Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.225 – 232

RESEARCH ARTICLE

META-LEARNING FROM LARGE ATTRIBUTED AND LARGE SAMPLED DATASETS USING META FEATURES

Patel Jaiminee J[1], Mr. A.R.Kazi[2]

[123]Computer Department & Gujarat Technological University, Gujarat, India

[1] jaimini101089@gmail.com; [2] arcazi.ce@gmail.com

***Abstract*— Now a days large variety of data classifiers are available so the people who are not expert in characteristics of data and their distribution, do not know which data classification method should be used to obtain good classification results for their given dataset. For this reason, choosing suitable classifier for given dataset is an important task. Meta-learning is a process to find or extract meta-features (the features which describe data itself) from given dataset. In this paper, we include five different categories of metafeatures for their suitability in predicting classification accuracies of several widely used classifiers.**

***Keywords*— Metalearning; metafeatures; learning algorithms**

I. INTRODUCTION

Classification consists of assigning a class label to a set of unclassified cases. It is two step process I. Learning II. Classification. For Example Loan officer needs to analyze data to learn which loan applicants are “safe” and which are “risky”. Now a days many types of classification algorithms are available like bayesian classification (naïve bayes), tree base (REP tree Decision stump), rule base (Zeror, Decision table, PART). So difficult task is to find out best classification algorithm for given dataset. The actual performance of a classifier compared to alternatives always depends on the characteristics of the data and how well they satisfy the assumptions made by the classier.

The idea of meta-learning is to learn about the learning algorithms themselves, i.e. to predict how well a learning algorithm will perform on a given dataset. This prediction is based on extracting meta-features (these are features that describe the dataset itself). They are derived from different properties of dataset. There are five types of metafeatures groups: simple, statistical, information-theoretic, model-based, and landmarking meta-features. Simple meta-features are directly derived from the data, e.g. the number of samples, the number of attributes or the number of classes. Statistical features describe statistical properties of the data, e.g. the “peakedness" or the asymmetry of a probability distribution. Information-theoretic features are typically based on entropy measures. More recently proposed types of meta-features are landmarking features and model-based features. The landmarking approach utilizes simple and fast computable classification algorithms. These classifiers are applied on the dataset and the resulting performance values are used as meta-features of the dataset. The model-based approach creates a model from the data and uses its properties as feature values [4].

These features are used to train a metalearning model on training data (in this case one dataset corresponds to one training sample). Afterwards, this model is applied on the meta-features of a new dataset. The result is the prediction of the suitability or performance of one or more target classifiers. Especially for non-experts in pattern recognition, meta-learning might significantly reduce the development time of a pattern recognition system by decreasing the required level of expertise for choosing a suitable classifier for a given problem [4].

II. CHALLENGES IN SELECTION OF CLASSIFIER

There are many classifier or classification algorithms are used in classification process. So it is difficult task to predict which classifier gives best result on given types of data. Because it is not true that any one classifier is best for all types of data. So selection of classifier based on our dataset is our challenging task.

Below Table1 shows one experimental result in which different classification algorithms are applied on only one dataset. It shows that different classification algorithms give different accuracy in same dataset. However, the no-free-lunch theorem [15] tells us that there is no learning scheme that can be uniformly better than all other learning schemes for all problem instances. Hence, no universal recommendation can be made for arbitrary data.

The actual performance of a classifier compared to alternatives always depends on the characteristics of the data and how well they satisfy the assumptions made by the classifier.

Classifier	***Accuracy***
Naïve Bayes	***64.28%***
PART	***35.71%***
Decision Table	***57.14%***
Decision Stump	***28.57%***

Table 1: Result of different classification algorithms on Wheather_Forcasting dataset.

Classifier	***Accuracy***
Naïve Bayes	***70.83%***
PART	***83.33%***
Decision Table	***75.00%***
Decision Stump	***70.83%***

Table 2: Result of different classification algorithms on Contact_Lense dataset.

Table 1 shows result of different classification algorithms on same dataset that is Whether_Forecasting. On that dataset Naïve bayes algorithm give beast accuracy. Table 2 shows result of different classification algorithms on same dataset that is Contact_Lense[1]. On that dataset PART and REP Tree algorithm give best accuracy. So we can say that any one classifier does not give best result on all types of data. So choosing suitable classifier for given data set is an important task. Because non expert do not know which classifier should be used to achieve good result. Meta learning is one approach for it.

III.METALEARNING

Metalearning is the process of predict how learning algorithm perform well on given types of dataset. This process is based on extracting metafeatures form given dataset. Metafetures describe properties of dataset and these are used to train metalearning model on given dataset.

A. Process of Metalearning[4]

The goal of meta-learning is to predict the actual performance outcome for each considered classifier independently instead of predicting the best out of a pair or out of all classifiers (classification). Therefore, a separate regression model is trained for each algorithm. The knowledge of the meta-learner is derived from the training data, which comprises of meta-features of multiple datasets and a target variable. The actual performance values of the classification algorithm serve as this quantitative target variable in this context. Each dataset results in one instance in the training data described by its meta-features and the computed performance of the target classifier. Using these measures, a regression model is learned that describes the relation between the meta-features and the expected performance of datasets.

If the performance of a classifier is to be predicted for a new dataset, the meta-features of this dataset are computed first. Then, the previously learned regression model is applied on these feature values. The result is the predicted performance of the classifier for this new dataset. This procedure is illustrated in Figure 1.

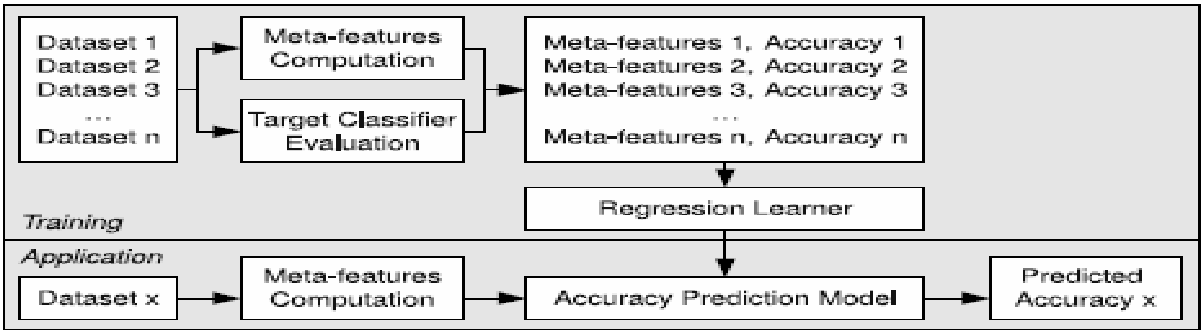

Figure. 1 Training using regression

For a recommendation or decision about which algorithm should be used, the results of the different regression models have to be compared. Since the predictions are quantitative values, multiple algorithms may be recommended if two or more predicted accuracies are close

A possible drawback of the regression approach is that for each target algorithm, a separate regression model has to be trained. This can be time consuming, especially when an exhaustive parameter optimization for the meta-model is performed. On the other hand, independent models also simplify adding or removing target classifiers. Additionally, the models for different target classifiers can be trained and evaluated independently. In particular, a different regression algorithm, different parameter values or different meta-features can be used.

B. Types of Metafeatures [4, 14]

Simple meta-features: number of samples, number of classes, number of attributes, number of nominal attributes, number of numerical attributes, ratio of nominal attributes, ratio of numerical attributes, dimensionality (number of attributes divided by number of samples)

Statistical meta-features: kurtosis, skewness, canonical discriminant correlation (cancor1), first normalized eigenvalues of canonical discriminant matrix (fract1), absolute correlation

Information-theoretic meta-features: normalized class entropy, normalized attribute entropy, joint entropy, mutual information, noise-signal-ratio, equivalent number of attributes

Model-based meta-features: For these features, a Decision Tree is trained without pruning. Different properties of this tree are used as feature values: number of leaves, number of nodes, nodes per attribute, nodes per sample, leaf corroboration. Additionally, the minimum, maximum, mean value and the standard deviation of the following measures are used: length of a branch (min-, max-, mean-, devBranch), number of nodes in a level (min- , max-, mean-, devLevel), number of occurrences of attributes in a split (min-, max-, mean-, devAtt)

Landmarking meta-features: The accuracy values of the following simple learners are used: Naive Bayes, Linear Discriminate Analysis, One-Nearest Neighbor, Decision Node, Random Node, Worst Node, Average Node.

C. Types of Learning Algorithms

There are many learning algorithms are available now a day like bayesian classification (naïve bayes), tree base (REP tree Decision stump), rule base (If –then rules, Zeror, Decision table, PART).

i) Bayesian based[2]:

Naive bias:

Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

P(c|x) is the posterior probability of class (target) given predictor (attribute).
P(c) is the prior probability of class.
P(x|c) is the likelihood which is the probability of predictor given class.
P(x) is the prior probability of predictor.

ii) Tree based[2]:

Decision Tree:

Decision tree is a classifier in the form of a tree structure
Decision node: specifies a test on a single attribute
Leaf node: indicates the value of the target attribute
Arc/edge: split of one attribute
Path: a disjunction of test to make the final decision
Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

Example:

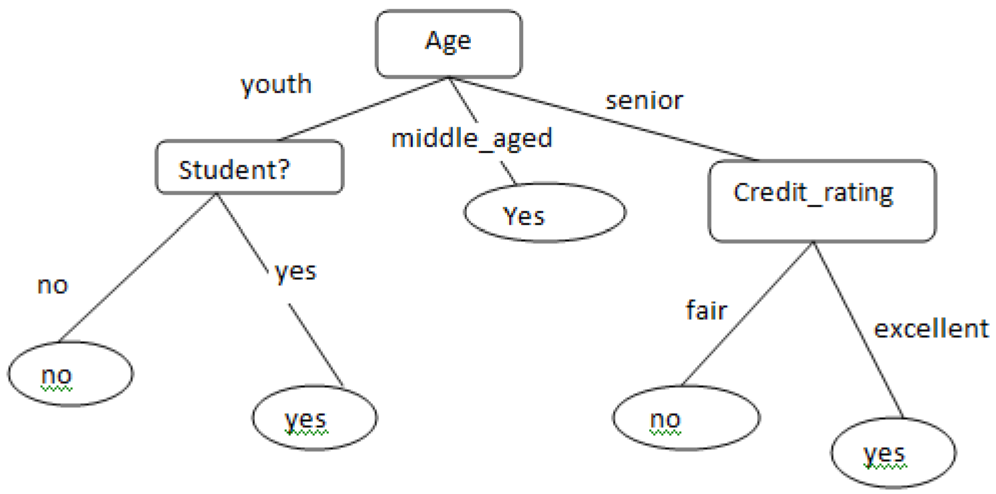

Figure.2. example of decision tree

iii) Rule Based[3]:

Using IF-THEN rules for classification:
Rules are good way for knowledge representation. A rule based classifier use set of IF_THEN rules for classification An IF-THEN rules has following form

IF condition THEN conclusion

Example:

R1: IF age=youth AND student=yes THEN buys_computer=yes.

The “IF” part of rule is known as rule antecedent or precondition. The “THEN” part is known as rule consequent. If the condition in a rule antecedent holds true for a given tuple, we can say that the rule antecedent is satisfied and that the rule cover the tuple.

Rules are assessed by its coverage and accuracy. Given a tuple, X, from a class labeled dataset, D, let Ncovers be the number of tuples covered by R; Ncorrect be the number of tuples covered by R and |D| be the number of tuples in D. So coverage and accuracy can be defined as

Coverage = Ncovers/|D|
Accuracy = Ncorrect/Ncovers

IV. PROPOSED WORK

Below figure shows preposed work that is used to find best classifier for given types of dataset using metalearning process

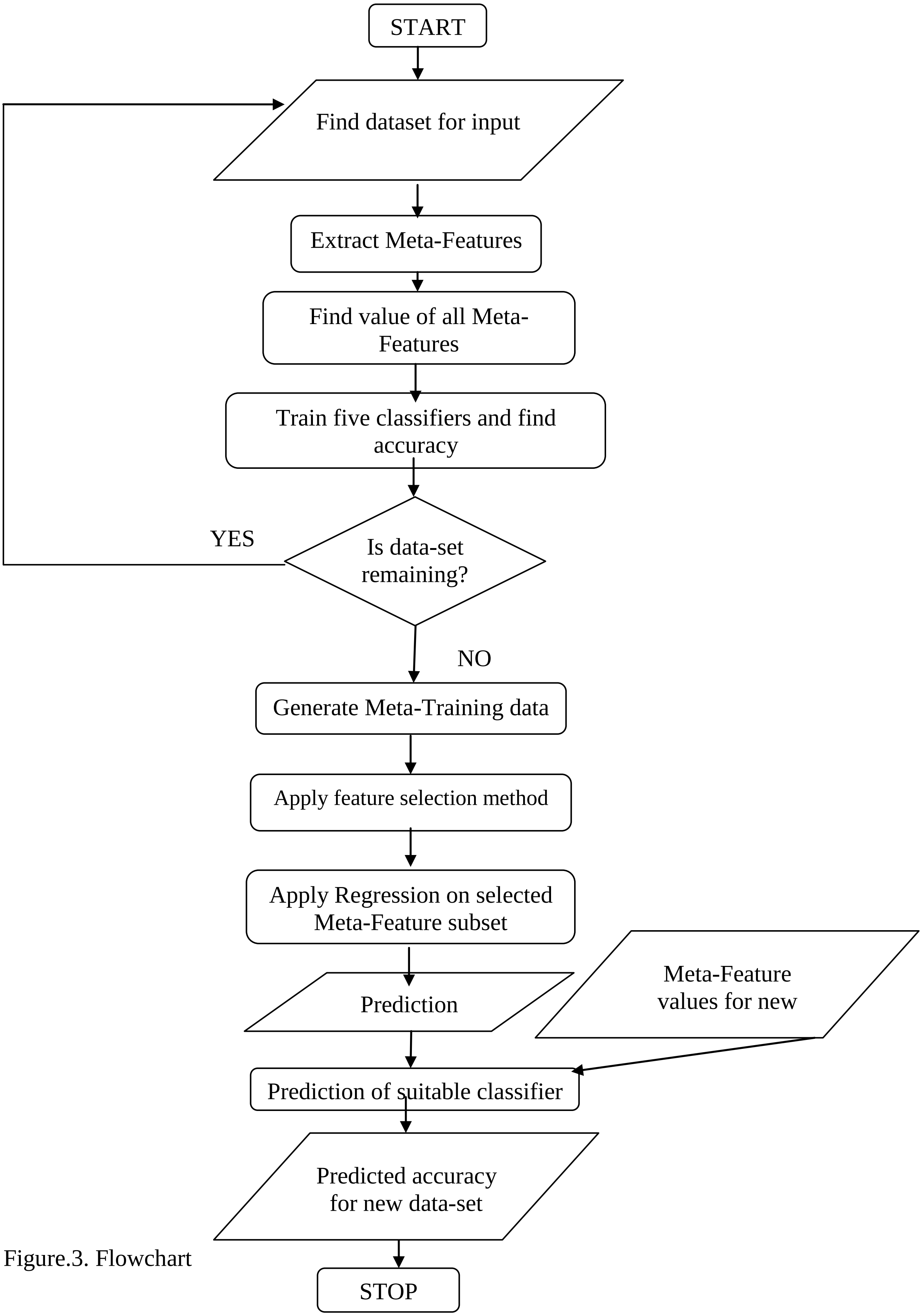

Figure.3. Flowchart

A. Find dataset for input:

In this step we have to find out dataset with different categories like dataset with more attributes, dataset with more samples, dataset with both more attributes and more samples, dataset with only nominal attributes, dataset with only numerical attributes, and dataset with both numerical and nominal attributes etc.

B. Extract metafeatures:

In this step from these datasets we have to find out different types of metafeatures like simple, statistical, information based, model based, and landmarking.

C. Find values of all metafeatures:

Next step is to find out values of all the metafeatures. These values of metafeatures are used in process of metalearning.

D. Train five classifiers

There are many learning algorithms are available now a day like bayesian classification (naïve bayes), tree base (REP tree Decision stump), and rule base (If –then rules, Zeror, Decision table, PART). In this step we have to select classifier which we want to train. Apply these classifiers one by one to generated values of metafeatures and find out accuracy values of all that selected classifier on selected metafeatures. After this step Meta training data are available.

E. Apply feature selection method

This step is used to find out which combination of meta-features is better, so that instead of finding all meta-features find only those are used to predict the best classifier.

F. Apply regression on selected metafeatures

On output of above step apply regressions so rules are generated and these rules are used to predict classifier.

G. Prediction for new dataset

In last step select new dataset and apply on prediction model which generate selected metafetures and prediction model will predict best classifier for it and also generate predicted accuracy.

V. CONCLUSIONS

Selection of good classifier for given types of dataset is an important task. Metalearning process is use for this task. Metalearning process use metafeatures (describe properties of dataset) to train classifier. Five different categories of metafeatures shown in this paper-simple, statistical, information based, model based, landmarking metafeatures. This paper describe prediction model that is used to find out best classification algorithm. Prediction model use metalearning process that is based on metafeatures.

REFERENCES

[1] http://www.tutorialspoint.com/data_mining/dm_classification_prediction. htm

[2] http://www.cs.waikato.ac.nz/~ml/weka/downloading.html

[3] Jiawei Han, Data Mining :Concept and Techniques http://www.mis.boun.edu.tr/gulser/index_files/DM%20Concepts%20%26%20Techniques%20_%20Han%26Kamber.pdf

[4] Matthias Reif, Faisal Shafait, Markus Goldstein, Thomas M.zeuel, Andreas Dengel, "Automatic Classifier Selection for Non-Experts"

[5] Nikita Bhatt, Amit Thakkar, Amit Ganatra ," A Survey & Current Research Challenges in Meta Learning Approaches based on Dataset Characteristics", IJSCE, ISSN: 2231-2307, Volume-2, Issue-1, March 2012

[6] Matthias Reif, Faisal Shafait, Andreas Dengel, "Meta features: Providing Meta Learners More Information", German Research center for AI Trippstader Str.122,67633

[7] Bensusan, H., Giraud-Carrier, C," how landmark performances can describe tasks", 2000

[8] Frnkranz, J., Petrak, J.:" An evaluation of landmarking variants", 2001

[9] Pfahringer, B., Bensusan, H., Giraud-Carrier, C, "Meta-learning by landmarking various learning algorithms", 2000

[10] Engels, R., Theusinger, C." Using a data metric for preprocessing advice for data mining applications.", 1998

[11] Segrera S, Pinho J, Moreno, M, "Information-theoretic measures for meta-learning", Springer Berlin / Heidelberg (2008)

[12] SilviuCacoveanu, CameliaVidrighin, RodicaPotolea , "Evolution Meta-Learning Framework For automatic Classifier Selection",2005.

[13] C. Giraud-Carrier, R.Vilalta , P. Brazdil, "Introduction to the special issue on meta-learning?, Machine Learning", 2004.

[14] Nikita Bhatt, Amit Thakkar, Amit Ganatra."Ranking of Classifiers based on Dataset Characteristics using ActivMeta Learning", International Journal of Computer Applications ,May-2013

[15] Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Comput. 8(7), 1341{1390 (1996)