China-USA Business Review, ISSN 1537-1514
March 2014, Vol. 13, No. 3, 199-205

Automatic Selection of Classification Algorithms for Non-Experts Using Meta-Features

Munehiro Nakamura
Kanazawa Institute of Technology, Kanazawa, Japan
Atsushi Otsuka, Haruhiko Kimura
Kanazawa University, Kanazawa, Japan

With the arrival of big-data society, methods for classifying real-world problems have attracted much attention for researchers and developers in various fields. In recent years, much effort has been devoted for improving performances of classification algorithms by adding functions or modifying their weaknesses. However, since a large variety of classification algorithms has been available, it is difficult for non-experts to find classification algorithms that achieve good results on a given data set. Therefore, if there is a system which automatically selects the best classification algorithm for a given data set, non-experts would receive various benefits such as saving time and effort. This paper presents a system of predicting the best possible classification algorithm for a given data set with respect to the accuracy. To the best of our knowledge, this is the first approach focused on predicting the best one. The main target users of the proposed system are non-experts who do not have knowledge and experience in data mining. The proposed system utilizes useful meta-features selected from existing meta-features to increase the performance of the prediction. The feature selection is conducted by a wrapper approach with the genetic search algorithm. In the proposed system, *K*-nearest neighbor algorithm is used to learn the selected meta-features and build a classification model for predicting future data. Experiments using 58 real-world data sets show that the proposed system predicted the best classification algorithm with 60.34% accuracy from the top five in 30 classification algorithms.

Keywords: feature selection, wrapper method, meta-feature, classifier, *k*-nearest neighbor

Introduction

The recent development of information society has increased needs for machine learning methods. For example, in marketing Customer Relationship Management (CRO) system enables a specific service for each customer by analyzing a wide variety of data such as customers attributions and action histories. In this field, automatic classification of a given data set plays an important role in decision making.

Classification algorithms (generally called as classifiers) are divided into several categories such as function-based classifiers (e.g., support vector machine and neural network), tree-based classifiers (e.g., J48

Munehiro Nakamura, Ph.D., Department of Information Science, Kanazawa Institute of Technology.
Atsushi Otsuka, B.D., Department of Natural Science and Technology, Kanazawa University.
Haruhiko Kimura, Ph.D., Department of Natural Science and Technology, Kanazawa University.
Correspondence concerning this article should be addressed to Munehiro Nakamura, Ogigaoka, Nonoichi-shi, Ishikawa, 921-8501, Japan. E-mail: m-nakamrua@blitz.ec.t.kanazawa-u.ac.jp.

and random forest), distance-based classifiers (e.g., *k*-nearest neighbor and *k*-star algorithm), and Bayesian classifiers. All existing classifiers have pros and cons. For example, while Support Vector Machine (SVM) is known as a powerful classification algorithm for binary class problem, it often shows poor classification performance on class-imbalanced problem.

It has been reported that no classifiers are better than any other classifiers with respect to the average performance on a set of problems (David, 1996). For instance, we do not know which classifier achieves good results for a given data set without a prior analysis. In fact, finding the best possible classifier is a challenging task specifically for non-experts because it requires knowledge and experience in this field.

This paper presents a system of estimating the best classifier for a given data set with respect to the accuracy. In the proposed system, 54 meta-features in five categories are used for the estimation. Among the meta-features, useful ones are selected by a feature selection method. Experiments for 58 real-world data sets show that the proposed system selects the best classification algorithm with 60.34% accuracy among five standard classification algorithms.

Research Questions

Classification algorithms such as SVMs and neural networks have already shown a great deal of success in practical applications. In many decades, much effort has been devoted for improving performances of classification algorithms by adding functions or modifying their weaknesses. For example, while SVM is one of the most powerful classification algorithms for binary class problem, there are three major problems in SVM. First, SVM tends to be biased to the majority class. This means that SVM builds a classification model to classify the majority class samples while the other class samples tend to be incorrectly classified. Second, sometimes SVM does not work well on multi-class problems. Third, SVM takes much computation time than simple classification algorithms. To solve these three problems, many modified versions of SVM have been proposed. Similarly, all of the existing classification algorithms have pros and cons as described in the previous section.

While novel classification algorithms have been proposed in various fields such as machine learning, bioinformatics, and data mining, we do not know which classification algorithm achieves good results for a given data set without a prior analysis. The most simplest way to find that the best classification algorithm is to apply each algorithm to a given data set. However, this procedure takes significant computation time and effort depending on the volume of a given data set. For these reasons, if there is a system which automatically selects the best classification algorithm for a given data set, users specifically non-experts would gain various benefits such as saving time and effort.

Research Methods

Meta-features are often used to evaluate classification algorithms in the machine learning community. Currently, meta-feature is broadly distinguished into five categories, namely, simple, statistical, information-theoretic, model-based, and landmarking (Matthias, Faisal, Markus, Thomas, & Andreas, 2014). Simple is a set of basic features such as number of samples, number of features, number of classes, and number of dimensionality. Statistical meta-features are kurtosis, skewness, canonical discriminant correlation, and so on. Information-theoretic meta-features are mutual information, normalized attribute entropy, noise-signal-ratio, and so on. Model-based meta-features are obtained from decision tree that build a

classification model without pruning. Landmarking meta-features are obtained from simple classification algorithms such as Naive Bayes, linear discriminant analysis, one-nearest neighbor, decision node, random node, worst node, and average node.

There exist many meta-features proposed in literatures (Hilan & Alexandros, 2001; Pavel, Carlos, & Joaquim, 2003; Yonghong, Peter, Carlos, & Pavel, 2002). Faisal, Matthias, Christian, and Thomas (2010) and Sarah, Faisal, Matthias, and Markus (2010) proposed 54 meta-features including existing ones. Since the meta-features would include unnecessary ones that reduce the accuracy for the classifier selection, we propose a method of selecting useful meta-features from the meta-features. In this paper, the meta-features are selected by a feature selection method based on the wrapper approach proposed in the literature (Ron, 1997).

The proposed system uses meta-features for predicting the best possible classifier on a given data set. Figure 1 shows a flow of the proposed system. The system is divided into the user side and system side. In the system side, a set of data sets is given by the developer. In the user side, the user only needs to prepare a data set that consists of attributes and a class attribute as shown in the left side of Figure 1.

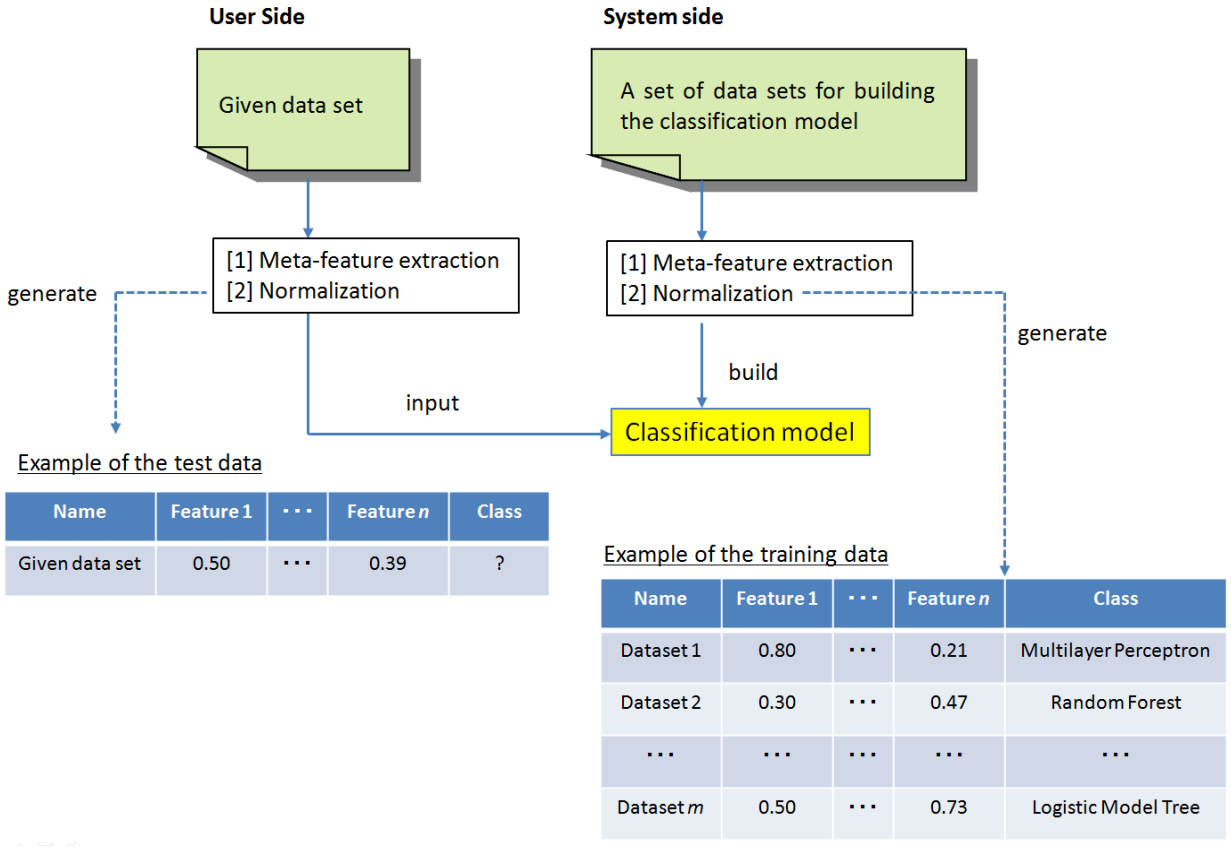

Figure 1. A flow of the proposed system.

In Figure 1, the system side builds a classification model that predicts the best possible classifier for a given data set. The table in the right side of the figure shows an example of the data for building the classification model. In the table, class represents the best classifier among a number of classifiers prior designated by the developer. The best classifier is determined by applying each of the classifiers to each of the

given data sets based on an evaluation value. There are broadly two types of evaluation values. One is error rate that represents how well a classification model classifies the instances used for building the model. Another is accuracy that represents how well a classification model classifies the instances in a given data set input by the user. From the view point of a user, the proposed system employs accuracy. As a representative evaluation value, the proposed system employs F_1 score which is defined as below:

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{1}$$

where *precision* and *recall* are defined respectively as below:

$$\text{precision} = \frac{tp}{tp + fp} \tag{2}$$

$$\text{recall} = \frac{tp}{tp + fn} \tag{3}$$

where *tp* is the number of correctly classified instances, *fp* is the number of incorrectly classified instances, *tn* is the number of correctly rejected instances, and *fn* is the number of incorrectly rejected instances.

Research Results

We have prepared 58 real-world data sets obtained from the UC Irvine Machine Learning Repository (Bache & Lichman, 2013) and benchmark data sets provided by Weka 3: Data Mining Software (Mark et al., 2009). For more details, Appendix A Table A1 shows all the data sets. The system was developed in the Eclipse platform. We have implemented 30 classifiers provided by Weka. Table 1 shows the 30 classifiers. After applying all the classifiers to the prepared data sets, we selected the top five classifiers regarding the number of the times that each classifier was selected as the best classifier. The five classifiers are MultilayerPerceptron, RandomForest (Leo, 2001), LMT (Niels, Mark, & Eibe, 2005), LADTree (Geoffrey, Bernhard, Richard, Eibe, & Mark, 2002), and FT (Joao, 2004). In this paper, the parameters for each classifier were set as default as well as most related works (Shawkat & Kate, 2006; Pavel et al., 2003; Alexandros & Melanie, 2001).

In the system construction, we chose *k*-Nearest Neighbor (*k*NN) algorithm to build the classification model where the number of *k* was configured as 1. As the feature selection method in the proposed system, we implemented the wrapper approach (Ron, 1997) with Genetic Search (GS) algorithm (David, 1989). Parameters for the GS algorithm were set as follows: crossoverProb = 0.6, maxGenerations = 20, mutationProb = 0.033, populationSize = 20, and reportFrequency = 20.

In order to evaluate the proposed system, we performed leave-one-out cross-validation on the 58 data sets. That is, one of the data sets was chosen as a given data set and the others were used to build the classification model, and this procedure was repeated until all the data sets were chosen as a given data set. Table 2 shows accuracy and F_1 score obtained in the evaluation experiment. In the table, the threshold was used for removing the meta-features that have less reliability than the threshold. The reliability was obtained from the GS algorithm. From the table, we can find that the proposed system achieved 60.3% accuracy, which is 15.5% higher than the accuracy without the feature selection.

Table 1

Thirty Classifiers in Six Categories Implemented in the Proposed System

Category	Classifier name	Category	Classifier name
Function	MultilayerPerceptron	Bayes	Bayes Net
	LibSVM		NaiveBayes
	SimpleLogistic		NaiveBayesUpdateable
	SMO	Misc	HyperPipes
Lazy	IB1		VFI
	IBk	Trees	DecisionStump
	Kstar		FT
	LWQ		J48
Rules	ConjunctiveRule		J48graft
	DecisionTable		LDATree
	JRip		LMT
	NNge		NBTree
	OneR		RandomForest
	PART		RandomTree
	ZeroR		REPTree

Table 2

Result of the Evaluation Experiment With Different Parameters for the Feature Selection Method

Number of meta-features	Threshold [%]	Accuracy [%]	F_1 score [%]
54	0	44.8	43.4
50	10	46.5	45.2
47	20	43.1	41.7
42	30	53.5	52.9
33	40	56.9	56.8
31	43	58.6	58.6
28	45	60.3	60.6
26	50	60.3	60.6
25	52	58.6	59.3
22	55	56.9	57.3
15	60	46.6	46.4
5	70	29.3	29.5

Table 3

Confusion Matrix for the Experimental Result in Table 2 When Threshold = 50

	Classified as				
Actual class	LDATree	FT	LMT	MP	RF
LDATree	4	0	2	1	2
FT	2	7	1	0	0
LMT	1	4	9	1	2
MP	2	0	0	10	0
RF	3	0	1	1	5

Table 3 shows the confusion matrix obtained in the experiment. Form the table, we can see that Multilayer Perceptron (MP) and FT achieved high accuracy, that is 83.3% and 70% respectively. On the other hand, many classifiers were incorrectly classified as LDATree. This suggests that the classification model was biased to

LDATree. To increase the classification performance, we need to reduce the bias by removing unnecessary instances that belong to LDATree. By the way, the interested readers can refer to Appendix A Table A2 that shows the meta-features selected by the GS algorithm when the threshold was set as 50.

Conclusions

This paper has presented a system of predicting the best possible classifier among various classifiers. Evaluation experiments showed that the feature selection works well on the proposed system. As future works, we would like to use optimized classification algorithms in the proposed system. In this case, since non-experts do not know how to optimize the best classifier, we need to predict the best parameters for the best classifier. We also would like to propose additional meta-feaures that increase the performance of the proposed system.

References

Alexandros, K., & Melanie, H. (2001). Feature selection for meta-learning. *Lecture in Notes in Computer Science, 2035,* 222-233.

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*. Retrieved from http://archive.ics.uci.edu/ml/

David, E. G. (1989). *Genetic algorithms in search, optimization and machine learning*. Boston, M.A.: Addison-Wesley Longman Publishing Co., Inc..

David, H. W. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computing, 8*(7), 1341-1390.

Faisal, S., Matthias, R., Christian, K., & Thomas, B. (2010). Pattern recognition engineering. Proceedings from *RapidMiner Community Meeting and Conference (RCOMM-10)*. Dortmund, Germany.

Geoffrey, H., Bernhard, P., Richard, K., Eibe, F., & Mark, H. (2002). Multiclass alternating decision trees. Proceedings from *ECML'02: The 13th European Conference on Machine Learning* (pp. 161-172). London, UK.

Hilan, B., & Alexandros, K. (2001). Estimating the predictive accuracy of a classifier. *Lecture Notes in Computer Science, 2167,* 25-36.

Joao, G. (2004). Functional trees. *Machine Learning, 55*(3), 219-250.

Leo, B. (2001). Random forests. *Machine Learning, 45*(1), 5-32.

Mark, H., Eibe, F., Geoffrey, H., Bernhard, P., Peter, R., & Ian, H. W. (2009). The WEKA data miming software: An update. *SIGKDD Explorations, 11*(1), 10-18.

Matthias, R., Faisal, S., Markus, G., Thomas, B., & Andreas, D. (2014). Automatic classifier selection for non-experts. *Pattern Analysis and Applications, 17*(1), 83-96.

Niels, L., Mark, H., & Eibe, F. (2005). Logistic model trees. *Machine Learning, 95*(1-2), 161-205.

Pavel, B. B., Carlos, S., & Joaquim, P. C. (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning, 50*(3), 251-277.

Sarah, D. A., Faisal, S., Matthias, R., & Markus, G. (2010). Landmarking for meta-learning using RapidMiner. Proceedings from *RapidMiner Community Meeting and Conference (RCOMM-10)*. Dortmund, Germany.

Shawkat, A., & Kate, A. S. (2006). On learning algorithm selection for classification. *Applied Soft Computing, 6*(2), 119-138.

Yonghong, P., Peter, A. F., Carlos, S., & Pavel, B. (2002). Improved dataset characterisation for meta-learning. *Lecture in Notes in Computer Science, 2534,* 193-208.

Appendix A

Table A1

Benchmark Data Sets Used to Evaluate the Proposed System

arrhythmia	hayes-roth_test	mfeat-morphological	sonar
audiology	hayes-roth_train	mfeat-zernike	soybean
balance-scale	heart-statlog	molecular-biology_promoters	spambase
breast-cancer	hepatitis	Mushroom	splice
breast-w	hypothyroid	Nursery	trains
bridges_version1	ionosphere	Optdigits	vehicle
car	iris.2D	page-blocks	vote
contact-lenses	iris	Pendigits	vowel
credit-a	kdd_synthetic_control	primary-tumor	waveform-5000
credit-g	kr-vs-kp	segment-challenge	weather.nominal
cylinder-bands	labor	segment-test	weather.numeric
dermatology	lung-cancer	shuttle-landing-control	wine
diabetes	lymph	Sick	zoo
ecoli	mfeat-fourier	solar-flare_1	
glass	mfeat-karhunen	solar-flare_2	

Table A2

Selected Meta-Features and Their Reliability for the Prediction

Feature name	Reliability [%]	Feature name	Reliability [%]
knn	91	max_symbols	60
dev_branch	89	numericalRate	60
average_node	84	min_attribute	58
min_branch	84	min_conditional_entropy	58
dev_mutual_information	73	Dimensionality	56
class_entropy	69	dev_level	55
min_mutual_information	67	max_branch	55
number of samples	67	n_leaves	55
mean_kurtosis	65	best_node	55
mean_symbols	65	max_level	53
min_entropy	62	mean_skewness	53
max_attribute	60	dev_symbols	53
max_entoropy	60	Numerical	51