Dynamic Selection of Classifiers Applied to High-Dimensional Small-Instance Data Sets: Problems and Challenges

Alexandre Maciel-Guerra(✉), Grazziela P. Figueredo, and Jamie Twycross

School of Computer Science, University of Nottingham, Computer Science Building, Wollaton Road, Nottingham NG8 1BB, UK
{Alexandre.MacielGuerra,Grazziela.Figueredo, Jamie.Twycross}@nottingham.ac.uk

Abstract. Dynamic selection (DS) of classifiers have been explored by researchers due to their overall ability to obtain higher accuracy on low-sample data sets when compared majority voting. Little literature, however, has employed DS to high-dimensional data sets with substantially more features than samples. Since, several studies have reported the benefits of applying feature selection methods to high-dimensional data sets, raised the following open research questions: 1. How DS methods perform for such data sets? 2. Do they perform better than majority voting? and 3. Does feature selection as a pre-processing step improve their performance? The performance of 21 DS methods was statistically compared against the performance of majority voting on 10 high-dimensional data sets and with a filter feature selection method. We found that majority voting is among the best ranked classifiers and none of the DS methods perform statistically better than it with and without feature selection. Moreover, we demonstrated that feature selection does improve the performance of DS methods.

Keywords: Ensemble learning · Dynamic integration of classifiers · Dynamic selection · Machine learning · Majority voting · High dimensional data sets

1 Introduction

Over the past decades, multiple classifier systems (MCS) became a very active area in pattern recognition. One of the most promising approaches involves dynamic selection (DS), in which different classifiers are selected for each unseen sample. Several authors have recently shown that dynamic selection (DS) methods obtain high performances in terms of accuracy on low dimensional datasets [7,9]. Nevertheless, many authors observed that DS techniques are still far from the upper bound performance of the oracle, which always predicts the correct label if at least one classifier in the ensemble predicts the correct

G. Nicosia et al. (Eds.): LOD 2020, LNCS 12565, pp. 637–649, 2020.
https://doi.org/10.1007/978-3-030-64583-0_56

label. With this in mind, some authors have reported and proposed solutions to improve the quality of the region of competence in low-dimensional datasets to increase their performance [7,20]. Over the past decade, DS techniques have been evaluated on low dimensional datasets and, to the best of our knowledge, there is no comprehensive work in the literature that verifies the performance of the state-of-art DS methods when dealing with high-dimensional small-instances datasets. Maciel-Guerra *et al.* (2019) [18] studied the performance of DS methods over a single high-dimensional protein microarray data set.

High-dimensional data sets with a small number of samples are typical in some domains, such as biology, medicine, bioinformatics and neuroimaging. Often in these areas data do not exist in abundance or is expensive to acquire [4]. In high dimensional data sets, many dimensions are irrelevant and/or redundant which can directly impact the quality of the regions of competence [16,21]. Feature selection methods have been employed to remove irrelevant features and filter methods are usually chosen due to their low computational cost [16,21].

The focus of this paper is, therefore, to evaluate how DS methods perform on high-dimensional small-instance data sets and compare it to majority voting which is the simplest MCS method. Despite the large number of papers published in DS, there is no comprehensive study available verifying the use of this methods on this specific type of data set. Following the recent study of Maciel-Guerra *et al.* (2019) [18] that studied the performance of DS methods over a single small instance high dimensional data set, we have three research questions, namely:

1. How DS methods perform in terms of accuracy?
2. Do they perform statistically better than majority voting?
3. Does feature selection as a pre-processing step improve their performance?

To answer these questions, 10 real-world benchmark data sets with a high number of features and a low number of samples are selected. Four data sets are text based while six are biomedical data sets relating to different types of cancer (lung, prostate, leukemia, colon, glioma and ovarian). Twenty-one DS methods available in the literature are compared against majority voting. The Iman-Davenport extension of the Friedman test [14] is used to statistically verify the performance of the classifiers over all data sets and the Bonferroni-Dunn test [10] is used as a post-hoc test to evaluate if any of the methods outperform statistically majority voting.

This paper is organised as follows. Section 2 provides background on the main topics of this paper. Section 3 introduces the experiments design with the data sets and statistical methods used. A discussion between the performance of DS methods and other MCS methods is conducted in Sect. 4. The conclusion and future research are given in Sect. 5.

2 Background

The quantity of data collected from multiple sources have increased greatly in the past decade, particularly in medicine and life sciences, which brings challenges

and opportunities. Heterogeneity, scalability, computational time and complexity are some of the challenges that impede progress to extract meaningful information from data [2,3]. High-dimensional data sets with a small number of samples are typical in some domains, such as biology, medicine, bioinformatics and neuroimaging [12]. We believe that approaches such as DS can improve the classification and increase knowledge discovery in high-dimensional data.

2.1 Dynamic Selection

An important task regarding classification ensembles is the decision as to which classifiers are required to be included to achieve high prediction accuracy. Static Selection (SS), Dynamic Classifier Selection (DCS) and Dynamic Ensemble Selection (DES) are the techniques commonly employed to determine the set of classifiers within the ensemble. SS works by selecting a group of classifiers for all new samples, while DCS and DES select a single or a group of classifiers for each new sample, respectively. Recently, DS methods have been preferred over static methods due to their ability to create different classifier configurations, i.e. different groups of classifiers are experts in different local regions of the feature space. As for many cases, different samples are associated with different classification difficulties and the ability to choose a group of classifiers can possibly overcome static selection methods limitations [6,9,15].

Table 1. DS methods information

Name	Selection criteria	DS Method	Region of Competence	Year
Classifier Rank (CR)	Ranking	DCS	k-NN	1993
Modified Classifier Rank (MCR)	Ranking	DCS	k-NN	1997
Overall Local Accuracy (OLA)	Accuracy	DCS	k-NN	1997
Local Class Accuracy (LCA)	Accuracy	DCS	k-NN	1997
A Priori	Probabilistic	DCS	k-NN	1999
A Posteriori	Probabilistic	DCS	k-NN	1999
Multiple Classifier Behaviour (MCB)	Behaviour	DCS	k-NN	2002
Modified Local Accuracy (MLA)	Accuracy	DCS	k-NN	2002
DES - kMeans	Accuracy & Diversity	DES	k-Means	2006
DES - K-Nearest Neighbour (DES-kNN)	Accuracy & Diveristy	DES	k-NN	2006

(*continued*)

Table 1. (*continued*)

Name	Selection criteria	DS Method	Region of Competence	Year
k-Nearest ORAcles Elimimante (KNORA-E)	Oracle	DES	k-NN	2008
k-Nearest ORAcles Union (KNORA-U)	Oracle	DES	k-NN	2008
DES - Exponential (DES-EXP)	Probabilistic	DES	All samples	2009
DES - Randomised Reference Classifier (DES-RRC)	Probabilistic	DES	All samples	2011
DES - Minimal Difference (DES-MD)	Probabilistic	DES	All samples	2011
DES - Kullback-Leibler Divergence (DES-KL)	Probabilistic	DES	All samples	2012
DES - Performance (DES-P)	Probabilistic	DES	All samples	2012
k-Nearest Output Profiles Elimiante (KNOP-E)	Behaviour	DES	k-NN	2013
k-Nearest Output Profiles Union (KNOP-U)	Behaviour	DES	k-NN	2013
Meta-learning - DES (Meta-DES)	Meta-learning	DES	k-NN	2015
Dynamic Selection on Complexity (DSOC)	Accuracy & Complexity	DCS	k-NN	2016

For DS methods to achieve optimum recognition rates they need to select the most competent classifiers for any given test sample, which can be done by measuring different selection criteria depending on the technique used (accuracy, ranking, behaviour, diversity, probabilistic, complexity and meta-learning). More information about each one of this different criteria can be found on the recent review by [6,9]. A local region of the feature space surrounding the test sample (Region of Competence) is used to estimate the competence of each classifier according to any selection criteria. The majority of DS techniques relies on k-Nearest Neighbours (k-NN) algorithms (Table 1) and the quality of the neighbourhood can have a huge impact on the performance of DS methods [6,7,9].

Table 1 shows the different DS methods found in the literature which were presented in the most recent review by Cruz *et al.* [9]. More information about each one can be found on their respective reference or on the recent reviews done by [6,9]. These methods were chosen due to their differences in the selection criteria and because they present the most important breakthroughs in the area over the past three decades.

2.2 High-Dimensional Data

Financial, risk management, computational biology, health studies are some of the areas where high-dimensional data sets can be produced. However, in some of these areas, such as biology and medicine, it might not be feasible to have thousands or millions of samples due to the nature of the disease or the access to samples [29]. DNA microarray is one example of these types of data sets where data collected from tissue and cell samples are used to measure the levels of gene expression. The number of genes is usually far higher than the number of patients in cancer research for instance [4].

Data sets with a high number of features usually poses challenges that are commonly referred as the "curse of dimensionality". One of the main aspects of this curse is *distance concentration*, which can directly affect machine learning application, specially the ones that deal with distance metrics such as k-NN. Concentration of distance refers to the tendency of distance to all points to become almost equal in high-dimensional spaces [1,24,25].

For these reasons, for any classifier to be successful (have a high accuracy level), it is usually necessary to have sufficient data to cover the feature space during training, so it can have as much information possible on the feature space to find the correct learning function to predict the output associated with new inputs [4,29]. If this is not the case, researchers frequently apply different feature selection techniques to remove unwanted (redundant, irrelevant, noisy) features and, consequently, improve the performance of classifiers [4].

2.3 Feature Selection

In two recent reviews Bólon-Canedo *et al.* [4,5] reported the benefits of applying feature selection methods to high-dimensional data sets, and highlighted the fact that feature selection methods are considered a *de facto* standard in machine learning and data analysis since its introduction.

Feature selection maps $\mathbf{x} \in \mathbb{R}^d \rightarrow \mathbf{y} \in \mathbb{R}^p$ where $p < d$. The reduction criteria usually either maintains or improves the accuracy of classifiers trained with this data, while simplifying the complexity of the model [4,5,13]. Feature selection methods can be classified into three main groups:

1. Filter methods: perform the feature selection as a pre-processing step. It is independent from the learning stage and relies only on the attributes of the data [5]. Despite the lower time consumption, one of the main disadvantages of filters is the fact that they do not interact with the learning method; which usually leads to worse performance when compared to other methods [4].
2. Wrapper methods: use a learning algorithm as a subroutine, measuring the usefulness of each subset of features with the prediction performance of the learning algorithm over a validation set [5]. Although usually wrapper methods show a better performance when compared with filter methods, they have a much higher computation cost which increases as the number of features in the data increases [4].

3. Embedded methods: the feature selection process is built into the learning method, so it can use the core of the learning method to rank the features by their importance [4,5].

Tsymbal *et al.* [28] and Pechenizkiy *et al.* [22] demonstrated the benefits of integrating feature selection methods to the DS framework. However, the data sets used had a sample-feature ratio higher than one. In addition, the filtering method proposed by Almeida (2014) [20] achieved higher performances in terms of accuracy only on datasets with less than 20 features and 3 classes. These authors were able to show that feature selection methods incorporated to the DS framework can improve the performance of DS methods on some data sets and overcome some of the problems related to high-dimensional data sets. In addition, Maciel-Guerra *et al.* (2019) studied a protein microarray data set to evaluate the performance of DS methods. The authors demonstrated that for this single data set, DS methods do not outperform majority voting.

3 Experiments

3.1 Data Sets

The experiments are conducted on 10 real-world high-dimensional data sets (Table 2. Nine of those data sets are obtained from the *Feature Selection data sets* (Arizona State University [17]) and another from the UCI machine learning repository [11]. We considered only data sets with small sample sizes

Table 2. Data sets attributes

Data set	Sample (s)	Features (f)	Ratio (s/f)	No. of classes	Distribution	Type	Source
Leukemia/ALLAML	72	7129	0.0101	2	65.3 - 34.7%	Microarary	[17]
Arcene	200	10000	0.02	2	56 - 44%	Mass spectrometry	[17]
Basehock	1993	4862	0.4099	2	49.9 - 50.1%	Text	[17]
Colon	62	2000	0.031	2	64.5 - 35.5%	Microarary	[17]
Dexter	600	20000	0.03	2	50 - 50%	Text	[11]
Gli85	85	22283	0.0038	2	30.6 - 69.4%	Microarary	[17]
Pcmac	1943	3289	0.5907	2	50.5 - 49.5%	Text	[17]
Prostate	102	5966	0.0171	2	49 - 51%	Microarary	[17]
Relathe	1427	4322	0.3302	2	54.6 - 45.4%	Text	[17]
Smk-Can	187	19993	0.0094	2	48.1 - 51.9%	Microarary	[17]

3.2 Experimental Design

All techniques are implemented using the *scikit-learn* [23] and the *DESlib* [8] libraries in Python. The experiments are conducted using 30 replicates. For each replicate, the data sets are randomly divided in 50% for the training set, 25% for the Region of Competence set and 25% for the test set as suggested by Cruz *et al.* [9]. These divisions are performed preserving the proportion of samples for

each class by using the stratified k-fold cross validation function in the *scikit-learn* [23] library.

The pool of classifiers is composed of 11 decision trees, as suggested by Woloszynski *et al.* [31], with pruning level set to 10. The pool is generated using the bagging technique, similarly to the methodology followed by Woloszynski in [30,31]. An odd number of classifiers is chosen to overcome decision ties. These classifiers are used due to their instability when trained with different sets of data, i.e., small differences on the training set can create different trees [31]. Following the recent survey on DS techniques [9], the size of the Region of Competence K is set to 7 neighbours for all the techniques based on k-NN. Moreover, as suggested by Cruz and Soares in [9,26,27], 30% of the base classifiers are selected using accuracy and diversity for the techniques DES-kNN and DES-kMeans. In addition, the number of clusters of DES-kMeans is set to 5.

3.3 Comparison of Techniques

The Friedman test F_F with Iman-Davenport correction [14] is employed for statistical comparison of multiple classifier system techniques as suggested by Cruz and Demsar in [9,10]. F_F ranks the algorithms for each data set separately, i.e. the best algorithm gets ranking 1, the second best ranking 2, and so on. In case of ties, average ranks are assigned. F_F is distributed according to the $\mathcal{X}_F^2$ distribution and the Iman-Davenport extension (Eq. 1) is distributed according to the F-distribution (Eq. 2) with $k-1$ and $(N-1)\times(k-1)$ degrees of freedom. The null-hypothesis states that all algorithms are equivalent and so their average ranks should be equal.

$$\mathcal{X}_F^2 = \frac{12N}{k(k+1)}\Big[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\Big] \tag{1}$$

$$F_F = \frac{(N-1)\mathcal{X}_F^2}{N(k-1)-\mathcal{X}_F^2} \tag{2}$$

where R_j is the average rank of the j-th classifier, k is the number of classifiers and N is the number of data sets.

The rank of each method is calculated using the weighted ranking approach proposed by Yu in [32], which considers the differences among the average performance metric values between classifiers for each data set [32]. The best performing algorithm is the one with the lowest average rank. Next, as suggested by [10], to compare all classifiers against a control, we use the Bonferroni-Dunn test with the following test equation to compare two classifiers:

$$z = (R_i - R_j)\Big/\sqrt{\frac{k(k+1)}{6N}} \tag{3}$$

where R_i is the rank of i-th classifier, k is the number of classifiers and N is the number of data sets. The z value is than used to find the corresponding p-value from the two-tailed normal distribution table, which is subsequently compared

to an appropriate significance level α. The Bonferroni-Dunn test subsequently divides α by $k-1$ to control the family-wise error rate. The level of $\alpha = 0.05$ is considered as significance level. Hence, the level of $p < 0.0022$ was considered as statistically significant.

4 Results and Discussion

Accuracy is calculated for all experiments and averaged over the 10 replications. In addition, the rank of all classifiers for each data set is calculated according to the weighted ranking approach proposed by Yu in [32] and averaged to measure the Z-score to find its respective p-value. With 22 classifiers and 10 data sets, the Friedman test is distributed according to the F distribution with $22-1 = 21$ and $(10-1) \times (22-1) = 189$ degrees of freedom. The critical value of F (21,189) for $\alpha = 0.0001$ is 2.8165.

Table 1 shows the 21 DS methods used. Nine are dynamic classifier selection methods (the first eight and the last one based on the date the paper was published) which select a single classifier from the pool of classifiers. The remaining methods are dynamic ensemble selection techniques, which select an ensemble of classifiers from the initial pool. These techniques are selected because they incorporate all the major breakthroughs in the area of dynamic selection on the past three decades as highlighted by Cruz *et al.* [9], i.e, the papers which proposed different selection techniques to be incorporated into the DS framework. We compare the average rank obtained by the majority voting method (static selection) against the 21 DS methods.

The first experiment assesses classifier performance without feature selection. Table 3 shows the average accuracy and standard deviation for each data set, the average rank, Z-score and p-value results for all the classifiers that had a rank lower than majority voting without feature selection. The F_F statistic is 4.7468, so the null-hypothesis can be rejected with 99.99% confidence. To compare all classifiers against a control, majority voting, the Bonferroni-Dunn test is used to measure the Z-score for each classifier. Even though there are 3 classifiers (KNORA-U, KNOP-U and DES-P) with a better rank than majority voting, none of them is statistically different from majority voting.

The second experiment (Table 4) employs the univariate feature selection method. Instead of selecting a specific number of features, a p-value is computed using the ANOVA F-test and a family wise error rate is used to select them with a 95% confidence level. For high-dimensional data sets it is necessary to compute a feature selection method to reduce the complexity of the problem. Nonetheless, this is not an easy task due to the "curse of dimensionality". Therefore, the feature selection method chosen must be fast to compute because of the large number of features. This is the reasoning for choosing a filter method as the feature selection approach. For this experiment, the F_F statistical value was 5.6171. Aposteriori, KNORA-U and KNOP-U had a lower rank when compared with majority voting, nevertheless, these ranks are not statistically different.

The aforementioned results show that for all the data sets we tested with more features than samples dynamic selection methods are statistically equivalent to

Table 3. Average accuracy, ranking, z-score and respective p-value for the classifiers that had a lower rank when compared with majority without feature selection and the oracle results

	knop u	knora u	des p	majority voting	oracle
Allaml	0.9333 ± 0.0563	0.9296 ± 0.0573	0.9296 ± 0.0573	0.9278 ± 0.0576	1 ± 0
Arcene	0.7067 ± 0.0646	0.7173 ± 0.0667	0.704 ± 0.0576	0.71 ± 0.0586	0.996 ± 0.0095
Basehock	0.9045 ± 0.0138	0.8922 ± 0.0132	0.8923 ± 0.0132	0.8917 ± 0.0128	0.9625 ± 0.013
Colon	0.7542 ± 0.0926	0.7604 ± 0.0983	0.7417 ± 0.1067	0.7438 ± 0.0932	0.9896 ± 0.0233
Dexter	0.8789 ± 0.0357	0.8722 ± 0.0366	0.8731 ± 0.0368	0.8729 ± 0.0367	0.992 ± 0.0111
Gli	0.8136 ± 0.0678	0.8212 ± 0.0713	0.8273 ± 0.0737	0.8152 ± 0.0713	0.9924 ± 0.0169
Pcmac	0.8648 ± 0.0162	0.8582 ± 0.0158	0.8576 ± 0.016	0.8575 ± 0.016	0.9421 ± 0.0261
Prostate	0.8782 ± 0.0724	0.8833 ± 0.064	0.8821 ± 0.062	0.8821 ± 0.0688	0.9936 ± 0.0143
Relathe	0.825 ± 0.0205	0.8085 ± 0.0226	0.8121 ± 0.0228	0.8076 ± 0.0217	0.9525 ± 0.0193
Smkcan	0.6298 ± 0.0637	0.6255 ± 0.0551	0.6262 ± 0.0661	0.6135 ± 0.053	0.9986 ± 0.0053
Rank	5,60	6,49	6,82	7,47	–
z score	0,6413	0,3374	0,2220	0	–
p-value	0,5213	0,7358	0,8243	1	–

Table 4. Average accuracy, ranking, z-score and respective p-value for the classifiers that had a lower rank when compared with majority with univariate feature selection based on the ANOVA-F test with Family-wise Error rate and the oracle results

	aposteriori	knop u	knora u	majority voting	oracle
Allaml	0.9111 ± 0.0682	0.9315 ± 0.0652	0.9333 ± 0.0664	0.9333 ± 0.0664	1 ± 0
Arcene	0.6907 ± 0.0593	0.7727 ± 0.065	0.7693 ± 0.0655	0.766 ± 0.0687	0.9913 ± 0.0123
Basehock	0.9048 ± 0.0144	0.9063 ± 0.0128	0.895 ± 0.0137	0.8929 ± 0.0136	0.9633 ± 0.0125
Colon	0.8625 ± 0.0987	0.7896 ± 0.0876	0.7896 ± 0.0876	0.7938 ± 0.0886	0.9688 ± 0.0419
Dexter	0.8949 ± 0.0262	0.9009 ± 0.0174	0.8976 ± 0.0166	0.8962 ± 0.0204	0.99 ± 0.0089
Gli	0.8864 ± 0.0721	0.8515 ± 0.0778	0.8545 ± 0.0866	0.8606 ± 0.0813	0.9924 ± 0.0169
Pcmac	0.8737 ± 0.0138	0.8684 ± 0.0186	0.8666 ± 0.0184	0.8641 ± 0.0153	0.9198 ± 0.0368
Prostate	0.8949 ± 0.0432	0.8936 ± 0.0656	0.8885 ± 0.0669	0.8897 ± 0.0657	0.9885 ± 0.0202
Relathe	0.8313 ± 0.0166	0.8274 ± 0.0204	0.8183 ± 0.0209	0.8139 ± 0.0204	0.9198 ± 0.0374
Smkcan	0.7553 ± 0.0568	0.7333 ± 0.0688	0.7369 ± 0.0685	0.7355 ± 0.074	0.9872 ± 0.0224
Rank	5,89	5,9	7,3	7,81	–
z score	0,6606	0,6585	0,1756	0	–
p-value	0,5089	0,5102	0,8606	1	–

a simple method such as majority voting. This result differs from the recent reviews in the literature [6,9] that showcased the higher performance of DS methods over majority voting on low-dimensions data sets. Nonetheless, the filter feature selection method chosen was able to reduce drastically the number of features (Table 5) and increase the performance of most classifiers over all data sets.

The type of data sets used in our work might explain the reasons of our findings. The data sets investigated have a far larger number of features compared to the number of instances. This situation poses a problem for machine learning

techniques for some reasons: (1) wrapper methods require a reasonable computational time to select a subset of features in a large search space, hence the selection of a filter technique to reduce the dimensionality; (2) it is likely that there is insufficient data to cover the entire feature space, because the reduction of dimensionality increased the performance of 97% of 22 classifiers over 10 data sets; (3) Euclidean distance does not work on high-dimensional spaces since points are equally distance from one another.

Table 5. Number of features after applying the filter univariate feature selection based on the ANOVA-F test with Family-wise Error rate

Data sets	Features before filter	Features after filter	Reduction
Allaml	7129	130	98,18%
Arcene	10000	937	90,63%
Basehock	4862	286	94,12%
Colon	2000	16	99,20%
Dexter	20000	36	99,82%
Gli	22283	265	98,81%
Pcmac	3289	59	98,21%
Prostate	5966	198	96,68%
Relathe	4322	126	97,08%
Smkcan	19993	63	99,68%

We focused on demonstrating that DS methods did not have high performance levels on data sets with high-dimensionality and low sample sizes when compared with a simple MCS method such as majority voting. The results suggest that the Euclidean distance used by most of the methods is not working and therefore an alternative must be proposed for these types of data set. Moreover, feature selection could be incorporate to the DS framework to select the most important features for each sample. Although the results suggest an increase in performance, they are still far from the oracle. This indicates that the features selected might still not be the best subset.

In addition, due to the properties of high-dimensional spaces, clusters can be masked [21]; and a phenomena called *local feature relevance* happens, i.e., different subsets of features are relevant for different clusters [16]. This might explain the reason why the accuracy after feature selection was still further apart from the oracle and further investigations must be conducted to overcome this issue and improve even further the results.

5 Conclusions

In this paper, we investigated how DS methods perform on high dimensional data sets, more specifically those with a sample-feature ratio below one. We compared

21 DS methods against the majority voting method. Our approach used the Friedman test with the Iman-Davenport correction to compare the averaged weighted ranking of each classifier for all data sets. If the null-hypothesis is rejected, the Bonferroni-Dunn test is used as a post-hoc test to compare all classifiers against a control (majority voting). Experiments with and without feature selection were performed and showed that for high dimensional data sets the DS methods are statistically equivalent to the majority voting. For both studies, with and without feature selection, the null-hypothesis of the F_F statistic was reject with a confidence of 99.99%. Moreover, on both studies, the Bonferroni-Dunn test showed that none of the best ranked classifiers are statistically different from the majority voting classifier, which contradicts most of the results in the literature. This paper extends the research done by Maciel-Guerra *et al.* (2019) [18] by using a more comprehensive list of data sets. Our results indicate that modifications to the traditional DS framework could be beneficial.

The future work will extend the study of DS methods on high dimensional data sets with modifications proposed to the way the region of competence works. As suggested by Aggarwal *et al.* [1], the use of L_1 norm and the natural extension the authors provide is more preferable for high dimensional spaces when compared with Euclidean distance. Therefore, it would be important to investigate whether different distance metrics can improve the region of competence. As suggested by Maciel-Guerra *et al.* (2020) [19], we will focus on subspace clustering which localise their search not only in terms of samples but in terms of features as well to overcome the issues presented by k-NN on high-dimensional data sets.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44503-X_27
2. Agrawal, D., et al.: Challenges and opportunities with big data: a white paper prepared for the computing community consortium committee of the computing research association. Computing Research Association (2012)
3. Ballard, C., Wang, W.: Dynamic ensemble selection methods for heterogeneous data mining. In: 2016 12th World Congress on Intelligent Control and Automation (WCICA), pp. 1021–1026, June 2016. https://doi.org/10.1109/WCICA.2016.7578244
4. Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J.M., Herrera, F.: A review of microarray datasets and applied feature selection methods. Inf. Sci. **282**, 111–135 (2014). https://doi.org/10.1016/j.ins.2014.05.042
5. Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A.: Feature selection for high-dimensional data. Prog. Artif. Intell. **5**(2), 65–75 (2016). https://doi.org/10.1007/s13748-015-0080-y
6. Britto Jr., A.S., Sabourin, R., Oliveira, L.E.S.: Dynamic selection of classifiers - a comprehensive review. Pattern Recogn. **47**(11), 3665–3680 (2014). https://doi.org/10.1016/j.patcog.2014.05.003

7. Cruz, R.M., Zakane, H.H., Sabourin, R., Cavalcanti, G.D.: Dynamic ensemble selection vs k-NN: why and when dynamic selection obtains higher classification performance? In: The Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, Canada (2017)
8. Cruz, R.M.O., Hafemann, L.G., Sabourin, R., Cavalcanti, G.D.C.: DESlib: A Dynamic ensemble selection library in Python. arXiv preprint arXiv:1802.04967 (2018)
9. Cruz, R.M., Sabourin, R., Cavalcanti, G.D.: Dynamic classifier selection: recent advances and perspectives. Inf. Fusion **41**(Supplement C), 195–216 (2018). https://doi.org/10.1016/j.inffus.2017.09.010
10. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
11. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
12. Donoho, D.L.: High-dimensional data analysis: the curses and blessings of dimensionality. In: AMS Conference on Math Challenges of the 21st century, pp. 1–33 (2000)
13. Ghojogh, B., et al.: Feature selection and feature extraction in pattern analysis: A literature review. ArXiv abs/1905.02845 (2019)
14. Iman, R.L., Davenport, J.M.: Approximations of the critical region of the fbietkan statistic. Commun. Stat. - Theory Methods **9**(6), 571–595 (1980). https://doi.org/10.1080/03610928008827904
15. Ko, A.H.R., Sabourin, R., Britto Jr., A.S.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recogn. **41**(5), 1718–1731 (2008). https://doi.org/10.1016/j.patcog.2007.10.015
16. Kriegel, H.P., Kröger, P., Zimek, A.: Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans. Knowl. Disc. Data **3**(1), 1–57 (2009)
17. Li, J., et al.: Feature selection datasets at arizona state university. http://featureselection.asu.edu/datasets.php. Accessed August 2018
18. Maciel-Guerra, A., Figueredo, G.P., Zuben, F.J.V., Marti, E., Twycross, J., Alcocer, M.J.C.: Microarray feature selection and dynamic selection of classifiers for early detection of insect bite hypersensitivity in horses. In: IEEE Congress on Evolutionary Computation, CEC 2019, Wellington, New Zealand, 10–13 June 2019, pp. 1157–1164. IEEE (2019). https://doi.org/10.1109/CEC.2019.8790319
19. Maciel-Guerra, A., Figueredo, G.P., Zuben, F.J.V., Marti, E., Twycross, J., Alcocer, M.J.C.: Subspace-based dynamic selection: a proof of concept using protein microarray data. In: WCCI - World Congress on Computational Intelligence, The International Joint Conference on Neural Networks (IJCNN) 2020, Glasgow, UK, 19–24 July 2020. IEEE (2020)
20. de Menezes Sabino Almeida, H.A.: Selecao dinamica de classificadores baseada em filtragem e em distancia adaptativa. Master's thesis, Federal University of Pernambuco, Recife, Brazil (2014)
21. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. SIGKDD Explor. **6**, 90–105 (2004). https://doi.org/10.1145/1007730.1007731
22. Pechenizkiy, M., Tsymbal, A., Puuronen, S., Patterson, D.: Feature extraction for dynamic integration of classifiers. Fundamenta Informaticae **77**(3), 243–275 (2007)
23. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)

24. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Nearest neighbors in high-dimensional data: the emergence and influence of hubs. In: Proceedings of the 26th International Conference on Machine Learning, ICML 2009, vol. 382, p. 109, January 2009. https://doi.org/10.1145/1553374.1553485
25. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Hubs in space: popular nearest neighbors in high-dimensional data. J. Mach. Learn. Res. **11**, 2487–2531 (2010)
26. Soares, R.G.F., Santana, A., Canuto, A.M.P., de Souto, M.C.P.: Using accuracy and diversity to select classifiers to build ensembles. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp. 1310–1316 (2006). https://doi.org/10.1109/IJCNN.2006.246844
27. de Souto, M.C.P., Soares, R.G.F., Santana, A., Canuto, A.M.P.: Empirical comparison of dynamic classifier selection methods based on diversity and accuracy for building ensembles. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1480–1487, June 2008. https://doi.org/10.1109/IJCNN.2008.4633992
28. Tsymbal, A., Puuronen, S., Skrypnyk, I.: Ensemble feature selection with dynamic integration of classifiers. In: International Congress on Computational Intelligence Methods and Applications CIMA2001 (2001). https://doi.org/10.1007/3-540-39963-1_44
29. Verleysen, M., François, D.: The curse of dimensionality in data mining and time series prediction. In: Cabestany, J., Prieto, A., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 758–770. Springer, Heidelberg (2005). https://doi.org/10.1007/11494669_93
30. Woloszynski, T., Kurzynski, M.: A probabilistic model of classifier competence for dynamic ensemble selection. Pattern Recogn. **44**(10–11), 2656–2668 (2011). https://doi.org/10.1016/j.patcog.2011.03.020
31. Woloszynski, T., Kurzynski, M., Podsiadlo, P., Stachowiak, G.W.: A measure of competence based on random classification for dynamic ensemble selection. Inf. Fusion **13**(3), 207–213 (2012). https://doi.org/10.1016/j.inffus.2011.03.007
32. Yu, Z., et al.: A new kind of nonparametric test for statistical comparison of multiple classifiers over multiple datasets. IEEE Trans. Cybern. **47**(12), 4418–4431 (2017). https://doi.org/10.1109/TCYB.2016.2611020