Metalearning to Choose the Level of Analysis in Nested Data: A Case Study on Error Detection in Foreign Trade Statistics

Mohammad Nozari Zarmehri
INESC Porto
Faculty of Engineering
University of Porto
Rua Dr. Roberto Frias, 378, 4200–465
Email: mohammad.nozari@fe.up.pt

Carlos Soares
INESC Porto
Faculty of Engineering
University of Porto
Rua Dr. Roberto Frias, 378, 4200–465
Email: csoares@fe.up.pt

***Abstract*—Traditionally, a single model is developed for a data mining task. As more data is being collected at a more detailed level, organizations are becoming more interested in having specific models for distinct parts of data (e.g. customer segments). From the business perspective, data can be divided naturally into different dimensions. Each of these dimensions is usually hierarchically organized (e.g. country, city, zip code), which means that, when developing a model for a given part of the problem (e.g. a zip code) the training data may be collected at different levels of this nested hierarchy (e.g. the same zip code, the city and the country it is located in). Selecting different levels of granularity may change the performance of the whole process, so the question is which level to use for a given part. We propose a metalearning model which recommends a level of granularity for the training data to learn the model that is expected to obtain the best performance. We apply decision tree and random forest algorithms for metalearning. At the base level, our experiment uses results obtained by outlier detection methods on the problem of detecting errors in foreign trade transactions. The results show that using metalearning help finding the best level of granularity.**

I. Introduction

Data Mining (DM) projects are often based on the raw data obtained from Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM) systems. The data collected will be used to obtain knowledge that can be used to improve the business processes of the organization. To ensure that these goals are achieved, a methodology like CRoss Industry Standard Process for Data Mining (CRISP-DM [1] should be followed.

Generally, a single model is created by DM algorithms for all data. For example, a single model is generated to predict sales for all products. But it has already been shown that there is no commonly best algorithm for a broad domain of problems [2]. As more data becomes available, organizations are not satisfied with single model for dealing with their prediction problems. For different areas of the data space, they want different models, that are well suited for those areas (e.g. different customer segments or different product categories) [3]. The first arisen question is how to split the data. We believe that Business Intelligence (BI) can be used for that purpose because a lot of effort has been invested into identifying the data dimensions that are relevant for the business (i.e. implemented as the data cubes).

The second question is the granularity of the split. In BI the values of a dimension are organized hierarchically. The best models for a given subset of the data may be obtained by training with data from other, related subsets (e.g. if the amount of the data available for a given product is small, a more reliable model may be obtained by training with data from other products in the same category) [4].

In addition, a data hierarchy has been carefully designed by experts. The corresponding subsets are meaningful from a business perspective. Therefore, subsets defined by Data Warehouse (DW) dimensions are generally expected to represent partitions of the data which may be useful to improve learning processes. However, since there are multiple levels, finding the best subset for learning a model by DM is a crucial task. One solution can be metalearning.

Metalearning is learning about learning processes [5]. It models the relationship between the characteristics of the data with the performance of the algorithms. It is often used to select the best algorithm for a specific problem, such as classification or regression.

In this paper we address this question: How to choose the right level of granularity, as defined by

978-1-4799-1959-8/15/$31.00 ©2015 IEEE

DW dimensions, to model a DM problem? We use a metalearning approach, in which the characteristics of the data are mapped to the performance of the learning algorithms at different levels of granularity.

The paper is organized as follows. Section II summarizes some background knowledge about the data, previous results, and metalearning fields. Section III describes our methodology for data analysis and metalearning to find the best level of granularity. The obtained results are presented in Section IV. The results are discussed in Section V. Finally, a conclusion and the future work are presented in Sections VI and VII, respectively.

II. Background

In this section, we start by introducing the case study and then we summarize the metalearning.

A. *Case study: error detection in foreign trade statistics*

Foreign trade statistics are important to describe the state of the economy of countries [6]. They are usually estimated by the different national statistics institutes based on data provided by companies. However, this data often contains errors because companies do not always appreciate the importance of providing accurate information. If undetected, these errors may, in some cases, have a significant impact on the value of the statistics. Therefore, national statistics institutes, such as the Portuguese Institute of Statistics (Instituto Nacional de Estatistica – INE), apply a combination of automatic and manual procedures to identify and correct those errors. Their goal is to detect as many errors as possible – to maximize the quality of the statistics – with as little manual effort as possible – to minimize the cost.

Some of the previous work on error detection has used outlier detection, classification and clustering approaches (e.g., [4], [6]). In general, satisfactory results have been obtained as some approaches were able to detect most of the erroneous transactions by choosing a small subset of suspicious transactions for manual inspection. However, this was not true for all products. This is partly due to the fact that some products have very few transactions. Given that each product is analyzed individually, the decision can be based on a very small set of data.

In [4], investigation of improvement of previous results by aggregating the data from different products based on the product taxonomy was done. The INE data (See Table I) contains all the transactions for months 1, 2, 3, 5, 6, 8, 9, 10 in 1998 and months 1, 2 in 1999. The products are organized in a 4-levels taxonomy. An example of such a taxonomy can be: Food (Level 4), Bread (Level 3), Sliced bread (Level 2), Pack of 16 slices (Level 1) (Fig. 1).

Each product is presented with a unique 8-digits product code (Level 1). By going up in the product taxonomy, the number of products in each level increases. Therefore, by grouping the transactions at a higher level of the product taxonomy may help obtaining better results when compared to an analysis at the product level (Level 1) itself, especially in the cases where the amount of data at this level is too small.

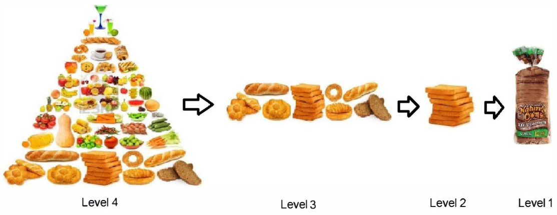


Fig. 1. An example of existing taxonomy in foods

According to previous work, the best results are obtained at different levels of the taxonomy for different products (Fig. 2). For example, the best results for the products on the right leaf are obtained at the third level of product taxonomy while for the products at the middle leaf, the best results are obtained at the second level (black models in Fig. 2). In spite of the fact that their results show that the aggregation is generally useful (Fig. 3), they also show that the best results for different products are obtained at different levels of granularity (Fig. 4).

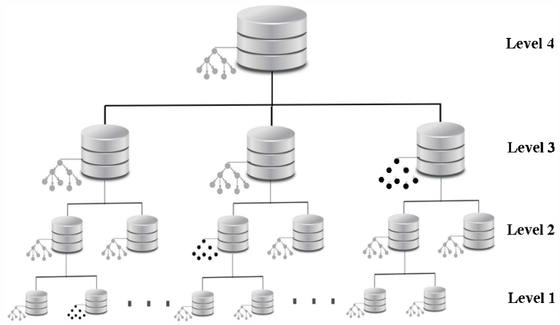


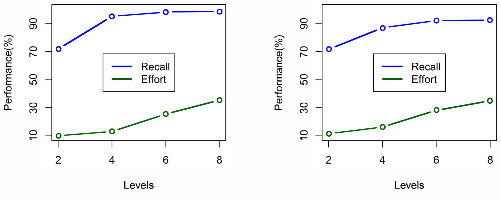
Fig. 2. Illustration of a hierarchy in datasets: for each category the best performance (black model) is obtained at different levels

B. *Metalearning*

The metalearning strategies has been studied during the last decade. Various meaning can be found in literature for metalearning [7]–[10]. The unity of

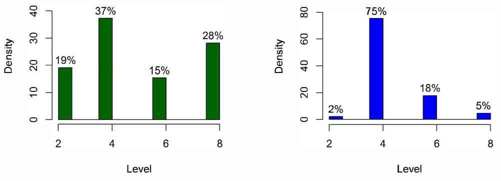
TABLE I
LIST OF FEATURES IN THE DATASET [6]

FEATURE	DESCRIPTION
RN	Row number
O	Origin: Type of method used to insert the data into the data set (1: Disk, 2: Manual, 3: Web)
IO	In/Out: Flow of materials (1: arrival, 2: dispatch)
L	Lot number: Group of transactions that were shipped together
DN	Document number: Official document number
OID	Operator ID: Company that is responsible for the transaction
M	Month: Month of the transaction
LN	Line number: Line number in the document
C	Country: Country of origin/destination
PC	Product code: Code of the product
W	Weight of the traded goods
TC	Total cost
I/E	Direction: (1: Import, 2: Export)
TCW	Total cost/weight
AvgWM	Average Weight/Month: Average weight of the transactions made in the same month of the product which the current transaction is from
SDWM	Standard Deviation of Weight/Month: Standard deviation of AvgWM
S	Score: Normalized distance of the Total cost/weight value to the average value [6]
TN	Transaction number: Number of transactions made in the same month of the product which the current transaction is from
E	Error: Target value (1: error, 0: normal transaction).



(a) (b)

Fig. 3. Aggregation is generally useful: (a) Import, (b) Export [4]



(a) (b)

Fig. 4. The best results are obtained at the different levels: (a) Best effort, (b) Best recall [4]

references agrees to consider the metalearning approach as a research structure for discovering learning algorithms that improve their efficiency [11]. The main idea is that high-quality dataset characteristics (metafeatures) provide some information to differentiate the performance of a set of given learning strategies [12].

The characterization of a classification problem and its effect on algorithm performance is investigated by Rendell and Cho [13]. They use the size and concentration of the classes as features. This idea was extended in 1992 by Aha [14]. The number of examples, number of classes, number of prototypes per class, number of relevant and irrelevant attributes, and the distribution range of examples and prototypes were the selected features. Aha creates rules for learning, i.e.,

if a given dataset has specific characteristics (C_1, C_2, ..., C_n) then algorithm A_1 should be selected.

In 1976, John Rice used the term algorithm selection [15] to describe the correlation of data characteristics of a specific problem with the performance of the algorithms. In addition, a metalearning algorithm for supporting the selection of learning algorithms is presented in [16].

The problem of selecting the level of granularity used for the training data used for different products can be addressed with a metalearning approach. It can be used to map the characteristics of the data with the ideal level of granularity [17]. DM models can be learned at different levels of granularity. Consequently, the data characteristics can be calculated and used to determine the best level for each product [6], [18].

On the far side of the task of algorithm selection problem, there are many other problems that the same idea can be derived. For example to select the best parameter settings (Levels in our case), a metalearning approach can be used. In 2005, Shawkat and Smith [19] utilized metalearning to find the best kernel to use with support vector machines (SVMs). By changing the kernel, the algorithm essentially changes and there will be different performances for each setting.

III. Methodology

Fig. 5 shows our proposed metalearning model. The problem is to find the best level of granularity to apply an outlier detection algorithm to have the best performance. For each unique product, metafeatures (see Section III-A) are calculated. The metadata is consist of the metafeatures and the best performance obtained by base-level. Then the metadata is used for training in the metalearning process.

In our experiment, there are different products, P_i for $i = 1, ..., n_1$ while n_1 is the number of unique products at the first level. Products are organized in a four-levels hierarchy ($C_i^j, j = 1, ..., k$)) where $k = 4$. Each product is distinguished by an 8-digits code. Level one contains all the products with the same 8-digits code. Level two includes all the products with the same first 6-digits code. Similarly, levels three and four contain the products with the same 4- and 2-digits code, respectively. For example, all the dried fruits have a product code starting with 11 while raisins, which are a specific type of dried fruits, have code 1155.

The best model for a given product depends on the product itself and the data available. For instance, given a very specific product with a lot of transactions, the best results can be obtained by learning a model on training data from that product only. On the other hand, if the product is very general behavior or has very few transactions, the best model can be obtained by training with the full dataset. It is expected that in other cases the best model can be obtained by training on data from intermediate levels of the hierarchy. Therefore, the question is, given a product, what subset of the data should be used for training.

By mapping the characteristics of the different subsets to the performance of learning algorithms, a metalearning approach can be used in this problem.

However, another question is raised. For large datasets, the number of candidate subsets is very large, so it is not feasible to consider all of them in a metalearning approach. When available, such as in our case study, a hierarchy of the observations (e.g. products, product families, ...) can be used to reduce the number of candidate sets of training data to be considered. So, for product P_i the datasets considered contain the transactions concerning the levels $C_i^1, C_i^2, \ldots, C_i^k$ (where $k = 4$ in our case study). A metalearning approach is then used to choose, for a given product P_i, the level of the hierarchy, $j = 1, \ldots, k = 4$ that contains the data which is expected to generate the best model for that product.

The metadata consists of algorithm performance data and metafeatures. Algorithm performance data is obtained by running base-level experiments. In these experiments, an outlier detection method (LOF [4], [20]) at each level, for each product. The data from a month is used for training and the accuracy of the model is tested with the data from the next month. The performance of outlier detection for the product i on the level j is indicated by PI_i^j.

The metafeatures for each level of the product hierarchy are also calculated. mf_i^j is a vector containing the calculated metafeatures for the product i at the level j.

Having PI_i^j and mf_i^j enables us to create the metadata set. Each metadata row includes the product code (P_i), metafeatures for all levels of the product code (mf_i^j, $\forall j \in \{1, \ldots, k\}$), and the best performance obtained by outlier detection methods for that product among all the levels ($PI_{best\,i}$, see Eq. 1). Eq. 2 shows the format of the metadata sets in our experiment.

$$PI_{best\,i} = \max_j(PI_i^j) \ , \ \forall j \in \{1, \ldots, k\} \quad (1)$$

$$P_i = \{mf_i^1, mf_i^2, mf_i^3, mf_i^4, PI_{best\,i}\}, \\ \forall i \in \{1, \ldots, n_1\} \quad (2)$$

By applying a learning algorithm on the metadata, we obtain a meta-model that can be used to predict the best level of granularity for each product code.

A. Metafeatures

The extracted metafeatures noted above (mf_i^j), is describe briefly in this section. A comprehensive study was done by [21] for feature selection. Totally 15 measures were proposed to describe the characteristics of the data. Their effectiveness through extensive experiments were evaluated. A list of all features which are used for this study with a brief description is provided in the Table II. The detail description of each metafeature is explained in [21].

IV. Experiments and Results

Two learning algorithms were used in our experiments: Random Forest (RF) [22]–[24] and Decision Tree (DT) [25]–[27]. A DT is a tree-like structure, splitting a data set into branch-like segments, by

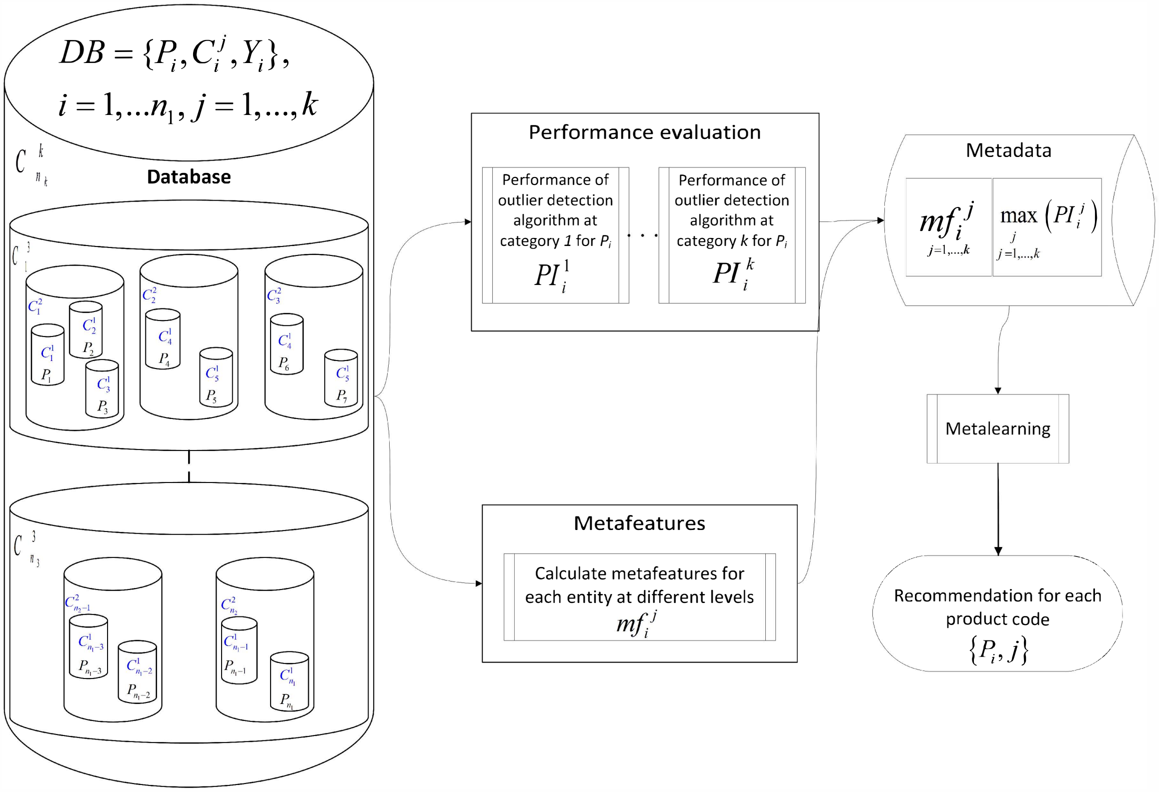

$DB = \{P_i, C_i^j, Y_i\}$,
$i = 1,...n_1, j = 1,...,k$
Database
Performance evaluation
Performance of outlier detection algorithm at category *1* for P_i
PI_i^1
Performance of outlier detection algorithm at category *k* for P_i
PI_i^k
Metadata
mf_i^j $j=1,...,k$
$\max_j (PI_i^j)$ $j=1,...,k$
Metalearning
Metafeatures
Calculate metafeatures for each entity at different levels
mf_i^j
Recommendation for each product code
$\{P_i, j\}$

Fig. 5. Methodology used for metalearning

TABLE II
EXTRACTED FEATURES USED IN METALEARNING

FEATURE NAME	DESCRIPTION
n.examples	Number of examples
n.attrs	Number of attributes
prop.symbolic.attrs	Proportion of symbolic attributes
prop.missing.values	Proportion of missing values
class.entropy	Class entropy
avg.mutual.information	Average mutual information
prop.h.outlier	Proportion of continuous attributes with outliers
avg.attr.entropy	Average attribute entropy
avg.symb.pair.mutual.infor	Average mutual information between pairs of symbolic attributes
avg.abs.attr.correlation	Average absolute correlation between continuous attributes
avg.skewness	Mean skewness of attributes
avg.abs.skewness	Mean absolute skewness of attributes
avg.kurtosis	Mean kurtosis of attributes
canonical.correlation.best.linear.combination	Canonical correlation of the best linear combination of attributes to distinguish between classes
relative.prop.best.linear.combination	Proportion of the total discrimination power explained by the best linear combination

evaluating a condition on an attribute in each node. The origin of the DT is a root node at the top of the tree. The decision taken after computing all attributes downward. The classification rules are the path from the root to the leaf. A RF is a construction of multiple DTs at the training time. Precisely, RF is a forest of uncorrelated trees such that each tree depends on the values of a random vector sampled and with the same distribution for all trees in the forest [23]. These algorithms are applied on the metadata and tried to predict the best level of granularity. There are four possible levels to apply the outlier detection algorithm: level 1 ($OD8$), level 2 ($OD6$), level 3 ($OD4$), and level 4($OD2$). While ”OD” stands for the outlier detection method and the suffixes (8, 6, 4, and 2) show the number of common digits among the product codes in the same level.

In our experiment, the two algorithms applied separately on the metadata and then the obtained model is applied on unseen observations. The accuracy of our model (See Section III) is compared with the baseline accuracy on different levels.

At the baseline, the LOF algorithm is applied on all the possible levels. Then the level with the best performance among all is selected as a *predicted level* for that product. Thereafter, by knowing the prediction of the level with the best performance for all product codes (the majority *predicted level* among all the product codes) and its actual value (the *predicted level* for a product code), the accuracy of the baseline is obtained.

The prediction accuracy of the model is our primary comparison measure. The accuracy of our model is compared with the accuracy of baseline.

The results are plotted in Fig. 6. The accuracy of applying the RF algorithm on the metadata is labeled as ”ML-RF”. The accuracy of DT is also plotted as ”ML-Tree”. The average accuracy is calculated in a month for all the product codes.

It is clear that the performance of the RF model is almost better than the performance of the baseline. The difference is significant between the RF and the baseline, almost two times. While the baseline itself is not accurate enough because the best level of granularity is different from one product code to another. Although the performance of the DT is not as good as the RF model, it shows better results than the baseline for September, 1998.

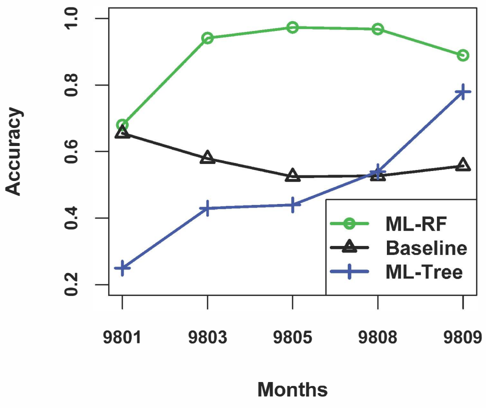


Fig. 6. Comparing the accuracy of metalearning approaches with the baseline

The overall results for all of the metrics mentioned in [4] (Recall, Accuracy, and Effort) are shown in Fig. 7. As long as the accuracy is acceptable, the prediction of each level can also be analyzed by other two metrics: Effort, and Recall. In this figure the Recall also is acceptable according to the satisfaction level of experts (more than 90%) although the recall is sustained a drop when the number of observations is not sufficient in the first level ($OD8$).

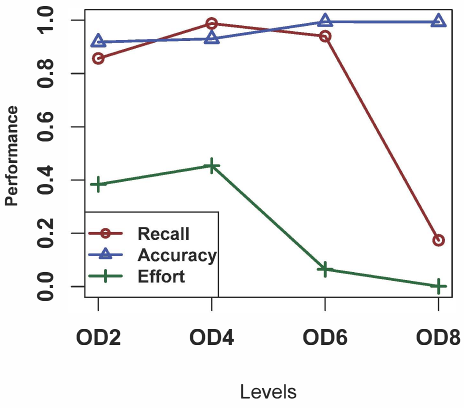


Fig. 7. Performance metrics using metalearning at different levels

The comparison of the Recall for metalearning approach and the baseline is also illustrated in Fig. 8. While the baseline sustains significantly more than 50% in recall degradation for higher levels, the RF incurs around 60% or more decrease in recall in the highest level. The DT algorithm performs variably on different levels. Nevertheless, for the lower levels, where the number of observations are very low, the baseline prevail the metalearning results.

Other metric which is noted previously is effort. Fig. 9 compares the effort obtained from metalearning approaches with the baseline. For the lower levels where the number of observations is low, the both metalearning algorithms outperform the baseline performance. This is due to the inaccurate model built on top of a few training examples at the baseline.

From our experiments, our metalearning model compares favorably with the baseline results across the different levels of granularity. However, the metalearning model do not always outperform the baseline. One of the reason for this problem can be the lack of enough observations for calculating the metafeatures. Therefore, the performance of the

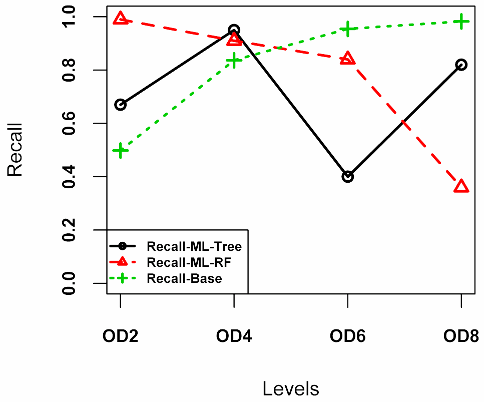


Fig. 8. Comparing metalearning with different algorithms with baseline: Recall

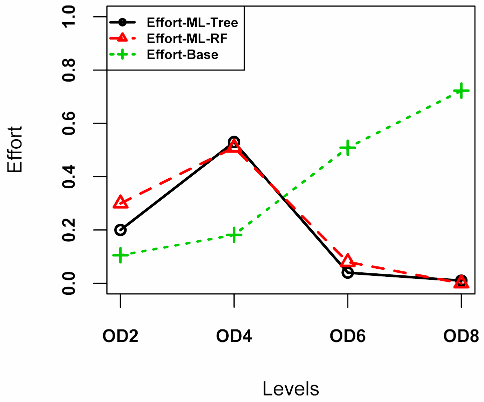


Fig. 9. Comparing metalearning with different algorithms with the baseline: Effort

metalearning model is highly depending on the metafeatures.

V. Discussion

As mentioned above, by increasing the available data, a single model cannot be useful for the whole data. Therefore, creating a specific model for subset of data can help improving the performance. The problem of splitting the data into different subsets is investigated in several ways. DW is a natural way to use the data dimension for splitting the data into subsets. This dimensions, equivalent to the different levels of hierarchy in the data, is used for prediction of the errors in the statistical data [4]. Their results showed that the best level of granularity differ from one product to another. So we use metalearning approaches to help predicting the best level of granularity for each individual product, separately.

For statistical data, there is no cost for aggregation of data at different levels. But in some case, like Vehicular networks data, there is a high cost to aggregate the data in higher levels. In such a network, the problem can be decision making for an individual car moving in a city. So it can use its data, data from its neighbors, higher level like Road Side Unit (RSU), or traffic center. As a result, there are involved cost like communication and time costs. The time can be crucial for safety application in VANETs. Considering these costs, different approaches for feature selection and metalearning is needed.

VI. Conclusion

Metalearning strategy aims at assisting the user in select appropriate learning algorithm for the particular data mining task. The problem of finding the error in the foreign trade statistics is important for providing accurate information. These errors may have a significant impact on the value of the statistics. In INE data, the products are organized in a 4-levels taxonomy. The objective of using metalearning is to recommend the best level for applying the outlier detection.

In this paper, we proposed a new model for metalearning to recommend the best level of granularity. The basic idea is to avoid applying the outlier detection at all levels of granularity to find the best one. Instead, the model recommends the best level of granularity for applying the outlier detection algorithm. The proposed model has been applied on the INE dataset for several months. 15 metafeatures are used for the purpose of data characterization.

Extensive experimental results have illustrated the improvement of accuracy of metalearning approaches comparing to the baseline. The results suggest that applying the RF algorithm on the metadata significantly improve the accuracy when compare to the baseline.

VII. Future Works

Due to different characteristics of datasets, one interesting work would be working with a dataset from VANETs. This dataset, as discussed in Section V, have lots of involved costs. Consequently, for selecting the best level of granularity, these costs also should be taken into account.

Feature selection is also another absorbing future work that can influence the performance of the metalearning approaches. Therefore, depending on the problem, different features should be selected.

Acknowledgment

This work is financed by the ERDF - European Regional Development Fund through the COMPETE programme (operational programme for competitiveness) within project GNOSIS, cf. ”FCOMP-01-0202-FEDER-038987”. It is also funded by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT) through projects ”NORTE-07-0124-FEDER-000057” and ”NORTE-07-0124-FEDER-000059”. The work is also financed by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projects ”FCOMP-01-0124-FEDER-037281” and ”SFRH/BD/71438/2010”.

References

[1] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, *{CRISP}-{DM} 1.0: Step-by-Step Data Mining Guide*, SPSS, 2000.

[2] D. Wolpert and W. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[3] F. Soulié-Fogelman, “Data Mining in the real world: What do we need and what do we have?” in *Proceedings of the KDD Workshop on Data Mining for Business Applications*, R. Ghani and C. Soares, Eds., 2006, pp. 44–48.

[4] M. Nozari Zarmehri and C. Soares, “Improving Data Mining Results by taking Advantage of the Data Warehouse Dimensions: A Case Study in Outlier Detection,” in *Encontro Nacional de Inteligência Artificial e Computacional.* São Carlos, Brazil: UFMG, LBD, 2014.

[5] R. Vilalta and Y. Drissi, “A perspective view and survey of meta-learning,” *Artificial Intelligence Review*, no. 1997, pp. 77–95, 2002.

[6] C. Soares, P. Brazdil, J. Costa, V. Cortez, and A. Carvalho, “Error Detection in Foreign Trade Data using Statistical and Machine Learning Algorithms,” in *Proceedings of the 3rd International Conference and Exhibition on the Practical Application of Knowledge Discovery and Data Mining*, N. Mackin, Ed., London, UK, 1999, pp. 183–188.

[7] P. Chan and S. J. Stolfo, “Experiments on multistrategy learning by meta-learning,” in *In Proc. Second Intl. Conference on Info. and Knowledge Mgmt*, 1993, pp. 314–323.

[8] P. Domingos, “Knowledge discovery via multiple models,” *Intelligent Data Analysis*, vol. 2, no. 3, pp. 187–202, 1998.

[9] K. ALEXANDROS and H. MELANIE, “Model selection via meta-learning: a comparative study,” *International Journal on Artificial Intelligence Tools*, vol. 10, no. 04, pp. 525–554, 2001.

[10] N. Schweighofer and K. Doya, “Meta-learning in reinforcement learning,” *Neural Networks*, vol. 16, no. 1, pp. 5–9, 2003.

[11] N. Jankowski, W. Duch, and K. Grabczewski, *Meta-learning in computational intelligence.* Springer Science & Business Media, 2011, vol. 358.

[12] O. Maimon and L. Rokach, *Data mining and knowledge discovery handbook.* Springer, 2005, vol. 2.

[13] L. Rendell and H. Cho, “Empirical learning as a function of concept character,” *Machine Learning*, vol. 5, no. 3, pp. 267–298, Aug. 1990.

[14] D. W. Aha, “Generalizing from Case Studies: A Case Study,” in *Proceedings of the Ninth International Workshop on Machine Learning*, ser. ML92. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992, pp. 1–10.

[15] J. R. Rice, “The algorithm selection problem,” ser. Advances in Computers, M. Rubinoff and M. C. Yovits, Eds. Elsevier, 1976, vol. 15, pp. 65 – 118.

[16] P. Brazdil, C. Soares, and J. D. Costa, “Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results,” *Machine Learning*, pp. 251–277, 2003.

[17] P. Brazdil, C. Giraud-carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*, ser. Cognitive Technologies, C. Sammut and G. I. Webb, Eds. Springer, 2009.

[18] L. Torgo and C. Soares, “Resource-bounded Outlier Detection Using Clustering Methods,” in *Data Mining for Business Applications*, ser. Frontiers in Artificial Intelligence and Applications, R. G. Carlos Soares, Ed. IOS Press, 2010, pp. 84–98.

[19] S. Ali and K. A. Smith, “Kernal Width Selection for SVM Classification,” *International Journal of Data Warehousing and Mining*, vol. 1, no. 4, pp. 78–97, Jan. 2005.

[20] L. Torgo, *Data Mining with R, learning with case studies.* Chapman and Hall/CRC, 2010. [Online]. Available: http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR

[21] Y. Peng, P. A. Flach, C. Soares, and P. Brazdil, “Improved dataset characterisation for meta-learning,” in *Discovery Science.* Springer, 2002, pp. 141–152.

[22] Y. Amit and D. Geman, “Shape Quantization and Recognition with Randomized Trees,” *Neural Computation, Massachusetts Institute of Technology*, vol. 9, no. 7, pp. 1545–1588, Oct. 1997.

[23] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32.

[24] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[25] L. Olshen, C. J. Stone *et al.*, “Classification and regression trees,” *Wadsworth International Group*, vol. 93, no. 99, p. 101, 1984.

[26] S. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[27] B. Ripley, *tree: Classification and regression trees*, 2014, r package version 1.0-35. [Online]. Available: http://CRAN.R-project.org/package=tree