Full Model Selection in the Space of Data Mining Operators

Quan Sun, Bernhard Pfahringer and Michael Mayo
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
{qs12, bernhard, mmayo}@cs.waikato.ac.nz

ABSTRACT

We propose a framework and a novel algorithm for the full model selection (FMS) problem. The proposed algorithm, combining both genetic algorithms (GA) and particle swarm optimization (PSO), is named GPS (which stands for **G**A-**P**SO-FM**S**), in which a GA is used for searching the optimal structure of a data mining solution, and PSO is used for searching the optimal parameter set for a particular structure instance. Given a classification or regression problem, GPS outputs a FMS solution as a directed acyclic graph consisting of diverse data mining operators that are applicable to the problem, including data cleansing, data sampling, feature transformation/selection and algorithm operators. The solution can also be represented graphically in a human readable form. Experimental results demonstrate the benefit of the algorithm.

Categories and Subject Descriptors

H.2.8 [**Database applications**]: Data Mining

General Terms

Algorithms, Experimentation

Keywords

Genetic Algorithm, Particle Swarm, Full Model Selection

1. DATA MINING IN THE DMO SPACE

We define a search space that consists of all data mining actions (operators) that are applicable to a given data set for a user-specified goal, such as a set of outlier filters, a set of feature selection methods, a set of data transformation techniques and a set of base learning algorithms. In this sense, we call the subject of interest "the space of data mining operators (DMO)", or simply "the DMO space". In this search space, a data mining solution is a directed acyclic graph (DAG) consisting of DMOs that are connected based on some relations. In this paper, we consider only cases where an exhaustive search is not feasible, and we are particularly interested in a search method that optimizes a problem by iteratively trying to improve a candidate FMS solution with regard to a given measure of quality. These methods are usually referred to as a "heuristic search", such as best-first search, local search (using neighborhood relation) and population-based evolutionary algorithms.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

2. THE GPS SEARCH STRATEGY

In this section, we propose a novel FMS algorithm for searching a FMS solution in the DMO space. The algorithm combines both genetic algorithm (GA) and particle swarm optimisation (PSO), in which a GA is used for searching the optimal template instance of a DMO template, and PSO is used for searching the optimal parameter set for a particular template instance. The proposed algorithm is called GPS (**G**A-**P**SO FM**S**). It can be seen as a realization and an application of the DMO framework. Before introducing the GPS algorithm, we first define a DMO template. Here, we assume a FMS solution consisting of five DMOs:

$DMO_{[data-cleansing]}$, $DMO_{[data-sampling]}$, $DMO_{[feature-transformation]}$, $DMO_{[feature-selection]}$, and $DMO_{[algorithm]}$.

Then, a DMO template for the FMS problem covered by GPS is defined as:

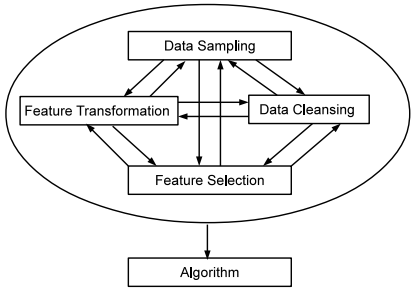
$$\begin{aligned} &solution \Longleftarrow \\ &DMO_{chain-search}(\\ &\quad DMO_{random-topology-search}(\\ &\quad DMO_{[data-cleansing]}, DMO_{[data-sampling]}, \\ &\quad DMO_{[feature-transformation]}, DMO_{[feature-selection]}), \\ &DMO_{[algorithm]}) \end{aligned} \quad (1)$$

Graphically, this template can be represented as Figure 1 (a). The four DMOs at the top can be performed in any order, then followed by an *Algorithm* DMO. Figure 1 (b) shows a solution instance of the DMO template. For each of the five DMOs we have defined in template (1), we have a pool of data mining tools available. For this research, the filters and algorithms in the Weka [2] machine learning package are used. Table 1 shows the tools that are included in the GPS system. The basic steps of the system are: for each GA iteration (generation), firstly a population of DMO template instances is randomly generated based on a predefined template (e.g., template (1) and Figure 1 (a)). Then, the placeholders of each template instance are randomly populated with the objects in the pools of DMOs (e.g., Figure 1 (b)). Then, PSO is used for searching an optimal parameter setting for each template instance (similar to the PSMS system [1]). The population of template instances is then sorted by their PSO-based evaluation scores. At the end of each GA iteration, typical GA operators, such as crossover and mutation, can be applied for generating new template instances which are used for replacing the template instances

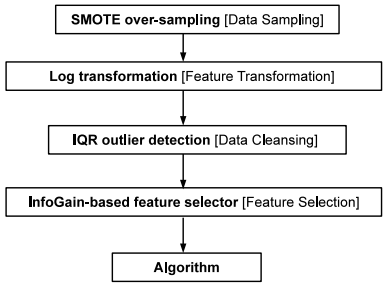
Table 1: Weka algorithms and filters that are used as the DMO objects in the GPS algorithm

Data Sampling	Data Cleansing	Feature Trans.	Feature Sel.
SMOTE oversampling, percent $p \in \{30, 50, 100, 200\}$	NumericCleaner	Normalize	CfsSubsetEval
Resample with replacement, percent $p \in \{30, 40, ..., 100\}$	RemoveUseless	Standardize	InfoGainAttributeEval
Resample without replacement, percent $p \in \{30, 40, ..., 90\}$	ReplaceMissingValues	Center	GainRatioAttributeEval
Do nothing	Do nothing	AddNoise	OneRAttributeEval
		Discretize	PrincipalComponents
		NominalToBinary	ChiSquaredAttributeEval
		NumericTransform	Do nothing
		Do nothing	

Algorithm	HyperParameters
Bagging with Random Tree	num.Bagging.Iterations $\in \{10, ..., 1000\}$, num.Atts.$\in \{1, ..., maxNumAtts\}$, depth.Tree $\in \{1, ...7\}$
Bagging with REPTree	num.Bagging.Iterations $\in \{10, ..., 1000\}$, num.Folds.$\in \{2, ..., 10\}$, depth.Tree $\in \{1, ...7\}$
AdaBoost.M1 with DecisionStump	num.Boosting.Iterations $\in \{10, ..., 1000\}$, useResample $\in \{True, False\}$
LogitBoost with DecisionStump	num.Boosting.Iterations $\in \{10, ..., 1000\}$, useResample $\in \{True, False\}$
Bagging with J48 Decision Tree	num.Bagging.Iterations $\in \{10, ..., 1000\}$, prune $\in \{True, False\}$, conf. $\in \{0.25, ..., 0.75\}$
RotationForest with REPTree	num.Iterations $\in \{10, ..., 1000\}$, removed. $\in \{20, ..., 50\}$, projection $\in \{PCA, RandomProjection\}$



(a) A graphical representation of the DMO template used by GPS



(b) A graphical representation of a DMO solution template instance

Figure 1: A full model defined by the GPS algorithm

with relatively *low* evaluation scores. The above procedure is repeated T times, where T is the number of GA generations. Finally, the template instance with the best evaluation score is returned as the GPS solution.

3. EXPERIMENTS

We experiment with ten real-world classification problems. To test the performance of the GPS algorithim, we implemented a variant of the PSMS system proposed in [1] with the DMO pools defined in Table 1. The PSMS system is an application of particle swarm optimisation (PSO) to the problem of full model selection for classification problems. In total, 3 feature transformation objects, 13 feature selection objects and 10 classifier objects are used in the PSMS system. A PSMS full model is defined as a 16-dimensional particle position [1]. The two systems are set to optimize the AUC performance and are tested under 30 different configurations. Figure 2 shows a summary of a comparison of AUC performance between GPS and PSMS under 30 different configurations.

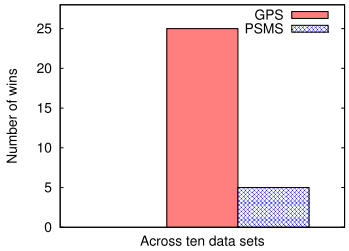


Figure 2: Comparison of predictive performance

4. CONCLUSIONS

In this paper, we proposed a framework and an algorithm for the full model selection problem in the DMO space. Our experiments show that the GPS algorithm outperforms the PSMS system, the state-of-the-art PSO-based FMS algorithm. In the longer version of this paper, we also theoretically examined the feasibility of using the divide and conquer idea for speeding up the GPS algorithm. Our results suggest that using the perfect binary tree as the internal tree structure is a viable approach when the training complexity of GPS is worse than linear. The success of the GPS algorithm on the diverse data sets selected for the study strongly suggests the applicability of the algorithm and the DMO framework to a wide range of FMS problems.

5. REFERENCES

[1] H. J. Escalante, M. Montes, and L. E. Sucar. Particle swarm model selection. *Journal of Machine Learning Research*, 10:405–440, 2009.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.