BIOINFORMATICS

Vol. 19 Suppl. 2 2003, pages ii179–ii188
DOI: 10.1093/bioinformatics/btg1076

Fast feature selection using a simple estimation of distribution algorithm: a case study on splice site prediction

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

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

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Motivation: Feature subset selection is an important preprocessing step for classification. In biology, where structures or processes are described by a large number of features, the elimination of irrelevant and redundant information in a reasonable amount of time has a number of advantages. It enables the classification system to achieve good or even better solutions with a restricted subset of features, allows for a faster classification, and it helps the human expert focus on a relevant subset of features, hence providing useful biological knowledge.

Results: We present a heuristic method based on Estimation of Distribution Algorithms to select relevant subsets of features for splice site prediction in *Arabidopsis thaliana*. We show that this method performs a fast detection of relevant feature subsets using the technique of constrained feature subsets. Compared to the traditional greedy methods the gain in speed can be up to one order of magnitude, with results being comparable or even better than the greedy methods. This makes it a very practical solution for classification tasks that can be solved using a relatively small amount of discriminative features (or feature dependencies), but where the initial set of potential discriminative features is rather large.

Keywords: Machine Learning, Feature Subset Selection, Estimation of Distribution Algorithms, Splice Site Prediction.

Contact: yvsae@gengenp.rug.ac.be

INTRODUCTION

For many biological processes, it is still not clear which elements contribute to the observed behaviour. One example is the occurrence of splice sites in gene sequences, an important characteristic for gene finding in genome sequencing projects. The DNA sequences 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 in the upstream part of the intron is called the donor site, the other site is termed the acceptor site. Due to the completion of sequencing the genome from several eukaryotes, much data became available, allowing the use of supervised learning methods to automate the process of splice site prediction. Here we used sequence data from the model plant *Arabidopsis thaliana*, for which more than 12000 full length cDNAs are now available, providing a large dataset of genes with documented exon/intron structure. To complete this task these learning methods need information sources to enable them to distinguish between true and false sites. These information sources are termed *features* in machine learning. 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 relevant to the classification task and others might be correlated, there is a need to search for a 'minimal' set of features with 'best' classification performance. Traditional Feature Subset Selection (FSS) methods are sequential and are based on a greedy heuristic (Kohavi and John, 1997). Sequential Forward Selection (SFS) starts with the empty feature set and iteratively adds features, while Sequential Backward

*To whom correspondence shoudl be addressed.

Elimination (SBE) starts with the full feature set and iteratively discards features. More advanced methods use heuristics to search the space of feature subsets, like e.g. genetic algorithms (Kudo and Sklansky, 2000; Siedelecky and Sklansky, 1988; Vafaie and De Jong, 1993). Recently, Estimation of Distribution Algorithms (EDAs) emerged as a more general framework of genetic algorithms (Mühlenbein and Paass, 1996). 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. In this paper we present an approach for feature subset selection for splice site prediction, using a simple EDA as a wrapper for feature subset selection. We will demonstrate the usefulness of EDAs when the number of features in the subset is constrained, resulting in a very practical algorithm being considerably faster than the sequential methods. The results of our experiments show that this approach is able to select feature subsets with equal or higher relevance than the traditional sequential methods, resulting in a better classification of the splice sites.

METHODS

Splice site data sets

The *A.thaliana* dataset was generated by aligning cognate 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 dataset containing 1495 genes. From each gene only those introns confirming the GT-AG consensus were used to construct the set of positive instances. All GT dinucleotides at the upstream border of these introns are positive donor instances and all AG dinucleotides at the downstream border of these introns are positive acceptor instances. The negative donor instances are defined as, for all genes, all GT dinucleotides that are located between 300 nucleotide positions upstream of the first donor and 300 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. Additional negative instances were extracted in the same range from the complementary DNA strand. These datasets and the way they were created (Degroeve *et al.*, 2002) can be found on http://www.psb.rug.ac.be/gps#eccb03.

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 were extracted from a local context around the splice site. 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 $p = q = 50$. This results in 100 position-dependent features, which were converted into binary format using sparse vector encoding yielding 400 binary features. Training sets with balanced class distribution were compiled by random selection of 1000 positive instances and 1000 negative instances (400AG.train.2000 and 400GT.train.2000). For the test sets we extracted all candidate splice sites within the interval as defined above from 50 independant genes (400AG.test.50 and 400GT.test.50). For feature subset selection, an independent dataset termed the *holdout set* is used to evaluate the subsets (see further). Two kinds of holdout sets for feature subset selection were created : balanced sets containing 1000 positive and 1000 negative instances not occurring in the training set (400AG.holdout.2000 and 400GT.holdout.2000), and unbalanced sets containing instances from another 50 independant genes (400AG.holdout.50 and 400GT.holdout.50). Table 1 summarizes the number of positive and negative instances in the unbalanced datasets.

Table 1. Class distribution of the unbalanced datasets

Dataset	# Positives	# Negatives	# Total
400AG.test.50	277	18516	18793
400AG.holdout.50	263	20886	21149
400GT.test.50	309	16077	16386
400GT.holdout.50	239	14937	15176

Estimation of Distribution Algorithms

During the last years, Estimation of Distribution Algorithms (EDAs) 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. EDAs 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 illustrates the main scheme of the EDA approach. After the generation of the initial population an iterative procedure is carried out until the termination

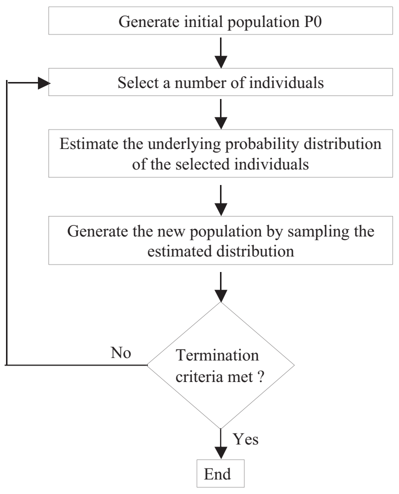
Generate initial population P0
Select a number of individuals
Estimate the underlying probability distribution of the selected individuals
Generate the new population by sampling the estimated distribution
Termination criteria met ?
No
Yes
End

Fig. 1. Schematic overview of the EDA algorithm.

criteria are met (e.g. a fixed number of iterations). In each step of the iteration a number of individuals is selected from the population (e.g. the better half) and from these a probability distribution of the encoded variables is estimated.

The actual estimation of the underlying probability distribution represents the core of the 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.

In the following step, the estimated probability distribution is used to generate the next population. This is done by sampling the probability distribution, i.e. generating individuals according to this distribution. A simple estimation algorithm is the Univariate Marginal Distribution Algorithm (UMDA, Mühlenbein, 1998). The UMDA simplifies the estimation by assuming all variables are independent. For each iteration l the probability model $p_l(x)$ is estimated as

$$p_l(x) = \prod_{i=1}^{n} p_l(x_i) = \prod_{i=1}^{n} p(x_i | D_{l-1}^{\mathrm{Se}})$$

where each $p_l(x_i)$ (the relative frequency) is estimated from the selected set of individuals of the previous generation D_{l-1}^{Se}. Generation of a new individual is then achieved by sampling a value from the distribution $p_l(x_i)$ for each variable x_i.

Classification models

As described above, our datasets 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 : the Naive Bayes Method (NBM) and the Support Vector Machine (SVM). These 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.

Support Vector Machines The Support Vector Machine (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

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

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

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

$$\text{constrained to: } 0 \le \alpha_i \le C \text{ and } \sum_{i=1}^{l} \alpha_i\, y_i = 0, \tag{2}$$

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:

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

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

$$K(\mathbf{x_i}, \mathbf{x_j}) = (s\, \mathbf{x_i} * \mathbf{x_j} + r)^d, \tag{4}$$

The degree d in the polynomial kernel describes the maximum order of feature interactions/dependencies. In the case of splice site prediction e.g. a kernel of degree 3 is able to model dependencies between up to 3 nucleotide positions (codon information). Similarly a kernel of degree 6 is able to extract dicodon (hexamer) information and a kernel of degree 9 can model tricodon (nonamer) information. Remark that these dependencies need not necessarily be between adjacent positions.

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

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

For the LSVM this function reduces to (1) with

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

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).

Naive Bayes Method The Naive Bayes Method (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:

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

The NBM 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:

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

The time complexity of the NBM is essentially linear in the number of training samples (McCallum *et al.*, 1998).

It is known that the NBM can achieve considerably better results when feature subset selection is applied, yet also the SVM can benefit from feature selection, although it already performs an implicit feature weighting (Guyon *et al.*, 2000).

Feature subset selection methods

To select an optimal subset of features from $\{F_1, \ldots, F_n\}$, given the instances in T, one needs to define what is meant by 'optimal subset of features' (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. As this is an optimization problem on its own, the evaluation of some subset needs to be calculated on a dataset. This dataset should be independent of both the training and test set and is termed the *holdout set*. As mentioned before we created two versions of the holdout set : balanced and unbalanced. The feature subset selection procedure then uses the training set to create a model, based on a particular subset of features. During the whole search process, only the holdout set is used for evaluation, and when the search has finished, a final evaluation on the test set is performed. We used a wrapper approach (Kohavi and John, 1997) for feature selection, embedding the classification methods within a greedy or heuristic framework.

Selection criterion The selection criterion used in most classification tasks is the accuracy ratio, defined as

$$\mathrm{ac} = \frac{\mathrm{TP + TN}}{\mathrm{TP + TN + FP + FN}}$$

where TP and TN denote the number of true positives/negatives, and FP and FN denote the number of false positives/negatives. However, the unbalanced class distribution of splice sites (the number of pseudo sites in a sequence is a number of magnitudes higher than the number of actual sites) makes things more complicated. Let Z be a set of instances where 98% of the instances are labelled as pseudo site. Then a simple classifier that always outputs the class 'pseudo' would have an accuracy ratio of 0.98. This would be hard to beat by a classifier that tries to capture the actual dependencies between features. Although current evaluation criteria such as sensitivity and specificity introduced by Snyder and Stormo (1995), or the correlation coefficient do not seem to suffer so drastically from this problem, they are still correlated to the class distribution of Z. A recent proposal, the $q9$ statistic (Zhang and Zhang, 2002), incorporates content-balancing, which means the measure is independent of the class distribution in Z.

$$q9 = \begin{cases} \frac{\mathrm{TN-FP}}{\mathrm{TN+FP}} \text{ if TP+FN = 0} \\ \frac{\mathrm{TP-FN}}{\mathrm{TP+FN}} \text{ if TN+FP = 0} \\ 1 - \sqrt{2}\sqrt{\left(\frac{\mathrm{FN}}{\mathrm{TP+FN}}\right)^2 + \left(\frac{\mathrm{FP}}{\mathrm{TN+FP}}\right)^2} \\ \text{ if TP+FN } \neq 0 \text{ and TN+FP} \neq 0 \end{cases} \tag{9}$$

This yields a number between -1 and $+1$, which can then be rescaled to a number between 0 and 1 by applying $q9 = \frac{1+q9}{2}$. As the class distribution between our test set and holdout sets differs considerably, we adopted the $q9$ statistic as the selection criterion to guide the search for good feature subsets.

Search algorithms A review of different search algorithms can be found in Kohavi and John (1997) and Boz (2002), and techniques to combine feature subset selection and naive Bayes have been discussed in the literature (Hall, 1999; Langley *et al.*, 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 greedy search and heuristic search with the EDA-approach. Both techniques were combined with the Support Vector Machine and the Naive Bayes Method.

SVM and NBM are known to perform well in high-dimensional input spaces because they implicitly avoid overfitting. This allows us to start the greedy search algorithm with the full feature set. 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. Another possibility would be to start with an empty feature set and iteratively add features. Such a method is termed a sequential forward selection (SFS) procedure. The strength of using backward feature elimination in comparison to forward selection is that correlated features are better discovered using SBE than with SFS.

We tested the SBE procedure both in combination with the SVM and the NBM. This was done as follows : at iteration l the feature set consists of n_l features and n_l models have to be trained, leaving out each feature once in each model. At iteration $l+1$ the feature set for the model with the best predictive performance (i.e. the best $q9$ statistic on the holdout set) is then chosen as the new feature subset.

For the heuristic approach we combined Estimation of Distribution Algorithms both with the Support Vector Machine and the Naive Bayes Method. The individuals in the population are represented as binary feature vectors, a 0 indicating an irrelevant/redundant 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. As the number of features for splice site prediction is quite large, we need to use rather large populations (up to 500 individuals) 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 the holdout set. Therefore, a fast classification algorithm and a fast estimation algorithm is preferred. In our experiments we used the NBM and SVM as the classification system, and the Univariate Marginal Distribution Algorithm (UMDA; Mühlenbein (1998)) as the estimation algorithm. It has to be noted that NBM and linear SVM are fast algorithms, the higher order polynomial SVMs are rather slow. Although using the naive assumption that parameters are independent both NBM 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.

Constraining feature subsets An important characteristic of using a string representation like in EDAs is the fact that we can easily constrain the size S of the feature subset (i.e. the number of features in the subset, so the number of 1's in the string) by adding or removing features. This can be used to explicitly search for the best feature subset given a number of features. To achieve a subset of the same size S using SBE, the algorithm starts with the full feature set, and iteratively discards features until S features are retained. If we denote the total number of features by N, then the SBE procedure needs

$$\text{Numeval}_{\text{SBE}} = \frac{N*(N+1) - S*(S+1)}{2} \tag{10}$$

model evaluations to complete this task. Each model evaluation involves the training of the model on the training set, and an evaluation on the holdout set. The average number of features over all model evaluations can then be calculated as

$$\frac{\sum_{i=S+1}^{N} i^2}{\text{Numeval}_{\text{SBE}}} \tag{11}$$

For an EDA with a population size of P, running for I iterations and using an elitist approach of E elitists (the best E individuals survive) the number of model evaluations can be calculated as

$$\text{Numeval}_{\text{EDA}} = P + (I-1)*(P-E) \tag{12}$$

Whereas the number of evaluations needed for SBE only depends on the total number of features and the size S of the subset, the number of evaluations for the EDA depends on P, I and E, but not directly on N and S. The average number of features over all model evaluations using the EDA-approach is fixed (S).

Implementation

The wrapper methods for feature selection were all implemented in C++, making use of the SVM^{light} implementation for Support Vector Machines (Joachims, 1998). Both

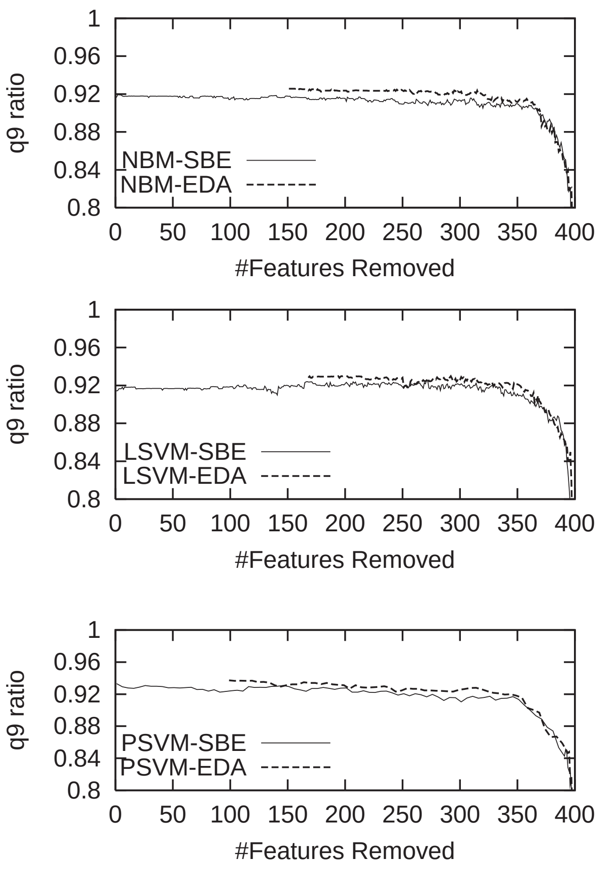

Fig. 2. Acceptor prediction : FSS on a balanced holdout set (400AG.holdout.2000).

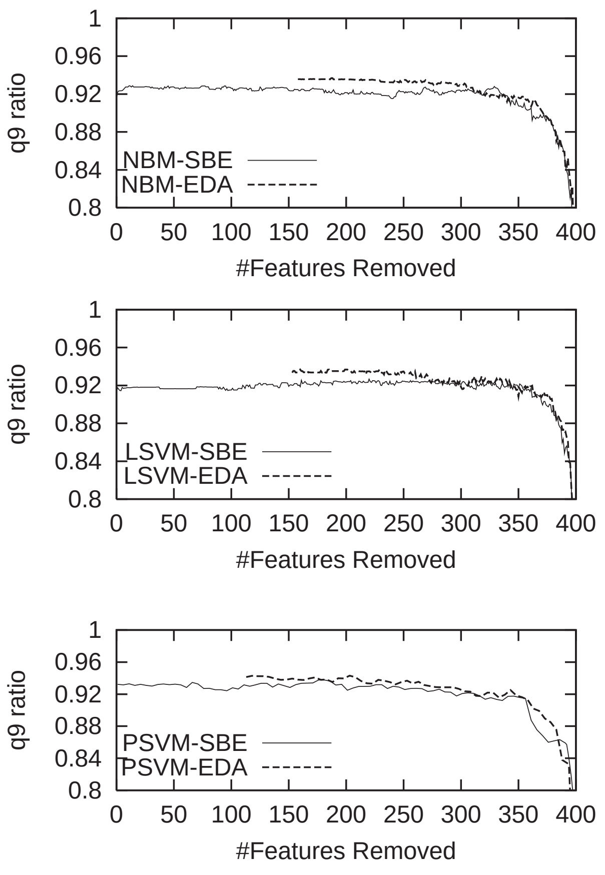

Fig. 3. Acceptor prediction : FSS on an unbalanced holdout set (400AG.holdout.50).

SBE and EDA are suitable candidates for parallellizing, providing a linear speedup of the selection process. This was done making use of the MPI libraries, available at http://www-unix.mcs.anl.gov/mpi/mpich. All experiments were run on a cluster of 5 dual-processor (1.2 Ghz) Linux machines running RedHat Linux 7.2.

RESULTS

All methods (NBM, LSVM and PSVM) were trained on a balanced dataset (400AG.train.2000 and 400GT.train.2000) and evaluated on an unbalanced test set (400AG.test.50 and 400GT.test.50). The experiments for feature subset selection were run on two holdout sets : a balanced data set (400AG.holdout.2000 and 400GT.holdout.50) and an unbalanced data set (400AG.holdout.50 and 400GT.holdout.50). For the polynomial SVM we used a ninth degree polynomial kernel function (d=9 in (4)), a linear coefficient of 0.01 (s=0.01) and a constant of 1 (r=1). For both the LSVM and PSVM we used a value of 0.05 for the c-parameter. These parameter values were determined experimentally using a cross-validation procedure.

Feature subset selection

To compare greedy and heuristic feature selection for splice site prediction we evaluated both techniques with NBM, LSVM and PSVM. 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 $q9$ ratio on the holdout set. Afterwards a greedy algorithm is applied to the solution found by the EDA, iteratively discarding features. For NBM and LSVM the population size P was set to 500, the number of elitists E to 50 and the number of iterations I to 150. As a standard practice we used the best half of the population to estimate the underlying probability distribution. For the PSVM the procedures for EDA-based and greedy FSS had to be adapted, as otherwise computations would take too long. Whereas in the greedy versions of NBM and LSVM only one feature was eliminated during each iteration, we eliminated five features at the time for each iteration in the case of PSVM. For the EDA approach P was changed to 100 and E to 10. Figure 2 compares the $q9$ ratios for NBM, LSVM and PSVM with a SBE and EDA method

Table 2. Acceptor prediction : evaluation of FSS on a balanced holdout set (400AG.holdout.2000)

Algorithm	# Features	CC	q9 ratio
NBM	150	31.26	90.94
SBE	80	29.89	90.51
	40	29.63	90.82
	Max	32.89	91.93
NBM	150	32.25 ± 0.41	91.75 ± 0.16
EDA	80	31.55 ± 0.77	91.55 ± 0.47
	40	28.18 ± 0.35	89.15 ± 0.35
	Unconstrained	33.18 ± 0.22	92.19 ± 0.18
LSVM	150	32.60	91.83
SBE	80	32.30	91.52
	40	29.86	90.31
	Max	33.52	92.57
LSVM	150	34.03 ± 0.43	92.55 ± 0.28
EDA	80	33.04 ± 0.64	92.32 ± 0.35
	40	29.47 ± 0.47	90.32 ± 0.48
	Unconstrained	34.65 ± 0.43	92.62 ± 0.44
PSVM	150	34.49	92.05
SBE	80	32.47	91.57
	40	28.57	90.00
	Max	35.77	93.30
PSVM	150	34.73 ± 0.84	92.68 ± 0.49
EDA	80	30.64 ± 0.44	90.75 ± 0.36
	40	26.82 ± 0.81	88.51 ± 0.84
	Unconstrained	36.24 ± 0.52	93.42 ± 0.35

Table 3. Acceptor prediction : evaluation of FSS on an unbalanced holdout set (400AG.holdout.50)

Algorithm	# Features	CC	q9 ratio
NBM	150	33.57	92.25
SBE	80	32.20	92.00
	40	30.04	90.38
	Max	34.20	92.87
NBM	150	35.19 ± 0.11	93.07 ± 0.10
EDA	80	33.49 ± 0.54	92.37 ± 0.59
	40	30.06 ± 0.37	90.17 ± 0.51
	Unconstrained	35.97 ± 0.25	93.14 ± 0.20
LSVM	150	34.95	92.43
SBE	80	33.20	91.92
	40	31.32	91.36
	Max	34.99	92.68
LSVM	150	36.88 ± 0.56	93.25 ± 0.46
EDA	80	35.04 ± 0.18	92.57 ± 0.19
	40	31.72 ± 0.44	91.37 ± 0.40
	Unconstrained	38.34 ± 0.32	93.31 ± 0.38
PSVM	150	35.02	92.57
SBE	80	32.94	91.35
	40	28.66	88.75
	Max	37.46	93.77
PSVM	150	36.20 ± 0.41	92.98 ± 0.37
EDA	80	32.46 ± 0.79	91.39 ± 0.80
	40	27.49 ± 0.49	88.47 ± 0.49
	Unconstrained	38.29 ± 0.38	93.94 ± 0.26

in the case of acceptor prediction and FSS on a balanced holdout set. On the x-axis the number of features that is eliminated so far is shown, starting at the origin with the full feature set. Figure 3 shows the $q9$ ratios in the case of acceptor prediction and FSS on an unbalanced holdout set. It is clearly observed that for each of the three classification methods, the EDA approach outperforms the SBE, being able to select more relevant features, yielding a better classification. Both greedy and heuristic feature selection achieve better results on the unbalanced holdout set. Similar trends were seen for donor prediction. The results and analysis for these sites are available online at http://www.psb.rug.ac.be/gps#eccb03.

Constraining the number of features

As already mentioned before, using the EDA-approach, the size S of the feature subset can be constrained by adapting the binary feature vectors. We compared the greedy and heuristic approach for three fixed values of S : 150, 80 and 40 features. The parameters we used were the same as described in the previous experiment, again adapting the parameters in the case of PSVM to enable computational feasibility. Table 2 summarizes the results for acceptor prediction with FSS on a balanced holdout set. The first column in the table describes the combination of the classification algorithm and the feature selection algorithm. In the second column the number of features that was used is displayed. It contains the evaluation values for fixed subsets of 150, 80 and 40 features. An additional value *Max* was added for SBE; this value indicates the best $q9$ ratio over all iterations. For the EDA the field *Unconstrained* indicates the value in the case the size of the feature subset is not constrained. This value is equal to the starting point of the EDA curves in Figures 2 and 3. The last two columns show the correlation coefficient (CC) and the $q9$ ratio. As the EDA-approach is a heuristic, we repeated these experiments in five independent runs. The results in the table show the mean and the standard deviation (Mean ± Stddev).

Similar results for FSS on an unbalanced holdout set are shown in Table 3. In the case where the size of the subset is constrained it is observed that the constrained EDA-

Table 4. Comparisons of the running times for constrained subsets on a Linux cluster (5 dual-processors at 1.2 Ghz)

Algorithm	# Features	# Evaluations	Average # Features	Time on 400AG.holdout.2000	Time on 400AG.holdout.50
NBM	150	68875	294.40	0 h 34 m	1 h 58 m
SBE	80	76960	275.98	0 h 36 m	2 h 09 m
	40	79380	269.48	0 h 37 m	2 h 11 m
NBM	150	67100	150	0 h 20 m	0 h 46 m
EDA	80	67100	80	0 h 09 m	0 h 21 m
	40	67100	40	0 h 05 m	0 h 11 m
LSVM	150	68875	294.40	2 h 15 m	2 h 38 m
SBE	80	76960	275.98	2 h 19 m	2 h 52 m
	40	79380	269.48	2 h 20 m	2 h 54 m
LSVM	150	67100	150	0 h 38 m	0 h 59 m
EDA	80	67100	80	0 h 17 m	0 h 27 m
	40	67100	40	0 h 14 m	0 h 19 m
PSVM	150	13875	296.26	9 h 11 m	62 h 02 m
SBE	80	15520	277.68	9 h 42 m	63 h 24 m
	40	16020	271.03	9 h 48 m	63 h 40 m
PSVM	150	13510	150	4 h 54 m	16 h 48 m
EDA	80	13510	80	2 h 48 m	9 h 38 m
	40	13510	40	1 h 52 m	6 h 16 m

The first two columns indicate the algorithm and the size of the constrained subset. The third column shows the number of model evaluations that is needed, calculated using formulas 10 and 12. The average number of features that has to be evaluated is shown in the fourth column and can be calculated with formula 11. The last two columns show the running time (in hours and minutes) that is needed for a balanced holdout set (400AG.holdout.2000) and an unbalanced holdout set (400AG.holdout.50)

approach performs comparable to the greedy method. A McNnemar statistical test with p=0.05 did not reveal any significant difference to prefer one method over the other on the basis of their q9 values. Furthermore the results of the EDA-approach can be obtained in much less time, showing the advantage of a simple EDA method like the UMDA.

To compare the running time (speed) of the greedy and heuristic algorithm, two aspects need to be considered : the number of model evaluations and the average number of features for a model evaluation. The number of model evaluations needed for both methods can be calculated using formulas 10 and 12. For the greedy algorithm the average number of features that has to be evaluated is calculated with formula 11, the heuristic algorithm with a constrained subset size has a fixed number of features. Table 4 shows the results of the time comparisons for constrained subsets. Clearly the EDA-approach needs considerably less model evaluations as the size of the feature subsets decreases, resulting in a faster feature selection algorithm. Furthermore, all models in the EDA-approach have a fixed number S of features to be trained on, whereas the greedy approach starts with the full feature set and gradually decreases the number of features until a set of S features remains. For both holdout sets it is observed that the EDA-approach provides a considerable speedup. The increase in speed depends on a number of things like e.g. complexity of the algorithm, size of the constrained subset and the holdout set. Whereas the increase in speed is not so much in the case of NBM on an unbalanced holdout set and a subset of 150 features, the speedup for the PSVM on the unbalanced holdout set for a fixed number of 40 features is about a factor 10.

These results support the view that the use of EDAs for feature selection provides a very practical approach when feature sets get larger or when very time-demanding algorithms (like e.g. PSVM) are used. In these cases the use of SBE becomes computationally infeasible and by tuning the EDA-parameters P, E and I, a relevant subset of S features can be discovered.

RELATED WORK

Genetic Algorithms (GAs) 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. It has to be pointed out that the EDA-UMDA approach is very similar to the compact GA (Harik *et al.*, 1998) or to a GA with uniform crossover.

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.

An extensive overview of splice site recognition, including new methods like Support Vector Machines can be found in Sonnenburg (2002), while a more general overview and a comparison of gene and splice site prediction is discussed in Mathé *et al.* (2002) and Zhang (2002).

CONCLUSIONS AND FUTURE WORK

The results displayed in this paper are showing that feature subset selection by estimation of distribution algorithms is able to select highly relevant features for splice site prediction. We presented a method that is scalable to larger feature sets and, when applied with a constraint on the size of the feature subset, provides a considerable gain in speed. This was obtained at no expense on efficiency, on the contrary. The method can be used for any optimisation problem where the feature set is sufficiently large, like e.g. gene selection in microarray datasets. Future research on splice site prediction will include position-independent information, possibly also structural information to achieve better results. Other future directions we would like to explore are the combination of EDAs with other 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 categorization. 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 Haussler,D. (ed.), *Proceedings of COLT*. ACN Press, pp. 144–152.

Boz,O. (2002) Feature subset selection by using sorted feature relevance. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA 2002)*.

Cantú-Paz,E. (2002) Feature subset selection by estimation of distribution algorithms. In Langdon,W.B., Cantú-Paz,E., Mathias,K., Roy,R., Davis,D., Poli,R., Balakrishnan,K., Honavar,V., Rudolph,G., Wegener,J., Bull,L., Potter,M.A., Schultz,A.C., Miller,J.F., Burke,E. and Jonoska,N. (eds), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Morgan Kaufmann, San Francisco, CA, pp. 754–761.

Degroeve,S., De Baets,B., Van de Peer,Y. and Rouzé,P. (2002) Feature subset selection for splice site prediction. *Bioinformatics*, **18**, 75–83.

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

Guyon,I., Weston,J., Barnhill,S. and Vapnik,V.N. (2000) Gene selection for cancer classification using support vector machines. *Machine Learning*.

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)*, Piscataway, IEEE Service Center, NJ, pp. 523–528.

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

Inza,I., Larrañaga,G., Etxebarria,R. and Sierra,B. (1999) Feature subset selection by Bayesian networks based on optimization. *Artificial Intelligence*, **27**, 143–164.

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

Kohavi,R. and John,G. (1997) Wrappers for feature subset selection. *Artificial Intelligence J.*, **97**, 273–324.

Kudo,M. and Sklansky,J. (2000) Comparison of algorithms that select features for pattern classifiers. *Pattern Recogn.*, **33**, 25–41.

Langley,P. and Sage,S. (1994) Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (1994)*. Morgan Kauffman, pp. 399–406.

Mathé,C., Sagot,M.F., Schiex,T. and Rouzé,P. (2002) Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res.*, **30**, 4103–4117.

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

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

Mühlenbein,H. (1996) The equation for response to selection and its use for prediction. *Evolutionary Computation*, **5**, 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 Klansky,J. (1988) On automatic feature selection. *Int. J. Pattern Recogn*, **2**, 197–220.

Snyder,E.E. and Stormo,G.D. (1995) Identification of protein regions in genomic DNA. *J. Mol. Biol.*, **248**, 1–18.

Sonnenburg,S. (2002) New Methods for Splice Site recognition, Diploma thesis, Humboldt-Universität zu Berlin.

Tishby,N., Pereira,F.C. and Bialek,W. (1999) The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computation*. pp. 368–377.

Vapnik,V.N. (1995) *The nature of statistical learning theory*. Springer.

Vafaie,H. and De Jong,K. (1993) Robust feature selection algorithms. In *Proceedings Fifth International Conference on Tools with Artificial Intelligence*. pp. 356–363.

Zhang,M.Q. (2002) Computational prediction of eukaryotic protein-coding genes. *Nat. Rev. Genet.*, **3**, 698–709.

Zhang,C-T. and Zhang,R. (2002) Evaluation of gene-finding algorithms by a content-balancing accuracy index. *J. Biomol. Struct. Dyn.*, **19**, 1045–1052.