BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **18**, No 4
Sofia • 2018 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2018-0045

A New Horizo-Vertical Distributed Feature Selection Approach

B. Zerhari[1], *A. Ait Lehcen*[2], *S. Mouline*[1]

[1]*LRIT, Associated Unit to CNRST-URAC No 29, Faculty of Sciences, Mohammed V University in Rabat, IT Center, Morocco*
[2]*LGS, National School of Applied Sciences* (*ENSA*), *Ibn Tofail University, Kenitra, Morocco*
E-mails: *zerhari.btissam@gmail.com ayoub.aitlahcen@univ-ibntofail.ac.ma salma.mouline@um5.ac.ma*

Abstract: *Feature selection technique has been a very active research topic that addresses the problem of reducing the dimensionality. Whereas, datasets are continuously growing over time both in samples and features number. As a result, handling both irrelevant and redundant features has become a real challenge. In this paper we propose a new straightforward framework which combines the horizontal and vertical distributed feature selection technique, called Horizo-Vertical Distributed Feature Selection approach* (*HVDFS*), *aimed at achieving good performances as well as reducing the number of features. The effectiveness of our approach is demonstrated on three well-known datasets compared to the centralized and the previous distributed approach, using four well-known classifiers.*

Keywords: *Feature selection, distributed approach, dimensionality reduction.*

1. Introduction

The growing size of datasets raises great challenge for both supervised and unsupervised learning. As can be observed, Datasets are continuously growing over time both in samples size and features number. There might be also problems with redundant and irrelevant features. High dimensionality implies massive memory requirements, therefore a high computational cost for learning process.

Accordingly, the need for dimensionality reduction technique has increased dramatically in recent years in order to have a small number of samples and/or a small number of features. One of these kinds of techniques is feature selection. Indeed, feature selection is acknowledged to address the problem of reducing the dimensionality by finding the most compact and informative set of features. In other words, Feature selection is defined as the process of identifying and removing irrelevant and redundant features with the goal of finding the best collection of feature subsets without significant loss of useful information or degradation of performance.

In the last few years, feature selection has been successfully applied in different domains and with different strategies (Fig. 1) to improve data storage and classification accuracy. Traditionally, feature selection methods have been designed to run in a centralized computing environment. However, it is not economic to process the whole data at once. On the other hand, most existing feature selection algorithms are not suitable for large amounts of data. In other words, their efficiency may significantly be deteriorated when dealing with a huge quantity of data to the point of becoming inapplicable. Therefore, over the last few years two major categories of distributed methods have been developed instead of the centralized approaches: horizontal and vertical approach. Data can be distributed either horizontally (by sample) [1-4] or vertically (by feature) [5-7]. Thereby, the problem of big dimensionality may be solved, as much as possible.

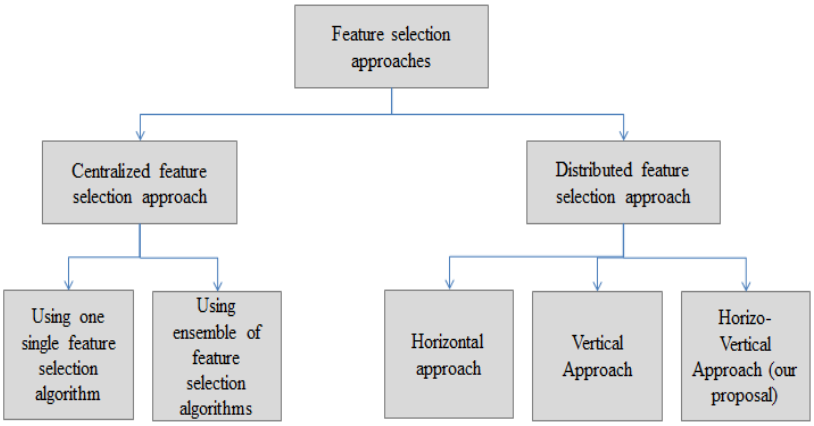

Fig. 1. Different feature selection approaches

In this work, we will propose a parallel framework for FS by distributing the data vertically and horizontally, aimed at achieving a good classification performance while reducing the input dimensionality. The idea is to split the data by both instances and features together. After having the data distributed in small subsets a filter is applied over different partitions of the data, these two first steps are repeated several rounds to obtain a stable set of features. Then, a merging procedure is performed which combine the partial results into a single subset of relevant features according to calculated threshold. Later, the two sets of features selected by the horizontal and vertical algorithm are combined to obtain the final set of features.

The remainder of this paper is organized as follows. Section 2 depicts a background of different techniques and methods used in this paper. Section 3 reviews related work. Section 4 explains the proposed methods Section 5 describes the conducted experiments Section 6 presents analysis and discussion results. Section 7 concludes the paper and gives proposals for further research.

2. Background

2.1. Feature selection algorithms

Feature selection technique has been a very active research topic since about ten years in the fields of artificial learning, data mining, image processing, and data analysis in bioinformatics [8-10]. It consists on choosing from a large set of features, a subset of relevant ones for the problem studied, which gives better model readability and interpretability in order to obtain a subset of features that accurately describes a given problem with a minimum degradation of performance.

The purpose of the selection is to find an optimal subset of features that has the following properties, while maintaining the physical meanings of the original feature sets: it must be composed of relevant features and must avoid redundant attributes. Moreover, this set must make it possible to best satisfy the objective set, namely the precision and speed of learning. There are several feature selection algorithms, the most well known in academic work are Chi-squared, InfoGain, Gain Ratio, ReliefF, Cfs, INTERACT, MRMR and Consistency-based Filter.

- Chi-squared [11]: This method computes the value of chi-squared for an attribute with respect to the class in order to evaluate the worth of this attribute. Otherwise, it is applied for determining the correlation between the decision classes and the attribute.
- InfoGain [12]: This method measures the information gain for an attribute with respect to the class in order to evaluate the worth of this attribute. The information gains are used to select the most influential attributes.
- Gain Ratio [13]: This method measures the gain ratio of an attribute with respect to the class in order to evaluate the worth of this attribute. The disadvantage of the Gain Ratio is that it computes a weight for an attribute without examining other available attributes. This can be a big problem especially if features are dependent.
- ReliefF [14]: This method choses randomly an instance and considers the value of the given attribute for the nearest instance of the same and different class, in order to evaluate the worth of this attribute. It is one of the most successful and most widely used feature selection algorithms.
- Cfs (Correlation-based feature selection) [15]: The attributes' subsets highly correlated with class are evaluated by this method while choosing the lower intercorrelated ones. CFS is a fully automatic algorithm, in other words it does not require the user to specify any thresholds or the number of features to be selected.
- Consistency-based Filter [16]: This method measures the level of consistency in the class values when the instances are projected into the subset of attributes, in order to evaluate the worth of a subset of features.
- MRMR (Minimum Redundancy Maximum Relevance) [17]: This algorithm tends to selects features that should be both minimally redundant among themselves and maximally relevant to the target classes. The features are ranked according to the minimal-redundancy-maximal-relevance criteria based on mutual information.
- INTERACT [18]: This algorithm is based on Symmetrical Uncertainty (SU); it combines an information measure and a consistency measure. The first part of the algorithm requires a threshold, in the second part searches, features are evaluated according to their C-contribution which relies on the calculation of inconsistency rate. INTERACT can handle feature interaction efficiently.

Feature selection, as a type of dimension reduction technique, has been proven to be effective and efficient in handling high dimensional data. Indeed, the removal of irrelevant and redundant features reduces the computational and storage costs without significant loss of information or negative degradation of the learning performance.

The growing of dataset sizes in last years presents some challenges to the traditional feature selection task. Currently, there are some attempts to replace centralized data mining by distributed techniques to perform parallel feature selection, as ways to reduce costs.

2.2. Centralized vs distributed approach

The standard approach to data mining is centralized. Even though, this process is easy to understand, and the data-mining software design is straightforward, there are a number of drawbacks to the centralized approach: Centralizing the entire dataset would be very costly and impractical because of the large number of data sources. On the other hand the performance of most systems today is limited by the memory capacity, so the need to distribute the storage is important and necessary [3].

Partitioning means dividing the original training set into smaller training sets to make parallel process. A different algorithm is trained on each subset. After that, the different outputs are combined in some fashion [2]. There are two main ways to partition the dataset: Horizontal Partitioning and Vertical Partitioning.

In horizontal partitioning the train dataset is partitioned by samples into several subsets (Fig. 2). Each one contains a subset of the instances and the same features as the original. This parallel processing will not only overcome the issue of exceeding memory size, but will also lead to creating an ensemble of diverse and accurate classifiers, each built from a disjoint partition but with the aggregate processing all of the data [1].

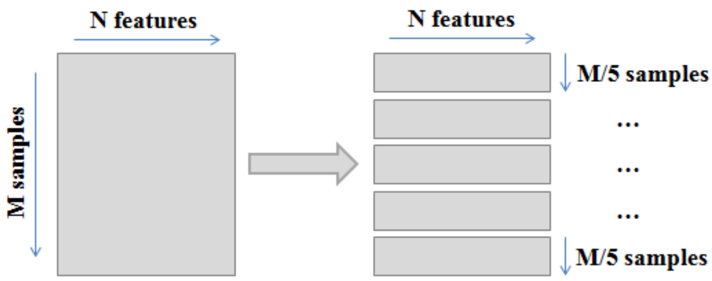

Fig. 2. Horizontal technique to partition data

In vertical partitioning the original dataset is partitioned by features into several datasets (Fig. 3). Each one has the same number of instances and contain a subset of features in the original dataset, each containing a subset of the original set of features. The idea is to simply give each classifier a different projection of the training set. There are three popular strategies for creating feature subset-based ensembles: Random-based, reduct-based and collective performance based strategy [6]. In this paper we opted for the first option.

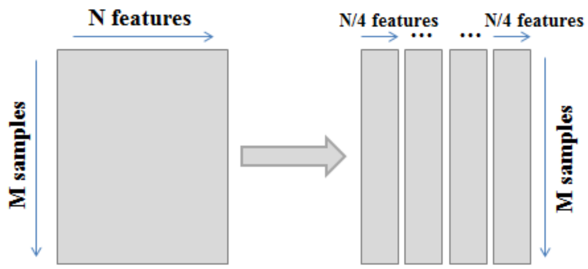

Fig. 3. Vertical technique to partition data

2.3. Classification

In the following subsection, the classifiers whereby we evaluated our proposal are briefly discussed. Four well-known classifiers are used each one of it is belonging to a different family, two linear (NB and SVM using a linear kernel) and two non-linear (C4.5 and KNN).

- C4.5 [19]: The C4.5 is decision tree algorithm, decision trees are among most popular classification techniques. They owe popularity to their straightforward process. C4.5 is based on the ID3 algorithm to which it brings several improvements. This algorithm can deal with both numerical and symbolic data.
- Naïve Bayes [20]: The naive Bayesian classifier is one of the simplest methods of supervised learning based on the Bayes theorem. The simplicity of programming, the ease of parameter estimation and the speed are the advantages of this probabilistic classifier.
- SVM [21]: The SVM is a supervised learning tool. This is one of the most successful learning algorithms, with the ability to compute complex models for the computational cost. This discriminative classifier aims to find the separation between two classes of objects with the idea that the larger separation, the more robust classification. In its simplest form, that of a linear separation and separable classes, the algorithm selects the hyperplane that separates the set of observations into two distinct classes in order to maximize the distance between the hyperplane and the most closer observations to the learning sample.
- KNN [22]: The KNN algorithm is one of the simplest artificial learning algorithms. In a classification context of a new observation, the simple founding idea is that the nearest neighbors vote on this observation. It predicts the class of a testing observation that is dominant among the K most similar observations. The KNN method is therefore a non-parametric neighbourhood method.

3. Related works

With the growing of data size, Feature selection algorithms face a major challenge since it is difficult to deal with a high number of input features and instances. The distributed approach has been receiving a growing amount of attention in last years. Recently several authors have investigated the distributed method to solve this problem.

There are two main ways to partition the dataset: Horizontal and Vertical Partitioning. In [23], for instance, an approach which splits the data horizontally was proposed. A filter is applied at each partition. Later, a merging procedure is performed in order to combine the results into a single subset of relevant features. The drawback of this approach is that, by involving a classifier in the process of selecting the optimal threshold, in some cases the time necessary for this task was higher than the time required by the feature selection process. Later, the same authors in [24], propose a new methodology for merging procedure using the theoretical complexity measures, applied to horizontal distributed feature selection process. Another work [25], addresses the problem of feature selection in a large P2P environment. The authors have developed a local distributed privacy preserving algorithm when the data are dispatched across a large number of machines. A new classifier combination strategy was presented in [26]. As result a fast and effective distributed datamining of large classifier ensembles is achieved. Evolutionary Feature Selection for Big Data Classification was proposed [27], using the MapReduce paradigm to obtain subsets of features from big datasets. Researcher in [28] examines a decision tree framework for space decomposition with grouped gain-ratio. Indeed, the original instance-space is hierarchically partitioned into multiple subspaces. After that, a distinct classifier is assigned to each subspace.

While not common, there are some other developments that distribute the data by features. In [29], for instance, a new approach was designed to predict from vertically partitioned data. Such data is collected via multiple channels; each local site builds a predictor based on the corresponding features using any base prediction algorithm. Approaches that partition data by attributes were introduced [30]; result shows that this method is simple and can even increase classification accuracy compared to a centralized approach. In their work [31], authors present parallel filter approach for vertically distributing the data. Although the experiments showed that execution time was considerably shortened compared to centralized technique. The drawback of this methodology was its dependence on the classifier used. More recently, the same authors propose a distributed approach based on data complexity measures [32], this method was carried out for both the horizontal and the vertical technique. To combine the partial outputs obtained from feature selection algorithm applied to each subset, a merging process using the theoretical complexity is applied to these feature subsets. In [33] a distributed privacy preserving protocol was proposed to perform feature selection for vertical and horizontal distributions using virtual dimension. In all of the aforementioned works, each approach has some vulnerability, either in terms of classification accuracy, execution runtime and/or storage requirements.

Following the recommendations in [32], the use of the distributed approaches is recommended instead of the traditional centralized methods when dealing with large datasets. On the other hand, the vertical distribution is recommended in case that better classification performance is more important than a smaller storage requirement. Nevertheless, the big drawback for the vertical distribution in which the features are distributed across the packets is that: it will be difficult to detect redundancy between these different packets. Whereas, the horizontal distribution is preferable in case of reducing the storage requirements and runtime is more important than classification accuracy.

In the next sections we will confirm the recommendations presented above, in addition we will show the advantage of our approach compared to other approaches in the literature.

4. Our approach: Horizo-vertical distributed feature selection method

We propose a new framework which combines the horizontal and vertical distributed feature selection scheme (Fig. 4), called Horizo-Vertical Distributed Feature Selection approach (HVDFS).

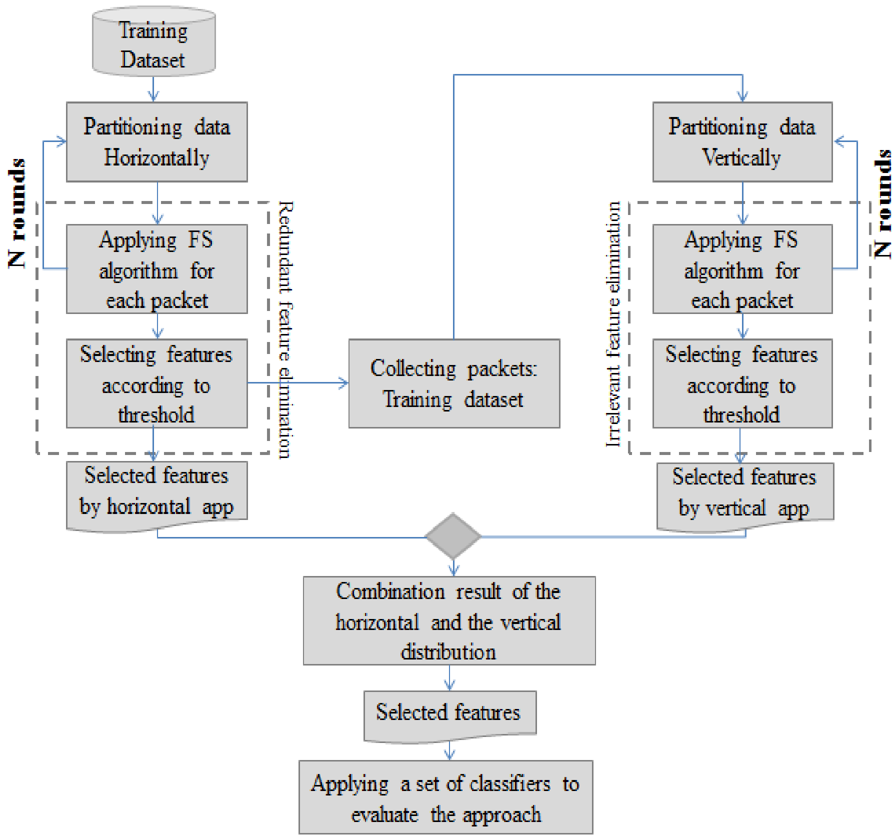

Fig. 4. Process of the horizo-vertical distribution feature selection approach

We will show later that the proposed framework is able to overcome a common drawback of the existing approaches previously mentioned. We can summarize the proposed procedure in the following five stages:

- Partition of the training datasets in several packets (by samples and features).
- Application of the feature selection methods to the subsets in several rounds to select relevant attributes.

- Combination of different outputs into a single feature subset for each distribution technique.
- Combination of the result of the horizontal and vertical distribution.
- Build classifiers to evaluate the selection features.

We partition dataset horizontally in disjoints partitions of the same size. After that, we apply a powerful feature selection algorithm on detecting irrelevant features to each of these partitions. Then, we combine the partial outputs into single features subsets. This procedure will be repeated on several rounds in order to capture enough information. On the other side we partition dataset vertically, this time we apply a feature selection algorithm that is more efficient in detecting redundant features such as the MRMR and Relief. Then the features selected are removed according to predefined threshold. Next, we combine the partial output from different partition. Finally, the results of the two algorithms are combined to obtain a final selection of features.

In order to evaluate the performance of our framework classification algorithms belonging to a different family (linear and non- linear) is used.

5. Experimental results

5.1. Data and tools

In order to evaluate the performance and effectiveness of the proposed distributed framework we use three datasets (Isolet, 11-Tumors and Modelon dataset) presented in Table 1 in terms of the number of features, training and test samples, classes and packets. The training datasets were divided maintaining the class distribution as following: 2/3 for training and 1/3 for testing.

We first present briefly the three datasets, the first and the second one can be free downloaded from the UCI Machine Learning Repository [34], the third one from Gene Expression Model Selector (GEMS) [35].

Table 1. Classification accuracy achieved by the horizontal distributed approach with different alpha values

Dataset	Number of Features	Number of Training	Number of Test	Number of Class	Number of Packets	Download
Isolet	617	6238	1559	26	5	[34]
11-Tumor	500	1600	800	2	3	[34]
Madelon	12,534	114	58	11	145	[35]

- Isolet: The Isolet dataset is composed of 150 speakers uttering 26 letters of the alphabet twice, which mean each speaker, contributed 52 training examples; thereby we have in total 7797 examples. The task is to classify a letter which has been uttered based on 617 features such as spectral coefficients, contour and sonorant features.
- 11-Tumor: This dataset contains 172 samples with 12,534 characteristics from 11 various human tumor types, a total of 12 classes.
- Madelon: The Madelon is a two-class classification problem, with 2400 samples points situated on the vertices of a five dimensional hypercubes and 500

continuous input variables assigned to each vertices. It was part of the NIPS 2003 feature selection challenge.

The distributed approach proposed herein can be used with any feature selection method; in our experiments we choose the CFS algorithm for detecting irrelevant features and MRMR for detecting the redundant ones, according to the recommendation in [36].

For testing the adequacy of our proposal, we select four widely used classifiers: C4.5, naïve Bayes, k-Nearest Neighbor and Support Vector Machine, more details are described in the previous section. We also compare our proposal with the horizontal and the vertical strategies proposed in [32], as well as the centralized approach in terms of number of selected features and classification accuracy. Furthermore, for the distributed approaches, it is mandatory to choose the adequate threshold to combine the partial outputs. For this reason we performed our proposed approach with different threshold to find the value that gives the best classification accuracy.

The experiments described here were conducted on a Windows 8 PC with 1.8 GHz Intel Core i5. All algorithms were implemented using Matlab 7.0 environment and the WEKA [37] Version 3.6 data mining toolkits with its default values.

5.2. Horizontal vs vertical vs horizo-vertical distribution

In this section we present and discuss experimental results in terms of classification accuracy and number of selected features. Four different approaches have been compared: centralized approach, horizontal distribution, vertical distribution and horizo-vertical distribution. For each of these strategies, we evaluate the classification accuracy using four classifiers C4.5, SVM, NB and kNN. In order to find the adequate threshold with the proposed distributed method we use different values of the threshold of votes (consensus, majority, complexity measure and logarithmic value suggested by Yu and Liu).

5.2.1. Accuracy evaluation

Keeping these in mind, we develop a novel approach which can efficiently deal with both irrelevant and redundant features, and obtain a good feature subset. Hence, good classification accuracy is achieved.

Table 2 shows the results obtained by the algorithms in terms of classification accuracy using C4.5, NB, kNN and SVM. For the centralized as well as horizontal, vertical and horizo-vertical distributed approaches.

Table 2. Comparing the centralized approach and the horizontal, vertical and our horizo-vertical distributed approach in term of Classification accuracy

Classification accuracy	Isolet				11-Tumor				Modelen			
	H	V	HV	C	H	V	HV	C	H	V	HV	C
C4.5	85	86	97	82	88	89	99	88	89	89	89	88
NB	80	85	98	81	86	87	94	86	86	85	87	86
SVM	82	86	98	82	88	88	99	87	87	89	90	86
KNN	83	86	98	82	88	89	99	87	88	90	91	87

Table 2 shows the classification accuracy achieved by each filter over the three datasets with the majority voting. As we can see, the best classification performances were obtained with our proposed approach for all classifiers measure.

5.2.2. Threshold impact

We compared different strategies of combination of results using our approach: threshold based on complexity measures, majority voting, consensus voting, and logarithmic measure.

- Complexity measures: These measures aim to identify data particularities which imply some difficulty for the classification task beyond estimates of error rates. They are a relatively recent proposal by Ho and Basu.
- Majority voting: Take the majority vote of the features selection algorithms [39].
- Consensus voting: All features selection algorithms must agree on the same vote.
- Logarithmic measure: This measure is calculated as value of the $m/\log m$ where m is the number of features in a given data set as suggested in [38]

Table 3. Comparing different combination strategies in term of Classification accuracy

Accuracy	Majority voting	Consensus voting	Logarithmic measure	Complexity measure
Horizontal	85	83	87	86
Vertical	88	85	88	87
Horizo-vertical	92	89	95	90

As expected (Table 3), the best classification accuracy for our algorithm was achieved by the logarithmic measure, compared to others strategies.

High runtimes are required to find the threshold of vote complexity measures. So, it is preferred to establish a fixed threshold and not performing a specific calculation.

5.2.3. Number of feature selection

Table 4 show the number of features selected by the centralized approach and the horizontal, vertical and our horizo-vertical distributed approach. As can be seen, the number of features selected by the vertical approaches was larger than those selected by our distributed methods as well as all others approaches. The reason behind is that, with the vertical partition the features were distributed across the packets, so detecting redundancy between features will be more difficult if they were in different partitions [32]. Whereas, there is no significant differences between the number of features selected by vertical and the centralized approach. On the other hand, a set of features selected by our distributed approach was smaller than in the case of the others approaches.

Table 4. Number of features selected by the centralized approach and the horizontal, vertical and our horizo-vertical distributed approach

Number of features	Isolet	11-Tumor	Modelon
Horizontal approach	93	236	15
Vertical approach	200	455	23
Horizo-vertical approach	14	57	10
Centralized approach	125	312	18

Researchers notice that a large number of features are not necessarily more informative because there is a risk that they are either irrelevant or redundant [40]. Therefore, selecting a small number of relevant features from large number of features in fair and reasonable way is essential for efficient classification.

We can conclude that the proposed algorithm not only reduces the number of features, but also improves the performances of the four well-known different types of classifiers.

6. Analysis and discussion

The experiments on three datasets with number of features variate between 500 and 12,534 and instances between 172 and 7779, showed that our proposal was able to achieve good classification accuracy as well as, to reduce the number of features selected.

Most existing feature selection algorithms do not scale well using standard filtering approach, and their efficiency deteriorated in a noteworthy way when it comes to large-scale data [24]. This can be very costly and impractical. That is the reason why it is highly advisable to divide datasets by one of the distributed approaches, depending on which factor determines the complexity of the problem (features, instances or both of them).

Following the recommendations in [32], the vertical approach has the drawback to not handle redundant features. Indeed, with the vertical partition, the features were distributed across the packets thereby; it will be more difficult to detect redundancy between them. Whereas, the horizontal partition is not recommended if better classification performance is more important than smaller storage requirement. When proposing the Horizo-vertical distributed approach, the goal is to enhance the classification performance whilst reducing the dimensionality by selecting small features set. Our proposal is straightforward process; it can be used with any feature selection algorithm. To summarize briefly our recommendation we can say is that it is preferrable to use the horizontal approach when we have a large number of instances, while it is recommended to use the vertical approach when we have a large number of features. Whereas our approach is more beneficial in the case of a large number of both instances and features.

A data set with n features and m instances is classified as small or large depending on the nature of the data and the domain in which it belongs. In [36], algorithms for dataset characteristics discretization are proposed.

7. Conclusion

A large number of features and instances can cause overfitting data [38]. In this new proposal, we are choosing to tackle the distributed feature selection approach because it is an important issue which directly impact the quality of the model classification. With our proposal, a process of distributed feature selection becomes able to overcome the drawbacks of the existing approaches previously mentioned.

In this work, we have proposed a new distributed feature selection process. The proposed approach has been able to successfully distribute the data using both features and instances, reducing the set of features selected and achieving good classification performance.

As future work, we plan to use the proposed approach with larger datasets of both samples and features and to try other combination strategies.

References

1. Chan, P. K., S. J. Stolfo. Toward Parallel and Distributed Learning by Meta-Learning. – In: Proc. of AAAI Workshop in Knowledge Discovery in Databases, 1993, pp. 227-240.
2. Ananthanarayana, V. S., D. K. Subramanian, M. N. Murty. Scalable, Distributed and Dynamic Mining of Association Rules. – In: Proc. of International Conference on High-Performance Computing, Springer, Berlin, Heidelberg, 2000, pp. 559-566.
3. Tsoumakas, G., I. Vlahavas. Distributed Data Mining of Large Classifier Ensembles. – In: Proc. of Companion Volume of the Second Hellenic Conference on Artificial Intelligence, 2002, pp. 249-256.
4. Das, K., K. Bhaduri, H. Kargupta. A Local Asynchronous Distributed Privacy Preserving Feature Selection Algorithm for Large Peer-To-Peer Networks. – Knowledge and Information Systems, Vol. **24**, 2010, No 3, pp. 341-367.
5. Sheela, M. A., K. Vijayalakshmi. Partition Based Perturbation for Privacy Preserving Distributed Data Mining. – Cybernetics and Information Technologies, Vol. **17**, 2017, No 2, pp. 44-55.
6. Skillicorn, D. B., S. M. McConnell. Distributed Prediction from Vertically Partitioned Data. – Journal of Parallel and Distributed Computing, Vol. **68**, 2008, No 1, pp. 16-36.
7. Rokach, L. Taxonomy for Characterizing Ensemble Methods in Classification Tasks: A Review and Annotated Bibliography. – Computational Statistics & Data Analysis, Vol. **53**, 2009, No 12, pp. 4046-4072.
8. Hasnat, A., A. U. Molla. Feature Selection in Cancer Microarray Data Using Multi-Objective Genetic Algorithm Combined with Correlation Coefficient. – In: Proc. of International Conference on Emerging Technological Trends, IEEE, 2016, pp. 1-6.
9. Saeys, Y., I. Inza, P. Larrañaga. A Review of Feature Selection Techniques in Bioinformatics. – Bioinformatics, Vol. **23**, 2007, No 19, pp. 2507-2517.
10. Ding, C., H. Peng. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. – Journal of Bioinformatics and Computational Biology, Vol. **3**, 2005, No 2, pp. 185-205.
11. Satorra, A., P. M. Bentler. A Scaled Difference Chi-Square Test Statistic for Moment Structure Analysis. – Psychometrika, Vol. **66**, 2001, No 4, pp. 507-514.
12. Kononenko, I. Estimating Attributes: Analysis and Extensions of RELIEF. – In: Proc. of European Conference on Machine Learning, 1994, pp. 171-182.
13. Dai, J., Q. Xu. Attribute Selection Based on Information Gain Ratio in Fuzzy Rough Set Theory with Application to Tumor Classification. – Applied Soft Computing, Vol. **13**, 2013, No 1, pp. 211-221.

14. S i k o n j a, M. R., I. K o n o n e n k o. An Adaptation of Relief for Attribute Estimation on Regression. Machine Learning. – In: Proc. of 14th International Conference on Machine Learning, Nashville, 1997, pp. 296-304.
15. H a l l, M. A. Correlation-Based Feature Subset Selection for Machine Learning. – In: Thesis Submitted in Partial Fulfilment of the Requirements of the Degree of Doctor of Philosophy at the University of Waikato, 1998.
16. D a s h, M. H. L i u. Consistency-Based Search in Feature Selection. – Artificial Intelligence, Vol. **151**, 2003, No 1-2, pp. 155-176.
17. P e n g, H., F. L o n g, C. D i n g. Feature Selection Based on Mutual Information Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **27**, 2005, No 8, pp. 1226-1238.
18. D e i s y, C., S. B a s k a r, N. R a m r a j, J. S. K o o r i. P. J e e v a n a n d a m. A Novel Information Theoretic-Interact Algorithm (IT-IN) for Feature Selection Using Three Machine Learning Algorithms. – Expert Systems with Applications, Vol. **37**, 2010, No 12, pp. 7589-7597.
19. Q u i n l a n. J. R. C4.5: Programs for Machine Learning. – In: Elsevier, Machine Learning, Morgan Kaufmann Publishers, 2014.
20. N i e l s e n, T. D., F. V. J e n s e n. Bayesian Networks and Decision Graphs. – In: Springer Science and Business Media, 2009.
21. C h e r k a s s k y, V., Y. M a. Practical Selection of SVM Parameters and Noise Estimation for SVM Regression. – Neural Networks, Vol. **17**, 2004, No 1, pp. 113-126.
22. G u o , G., H. W a n g, D. B e l l, Y. B i, K. G r e e r. KNN Model-Based Approach in Classification. – In: Proc. of OTM Confederated International Conferences on the Move to Meaningful Internet Systems, Springer, Berlin, Heidelberg, 2003, pp. 986-996.
23. B o l ó n-C a n e d o, V., N. S á n c h e z-M a r o ñ o, J. C e r v i ñ o-R a b u ñ a l. Scaling up Feature Selection: A Distributed Filter Approach. – In: Proc. of Conference of the Spanish Association for Artificial Intelligence, Springer, Berlin, Heidelberg, 2013, pp. 121-130.
24. B o l ó n-C a n e d o, V., N. S á n c h e z-M a r o ñ o, A. A l o n s o-B e t a n z o s. A Distributed Feature Selection Approach Based on a Complexity Measure. – In: Proc. of International Work-Conference on Artificial Neural Networks. Springer, Cham, 2015, pp. 15-28.
25. D a s, K., K. B h a d u r i, H. K a r g u p t a. A Local Asynchronous Distributed Privacy Preserving Feature Selection Algorithm for Large Peer-to-Peer Networks. – Knowledge and Information Systems, Vol. **24**, 2010, No 3, pp. 341-367.
26. T s o u m a k a s, G., I. V l a h a v a s. Distributed Data Mining of Large Classifier Ensembles. – In: Proc. of Companion Volume of the Second Hellenic Conference on Artificial Intelligence. 2002.
27. P e r a l t a, D., S. D e l Río, S. R a m í r e z-G a l l e g o, I. T r i g u e r o, J. M. B e n i t e z, F. H e r r e r a. Evolutionary Feature Selection for Big Data Classification: A Mapreduce Approach. – Mathematical Problems in Engineering, Vol. **2015**, 2015.
28. C o h e n, S., L. R o k a c h, O. M a i m o n. Decision-Tree Instance-Space Decomposition with Grouped Gain-Ratio. – Information Sciences, Vol. **177**, 2007, No 17, pp. 3592-3612.
29. S k i l l i c o r n, D. B., M. M. S a b i n e. Distributed Prediction from Vertically Partitioned Data. – Journal of Parallel and Distributed Computing, Vol. **68**, 2008, No 1, pp. 16-36.
30. M c C o n n e l l, S., D. B. S k i l l i c o r n. Building Predictors from Vertically Distributed Data. – In: Proc. of the Conference of the Centre for Advanced Studies on Collaborative Research. IBM Press, 2004, pp. 150-162.
31. B o l ó n-C a n e d o, V., N. S á n c h e z-M a r o n o, J. C e r v i n o-R a b u n a l. Toward Parallel Feature Selection from Vertically Partitioned Data. – In: Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium, 2014, pp. 23-25.
32. M o r á n-F e r n á n d e z, L., V. B o l ó n-C a n e d o, A. A l o n s o-B e t a n z o s. Centralized vs. Distributed Feature Selection Methods Based on Data Complexity Measures. – Knowledge-Based Systems, Vol. **117**, 2017, pp. 27-45.
33. B a n e r j e e, M., S. C h a k r a v a r t y. Privacy Preserving Feature Selection for Distributed Data Using Virtual Dimension. – In: Proc. of 20th ACM International Conference on Information and Knowledge Management. ACM. 2011.

34. B a c h e, K., M. L i n c h m a n. UCI Machine Learning Repository. – In: University of California, Irvine, School of Information and Computer Sciences. Online; Accessed January 2016. **http://archive.ics.uci.edu/ml/**
35. University, V. Gene Expression Model Selector. Online. Accessed January 2016. **http://www.gems-system.org/**
36. O r e s k i, D., O. S t j e p a n, K. B o z i d a r. Effects of Dataset Characteristics on the Performance of Feature Selection Techniques. – Applied Soft Computing, Vol. **52**, 2017, pp. 109-119.
37. H a l l, M., E. F r a n k, G. H o l m e s, B. P f a h r i n g e r, P. R e u t e m a n n, I. H. W i t t e n. The Weka Data Mining Software: An Update. – ACM SIGKDD Explorations Newsletter, Vol. **11**, 2009, No 1, pp. 10-18.
38. S o n g, Q., N. J i n g j i e, W. G u a n g t a o. A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data. – IEEE Transactions on Knowledge and Data Engineering, Vol. **25**, 2013, No 1, pp. 1-14.
39. D h a r m a r a j, R. P., J. B. P a t i l. Malicious URLs Detection Using Decision Tree Classifiers and Majority Voting Technique. – Cybernetics and Information Technologies, Vol. **18**, 2018, No 1, pp. 11-29.
40. X i n g, E., M. J o r d a n, R. K a r p. Feature Selection for High-Dimensional Genomic Microarray Data. – In: Proc. of Eighteenth International Conference on Machine Learning, 2001, pp. 601-608.

Received 05.04.2018; Accepted 08.09.2018