FS-XCS vs. GRD-XCS: An analysis using high-dimensional DNA microarray gene expression data sets

Mani Abedini[1], Michael Kirley[1], and Raymond Chiong[1,2]

[1] Department of Computing and Information Systems,
The University of Melbourne, Victoria 3010, Australia
{mabedini,mkirley,rchiong}@csse.unimelb.edu.au
[2] Faculty of Higher Education Lilydale,
Swinburne University of Technology, Victoria 3140, Australia
rchiong@swin.edu.au

Abstract. XCS, a Genetic Based Machine Learning model that combines reinforcement learning with evolutionary algorithms to evolve a population of classifiers in the form of condition-action rules, has been used successfully for many classification tasks. However, like many other machine learning algorithms, XCS becomes less effective when it is applied to high-dimensional data sets. In this paper, we present an analysis of two XCS extensions – FS-XCS and GRD-XCS – in an attempt to overcome the dimensionality issue. FS-XCS is a standard combination of a feature selection method and XCS. As for GRD-XCS, we use feature quality information to bias the evolutionary operators without removing any features from the data sets. Comprehensive numerical simulation experiments show that both approaches can effectively enhance the learning performance of XCS. While GRD-XCS has obtained significantly more accurate classification results than FS-XCS, the latter has produced much quicker execution time than the former.

1 Introduction

Classification tasks arise in many areas of science and engineering. One such example is disease classification based on gene expression profiles in bioinformatics. Gene expression profiles provide important insights into, and further our understanding of, biological processes. They are key tools used in medical diagnosis, treatment, and drug design [21]. From a clinical perspective, the classification of gene expression data is an important problem and a very active research area (see [3] for a review). DNA microarray technology has advanced a great deal in recent years. It is possible to simultaneously measure the expression levels of thousands of genes under particular experimental environments and conditions [22]. However, the number of samples tends to be much smaller than the number of genes (features)[1]. Consequently, the high dimensionality of a given

[1] Generally speaking, the number of samples must be larger than the number of features for good classification performance.

data set poses many statistical and analytical challenges, which often degrade the performance of classification methods used.

XCS – the eXtended Classifier System – is a Genetic Based Machine Learning (GBML) method that has been successfully used for a wide variety of classification applications, including medical data mining. XCS can learn from sample data in multiple iterative cycles. This is a great characteristic, but it also exhibits two common pitfalls that most classification methods have: sensitivity to data noise and "the curse of dimensionality" [22]. Both issues can easily jeopardise the learning process. A well-known solution is to use a cleansing stage. For example, feature selection/ranking techniques can remove unnecessary features from the data set. Reducing the dimensionality and removing noisy features can improve learning performance. Nevertheless, there exist data sets with highly co-expressed features, such as those studying Epistasis phenomena, that do not allow effective feature reduction. Examples of this include protein structure prediction and protein-protein interaction.

In this paper, we study two extensions of XCS inspired by feature selection techniques commonly used in machine learning: FS-XCS with effective feature reduction in place and GRD-XCS [1] that does not remove any features. The proposed model uses some prior knowledge, provided by a feature ranking method, to bias the discovery operators of XCS. A series of comprehensive numerical experiments on high-dimensional medical data sets has been conducted. The results of these simulation experiments suggest that both extensions can effectively enhance the XCS's learning performance. While GRD-XCS has performed significantly more accurate than FS-XCS, the latter is shown to have much quicker execution time compared to the former.

The remainder of this paper is organised as follows: Section 2 briefly describes some related work on XCS. In Section 3, we present the details of our proposed model. Section 4 discusses the experimental settings and results. Finally, we draw conclusion and highlight future possibilities in Section 5.

2 Related Work

GBML concerns applying evolutionary algorithms (EAs) to machine learning. EAs belong to the family of nature-inspired optimisation algorithms [9, 10]. As a manifestation of population-based, stochastic search algorithms that mimic natural evolution, EAs use genetic operators such as crossover and mutation for the search process to generate new solutions through a repeated application of variation and selection [11].

It is well documented in the evolutionary computation literature that the implementation of EA's genetic operators can influence the trajectory of the evolving population. However, there has been a paucity of studies focused specifically on the impact of selected evolutionary operator implementations in Learning Classifier Systems (LCSs), a type of GBML algorithm for rule induction. Here, we briefly describe some of the key studies related to LCSs in general and XCS – a Michigan-style LCS – in particular.

In one of the first studies focused on the rule discovery component specifically for XCS, Butz et al. [7] have shown that uniform crossover can ensure successful learning in many tasks. In subsequent work, Butz et al. [6] introduced an informed crossover operator, which extended the usual uniform operator such that exchanges of effective building blocks occurred. This approach helped to avoid the over-generalisation phenomena inherent in XCS [14]. In other work, Bacardit et al. [4] customised the GAssist crossover operator to switch between the standard crossover or a new simple crossover, SX. The SX operator uses a heuristic selection approach to take a minimum number of rules from the parents (more than two), which can obtain maximum accuracy. Morales-Ortigosa et al. [16] have also proposed a new XCS crossover operator, BLX, which allowed for the creation of multiple offspring with a diversity parameter to control differences between offspring and parents. In a more comprehensive overview paper, Morales-Ortigosa et al. [17] presented a systematic experimental analysis of the rule discovery component in LCSs. Subsequently, they developed crossover operators to enhance the discovery component based on evolution strategies with significant performance improvements.

Other work focused on biased evolutionary operators in LCSs include the work of Jos-Revuelta [18], who introduced a hybridised Genetic Algorithm-Tabu Search (GA-TS) method that employed modified mutation and crossover operators. Here, the operator probabilities were tuned by analysing all the fitness values of individuals during the evolution process. Wang et al. [20] used *Information Gain* as part of the fitness function in an EA. They reported improved results when comparing their model to other machine learning algorithms. Recently, Huerta et al. [5] combined *linear discriminant analysis* with a GA to evaluate the fitness of individuals and associated discriminate coefficients for crossover and mutation operators. Moore et al. [15] argued that biasing the initial population, based on expert knowledge preprocessing, would lead to improved performance of the evolutionary based model. In their approach, a statistical method, *Tuned ReliefF*, was used to determine the dependencies between features to seed the initial population. A modified fitness function and a new guided mutation operator based on features dependency was also introduced, leading to significantly improved performance.

3 The Model

We have designed and developed two extensions of XCS, both inspired by feature selection techniques commonly used in machine learning. The first extension, which we call FS-XCS, is a combination of a Feature Selection method and the original XCS. The second extension, which we call GRD-XCS, incorporates a probabilistically Guided Rule Discovery mechanism for FS-XCS. The motivation behind both extensions was to improve classifier performance (in terms of accuracy and execution time), especially for high-dimensional classification problems.

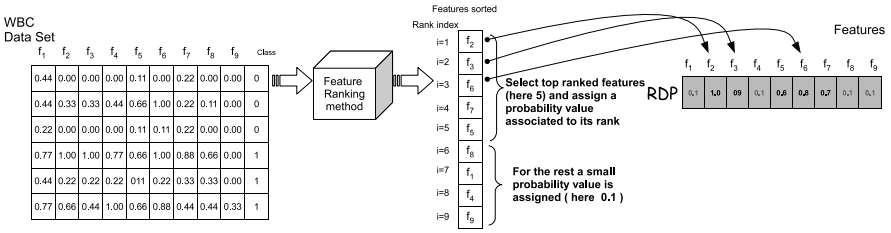


Fig. 1. Here, Information Gain has been used to rank the features. The top Ω features (in this example $\Omega = 5$) are selected and allocated relatively large probability values $\in [\gamma, 1]$. The RDP vector maintains these values. The probability value of the highest ranked feature is set to 1.0. Other features receive smaller probability values relative to their rank (in this example γ =0.5). Features that are not selected based on Information Gain are assigned a very small probability value (in this example $\xi = 0.1$).

FS-XCS uses feature ranking methods to reduce the dimension of a given data set before XCS starts to process the data set. It is a fairly straightforward hybrid approach. However, in GRD-XCS information gathered from feature ranking methods is used to build a probability model that biases the evolutionary operators of XCS. The feature ranking probability distribution values are recorded in a Rule Discovery Probability (RDP) vector. Each value of the RDP vector ($\in [0, 1.0]$) is associated with a corresponding feature. The RDP vector is then used to bias the feature-wise uniform *crossover*, *mutation*, and *don't care* operators, which are part of the XCS rule discovery component.

The actual values in the RDP vector are calculated based on the rank of the corresponding feature as described below:

$$RDP_i = \begin{cases} \frac{1-\gamma}{\Omega} \times (\Omega - i) + \gamma & \text{if } i \leq \Omega \\ \xi & \text{otherwise} \end{cases} \quad (1)$$

where i represents the rank index in ascending order for the selected top ranked features Ω. The probability values associated with the top ranked features would be some relatively large values ($\in [\gamma, 1]$) depending on the feature rank; for the others a very low probability value ξ is given. Thus, all features have a chance to participate in the rule discovery process. However, the Ω-top ranked features have a greater chance of being selected (see Figure 1 for an example).

GRD-XCS uses the probability values recorded in the RDP vector in the preprocessing phase to bias the evolutionary operators used in the rule discovery phase of XCS. The modified algorithms describing the *crossover*, *mutation* and *don't care* operators in GRD-XCS are very similar to standard XCS operators:

- GRD-XCS *crossover* operator: This is a hybrid uniform/n-point function. An additional check of each feature is carried out before the exchange of genetic material. If $Random[0, 1) < RDP[i]$ then feature i is swapped between the selected parents (Algorithm 1).

Algorithm 1 Guided Uniform Crossover algorithm

Require: Individuals: Cl_1,Cl_2 $\in[A]$, Probability Vector: RDP, Crossover Probability: χ

```
if Random[0,1) < χ then
  for i = 1 To SizeOf(Features) do
    if Random[0,1) < RDP[i] then
      SWAP(Cl1[i],Cl2[i])
    end if
  end for
end if
```

Algorithm 2 Guided Mutation algorithm

Require: Individual: Cl $\in[A]$, Probability Vector: RDP, Mutation Probability: μ

```
for i = 1 To SizeOf(Features) do
  if Random[0,1) < RDP[i] ×μ then
    Mutate(Cl[i])
  end if
end for
```

Algorithm 3 Guided Don't Care algorithm

Require: Individuals: Cl $\in[A]$, Probability Vector: RDP, Don't Care Probability: $P_{\#}$

```
for i = 1 To SizeOf(Features) do
  if Random[0,1) < (1 − RDP[i]) ×P# then
    P1[i] ← #
  end if
end for
```

- GRD-XCS *mutation* operator: It uses the RDP vector to determine if feature i is to undergo mutation; the base-line mutation probability is multiplied by RDP for each feature. Therefore, the mutation probability is not a uniform distribution anymore. The more informative features have better chance to be selected for mutation (Algorithm 2).
- GRD-XCS *don't care* operator: In this special mutation operator, the values in the RDP vector are used in the reverse order. That is, if feature i has been selected to be mutated and $Random[0,1) < (1 - RDP[i])$, then feature i is changed to # ("don't care") (see Algorithm 3).

The application of the RDP vector reduces the crossover and mutation probabilities for "uninformative" features. However, it increases the "don't care" operator probability for the same feature. Therefore, the more informative features should appear in rules more often than the "uninformative" ones.

4 Experiments and Results

We have conducted a series of independent experiments to compare the performance of FS-XCS and GRD-XCS. A suite of feature selection techniques have

Table 1. Data set details

Data Set	#Instances	#Features	Cross Validation	Reference
High-dimensional data sets (Microarray DNA gene expression)				
Breast cancer	22	3226	3	[13]
Colon cancer	62	2000	10	[2]
Leukemia cancer	72	7129	10	[12]
Prostate cancer	136	12600	10	[19]

been tested: Correlation based Feature Selection (CFS), Gain Ratio, Information Gain, One Rule, ReliefF and Support Vector Machine (SVM). Four DNA microarray gene expression data sets have been used in the experiments. The details of these data sets are reported in Table 1.

Our algorithms were implemented in C++, based on the Butz's XCS code[2]. The WEKA package (version 3.6.1)[3] was used for feature ranking. All experiments were performed on the *VPAC*[4] Tango Cluster server. Tango has 111 computing nodes. Each node is equipped with two 2.3 GHz AMD based quad core Opteron processors, 32GB of RAM and four 320GB hard drives. Tango's operating system is the Linux distribution CentOS (version 5).

4.1 Parameter settings

Default parameter values as recommended in [8] have largely been used to configure the underlying XCS model. For parameters specific to our proposed model, we have carried out a detailed analysis to determine the optimal settings. In particular, we have tested a range of Ω values $\Omega = 10$, 20, 32, 64, 128, 256 and population sizes $pop\_size = 500$, 1000, 2000, 5000. The analysis suggested that $\Omega = 20$ with a population size of 2000 can provide an acceptable accuracy level within reasonable execution time for FS-XCS. As for GRD-XCS, the setting of $\Omega = 128$ and $pop\_size = 500$ was found to have produced the best results. As such, these parameter values were used for the results presented in Section 4.3.

The limits used in probability value calculations in Equation 1 were set to $\gamma = 0.5$ and $\xi = 0.1$. In all experiments, the number of iterations was capped at 5000.

4.2 Evaluation

For each scenario (parameter value–data set combination), we performed N-fold cross validation experiments over 100 trials (see Table 1). The average accuracy

[2] The source code is available at the Illinois Genetic Algorithms Laboratory (IlliGAL) site http://www.illigal.org/

[3] Weka 3 is an open source data mining tool (in Java), with a collection of machine learning algorithms developed by the Machine Learning Group at University of Waikato – http://www.cs.waikato.ac.nz/ml/weka/

[4] Victorian Partnership for Advanced Computing: www.vpac.org

Table 2. Average accuracy (measured by AUC values) of the base-line XCS, FS-XCS and GRD-XCS on all selected microarray gene expression data sets.

base-line XCS	FS-XCS	GRD-XCS
0.77	0.88	0.98

values for specific parameter combinations have been reported using the Area Under the ROC Curve – the AUC value. The ROC curve is a graphical way to depict the tradeoff between the *True Positive rate* (TPR) on the Y axis and the *False Positive rate* (FPR) on the X axis. The AUC values obtained from the ROC graphs allow for easy comparison between two or more plots. Larger AUC values represent higher overall accuracy.

Appropriate statistical analyses using paired t-tests were conducted to determine whether there were statistically significant differences between particular scenarios in terms of both accuracy and execution time. Scatter plots of the observed and fitted values and Q-Q plots were used to verify normality assumptions. These statistical analyses were performed using the IBM SPSS Statistics (version 19) software.

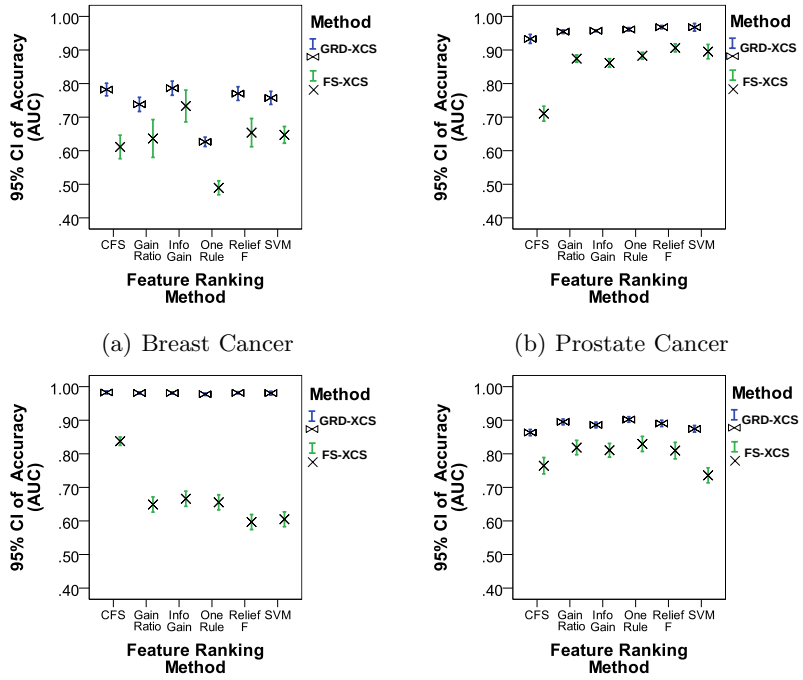
4.3 FS-XCS vs. GRD-XCS

To begin with, we have compared the average accuracy of FS-XCS and GRD-XCS with the base-line XCS (without feature selection) using all the aforementioned feature ranking methods on the microarray gene expression data sets listed in Table 1. The results, as shown in Table 2, indicate that GRD-XCS has an overall better accuracy than FS-XCS: the average FS-XCS accuracy using various feature selection techniques is 0.88 while the average accuracy of GRD-XCS using the same feature ranking methods is 0.98. Meanwhile, both FS-XCS and GRD-XCS are better than the base-line XCS – the latter has managed only an average accuracy of 0.77. For the rest of this section, we will focus on a detailed comparison between FS-XCS and GRD-XCS.

Figure 2 shows the AUC values of FS-XCS and GRD-XCS when different feature ranking methods are used. From the figure, it is clear that GRD-XCS is significantly more accurate than FS-XCS. The accuracy result of both FS-XCS and GRD-XCS for every feature ranking method, except Information Gain over the Breast Cancer data set, is significantly different ($p < 0.001$).

In Figure 3, FS-XCS is shown to be significantly faster than GRD-XCS ($p < 0.001$) in terms of execution time. This is much expected since FS-XCS works with only a fraction of the original data set size (i.e., 20 features) while GRD-XCS still accepts the entire data set with thousands of features. The only exception is when Gain Ratio has been applied over the Breast Cancer data set – in this case there is strong evidence that both FS-XCS and GRD-XCS have significantly equal average execution time ($p = 0.94$).

Figures 4 and 5 depict some general insight into the population diversity. In the majority of cases, GRD-XCS has less diversity.



(a) Breast Cancer

(b) Prostate Cancer

(c) Leukemia

(d) Colon Cancer

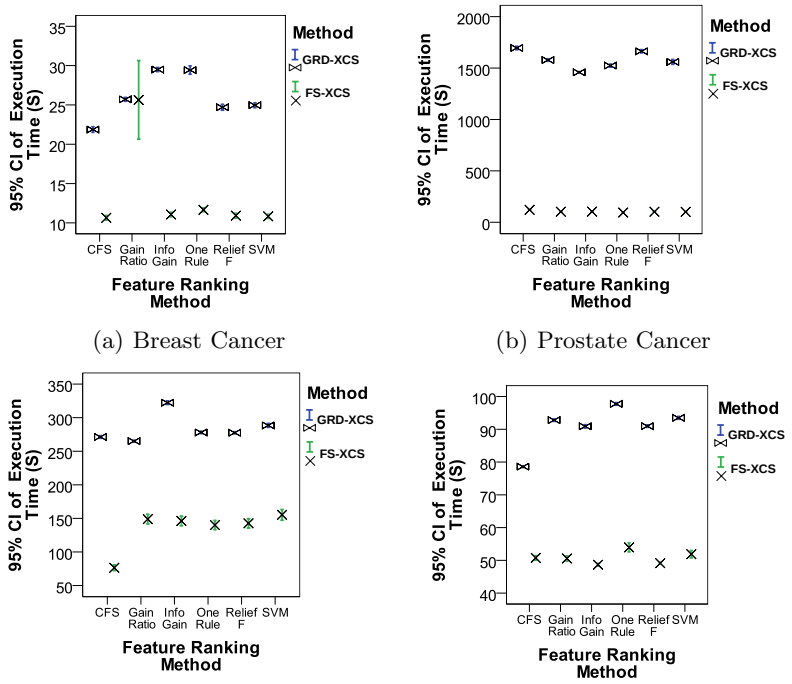
Fig. 2. The accuracy (AUC) of FS-XCS vs. GRD-XCS when various feature ranking methods are applied.

The average length of each classifier in GRD-XCS is almost always significantly smaller than FS-XCS ($p < 0.05$). The significant similar cases are Gain Ratio ($p = 0.80$) and ReliefF ($p = 0.26$) on the Prostate Cancer data set.

The average number of macro classifiers in GRD-XCS is significantly smaller than the average number of macro classifiers in FS-XCS. As can be seen in Figures 5(b) and (d), the difference is getting more obvious when the dimensionality increases (for Prostate Cancer and Colon Cancer). However, there is a different story for the Breast Cancer data set where the average number of macro classifiers in the GRD-XCS population is larger than FS-XCS. It would be a fair conclusion to say that GRD-XCS is exploring the solution space in a more focused manner than FS-XCS. In other words, the guided rule discovery approach forces the learning process to generate less diverse testing hypothesis; however this behaviour can evolve more accurate classifiers.

5 Conclusion and Future Work

In this paper, we have analysed the performance of FS-XCS and GRD-XCS based on some high-dimensional classification problems. Comprehensive numer-



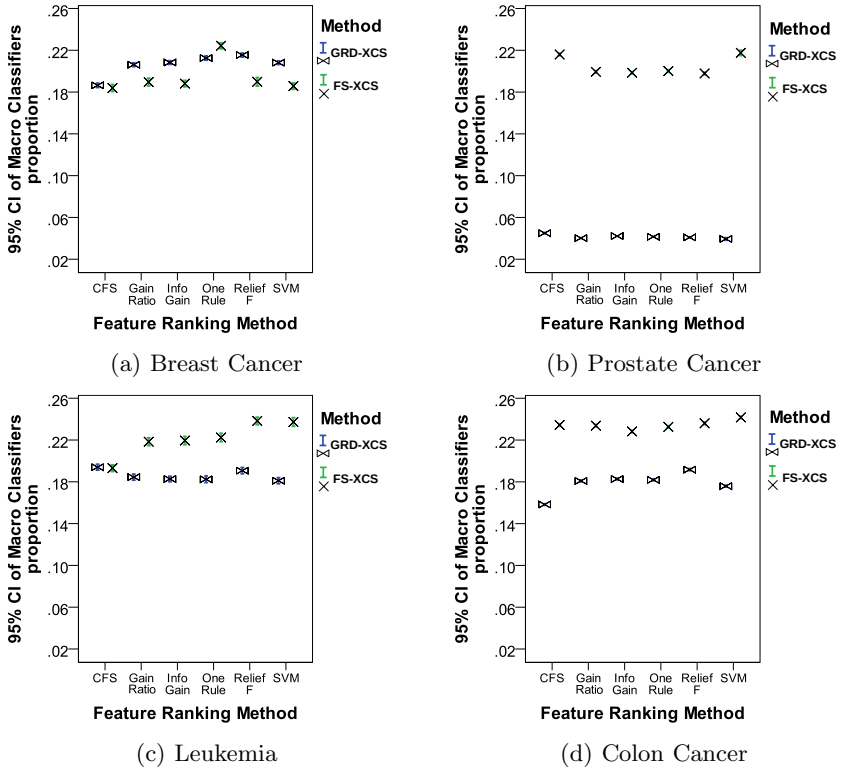
(a) Breast Cancer (b) Prostate Cancer

(c) Leukemia (d) Colon Cancer

Fig. 3. The execution time (in seconds) of FS-XCS vs. GRD-XCS when various feature ranking methods are applied.

ical simulations have established that GRD-XCS is significantly more accurate than FS-XCS in terms of classification results. On the other hand, FS-XCS is significantly faster than GRD-XCS in terms of execution time. The results of FS-XCS suggest that normally 20 top-ranked features would be enough to build a good classifier, although this classifier is significantly less accurate than the equivalent GRD-XCS model. Nevertheless, both models have performed better than the base-line XCS.

To sum up, using feature selection to highlight the more informative features and using them to guide the XCS rule discovery process is better than applying feature reduction approaches. This is mainly due to the fact that GRD-XCS can transform poor classifiers (created from the uninformative features) into highly accurate classifiers. From the empirical analysis presented it is clear that the performance of different feature selection techniques varies inevitably depending on the data set characteristic. Future work will therefore attempt to rectify this through the idea of ensemble learning. That is, we can build an ensemble classifier from multiple XCS based models (may it be FS-XCS or GRD-XCS). Each of these XCS cores can use a distinctive feature selection method. The



(a) Breast Cancer (b) Prostate Cancer

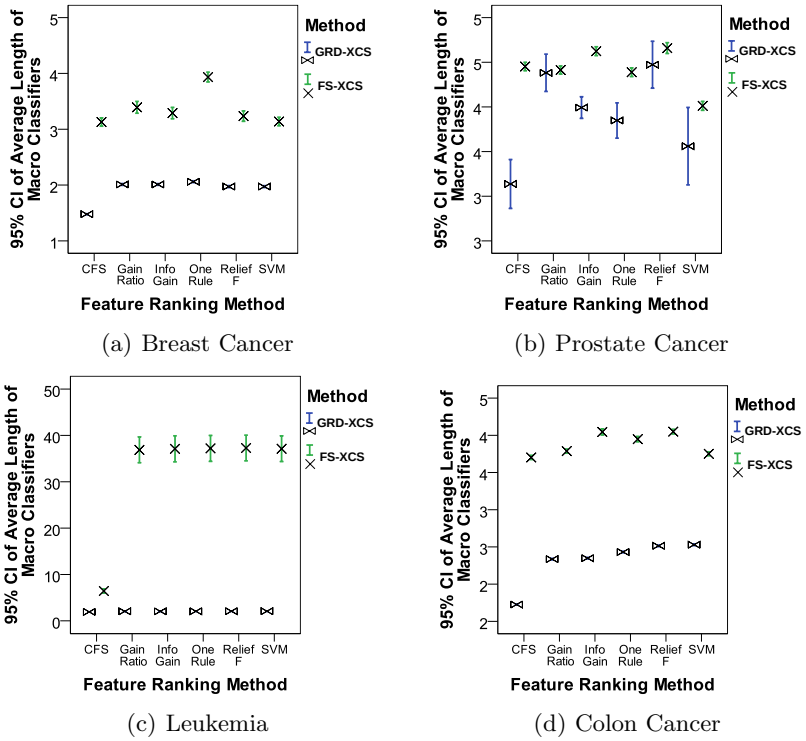
(c) Leukemia (d) Colon Cancer

Fig. 4. The proportion of macro classifiers to the population size of FS-XCS vs. GRD-XCS when various feature ranking methods are applied.

results of all XCS cores are then combined to form the ensemble result – for instance by using a majority voting technique.

References

1. M. Abedini and M. Kirley. An enhanced XCS rule discovery module using feature ranking. *International Journal of Machine Learning and Cybernetics*, 10.1007/s13042-012-0085-9, 2012.
2. U. Alon, N. Barkai, D. A. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. of the National Academy of Sciences of the USA*, 96:6745–6750, 1999.
3. M. H. Asyali, D. Colak, O. Demirkaya, and M. S. Inan. Gene expression profile classification: A review. *Current Bioinformatics*, 1(1):55–73, 2006.
4. J. Bacardit and N. Krasnogor. Smart crossover operator with multiple parents for a Pittsburgh learning classifier system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 1441–1448. ACM Press, 2006.



(a) Breast Cancer (b) Prostate Cancer

(c) Leukemia (d) Colon Cancer

Fig. 5. The average length of macro classifiers (rules) of FS-XCS vs. GRD-XCS when various feature ranking methods are applied.

5. E. Bonilla Huerta, J. C. Hernandez Hernandez, and L. A. Hernandez Montiel. A new combined filter-wrapper framework for gene subset selection with specialized genetic operators. In *Advances in Pattern Recognition*, volume 6256 of *Lecture Notes in Computer Science*, pages 250–259. Springer, 2010.
6. M. Butz, M. Pelikan, X. Lloral, and David E. Goldberg. Automated global structure extraction for effective local building block processing in XCS. *Evolutionary Computation*, 14(3):345–380, 2006.
7. M. V. Butz, D. E. Goldberg, and K. Tharakunnel. Analysis and improvement of fitness exploitation in XCS: Bounding models, tournament selection, and bilateral accuracy. *Evolutionary Computation*, 11(3):239–277, 2003.
8. M. V. Butz and S. W. Wilson. An algorithmic description of XCS. *Soft Computing*, 6(3–4):144–153, 2002.
9. R. Chiong, editor. *Nature-Inspired Algorithms for Optimisation*. Springer, 2009.
10. R. Chiong, F. Neri, and R. I. McKay. Nature that breeds solutions. In R. Chiong, editor, *Nature-Inspired Informatics for Intelligent Applications and Knowledge Discovery: Implications in Business, Science and Engineering*, chapter 1, pages 1–24. Information Science Reference, Hershey, PA, 2009.
11. R. Chiong, T. Weise, and Z. Michalewicz, editors. *Variants of Evolutionary Algorithms for Real-World Applications*. Springer, 2012.

12. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
13. I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O. P. Kallioniemi, B. Wilfond, A. Borg, and J. Trent. Gene-expression profiles in hereditary breast cancer. *The New England Journal of Medicine*, 344(8):539–548, 2001.
14. P. L. Lanzi. A study of the generalization capabilities of XCS. In Thomas Bäck, editor, *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 418–425. Morgan Kaufmann, 1997.
15. J. H. Moore and B. C. White. Exploiting expert knowledge in genetic programming for genome-wide genetic analysis. In *PPSN*, volume 4193 of *Lecture Notes in Computer Science*, pages 969–977. Springer, 2006.
16. S. Morales-Ortigosa, A. Orriols-Puig, and E. Bernadó-Mansilla. New crossover operator for evolutionary rule discovery in XCS. In *Proceedings of the 8th International Conference on Hybrid Intelligent Systems*, pages 867–872. IEEE Computer Society, 2008.
17. S. Morales-Ortigosa, A. Orriols-Puig, and E. Bernadó-Mansilla. Analysis and improvement of the genetic discovery component of XCS. *International Journal of Hybrid Intelligent Systems*, 6(2):81–95, 2009.
18. L. M. San Jose-Revuelta. *A Hybrid GA-TS Technique with Dynamic Operators and its Application to Channel Equalization and Fiber Tracking*. I-Tech Education and Publishing, 2008.
19. D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, and A. A. Renshaw. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.
20. P. Wang, T. Weise, and R. Chiong. Novel evolutionary algorithms for supervised classification problems: An experimental study. *Evolutionary Intelligence*, 4(1):3–16, 2011.
21. F.-X. Wu, W. J. Zhang, and A. J. Kusalik. On determination of minimum sample size for discovery of temporal gene expression patterns. In *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences*, pages 96–103, 2006.
22. Y. Zhang and J. C. Rajapakse. *Machine Learning in Bioinformatics*. Wiley Series in Bioinformatics. 1'st edition, 2008.