Ensemble Learning with Evolutionary Computation: Application to Feature Ranking

Kees Jong[1], Elena Marchiori[1], and Michèle Sebag[2]

[1] Department of Mathematics and Computer Science
Vrije Universiteit Amsterdam, The Netherlands
{cjong,elena}@cs.vu.nl
[2] Laboratoire de Recherche en Informatique, CNRS-INRIA
Université Paris-Sud Orsay, France
sebag@lri.fr

Abstract. Exploiting the diversity of hypotheses produced by evolutionary learning, a new ensemble approach for Feature Selection is presented, aggregating the feature rankings extracted from the hypotheses. A statistical model is devised to enable the direct evaluation of the approach; comparative experimental results show its good behavior on non-linear concepts when the features outnumber the examples.

1 Introduction

Among the major advances of Machine Learning in the last decade is Ensemble Learning [6, 3, 5], based on the vote of hypotheses extracted from the dataset along independent (bagging [3]) or iterative (boosting [6]) procedures. The variance of the learning error is reduced through the vote mechanism, thereby improving the global learning performance.

Indeed, stochastic algorithms and specifically evolutionary learners are natural candidates for ensemble learning approaches; they can provide collections of hypotheses, extracted using many runs or within a single run using diversity enforcing heuristics; see among many others [8, 12, 19].

In this paper, a new ensemble approach aimed at Feature Selection and Feature Ranking, referred to as Ensemble Feature Ranking, is presented, and implemented using evolutionary learning.

Feature Selection (FS) is commonly viewed as a major bottleneck of Supervised Machine Learning and Data Mining [13, 9]. For the sake of the learning performance, it is highly desirable to discard irrelevant features prior to learning, especially when the features significantly outnumber the examples. FS can be formalized as a combinatorial optimization problem, finding the feature set maximizing the quality of the hypothesis learned from these features. Global approaches to this optimization problem, referred to as wrapping methods, actually use an embedded learning algorithm to evaluate a feature set [21, 13]. For this reason, basic wrapping approaches hardly scale up to large size problems, though some progress has been done using ensemble-like evolutionary approaches [8, 19].

A relaxed formalization of FS is concerned with feature ranking (FR) [9]. In the FR approach, one selects the top ranked features, the number of which is either specified by the user [10] or analytically determined [20].

Ensemble Feature Ranking proceeds by exploiting a set of hypotheses independently learned from the dataset. Each hypothesis induces a ranking on the features, and EFR achieves the aggregation of these feature rankings. Based on the same principles as ensemble learning, the performance of an EFR increases with the ensemble size if it aggregates weakly competent feature rankings (misordering two features with probability $p < \frac{1}{2}$) [14].

EFR is implemented using an evolutionary learning algorithm termed *ROGER* for *ROC-based Genetic Learner*, first presented in [18]. *ROGER* extracts hypotheses optimizing the recently investigated AUC learning criterion, the area under the Receiver Operating Characteristics (ROC) curve [2, 15, **?**].

The paper is organized as follows. Section 2 briefly reviews the state of the art in Feature Selection and Ranking. An overview of Ensemble Feature Ranking is given in Section 3, describing the *ROGER* algorithm for the sake of completeness. The experimental validation setting is described in Section 4; a statistical model is devised to enable a direct evaluation of FR algorithms, inspired from the phase transition paradigm developed in the CSP community [11] and imported in the ML community by [7]. Section 5 reports on the comparative empirical validation results, and the paper ends with a discussion and perspectives for further research.

2 State of the art

This section introduces some Feature Ranking algorithms, referring to [9] for a more comprehensive discussion.

Notations used throughout the paper are first introduced. Only binary concept learning is considered in the following. The training set $\mathcal{E}$ includes n examples described from d continuous features, $\mathcal{E} = \{(\mathbf{x}_i, y_i),\ \mathbf{x}_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\},\ i = 1 \ldots n\}$, where label y_i indicates whether the i-th example pertains to the target concept (positive example) or not (negative example).

2.1 Univariate Feature Ranking and Iterated Selection

In univariate approaches, a score is associated to each feature independently from the others. In counterpart for this simplicity, univariate approaches are adversely affected by disjunctive concepts and redundant features.

The feature score is computed after a statistical test, quantifying how well this feature discriminates positive and negative examples. For instance the Mann-Whitney test associates to the k-th feature the score $MW(k)$ defined as:

$$MW(k) = Max(\ Pr(x_{i,k} > x_{j,k} \mid y_i > y_j), \quad Pr(x_{i,k} < x_{j,k} \mid y_i > y_j))$$

where $Pr(x_{i,k} > x_{j,k} \mid y_i > y_j)$ denotes the fraction of pairs of (positive,negative) examples such that feature k ranks the positive example higher than the negative one. This criterion coincides with the Wilcoxon rank sum test, which is equivalent to the AUC criterion [22].

A sophisticated extension of univariate approaches, based on an iterative selection process, is presented in [20]. The score associated to each feature is proportional to its cosine with the target concept ($score(k) = \sum_{i=1}^{n} x_{i,k} . y_i / \sqrt{\sum_{i=1}^{n} x_{i,k}^2}$).

This two-step iterative process i) determines the current feature k maximizing the above score; ii) projects all remaining features and the target concept on the hyperplane perpendicular to feature k, thereby overcoming the limitations of univariate approaches with respect to redundant features.

2.2 ML-based Feature Ranking

An alternative to univariate approaches is to exploit the output of a machine learning algorithm, which assumedly takes into account every feature in relation with the other ones [13].

Actually, a linear hypothesis ($h(\mathbf{x}) = \sum_{i=1}^{N} w_i x_i [+b]$) induces a feature ranking, associating to each feature k the absolute value of weight w_k; the higher the score, the more relevant the feature is *in combination with the other features*.

A two-step iterative process, termed SVM-Recursive Feature Elimination, is proposed by [10]. In each step, i) a linear SVM is learned, the features are ranked after the associated absolute weight; ii) the worst features are filtered out.

Another approach, based on linear regression [1], uses a randomized approach for better robustness, extracting various hypotheses from subsamples of the dataset, and associating to each feature its average weight.

Another related work is concerned with learning an ensemble of GA-based hypotheses extracted along independent runs [8], where: i) the underlying GA-inducer searches good feature subsets; and ii) the quality of a feature subset is measured from the accuracy of a k-nearest neighbor or euclidean decision table classification process, based on these features.

3 Ensemble Feature Ranking

This section introduces Ensemble Feature Ranking. EFR is implemented using the evolutionary learning algorithm *ROGER*, which is first described for the sake of completeness.

3.1 *ROGER*: ROc-based GEnetic learneR

ROGER is an evolution strategy over a continuous hypothesis space, that maximizes the area under the ROC curve (AUC) associated to each hypothesis h. As mentioned already, AUC is equivalent to the Wilcoxon statistics [22]:

$$AUC(h) = Pr(h(\mathbf{x}_i) > h(\mathbf{x}_j) \mid y_i > y_j)$$

The maximal fitness 1 is thus attained for a hypothesis h ranking all positive examples higher than negative examples (separating hypothesis).
Previous experiments using simple linear hypotheses ($h = \mathbf{w} \in \mathbb{R}^d$, $h(\mathbf{x}) =< \mathbf{w}, \mathbf{x} >$) show good learning performances compared to state-of-art linear Support Vector Machines [4], *ROGER* reaching a similar predictive accuracy in a fraction of the SVM computational time [18].

Thanks to the flexibility of evolutionary search, *ROGER* can explore complex search spaces provided that they can be parameterized in a compact way. In this paper, *ROGER* considers hypotheses defined as the weighted L_1 distance to some point $\mathbf{c}$ of the instance space. Formally, $h \in \mathbb{R}^{2d}$ is characterized as $(w_1, \ldots, w_d, c_1, \ldots, c_d)$, with

$$h(\mathbf{x}) = \sum_{i=1}^{d} w_i \times |x_i - c_i|$$

This way, *ROGER* explores a limited kind of non linear hypotheses, while exploring search space $\mathbb{R}^{2d}$ with linear size in the number of features. In the meanwhile, such non-linear hypotheses still allow for feature ranking (in contrast with quadratic or Gaussian functions), again associating to feature k the absolute value of weight w_k.

3.2 Ensemble feature ranking

Let $h_1, \ldots h_T$ denote T hypotheses. At the moment, the h_ts are the best hypotheses extracted by *ROGER* along T independent runs; using diversity enforcing heuristics to extract several hypotheses from a single run is a perspective for further research. For the sake of simplicity and by abuse of notations, let us denote $h(k)$ the absolute value of the k-th weight in h. With no loss of generality, hypotheses h_t are normalized ($\sum_k h_t(k) = 1$).

The goal is to construct an ensemble feature ranking h^*, aggregating the feature rankings derived from the h_ts. The justification for such an aggregation, presented in [14], is based on the same principles as ensemble learning; assuming that an elementary feature ranking offers an advantage over random guessing (the probability p of misranking a pair of features being less than $\frac{1}{2}$), the aggregation mechanism allows for amplifying this advantage as the ensemble size T increases [5].

Several aggregation procedures have been considered. The first one defines $h^*(k)$ as the proportion of features ℓ that are ranked before k by a majority of h_ts ($h^*(k) = \#\{\ell, \#\{t \ / \ h_t(\ell) < h_t(k)\} > \frac{T}{2}\}$, where $\#A$ stands for the size of set A). The convergence of this aggregation procedure wrt the ensemble size T has been analytically studied in [14].
The second (respectively the third) aggregation procedure associates to each feature k the median (resp. the maximal) value in $\{h_t(k), t = 1 \ldots T\}$.
A perspective for further research is concerned with a better exploitation of the order statistics of the $h_t(k)$ (e.g. setting $h^*(k)$ as some quantile of the set $\{h_t(k)\}$).

4 Statistical Validation Model

Before proceeding to experimental validation, it must be noted that the performance of a feature selection algorithm is commonly computed from the performance of a learning algorithm based on the selected features, which makes it difficult to compare FS algorithms standalone.

To sidestep this difficulty, a statistical model is devised, enabling the direct evaluation of the proposed FR approach. This model is inspired from the statistical complexity analysis paradigm developed in the Constraint Satisfaction community [11], and first imported in the Machine Learning community by Giordana and Saitta [7].

In this framework, the problem space is defined along a set of order parameters (e.g. the constraint density and tightness in CSPs [11]). The performance of a given algorithm is viewed as a random variable, observed in the problem space. To each point in the problem space (values of the order parameters), one associates the average behavior of the algorithm over all problem instances with same value of the order parameters.

4.1 Order parameters

Seven order parameters are defined for Feature Selection: i) the number d of features; ii) the number n of examples; iii) the number r of relevant features, where a feature is said to be relevant iff it is involved in the definition of the target concept, see next; iv) the type l of target concept, linear ($l = 1$) or non-linear ($l = 2$), with

$$l = 1: \quad y(\mathbf{x}) = 1 \quad \text{iff} \quad (\textstyle\sum_{i=1}^{r} x_i > s) \tag{1.1}$$
$$l = 2: \quad y(\mathbf{x}) = 1 \quad \text{iff} \quad (\textstyle\sum_{i=1}^{r} (x_i - .5)^2 < s) \tag{1.2}$$

v) the redundancy k of the relevant features (practically, redundancy ($k = 1$) is implemented by replacing r of the irrelevant features, by linear random combinations of the r relevant ones); vi) the noise rate e in the class labels: the class associated to each example is flipped with probability e; vii) the noise rate σ in the feature values, where a Gaussian noise $\mathcal{N}(0, \sigma)$ is added to each feature value.

4.2 Artificial problem generator

For each point $(d, n, r, l, k, e, \sigma)$ in the problem space, independent instances of learning problems are generated after the following distribution.

All d features of all n examples are drawn uniformly in $[0, 1]$. The label of each example is computed as in equation (1.1) (for $l = 1$) or equation (1.2) (for $l = 2$)[1]. In case of redundancy ($k = 1$), r irrelevant features are selected

[1] The threshold s referred to in the target concept definition is set to $r/2$ in equation (1.1) (respectively $r/12$ in equation (1.2)), guaranteeing a balanced distribution of positive and negative examples. The additional difficulties due to skewed example distributions are not considered in this study.

and replaced by linear combinations of the r relevant ones. Last, the example labels are randomly flipped with probability e, and the features are perturbed by addition of a Gaussian noise with variance σ.

The above generator differs from the generator proposed in [9] in several respects. [9] only considers linear target concepts, defined from a linear combination of the relevant features; this way, the target concept differentially depends on relevant features, whilst all features have same relevance in our model. In contrast, the proposed model investigates linear as well as a (limited kind of) non-linear concepts.

4.3 Format of the results

Feature rankings are evaluated and compared using a ROC-inspired setting. To each index $i \in \{1, d\}$ is associated the fraction of true relevant features (respectively, the fraction of irrelevant, or false relevant, features) with rank higher than i, denoted $TR(i)$ (resp. $FR(i)$). The curve $\{(FR(i), TR(i)), i = 1, \ldots, d\}$ is referred to as ROC-FS curve associated to $\mathcal{O}$.

The ROC-FS curve shows the trade-off achieved by the algorithm between the two objectives of setting high ranks (resp. low ranks) to relevant (resp. irrelevant) features. The ROC-FS curve associated to a perfect ranking (ranking all relevant features before irrelevant ones), reaches the global optimum $(0, 1)$.

The inspection of the ROC-FS curves shows whether a Feature Ranking algorithm consistently dominates over another one. The curve also gives a precise picture of the algorithm performance; the beginning of the curve shows whether the top ranked features are actually relevant, suggesting an iterative selection approach as in [20]; the end of the curve shows whether the low ranked features are actually irrelevant, suggesting a recursive elimination procedure as in [10].

Finally, three indicators of performance are defined on a feature ranking algorithm. The first indicator measures the probability for the best (top) ranked feature to be relevant, noted p_b, reflecting the FR potential for a selection procedure. The second indicator measures the worst rank of a relevant feature, divided by d, noted p_w, reflecting the FR potential for an elimination procedure. A third indicator is the area under the ROC-FS curve (AUC), taken as global indicator of performance (the optimal value 1 being obtained for a perfect ranking).

5 Experimental Analysis

This section reports on the experimental validation of the EFR algorithm described in section 3, compared to the baseline results provided by the cosine criterion [20].

5.1 Experimental setting

A principled experimental validation has been conducted along the formal model defined in the previous section. The number d of features is set to 100, 200 and

500. The number r of relevant features is set to $d/20, d/10$ and $d/5$. The number n of examples is set to $d/2$, d and $2d$. Linear and non-linear target concepts are considered ($l = 1$ or 2), with redundant ($k = 1$) and non-redundant ($k = 0$) feature sets. Last, the label noise e is set to 0, 5 and 10%, and the variance σ of the feature Gaussian noise is set to 0., .05 and .10.

In total 972 points $(d, r, m, l, k, e, \sigma)$ of the problem space are considered. For each point, 20 datasets are independently generated. For each dataset, 15 independent *ROGER* runs are executed to construct an ensemble feature ranking $\mathcal{O}$; the associated indicators p_b, p_w and AUC are computed, and their median over all datasets with same order parameters is reported.

The reference results are obtained similarly from the cosine criterion [20]: for each point of the problem space, 30 datasets are independently generated, the cosine-based feature ranking is evaluated from indicators p_b, p_w and AUC, and the indicator median over all 30 datasets is reported.

Computational runtimes are measured on PC Pentium-IV. *ROGER* is parameterized as a (20+200)-ES with self adaptive mutation, uniform crossover with rate .6, uniform initialization in $[0, 1]$, and a maximum number of 50,000 fitness evaluations[2].

5.2 Reference results

The performance of the cosine criterion for linear and non-linear concepts is illustrated on Fig. 1, for $d = 100, n = 50, r = 10, k = 0$.

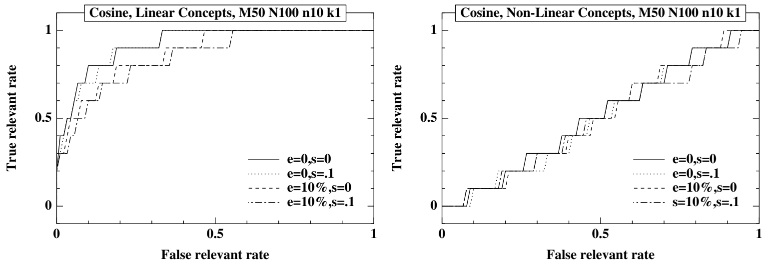
The performance indicators summarized in Table 1.(a), show an outstanding behavior on linear concepts; complementary results, omitted due to space limitations, show similar trends for redundant problems $k = 1$ and higher values of d. With twice as many features as examples, the probability p_b of top ranking a relevant feature is around 90%. A graceful degradation of p_b is observed as the noise rate increases, more sensitive to the label noise than to the feature noise. The relevant features are in the top p_w features, where p_w varies from 1/3 to roughly 1/2. The performance steadily improves when the number of examples increases.

In opposition, the cosine criterion behaves no better than random ranking for non-linear concepts; this is visible as the ROC-FS curve is close to the diagonal, and the situation does not improve by doubling the number of examples.

5.3 Evolutionary Feature Ranking

The performance of EFR is measured under the same conditions (Fig. 2, Table 1.(b)). EFR is clearly outperformed by the cosine criterion in the linear case. With twice as many features as examples, the probability p_b of top ranking a relevant feature ranges between 35 and 50% (non redundant features), against 80 and 90% for the reference results. When the number of examples increases,

[2] All datasets and *ROGER* results are available at
`http://www.lri.fr/∼sebag/EFRDatasets` and `http://www.lri.fr/∼sebag/EFResults`.

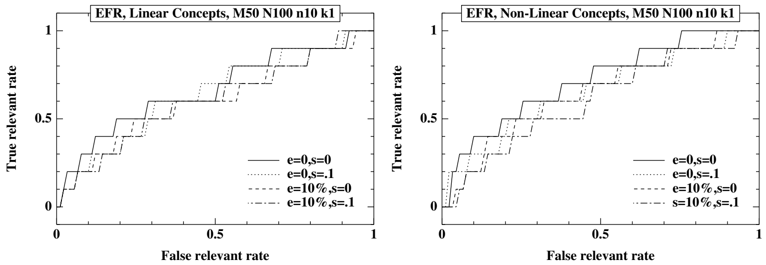


(a) Linear concepts (b) Non-linear concepts

Fig. 1. Cosine criterion: Median ROC-FS curves over 30 training sets on Linear and Non-Linear concepts, with $d = 100$, $n = d/2$, $r = d/10$, Non redundant features.

Table 1. Comparative results of the cosine ranking criterion and EFR: Probability p_b of top ranking a relevant feature, Median relative rank p_w of the worst ranked relevant feature, Area under the ROC-FS curve.

					(a) Cosine criterion						(b) Ensemble Feature Ranking					
					Linear			Non-Linear			Linear			Non-Linear		
n	d	r	e	σ	p_b	p_w	AUC	p_b	p_w	AUC	p_b	p_w	AUC	p_b	p_w	AUC
50	100	10	0	0	0.87	.33	0.920	0.03	.93	0.49	0.5	.92	0.67	0.20	.75	0.71
50	100	10	0	0.1	0.9	.33	0.916	0.03	.94	0.49	0.5	.80	0.63	0.45	.82	0.68
50	100	10	10%	0	0.87	.47	0.87	0.1	.93	0.49	0.35	.94	0.61	0.25	.81	0.68
50	100	10	10%	0.1	0.8	.56	0.848	0.03	.93	0.51	0.35	.89	0.62	0.25	.88	0.61
100	100	10	0	0	1	.18	0.97	0	.91	0.53	0.85	.79	0.79	0.55	.63	0.81
100	100	10	0	0.1	1	.22	0.966	0.03	.90	0.52	0.50	.74	0.77	0.60	.72	0.78
100	100	10	10%	0	0.93	.29	0.944	0.17	.92	0.52	0.55	.77	0.72	0.65	.78	0.77
100	100	10	10%	0.1	0.93	.36	0.934	0.1	.92	0.52	0.65	.82	0.75	0.40	.72	0.75



(a) Linear concept (b) Non-linear concept

Fig. 2. EFR performance: Median ROC-FS curves over 20 training sets on Linear and Non-Linear concepts, with $d = 100$, $n = d/2$, $r = d/10$, Non redundant features.

p_b increases as expected; but p_b reaches 55 to 85% against 93 to 100% for the reference results.

In opposition, EFR does significantly better than the reference criterion in the non-linear case. Probability p_b ranges around 30%, compared to 3% and 10% for the reference results with $n = 50$ and p_b increases up to circa 55% when n increases up to 100.

With respect to computational cost, the cosine criterion is linear in the number of examples and in $d \log d$ wrt the number of features; the runtime is negligible in the experiment range.

The computational complexity of EFR is likewise linear in the number of examples. The complexity wrt the number of features d is more difficult to assess as d governs the size of the *ROGER* search space ($[0,1]^{2d}$). The total cost is less than 6 minutes (for 20 data sets $\times$ 15 *ROGER* runs) for $n = 50, d = 100$ and less than 12 minutes for $n = 100, d = 100$. The scalability is demonstrated in the experiment range as the cost for $n = 50, d = 500$ is less than 23 minutes.

6 Discussion and Perspectives

The contribution of this paper is based on the exploitation of the diverse hypotheses extracted along independent runs of evolutionary learning algorithms, here *ROGER*. This collection of hypotheses is exploited for ensemble-based feature ranking, extending the ensemble learning approach [3] to Feature Selection and Ranking [9].

As should have been expected, the performances of the Evolutionary Feature Ranker presented are not competitive with the state of the art for linear concepts. However, the flexibility of the hypothesis search space explored by *ROGER* allows for a breakthrough in (a limited case of) non-linear concepts, even when the number of examples is a fraction of the number of features.

These results are based on experimental validation over 9,000 datasets, conducted after a statistical model of Feature Ranking problems. Experiments on real-world data are underway to better investigate the EFR performance, and the limitations of the simple model of non-linear concepts proposed.

Further research will take advantage of multi-modal evolutionary optimization heuristics to extract diverse hypotheses from each *ROGER* run, hopefully reducing the overall computational cost of the approach and addressing more complex learning concepts (e.g. disjunctive concepts).

Acknowledgments

We thank A. Cornuéjols and J. Mary for many discussions. The third author is partially supported by the PASCAL Network of Excellence, IST-2002-506778.

References

1. J. Bi, K.P. Bennett, M. Embrechts, C.M. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *J. of Machine Learning Research*, 3:1229–1243, 2003.
2. A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 1997.
3. L. Breiman. Arcing classifiers. *Annals of Statistics*, 26(3):801–845, 1998.
4. R. Collobert and S. Bengio. Svmtorch: Support vector machines for large-scale regression problems. *J. of Machine Learning Research*, 1:143–160, 2001.
5. R. Esposito and L. Saitta. Monte Carlo theory as an explanation of bagging and boosting. In *Proc. of IJCAI'03*, pp. 499–504. 2003.
6. Y. Freund and R.E. Shapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *Proc. ICML'96*, pp. 148–156. Morgan Kaufmann, 1996.
7. A. Giordana and L. Saitta. Phase transitions in relational learning. *Machine Learning*, 41:217–251, 2000.
8. C. Guerra-Salcedo and D. Whitley. Genetic approach to feature selection for ensemble creation. In *Proc. GECCO'99*, pp. 236–243, 1999.
9. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. of Machine Learning Research*, 3:1157–1182, 2003.
10. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.
11. T. Hogg, B.A. Huberman, and C.P. Williams (Eds). *Artificial Intelligence: Special Issue on Frontiers in Problem Solving: Phase Transitions and Complexity*, volume 81(1-2). Elsevier, 1996.
12. K. Imamura, R.B. Heckendorn, T. Soule, and J.A. Foster. Abstention reduces errors; decision abstaining n-version genetic programming. In *Proc. GECCO'02*, pp. 796–803. Morgan Kaufmann, 2002.
13. G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proc. ICML'94*, pp. 121–129. Morgan Kaufmann, 1994.
14. K. Jong, J. Mary, A. Cornuéjols, E. Marchiori, and M. Sebag. Ensemble feature ranking. In *Proc. ECML-PKDD'04*, 2004. to appear.
15. C.X. Ling, J. Hunag, and H. Zhang. AUC: a better measure than accuracy in comparing learning algorithms. In *Proc. of IJCAI'03*, 2003.
16. S. Rosset. Model selection via the auc. In *Proc. ICML'04*. Morgan Kaufmann, 2004, to appear.
17. M. Sebag, J. Azé, and N. Lucas. Roc-based evolutionary learning: Application to medical data mining. In *Artificial Evolution VI*, pp. 384–396. Springer Verlag LNCS 2936, 2004.
18. D. Song, M. I. Heywood, and A. Nur Zincir-Heywood. A linear genetic programming approach to intrusion detection. In *Proc. GECCO'02*, pp. 2325–2336. Springer-Verlag, 2003.
19. H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar. Ranking a random feature for variable and feature selection. *J. of Machine Learning Research*, 3:1399–1414, 2003.
20. H. Vafaie and K. De Jong. Genetic algorithms as a tool for feature selection in machine learning. In *Proc. ICTAI'92*, pp. 200–204, 1992.
21. L. Yan, R. H. Dodier, M. Mozer, and R. H. Wolniewicz. Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic. In *Proc. ICML'03*, pp. 848–855. Morgan Kaufmann, 2003.