Adaptive Artificial Datasets to Discover the Effects of Domain Features for Classification Tasks

Syahaneim Marzukhi
Faculty of Engineering
Victoria University of Wellington, Wellington, NZ
|marzuksyah@myvuw.ac.nz

Will N. Browne
Faculty of Engineering
Victoria University of Wellington, Wellington, NZ
will.browne@vuw.ac.nz

Mengjie Zhang
Faculty of Engineering
Victoria University of Wellington, Wellington, NZ
mengjie.zhang@ecs.vuw.ac.nz

ABSTRACT

This paper described an automated pattern generator to generate various synthetic data sets for classification problems, where the problem's complexity can be manipulated autonomously. The Tabu Search technique has been applied in the pattern generator to discover the best combination of domain features in order to adjust the complexity levels of the problem. Experiments confirm that the pattern generator was able to tune the problem's complexity so that it can either increase or decrease the classification performance. The novel contributions in this work enable the effect of domain features that alter classification performance, to become human readable. This work provides a new method for generating artificial datasets at various levels of difficulty where the difficulty levels can be tuned autonomously.

Categories and Subject Descriptors

F.1.1 [**Models of Computation**]: Genetics-Based Machine Learning, Learning Classifier Systems

General Terms

Algorithm, Performance

Keywords

Pattern classification, Learning Classifier Systems

1. INTRODUCTION

The main goal of the work is to design a new pattern generation agent that utilizes *Tabu Search* technique to generate various synthetic datasets for classification with different levels of complexity based on the classification agent's ability to learn. The objectives here is for the pattern generation agent to autonomously tune and adjust the problem's complexity based on the classification agent's ability. Class balance, noise, number of instances, and many other parameters are to be set autonomously either to increase or decrease the problem's difficulties required.

2. METHODS

2.1 Two-Cornered LCSs

In the *Two-Cornered LCSs* [1], the system consists of two main agents, the pattern generation agent (i.e. the Sender(S)) and the pattern classification agent (i.e. the Receiver (R)). The pattern classification agent is developed based upon *Accuracy-based LCSs with real-value encoding (XCSR).* The algorithmic description for problem generation and classification is shown in Algorithm 1. S initializes a random *meta-problem* containing a list of parameters (i.e `<Fn Fc Fd Fi Fr Fan Fcn Fcbl Fcbd>`) for synthetic dataset generation. `Fn` is number of features, `Fc` is number of conjunction, `Fd` is number of disjunction, `Fi` is number of irrelevant features, `Fr` is number of redundant features, `Fan` and `Fcn` are level of noise that apply to action and condition, `Fcbl` percentage of class balance and `Fcbd` percentage of decision boundary. The dataset consists of a set of n instances, where each instance is defined by `F` features. Each instance n is created *on-the-fly* based on the specified problem within the interval of $[0, 1]$ and is labeled accordingly. Using Tabu Search technique, S searches for the best combination of `F` for the set task. Based on R's classification performance, S changes the combination of `F`. If S's objective is to increase R's performance, S attempt to find the best combination of features that can maximize R's classification performance (refer to Table 1).

3. EXPERIMENTS AND RESULTS

In *Experiment 1*, different combinations of the problem features (i.e. increasing and decreasing value of `Fan, Fcn and Fcbl`) on four problem domains (i.e. `Fn=2 to 5`) are enumerated to analyze R's performance with respect to those changes (Figure 1 and Figure 2). If there is no gradient in difficulty than it would be impossible for S either to make the problem 'harder' or 'easier' for R to learn.

In *Experiment 2*, TS was applied in S to search for the best combination of features in the problem with the objective to *maximize* R's performance (Figure 3). S was started with a predefined problem (i.e. `<Fc=1 Fan=50 Fcn=50 Fcbl=70 Fcbd=25>`) that was likely to be a 'hard' problem. Figure 4 shows R's classification performance when TS is applied in S to *minimize* R's performance. S was started with a predefined problem (i.e. `<Fc=1 Fan=5 Fcn=5 Fcbl=50 Fcbd=5>`) that was likely to be an 'easy' problem.

4. CONCLUSIONS AND FUTURE WORK

Generating datasets through specifying features has led to

Algorithm 1: Algorithmic description for problem generation and classification, Subscript R Receiver.

```
begin
    problem ← Sender : generate initial problem to Receiver.
    for (problem less than maximum problems) do
        for (instance less than maximum instance in dataset) do
            instance ← Sender : generate instance based on problem.
            Receiver ← pattern : perceive instance from Sender.
            GENERATE MATCH SET $[M]_R$ out of $[P]_R$ using instance.
            GENERATE PREDICTION ARRAY $PA_R$ out of $[M]_R$.
            class ← SELECT ACTION according to $PA_R$ .
            GENERATE ACTION SET $[A]_R$ out of $[M]_R$ according to class.
            Receiver : execute action class.
            reward ← Sender : Sender check class and send reward back to Receiver.
            if Receiver : end loop then
                prediction ← reward : update prediction with current reward.
                UPDATE SET $[A]_R$ using prediction possibly deletion in$[P]_R$.
                RUN GA in $[A]_R$ considering instance insertion in $[P]_R$.
            end
            if instance equal to maximum instance in dataset then
                classificationPerformance : calculate Receiver classification performance.
            end
        end
        if Sender : end loop then
            problem ← APPLY TS on problem based on classificationPerformance.
        end
    end
end
```

Table 1: **Changes in features F using Tabu Search.**

INITIAL SOLUTION: 2, 1, 0, 0, 0, 50, 50, 70, 25
INITIAL PERFORMANCE: 44.0
BEST PERFORMANCE: 96.0
BEST SOLUTION: 2, 1, 0, 0, 0, 1, 50, 70, 25

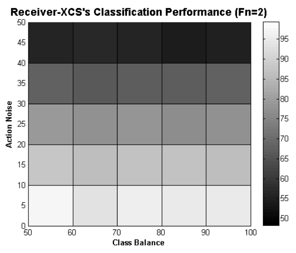

Figure 1: Trade-off surface of R's performance (average of R's classification performance from 30 runs, when Fn=2, while value of Fan and Fcbl is incremented by 5).

a system that can tune datasets to adjust the performance of the agents in a desired manner. An enumerative analysis of the potential datasets, identified the performance gradients but the agent identified useful gradients more efficiently. Important features, which control the ease of learning within the problem domain for the classification system, were identified. In future, the developed system will be used to modify the process of generating classification problems for *Three-Cornered LCSs Framework*, where the problem domain will tune autonomously depending on the two different agents' ability to learn (i.e. the Receiver and the Interceptor (I) which used different techniques of learning) [2].

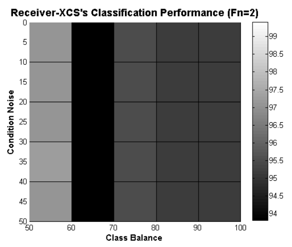

Figure 2: Trade-off surface of R's performance (average of R's classification performance from 30 runs, when Fn=2, while value of Fcn and Fcbl is incremented by 5).

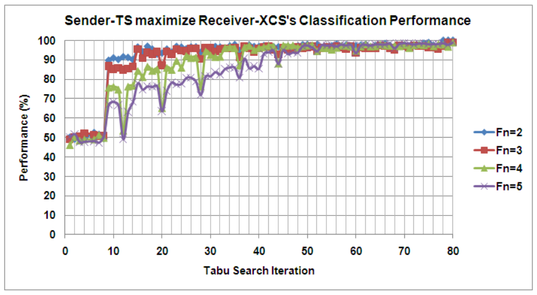

Figure 3: Average of R's classification performance from 30 runs on 4 problem domains, where TS is used in S for adjusting the difficulty levels (i.e. from 'hard' to 'easy').

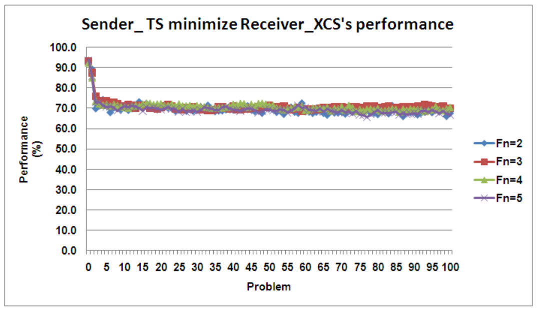

Figure 4: Average of R's classification performance from 30 runs on 4 problem domains, where TS is used in S for adjusting the difficulty levels (i.e. from 'easy' to 'hard').

5. REFERENCES

[1] S. Marzukhi, W. N. Browne, and M. Zhang. Two-cornered learning classifier systems for pattern generation and classification. In *The 12th Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 895–902. ACM, 2012.

[2] S. Wilson. Coevolution of Pattern Generators and Recognizers. *Lecture Notes in Computer Science (LNCS)*, Volume 6471/2010(1):38–46, 2010.