: Learning from noisy labels with deep neural networks. arXiv preprint arXiv:1406.2080 (2014)
46. Tan, S.: Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Systems with Applications **28**, 667–671 (2005)
47. Thiel, C.: Classification on soft labels is robust against label noise. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 65–73. Springer Berlin Heidelberg (2008)
48. 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)
49. Tomašev, N., Mladenić, D.: Nearest neighbor voting in high dimensional data: Learning from past occurrences. Computer Science and Information Systems **9**, 691–712 (2012)
50. Tomašev, N., Mladenić, D.: Class imbalance and the curse of minority hubs. Knowledge-Based Systems (2013)
51. Tomašev, N., Mladenić, D.: Hub co-occurrence modeling for robust high-dimensional knn classification. In: Proceedings of the ECML conference. Springer-Verlag, Berlin, Germany (2013)
52. 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)
53. Tomašev, N., Radovanović, M., Mladenić, D., Ivanović, M.: Hubness-based fuzzy measures for high-dimensional k-nearest neighbor classification. International Journal of Machine Learning and Cybernetics (2013)
54. Tomašev, N., Radovanović, M., Mladenić, D., Ivanović, M.: The role of hubness in clustering high-dimensional data. IEEE Transactions on Knowledge and Data Engineering **99**(PrePrints), 1 (2013). DOI 10.1109/TKDE.2013.25
55. Tomašev, N., Rupnik, J., Mladenić, D.: The role of hubs in cross-lingual supervised document retrieval. In: Proceedings of the Pacific Asian Knowledge Discovery and Data Mining Conference, pp. 185–196. Springer-Verlag, Berlin / Heidelberg, Germany (2013)
56. Valizadegan, H., Tan, P.N.: Kernel based detection of mislabeled training examples. In: SDM. SIAM (2007)
57. Venkataraman, S., Metaxas, D., Fradkin, D., Kulikowski, C., Muchnik, I.: Distinguishing mislabeled data from correctly labeled data in classifier design. In: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '04, pp. 668–672. IEEE Computer Society, Washington, DC, USA (2004)
58. Verbaeten, S., Van Assche, A.: Ensemble methods for noise elimination in classification problems. In: Proceedings of the 4th International Conference on Multiple Classifier Systems, MCS'03, pp. 317–325. Springer-Verlag, Berlin, Heidelberg (2003)
59. Wang, J., Neskovic, P., Cooper, L.N.: Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recognition Letters **28**, 207–213 (2007)
60. Welinder, P., Perona, P.: Online crowdsourcing: rating annotators and obtaining cost-effective labels. In: Workshop on Advancing Computer Vision with Humans in the Loop (2010)
61. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd International Conference on Machine Learning, ICML '06, pp. 1033–1040. ACM, New York, NY, USA (2006)
62. 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 (2011)
63. Zeng, X., Martinez, T.R.: An algorithm for correcting mislabeled data. Intelligent Data Analysis **5**(6), 491–502 (2001)
64. Zhao, J., Xu, K.: Enhancing the robustness of scale-free networks. Computing Research Repository (2009)
65. Zhong, S., Tang, W., Khoshgoftaar, T.M.: Boosted noise filters for identifying mislabeled data (2005)
66. Zhu, X., Wu, X., Chen, Q.: Eliminating class noise in large datasets. In: Proceedings of International Conference on Machine Learning (ICML2003, pp. 920–927 (2003)

Table 2 Experiments on ImageNet quantized image data. Classification accuracy is given for kNN, NWKNN, AKNN, hw-kNN, HIKNN, NHBNN and h-FNN, for $k = 5$. The symbols •/◦ denote statistically significant worse/better performance ($p < 0.05$) compared to kNN. The best result in each line is in bold.

(a) Correct labels, no noise

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
iNet3ImbSift	88.0 ± 1.7	84.3 ± 1.9 •	**89.6 ± 1.7** ◦	89.2 ± 1.6	89.4 ± 1.5	86.3 ± 1.7	88.8 ± 1.6
iNet4ImbSift	70.3 ± 1.5	69.4 ± 1.5	71.4 ± 1.5	71.4 ± 1.5	**72.0 ± 1.4** ◦	69.4 ± 1.5	**72.0 ± 1.5** ◦
iNet5ImbSift	67.3 ± 1.8	63.9 ± 1.8 •	70.4 ± 1.6 ◦	70.1 ± 1.6 ◦	**70.6 ± 1.6** ◦	63.6 ± 1.6 •	69.8 ± 1.5 ◦
iNet6ImbSift	67.9 ± 1.7	65.5 ± 1.7 •	69.7 ± 1.5	70.1 ± 1.6 ◦	**71.2 ± 1.6** ◦	67.2 ± 1.7	70.7 ± 1.6 ◦
iNet7ImbSift	63.9 ± 2.1	63.3 ± 2.0	67.8 ± 2.0 ◦	67.7 ± 1.9 ◦	**68.0 ± 2.1** ◦	65.4 ± 2.0	67.9 ± 2.0 ◦
iNet3Sift	84.5 ± 1.4	83.3 ± 1.5	85.2 ± 1.4	85.6 ± 1.5 ◦	**85.7 ± 1.4** ◦	85.1 ± 1.5	85.0 ± 1.4
iNet4Sift	67.5 ± 1.2	67.0 ± 1.1	67.9 ± 1.2	68.8 ± 1.2 ◦	**69.7 ± 1.1** ◦	69.1 ± 1.2 ◦	69.3 ± 1.2 ◦
iNet5Sift	62.4 ± 1.4	61.8 ± 1.3	65.3 ± 1.3 ◦	65.7 ± 1.2 ◦	66.7 ± 1.2 ◦	65.0 ± 1.2 ◦	**67.4 ± 1.1** ◦
iNet6Sift	65.5 ± 1.3	64.8 ± 1.3	66.4 ± 1.2	67.2 ± 1.4 ◦	**68.1 ± 1.4** ◦	66.9 ± 1.2 ◦	67.5 ± 1.3 ◦
iNet7Sift	60.4 ± 1.1	60.5 ± 1.1	62.0 ± 1.0 ◦	62.8 ± 0.9 ◦	**63.4 ± 1.0** ◦	63.0 ± 0.9 ◦	63.3 ± 0.9 ◦
iNet3Haar	**99.9 ± 0.1**	99.8 ± 0.1	99.6 ± 0.2	**99.9 ± 0.1**	**99.9 ± 0.1**	99.8 ± 0.1	**99.9 ± 0.1**
iNet4Haar	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**
iNet5Haar	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**	**99.9 ± 0.0**
iNet6Haar	99.8 ± 0.1	**99.9 ± 0.0**	99.6 ± 0.1	99.8 ± 0.1	99.8 ± 0.1	99.8 ± 0.0	99.8 ± 0.1
iNet7Haar	99.8 ± 0.0	**99.9 ± 0.0**	99.8 ± 0.0	99.8 ± 0.0	**99.9 ± 0.0**	99.8 ± 0.0	**99.9 ± 0.0**
AVG	79.8	78.9	81.0	81.2	**81.6**	80.0	81.4

(b) Noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
iNet3ImbSift	74.8 ± 2.6	69.3 ± 2.6 •	85.0 ± 1.9 ◦	81.1 ± 2.1 ◦	84.1 ± 2.0 ◦	76.3 ± 2.3	**87.5 ± 1.7** ◦
iNet4ImbSift	59.4 ± 1.7	56.6 ± 1.9 •	64.0 ± 1.4 ◦	60.3 ± 1.7	65.8 ± 1.8 ◦	66.5 ± 1.7 ◦	**69.0 ± 1.8** ◦
iNet5ImbSift	56.7 ± 1.6	53.0 ± 1.6 •	64.5 ± 1.6 ◦	59.0 ± 1.5 ◦	64.5 ± 1.6 ◦	60.7 ± 1.8 ◦	**67.8 ± 1.6** ◦
iNet6ImbSift	59.0 ± 1.9	55.8 ± 2.0 •	65.7 ± 1.6 ◦	61.2 ± 1.8 ◦	65.0 ± 1.9 ◦	62.5 ± 1.8 ◦	**67.5 ± 1.8** ◦
iNet7ImbSift	56.8 ± 1.9	53.8 ± 2.0 •	61.2 ± 1.9 ◦	59.9 ± 2.0 ◦	63.0 ± 2.0 ◦	60.5 ± 1.9 ◦	**65.7 ± 2.0** ◦
iNet3Sift	72.7 ± 1.8	69.8 ± 1.8 •	78.1 ± 1.4 ◦	75.4 ± 1.5 ◦	78.8 ± 1.6 ◦	80.3 ± 1.6 ◦	**82.4 ± 1.4** ◦
iNet4Sift	57.8 ± 1.4	56.6 ± 1.5 •	60.0 ± 1.1 ◦	58.9 ± 1.6	63.3 ± 1.5 ◦	**66.6 ± 1.4** ◦	**66.6 ± 1.4** ◦
iNet5Sift	52.6 ± 1.3	51.2 ± 1.3 •	57.3 ± 1.3 ◦	55.2 ± 1.2 ◦	59.1 ± 1.1 ◦	62.0 ± 1.1 ◦	**63.9 ± 1.2** ◦
iNet6Sift	56.8 ± 1.3	54.4 ± 1.2 •	59.0 ± 1.4 ◦	58.6 ± 1.4 ◦	61.8 ± 1.3 ◦	62.8 ± 1.4 ◦	**64.6 ± 1.4** ◦
iNet7Sift	53.6 ± 1.1	53.3 ± 1.0	52.0 ± 0.9 •	53.3 ± 0.9	58.2 ± 1.0 ◦	60.4 ± 0.9 ◦	**60.6 ± 1.0** ◦
iNet3Haar	88.2 ± 1.2	85.8 ± 1.4 •	94.3 ± 1.0 ◦	93.2 ± 1.1 ◦	96.0 ± 0.7 ◦	98.3 ± 0.6 ◦	**99.0 ± 0.4** ◦
iNet4Haar	90.4 ± 0.8	89.5 ± 0.9	95.8 ± 0.6 ◦	94.8 ± 0.7 ◦	97.9 ± 0.4 ◦	**99.6 ± 0.1** ◦	99.5 ± 0.1 ◦
iNet5Haar	92.3 ± 0.7	91.2 ± 0.7 •	96.2 ± 0.9 ◦	95.0 ± 0.5 ◦	98.3 ± 0.3 ◦	**99.7 ± 0.1** ◦	99.6 ± 0.1 ◦
iNet6Haar	93.6 ± 0.7	91.3 ± 0.8 •	98.1 ± 0.4 ◦	95.8 ± 0.5 ◦	98.0 ± 0.3 ◦	99.2 ± 0.2 ◦	**99.4 ± 0.2** ◦
iNet7Haar	93.6 ± 0.5	92.7 ± 0.5 •	94.3 ± 0.6 ◦	95.8 ± 0.4 ◦	98.5 ± 0.2 ◦	**99.7 ± 0.0** ◦	99.6 ± 0.0 ◦
AVG	70.5	68.3	75.0	73.2	76.8	77.0	**79.6**

(c) $N_k(x)$-proportional noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
iNet3ImbSift	45.1 ± 2.3	35.3 ± 2.3 •	81.9 ± 1.8 ◦	69.4 ± 2.4 ◦	75.2 ± 2.2 ◦	69.2 ± 2.5 ◦	**86.2 ± 1.8** ◦
iNet4ImbSift	30.6 ± 1.4	28.0 ± 1.3 •	62.1 ± 1.8 ◦	46.0 ± 1.6 ◦	56.2 ± 1.6 ◦	64.3 ± 1.7 ◦	**68.8 ± 1.8** ◦
iNet5ImbSift	26.9 ± 1.6	23.8 ± 1.6 •	63.8 ± 1.8 ◦	42.3 ± 1.8 ◦	52.6 ± 1.7 ◦	56.9 ± 1.7 ◦	**67.3 ± 1.6** ◦
iNet6ImbSift	36.9 ± 1.7	32.2 ± 1.6 •	64.8 ± 1.7 ◦	51.2 ± 1.8 ◦	58.5 ± 1.6 ◦	61.3 ± 1.7 ◦	**67.8 ± 1.7** ◦
iNet7ImbSift	33.1 ± 2.0	28.6 ± 1.9 •	60.0 ± 2.5 ◦	47.9 ± 2.3 ◦	55.2 ± 2.2 ◦	58.1 ± 2.0 ◦	**65.2 ± 2.2** ◦
iNet3Sift	33.5 ± 2.1	29.8 ± 2.1 •	75.2 ± 1.9 ◦	55.7 ± 2.1 ◦	62.1 ± 2.1 ◦	76.7 ± 1.9 ◦	**80.5 ± 1.7** ◦
iNet4Sift	27.9 ± 1.3	27.2 ± 1.2	56.3 ± 1.4 ◦	43.6 ± 1.4 ◦	49.7 ± 1.5 ◦	65.3 ± 1.3 ◦	**65.8 ± 1.3** ◦
iNet5Sift	29.4 ± 1.2	27.6 ± 1.2 •	54.2 ± 1.3 ◦	40.3 ± 1.2 ◦	45.7 ± 1.3 ◦	60.7 ± 1.0 ◦	**63.0 ± 1.2** ◦
iNet6Sift	31.6 ± 1.3	28.8 ± 1.3 •	55.1 ± 1.4 ◦	45.2 ± 1.5 ◦	51.5 ± 1.6 ◦	61.5 ± 1.4 ◦	**64.2 ± 1.5** ◦
iNet7Sift	26.4 ± 0.9	26.0 ± 0.9	51.8 ± 1.1 ◦	38.6 ± 1.1 ◦	46.9 ± 1.2 ◦	59.5 ± 1.0 ◦	**60.3 ± 1.0** ◦
iNet3Haar	61.8 ± 2.1	58.5 ± 2.0 •	90.5 ± 1.3 ◦	87.7 ± 1.6 ◦	93.3 ± 1.1 ◦	98.4 ± 0.5 ◦	**99.5 ± 0.2** ◦
iNet4Haar	63.1 ± 1.3	63.5 ± 1.3	92.6 ± 1.2 ◦	88.0 ± 1.0 ◦	96.1 ± 0.6 ◦	99.7 ± 0.1 ◦	**99.7 ± 0.1** ◦
iNet5Haar	66.0 ± 1.2	65.3 ± 1.3	94.1 ± 0.9 ◦	89.2 ± 0.9 ◦	97.2 ± 0.4 ◦	99.6 ± 0.1 ◦	**99.7 ± 0.1** ◦
iNet6Haar	73.2 ± 1.2	70.0 ± 1.1 •	96.4 ± 0.5 ◦	91.5 ± 0.7 ◦	97.4 ± 0.4 ◦	99.4 ± 0.2 ◦	**99.6 ± 0.1** ◦
iNet7Haar	71.0 ± 0.9	70.8 ± 0.8	93.2 ± 0.7 ◦	90.3 ± 0.5 ◦	97.8 ± 0.3 ◦	99.8 ± 0.0 ◦	**99.8 ± 0.0** ◦
AVG	43.8	41.0	72.8	61.8	69.0	75.4	**79.2**

Table 3 Experiments on high-dimensional 10-class Gaussian mixtures. Classification accuracy is given for kNN, NWKNN, AKNN, hw-kNN, HIKNN, NHBNN and h-FNN, for $k = 5$. The symbols •/◦ denote statistically significant worse/better performance ($p < 0.05$) compared to kNN. The best result in each line is in bold.

(a) Correct labels, no noise

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
GM1	48.8 ± 3.0	49.5 ± 3.2	**79.2 ± 2.8** ◦	64.9 ± 2.8 ◦	63.1 ± 3.0 ◦	73.3 ± 2.4 ◦	69.9 ± 2.7 ◦
GM2	54.3 ± 2.9	55.7 ± 2.9 ◦	**82.4 ± 2.2** ◦	73.8 ± 2.8 ◦	69.3 ± 2.8 ◦	78.2 ± 2.4 ◦	76.5 ± 2.5 ◦
GM3	68.5 ± 2.6	65.7 ± 2.6 •	**87.0 ± 1.5** ◦	82.2 ± 1.7 ◦	81.3 ± 1.9 ◦	84.8 ± 1.7 ◦	84.4 ± 1.8 ◦
GM4	57.2 ± 2.2	58.5 ± 2.0	**83.4 ± 2.0** ◦	69.9 ± 2.3 ◦	68.7 ± 2.4 ◦	77.3 ± 2.1 ◦	75.0 ± 2.3 ◦
GM5	63.7 ± 2.6	62.6 ± 2.6	**83.2 ± 1.9** ◦	77.7 ± 2.0 ◦	77.0 ± 2.2 ◦	82.5 ± 2.1 ◦	81.9 ± 2.1 ◦
GM6	65.1 ± 2.7	63.4 ± 2.8 •	**80.0 ± 2.4** ◦	78.7 ± 2.1 ◦	76.8 ± 2.3 ◦	81.8 ± 2.3 ◦	**80.0 ± 2.5** ◦
GM7	69.9 ± 2.1	68.1 ± 2.2 •	**90.7 ± 1.6** ◦	82.3 ± 1.9 ◦	81.3 ± 1.9 ◦	85.7 ± 1.4 ◦	84.7 ± 1.8 ◦
GM8	72.3 ± 2.4	71.1 ± 2.5	**84.9 ± 1.9** ◦	79.6 ± 2.1 ◦	79.7 ± 2.2 ◦	83.6 ± 2.0 ◦	83.1 ± 2.0 ◦
GM9	62.3 ± 2.5	61.7 ± 2.5	**84.0 ± 1.8** ◦	73.0 ± 2.3 ◦	72.8 ± 2.3 ◦	81.8 ± 2.1 ◦	78.8 ± 2.2 ◦
GM10	63.3 ± 3.0	64.1 ± 2.8	**80.0 ± 2.5** ◦	75.3 ± 2.5 ◦	73.3 ± 2.4 ◦	81.3 ± 2.0 ◦	79.0 ± 2.3 ◦
AVG	62.6	62.0	**83.5**	75.7	74.3	81.0	79.3

(b) Noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
GM1	46.2 ± 3.1	44.5 ± 3.2	**71.3 ± 2.4** ◦	53.8 ± 3.0 ◦	58.2 ± 3.1 ◦	66.4 ± 2.5 ◦	66.1 ± 3.1 ◦
GM2	51.5 ± 2.5	51.1 ± 2.4	**75.2 ± 2.3** ◦	58.9 ± 2.3 ◦	64.0 ± 2.2 ◦	71.7 ± 2.1 ◦	73.4 ± 2.0 ◦
GM3	60.9 ± 2.4	58.0 ± 2.4 •	81.1 ± 2.0 ◦	67.4 ± 2.3 ◦	73.4 ± 2.3 ◦	81.5 ± 1.8 ◦	**82.0 ± 1.7** ◦
GM4	50.7 ± 2.7	50.7 ± 2.7	**71.7 ± 2.3** ◦	57.9 ± 2.6 ◦	62.8 ± 2.7 ◦	69.7 ± 2.2 ◦	71.1 ± 2.2 ◦
GM5	56.1 ± 2.1	53.4 ± 2.3 •	73.1 ± 1.9 ◦	63.0 ± 2.3 ◦	69.2 ± 2.0 ◦	76.3 ± 2.0 ◦	**77.4 ± 2.0** ◦
GM6	58.6 ± 2.9	56.7 ± 3.0 •	75.3 ± 2.7 ◦	66.4 ± 2.6 ◦	71.3 ± 2.8 ◦	75.2 ± 2.5 ◦	**76.7 ± 2.6** ◦
GM7	62.1 ± 2.8	59.1 ± 2.7 •	82.3 ± 2.1 ◦	68.8 ± 2.5 ◦	74.9 ± 2.3 ◦	81.6 ± 2.0 ◦	**82.8 ± 2.1** ◦
GM8	62.1 ± 2.4	59.8 ± 2.6 •	79.5 ± 2.1 ◦	65.7 ± 2.6 ◦	73.4 ± 2.1 ◦	78.9 ± 1.8 ◦	**79.9 ± 2.0** ◦
GM9	56.4 ± 2.3	56.1 ± 2.4	**76.5 ± 2.3** ◦	62.1 ± 2.3 ◦	67.7 ± 2.4 ◦	76.3 ± 2.3 ◦	76.0 ± 2.2 ◦
GM10	55.4 ± 2.7	54.8 ± 2.8	72.7 ± 2.5 ◦	61.3 ± 2.7 ◦	65.9 ± 2.6 ◦	73.9 ± 2.4 ◦	**75.0 ± 2.4** ◦
AVG	56.0	54.4	75.9	62.5	68.1	75.1	**76.0**

(c) $N_k(x)$-proportional noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
GM1	28.3 ± 3.1	24.5 ± 2.7 •	63.7 ± 3.1 ◦	42.3 ± 2.9 ◦	48.5 ± 2.8 ◦	**64.3 ± 2.9** ◦	62.2 ± 3.2 ◦
GM2	29.3 ± 2.7	25.8 ± 2.6 •	68.8 ± 2.2 ◦	46.7 ± 2.6 ◦	53.9 ± 2.8 ◦	**70.2 ± 2.6** ◦	69.7 ± 2.6 ◦
GM3	34.1 ± 2.5	31.8 ± 2.7 •	76.9 ± 2.2 ◦	54.5 ± 2.9 ◦	64.9 ± 2.4 ◦	79.7 ± 2.0 ◦	**80.3 ± 2.0** ◦
GM4	32.2 ± 2.4	30.3 ± 2.4 •	62.3 ± 2.8 ◦	49.7 ± 2.4 ◦	57.9 ± 2.5 ◦	69.2 ± 2.0 ◦	**71.7 ± 2.2** ◦
GM5	31.0 ± 2.2	28.2 ± 2.2 •	70.9 ± 2.2 ◦	51.1 ± 2.3 ◦	62.4 ± 2.2 ◦	75.1 ± 2.0 ◦	**76.7 ± 2.1** ◦
GM6	38.9 ± 2.6	37.0 ± 2.6 •	68.2 ± 2.7 ◦	55.5 ± 2.8 ◦	64.5 ± 2.5 ◦	73.3 ± 2.2 ◦	**74.8 ± 2.1** ◦
GM7	37.5 ± 2.4	34.7 ± 2.5 •	77.7 ± 1.9 ◦	58.6 ± 2.7 ◦	68.5 ± 2.5 ◦	81.5 ± 2.0 ◦	**81.6 ± 2.0** ◦
GM8	39.0 ± 2.8	36.1 ± 2.7 •	74.4 ± 2.4 ◦	57.7 ± 2.6 ◦	67.1 ± 2.5 ◦	**78.8 ± 2.1** ◦	77.3 ± 1.9 ◦
GM9	33.0 ± 2.4	31.7 ± 2.4	70.4 ± 2.4 ◦	50.5 ± 2.6 ◦	59.9 ± 2.7 ◦	74.8 ± 2.3 ◦	**76.0 ± 2.2** ◦
GM10	35.3 ± 2.9	32.8 ± 2.9 •	67.7 ± 2.7 ◦	52.3 ± 2.9 ◦	60.0 ± 2.7 ◦	73.9 ± 2.4 ◦	**76.6 ± 2.6** ◦
AVG	33.8	31.3	70.1	51.9	60.8	74.1	**74.7**

Table 4 Experiments on UCI data. Classification accuracy is given for kNN, NWKNN, AKNN, hw-kNN, HIKNN, NHBNN and h-FNN, for $k = 5$. The symbols •/○ denote statistically significant worse/better performance ($p < 0.05$) compared to kNN. The best result in each line is in bold.

(a) Correct labels, no noise

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
arrhythmia	60.4 ± 4.8	58.6 ± 5.2	57.9 ± 4.4 •	60.3 ± 4.7	61.2 ± 4.9	54.7 ± 4.5 •	**62.1 ± 5.1**
ozone	93.3 ± 0.9	91.0 ± 1.1 •	93.6 ± 0.9	93.5 ± 0.9	**93.7 ± 0.9**	88.7 ± 1.3 •	**93.7 ± 0.9**
ecoli	86.7 ± 3.7	85.4 ± 3.9	85.3 ± 4.0	86.4 ± 3.9	86.6 ± 3.8	85.9 ± 4.1	**87.3 ± 3.7**
gisette	96.2 ± 0.5	96.2 ± 0.5	**97.4 ± 0.4** ○	97.1 ± 0.4 ○	96.8 ± 0.4 ○	96.8 ± 0.5 ○	96.7 ± 0.5
glass	68.0 ± 6.9	68.1 ± 6.9	66.1 ± 7.4	67.6 ± 6.8	**68.7 ± 6.8**	65.2 ± 7.5	66.9 ± 6.9
haberman	71.2 ± 4.4	67.6 ± 5.3 •	**72.2 ± 5.1**	71.5 ± 4.5	71.1 ± 4.9	69.9 ± 5.6	71.8 ± 5.0
ionosphere	84.3 ± 4.2	85.2 ± 4.1	**94.7 ± 2.9** ○	88.4 ± 3.6 ○	87.8 ± 3.9 ○	92.2 ± 3.5 ○	89.8 ± 3.7 ○
mfeat-factors	95.3 ± 1.0	**95.6 ± 1.0** ○	92.9 ± 1.3 •	94.7 ± 1.1	**95.6 ± 1.0** ○	94.8 ± 1.1	95.2 ± 1.0
mfeat-fourier	**84.0 ± 1.5**	83.8 ± 1.4	80.2 ± 1.9 •	83.8 ± 1.7	83.8 ± 1.7	83.5 ± 1.5	83.8 ± 1.5
mfeat-karhunen	97.6 ± 0.7	97.6 ± 0.7	96.7 ± 0.7 •	97.4 ± 0.7	**97.7 ± 0.7**	97.5 ± 0.7	97.6 ± 0.7
Iris	96.5 ± 2.8	96.5 ± 2.7	95.8 ± 3.2	96.8 ± 2.7	96.9 ± 2.7	97.1 ± 2.6	**97.3 ± 2.7**
segment	94.6 ± 1.0	**96.5 ± 0.7** ○	93.7 ± 1.1 •	94.8 ± 1.0	96.2 ± 0.8 ○	95.2 ± 0.9	95.4 ± 0.9 ○
sonar	80.9 ± 6.3	82.4 ± 6.0	81.0 ± 6.3	79.7 ± 6.5	**82.7 ± 6.0**	80.1 ± 6.6	80.9 ± 6.2
vehicle	65.5 ± 3.6	**65.8 ± 3.7**	62.7 ± 3.3 •	65.0 ± 3.6	64.9 ± 3.6	61.8 ± 3.6 •	63.4 ± 3.3 •
ovarian	92.7 ± 3.9	93.0 ± 3.7	89.1 ± 4.4 •	92.7 ± 3.7	**93.4 ± 3.5**	**93.4 ± 3.0**	93.3 ± 3.6
AVG	84.5	84.2	84.0	84.7	**85.1**	83.8	85.0

(b) Noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
arrhythmia	54.2 ± 5.4	45.6 ± 5.5 •	43.8 ± 4.8 •	52.0 ± 5.2	58.4 ± 5.6 ○	42.9 ± 5.7 •	**59.1 ± 5.7** ○
ozone	77.1 ± 1.6	72.9 ± 1.7 •	85.3 ± 1.4 ○	81.0 ± 1.8 ○	81.5 ± 1.8 ○	66.5 ± 2.1 •	**86.3 ± 1.4** ○
ecoli	74.9 ± 5.3	67.3 ± 6.0 •	77.7 ± 5.5	78.8 ± 5.5 ○	81.7 ± 4.5 ○	78.0 ± 5.4	**84.1 ± 4.4** ○
gisette	79.8 ± 1.1	79.8 ± 1.1	89.7 ± 0.8 ○	84.9 ± 0.9 ○	85.3 ± 0.9 ○	**92.2 ± 0.7** ○	91.5 ± 0.7 ○
glass	61.4 ± 6.6	57.4 ± 6.4 •	50.2 ± 9.1 •	58.4 ± 6.9	**62.1 ± 6.7**	58.4 ± 7.4	60.7 ± 7.3
haberman	**65.5 ± 6.5**	61.4 ± 6.7 •	55.6 ± 6.8 •	65.1 ± 6.8	64.2 ± 7.4	60.7 ± 6.6 •	64.6 ± 7.2
ionosphere	75.9 ± 5.8	72.3 ± 5.7 •	76.4 ± 5.4	80.3 ± 5.3 ○	79.6 ± 4.9 ○	85.3 ± 4.5 ○	**85.4 ± 4.3** ○
mfeat-factors	89.3 ± 1.4	88.5 ± 1.4	88.5 ± 1.6	90.2 ± 1.2	93.2 ± 1.0 ○	93.7 ± 1.1 ○	**94.3 ± 1.0** ○
mfeat-fourier	77.8 ± 1.8	76.9 ± 1.8	76.2 ± 2.3 •	77.5 ± 1.8	81.3 ± 1.6 ○	82.7 ± 1.7 ○	**83.0 ± 1.7** ○
mfeat-karhunen	91.5 ± 1.3	90.9 ± 1.4	92.9 ± 1.0 ○	92.8 ± 1.3 ○	95.4 ± 0.9 ○	**96.6 ± 0.9** ○	**96.6 ± 0.8** ○
Iris	80.6 ± 8.7	77.6 ± 8.9	89.1 ± 5.7 ○	81.4 ± 7.6	82.2 ± 7.9	**89.3 ± 7.0** ○	83.6 ± 7.4
segment	87.4 ± 1.6	78.1 ± 1.9 •	86.4 ± 1.5	89.0 ± 1.6 ○	86.8 ± 1.6	91.3 ± 1.4 ○	**91.5 ± 1.3** ○
sonar	64.3 ± 7.8	66.2 ± 7.8	57.3 ± 7.5 •	66.2 ± 6.6	**66.9 ± 6.1** ○	63.1 ± 7.3	65.7 ± 7.2
vehicle	56.2 ± 3.9	55.3 ± 3.8	57.7 ± 3.3	58.7 ± 3.3 ○	59.8 ± 3.4 ○	59.3 ± 3.7 ○	**60.5 ± 3.6** ○
ovarian	80.6 ± 5.7	80.4 ± 5.6	76.0 ± 5.8 •	80.6 ± 5.4	82.2 ± 5.3	**83.5 ± 5.0** ○	82.2 ± 5.5
AVG	74.4	71.4	73.5	75.8	77.4	76.2	**79.3**

(c) $N_k(x)$-proportional noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
arrhythmia	43.3 ± 5.7	36.2 ± 5.2 •	45.2 ± 5.3	46.2 ± 5.2	56.4 ± 5.3 ○	39.2 ± 5.1	**60.2 ± 5.1** ○
ozone	64.9 ± 1.9	60.7 ± 2.0 •	83.3 ± 1.4 ○	79.0 ± 1.8 ○	78.0 ± 1.9 ○	64.5 ± 2.1	**85.0 ± 1.5** ○
ecoli	73.1 ± 5.3	62.3 ± 6.1 •	79.4 ± 5.0 ○	78.7 ± 5.0 ○	80.6 ± 4.7 ○	77.4 ± 4.9 ○	**83.9 ± 4.5** ○
gisette	49.6 ± 1.4	49.6 ± 1.4	78.9 ± 1.2 ○	76.4 ± 1.1 ○	71.6 ± 1.3 ○	90.7 ± 0.8 ○	**90.9 ± 0.8** ○
glass	59.9 ± 7.8	56.8 ± 7.6	55.3 ± 7.4 •	64.0 ± 7.4	65.5 ± 7.3 ○	61.8 ± 8.0	**66.3 ± 7.3** ○
haberman	57.2 ± 5.1	54.5 ± 5.5 •	**64.9 ± 5.3** ○	60.7 ± 5.0 ○	59.3 ± 5.5	57.7 ± 5.9	61.7 ± 5.2 ○
ionosphere	45.3 ± 5.8	44.3 ± 5.7	57.7 ± 6.0 ○	63.9 ± 5.0 ○	69.4 ± 5.2 ○	**79.8 ± 4.2** ○	78.7 ± 3.9 ○
mfeat-factors	82.8 ± 1.9	80.7 ± 2.0 •	89.8 ± 1.6 ○	88.8 ± 1.6 ○	92.7 ± 1.3 ○	93.7 ± 1.1 ○	**94.0 ± 1.2** ○
mfeat-fourier	71.7 ± 1.8	70.8 ± 1.7	77.1 ± 1.7 ○	78.5 ± 1.6 ○	81.7 ± 1.5 ○	82.9 ± 1.4 ○	**83.4 ± 1.3** ○
mfeat-karhunen	85.2 ± 1.8	84.8 ± 1.9	92.6 ± 1.3 ○	92.7 ± 1.3 ○	96.4 ± 0.8 ○	96.9 ± 0.7 ○	**97.3 ± 0.6** ○
Iris	79.6 ± 7.8	71.8 ± 8.7 •	87.8 ± 7.2 ○	91.1 ± 5.3 ○	89.7 ± 5.5 ○	**96.0 ± 3.6** ○	94.4 ± 4.2 ○
segment	85.7 ± 1.7	76.7 ± 1.9 •	85.0 ± 1.5	90.4 ± 1.4 ○	88.5 ± 1.3 ○	**92.9 ± 1.1** ○	92.3 ± 1.2 ○
sonar	64.0 ± 7.8	65.1 ± 7.7	65.2 ± 6.7	65.7 ± 6.7	66.1 ± 6.8	**69.1 ± 6.7** ○	66.6 ± 6.8
vehicle	55.3 ± 3.7	54.6 ± 3.6	57.5 ± 3.4	59.7 ± 3.4 ○	**61.3 ± 3.4** ○	59.1 ± 3.7 ○	61.0 ± 3.5 ○
ovarian	61.7 ± 6.5	62.8 ± 6.7	74.4 ± 6.3 ○	76.1 ± 5.9 ○	77.0 ± 6.0 ○	79.6 ± 5.4 ○	**81.7 ± 5.0** ○
AVG	65.3	62.1	72.9	74.1	75.6	76.1	**79.8**

Table 5 Experiments on UCR time series data. Classification accuracy is given for kNN, NWKNN, AKNN, hw-kNN, HIKNN, NHBNN and h-FNN, for $k = 5$. The symbols •/○ denote statistically significant worse/better performance ($p < 0.05$) compared to kNN. The best result in each line is in bold.

(a) Correct labels, no noise

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
CricketX	79.3 ± 2.8	81.9 ± 2.8 ○	76.0 ± 3.1 •	79.9 ± 2.4	**82.4 ± 2.6** ○	79.3 ± 2.9	80.9 ± 2.9 ○
CricketY	79.9 ± 2.8	82.3 ± 2.5 ○	73.4 ± 2.8 •	79.3 ± 2.7	**82.5 ± 2.5** ○	80.0 ± 2.7	81.9 ± 2.4 ○
CricketZ	80.9 ± 3.2	83.0 ± 2.7 ○	76.7 ± 3.1 •	80.7 ± 3.1	**83.5 ± 2.7** ○	79.6 ± 3.2	81.6 ± 2.8
FacesUCR	97.3 ± 0.7	**98.2 ± 0.6** ○	97.0 ± 0.7	97.2 ± 0.7	**98.2 ± 0.5** ○	97.5 ± 0.6	98.0 ± 0.6 ○
MedicalImages	79.9 ± 2.5	81.4 ± 2.3 ○	77.9 ± 2.8 •	80.3 ± 2.3	**81.9 ± 2.3** ○	74.9 ± 2.7 •	80.8 ± 2.3
MALLAT	98.6 ± 0.4	**98.9 ± 0.4** ○	98.6 ± 0.4	98.7 ± 0.5	**98.9 ± 0.4** ○	98.6 ± 0.4	98.8 ± 0.4
Motes	94.1 ± 1.3	94.9 ± 1.3 ○	94.0 ± 1.5	94.5 ± 1.2	**95.3 ± 1.2** ○	94.6 ± 1.2	95.0 ± 1.3 ○
OliveOil	82.4 ± 10.8	**89.4 ± 8.8** ○	80.3 ± 13.0	84.8 ± 11.1	87.4 ± 9.4	83.9 ± 11.8	83.3 ± 11.5
SonyAIBO	97.3 ± 1.3	97.3 ± 1.3	**98.7 ± 1.0** ○	97.9 ± 1.2	97.7 ± 1.3	97.8 ± 1.3	98.1 ± 1.2 ○
SonyAIBOII	96.4 ± 1.3	97.0 ± 1.2 ○	96.7 ± 1.3	96.8 ± 1.2	**97.2 ± 1.1** ○	96.9 ± 1.1	**97.2 ± 1.1** ○
SwedishLeaf	84.0 ± 2.0	84.6 ± 2.0	84.6 ± 2.0	84.9 ± 2.2	**85.6** ± 2.1 ○	**85.5 ± 2.0** ○	85.2 ± 2.0 ○
Symbols	97.7 ± 0.9	**98.3 ± 0.9** ○	97.5 ± 1.0	98.0 ± 0.9	98.2 ± 1.0 ○	97.9 ± 0.9	98.0 ± 1.0
SyntheticControl	99.2 ± 0.7	99.2 ± 0.7	97.5 ± 1.4 •	**99.5 ± 0.6** ○	99.2 ± 0.7	99.4 ± 0.6	99.4 ± 0.7
Trace	99.5 ± 1.0	**100.0 ± 0.0** ○	99.5 ± 1.0	98.9 ± 4.6 •	**100.0 ± 0.0** ○	99.5 ± 1.0	99.9 ± 0.3 ○
TwoPatterns	**100.0 ± 0.0**	**100.0 ± 0.0**	99.9 ± 0.0	99.9 ± 0.0	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
AVG	91.1	92.4	89.9	91.4	**92.5**	91.0	91.9

(b) Noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
CricketX	74.0 ± 3.7	67.7 ± 3.8 •	66.1 ± 3.8 •	73.0 ± 3.7	76.1 ± 3.4 ○	75.8 ± 3.3	**78.1 ± 3.0** ○
CricketY	72.1 ± 3.2	69.2 ± 3.3 •	63.1 ± 3.9 •	71.0 ± 3.1	75.5 ± 3.2 ○	75.8 ± 3.2 ○	**78.0 ± 3.2** ○
CricketZ	74.4 ± 3.6	70.1 ± 3.5 •	62.9 ± 3.8 •	73.8 ± 3.5	77.1 ± 3.0 ○	76.2 ± 3.8	**78.4 ± 3.2** ○
FacesUCR	93.9 ± 1.0	89.8 ± 1.5 •	92.4 ± 1.2 •	94.4 ± 1.1	95.8 ± 0.8 ○	96.4 ± 0.8 ○	**97.2 ± 0.7** ○
MedicalImages	72.6 ± 2.4	59.6 ± 3.0 •	73.5 ± 2.8	72.9 ± 2.2	73.1 ± 2.5	64.1 ± 3.1 •	**76.9 ± 2.2** ○
MALLAT	92.8 ± 1.1	83.7 ± 1.5 •	95.3 ± 1.1 ○	94.6 ± 1.1 ○	91.5 ± 1.2 •	**97.9 ± 0.6** ○	97.4 ± 0.7 ○
Motes	79.2 ± 2.5	74.9 ± 2.8 •	83.8 ± 2.3 ○	84.1 ± 2.4 ○	82.4 ± 2.6 ○	**87.1 ± 2.3** ○	85.7 ± 2.3 ○
OliveOil	72.8 ± 13.7	74.4 ± 12.4	76.1 ± 12.2	74.0 ± 13.6	73.7 ± 15.7	78.8 ± 13.9 ○	**79.6 ± 13.8** ○
SonyAIBO	77.1 ± 3.8	76.9 ± 3.8	88.3 ± 2.9 ○	86.5 ± 3.0 ○	88.3 ± 3.0 ○	**93.0 ± 2.4** ○	92.4 ± 2.7 ○
SonyAIBOII	79.8 ± 2.9	78.0 ± 3.2	85.2 ± 2.7 ○	85.0 ± 2.7 ○	86.0 ± 2.6 ○	88.5 ± 2.4 ○	**89.2 ± 2.1** ○
SwedishLeaf	79.4 ± 2.7	77.0 ± 2.8 •	75.5 ± 2.9 •	80.0 ± 2.6	82.5 ± 2.6 ○	**83.5 ± 2.5** ○	83.3 ± 2.6 ○
Symbols	93.6 ± 1.7	84.8 ± 2.4 •	90.7 ± 1.9 •	95.0 ± 1.4 ○	93.7 ± 1.4	**97.4 ± 0.9** ○	96.6 ± 1.1 ○
SyntheticControl	92.2 ± 2.5	92.2 ± 2.6	91.9 ± 2.4	92.5 ± 2.8	96.8 ± 1.7 ○	**99.0 ± 0.9** ○	98.4 ± 1.1 ○
Trace	89.0 ± 5.8	86.5 ± 5.9 •	93.2 ± 4.0 ○	94.1 ± 4.4 ○	93.0 ± 4.2 ○	**97.6 ± 2.8** ○	95.3 ± 3.6 ○
TwoPatterns	91.2 ± 0.8	90.6 ± 0.9 •	95.7 ± 0.5 ○	95.4 ± 0.5 ○	97.5 ± 0.4 ○	**99.5 ± 0.2** ○	99.3 ± 0.2 ○
AVG	82.3	78.4	82.3	84.4	85.5	87.4	**88.4**

(c) $N_k(x)$-proportional noise rate $\eta = 0.3$

Data set	kNN	NWKNN	AKNN	hw-kNN	HIKNN	NHBKNN	h-FNN
CricketX	68.4 ± 3.7	63.3 ± 3.8 •	63.0 ± 4.0 •	72.5 ± 3.3 ○	77.8 ± 3.4 ○	76.4 ± 3.2 ○	**78.5 ± 3.0** ○
CricketY	68.1 ± 4.0	62.3 ± 4.2 •	63.2 ± 3.8 •	72.4 ± 3.5 ○	75.3 ± 3.4 ○	76.3 ± 3.5 ○	**79.0 ± 3.4** ○
CricketZ	69.1 ± 4.1	63.4 ± 4.4 •	62.0 ± 4.2 •	72.7 ± 4.2 ○	76.0 ± 3.9 ○	74.7 ± 3.9 ○	**76.7 ± 3.9** ○
FacesUCR	87.0 ± 1.6	82.5 ± 1.8 •	90.7 ± 1.4 ○	93.1 ± 1.1 ○	96.0 ± 0.8 ○	96.7 ± 0.8 ○	**97.5 ± 0.6** ○
MedicalImages	70.9 ± 2.5	58.9 ± 3.0 •	71.6 ± 3.0	73.5 ± 2.7 ○	73.4 ± 2.8 ○	63.3 ± 3.3 •	**76.8 ± 2.8** ○
MALLAT	85.7 ± 1.6	75.8 ± 2.2 •	96.0 ± 0.9 ○	93.9 ± 0.9 ○	91.8 ± 1.2 ○	**97.8 ± 0.6** ○	97.7 ± 0.6 ○
Motes	66.8 ± 2.6	65.5 ± 2.6	80.5 ± 2.5 ○	80.0 ± 2.3 ○	78.7 ± 2.2 ○	**83.3 ± 2.1** ○	83.0 ± 2.0 ○
OliveOil	63.4 ± 15.7	65.1 ± 15.1	69.3 ± 14.4	77.2 ± 12.6 ○	80.6 ± 10.9 ○	**81.0 ± 11.7** ○	79.5 ± 11.0 ○
SonyAIBO	61.8 ± 4.1	61.4 ± 4.0	85.0 ± 3.2 ○	81.6 ± 3.5 ○	82.5 ± 3.4 ○	89.3 ± 2.5 ○	**89.5 ± 2.7** ○
SonyAIBOII	66.7 ± 3.4	65.9 ± 3.2	83.7 ± 2.7 ○	83.2 ± 2.9 ○	85.0 ± 2.6 ○	89.6 ± 2.6 ○	**91.1 ± 2.3** ○
SwedishLeaf	72.2 ± 3.0	69.8 ± 2.8 •	75.5 ± 2.8 ○	77.1 ± 3.0 ○	82.3 ± 2.6 ○	83.2 ± 2.3 ○	**84.0 ± 2.4** ○
Symbols	81.6 ± 3.1	72.3 ± 3.1 •	91.8 ± 1.6 ○	92.3 ± 1.8 ○	92.3 ± 1.8 ○	**96.8 ± 1.2** ○	96.5 ± 1.3 ○
SyntheticControl	74.2 ± 4.0	73.7 ± 4.1	90.4 ± 2.6 ○	91.4 ± 2.3 ○	97.5 ± 1.3 ○	99.2 ± 0.7 ○	**99.2 ± 0.7** ○
Trace	82.0 ± 5.9	80.9 ± 6.1	89.2 ± 4.8 ○	95.9 ± 3.3 ○	94.6 ± 3.9 ○	**97.6 ± 2.4** ○	97.0 ± 2.8 ○
TwoPatterns	75.1 ± 1.3	74.4 ± 1.3	93.8 ± 0.7 ○	92.9 ± 0.7 ○	97.2 ± 0.4 ○	**99.7 ± 0.1** ○	99.6 ± 0.1 ○
AVG	72.9	69.0	80.4	83.3	85.4	87.0	**88.4**

Table 6 Experiments on non-kNN classifiers: decision trees, neural networks and support vector machines. Classification accuracy is given for J48, MLP and SVM with the polynomial and RBF kernel. The three included result columns for the multi-layer perceptron correspond to its operating modes using 5 hidden nodes (MLP(5)), 20 hidden nodes (MLP(20)) and two layers of 20 and 5 hidden nodes, respectively (MLP(20,5)). In most data domains, there were no significant differences between the algorithm performance under the uniform random label noise model and the hubness-proportional random label noise model.

Domain	Noise Model	J48	MLP (5)	MLP (20)	MLP (20, 5)	SVM (poly)	SVM (RBF)
ImageNet	no noise	73.3	73.1	70.9	67.1	85.0	84.8
	random	57.8	65.7	59.9	64.1	80.3	79.9
	hubness-pr.	57.5	68.2	64.7	65.8	80.0	80.0
Gaussian mixtures	no noise	47.1	84.0	95.1	91.1	96.4	96.5
	random	36.2	69.4	75.6	71.2	92.2	92.6
	hubness-pr.	36.5	69.0	74.7	70.4	91.7	91.9
UCI	no noise	82.7	82.1	83.5	81.4	87.1	78.1
	random	66.8	74.9	74.2	72.2	78.8	70.0
	hubness-pr.	66.0	75.0	74.0	72.4	78.8	69.1
UCR	no noise	77.0	77.3	83.8	79.7	85.1	81.7
	random	57.2	67.1	68.9	64.7	75.1	75.0
	hubness-pr.	56.1	67.3	68.4	65.5	75.4	74.6