Towards a Framework for Designing Full Model Selection and Optimization Systems

Quan Sun, Bernhard Pfahringer and Michael Mayo

Machine Learning Group
Department of Computer Science
The University of Waikato, New Zealand

May 16, 2013

Motivations

End-users of ML/DM now have to face the new problem of how to choose a combination of data processing tools and algorithms.

This problem is usually termed the Full Model Selection problem.

Full Model Selection (FMS)

The FMS problem consists of the following[1]:

Given a pool of preprocessing methods, feature selection and learning algorithms, select the combination of these that obtains the lowest classification error for a given data set.

FMS tasks also include the selection of hyperparameters for the considered methods, resulting in a vast search space.

[1] Escalante et al., Particle Swarm Model Selection. JMLR (2009)

Data Mining Operators

We attempt to define a search space that consists of all data mining actions (operators) that are available to a given data set for a user-specified goal, such as:

- a set of outlier filters
- a set of feature generation, transformation and selection methods
- a set of learning algorithms
- ...

In this sense, we call the subject of interest "the space of data mining operators (DMO)", or simply "the DMO space".

The DMO Space

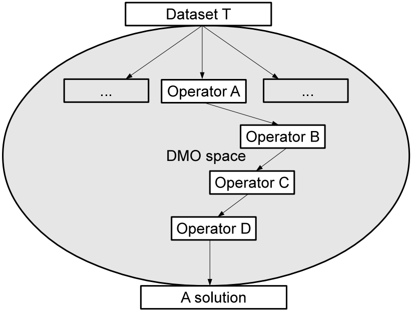


Figure: An illustration of the DMO space

Due to the resources at hand, usually we do not search in an infinite DMO space, and, moreover, we can make the DMO space a finite space by defining the DMOs that are to be included.

Related Work - The PSMS System

- The PSMS system (Escalante et al., JMLR, 2009) is an application of Particle Swarm Optimization (PSO) to the problem of FMS for binary classification problems.
- In total, 3 feature transformation objects, 13 feature selection objects and 10 classifier objects are used in the PSMS system.
- A full PSMS model is defined as a 16-dimensional particle position.

Related Work - The PSMS System

From the system architecture point of view, PSMS assumes a full model has three components: feature transformation, feature selection, and learning algorithm.

In the DMO framework, we can define the following DMO template for the search space covered by the PSMS system:

$$solution \Longleftarrow \quad DMO_{chain-search}(DMO_{random-topology-search}(DMO_{[feature-transformation]}, DMO_{[feature-selection]}), DMO_{[algorithm]})$$

The GPS Search Strategy

We propose a search strategy, which combines both a genetic algorithm (GA) and particle swarm optimization (PSO).

- GA is used for searching the optimal template structure of a DMO solution (structure space)
- PSO is used for searching the optimal parameter set for a particular solution instance (parameter space)

The algorithm is named GPS (**G**A-**P**SO FM**S**). It can be seen as a realization and an application of the DMO framework.

The GPS Search Strategy

We assume a FMS solution consists of five DMOs:
$DMO_{[data-cleansing]}$, $DMO_{[data-sampling]}$, $DMO_{[feature-transformation]}$, $DMO_{[feature-selection]}$, and $DMO_{[algorithm]}$.

A DMO template for the FMS problem covered by GPS is defined as:

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

The GPS Search Strategy

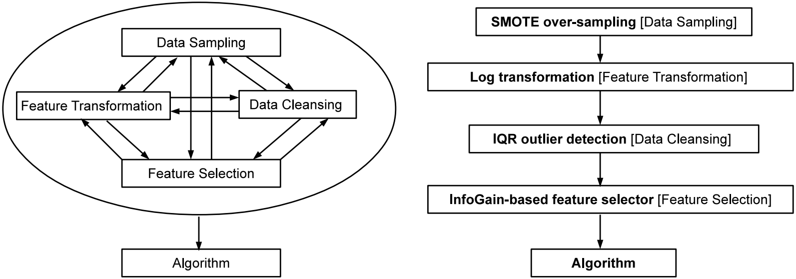


Figure: Left: a graphical representation of the DMO template used by GPS; Right: a DMO solution instance

The GPS Algorithm

Algorithm 1 Pseudocode of the GPS strategy for searching a FMS solution

procedure GPS(T,P,M,W,G)

Input:

T (number of generations for GA), P (population size for GA), M (number of evolutions for PSO), W (swarm size for PSO), G (goal metric)

Get P random template instances based on template (3).

Populate template instances with objects in the DMO pools (Table 2)

for $i \leftarrow 1$ to T **do**

Use a standard PSO procedure **PSO**(M,W,G,I) to search for the optimal parameters for each template instance I (optimising the goal metric G), and assign an evaluation score to each template instance I. This procedure is similar to the PSMS system [3].

Do *crossover* // single point crossover among the top 20% template instances.

Do *mutation* // randomly choose 30% template instances from the population, and randomly change one DMO in each template instance.

Replace the worst N template instances with the N new template instances generated in above two steps, here we use $N = (20\% + 30\%) \times P$.

$solution_{best} \leftarrow population_{best}$

end for

return $solution_{best}$

end procedure

Generate initial solution structures and instances

Optimize a solution instance

Generate new solution structures and instances

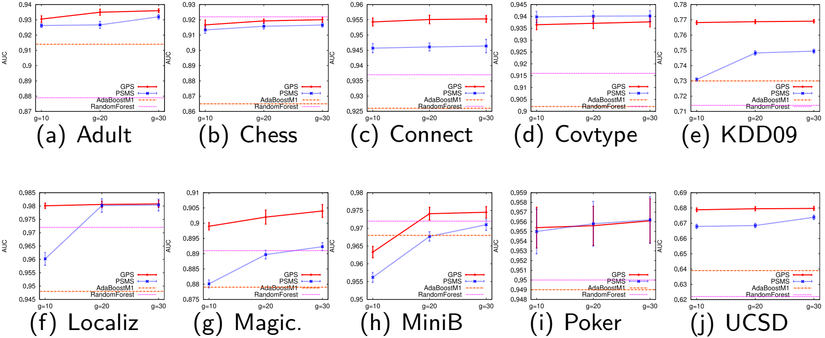
Experiments

1. Comparing GPS to PSMS and other algorithms
2. Speeding up the GPS system

Experiments: Datasets

Original data sets			Final binary data sets
Data set with release year	#Insts	Atts:Classes	Class distribution (#Insts)
Adult 96	48,842	14:2	23% vs 77% (10,000)
Chess 94	28,056	6:18	48% vs 52% (8,747)
Connect-4 95	67,557	42:3	26% vs 74% (10,000)
Covtype 98	581,012	54:7	43% vs 57% (10,000)
KDD09 Customer Churn 09	50,000	190:2	8% vs 92% (10,000)
Localization Person Activity 10	164,860	8:11	37% vs 63% (10,000)
MAGIC Gamma Telescope 07	19,020	11:2	35% vs 65% (10,000)
MiniBooNE Particle 10	130,065	50:2	28% vs 72% (10,000)
Poker Hand 07	1,025,010	11:10	45% vs 55% (10,000)
UCSD FICO Contest 10	130,475	334:2	9% vs 91% (10,000)

Experiment 1: Comparing GPS to PSMS



(a) Adult (b) Chess (c) Connect (d) Covtype (e) KDD09

(f) Localiz (g) Magic. (h) MiniB (i) Poker (j) UCSD

- GPS wins in 83% (25 out of 30) evaluation setups (benefit of combining GA and PSO for the FMS problem)
- The best performance of both GPS and PSMS outperform AdaBoost. and RF on 9 datasets (advantage of a full model over the single algorithm model)
- GPS outperforms the baseline algorithms with big margin on imbalanced datasets

Experiment 2: Speeding Up the GPS System

Some observations

- The training complexity of the GPS algorithm depends on the base learners found and evaluated during the search.
- The main cost for GPS is the cost for estimating a base learner's performance (e.g., cross-validation).

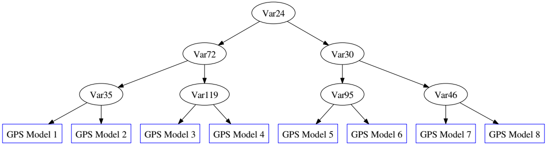
Users may have to wait for several hours, or even days on relatively large data sets. Therefore, in this work we also present a strategy for speeding up the GPS algorithm.

Experiment 2: Speeding Up the GPS System

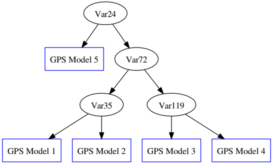
Full Model Trees (FMT):

Instead of training the GPS algorithm on the full training data, we build GPS models at the leaves of a tree structure.

Experiment 2: Speeding Up the GPS System



(k) FMT-perfect



(l) FMT-random

We compare GPS to Full Model Tree with two different tree structures, namely, the perfect binary tree and the random binary tree.

Experiment 2: Speeding Up the GPS System

Theoretically (theorem 1 and 2, pp. 9-10), the two Full Model Tree variants are faster than GPS-0 in the case that the (empirical) training complexity of GPS-0 is worse than linear.

Experiment 2: Speeding Up the GPS System

Table: Performance and runtime of the GPS and the Full Model Tree algorithms; A "$\ominus$" indicates that in terms of AUC, the GPS algorithm is significantly better than the respective algorithm; A "$\Diamond$" indicates that in terms of runtime, the GPS algorithm is significantly slower than the respective algorithm

Dataset	**GPS**	**FMT-perf.**	**FMT-rand.**	**GPS**	**FMT-perf.**	**FMT-rand.**
	AUC			Runtime (mins)		
Adult	0.94	0.93	0.93 $\ominus$	45 $\pm$ 6	37 $\pm$ 4 $\Diamond$	48 $\pm$ 11
Connect-4.	0.95	0.95	0.95 $\ominus$	91 $\pm$ 5	77 $\pm$ 9 $\Diamond$	74 $\pm$ 14 $\Diamond$
KDD Cup.	0.77	0.77	0.76 $\ominus$	178 $\pm$ 9	157 $\pm$ 11 $\Diamond$	189 $\pm$ 8
Mini.B.E.	0.98	0.98 $\ominus$	0.97 $\ominus$	124 $\pm$ 7	123 $\pm$ 9	135 $\pm$ 12
UCSD.	0.68	0.68 $\ominus$	0.67 $\ominus$	487 $\pm$ 16	417 $\pm$ 19 $\Diamond$	476 $\pm$ 17

Summary

- The DMO framework for designing new FMS algorithms
- The GPS algorithm for FMS
- Speeding up GPS with the perfect-binary-tree structure

A GUI for FMS

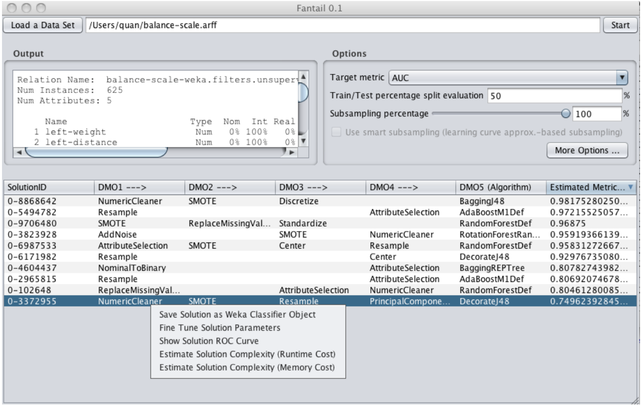


Figure: A proof-of-concept system based on the DMO framework using the GPS algorithm as the optimisation engine.

Questions

Thank you :-)