Learning from positive and unlabeled examples by enforcing statistical significance

Pierre Geurts
Department of EE and CS & GIGA-R, University of Liège, Belgium

Abstract

Given a finite but large set of objects described by a vector of features, only a small subset of which have been labeled as 'positive' with respect to a class of interest, we consider the problem of characterizing the positive class. We formalize this as the problem of learning a feature based score function that minimizes the p-value of a non parametric statistical hypothesis test. For linear score functions over the original feature space or over one of its kernelized versions, we provide a solution of this problem computed by a one-class SVM applied on a *surrogate dataset* obtained by sampling subsets of the overall set of objects and representing them by their average feature-vector shifted by the average feature-vector of the original sample of positive examples. We carry out experiments with this method on the prediction of targets of transcription factors in two different organisms, *E. Coli* and *S. Cererevisiae*. Our method extends enrichment analysis commonly carried out in Bioinformatics and its results outperform common solutions to this problem.

1 MOTIVATION

Machine learning algorithms are being applied successfully in a growing number of diverse domains. Standard supervised learning methods assume that the training set is a sample of input-output pairs i.i.d. from some probability distribution, but there are many cases where this assumption does not hold.

In this paper, we are motivated by applications in bioinformatics, where machine learning algorithms are used to predict the functions of some biological objects such as genes or proteins. For example, we will carry out experiments later on the prediction of the genes that are the target of some transcription factors [15, 5]. Other applications include the identification of microRNA genes[21], the inference of protein-protein interactions [3] or the discovery of disease-specific genes [4]. In this latter application for example, a small set of genes involved in one disease is known and we would like to predict which other genes among all human genes are also involved in that disease. Typically, the number of genes associated to a disease is very small compared to the complete human genome, there are no clearly identified negative genes (because it is difficult to design experiments that would prove that a gene is not related to a disease), and all human genes of interest for which we want to obtain predictions are given in advance.

The class of problems we are interested in can be abstracted as follows: we assume that we have a finite universe of N objects each one described by a vector of features and out of which N_+ are already known to share some function, and the goal is to rank the remaining $N - N_+$ unlabeled objects by decreasing order of their probability to share that function, by inferring a score function computed from the vector of features. The most prominent characteristics of this problem with respect to classical supervised learning problems are as follows:

- There are only positive and unlabeled objects and no clearly identified negative ones.
- N_+ is typically very small with respect to N.
- Although N can be very large, all possible unlabeled objects are supposed to be known and we are interested only in making predictions for this finite set of objects (transductive).

Several solutions have been proposed for these problems. The simplest one is probably to directly apply

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

standard supervised classification methods assuming that all unlabeled examples are negative and using cross-validation to obtain non-trivial predictions for the unlabeled examples (e.g., [15, 9]). It is shown in [9] that, under the assumption that the labeled examples are selected randomly from the positive examples, this approach predicts class conditional probabilities that differ by only a constant factor from the conditional probabilities predicted by a model trained from the true labeling. This assumption is however unlikely to be met in our setting where we can not even assume that examples have been independently drawn from some probability distribution. Instead of considering all unlabeled examples as negative, one could also select a subset of reliable negative examples from the unlabeled ones, for example using some prior knowledge about the problem [3, 21, 5] or other algorithmic solutions [22]. Another approach is to forget the unlabeled examples and learn a model only from the positive ones; methods such as one-class SVM [17] have been used by several researchers for that purpose [20, 4, 21]. Some other specific algorithms have also been proposed [6, 14]. In particular [6] proposes a variant of the C4.5 algorithm for learning with positive and unlabeled examples only.

In this paper, we propose an extension of gene set enrichment analysis methods that are commonly used in bioinformatics [2, 7]. Given a small set of genes, determined by some experimental analysis or some prior functional categorization, the goal of these methods is to determine what these genes have in common that a set of random genes drawn from the genome of interest does not, by exploiting statistical hypothesis tests to determine for which features the subset of genes is 'enriched'. In this paper, we propose to formalize and generalize this research process as the problem of defining a hypothesis space of scoring functions of features and a statistical testing procedure, and from them to learn a function of the features whose scoring appears as maximally significant according to the statistical test. In doing this, we analyze the nature of the multiple testing problem in this context and provide a procedure for avoiding its possible optimistic bias. We then consider the particular case of linear scoring functions, and derive an efficient algorithm based on an adaptation of one-class SVM models.

The rest of paper is structured as follows. In Section 2, we formalize the framework and provide an algorithm for linear models. In Section 3, we experimentally compare this algorithm with state-of-the-art methods on the prediction of transcription factor targets. Section 4 concludes the paper.

2 ENFORCING STATISTICAL SIGNIFICANCE

After a brief overview of statistical hypothesis testing whose aim is also to introduce the terminology used in the paper, we formulate our approach and then address the case of learning linear score functions.

2.1 Statistical Hypothesis Testing

Statistical hypothesis testing aims at making decisions from experimental data. The definition of a statistical test starts with the statement of a null hypothesis H_0 that one wants to challenge using the data. Then, one defines an appropriate statistic T and computes its value T_{obs} from the data. Such a statistic can be any function computed from the data, but in practice one will target statistics that are expected to be small if H_0 is true and large otherwise (or vice-versa). One then seeks to compute the probability to observe a value of the statistic as extreme as T_{obs} assuming that the null hypothesis is true. This probability is called the *p-value* of the test; when the p-value is smaller than a pre-defined significance threshold α, one rejects the null hypothesis. Computationally, the key point is the determination of the p-value given the observed data only. One can distinguish parametric and non-parametric tests. Parametric tests make some additional assumptions such as normality which take advantage of prior knowledge and/or make it possible to compute analytically the p-value. Non-parametric tests on the other hand do not require additional assumptions but they then rely on ranking statistics or resampling methods to compute the p-value.

In many applications, several hypotheses are in fact tested in parallel for example to determine which among several random variables (or features) are significantly correlated with a specific target variable. In this case, considering as significant all tests with a p-value smaller than the significance level α can be dangerous. Indeed, among N statistical tests, one can expect to obtain on the average $\alpha.N$ of them with a p-value lower than α, under the assumption that all null hypotheses are true. This is the so-called multiple testing problem [16], which can be addressed in several ways. One common technique is to control the family-wise error rate (FWER) instead of the p-value of each individual hypothesis. For a given p-value threshold, the family-wise error rate is the probability of having at least one false positive among the tests with p-values lower than this threshold. Practically, one ranks the multiple tests by increasing p-value, then computes the family-wise error rate for these nested subsets and stops to add additional tests as soon as the family-wise error rate exceeds the pre-specified significance level

α. The family-wise error rate associated to each test in the ranking can thus be considered as an *adjusted* or *corrected* p-value taking into account the multiplicity of the tests. Like for the computation of the p-value, there exist parametric and non-parametric methods, among which permutation based, to compute adjusted p-values for controlling the FWER [13].

2.2 Learning a Maximally Significant Test Statistic

In our context, let us denote by S_+, S_U, and S respectively the set of positive objects, the set of unlabeled objects and $S = S_+ \cup S_U$ the universe of objects and by N_+, N_U, and N their respective cardinalities, and let us denote by $x(o) \in \mathcal{X}$ the vector of features describing all objects of S. We state the general objective of learning a test statistic for our problem as follows:

Find a function $f : \mathcal{X} \to \mathbb{R}$ such that for some given statistic $g : \mathbb{R}^{N_+} \to \mathbb{R}$, the probability of observing a combined value $g(\{f(x(o))|o \in S'\})$ for a subset S' of size N_+ drawn at random from S greater than $g(\{f(x(o))|o \in S_+\})$ is as small as possible:

$$f^* = \arg\min_{f \in \mathcal{F}} p(f, g, S_+, S) \tag{1}$$

$$p(f, g, S_+, S) \stackrel{\triangle}{=} P_{S' \in R(S,N_+)}(g(f(S')) \geq g(f(S_+))), \tag{2}$$

where we denote by $f(S)$ the ensemble of values taken by f for the objects o in S, and by $R(S, N_+)$ the collection of all subsets S' of size N_+ that can be drawn in S, $R(S, N_+) = \{S' \subset S : |S'| = N_+\}$ and by $P_{S' \in R(S,N_+)}(.)$ the probability of the occurrence of the event in argument when subsets S' are drawn uniformly in $R(S, N_+)$:

$$P_{S' \in R(S,N_+)}(E(S')) = \frac{1}{|R(S, N_+)|} \sum_{S' \in R(S,N_+)} 1(E(S')),$$

where $E(S')$ denotes any event depending on S' and $1(.)$ is the indicator function equal to 1 when its argument is true, 0 otherwise.

The function f to be learned computes some score on the objects from their feature vector, while g is a given (fixed) function that aggregates these scores to provide a statistic on a subset of objects. Since we want to maximize the significance of the test, $g(f(S))$ should grow with the values in the set $f(S)$. Following the statistical terminology introduced in the previous section, the probability in (2) for a given (f, g) pair will be called its *apparent* p-value, with lower p-values meaning more significantly higher values of f in S_+ and thus a better score function f. The goal is thus to find a score function whose values on S_+ are higher than under the null hypothesis that S_+ would have been drawn at random from S. Good candidate positive examples among the unlabeled ones are then predicted as those that correspond to the highest values of f.

The solution as formulated in (1) and (2) corresponds to a non-parametric test as it does not make any hypothesis about the distribution of the data S and, at least theoretically, the p-value in (2) can be computed from the data simply by enumerating all possible subsets S' in S. It can also be considered as a permutation test of independence between f and a binary random variable taking a value of 1 for the examples in S_+ and 0 elsewhere. Indeed, selecting a random subset of size N_+ in S is equivalent to permuting the value of this binary variable in S.

Addressing the multiple testing problem. The problem as formulated in (2) is not regularized: the larger the space $\mathcal{F}$ of functions that is explored, the more chance we have to find a function f^* with a p-value smaller than any given $\alpha > 0$ even if our null hypothesis, stating that S_+ is a random subset of objects from S, is true. Adopting a strategy similar to the solutions proposed for the multiple statistical testing problem, one should thus also control the FWER when selecting a function in a space $\mathcal{F}$ of candidate functions. If we are only interested in an unbiased estimate of the p-value of f^* determined by (1), we can compute an adjusted p-value as follows:

$$p_{adj}(f^*, \mathcal{F}, S_+, S) = \tag{3}$$
$$P_{S'' \in R(S,N_+)}(\min_{f \in \mathcal{F}} p(f, g, S'', S) \leq p(f^*, g, S_+, S)).$$

This adjusted p-value (3) measures the proportion of random subsets that can be separated by a function in $\mathcal{F}$ from other random subsets with an apparent p-value as low as the one determined for S_+. It thus estimates the probability of making a mistake when declaring the optimal f^* in $\mathcal{F}$ as significant. The lower it is, the more significant is the score function f^* found.

This adjusted p-value depends both on the candidate function space $\mathcal{F}$ and on the problem. When the observed (or apparent) p-value $p(f^*, g, S_+, S)$ is fixed, the larger $\mathcal{F}$, the easier it will be to minimize $p(f, g, S'', S)$ for a random subset S'' and thus the higher the adjusted p-value. On the other hand, larger $\mathcal{F}$ means also a lower uncorrected p-value $p(f^*, g, S_+, S)$ and thus makes it harder finding a random subset that can be separated with an as low p-value. There should thus be a tradeoff in terms of the complexity of $\mathcal{F}$ which could be determined by investigating the variation of $p_{adj}(f^*, \mathcal{F}_i, S_+, S)$ for a nested collection of hypothesis spaces $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \ldots$

Let us also notice that, for a fixed $p(f^*, g, S_+, S)$, this adjusted p-value can be thought of as a measure of the

complexity of the function space $\mathcal{F}$ related to the data S, and it is possible to draw some similarity between this measure and the empirical Rademacher complexity [1]. Indeed, both measures evaluate the capability of a function space to separate randomly labeled versions of a data sample S. A difference however is that the p-value (3) also takes into account the quality of the fit of the observed data through $p(f^*, g, S_+, S)$, which is not the case of the empirical Rademacher complexity.

Monte-Carlo estimation of p-values. In principle, the p-value (3) can be computed directly from the data, either exactly by drawing all subsets $S'' \in R(S, N_+)$ or approximately by drawing only a number of them by Monte-Carlo (3):

$$\hat{p}_{adj}(f^*, \mathcal{F}, S_+, S) = \frac{1}{T}\sum_{i=1}^{T} 1(\min_{f\in F} p(f, g, S_i'', S) \leq p(f^*, g, S_+, S)), \quad (4)$$

where the S_i'' denote random subsets of size N_+ drawn from S.

In turn, the values of (2) used twice in this formula may also be estimated by Monte-Carlo, by using a similar approximation, namely

$$\hat{p}(f, g, S', S) = \frac{1}{T}\sum_{i=1}^{T} 1(g(f(S_i')) \geq g(f(S'))), \quad (5)$$

where the S_i' denote random subsets of size N_+ drawn from S.

But since for each subset S_i, one has to compute $\min_f \hat{p}(f, g, S_i, S)$ over $\mathcal{F}$ that also requires an additional sampling of random subsets S_i' in S and as many minimizations over $\mathcal{F}$ as there are subsets S_i'', in general, this procedure may not be practically feasible. Nevertheless, if $\mathcal{F}$ is finite (and not too large), one can directly exploit the efficient resampling techniques proposed in [13] to avoid the double permutation procedure and render the computation of (3) tractable. But, since in what follows, we will consider infinite function spaces, these exact or approximate Monte-Carlo computations of (3) for direct minimization will not be practically feasible in general.

Choice of a statistic. The statistic g allows to introduce various hypotheses in the problem. In what follows, we will restrict ourselves to a function g that simply sums up the values of f in the given sample. Equations (1) and (2) thus translate into the following problem:

$$f^* = \arg\min_{f\in\mathcal{F}} P_{S'\in R(S,N_+)}(\sum_{o'\in S'} f(x(o')) \geq \sum_{o\in S_+} f(x(o))). \quad (6)$$

We will see that this will lead to a simple algorithm in the case of linear score functions.[1]

Special cases. Before we address the case of linear functions f, let us now discuss how our general framework relates to some standard non parametric tests for two specific forms of the output space of the functions in $\mathcal{F}$.

Let us first assume that functions in $\mathcal{F}$ can take only their value in $\{0, 1\}$. In this case, $\sum_{o\in S'} f(x(o))$ is the number of objects in S' for which $f(x(o))$ is equal to 1 and the p-value becomes the probability that there are more objects o for which $f(x(o)) = 1$ in a random S' than in S_+. It thus corresponds exactly to the p-value of a (one-tailed) Fisher exact test for measuring the association between the two (binary) classifications of the objects, according to their membership to S_+ and to their values for f. This p-value can thus be computed analytically without enumerating all possible random subsets S' by computing the sum of several hypergeometric probabilities. When N is large, it can also be approximated by a chi-square statistic.

Another sensible choice is to use functions f that output a ranking of the objects, ie. functions that associate to each object o its rank among the N objects in S as a function of its input features $x(o)$. In this case, for a given ranking function f, the p-value in (2) is the probability that the sum of ranks in a random sample would be greater or equal to the observed one in S_+. It corresponds exactly to the p-value of a Wilcoxon rank sum test, that can be computed exactly for small sample sizes or be approximated by a normal distribution for larger sample sizes. Since the area under the ROC curve (AUC) is proportional to the rank sum statistic, this p-value measures also the probability that the AUC of the ranking f for distinguishing objects in S_+ from unlabeled objects is greater than expected by chance. When one considers functions f that are rankings only, minimizing probability (2) thus amounts at maximizing the AUC.

2.3 Linear Score Functions

In this section, we particularize the proposed framework to the case of linear score functions over the original feature space. The extension to non-linear func-

[1] Other alternatives could be imagined inspired by classical statistical tests. For example, instead of a simple sum, one could instead derive a t-test like statistic that also takes into account the variation of the values of f in the subset, e.g., $g(f(S)) = \mu(f(S))/\sigma(f(S))$, where $\mu(f(S))$ and $\sigma(f(S))$ denote respectively the mean and standard deviation of the values in $f(S)$. This would ensure that the values of f in addition to be larger on the average in S_+ than in a random subset are also close to each other.

tions with kernels is straightforward and is addressed in the Appendix.

When all features are numerical ($x(o) \in \mathcal{X} = I\!R^n$), one can consider score functions f that are linear combinations of the attribute values, ie.[2]:

$$\mathcal{F} = \{f : I\!R^n \rightarrow I\!R | f(x) = w^T x, w \in I\!R^n\}. \quad (7)$$

In this case, the optimization problem (6) may be written:

$$w^* = \arg\min_w P_{S' \in R(S,N_+)}(\sum_{o \in S'} w^T x(o) \geq \sum_{o \in S_+} w^T x(o)), \quad (8)$$

meaning to find a direction w^* such that the average projection of the objects in S_+ on this direction is as often as possible greater than the average projection of the objects in a random subset S'. The average of a linear projection being the projection of the average, this objective may be further rewritten:

$$w^* = \arg\min_w P_{S' \in R(S,N_+)}(w^T x'(S') > 0), \quad (9)$$

where we define:

$$x'(S') = \frac{1}{N_+} \sum_{o \in S'} x(o) - \frac{1}{N_+} \sum_{o \in S_+} x(o), \quad (10)$$

as the center of mass of the feature vectors in S' relative to the center of mass of feature vectors in S_+.

Estimating the probability in (9) by Monte-Carlo from T random subsets $\{S'_1, S'_2, \ldots, S'_T\}$, one thus gets the following minimization problem:

$$w^* = \arg\min_w \frac{1}{T} \sum_{S'_i} 1(w^T x'(S'_i) \geq 0). \quad (11)$$

Formulations of the problem in (9) and (11) can be interpreted as finding a hyperplane that separates as many vectors $x'(S')$ as possible from the origin, the latter coinciding with the target subset S_+ given the centering in (10). This problem is similar to the problem solved by one-class SVM [17] that also amounts at separating some input feature vectors from the origin.

When it is possible to find a vector w that leads to a null (estimated) p-value, i.e. the data is linearly separable, then there are actually an infinite number of such directions. In this case, we propose to adopt the same strategy as in one-class SVM, i.e. look for the direction w that maximizes the margin. This direction can be obtained by solving the following optimization problem, for any $\rho > 0$ [17]:

$$\min_w \frac{1}{2}||w||^2 \text{ subject to } w^T x'(S'_i) \leq -\rho, i = 1, \ldots, T \quad (12)$$

[2]Note that since f is only used to make pairwise comparisons in (2), there is no need to introduce a bias term.

Translated to our context, the resulting w is such that when using $f(x) = \frac{{w^*}^T x}{||w^*||^2}$ as a score function, it maximizes the gap between the score of S_+ and the score of its closest random subsets:

$$\min_{S' \in \{S'_1, \ldots, S'_T\}} \sum_{o \in S'} f(x(o)) - \sum_{o \in S_+} f(x(o))) = \frac{\rho}{||w||^2}. \quad (13)$$

Although maximizing this gap does not influence the apparent p-value (which remains equal to zero), we show below that this can be interpreted as choosing a linear combination of minimal adjusted p-value, among all those of zero apparent p-value.

When the vectors $x'(S'_i)$ are not linearly separable from the origin, it is not possible to drive the apparent p-value in (11) to zero and problem (12) has no solution. Again, we will adopt in the case the same strategy as in one-class SVM and allow that some subsets S'_i cross the hyperplane by introducing slack variables in the optimization problem. Adapting soft-margin one-class SVM to our problem, we thus propose to solve the following optimization problem to obtain w^*:

$$\min_{w \in I\!R^n, \xi \in I\!R^T, \rho \in I\!R} \frac{1}{2}||w||^2 + \frac{1}{\nu T} \sum_i (\xi_i - \rho) \quad (14)$$

$$\text{subject to } w^T x'(S'_i) \leq -\rho + \xi_i, \xi_i \geq 0, \forall i = 1, \ldots, T. \quad (15)$$

Note that, although the optimization includes a bias term ρ, this term is not exploited in the final score function. The parameter $\nu \in]0, 1]$ is introduced to balance the minimization of the p-value and the maximization of the margin. As ν increases, the margin will increase but the p-value will also increase. Transposing results in [17], assuming that the solution to (14) is such that $\rho > 0$, ν is an upper bound on the fraction of vectors $x'(S'_i)$ that fall on the same side of the hyperplane $w^T x + \rho = 0$ as the origin. ν is thus also an upper bound on the fraction of vectors $x'(S'_i)$ such that $w^T x'(S'_i) \geq 0$ and thus an upper bound on the estimated p-value (11).

On the link between margin and adjusted p-value. Let us assume that the random subsets S'_i are fixed and that it is possible to linearly separate all the T vectors $x'(S'_i)$ from the origin. Let us further consider that we use the following algorithm to learn a linear score function for a given subset S_+:

- Compute the solution w^* of (12)
- If the resulting (maximum) margin $m^* = \frac{\rho}{||w^*||^2}$ is greater than m, then output the linear score function $f(x) = {w^*}^T x$
- Otherwise return the function $f(x) = 0, \forall x$.

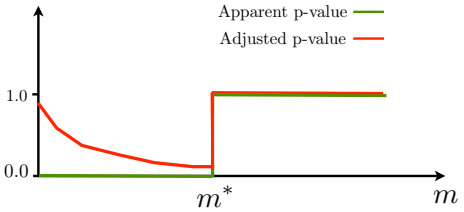

Figure 1: Evolution of the apparent and adjusted p-values with the margin threshold

This algorithm returns a trivial solution as soon as the margin corresponding to the positive set S_+ is not large enough. m is thus a parameter of the algorithm that controls its complexity. The evolution of the apparent and adjusted p-values of this algorithm when m is increased is depicted in Figure 1. When m is equal to 0, the algorithm returns a solution as soon as the subset is separable. Since we assume that this is the case of S_+, its apparent p-value is thus zero for $m = 0$. It also remains so while m remains lower than the maximum margin m^*. When m is greater than m^*, it is not possible anymore to find a score function with a margin greater than m and thus the apparent p-value becomes 1. For a given $m < m^*$, the adjusted p-value is the proportion of random subsets that can be separated from the other random subsets with a margin of at least m (for which the apparent p-value computed by the algorithm is 0) and thus it is monotonically decreasing with m. When $m > m^*$, the p-value of S_+ being 1, it is always greater or equal to the p-value that can be obtained on any random subsets and thus the adjusted p-value equals 1. This shows that the value of the parameter m that minimizes the adjusted p-value is m^* and it corresponds to the linear score function which is the solution of (12). Maximizing the margin can thus be interpreted as minimizing the adjusted p-value.

In the non-separable case, the analysis must also take into account the effect of the parameter ν, and so the relation between margin maximization and adjusted p-value is less straightforward. Intuitively, however, we think that also here the regularization introduced by the formulation (14) should have a beneficial effect in terms of minimizing the adjusted p-value.

3 EXPERIMENTS

In this section, we carry out experiments with the proposed method on the prediction of regulatory networks in *Escherichia coli* and Yeast *S. cerevisiae* from microarray expression data.

Problem and Datasets. The elucidation of regulatory networks is an important problem of systems biology. Transcription factors (TF) are proteins that control the transcription of genes into messenger RNA and play a key role in transcriptional regulation. For a given organism, one has typically a partial knowledge only of which genes are regulated by a given TF. Several researchers have proposed the use of learning techniques to complete this partial knowledge by integrating various experimental data about genes [15, 10, 5]. From a subset of genes that are known to be regulated by a given transcription factor (the positive set), the goal is to predict what are the genes in the unlabeled set that are the more likely to be also regulated by this TF. This problem is a very good representative of the setting that we introduced earlier. Typically, for a given transcription factor, there are only a few positive examples of genes that it regulates and negative examples are usually not reported in the databases. All genes (and their describing features) are furthermore known in advance and they can thus not be assumed to be drawn from some probability distribution.

We gathered two datasets corresponding to two different organisms, the yeast Saccharomyces cerevisae (Yeast) and the bacteria E.Coli (Ecoli). The datasets contain respectively 6178 genes and 4345 genes. As input features from which to infer the regulation, we selected microarray expression data. For Yeast, we used the expression data produced in [19] and [8], totalizing 157 numerical features for 6178 genes. For Ecoli, we use the expression data collected in [11][3], yielding 445 numerical features for 4345 genes. The regulatory network for Yeast was obtained from [18]. It contains the regulated genes for 80 different transcription factors, totalizing 1164 interactions involving 606 genes. The number of known regulated genes per transcription factor ranges from 4 to 57 (median 10). The regulatory network for Ecoli was obtained from RegulonDB [12][4]. It contains the regulated genes for 154 transcription factors and totalizes 3293 interactions involving 1164 genes. For Ecoli, [15] noticed that these 4345 genes comprises several operons, ie. groups of genes that are regulated together and whose expressions are also very close. These operons introduce an undesirable positive bias in the predictors if no particular care is taken. While [15] adapted the cross-validation procedure to ensure that no operon was split between the training and test set, we adopted a simple strategy that consists in only keeping in our dataset one representative gene randomly selected in each operon. The

[3]We downloaded the v3 build 1 release from `http://gardnerlab.bu.edu/data/PLoS_2007/data_and_validation.html` and use the GCRMA normalized data in our experiments.

[4]We downloaded the preprocessed files exploited in [11] from `http://gardnerlab.bu.edu/data/PLoS_2007/data_and_validation.html`.

list of operons was downloaded from regulonDB [12] and the sampling resulted in a reduction of the number of genes in the dataset from 4345 to 2925. Given that we will use a two-stage cross-validation in our experiments, we have furthermore removed from the Ecoli dataset all transcription factors which have less than 4 known target genes. The final dataset contains 2925 genes, 63 transcription factors, 1446 interactions involving 554 genes with a number of regulated genes per TF ranging from 4 to 357 genes (median 6).

Note that our goal in this paper is not to get the best results on the problem of predicting regulatory networks but merely to assess the proposed algorithms. The former would imply the consideration of additional input features and up-to-date training networks.

Compared Methods and Protocol. In our experiment, we compare five differents methods. Except the last one, all these methods rank the genes using a linear score function:

PU: our method with a linear score function as described in Section 2. This method has two parameters: the number of random subsets T and the regularization parameter of one-class SVM, ν. The first one has been fixed to 1000 in all our experiments. The optimal value of the second parameter will be determined by cross-validation (see below).

2SVM: a two-class support vector machines applied using the unlabeled examples as negative examples. To get an unbiased prediction for all unlabeled examples, we adopted the same strategy as in [15]: the set of unlabeled examples is splitted into k subsets of approximatively the same sizes, a model is learned using in turn $k-1$ of these subsets as the negative examples and all positive examples. This model is then used to compute the score for the hold-out subset. A linear kernel was used and both the regularization parameter of SVM, C, and the number of subsets k were determined by cross-validation. Note that, since we are only interested in the ranking of the unlabeled examples and not the absolute score values, this method is also equivalent to the method in [9, 5].

1SVM: a one-class SVM trained using only the positive examples. Following [4], we centered the data so that the origin coincides with the center of mass of the unlabeled examples. This modification turns out to be crucial to obtain good accuracy in our experiments. We use a linear kernel and the regularization parameter ν was determined by cross-validation.

CML: a simple method that ranks the genes according to the following linear score: $f(x) = (\bar{x}(S_+) - \bar{x}(S_U))^T x$ where $\bar{x}(S_+)$ (resp. $\bar{x}(S_U)$) computes the center of mass in S_+ (resp. S_U). This function projects each example on the line connecting the positive and unlabeled centers of mass. This method can be seen as an extreme case of both 1SVM that distinguish the positive from the center of mass of the negatives and PU that distinguish the negatives from the center of masses of the positives.

CORR: this method simply ranks the genes according to their average correlation with the genes in the positive set. This method is based on the assumption that the expressions of genes that are co-regulated by the same TFs should be correlated.

To compare these methods, we adopted a protocol similar to the one proposed in [4]: the set of positive examples was divided into 5 subsets of approximatively the same size[5]. Each subset was removed in turn from the positive set and introduced among the unlabeled examples. Then a model was learned with each method to rank the unlabeled examples, including the holdout positive subset. The rank of each positive gene among the unlabeled examples (not counting the other positive genes) was computed and normalized by the total number of unlabeled examples. A ROC curve was then obtained from all these ranking by computing as a function of x the proportion of positive genes that are ranked among the top x% of the unlabeled genes. The area under this curve (AUC) was then computed as well as the average normalized rank of all positive genes. When a given method depends on the value of some parameters, their optimal value was determined by an additional internal cross-validation loop following the same scheme as the external one (again with 5 fold cross-validation or leave-one-out)[6].

Results. Table 1 reports average results over the 80 and 61 task respectively. We observe that on the average, the PU method is better than all other tested methods on the Yeast dataset and it ranks second, after 2SVM, on the Ecoli dataset. On Yeast, 1SVM is almost as good as PU and better than 2SVM, while on Ecoli, 1SVM is much worse than these two methods.

These results are confirmed by the pairwise comparisons displayed in Table 2. In Yeast, PU is better than 1SVM and 2SVM for 70% of the TFs and it improves these methods respectively by 3% and 9%. In Ecoli, PU is only better than 2SVM in 30% of the cases (nevertheless with an average improvement of 9% of the AUC in these cases) and better than 1SVM in 80% of the cases (with an improvement of more than 10%). The two simple methods, CML and CORR, are competitive in Yeast with CML even better than

[5] When S_+ contained less than 5 objects, a leave-one-out procedure was used.

[6] ν for PU and 1SVM was optimized in $\{0.001, 0.01, 0.1, 0.25, 0.5, 0.75\}$, C for 2SVM was searched in $\{0.0001, 0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$, and the number of folds k in $\{2, 3, 4\}$. These values were narrowed down by preliminary experiments on a subset of the data.

Table 1: Average results (± standard deviations) on the two regulatory networks

Method	PU	2SVM	1SVM	CML	CORR
Yeast, 80 TFs					
AUC	0.880±0.095	0.843±0.120	0.874±0.093	0.865±0.111	0.843±0.117
Avg rank	0.145±0.106	0.186±0.130	0.152±0.102	0.161±0.121	0.184±0.126
Ecoli, 61 TFs					
AUC	0.850±0.119	0.863±0.118	0.791±0.114	0.747±128	0.828±0.136
Avg rank	0.178±0.134	0.166±0.134	0.243±0.127	0.290±0.139	0.200±0.146

Table 2: Pairwise comparison of all methods. In each cell, we report the number of TFs where the row method is better than the column method (in terms of AUC) as well as the average relative improvement (in %) of AUC of the row method versus column method for those TFs for which the former outperforms the latter.

Yeast, 80 TFs

Method	PU	2SVM	1SVM	CML	CORR
PU	-	55/9.3	56/2.7	62/3.4	67/6.5
2SVM	25/3.6	-	28/4.7	30/7.9	38/8.3
1SVM	24/3.9	52/9.5	-	42/4.7	65/5.8
CML	18/1.8	50/9.4	38/1.9	-	61/4.2
CORR	13/1.5	42/7.3	15/1.5	19/1.4	-

Ecoli, 61 TFs

Method	PU	2SVM	1SVM	CML	CORR
PU	-	19/8.5	49/11.1	55/17.5	38/7.2
2SVM	42/6.2	-	49/14.2	53/21.7	43/10.0
1SVM	12/5.8	12/9.1	-	45/11.0	19/10.8
CML	6/4.9	8/8.1	16/3.7	-	12/9.7
CORR	23/3.5	18/7.0	42/11.5	49/17.4	-

2SVM in 60% of the TFs. In Ecoli, the CML method is much worse than all methods but the correlation based method is still better than 1SVM in 70% of the TFs and it improves it by more than 10% in average.

The fact that 1SVM and the two simple methods are competitive in Yeast and that they are not in Ecoli suggests that the unlabeled examples are more informative in Ecoli than in Yeast. This might come from the fact that input features in Ecoli are compiled from very diverse experiments while they are more narrow in scope in the Yeast dataset where both feature subsets measure expression values related to the cell cycle.

4 DISCUSSION

We presented in this paper a method for learning with positive and unlabeled examples. The main idea behind this method is to explicitely derive a test statistic that minimizes the p-value of a non-parametric test trying to characterize the positive examples among the set of all examples, positive and unlabeled. With linear score functions, the resulting method is equivalent to a one-class SVM applied on aggregated features in random subsets of objects. Experiments on the prediction of Yeast and Ecoli regulatory interactions (totalizing 141 distinct classification problems) show that the method is most often better than one-class SVM and competitive with two-class SVM.

There is a fundamental difference between our formulation and the standard use of one-class SVM for PU learning problems. Indeed, the idea behind one-class SVM is that the unlabeled examples are outliers with respect to the positive set. In constrast, our method is based on the idea that positive examples are the outliers inside the set of all examples. We believe that the latter hypothesis is more natural in the targeted applications. It also naturally leads to the exploitation of the negative examples, which are neglected by the standard one-class SVM solution. With respect to two-class SVM, we do not make the assumption that all unlabeled examples are negative and do not have to use cross-validation to get an unbiased prediction for the unlabeled examples. Our method is also computationally advantageous with respect to two-class SVM as it requires only one training on a subset of examples, whose size is furthermore controlled by the user through the number of random subsamples T.

In terms of future works, we would like to exploit this approach with more complex scoring functions, beginning with non-linear kernels. There also exist extensions of one-class SVM model for multiple kernel learning [4], whose applications in our framework would be straightforward. More generally, we believe that the idea of formulating learning problems as searching for good statistical tests is a general framework that deserves to be further studied and could lead to the design of novel algorithms. In particular, the link between multiple hypothesis testing, adjusted p-values and regularization is an interesting future direction for theoretical research.

Acknowledgements.

The author would like to thank Louis Wehenkel for useful comments about the manuscript. PG is a research fellow of the FNRS, Belgium. This work is partially supported by the Interuniversity Attraction Poles Programme (IAP P6/25 BIOMAGNET), initiated by the Belgian State, Science Policy Office and by the European Network of Excellence, PASCAL2.

References

[1] Bartlett, P., Mendelson, S.: Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research 3(3), 463–482 (2003)

[2] Beissbarth, T., Speed, T.P.: Gostat: find statistically overrepresented gene ontologies within a group of genes. Bioinformatics 20(9), 1464–5 (Jun 2004)

[3] Ben-Hur, A., Noble, W.: Choosing negative examples for the prediction of protein-protein interactions. BMC bioinformatics 7(Suppl 1), S2 (2006)

[4] Bie, T.D., Tranchevent, L.C., van Oeffelen, L.M.M., Moreau, Y.: Kernel-based data fusion for gene prioritization. Bioinformatics 23(13), i125–32 (Jul 2007)

[5] Cerulo, L., Elkan, C., Ceccarelli, M.: Learning gene regulatory networks from only positive and unlabeled data. BMC bioinformatics (Jan 2010)

[6] Denis, F., Gilleron, R., Letouzey, F.: Learning from positive and unlabeled examples. Theoretical Computer Science 348(1), 70–83 (2005)

[7] Dennis, G., Sherman, B.T., Hosack, D.A., Yang, J., Gao, W., Lane, H.C., Lempicki, R.A.: David: Database for annotation, visualization, and integrated discovery. Genome Biol 4(5), P3 (Jan 2003)

[8] Eisen, M., Spellman, P., Patrick, O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. 95, 14863–14868 (1998)

[9] Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining pp. 213–220 (2008)

[10] Ernst, J., Beg, Q.K., Kay, K.A., Balázsi, G., Oltvai, Z.N., Bar-Joseph, Z., Stormo, G.: A semi-supervised method for predicting transcription factor–gene interactions in escherichia coli. PLoS Computational Biology 4(3), e1000044 (Mar 2008)

[11] Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., Gardner, T.S.: Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol 5(1), e8 (Jan 2007)

[12] Gama-Castro *et al.*, S.: Regulondb (version 6.0): gene regulation model of escherichia coli k-12 beyond transcription, active (experimental) annotated promoters and textpresso navigation. Nucleic Acids Res 36(Database issue), D120–4 (Jan 2008)

[13] Ge, Y., Dudoit, S., Speed, T.: Resampling-based multiple testing for microarray data analysis. Test 12(1), 1–77 (2003)

[14] Lee, W., Liu, B.: Learning with positive and unlabeled examples using weighted logistic regression. Proceedings of the International Conference on Machine Learning 20(1), 448 (2003)

[15] Mordelet, F., Vert, J.P.: Sirene: supervised inference of regulatory networks. Bioinformatics 24(16), i76–i82 (May 2008)

[16] Noble, W.S.: How does multiple testing correction work? Nature Biotechnology 27(12), 1135–1137 (Jan 2009)

[17] Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., Williamson, R.: Estimating the support of a high-dimensional distribution. Neural computation 13(7), 1443–1471 (2001)

[18] Simonis, N., Wodak, S.J., Cohen, G.N., van Helden, J.: Combining pattern discovery and discriminant analysis to predict gene co-regulation. Bioinformatics 20(15), 2370–9 (Oct 2004)

[19] Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell 9(12), 3273–3297 (1998)

[20] Wang, C., Ding, C., Meraz, R., Holbrook, S.: Psol: a positive sample only learning algorithm for finding non-coding RNA genes. Bioinformatics (Jan 2006)

[21] Yousef, M., Jung, S., Showe, L.C., Showe, M.K.: Learning from positive examples when the negative class is undetermined- microRNA gene identification. Algorithms Mol Biol 3(1), 2 (Jan 2008)

[22] Yu, H., Han, J., Chang, K.: Pebl: Positive example based learning for web page classification using SVM. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining p. 248 (2002)

APPENDIX—SUPPLEMENTARY MATERIAL

Kernelization.

In this appendix, we explicitly derive the kernelization of the approach proposed in Section 2.3 to learn a function f which is a non-linear function of the inputs.

Assuming that each object is described by a vector $\phi_x(o)$ lying in the feature space Φ corresponding to some kernel k, then the transformed vectors $x'(S')$ defined in Eqn. (10) become vectors $\phi'_x(S')$ from Φ computed by:

$$\phi'_x(S') = \frac{1}{N_+}\sum_{o\in S'}\phi_x(o) - \frac{1}{N_+}\sum_{o\in S_+}\phi_x(o),$$

and the dot-products $k'(S'_i, S'_j) = \phi'_x(S'_i)^T\phi'_x(S'_j)$ between two such vectors is given by:

$$\frac{1}{N_+^2}\Big(\sum_{o_1\in S'_1, o_2\in S'_2} k(o_1,o_2) - \sum_{o_1\in S'_1, o_+\in S'_2} k(o_1,o_+)$$
$$- \sum_{o_2\in S'_2, o_+\in S'_2} k(o_2,o_+) + \sum_{o_{+,1}\in S_+, o_{+,2}\in S_+} k(o_{+,1},o_{+,2})\Big),$$

which can be computed from the sole knowledge of the kernel k. We can thus directly use the kernel formulation of one-class SVM to learn a vector w of the following form:

$$w = \sum_{i=1}^{T}\alpha_i\phi'_x(S'_i).$$

To make predictions, objects in S can then be ranked according to:

$$f(\phi_x(o)) = \sum_{i=1}^{T}\alpha_i\phi'_x(S'_i)^T\phi_x(o),$$

which may be written in terms of the kernel k as follows:

$$f(\phi_x(o)) = \sum_{i=1}^{T}\alpha_i\frac{1}{N_+}\sum_{o'\in S'_i}k(o',o) - \frac{1}{N_+}\sum_{o'\in S_+}k(o',o),$$

making use of the fact that $\sum_i \alpha_i = 1$.