Semi-supervised Naive Hubness-Bayesian k-Nearest Neighbor for Gene Expression Data

Krisztian Buza

BioIntelligence Lab, Institute of Genomic Medicine and Rare Disorders
Semmelweis University, Budapest, Hungary
buza@biointelligence.hu
http://www.biointelligence.hu

Abstract. Classification of gene expression data is the common denominator of various biomedical recognition tasks. However, obtaining class labels for large training samples may be difficult or even impossible in many cases. Therefore, semi-supervised classification techniques are required as semi-supervised classifiers take advantage of the unlabeled data. Furthermore, gene expression data is high-dimensional which gives rise to the phenomena known under the umbrella of the *curse of dimensionality*, one of its recently explored aspects being the presence of hubs or *hubness* for short. Therefore, hubness-aware classifiers were developed recently, such as Hubness-Bayesian k-Nearest Neighbor (NHBNN). In this paper, we propose a semi-supervised extension of NHBNN and show in experiments on publicly available gene expression data that the proposed classifier outperforms all its examined competitors.

Keywords: Semi-supervised classification, gene expression data, high dimensionality

1 Introduction

Proteins play essential role in almost all biological processes at the cellular level. Genes are particular subsequences of the DNA that code for proteins. While each cell of the organism has the same DNA, activation levels of genes may vary in different tissues: informally speaking, the expression level of a gene means how frequently the corresponding DNA fragment is transcribed to RNA and translated to proteins. Various tissues are characterized by different gene expression patterns, furthermore, diseases such as cancer may be associated with characteristic gene expression patterns.

Classification of gene expression data may contribute to diagnosis of various diseases such as colon cancer, lymphoma, lung cancer and subtypes of breast cancer [7]. However, the classification task is challenging for several reasons. Usually, the expression levels of several thousands of genes are measured, therefore, the data is high-dimensional which gives rise to the phenomena known under the umbrella of the *curse of dimensionality*. While well-studied aspects of the curse are the sparsity and distance concentration, see e.g. [17], a recently

explored aspect of the curse is the presence of hubs [13], i.e., instances that are similar to surprisingly many other instances. A hub is said to be bad if its class label differs from the class labels of those instances that have this hub as one of their k-nearest neighbors. In the context of k-nearest neighbor classification, bad hubs were shown to be responsible for a surprisingly large portion of the total classification error. Therefore, hubness-aware classifiers were developed, such as the Naive Hubness Bayesian k-Nearest Neighbor, or NHBNN for short [21].

Hubness-aware classifiers were shown to work well with various types of noise [18] which is particularly relevant from the point of view of the current study as gene expression data is often noisy due to measurement uncertainty. Furthermore, it may be expensive (or in case of rare diseases even impossible) to collect *large* amount of labeled data, therefore, we have to account for the fact that only relatively few labeled instances are available which may not reflect the structure of the classes well enough. Therefore, besides learning from labeled data, the classification algorithm should be able to use unlabeled data too in order to discover the structure of the classes.

Therefore, in this paper we introduce a semi-supervised hubness-aware classifier. In particular, our approach is an extension of the aforementioned NHBNN. As we will show, straight forward incorporation of semi-supervised classification techniques with NHBNN leads to suboptimal results, therefore, we develop a hubness-aware inductive semi-supervised classification schema. To our best knowledge, this paper is the first that studies hubness-aware semi-supervised classification of gene expression data.

2 Background

Semi-supervised classification, often in a general data mining context, i.e., without special focus on the analysis of genetic data, has been studied intensively, see e.g. [5],[10] and the references therein for related works on semi-supervised classification.

Although the difficulties related to the analysis of high dimensional data are often referred to as the *curse of dimensionality* [3], and some results even suggest that the notion of distances between instances of a dataset becomes meaningless in high-dimensional spaces [17], algorithms developed recently under the umbrella of hubness-aware data mining try to address the curse of dimensionality, see e.g. [4],[11],[12],[14],[20],[22],[23], and [19] for a survey. Hubs were observed in gene expression data [8],[14] and hubness was brought into relation with the performance of the SUCCESS semi-supervised time-series classifier [9], however, none of the aforementioned works focused on hubness-aware classifiers in semi-supervised mode, i.e., when the classifier is allowed to learn both from labeled and unlabeled instances.

In order to ensure that our study is self-contained, next, we review the Naive Hubness Bayesian k-Nearest Neighbor (NHBNN) classifier [21] and the self-training semi-supervised learning technique. The presentation of NHBNN and self-training is based on [19] and [10] respectively.

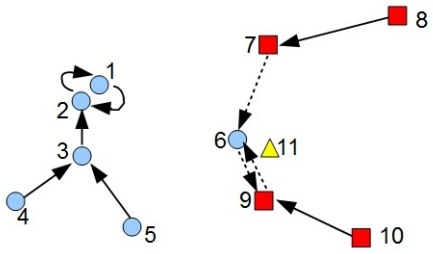


Fig. 1. Running example used to illustrate NHBNN. Labeled training instances belong to two classes, denoted by circles and rectangles. From each labeled training instance, a directed edge points to its first nearest neighbor among the labeled training instances. The triangle is an instance to be classified. For details, see the description of NHBNN.

2.1 NHBNN: Naive Hubness Bayesian *k*-Nearest Neighbor

We aim at classifying instance x^*, i.e., we want to determine its unknown class label y^*. We use $\mathcal{N}_k(x^*)$ to denote the set of k-nearest neighbors of x^*. For each class C, Naive Hubness Bayesian k-Nearest Neighbor (NHBNN) estimates $P(y^* = C|\mathcal{N}_k(x^*))$, i.e., the probability that x^* belongs to class C given its nearest neighbors. Subsequently, NHBNN selects the class with highest probability.

NHBNN follows a Bayesian approach to assess $P(y^* = C|\mathcal{N}_k(x^*))$. For each labeled training instance x, one can estimate the probability of the event that x appears as one of the k-nearest neighbors of any labeled training instance belonging to class C. This probability is denoted by $P(x \in \mathcal{N}_k|C)$. While calculating nearest neighbors, throughout this paper, an instance x is *never* treated as the nearest neighbor of itself, i.e., $x \notin \mathcal{N}_k(x)$.

Assuming conditional independence between the nearest neighbors given the class, $P(y^* = C|\mathcal{N}_k(x^*))$ can be assessed as follows:

$$P(y^* = C|\mathcal{N}_k(x^*)) \quad \propto \quad P(C) \prod_{x_i \in \mathcal{N}_k(x^*)} P(x_i \in \mathcal{N}_k|C). \tag{1}$$

where $P(C)$ denotes the prior probability of the event that an instance belongs to class C. From the labeled training data, $P(C)$ can be estimated as $P(C) \approx \frac{|\mathcal{D}_C^{lab}|}{|\mathcal{D}^{lab}|}$, where $|\mathcal{D}_C^{lab}|$ denotes the number of labeled training instances belonging to class C and $|\mathcal{D}^{lab}|$ is the total number of labeled training instances. The maximum likelihood estimate of $P(x_i \in \mathcal{N}_k|C)$ is the fraction

$$P(x_i \in \mathcal{N}_k|C) \approx \frac{N_{k,C}(x_i)}{|\mathcal{D}_C^{lab}|}, \tag{2}$$

where $N_{k,C}(x_i)$ denotes how many times x_i occurs as one of the k-nearest neighbors of labeled training instances belonging to class C.

Example. Fig. 1 shows a simple two-dimensional example, i.e., instances correspond to points of the plane. In this example, we use $k = 1$. In Fig. 1, a directed edge points from each labeled training instance to its first nearest neighbor among the labeled training instances. In other words: the nearest neighbor relationships shown in the Fig. 1 are calculated solely on the *labeled training data*.

Out of the 10 labeled training instances, 6 belong to the class of circles (C_1) and 4 belong to the class of rectangles (C_2). Thus: $|\mathcal{D}_{C_1}^{lab}| = 6$, $|\mathcal{D}_{C_2}^{lab}| = 4$, $P(C_1) = 0.6$ and $P(C_2) = 0.4$. Next, we calculate $N_{k,C}(x_i)$ for both classes and classify instance 11 using its first nearest neighbor, i.e., x_6. In particular, Eq. (2) leads to $P(x_6 \in \mathcal{N}_1|C_1) \approx \frac{N_{1,C_1}(x_6)}{|\mathcal{D}_{C_1}^{lab}|} = \frac{0}{6} = 0$ and $P(x_6 \in \mathcal{N}_1|C_2) \approx \frac{N_{1,C_2}(x_6)}{|\mathcal{D}_{C_2}^{lab}|} = \frac{2}{4} = 0.5$. According to Eq. (1) we calculate $P(y_{11} = C_1|\mathcal{N}_2(x_{11})) \propto 0.6 \times 0 = 0$ and $P(y_{11} = C_2|\mathcal{N}_2(x_{11})) \propto 0.4 \times 0.5 = 0.2$. As $P(y_{11} = C_2|\mathcal{N}_2(x_{11})) > P(y_{11} = C_1|\mathcal{N}_2(x_{11}))$, instance 11 will be classified as rectangle.

The previous example also illustrates that estimating $P(x_i \in \mathcal{N}_k|C)$ according to (2) may simply lead to zero probabilities. In order to avoid it, we can use a simple Laplace-estimate for $P(x_i \in \mathcal{N}_k|C)$ as follows:

$$P(x_i \in \mathcal{N}_k|C) \approx \frac{N_{k,C}(x_i) + m}{|\mathcal{D}_C^{lab}| + mq}, \tag{3}$$

where $m > 0$ and q denotes the number of classes. Informally, this estimate can be interpreted as follows: we consider m additional pseudo-instances from each class and we assume that x_i appears as one of the k-nearest neighbors of the pseudo-instances from class C. We use $m = 1$ in our experiments.

Even though k-occurrences are highly correlated, as shown in [19] and [21], NHBNN offers improvement over the basic kNN. This is in accordance with other results from the literature that state that Naive Bayes can deliver good results even in cases with high independence assumption violation [15].

2.2 Self-training

Self-training is one of the most commonly used semi-supervised algorithms. Self-training is a wrapper method around a supervised classifier, i.e., one may use self-training to enhance various classifiers. To apply self-training, for each instance x^* to be classified, besides its predicted class label, the classifier must be able to output a certainty score, i.e., an estimation of how likely the predicted class label is correct.

Self-training is an iterative process during which the set of labeled instances is grown until all the instances become labeled. Let L_t denote the set of labeled instances in the t-th iteration ($t \geq 0$) while U_t shall denote the set of unlabeled instances in the t-th iteration. L_0 denotes the instances that are labeled initially, i.e., the labeled training data, while U_0 denotes the set of initially unlabeled instances. In each iteration of self-training, the base classifier is trained on the labeled set L_t. Then, the base classifier is used to classify the unlabeled

```
SELF-TRAINING(L, U)
1  L_0 = L
2  U_0 = U
3  t = 0
4  repeat
5      M = SUPERVISED-LEARNING(L_t)
6      x_best = arg max_{x∈U_t} CERTAINTY(M, x)
7      ŷ = CLASSIFY(M, x_best)
8      L_{t+1} = L_t ∪ {(x_best, ŷ)}
9      U_{t+1} = U_t \ {x_best}
10     t = t + 1
11 until |U_t| == 0
12 return M
```

Fig. 2. Simple self-training algorithm.

instances. Finally, the instance with highest certainty score is selected. This instance, together with its predicted label $\hat{y}$, is added to the set of labeled instances, in order to construct L_{t+1} the set of labeled instance for the next iteration. The pseudocode of this algorithm is shown in Figure 2. In context of nearest neighbor classification, the algorithm is illustrated in Figure 3.

If an unlabeled instance is classified incorrectly and this instance is added to the training data of the subsequent iterations, this may cause a chain of classification errors. Therefore, as noted in [6], it may be worth to stop self-training after a moderate number of iterations and use the resulting model to label all the remaining unlabeled instances.

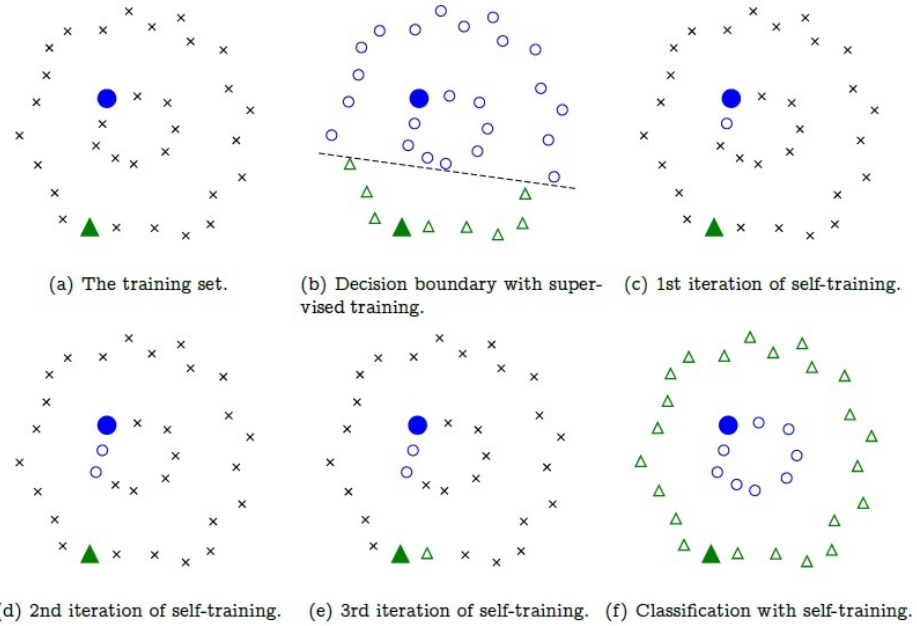
3 Certainty Estimation for NHBNN

In order to allow NHBNN to be used in self-training mode, we only need to define an appropriate certainty score. A straight-forward certainty score may be based on the probability estimates as follows:

$$certainty(x^*) = \frac{P(C') \prod\limits_{x_i \in \mathcal{N}_k(x^*)} P(x_i \in \mathcal{N}_k | C')}{\sum\limits_{C_j \in \mathcal{C}} \left(P(C_j) \prod\limits_{x_i \in \mathcal{N}_k(x^*)} P(x_i \in \mathcal{N}_k | C_j) \right)}. \tag{4}$$

where C' denotes the class with maximal estimated probability and $\mathcal{C}$ denotes the set of all the classes. In the example shown in Fig. 1, the above certainty estimate gives $0.2/(0 + 0.2) = 1$ when classifying instance 11.

However, this certainty estimate does not take into account that, usually, unlabeled instances appearing as nearest neighbors of many labeled instances can be classified more accurately as these instance are expected to be located "centrally" in the dataset, i.e., they appear in relatively dense regions of the



(a) The training set. (b) Decision boundary with supervised training. (c) 1st iteration of self-training.

(d) 2nd iteration of self-training. (e) 3rd iteration of self-training. (f) Classification with self-training.

Fig. 3. Self-training with nearest neighbor. There are two classes, circles and triangles. Bold symbols correspond to instances of the initially labeled training set L_0, while unlabeled instances are marked with crosses, see Subfigure (a). Subfigures (c) – (e) show the first three iterations of Self-Training. The final output of self-training is shown in Subfigure (f).

data, see e.g. [22]. Therefore, we propose to use the following hubness-aware certainty score:

$$hc(x^*) = \frac{N_k(x^*)P(C') \prod\limits_{x_i \in \mathcal{N}_k(x^*)} P(x_i \in \mathcal{N}_k | C')}{\sum\limits_{C_j \in \mathcal{C}} \left(P(C_j) \prod\limits_{x_i \in \mathcal{N}_k(x^*)} P(x_i \in \mathcal{N}_k | C_j) \right)}, \tag{5}$$

where $N_k(x^*)$ denotes how many times instance x^* appears as nearest neighbors of other instances when considering the labeled training data $\mathcal{D}^{lab}$ together with the unlabeled instance x^*, i.e., $\mathcal{D}^{lab} \cup \{x^*\}$. Please note that in order to calculate $hc(x^*)$, we do not take other unlabeled instances into account.

In the example shown in Fig. 1, the above certainty estimate gives $(2 \times 0.2)/(0+0.2) = 2$ when classifying instance 11, as instance 11 appears as nearest neighbor of instance 6 and instance 9 when considering all the eleven instances for the computation of the nearest neighbor relationships (we assume that the distance between instance 11 and instance 9 is lower that the distance between instance 9 and instance 6, therefore, instance 11 will be the nearest neighbor of instance 9 when considering all the instances).

4 Experimental Evaluation

Datasets. We used publicly available gene expression data of breast cancer tissues [16], colon cancer tissues [1], and lung cancer tissues [2]. In these datasets, the expression levels of 7650, 6500 and 12,600 genes have been measured for 95, 62 and 203 patients in the breast cancer, colon cancer and lung cancer datasets respectively. The breast and colon cancer datasets had two classes, while the lung cancer dataset had five classes. In all the cases, classes correspond to subtypes of the disease or healthy tissues, see [7] for details. Out of the five classes of the lung cancer dataset, we ignored one because extraordinarily few instances (in particular, only six instances) belonged to that class.

Experimental protocol. In order to simulate scenarios in which the available training data is not fully representative, we considered five randomly selected instances per class as labeled training data. This results in balanced distribution of classes in the labeled training data whereas the entire datasets were class-imbalanced [7]. We repeated all the experiments 100-times with 100 different initial random selection of the labeled training instances. We measured the performance of the classifiers in terms of classification accuracy, i.e., the fraction of correctly classified "unlabeled instances". Note that the true class labels of the "unlabeled instances" were given in the datasets, however, these true class labels were used for evaluation purposes only, i.e., the labels of the "unlabeled instances" were unknown to the classifier. We report the average and standard deviation of the accuracies achieved in the aforementioned 100 runs. Additionally, we used t-test at significance level of 0.01 to judge if the differences between our approach and the baselines are statistically significant.

Compared Methods. We compared the following approaches:

- NHBNN-HS, i.e., NHBNN in self-training mode with the proposed hubness-aware certainty score according to Formula (5),
- NHBNN-Simple, i.e., NHBNN in self-training mode with the straight-forward certainty score according to Formula (4),
- k-NN in self-training mode with the proposed hubness-aware certainty score according to Formula (5),
- NHBNN-SV, i.e., supervised NHBNN that uses only the labeled training instances but does not learn from the unlabeled data.

In accordance with [20], by default, we used $k = 5$ for all the aforementioned variants of NHBNN and k-NN. Note, however, that we performed experiments with other k values as well and we observed similar trends. In order to avoid the propagation of errors, in accordance with [6], in case of semi-supervised classifiers, we performed 20 iterations of self-training, i.e., 20 instances were labeled and added to the training set iteratively and then the model resulting after the 20th iteration was used to label all the remaining unlabeled instances. We performed experiments with other number of self-training iterations as well and we observed similar trends regarding the order of the semi-supervised approaches.

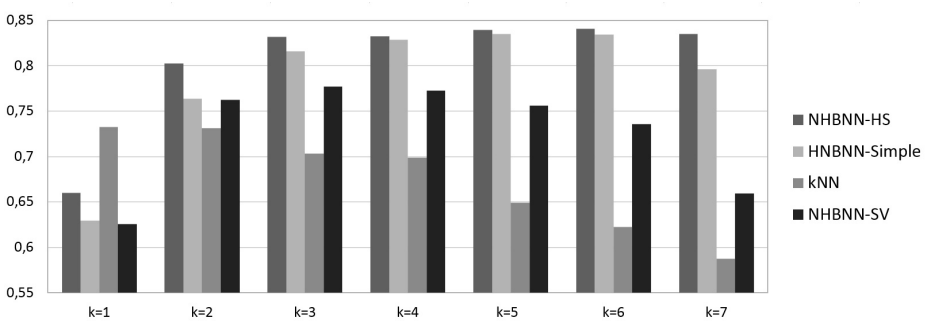


Fig. 4. Accuracy of our approach, NHBNN-HS, and its competitors for various k values on the BreastCancer dataset.

Table 1. Accuracy ± standard deviation of our approach, NHBNN-HS and the baselines averaged over 100 runs. Bold font denotes the best approach for each dataset. The symbol •/◦ denotes if the difference between NHBNN-HS and its competitor is statistically significant (•) or not (◦) according to t-test at significance level of 0.01.

	BreastCancer	ColonCancer	LungCancer
NHBNN-HS	**0.840 ± 0.044**	**0.794 ± 0.073**	**0.784 ± 0.152**
NHBNN-Simple	0.835 ± 0.049 ◦	0.790 ± 0.082 ◦	0.679 ± 0.114 •
k-NN	0.649 ± 0.155 •	0.650 ± 0.162 •	0.674 ± 0.329 •
NHBNN-SV	0.756 ± 0.103 •	0.637 ± 0.139 •	0.617 ± 0.125 •

Results. Our results are summarized in Table 1. The results show that our approach, NHBNN-HS, consistently outperforms the baselines on all the three datasets. As we can see, both the choice of the algorithm and the certainty score matters: both NHBNN in self-training mode with the straight forward certainty score and k-NN with the hubness-aware certainty score achieve suboptimal accuracy compared with our approach NHBNN-HS. Furthermore, as we expected, semi-supervised classification outperforms supervised classification as it can be seen from the comparison against NHBNN-SV. Fig. 4 shows that NHBNN-HS systematically outperforms its competitors for various k values, except for $k = 1$.

Additionally, we tried support vector machines from the Weka software package with polynomial and RBF kernels with various settings of the complexity constant and the exponent of the polynomial kernel. According to our observations, self-training was not able to substantially improve the performance of SVMs overall: SVMs without self-training performed as well as (or sometimes even better than) SVMs with self-training. More importantly, NHBNN-HS was competitive to SVMs too: for example on the Breast Cancer and Colon Cancer datasets, best performing SVMs achieved classification accuracy of 0.781 and 0.705 respectively. We also note that the model built by NHBNN is more interpretable to human experts than the model built by an SVM.

5 Conclusions and Outlook

In many applications, obtaining reliable class labels for large training samples may be difficult or even impossible. Therefore, semi-supervised classification techniques are required as they are able to take advantage of unlabeled data. Some of the most prominent recent methods developed for the classification of high-dimensional data follow the paradigm of hubness-aware data mining. However, hubness-aware classifiers have not been used for semi-supervised classification tasks previously. Therefore, in this paper, we introduced a semi-supervised hubness-aware classifier and we showed that it outperforms all the examined relevant baselines on the classification of gene expression data. While classification of gene expression data is highly relevant to due to its biomedical applications, we expect that hubness-aware semi-supervised classifiers will also be utilized in various other classification tasks in the future.

Acknowledgment

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

References

1. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences 96(12), 6745–6750 (1999)
2. Bhattacharjee, A., Richards, W.G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., et al.: Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. Proceedings of the National Academy of Sciences 98(24), 13790–13795 (2001)
3. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
4. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: INSIGHT: Efficient and Effective Instance Selection for Time-Series Classification. In: 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence (LNCS/LNAI), vol. 6635, pp. 149–160. Springer (2011)
5. Chapelle, O., Schölkopf, B., Zien, A., et al.: Semi-supervised learning. MIT press Cambridge (2006)
6. Guillaumin, M., Verbeek, J., Schmid, C.: Multimodal semi-supervised learning for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 902–909. IEEE (2010)
7. Lin, W.J., Chen, J.J.: Class-imbalanced classifiers for high-dimensional data. Briefings in bioinformatics 14(1), 13–26 (2013)

8. Marussy, K.: The curse of intrinsic dimensionality in genome expression classification. In: Students' Scientific Conference, Budapest University of Technology and Economics (2014)
9. Marussy, K., Buza, K.: Hubness-based indicators for semi-supervised time-series clas-sification. In: Proc. 8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications. pp. 97–108 (2013)
10. Marussy, K., Buza, K.: Success: A new approach for semi-supervised classification of time-series. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L., Zurada, J. (eds.) Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 7894, pp. 437–447. Springer Berlin Heidelberg (2013)
11. Radovanović, M., Nanopoulos, A., Ivanović, M.: Nearest Neighbors in High-Dimensional Data: The Emergence and Influence of Hubs. In: Proceedings of the 26rd International Conference on Machine Learning (ICML). pp. 865–872. ACM (2009)
12. Radovanović, M., Nanopoulos, A., Ivanović, M.: Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. The Journal of Machine Learning Research (JMLR) 11, 2487–2531 (2010)
13. Radovanović, M., Nanopoulos, A., Ivanović, M.: Time-Series Classification in Many Intrinsic Dimensions. In: Proceedings of the 10th SIAM International Conference on Data Mining (SDM). pp. 677–688 (2010)
14. Radovanović, M.: Representations and Metrics in High-Dimensional Data Mining. Izdavačka knjižarnica Zorana Stojanovića, Novi Sad, Serbia (2011)
15. Rish, I.: An empirical study of the naive Bayes classifier. In: Proc. IJCAI Workshop on Empirical Methods in Artificial Intelligence (2001)
16. Sotiriou, C., Neo, S.Y., McShane, L.M., Korn, E.L., Long, P.M., Jazaeri, A., Martiat, P., Fox, S.B., Harris, A.L., Liu, E.T.: Breast cancer classification and prognosis based on gene expression profiles from a population-based study. Proceedings of the National Academy of Sciences 100(18), 10393–10398 (2003)
17. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley (2005)
18. Tomašev, N., Buza, K.: Hubness-aware knn classification of high-dimensional data in presence of label noise. Neurocomputing (2015)
19. Tomašev, N., Buza, K., Marussy, K., Kis, P.B.: Hubness-aware classification, instance selection and feature construction: Survey and extensions to time-series. In: Feature Selection for Data and Pattern Recognition, pp. 231–262. Springer (2015)
20. Tomašev, N., Mladenić, D.: Nearest neighbor voting in high dimensional data: Learning from past occurrences. Computer Science and Information Systems 9, 691–712 (2012)
21. Tomašev, N., Radovanović, M., Mladenić, D., Ivanovicć, M.: A probabilistic approach to nearest neighbor classification: Naive hubness Bayesian k-nearest neighbor. In: Proceeding of the CIKM conference (2011)
22. Tomašev, N., Radovanović, M., Mladenić, D., Ivanovic, M.: The role of hubness in clustering high-dimensional data. In: PAKDD (1)'11. pp. 183–195 (2011)
23. 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)