

# Feature relevance estimation by evolving probabilistic dependency networks with weighted kernel machines \*

Nestor Rodriguez 

**Advisor:** Sergio A. Rojas, PhD.

Friday 18<sup>th</sup> October, 2013

## Abstract

This thesis proposal focuses on the problem of estimating relevance of observed variables for a classification task in high dimensional spaces, which is known as feature subset selection or feature relevance determination by the machine learning community. The main goal of the thesis is the design of a novel feature relevance estimation method by combining techniques for estimation of probabilistic dependency networks with weighted kernel machines<sup>†</sup>. The method is intended to work within a population-based stochastic search framework where relevant (but unknown) variables from a given dataset<sup>§</sup> are found by iteratively evolving a set of relevance estimation candidates. These candidates will consist of parameters of multivariate conditional probability distributions. The distributions could be seen as dependency networks of how variables influence each other. With this information a subset of the most relevant features from the original sample can be selected to perform classification, the suitability of the subset being assessed as its predictive classification accuracy. Weighted kernel classifiers would be used for this purpose. It is expected that the method may provide additional information about dependency among such variables as a byproduct of the selection process.

---

\* A MSc thesis proposal

<sup>†</sup> Engineering School, District University of Bogota, Colombia

<sup>‡</sup> The topics of feature selection methods, estimation of probability distribution algorithms and kernel machines are briefly reviewed in section 6. The interested reader is referred to [9, 11, 26] for details.

<sup>§</sup> In the context of this paper *dataset* means data arranged in a tabular way where columns represent variables and rows represents instances. Other types of formats (such as text, images, graphs, etc.) are not considered.

# Contents

<b>1</b>	<b>Problem, Motivation and Research Hypothesis</b>	<b>4</b>
1.1	Problem and Motivation . . . . .	4
1.2	Research Hypothesis . . . . .	7
<b>2</b>	<b>Idea and Proposal</b>	<b>8</b>
<b>3</b>	<b>Goals</b>	<b>10</b>
<b>4</b>	<b>Methodology</b>	<b>11</b>
4.1	Literature Review . . . . .	11
4.2	Algorithm Design . . . . .	11
4.3	Algorithm Implementation . . . . .	11
4.4	Experimentation . . . . .	11
4.5	Analysis of Results . . . . .	12
4.6	Conclusions and Documentation . . . . .	12
<b>5</b>	<b>Timeline</b>	<b>13</b>
<b>6</b>	<b>Literature Review</b>	<b>14</b>
6.1	Feature Selection Techniques . . . . .	14
6.1.1	The problem of feature selection . . . . .	14
6.1.2	Filter Approach . . . . .	14
6.1.3	Wrapper Approach . . . . .	15
6.1.4	Embedded Approach . . . . .	15
6.2	Machine Learning . . . . .	15
6.2.1	Kernel Machines . . . . .	15
6.2.2	Linear Learning Machines . . . . .	16
6.3	Evolutionary Estimation of Probabilistic Distributions . . . . .	17
6.3.1	Univariate EDAs . . . . .	17
6.3.2	Multivariate EDAs . . . . .	19
6.4	Feature Selection Techniques Using Estimation of Distribution Algorithms	20

## List of Algorithms

1	Preliminary pseudocode of the expected method described in this proposal .	9
2	The Perceptron . . . . .	17
3	The Compact Genetic Algorithm . . . . .	19

## List of Figures

1	Correlation analysis of input variables in relation to overweight conditions of insurance customers. . . . .	4
2	Using relevant features customers can be easily segmented with a simple linear function. . . . .	5
3	The components involved in this thesis proposal. . . . .	8
4	Preliminary depiction of the expected method described in this proposal.	9

5	Timeline for the thesis proposal. . . . .	13
6	A feature selection taxonomy. . . . .	15
7	Transformation from input space to a feature space simplifies the classification task. . . . .	16
8	Univariate Estimation of Distribution Algorithm flowchart. . . . .	18
9	Diagram of probability models used in most popular multivariate EDAs. . . . .	19
10	Main components of the FSS-EBNA algorithm. . . . .	20
11	Main components of the wKIERA algorithm. . . . .	20

# 1 Problem, Motivation and Research Hypothesis

## 1.1 Problem and Motivation

Feature subset selection (FSS) is a machine learning technique for dimensionality reduction in datasets where irrelevant or noisy variables may appear. It is useful in problems where few from a big set of observed variables explain a given pattern in classification, prediction or clustering of data cases (e.g. diagnosis of a disease, market analysis, image segmentation, etc.), and therefore it allows for an expert to focus on data that is really important, reducing the expenditure of costly or time-consuming experiments.

To illustrate the problem, consider the situation where an insurance company is evaluating the factors that have an impact on people's overweight conditions that lead to claims in illness insurance policies. Upon registration, they collect information of both parents' weight and birthday. They want to know which of these variables are related to the development of the condition. Now let us suppose that one-year time is required by an insurance analyst to process the information of just one variable. Then it would be desirable to discard irrelevant variables in order to save him valuable time. An FSS method could help to select the relevant ones.

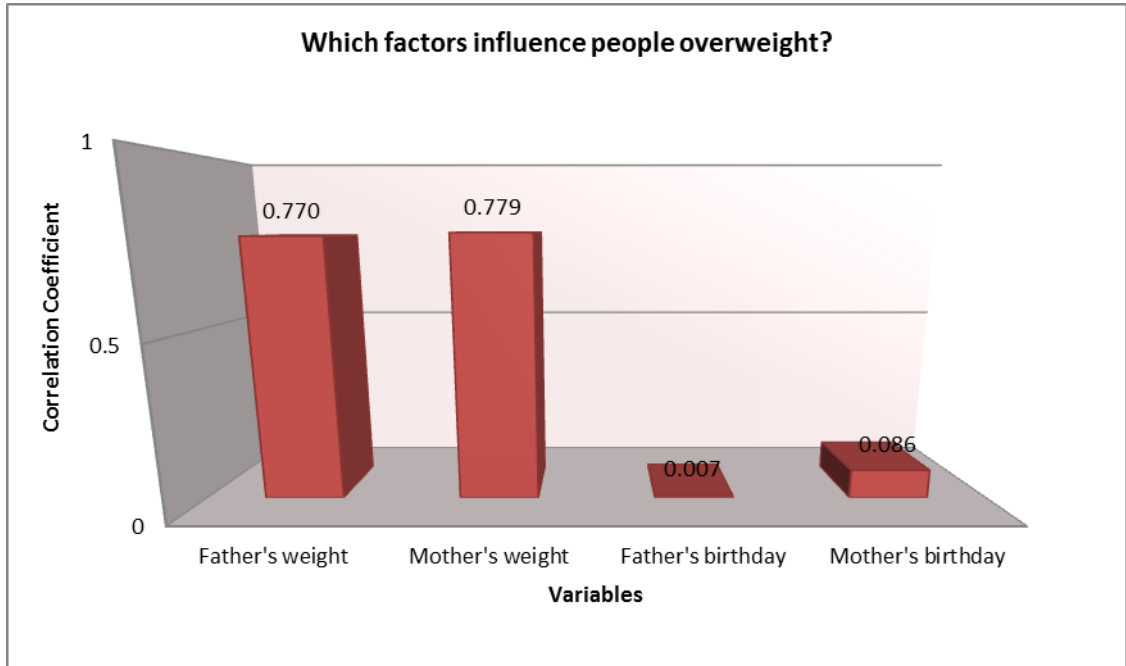


Figure 1: Correlation analysis of input variables in relation to overweight conditions of insurance customers (for the fictional example in the text).

Figure 1 depicts the result of applying a correlation-based feature selection method on this fictional problem; it is evident that half of the variables, namely those related to birth dates, are irrelevant and can be ignored for the problem of interest. Thus, dimensionality reduction may provide insights in data mining tasks, yielding a better understanding of objects or phenomena behavior and is useful to speed up data analysis and to improve generalization performance. For instance, in the previous insurance company example,

Figure 2 shows how sample cases can be now rendered in the 2D space obtained with the relevant variables; they can be easily discriminated with a simple hyperplane. By focusing on those two relevant variables, the analyst would have saved two precious years of working time and the company two equally valuable years of salary payments.

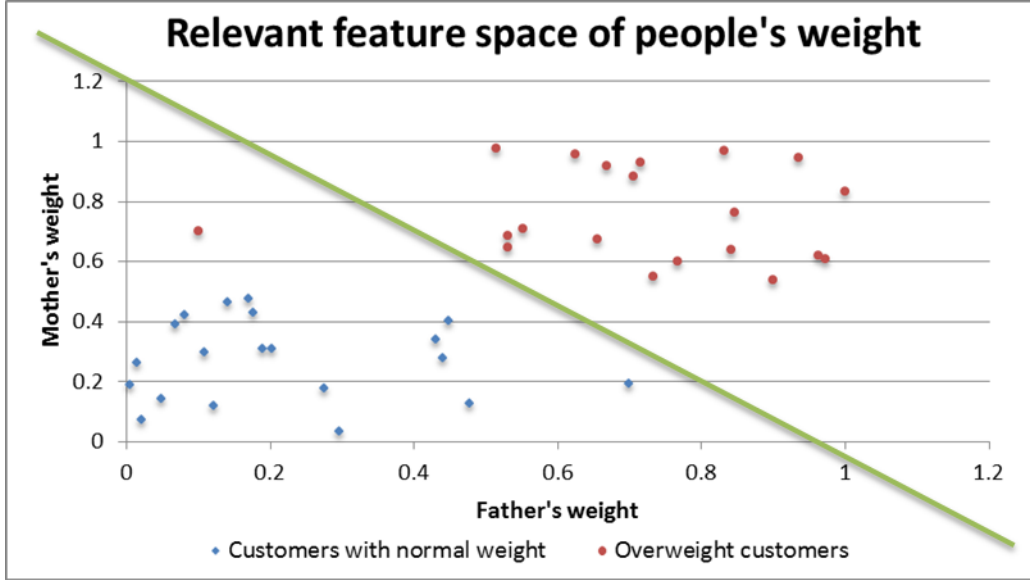


Figure 2: Using relevant features customers can be easily segmented with a simple linear function.

Some FSS techniques assume variables are independent. This assumption might not be appropriate for some real world problem since variables may influence others and these interactions are usually hidden. A good example is in the field of bioinformatics where experts analyze proteins relevance in diseases to design inhibitory vaccines; using state-of-the-art high-throughput technologies (e.g. mass-spectrometry [12]) they are able to collect large amounts of data about activation of protein mechanisms, but the information about how these interactions occur is unknown or difficult to obtain, not to mention that just a few proteins from a large proteomic spectrum would be related to the activation of the disease.

The independent assumptions (univariate FSS methods, see section 6.1) are very popular because of their low computational cost [18] but they do not provide additional insights about data relationships. On the other hand, missing information about those dependencies may affect negatively the prediction accuracy of the feature subset. As an alternative approach, multivariate FSS methods search for feature subsets and possible dependency relationships between them at the cost of an increase in computational complexity (recall that FSS is a combinatorial problem known to be NP-Hard [9, 20]). The challenge is then to design novel feature selection methods that take advantage of multivariate power combined with high-accuracy classifiers to obtain improved prediction and explanatory performance.

On the other hand, kernel classifiers have recently emerged as powerful high-accuracy classifiers with robust generalization abilities [26]. They use linear functions to perform classification, allowing for non-linear discriminatory borderlines in the input space by means of a kernel function. This function is a mapping of similarity measures in a

transformed space where nonlinearities can be solved linearly. Two widely used kernel functions are the RBF kernel and the polynomial kernel (Eq.(1) and Eq.(2) respectively):



$$K_{\sigma}(\bar{x}, \bar{z}) = \exp\left(-\sigma \sum_k (x_k - z_k)^2\right) \quad (1)$$

and

$$K_d(\bar{x}, \bar{z}) = \langle \bar{x}, \bar{z} \rangle^d, \quad (2)$$

where  $\bar{x}, \bar{z} \in X$ ,  $X \subseteq \mathbb{R}^D$  represents the input space and  $D$  the number of variables or dimensions. The parameter  $\sigma$  and  $d$  define the width of the RBF and polynomial kernel respectively. Modified weighted versions of these kernels introduce scale factors  $\bar{w} = \{w_1 \dots w_{\ell}\}$  to weight the amount that each dimension contributes to the total computation [4]. The weighted RBF and weighted polynomial kernels are then defined as Eq.(3) and (4):

$$K_{\sigma}(\bar{x}, \bar{z}) = \exp\left(-\sigma \sum_k^{\ell} w_k (x_k - z_k)^2\right) \quad (3)$$

and

$$K_d(\bar{x}, \bar{z}) = \left(\sum_k^{\ell} w_k (x_k \cdot z_k)\right)^d. \quad (4)$$

Some FSS methods have been proposed in connection with kernel functions, for instance the weighted kernels in Eq. (3) and (4) have been used to define a FSS embedded method (the weighted kernel iterative estimation of relevance algorithm, wKIERA [21]). The idea in this case is to consider the weight vector  $\bar{w}$  of the kernel as a relevancy estimator of input variables. The vector is tuned by a univariate estimation of distribution algorithm [3] which iteratively refines the distribution by a population-based technique inspired in genetic algorithms (GA) [8]. The accuracy of the kernel classifier coupled with the weight vector candidates is taken as the fitness measure of the GA. The authors showed promising results of this algorithm in selecting relevant variables in a number of different classification tasks, including problems with linear and non-linear hypothesis targets. This FSS method however, is designed on the basis of the independent assumption mentioned above and consequently does not provide the power of explanation of multivariate methods and does not account for information of relationship between the variables. One of the motivations of this thesis proposal is to build upon this method and design a refined version that incorporates techniques for multivariate feature selection within an embedded population-based stochastic algorithm. Dependency networks, also known as belief or Bayes networks [11], would be preliminary chosen to represent how variables depend and influence each other, and one of the goals of the thesis is to design and new algorithms to perform relevance estimation within such framework and to make feasible the approach.

Finally, it is worth to note that the work by [17] propose a FSS multivariate method that accounts for dependencies between variables by estimating belief networks. Their

approach however is a wrapper method (see section 6.1) and they do not incorporate the powerfulness of nonlinear kernel classifiers. We will however review this study, and as far as possible compare the different approaches.

## 1.2 Research Hypothesis



We will consider the following research hypothesis: Given a set of observations taken from a particular phenomenon where dependencies among variables exist but are hidden, it is feasible to select a subset of variables that are relevant in a classification task with a method that combines iterative estimation of dependency networks coupled with weighted-kernel classifiers, in order to produce a predictor model that improves the understanding of the problem domain when compared with other techniques based on independence assumptions.

## 2 Idea and Proposal

The new FSS method will involve the following main components (see Figure 3). The *data*, which is the sample of observed variables for a given problem; the overall goal of the method will be to identify relevant feature subsets and as much as possible information about dependencies that explain significant patterns hidden in the data. The *dependency network model* represents the relationships among variables and how they affect each others.

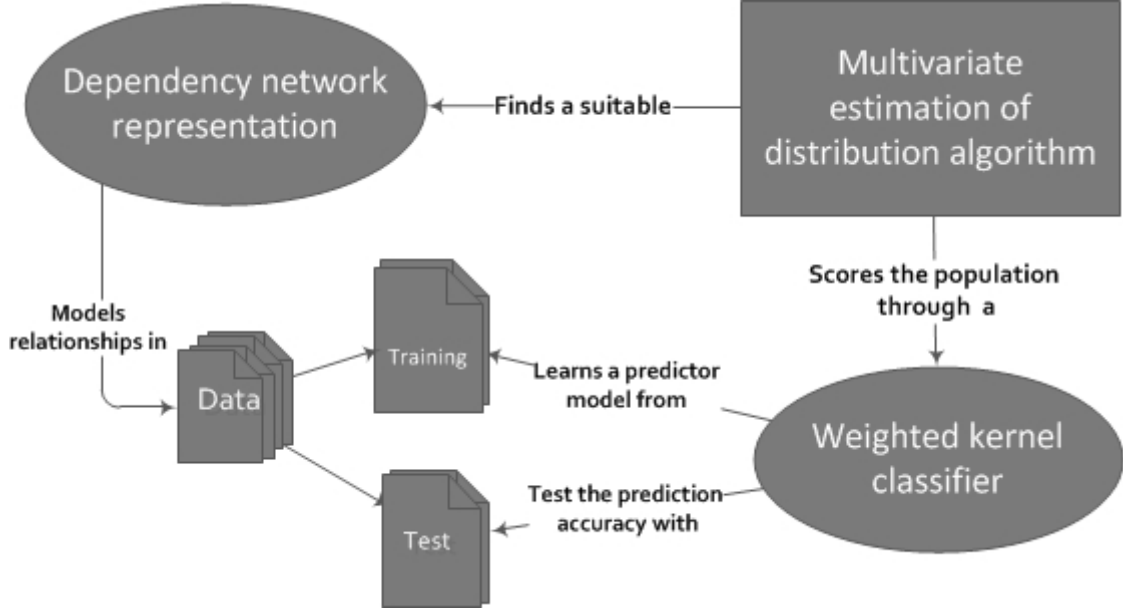


Figure 3: The components involved in this thesis proposal.

The *multivariate estimation of distribution algorithm* is a technique to infer the parameters and structure of the dependency network by estimating a probability distribution of relevancy from a pool of candidate weight vectors; the distribution and candidates are adjusted within a framework of a stochastic population-based evolutionary algorithm. Lastly, the *weighted kernel classifier* will perform the iterative classification using relevant feature subsets according to the evolved relevance distributions.

A draft schematic flowchart of the conceived method is shown in Figure 4. Preliminary pseudo-code of the method is summarized in Algorithm 1.



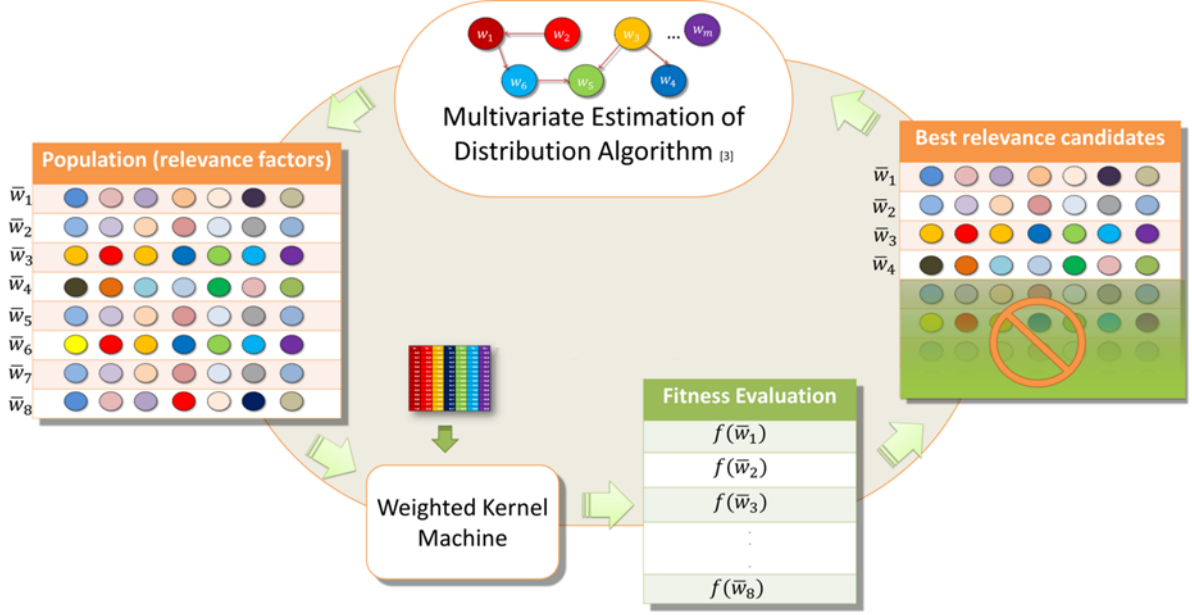


Figure 4: Preliminary depiction of the expected method described in this proposal.

---


**Algorithm 1** Preliminary pseudocode of the expected method described in this proposal

---

**Inputs:** Given a dataset  $\mathcal{D}$ , a weighted kernel  $\kappa_\omega$  and a classifier  $\mathcal{A}$

Let  $\beta$  represents a dependency network distribution initialized with an independent joint distribution.

**repeat**

Split  $\mathcal{D}$  in training  $\mathcal{D}_t$  and testing  data

$\bar{\Omega} \leftarrow$  Sample  $k$  candidates from  $\beta$  

**for**  $\omega_j \in \bar{\Omega}$  **do**

Train classifier:  $h_j \leftarrow \mathcal{A}(\mathcal{D}_t, \kappa_{\omega_j})$

Test classifier:  $e_j \leftarrow \text{error}(h_j, \mathcal{D}_s, \kappa_{\omega_j})$

**end for**

$\bar{\Omega}' \leftarrow \text{bestCandidates}(\bar{\Omega}, \bar{e})$

Re-estimate dependency network:  $\beta \leftarrow \text{reEstimate}(\bar{\Omega}')$

**until**  $\beta$  has converged or maximum number of iterations reached

---

### 3 Goals



- Design an algorithm for feature relevance estimation by estimating dependency networks combined with weighted kernel classification methods.
- Verify the feasibility of the algorithm in high dimensional feature spaces using toy and real datasets (e.g bioinformatics).
- Compare algorithm performance with respect to other feature selection methods (e.g. score-based filters, stochastic population-based wrappers and weighted-kernel-based embedded methods).



## 4 Methodology

This section explains the intended methodology for this thesis.

### 4.1 Literature Review

In this stage a review of the state-of-the-art in population-based stochastic search methods for feature selection, estimation of dependency networks and weighted kernel classifiers will be carried out. The outcome of this stage will be a literature review report.

### 4.2 Algorithm Design

In this stage we will study in detail the elements described in Figure 3, 4, and Algorithm 1 in order to provide the final design of the proposed method, according to the goals stated in Section 3.

### 4.3 Algorithm Implementation

In this stage we will choose the development tools (programming language and developing environment, source code management) and carry out the implementation of the algorithm, along with its documentation and a supporting website. The outcome of this stage will be a ready-to-use computer program.

### 4.4 Experimentation

In this stage we will test the program using toy and real datasets in controlled conditions. We will also compare its performance with other feature selection techniques. The experiments will involve the following tasks:

1. Dataset preparation: This task comprises the creation of synthetic tabular datasets with different testing properties (number of dimensions, variable dependencies, linear or non-linear concepts, etc) and search and preparation of real problem datasets.
2. Experimental design: This task comprises the design of different scenarios and hypothesis to test (input parameters, performance measures, etc.).
3. Experiments execution: This task comprises setting up the experimental platform, implementation of required algorithms, running up the designed experiments and collections of results in a suitable form for interpretations (tables, figures, etc.).

## **4.5 Analysis of Results**

In this stage the results obtained in the previous stages will be analyzed and the the feasibility of the algorithm will be discussed in view of the proposed research hypothesis and goals.

## **4.6 Conclusions and Documentation**

This stage involves writing up the final technical report including conclusions and views derived from the research work that was conducted and recommendations for future work.

## 5 Timeline

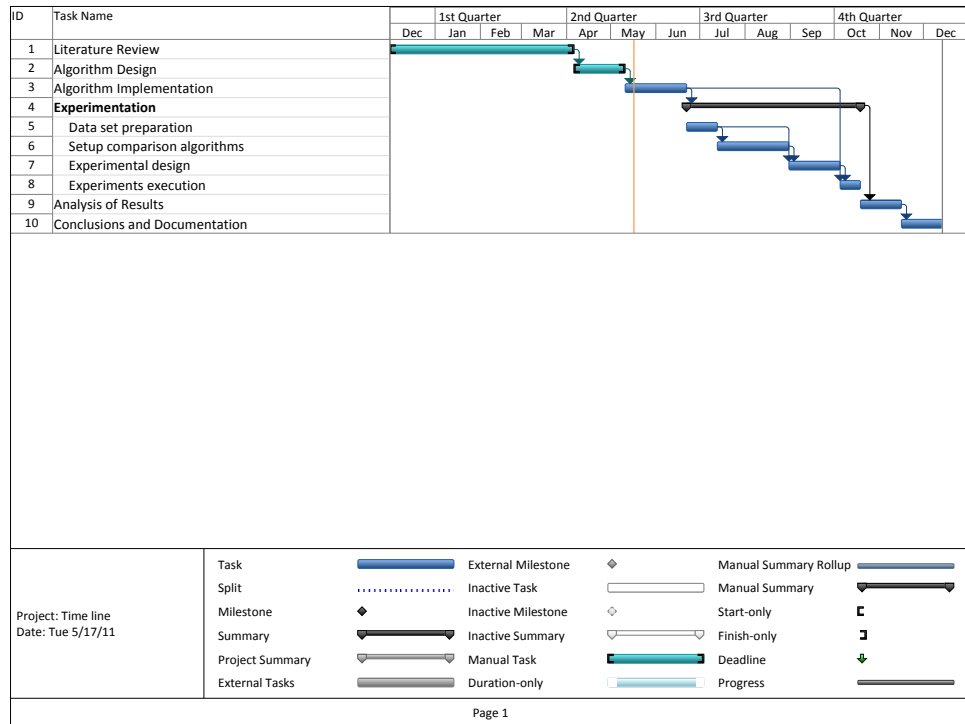


Figure 5: Timeline for the thesis proposal.

## 6 Literature Review

In this section we provide a summarized review of the main elements involved in this proposal, namely feature selection techniques, kernel classification machines, and estimation of distribution algorithms.

### 6.1 Feature Selection Techniques

#### 6.1.1 The problem of feature selection

Nowadays the advances in technologies for data collection (the genome project, particle colliders, internet-based social networks) pose increased challenges for data analysis due to larger sizes and higher dimensionality of the observed samples. It is reasonable however to assume that the collected variables may exhibit redundancy, inconsistency, noisy and irrelevant data and therefore will be difficult to analyze by the human eye. New automated mechanisms are required to identify significant variables for pattern discovery and data mining.

Feature selection is gaining in importance and has recently become an active field of research in disciplines such as knowledge discovery, machine learning, pattern recognition, bioinformatics, geoinformatics, etc. While FSS are techniques for dimensionality reduction, they maintain the original variables compared to other entropy-based, data compression or statistical methods that modify the original data representation instead of providing a subset of significant features with useful information for the domain experts [24]. The main goal of these techniques is to obtain a better understanding of the data for visualization purposes or to speed up data analysis.

FSS techniques are helpful in improving prediction models for supervised learning, detecting dimensions of tight data conglomeration in clustering tasks and a deeper understanding of process for data generation. Nonetheless, the design of novel FSS techniques aimed to provide more useful information also incurs in new levels of complexity giving rise to new challenges pertaining to feasible and efficient computational techniques.

In the classification context, feature selection techniques are categorized in three main groups depending on how they interact with a classification model (see Figure 6): filters, wrappers and embedded. A detailed description of this categorization is given below (based on the study in [18]).

#### 6.1.2 Filter Approach

Methods in this category assess the relevance of variables based on the intrinsic properties of data. The search in feature space is separated from the search in the space of classification hypothesis. In most cases a feature relevance score is calculated (i.e. using mutual-information or data correlation) and low-scoring features are removed. These methods have the ability to scale to higher dimensions since they exhibit a low computational cost.

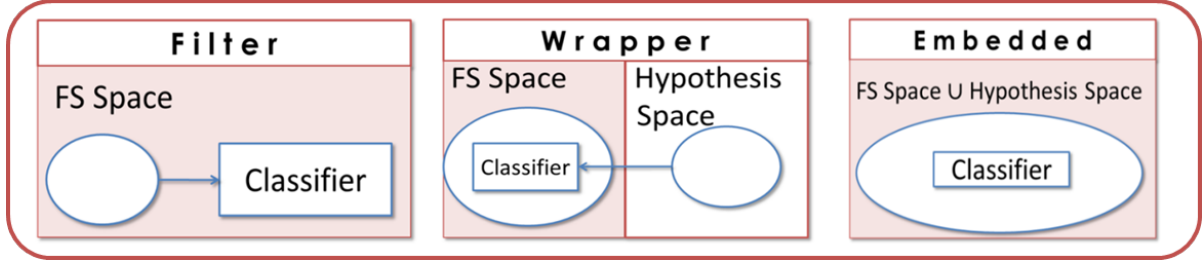


Figure 6: A feature selection taxonomy. The feature selector category depends on how the feature and hypothesis space are combined during classifier learning (left and right box taken from [24]).

### 6.1.3 Wrapper Approach

Wrappers establish a symbiotic relationship between the feature subset search and the classification algorithm. The aim of these type of methods is not only to find relevant features but also to find the best suited classification model to those features. However the interaction with the classifier may induce new problems such as a higher risk of overfitting and a higher demand of computational resources.

### 6.1.4 Embedded Approach

Embedded methods incorporate the feature selection and the classification model construction as a single process. As a result feature and hypothesis spaces are combined providing a richer source of information during the search although incurring as well in additional computational costs.

## 6.2 Machine Learning

Supervised learning aims to find optimal learning algorithms with high generalization and prediction performance to produce classification models of the training set of examples and their associated labels. The prediction accuracy of the resulting classifier is then evaluated with a test set. Among these techniques, linear classifiers are widely used because of their theoretical simplicity and computational efficiency. Application of linear classifiers, in real world problems where nonlinearities arise, has been possible due to the advances in kernel machines [26].

### 6.2.1 Kernel Machines

These algorithms use kernel functions to compute similarity measures of the input samples in a feature space where nonlinearities might be easier to solve by linear classifiers. Figure 7 shows an example of feature mapping where in the original space data can only be separated with a nonlinear function but becomes linearly separable if the data is

transformed into a feature space. When using the linear classifier the feature space mapping is implicit since the machine only uses inner products of the transformed examples which are computed using kernel function.

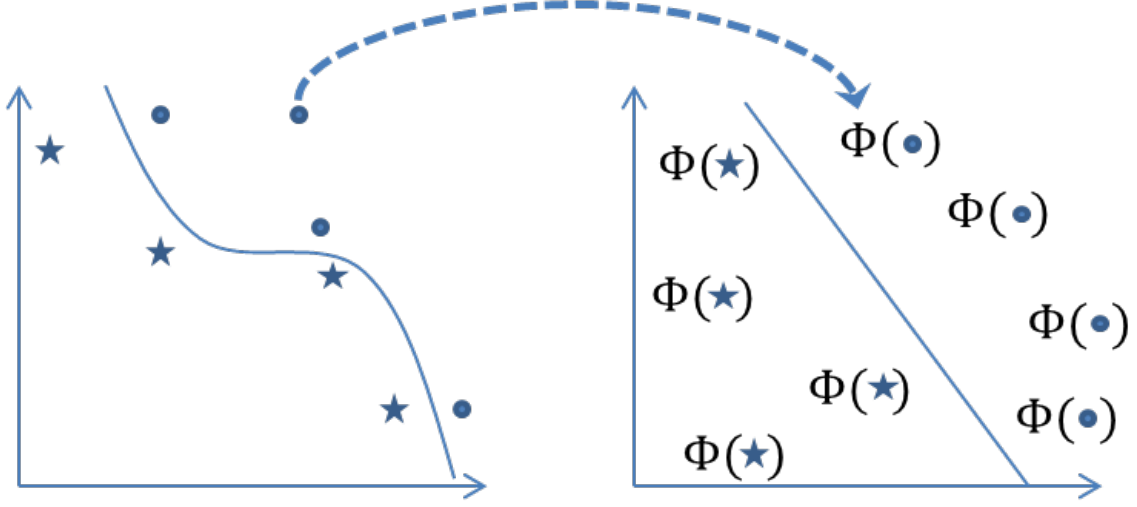


Figure 7: Transformation from input space to a feature space simplifies the classification task (taken from [26]).

A kernel is a function  $K$  such that for all  $\bar{x}, \bar{z} \in X$

$$K(\bar{x}, \bar{z}) = \langle \phi(\bar{x}), \phi(\bar{z}) \rangle,$$

where  $\phi$  is a mapping from the input space  $X$  to an (inner product) feature space. In order to be a kernel, the function must comply with the conditions defined by Mercer's theorem [26]. Eq. (1-4) are examples of kernel functions.

### 6.2.2 Linear Learning Machines

A linear classifier is a linear function  $f(\bar{x})$  where  $\bar{x} \in X$  and  $\bar{w} \in \mathbb{R}^D$ , that can be written as:

$$f(\bar{x}) = \text{sgn}(\langle \bar{w}, \bar{x} \rangle) = \text{sgn}\left(\sum_i w_i x_i\right) \quad (5)$$

A given sample  $\bar{x}$  is assigned to the positive class if  $f(\bar{x}) = 0$  or otherwise to the negative class. There are number of training algorithms to learn a classification vector  $\bar{w}$ , including the perceptron [22] and the linear support vector machine [6]. For illustration purposes, Algorithm 2 shows the learning procedure of the perceptron.

The kernelized version of the linear classifier allows for classification of nonlinearly separable datasets, as mentioned before. The classification function is written consequently as:



$$f(\bar{x}) = \text{sgn}\left(\sum_k \alpha_k K(\bar{x}_k, \bar{x})\right), \quad (6)$$

where  $K$  represents a kernel function and  $\alpha_k$  the classifier parameters that can be learned in this case using the kernel perceptron [7] or the support vector machine [5, 6].

---

**Algorithm 2** The Perceptron

---

**Inputs:** Given a linearly separable training set  $S$  and learning rate  $\eta \in \mathbb{R}^+$

$w \leftarrow 0$ ;  $k \leftarrow 0$

**repeat**

**for**  $i = 1, 2, \dots, \ell$  **do**

**if**  $y_i(\langle w_k, x_i \rangle) \leq 0$  **then**

$w_{k+1} \leftarrow w_k + \eta y_i x_i$

$k \leftarrow k + 1$

**end if**

**end for**

**until** no mistake made within the *for* loop

**Outputs:**  $(w_k)$  where  $k$  is the number of mistakes

---

### 6.3 Evolutionary Estimation of Probabilistic Distributions

Genetic algorithms (GA) are search stochastic methods inspired in the theory of natural selection of Darwin. The idea is to evolve a population of candidates coding the parameters for the solution of an optimization problem, using genetic operations such as chromosome recombination and mutation [8]. A novel technique known as Estimation of Distribution Algorithms (EDAs) has recently emerged motivated by GAs but from a statistical viewpoint. They have proven to be better suited in many applications than canonical GAs [3].

The main distinctive aspect of EDAs is that they search for a probabilistic distribution model representing the population of candidates. Instead of using genetic operations, these algorithms are based on well-known statistical techniques to estimate the parameters of a distribution function, and the evolution is guided by sampling the evolving probabilistic model. The complexity of the algorithm lies in the robustness of this probability model and in how it is iteratively re-estimated. The probabilistic models can be as simple as a marginal distribution or as complex as a joint multivariate distribution expressing high order interactions among the observed variables. These categories are briefly described below following the review in [18].

#### 6.3.1 Univariate EDAs

Univariate algorithms use a marginal probability model encoded in a probability vector. They are not computational intensive because they assume no interaction between variables. The probability vector is re-estimated using the fittest subset of the population

of candidates, as depicted in Figure 8. There are three widely used algorithms for this category: Univariate Marginal Distribution Algorithm (UMDA [13]), Population-based Incremental Learning (PBIL [2]) and the Compact Genetic Algorithm (cGA [1, 10]).

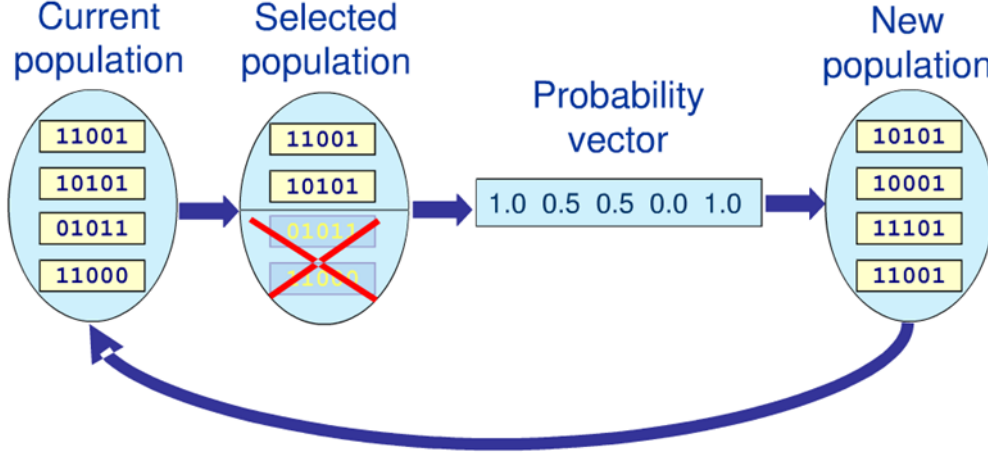


Figure 8: Univariate Estimation of Distribution Algorithm flowchart.

UMDA estimates the entire probability vector every iteration. The probability distribution is factorized as a product of independent univariate marginal distributions, which are estimated from marginal frequencies. The probability of the  $i$ -th variable being relevant is given by the Eq.(7), where  $S$  is the subpopulation of  $N$  fittest candidates, and  $\delta$  is the Kronecker delta.

$$p(x_i) = \prod_j \frac{\sum_j^N \delta(X_i = x_i | S)}{N} \quad (7)$$

PBIL was designed to work in a  $n$ -dimensional binary space  $\{0, 1\}^n$ . Unlike UMDA, PBIL does not estimate a new probability vector every iteration  $t$ . Instead, fittest candidates are chosen to update the probability vector at a learning rate  $\alpha = (0, 1]$ . The updating rule is shown in Eq.(8). Notice that PBIL becomes UMDA when  $\alpha = 1$ .

$$p_t(x_i) = (1 - \alpha)p_{t-1}(x_i) + \alpha \frac{1}{N} \sum_k^N S_{ki} \quad (8)$$

Lastly, cGA has become popular for the higher efficiency to solve very large scale problems with millions to billions of variables with a lower computational demand than canonic GA [25]. cGA has a low memory consumption because only two candidates per iteration are generated. Both compete and the winner updates the probability vector at a learning rate  $1/n$  where  $n$  is the population size parameter. Algorithm 3 shows the pseudocode of the cGA .

---

**Algorithm 3** The Compact Genetic Algorithm

---

**Inputs:** Population size  $n$  and chromosome length  $\ell$

$p \leftarrow$  initialize a uniform probability vector.

**repeat**

$[a, b] \leftarrow \text{generateTwoIndividuals}(p)$

$[winner, loser] \leftarrow \text{compete}(a, b)$

**for**  $i = 1, 2, \dots, \ell$  **do**

**if**  $winner[i] \neq loser[i]$  **then**

**if**  $winner[i] = 1$  **then**

$p[i] \leftarrow p[i] + \frac{1}{n}$

**else**

$p[i] \leftarrow p[i] - \frac{1}{n}$

**end if**

**end if**

**end for**

**until**  $p$  has converged

**Outputs:**  $p$  represents the final solution

---

### 6.3.2 Multivariate EDAs

Multivariate EDAs use joint statistics models of higher order to represent interaction among variables. These models are usually represented as probabilistic graphical models (see Figure 9). As mentioned above, EDA's complexity increases when a higher order of interaction among variables are desired. Factorized Distribution Algorithm (FDA [14]), Estimation of Bayesian Networks Algorithm (EBNA [15]) and Bayesian optimization algorithm (BOA [19]) belong to this category. BOA and EBNA both use Bayesian network structures but differs in the score metric they use to select the appropriate network structure. BOA uses Bayesian Dirichlet equivalence score (BDe) while EBNA uses K2+Penalization and Bayesian Information Criterion (BIC). Similarly, FDA uses Boltzmann selection for Boltzman distribution.

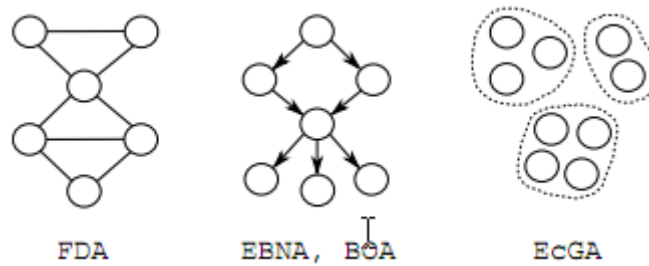


Figure 9: Diagram of probability models used in most popular multivariate EDAs( taken from [18]).

## 6.4 Feature Selection Techniques Using Estimation of Distribution Algorithms

Estimation of Distribution Algorithms for feature subset selection and feature relevance estimation has been proposed with promising results in large scale domains such as bioinformatics [16, 23]. Among these techniques, particularly FSS-EBNA [17] and wKIERA [21] have been used for selection of relevant feature subsets. FSS-EBNA uses a multi-variate estimation of distribution algorithm (EBNA) as the search engine for exploring the feature space and a Naive-Bayes classifier (NB) to predict the class for each instance; Figure 10 illustrates the main components for this algorithm.

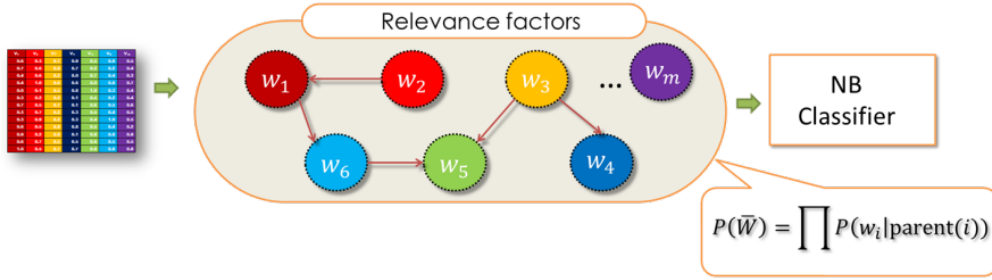


Figure 10: Main components of the FSS-EBNA algorithm.

On the other hand wKIERA uses a univariate estimation of distribution algorithm for searching the best suited representation of features relevance and a RBF weighted kernel machine with a perceptron classifier as the learning algorithm; Figure 11 illustrates the main components of this algorithm.

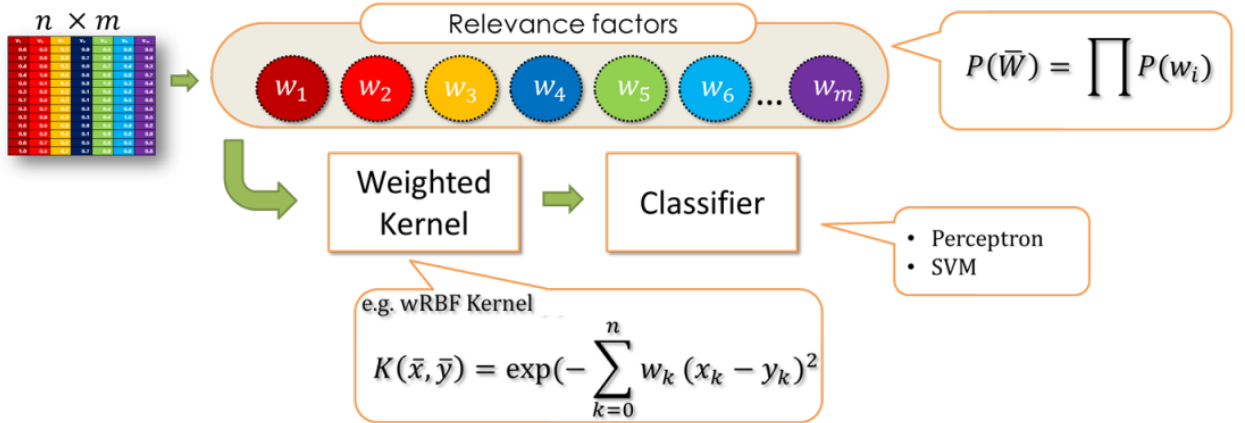


Figure 11: Main components of the wKIERA algorithm.



## References

- [1] Shumeet Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. In *The Proceedings of the 12th Annual Conference on Machine Learning*, pages 38 – 46. Morgan Kaufmann Publishers, 1995.
- [2] Shummet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Carnegie Mellon University Pittsburgh, PA, USA, Pittsburgh, PA, USA, 1994.
- [3] Endika Bengoetxea, Pedro Larrañaga, Isabelle Bloch, and Aymeric Perchant. Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 2134 of *Lecture Notes in Computer Science*, pages 454–469. Springer Berlin / Heidelberg, 2001.
- [4] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002. 10.1023/A:1012450327387.
- [5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. 10.1007/BF00994018.
- [6] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines: and other kernel-based learning methods*. Cambridge University Press, 1 edition, March 2000.
- [7] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296, December 1999.
- [8] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [9] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Machine Learning*, 3:1157–1182, March 2003.
- [10] Georges Harik, Fernando G. Lobo, and David E. Goldberg. The compact genetic algorithm. In *IEEE Transactions on Evolutionary Computation*, pages 523–528, 1998.
- [11] David Heckerman. *A tutorial on learning with Bayesian networks*, pages 301–354. MIT Press, Cambridge, MA, USA, 1999.
- [12] Mary S. Lipton and Ljiljana Pasa-Tolic. *Mass Spectrometry of Proteins and Peptides: Methods and Protocols*. Humman Press, 2008.

- [13] H. Muhlenbein and G. Paag. From recombination of genes to the estimation of distributions: I. binary parameters. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature PPSN IV*, volume 1141 of *Lecture Notes in Computer Science*, pages 178–187. Springer Berlin / Heidelberg, 1996.
- [14] Heinz Muhlenbein, Thilo Mahnig, and Alberto Ochoa Rodriguez. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5:215–247, July 1999.
- [15] Pedro Larra naga, Ramon Etxeberria, Jose A. Lozano, and Jose M. Pe na. Combinatorial optimization by learning and simulation of bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 343–352. Morgan Kaufmann, 2000.
- [16] Iñaki Inza, Marisa Merino, Pedro Larra naga, Jorge Quiroga, Basilio Sierra, and Marcos Giralá. Feature subset selection by genetic algorithms and estimation of distribution algorithms. a case study in the survival of cirrhotic patients treated with tips. *BioData Mining*, 2000.
- [17] Iñaki Inza, Pedro Larra naga, Ramon Etxeberria, and Basilio Sierra. Feature subset selection by bayesian network-based optimization. *Artificial Intelligence*, pages 157–184, 2000.
- [18] Rubn Arma nanzas, Iñaki Inza, Roberto Santana, Yvan Saeys, Jose Luis Flores, Jose Antonio Lozano, Yves Van De Peer, Rosa Blanco, Vctor Robles, Concha Bielza, and Pedro Larra naga. A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, 2008.
- [19] Martin Pelikan, David E. Goldberg, and Erick Cantu-Paz. Boa: The bayesian optimization algorithm. In *Genetic and Evolutionary Computation Conference (GECCO-1999)*, pages 525–532. Morgan Kaufmann, 1999.
- [20] Sergio Rojas and Delmiro Fernandez-Reyes. Adapting multiple kernel parameters for support vector machines using genetic algorithms. In *2005 IEEE Congress on Evolutionary Computation (CEC-2005)*, 2005.
- [21] Sergio Rojas, Emily Hsieh, Dan Agranoff, Sanjeev Krishna, and Delmiro Fernandez-Reyes. Estimation of relevant variables on high-dimensional biological patterns using iterated weighted kernel functions. *PLoS ONE*, 3:e1806, 03 2008.
- [22] Frank Rosenblatt. *The perceptron: a probabilistic model for information storage and organization in the brain*, pages 386–408. American Psychological Association, 1956.
- [23] Yvan Saeys, Sven Degroeve, Dirk Aeyels, and Yves Van De Peer. Fast feature selection using a simple estimation of distribution algorithm: A case study on splice site prediction. *Bioinformatics*, 19, 2003.
- [24] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23:2507–2517, September 2007.

- [25] Kumara Sastry, David E. Goldberg, and Xavier Llorca. Towards billion bit optimization via parallel estimation of distribution algorithm. In *Genetic and Evolutionary Computation Conference (GECCO-2007)*, pages 577–584, 2007.
- [26] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.