A Probabilistic Approach to Nearest-Neighbor Classification: Naive Hubness Bayesian kNN

Nenad Tomašev  
Institute Jožef Stefan  
Jamova 39  
Ljubljana, Slovenia  
nenad.tomasev@ijs.si

Miloš Radovanović  
Department of Mathematics and Informatics  
Trg D. Obradovića 4  
Novi Sad, Serbia  
radacha@DMI.uns.ac.rs

Dunja Mladenić  
Institute Jožef Stefan  
Jamova 39  
Ljubljana, Slovenia  
dunja.mladenic@ijs.si

Mirjana Ivanović  
Department of Mathematics and Informatics  
Trg D. Obradovića 4  
Novi Sad, Serbia  
mira@DMI.uns.ac.rs

ABSTRACT
Most machine-learning tasks, including classification, involve dealing with high-dimensional data. It was recently shown that the phenomenon of hubness, inherent to high-dimensional data, can be exploited to improve methods based on nearest neighbors (NNs). Hubness refers to the emergence of points (hubs) that appear among the k NNs of many other points in the data, and constitute influential points for kNN classification. In this paper, we present a new probabilistic approach to kNN classification, naive hubness Bayesian k-nearest neighbor (NHBNN), which employs hubness for computing class likelihood estimates. Experiments show that NHBNN compares favorably to different variants of the kNN classifier, including probabilistic kNN (PNN) which is often used as an underlying probabilistic framework for NN classification, signifying that NHBNN is a promising alternative framework for developing probabilistic NN algorithms.

Categories and Subject Descriptors  
1.2.6 [Artificial Intelligence]: Learning; 1.5.1 [Pattern Recognition]: Models

General Terms  
Algorithms, Experimentation, Theory

1. INTRODUCTION
High dimensionality is known to pose certain limitations on the effectiveness of many machine-learning algorithms. Nowadays, the majority of data one has to deal with is, unfortunately, high-dimensional. This includes text, images, video, medical records, data streams etc. All such data is sparse, since data size requirements for proper density estimates rise exponentially with the number of dimensions. Furthermore, there exists the phenomenon of distance concentration, which has been thoroughly explored in the past [4]. In high-dimensional spaces, distances between points drawn from the same distribution tend to become relatively similar to one another, so some doubts have been raised about the usefulness of the notion of nearest neighbors in high-dimensional settings [3]. Nevertheless, k-nearest neighbor (kNN) classifier is a widely used classification method that operates by a majority vote on a k-nearest-neighbor set [1].

Hubness is a phenomenon related specifically to nearest-neighbor methods. Let \( N_k(x) \) denote the number of k-occurrences of point \( x \), i.e., the number of times \( x \) occurs in \( k \)-nearest-neighbor lists of other data points. When the (intrinsic) dimensionality of data is high, significant skew in the distribution of \( N_k(x) \) can be observed. This leads to the emergence of hubs, data points which are included in many more neighbor lists than other points. The number of k-occurrences of point \( x \), \( N_k(x) \), will be referred to as the hubness score of \( x \). A hubness-based weighted voting scheme for kNN, proposed in [8], was shown to often lead to improvement when working with high-dimensional data.

Our goal is to examine if it is possible to incorporate hubness information for Bayesian class prediction. Hubness is interpreted as prior information about occurrences of elements in \( k \)-neighbor sets and is used to set up a Bayesian \( k \)-nearest neighbor model. The proposed approach is fundamentally different from related methods, from which a representative selection is reviewed in Section 2. Section 3 presents our probabilistic framework for using hubness, and proposes the naive hubness Bayesian \( k \)-nearest neighbor classification algorithm (NHBNN). Experimental comparison of NHBNN with related approaches is performed in Section 4, while the last section gives concluding remarks and an outline for future work.

2. RELATED WORK
Introducing weights to votes in kNN is not all that uncommon. Defining vote weights \( w_k(x_i) \) based on hubness
was introduced in [8]. The hubness of each point can be decomposed into two parts, which will be referred to as good hubness and bad hubness, respectively: \( N_h(x_i) = GN_h(x_i) + BN_h(x_i) \). Good hubness represents those \( k \)-occurrences of \( x_i \) where labels match, i.e., the label of \( x_i \) and the label of the data point which has \( x_i \) in its neighbor set are the same. Bad hubness, on the other hand, originates from label mismatches. In high-dimensional spaces, hubs become increasingly influential, since they are found in many \( k \)-neighbor sets. We can distinguish between good hubs and bad hubs based on how often label mismatches occur. Bad hubs exhibit a distinctly detrimental influence. Hubness-weighted \( k\)NN [8] aims to reduce the influence of bad hubs, which is how it improves on the basic \( k\)NN classifier. The approach is as follows. Standardized bad hubness is computed as: \( h_b(x_i,k) = (B N_h(x_i) - \mu_{BN_h})/\sigma_{BN_h} \), where \( \mu_{BN_h} \) is mean bad hubness and \( \sigma_{BN_h} \) the standard deviation. The weight associated with \( x_i \) is then \( w_b(x_i) = e^{-h_b(x_i,k)} \).

One of the disadvantages of the standard \( k\)NN classifier is that it does not output meaningful probabilities associated with class prediction. Performing a count of the number of votes given to each class and making the class probabilities somehow proportional to those counts is, admittedly, simple enough – but is not a really good measure of confidence, somehow proportional to those counts is, admittedly, simple enough. A Bayesian solution to this problem was proposed in [6] and is known as probabilistic \( k \)-nearest neighbor (PNN). It has been shown that PNN does not offer an improvement in accuracy over the basic \( k\)NN. Extensions of PNN have been proposed which improve its overall performance [2, 5].

3. BAYESIAN VIEW ON HUBNESS

In our approach, we wished to exploit the information provided by reverse nearest neighbors, while being aware of hubs in high dimensions. The way in which hubness had previously been exploited in hubness-weighted \( k\)NN is somewhat rigid, so we decided to extend it. In multi-class scenarios, all label mismatches in \( k \)-occurrences of point \( x \) were collapsed to a single quantity – bad hubness. Our idea was to observe class-specific label mismatches. This led us to define class hubness. Thereby, the total number of \( k \)-occurrences is decomposed in the following manner: \( N_h(x_i) = \sum_{c=1}^{C} N_{h,c}(x_i) \). \( N_{h,c}(x_i) \) signifies the number of \( k \)-occurrences of point \( x_i \) in neighborhoods of elements of class \( c \). Let \( D_h(x) \) denote the \( k \)-neighbor set of \( x \). We interpret the \( k \)-occurrence of \( x_i \) in \( D_h(x) \) as an event which carries information that gives certain evidence to class affiliation of \( x \).

Before we continue, we should first emphasize the main difficulty of such reverse nearest neighbor reasoning. Some points may never appear in \( k\)NN sets and therefore have no reverse nearest neighbors. In high dimensional data, this happens more often, due to the hubness phenomenon. Such points which appear either never or rarely as nearest neighbors we will refer to as anti-hubs.

On the other hand, in hubs there is a lot of occurrence information to infer from. Suppose that \( x_h \) is one such hub and \( c_h \) its label. In some cases, the labels of the majority of reverse neighbors of \( x_h \) might be different from \( c_h \). In such cases, when we see \( x_h \in D_h(x) \), this should not be considered a strong evidence in favor of the class \( c_h \). This is why we feel that taking class hubness into consideration may prove quite beneficial.

Let \( D = (X, Y) \) be the data set, so that \( y_i \in Y \) is the label of \( x_i \in X \). We would like to estimate \( p(y = c | D_h(x)) \). We view \( D_h(x) \) as being the result of \( k \) random draws from \( D \) where each \( x_i \) has probability \( p(x_i | D_h(x), x) \) of being drawn at time \( t \), where \( t \in \{1, 2, \ldots, k \} \). Since the concepts of \( t \)-neighbor and \((t+1) \)-neighbor are quite similar, we also approximate by disregarding the exact position of \( x_i \) in the list, which endows us with more information to work with.

Let \( x_i \in D \) and \( x_{it} \), for \( t \in \{1, 2, \ldots, k \} \) be the \( k \) nearest neighbors of \( x_i \). We focus on a naive Bayesian estimate which takes the form shown in Eq. 1:

\[
p(y_i = c | D_h(x_i)) \propto p(y_i = c) \prod_{t=1}^{k} p(x_{it} \in D_h(x_i) | y_i = c).
\] (1)

The independence assumption obviously does not hold in most cases. However, it was shown in the past that this is not necessary for the naive Bayesian approach to work [9]. The main problem lies in estimating the probabilities on the right-hand side of the expression, especially for anti-hubs. We will treat them as a special case. Also, each point was included to its own neighbor set at the 0th position, ensuring that each \( N_h(x_i) \geq 1 \).

For clarity we shorten the notation by denoting \( p(x_{it} \in D_h(x_i) | y_i = c) \) as \( p_{x_i,h,c}(x_{it}) \), and by \( n_c \) the number of elements of class \( c \). Also, let \( N_{h,c}(c_1) \) be the total number of \( k \)-occurrences of elements from class \( c_1 \) in neighborhoods of elements belonging to \( c \).

Let \( (x, y) \) be an element from the neighborhood of \( x_i \). Also, let \( (x_g, y_g) \in \bigcup_{y_g = y_{it} = \emptyset} D_h(x_i) \). The derivation of the probability estimate is given in Eq. 2:

\[
p_{x_i,h,c}(x_{it}) = p_{x_i,h}(y = y_{it} | y_i = c) \cdot p_{x_i,k}(x = x_{it} | y = y_{it})
\approx p_{c,h}(y = y_{it}) \cdot p_{c,k}(x = x_{it} | y = y_{it})
= \frac{N_{h,c}(y_{it})}{n_c(k+1)} \sum_{j:y_j = y_{it}} N_{h,c}(x_{jt}) = \frac{N_{h,c}(x_{it})}{n_c(k+1)}
\approx \frac{N_{h,c}(x_{it}) + \lambda}{n_c(k+1) + \lambda|D|} = \bar{p}_{x_i,h,c}(x_{it}).
\] (2)

\( \lambda \) is the Laplace estimator, ensuring non-zero probabilities. When dealing with anti-hubs, however, the outlined approximation can not be expected to yield reliable probability estimates. This is why for anti-hubs we partly infer the probability from what we know about the typical points from the same class and their hubness, as shown in Eq. 3.

\[
p_{x_i,h,c}(x_{it}) = \begin{cases} \bar{p}_{x_i,h,c}(x_{it}), & \text{if } N_h(x_{it}) > \theta, \\ \bar{u}_{x_i,h,c}(x_{it}), & \text{if } N_h(x_{it}) \leq \theta. \end{cases}
\] (3)

\( \bar{u}_{x_i,h,c}(x_{it}) = \alpha_{h} \cdot \bar{p}_{x_i,h,c}(x_{it}) + (1 - \alpha_{h}) \cdot \bar{p}_{x_i,h,c}(x_{it}). \)

We propose two approaches to defining \( \bar{p}_{x_i,h,c}(x_{it}) \). Both are based on the approximation given in Eq. 4:

\[
N_{h,c}(x_{it}) \approx N_{h,c}(y_{it})/n_{y_{it}}.
\] (4)

We will refer to using this estimate as the global approximate approach. We could also take local information into account by inferring \( N_{h,c}(y_{it}) \) from some \( k_{est} \)-neighborhood of \( x_{it} \) instead, as given in Eq. 5. We will refer to this estimate as the local approximate approach. In our experiments we invariably used \( k_{est} = 20 \).
In the end we obtain $\hat{y}_{x_i,c,k}(x_{it})$ from Eq. 3 by plugging either the local or the global class hubness estimate instead of the original $N_{k,c}(x_{it})$ into $\hat{y}_{x_i,c,k}(x_{it})$. In other words:

$$
\hat{y}_{x_i,c,k}(x_{it}) = \begin{cases} 
N_{k,c}(y_{it}) + \lambda, & \text{if global,} \\
N_{k,c}(y_{it} | x_{it}) + \lambda, & \text{if local,}
\end{cases}
$$

(5)

Class affiliation of $x$ is determined as $y = \arg \max_y p(y = c | D_k(x))$. In case of an unlikely tie we assign according to the prior class probabilities, i.e., the majority class. We name the algorithm which performs this classification, based on approximations given in Eqs. 1, 2, 3, 4, and 5, the naive hubness Bayesian $k$-nearest neighbor (NHBNN). Several parameters are required for the algorithm to work, but they can also be inferred from the training set by leave-one-out validation.

Algorithm 1 NHBNN: Training

Input: $(X, Y, k, k_{est}, \theta, \alpha_k)$
training set $T = (X, Y) \subseteq \mathbb{R}^{d \times n}$
number of neighbors $k \in \{1, 2, \ldots, n-1\}$
local estimate width $k_{est} \in \{1, 2, \ldots, n-1\}$
hubness score threshold $\theta \in [0, \max_{x_i \in T} N_k(x_i))$
hubness prevalence $\alpha_k \in [0, 1]$

Train:
calculateClassPriors()
kNeighbors = findNeighborSets$(T, k)$
kEstNeighbors = findNeighborSets$(T, k_{est})$
for all $(x_i, y_i) \in (X, Y)$ do
  for all $c = 1 \ldots C$ do
    count $N_{k,c}(x_i)$
    $N_{k,c}(y_i) = N_{k,c}(y_i) + 1$
  for all $\ell = 1 \ldots C$ do
    count $N_{x_i,k_{est},k,c}(\ell)$
end for
end for
$N_k(x_i) = \sum_{c=1}^C N_{k,c}(x_i)$
end for
{only after the global estimates are calculated}

{we finish by calculating the probabilities}
for all $(x_i, y_i) \in (X, Y)$ do
  calculate $\hat{y}_{x_i,c,k}(x_{it})$ by Eq. 3
end for

4. EXPERIMENTAL EVALUATION

The data sets used for evaluation are summarized in Table 1. Eighteen data sets of various dimensionalities originate from the University of California, Irvine (UCI) Machine Learning Repository, two high-dimensional gene expression microarray data sets (colonTumor, ovarian) are from the Kent Ridge (KR) Bio-Medical Data Set Repository, and five data sets represent subsets of images taken from the ImageNet repository. We extracted SIFT features from all images, formed quantized 400-dimensional representations [11], and then appended 16-bin color distribution information to obtain the final 416-dimensional feature vectors. More details on these five image data sets are given in [10].

All data sets are summarized in Table 1, whose columns respectively denote data-set name, size, dimensionality ($d$), number of classes ($C$), and the observed skewness of the distribution of $N_5$ ($S_{N_5}$) [8]. Euclidean distance was used as a dissimilarity measure for the first 20 data sets from the table and Manhattan distance for the last five image data sets. We see that the hubness phenomenon is much more pronounced in image data.

Experiments were performed in three different setups. The first two setups used only the first 20 listed data sets. The image data sets were considered separately, for reasons that will be explained later.

In the first round of experiments we examined how NHBNN performs for a fixed $k$ value and some fixed parameter configuration over all data sets. The conclusion from the results from this round (omitted due to space considerations) is that global and local versions of NHBNN exhibit similar performance, and are both superior to $k$NN and hw-$k$NN.

In the second round of experiments our aim was to compare the algorithms when automatically searching for proper parameters (as well as estimation schemes) on each fold of 10 times 10-fold cross validation, with the recorded accuracies compared using the corrected resampled $t$-test. Here, we also included PNN in the comparison with NHBNN. For $k$NN, hw-$k$NN and NHBNN we used $k \in \{1, 2, \ldots, 10\}$. For PNN we allowed a larger maximum neighborhood size, this being the way it was used in the original article [6]. The maximum $k$ for PNN was set to $\min(500, \lceil n/4 \rceil)$, where $n$ is the data-set size. We set PNN to use 1500 evaluation points. The results are shown in Table 2.

In both experimental settings, NHBNN performs quite well. Significant improvements over $k$NN were observed, with 5-1 and 4-0 in statistically significant wins in the two observed experimental setups, respectively. NHBNN compares favorably to PNN, since it achieves 7-0 in statistically significant wins.

In the first two experimental setups no significant difference between hw-$k$NN and NHBNN was observed. This is not all that surprising, since most UCI data sets are binary classification problems, low-to-medium (intrinsically) dimensional with small skewness in the distribution of $k$-occurrences, as seen in Table 1. Furthermore, many of the
Table 2: Comparison of NHBNN with several other classifiers, featuring automatic search for best parameters and approaches on each cross-validation fold. The symbols ¥/• denote statistically significant better/worse performance with respect to NHBBN, at significance level $p < 0.01$.

<table>
<thead>
<tr>
<th>Data set</th>
<th>$k$NN</th>
<th>hw-$k$NN</th>
<th>PNN</th>
<th>NHBBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>arccne</td>
<td>70.5±14.6</td>
<td>73.5±14.3</td>
<td>74.±14.0</td>
<td>74.9±14.2</td>
</tr>
<tr>
<td>colonlmumor</td>
<td>64.9±19.9</td>
<td>73.3±19.2</td>
<td>72.6±18.8</td>
<td>74.5±18.1</td>
</tr>
<tr>
<td>dexter</td>
<td>60.2±19.1</td>
<td>69.7±9.3</td>
<td>65.3±13.3</td>
<td>66.9±7.8</td>
</tr>
<tr>
<td>diabetes</td>
<td>76.6±5.2</td>
<td>72.0±4.6</td>
<td>73.3±5.1</td>
<td>71.4±4.6</td>
</tr>
<tr>
<td>ecoli</td>
<td>85.4±5.9</td>
<td>85.1±5.9</td>
<td>82.8±6.9</td>
<td>83.9±6.4</td>
</tr>
<tr>
<td>glass</td>
<td>73.1±10.2</td>
<td>68.8±9.6</td>
<td>63.3±9.3</td>
<td>68.1±10.9</td>
</tr>
<tr>
<td>ionosphere</td>
<td>89.9±4.9</td>
<td>87.0±5.9</td>
<td>78.0±8.4</td>
<td>91.7±5.0</td>
</tr>
<tr>
<td>iris</td>
<td>96.9±4.0</td>
<td>95.3±4.8</td>
<td>95.4±4.8</td>
<td>95.3±5.8</td>
</tr>
<tr>
<td>mfeat-factors</td>
<td>92.9±1.5</td>
<td>95.4±1.4</td>
<td>98.9±1.5</td>
<td>95.8±1.4</td>
</tr>
<tr>
<td>mfeat-fourier</td>
<td>77.4±3.0</td>
<td>80.3±2.6</td>
<td>75.2±5.0</td>
<td>81.2±2.7</td>
</tr>
<tr>
<td>ovarian</td>
<td>92.8±5.1</td>
<td>93.9±5.0</td>
<td>89.2±6.9</td>
<td>92.1±5.6</td>
</tr>
<tr>
<td>page-blocks</td>
<td>93.5±1.0</td>
<td>96.0±0.8</td>
<td>92.1±1.5</td>
<td>95.4±1.3</td>
</tr>
<tr>
<td>parkinsons</td>
<td>82.4±7.4</td>
<td>92.2±5.7</td>
<td>86.7±8.5</td>
<td>91.6±6.9</td>
</tr>
<tr>
<td>segment</td>
<td>89.9±2.1</td>
<td>91.2±1.7</td>
<td>71.2±7.5</td>
<td>90.7±1.7</td>
</tr>
<tr>
<td>sonar</td>
<td>83.0±8.1</td>
<td>85.1±6.9</td>
<td>72.6±11.0</td>
<td>84.0±7.8</td>
</tr>
<tr>
<td>spectrometer</td>
<td>77.7±5.2</td>
<td>81.3±4.9</td>
<td>81.6±5.1</td>
<td>81.7±5.4</td>
</tr>
<tr>
<td>vehicle</td>
<td>60.7±4.5</td>
<td>66.4±5.0</td>
<td>55.7±7.1</td>
<td>64.0±5.5</td>
</tr>
<tr>
<td>wdbc</td>
<td>95.4±3.1</td>
<td>96.6±2.3</td>
<td>95.1±3.2</td>
<td>98.8±3.8</td>
</tr>
<tr>
<td>wine</td>
<td>98.4±2.8</td>
<td>95.3±4.8</td>
<td>95.9±4.8</td>
<td>95.4±4.8</td>
</tr>
<tr>
<td>yeast</td>
<td>58.9±4.0</td>
<td>52.9±4.0</td>
<td>54.9±4.9</td>
<td>54.9±3.8</td>
</tr>
</tbody>
</table>

Average 80.94 82.56 78.29 82.60

UCI data are small, rendering statistical comparisons more difficult. This is why we decided to run a separate comparison on the five ImageNet data sets. They are multi-class classification problems and also exhibit a very high skew in the distribution of $N_k(x)$.

The three algorithms – $k$NN, hw-$k$NN and NHBBN were compared on five ImageNet subsets and the results are given in Table 3. The comparisons were performed for a fixed neighborhood of $k = 5$. The threshold $\theta$ in NHBBN was set to zero so that neither local or global estimate is used, only what information is available from the point label and neighborhood. We opted for such a configuration to show that, even though the local and global approach described in the paper provide greater flexibility in dealing with anti-hubs, they are by no means necessary for the basic idea to function.

The results from this third experimental setup differ greatly from the previous two, showing an even more convincing difference between NHBBN and $k$NN on high-dimensional data, as well as enabling us to differentiate between hw-$k$NN and NHBBN. On these data sets, we see that using class-specific hubs does, indeed, lead to better results. The improvement is usually modest, but this is due to the fact that both algorithms exploit the same phenomenon – hubness. However, NHBBN is to be preferred in cases when dealing with intrinsically high-dimensional multi-class data.

5. CONCLUSIONS AND FUTURE WORK

We presented a novel algorithm for probabilistic NN classification, naive hubness Bayesian $k$-NN (NHBBN). Hubness is a phenomenon inherent to high-dimensional data which has only recently been taken into serious consideration and has never before been used in a Bayesian framework. We have shown in this paper that taking point hubness into account may be beneficial to nearest-neighbor methods and that it should be more thoroughly investigated. The proposed algorithm differs in its conception greatly from the probabilistic $k$-nearest neighbor (PNN) and related methods, therefore representing an alternative approach that is open for further improvements.

Acknowledgments. This work was supported by bilateral project between Slovenia and Serbia “Correlating Images and Words: Enhancing Image Analysis Through Machine Learning and Semantic Technologies,” the Slovenian Research Agency, the Serbian Ministry of Education and Science through project no. O1174023, “Intelligent Techniques and Their Integration into Wide-Spectrum Decision Support,” and the ICT Programme of the EC PlanetData (ICT-257641).

Table 3: Comparison of NHBNN with $k$NN and hw-$k$NN on ImageNet data for $k = 5$. The symbols ¥/• denote statistically significant better/worse performance with respect to NHBBN for $p < 0.01$ and * denotes those cases when NHBBN is deemed better at significance level $0.01 \leq p < 0.05$.

<table>
<thead>
<tr>
<th>Data set</th>
<th>$k$NN</th>
<th>hw-$k$NN</th>
<th>NHBBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImgdNet-s3</td>
<td>72.9±2.7</td>
<td>80.8±2.3</td>
<td>81.8±2.3</td>
</tr>
<tr>
<td>ImgdNet-4</td>
<td>56.2±2.0</td>
<td>63.3±1.9</td>
<td>64.6±1.9</td>
</tr>
<tr>
<td>ImgdNet-s5</td>
<td>46.6±2.0</td>
<td>56.3±1.7</td>
<td>61.8±1.9</td>
</tr>
<tr>
<td>ImgdNet-s6</td>
<td>60.1±2.2</td>
<td>68.1±1.6</td>
<td>69.4±1.7</td>
</tr>
<tr>
<td>ImgdNet-s7</td>
<td>43.4±1.7</td>
<td>55.3±1.5</td>
<td>58.2±1.5</td>
</tr>
</tbody>
</table>

Average 55.66 64.72 67.16

6. REFERENCES