Selecting Relevant Features for Splice Site Prediction by Estimation of Distribution Algorithms

Yvan Saeys[1], Sven Degroeve[1], Dirk Aeyels[2], Yves Van de Peer[1] and Pierre Rouzé[1,3]

[1] Department of Plant Systems Biology, Ghent University, Flanders Interuniversity Institute of Biotechnology (VIB), K.L. Ledeganckstraat 35, Ghent, 9000, Belgium
[2] SYSTeMS Research Group, Ghent University, Technologiepark - Zwijnaarde 9, Zwijnaarde, 9052, Belgium
[3] Laboratoire associé de l'INRA (France)

1 Introduction

For many biological processes, it is still not clear which elements contribute to the observed behaviour. One example is the occurence of splice sites in gene sequences, an important characteristic for gene finding in genome sequencing projects, as well as for better understanding the molecular mechanism of gene expression. The DNA sequence of most genes are coding for messenger RNA (mRNA) themselves encoding proteins. While in lower organisms (prokaryotes) the mRNA is a mere copy of a fragment of the DNA, in higher organisms (eukaryotes) the DNA contains non-coding segments in genes (introns) which should be precisely spliced out to produce the mRNA. The splice sites we refer to here are the border sides of such introns. The splice site on the left (upstream part) of the intron is called the donor site, the other site is termed the acceptor site. Due to the completion of the sequencing of the genome of *Arabidopsis thaliana*, a model system for plants, much data became available, allowing the use of supervised learning methods to automate the process of splice site prediction. As it is not clear which features are relevant for an accurate splice site prediction these learning methods are usually provided with many features, assuming that this will increase the probability of including relevant information. Since not all features are important, and some of the features might be correlated, there is a need to search for an optimal subset of features that maximizes the predictive accuracy of the classification system. Traditional Feature Subset Selection (FSS) methods are sequential and are based on a greedy heuristic. Sequential Forward Elimination (SFE) starts with the empty feature set and iteratively adds features, while Sequential Backward Elimination (SBE) starts with the full feature set and iteratively discards features. More advanced methods use heurstics to search the space of feature subsets, like e.g. genetic algorithms. Recently, Estimation of Distribution Algorithms (EDAs; Mühlenbein and Paaβ, 1996) emerged as a more general framework of genetic algorithms. Instead of using the traditional crossover and mutation operators to create the new population, a more statistical approach is used to estimate the distribution of the parameters from a selected group of individuals. Creation of the new population is then performed by sampling individuals from the estimated distribution. EDAs have proven to outperform the standard genetic algorithms in many problems where multiple dependencies among parameters exist, and they usually need fewer fitness evaluations to obtain good solutions. We combined the use of Estimation of Distribution Algorithms with a wrapper approach for feature subset selection. The results of our experiments show that it clearly outperforms the traditional sequential methods for FSS.

2 Methods

2.1 Splice site data sets

The *Arabidopsis thaliana* data set was generated by aligning mRNAs obtained from the public EMBL database with the BAC-sequences that were used for the *Arabidopsis* chromosome assembly. Redundant genes were excluded resulting in a data set containing 1495 genes. From each gene only these introns confirming the GT-AG consensus were used to construct the set of positive instances. All GT dinucleotides at the start of these introns are positive donor instances and all AG dinucleotides at the end of these introns are positive acceptor instances. The negative donor instances are de-

fined as, for all genes, all GT dinucleotides that are located between 100 nucleotide positions upstream of the first donor and 100 nucleotide positions downstream of the last acceptor in that gene and that are not donor sites. The negative acceptor instances are defined as all AG dinucleotides within the same range and that are not acceptor sites. More information on the procedure for creating these datasets can be found in (Degroeve et al., 2002).

Splice site prediction can be divided into two subtasks : prediction of donor sites and prediction of acceptor sites. Each of these subtasks can be formally stated as a two-class classification task : {donor site, non-donor site} and {acceptor site, non-acceptor site}. The features describing the positive and negative instances can be divided into two subsets : position-dependent and position-independent features. In our experiments we used a fixed window of p nucleotide positions to the left (upstream the splice site) and q positions to the right (downstream the splice site) where $p = q = 50$. From this local context, 100 position-dependent and 128 position-independent features were extracted. The position-dependent features were then converted into binary format using sparse vector encoding. This results in 400 position-dependent and 128 position-independent features. A training data set with balanced class distribution was compiled by random selection of 1000 positive instances and 1000 negative instances (GT_{2000} and AG_{2000}). For the test data set we extracted all candidate splice sites within the interval as defined above from 50 genes. This results in a test data set GT_{50} with 281 positive and 7505 negative instances and a test data set AG_{50} with 281 positive and 7643 negative instances.

2.2 Estimation of Distribution Algorithms

During the last years, Estimation of Distribution Algorithms (EDA's) emerged as a more general framework for genetic algorithms. The main critics for standard genetic algorithms include the large number of parameters that have to be tuned, the difficult prediction of the movements of the populations in the search space and the fact that there is no mechanism for capturing the relations among the variables of the problem. EDA's try to overcome these difficulties by providing a more statistical analysis of the selected individuals, thereby explicitly modelling the relationships among the variables.

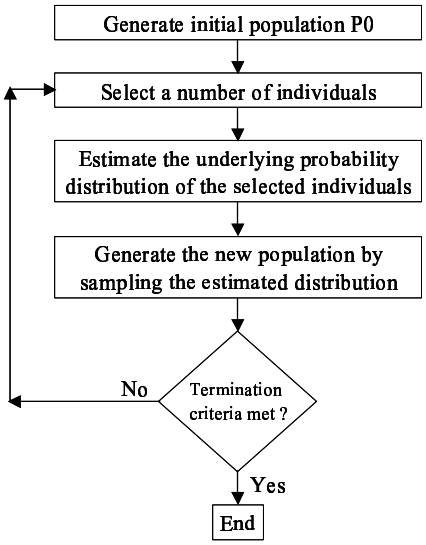

Figure 1: Schematic overview of the EDA algorithm.

Figure 1 illustrates the main scheme of the EDA approach. The actual estimation of the underlying probability distribution of the selected individuals represents the core of the new EDA paradigm, and can be considered an optimization problem on its own. Depending on the domain (discrete or continuous), different estimation algorithms with varying complexity (modelling univariate, bivariate or multivariate dependencies) were designed. For an overview see Larrañaga and Lozano (2001). In the most complex case of multivariate dependencies, Bayesian Networks are frequently used. A greedy search algorithm is then used to find a suitable (and often constrained) network that is likely to generate the selected individuals.

2.3 Classification models

As described above, our data sets contain positive and negative instances that are described by q nucleotide positions downstream and p nucleotide positions upstream the consensus. Formally, a data set T contains l instances $\mathbf{x_i}$ $(i = 1, \ldots, l)$ with each $\mathbf{x_i}$ labelled as y^+ or y^- (known as *classes*), indicating a positive or negative instance, respectively. Each index x_{ij} $(j = 1, \ldots, n)$ in vector $\mathbf{x_i}$ is a feature F_j.

Two methods for discriminating between positive and negative instances are described below. They are supervised classification methods that induce a decision function from the instances in T which can then be used to classify a new instance $\mathbf{z}$ not seen in T.

2.3.1 Support Vector Machines

The Support Vector Machine (SVM) (Boser et al., 1992; Vapnik, 1995) is a data-driven method for solving two-class classification tasks. The Linear SVM (LSVM) separates the two classes in T with a hyperplane in the feature space such that:

(a) the "largest" possible fraction of instances of the same class is on the same side of the hyperplane, and

(b) the distance of either class from the hyperplane is maximal.

The prediction of a LSVM for an unseen instance $\mathbf{z}$ is 1 (classified as a positive instance) or -1 (classified as a negative instance), given by the decision function

$$pred(\mathbf{z}) = \mathrm{sgn}(\mathbf{w} * \mathbf{z} + b). \quad (1)$$

The hyperplane is computed by maximizing a vector of Lagrange multipliers α in

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \, \alpha_j \, y_i \, y_j \, K(\mathbf{x_i}, \mathbf{x_j}),$$

$$\text{constrained to: } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{l} \alpha_i \, y_i = 0, \quad (2)$$

where C is a parameter set by the user to regulate the effect of outliers and noise, i.e. it defines the meaning of the word "largest" in (a).
Function K is a kernel function and maps the features in T, called the input space, into a feature space defined by K in which then a linear class separation is performed. For the LSVM this mapping is a linear mapping:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} * \mathbf{x_j}. \quad (3)$$

The non-linear mappings used in this paper is the Polynomial-SVM (PSVM):

$$K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} * \mathbf{x_j} + 1)^d, \quad (4)$$

After calculating the α_i's in (2), the decision function (1) becomes:

$$pred(\mathbf{z}) = \mathrm{sgn}(\sum_{i=1}^{l} \alpha_i \, y_i \, K(\mathbf{x_i}, \mathbf{z}) + b). \quad (5)$$

For the LSVM this function reduces to (1) with

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i \, \mathbf{x_i} \, y_i \, . \quad (6)$$

In (5) each α_i is associated with $\mathbf{x_i}$. After optimizing (2) many α_i's will become zero and the corresponding $\mathbf{x_i}$ will not be used in the decision function (5). All $\mathbf{x_i}$ for which the α_i is not zero are called the support vectors. Typically the size of the set of support vectors is much smaller than l.
The run-time complexity for training a Support Vector Machine is low order polynomial, usually approximately quadratic in the number of training samples (Hush and Scovel , 2000; Joachims, 1998).

2.3.2 Naive Bayes Classifier

The NBC (Duda and Hart, 1973) follows the Bayes optimal decision rule, that tells us to assign a class y^c (c in {+,-}) to an unseen instance $\mathbf{z}$ with features $(F_1^z, F_2^z, \ldots, F_n^z)$ that maximizes $P(y^c|F_1^z, \ldots, F_n^z)$, or the probability of the class y^c given the features $(F_1^z, F_2^z, \ldots, F_n^z)$. By using Bayes' rule we can write $pred(\mathbf{z}) = y^c$ as:

$$y^c = \mathrm{argmax}_c \frac{P(F_1^z, \ldots, F_n^z|y^c) \times P(y^c)}{P(F_1^z, \ldots, F_n^z)} \quad (7)$$

The naive Bayes method then simplifies the problem of estimating $P(F_1^z ... F_n^z|y^c)$ by making the arguable naive independence assumption that the probability of the features given the class is the product of the probabilities of the individual features given the class:

$$P(F_1^z, \ldots, F_n^z|y^c) = \prod_{1 \leq j \leq n} P(F_j^z|y^c). \quad (8)$$

The time complexity of the naive Bayes method is essentially linear in the number of training samples (McCallum and Nigam, 1998).

2.4 Feature subset selection methods

For selecting an optimal subset of features from $\{F_1, F_2, \ldots, F_n\}$, given the instances in T, one needs to define what is meant by "optimal subset of feature" (referred to as the *selection criterion*), and define a *search algorithm* to search for this optimal subset of features in the space of feature subset candidates. A review of different search algorithms can be found in (Kohavi and John, 1997; Boz, 2002) and techniques to combine feature subset selection and naive Bayes have been discussed in the literature (Hall, 1999; Langley and Sage, 1994)
As the number of feature subsets increases exponentially with increasing n (number of features) and n is relatively large, two techniques are justified : greedy search and heuristics. In this paper we will compare three methods : greedy search combined with SVM, greedy search combined with NBC, and heuristic search by using the EDA-approach combined with the NBC.

2.4.1 Greedy search

Both the SVM and NBC are known to perform well in high-dimensional input spaces because they implicitly avoid overfitting. This allows us to start the search algorithm with the full feature set. This set is known to be a good point to start our search in the space of feature subset candidates. The candidate space is explored with just one operator which eliminates a feature from the current subset. This bottom-up search procedure is called a sequential backward elimination (SBE) procedure and is greedy enough for our data sets. A simple criterion consists in selecting that feature that decreases the predictive performance of the model the least. We tested the SBE procedure both on the SVM and the NBC. For the NBC this can be done as follows. At iteration l the feature set consists of n_l features and n_l models can be trained, leaving out each feature once in each model. At iteration $l+1$ the feature set is then chosen belonging to the model with the best predictive performance. For SVM a slightly varying procedure is used : at iteration i for each feature F still in the feature subset we evaluate the generalization performance of the SVM model when setting F to its mean value in T (training set). This means that when considering a feature F for elimination, no model is retrained, but the alphas (2) of the previous model are reused, while setting x_{ij} to its mean value for all training instances. More details can be found in (Degroeve et al., 2002; Guyon et al., 2000).

2.4.2 Heuristic search

We combined the use of Estimation of Distribution Algorithms with a wrapper approach (Kohavi and John, 1997) for feature subset selection. The individuals in the population are represented as binary feature vectors, a 0 indicating an irrelevant feature, a 1 indicating a relevant feature. The goal of the EDA is then to look for the best subset with respect to some optimization criterion. In our case, the optimization criterion is a combination of the accuracy of the classification system coupled to the EDA and the number of features used to reach this accuracy. If two feature subsets achieve the same accuracy with respect to the same classification system, then the subset with the least number of features will result in a better fitness.
As the number of features in our real-world problem is quite large, we need to use quite large populations to allow a good estimation. Furthermore, a considerable amount of time is spent in analysing the fitness of each individual. For each individual a new model has to be trained, and this model has to be evaluated on a test set. Therefore, we need a fast classification algorithm and a fast estimation algorithm. In our experiments we used the NBC as the classification system, and the Univariate Marginal Distribution Algorithm (UMDA; Mühlenbein, 1998) as the estimation algorithm. Just like NBC, UMDA simplifies the estimation by assuming all features are independent : $p_l(x) = \prod_{i=1}^{n} p_l(x_i)$. Although using the naive assumption that parameters are independent both NBC and UMDA have shown to perform well in several fields such as text and image classification. As an adaptation to the standard UMDA we slightly modified the algorithm by replacing zero/one probabilities by very small/large probabilities.

3 Results

3.1 Ignoring feature dependencies

To investigate the effect of ignoring feature dependencies, we first compare the simple EDA-UMDA approach to a standard genetic algorithm (GA) with two-point crossover. Both

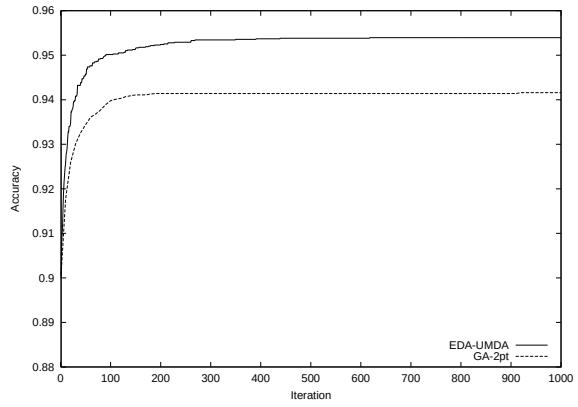

Figure 2: EDA versus GA fitness comparison.

EDA and GA used a NBC as a classification system. In both algorithms the fitness is defined as the accuracy obtained by the NBC on the test set for the selected subset of features. Both algorithms used a population of 500 individuals and an elitist approach where in each iteration the 50 best individuals survive. In both cases truncation selection was used, for the GA the crossover and mutation rate were set to 0.8 and 0.01. Figure 2 shows the comparison of both algorithms for a feature set of 528 features. Similar results were obtained on the feature set of 400 features. Clearly the EDA converges faster than the traditional GA (the EDA needs less iterations to achieve the same fitness as the GA) and also converges to better solutions than the GA. This suggests that ignoring feature dependencies can lead to good solutions. A similar conclusion was stated in (Cantú-Paz, 2002), where the author concluded that the complicated dependency learning EDA's are not needed, and the simple compact GA (Harik et al., 1998) will suffice. It has to be pointed out that the EDA-UMDA approach is very similar to the compact GA, or to a GA with uniform crossover.

3.2 Feature subset selection

To compare greedy and heuristic feature selection for splice site prediction we evaluated both techniques for the Naive Bayes Classifier. The greedy algorithm starts with the full feature set and iteratively removes features, the heuristic algorithm finds an optimal subset of features with regard to the classification accuracy of the NBC. Afterwards the greedy algorithm is applied to the solution found by the EDA, iteratively discarding features. Furthermore we evaluated the results for a Support Vector Machine with a second degree polynomial kernel (PSVM) also using the greedy approach. As a selection criterion, different measures can be used : accuracy, correlation coefficient, harmonic mean of sensitivity and specificity, or a combination of measures such as a weighted sum of accuracy and the number of eliminated features. The figures show the results when such a combination of the accuracy and the number of feature is used as selection criterion, but similar results are obtained with the other possible criteria.

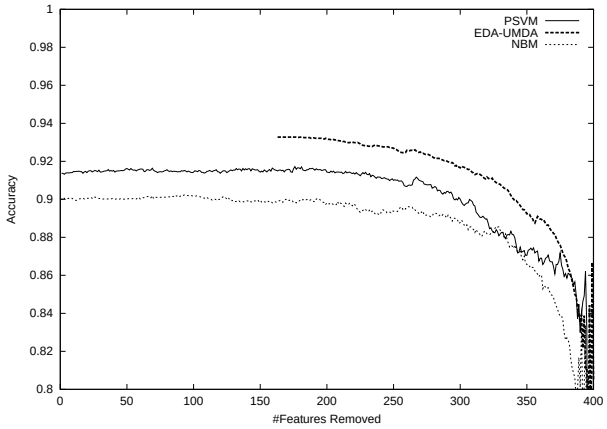

Figure 3: FSS on a set of position-dependent features.

Figure 3 and 4 show the accuracy of the three FSS methods, once for a feature set containing only position dependent features (400 features, figure 3) and once for a feature set containing position-dependent and position-independent features (528 features, figure 4). In the case of position-dependent features the SVM-model clearly performs better than the NBC. However, when applying the EDA heuristic method to the NBC, it clearly outperforms the SVM with a greedy approach. In the case of a mixture between position-dependent and position-independent features, the SVM-model starts with higher accuracy, but the NBM ap-

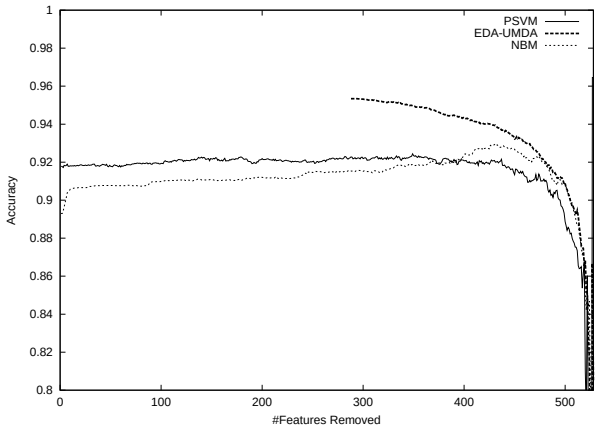

Figure 4: FSS on a set of position-dependent and position-independent features.

proach achieves higher accuracies with less features. Also here the EDA heuristic method achieves a better accuracy than both SVM and NBC with a greedy approach.

3.3 Biological relevance

The use of more position-independant features strongly influences the number of features needed to obtain good classification results. This can be demonstrated by comparing a set containing only position-dependent features with a set consisting of both position-dependent and position-independent features. For both feature sets we performed feature selection with EDA-UMDA, combined with the SBE greedy search algorithm. We then selected for each set the features that were sufficient and necessary to obtain a sensitivity of 0.93 and a specificity of 0.25. In the case of a set with only position-dependent features, 47 features were needed to achieve these values for accuracy and specificity. In the case of an extended set including also all triplets, only 25 features were needed to achieve these values. Clearly these features thus capture significant position-independent characteristics allowing a better discrimination of splice sites.

4 Related work

Genetic algorithms have been frequently used for feature subset selection in small scale (less than 100 features) domains (Kudo and Sklansky , 2000; Siedelecky and Sklansky, 1988; Vafaie and De Jong, 1993). The use of EDA's for feature subset selection was pioneered by (Inza et al., 1999) and the use of EDA's for FSS in large scale domains was reported to yield good results (Larrañaga and Lozano, 2001). Cantú-Paz (2002) compared several EDA's with the simple GA for small scale domains (at most 35 features) using a Naive Bayes classifier, and concluded that the complicated dependency learning EDA's are not significantly better than the simple compact GA.
Recently the technique of feature distributional clustering was combined with Support Vector Machines for text categorization (Bekkerman et al., 2001). This method performs feature selection by distributional clustering of words via the information bottleneck method (Tishby et al. , 1999) and can be considered a sophisticated filter method.

5 Conclusions and future work

Feature subset selection by estimation of distribution algorithms is able to select highly relevant features for splice site prediction. Future research will focus on including more position-independent information, possibly also structural information to achieve better results. More advanced feature selection frameworks such as a combined filter/wrapper model will then be needed, as the feature sets will get very large. Other interesting future directions are the combination of EDA with more advanced classification systems, and the development of faster estimation algorithms for multiple dependencies.

References

Bekkerman, R., El-Yaniv, R., Tishby, N. and Winter, Y. 2001. *On Feature Distributional Clustering for Text Categorizatio.* In *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval, pp. 146-153.*

Boser,B., Guyon,I. and Vapnik,V.N. 1992. *A training algorithm for optimal margin classifiers.* In *Proc. COLT (Haussler,D. ,ed.), ACN Press, 144-152.*

Boz, O. 2002. *Feature subset selection by using sorted feature relevance.* In *Proc. Int. Conf. on Machine Learning and Applications (ICMLA 2002).*

Cantú-Paz, E. 2002. *Feature subset selection by estimation of distribution algorithms.* In *W. B.*

Langdon, E. Cantu-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, N. Jonoska (Eds.), GECCO-2002: Proceedings of the Genetic and Evolutionary Computation Conference. (pp. 754). San Francisco, CA: Morgan Kaufmann.

Degroeve,S., De Baets,B., Van de Peer,Y. and Rouzé,P. 2002. *Feature Subset Selection for Splice Site Prediction.* In *Bioinformatics, 18-2:75-83.*

Duda,R.O. and Hart,P.E. 1973. *Pattern classification and scene analysis. New York, NY, Wiley.*

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. 2000. *Gene selection for cancer classification using support vector machines.* In *Machine Learning, 46:389-422.*

Hall, M.A. 1999. *Correlation based feature selection for machine learning. Doctoral dissertation, Department of Computer Science, The University of Waikato, Hamilton, New Zealand.*

Harik,G.R., Lobo, G.G. and Goldberg, D.E. 1998. *The compact genetic algorithm.* In *Proceedings of the International Conference on Evolutionary Computation 1998 (ECEC '98), pp. 523-528. Piscataway, NJ: IEEE Service Center.*

Hush, D. and Scovel, C. 2000. *Polynomial-time decomposition for support vector machines. Technical report, Los Alamos National Laboratory, Los Alamos, NM 87545.*

Kudo, M. and Sklansky, J. 2000. *Comparison of algorithms that select features for pattern classifiers.* In *Pattern Recogn. 33:25-41.*

Inza, I., Larrañaga, P., Etxebarria, R. and Sierra, B. 1999. *Feature subset selection by Bayesian networks based on optimization.* In *Artifical Intelligence, 27(2):143-164.*

Joachims, T. 1998. *Making large-scale support vector machine learning practical.* In *B. Scholkopf, C. Burges, A. Smola. Advances in Kernel Methods: Support Vector Machines, MIT Press, Cambridge, MA, December 1998.*

Kohavi,R. and John,G. 1997. *Wrappers for feature subset selection.* In *Artificial Intelligence Journal, 97:273-324.*

Langley, P. and Sage, S. 1994. *Induction of selective Bayesian classifiers..* In *Proceedings of the Tenth Conference on Uncertainty in Artifical Intelligence, pp. 399-406. Morgan Kaufmann, Seattle, WA.*

McCallum, A. and Nigam, K. 1998. *A comparison of event models for naive Bayes text classification..* In *AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48. AAAI Press, 1998..*

Mühlenbein,H. and Paaβ G. 1996. *From recombination of genes to the estimation of distributions. Binary parameters..* In *Lecture Notes in Computer Science 1411 : Parallel Problem Solving from Nature, PPSN IV, (pp. 178-187).*

Mühlenbein,H. 1998. *The equation for response to selection and its use for prediction.* In *Evolutionary Computation 5, pp. 303-346.*

Larrañaga,P. and Lozano,J.A. 2001. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic Publishers.*

Siedelecky, W. and Sklansky, J. 1988. *On automatic feature selection. Int. J. Pattern Recogn. 2: 197-220.*

Tishby, N., Pereira, F.C. and Bialek, W. 1999. *The information bottleneck method.* In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing, pp. 368-377.*

Vapnik,V.N. 1995. *The nature of statistical learning theory. Springer-Verlag.*

Vafaie, H. and De Jong, K. 1993. *Robust feature selection algorithms. Proceedings Fifht International Conference on Tools with Artificial Intelligence, pp. 356-363.*