Kernel methods for Multi-labelled classification and Categorical regression problems.

André Elisseeff, Jason Weston
BIOwulf Technologies
305 Broadway,
New-York, NY 10007
{andre,jason}@barhilltechnologies.com

Abstract

This report presents a SVM like learning system to handle multi-label problems. Such problems arise naturally in bio-informatics. Consider for instance the MIPS Yeast genome database in [12], it is formed by around 3,300 genes associated to their functional classes. One gene can have many classes, and different genes do not belong to the same number of functional categories. Such a problem can not be solved directly with classical approaches and it is generally decomposed into many two-class problems. The binary decomposition has been done partially by different researchers [12] on the Yeast dataset but it does not provide a satisfactory answer. We explore in this report a new direct approach. It is based on a large margin ranking system that shares a lot of common properties with Support Vector Machines. We tested it on a toy problem and on real datasets with positive results.

We also present a new method to do feature selection with multi-labelled datasets. Our method is based on the multiplicative update rule defined in [17].

1 Introduction

Many problems in Text Mining or in Bioinformatics are multi-labelled. That is, each point in a learning set is associated to a set of labels. Consider for instance the classification task of determining the subjects of a document, or of relating one protein to its many effects on a cell. In either case, the learning task would be to output a set of labels whose size is not known in advance: one document can for instance talk about food, meat, and finance, although another one would concern only food, and fat. Two-class, multi-class classification or ordinal regression problems can all be cast into multi-label ones[1]. This makes the latter quite attractive but at the same time it gives a warning: their generality hides their difficulty to solve them. The number of publications is not going to contradict this statement: we are aware of only few works about the subject [9, 10, 13] and they all concern text mining applications.

In [9] a natural method to learn a multi-label task by decomposing it into many two-class problems is presented. Assume that the maximum size of the label sets is bounded by Q, then each label set can be coded as a Q length vector y of $\{+1, -1\}$ components, where $y_k = +1$ *iff* the label k is in the label set. Learning can then be done by defining Q two-class systems that learn how to predict the y_k's. Such an approach is referred to as the *binary approach*. As has been pointed out in [10, 13], it does not take into account the correlation between labels which can result in poor expressive power (see section 4.1 for a more detailed discussion). A different method that has been studied in [13] consists in computing a ranking of the labels for each input. Ranking labels is not the same as providing a set of labels but, when the size of the set is known, both are equivalent. However, in their original work, Schapire and Singer [13] only consider ranking and do not provide a method to compute the size of the label sets. In [10] and for text mining application, a direct computation of the label sets is proposed. Given one input,

[1] They all correspond to label sets of size one.

here one document, it computes the posterior probabilities of all the label sets and takes the maximum. It is not really practical when the number of label Q is large[2] , but an approximation scheme exists, which provides a direct way of computing a set of labels. Unfortunately, the extension of this approach to other problems rather than Text Mining is not discussed in the original work. As a general comment, it is worthwhile noticing that multi-labelled problems have been mainly addressed in the context of Text Mining.

In Schapire and Singer's work about Boostexter, one of the only general purpose multi-label ranking system [13],they observe that overfitting occurs on learning sets of relatively small size ($< 1,000$). They conclude that controlling the complexity of the overall learning system is an important research goal. The aim of the current paper is to provide a way of controlling this complexity while having a small empirical error. For that purpose, we consider only architectures based on linear models and follow the same reasoning as for the definition of Support Vector Machines (SVM) [3, 16]. Defining a cost function (section 2) and margin for multi-label models, we focus our attention mainly on two approaches: the first one is the binary approach and is explained in section 3, the second one is based on a ranking method combined with a predictor of the size of the sets. It is explained in section 4. Section 5 presents the categorical regression classification problem and shows how it can be solved easily by interpreting it as a multi-label one. Section 6 present first experiments on biological data. In section 7, we explain the feature selection method for multi-label problems.

Before entering the core of the paper, let us review some basic notations: all sets of labels will be denoted by bold letters and all vectors will be typed in normal font. When some confusion may arise between a vector and a scalar, their type will be made explicit in the text. We will denote by $|\mathbf{Y}|$ the size of the set $\mathbf{Y}$, and dot product will be written as $\langle ., . \rangle$. Finally, $x \sim D$ will mean that x is drawn according to the distribution D.

2 Cost functions for multi-label problems

Let $\mathcal{X} = \mathbb{R}^d$ be a d-dimensional input space. We consider as an output space the space formed by all the sets of integer between 1 and Q identified here as the labels of the learning problem. Such an output space contains 2^Q elements and one output corresponds to one set of labels. The learning problem we are interested in is to find from a learning set $S = \{(x_1, \mathbf{Y}_1), \ldots, (x_m, \mathbf{Y}_m)\} \subset (\mathcal{X} \times \mathcal{Y})^m$, drawn identically and independently from an unknown distribution D, a function f such that the following generalization error is as low as possible:

$$R(f) = E_{(x,\mathbf{Y})\sim D}\left[c(f, x, \mathbf{Y})\right] \tag{1}$$

The function c is a real-valued loss and can take different forms depending on how $f(x)$ is computed. Here, we consider only linear models. Given Q vectors $w_1, .., w_Q$ in $\mathbb{R}^d$ and Q bias $b_1, .., b_Q$ in $\mathbb{R}$, we follow two schemes:

1. *The binary approach:*

$$f(x) = \text{sign}\left(\langle w_1, x\rangle + b_1, \ldots, \langle w_Q, x\rangle + b_Q\right) \tag{2}$$

   where the sign function applies component-wise. The value of $f(x)$ is a binary vector from which the set of labels can be retrieved easily by stating that label k is in the set iff $\text{sign}(\langle w_k, x\rangle + b_k) > 0$. For example this can be achieved by using SVM for each binary problem and applying the later rule [9]. This way of computing the output is studied in the next section.

2. *The ranking approach:* assume that $s(x)$, the size of the label set for the input x, is known, we define:

$$r_k(x) = \langle w_k, x\rangle + b_k$$

   and consider that a label k is in the label set of x iff $r_k(x)$ is among the first $s(x)$ elements $(r_1(x), .., r_Q(x))$. The algorithm Boostexter [14] is an example of such a system. The ranking approach is analyzed more precisely in section 4.

[2]Exploring all label sets require 2^Q computational steps.

We consider the same loss functions as in [13, 14] for any multi-label system built from real functions $(f_1, .., f_Q)$. It includes the so-called *Hamming Loss* defined as

$$HL(f, x, \mathbf{Y}) = \frac{1}{Q} |f(x) \Delta \mathbf{Y}|$$

where Δ stands for the symmetric difference of sets. Note that the more $f(x)$ is different from $\mathbf{Y}$, the higher is the error. Missing one label in $\mathbf{Y}$ is less important than missing two, which seems quite natural in many real situations. As an example, consider that the labels are possible diseases related to different genes. Many genes can share the same disease and lead to many diseases at the same time. Predicting only one disease although there are three is worse than predicting only two. Having a Hamming Loss of 0.1 means that the expected number of times a pair (x, y_k) has been misclassified is 0.1. When $|\mathbf{Y}| = 1$ a multi-label system is in fact a multi-class one and the Hamming Loss is $\frac{2}{Q}$ times the loss of the usual classification loss.

We also consider the *one-error*:

$$\text{1-err}(f, x, \mathbf{Y}) = \begin{cases} 0 & \text{if argmax}_k f_k(x) \in \mathbf{Y} \\ 1 & \text{otherwise} \end{cases}$$

which is exactly the same as the classification error for multi-class problems (it ignores the rankings apart from the highest ranked one and so does not address the quality of the other labels). We believe this loss is not appropriate for multi-label problems as we define them but it can be useful for other cases, in particular when what matters is not to output the set of labels but only to give one of the correct labels. Consider for instance horse races and assume that the target is to bet on one horse that will be among the top 5. In that case the rank among the top 5 is not important and the loss to minimize is the one-error. On the other hand, if the goal is to predict all the horses in the top 5, then Hamming Loss should be preferred.

Other losses concern only ranking systems (a system that specifies a ranking but no set size predictor $s(x)$). Let us denote by $\bar{\mathbf{Y}}$ the complementary set of $\mathbf{Y}$ in $\{1, .., Q\}$. We define the *Ranking Loss* [14] to be:

$$\begin{aligned} RL(f, x, \mathbf{Y}) &= \frac{1}{|\mathbf{Y}| \, |\bar{\mathbf{Y}}|} \left| (i, j) \in \mathbf{Y} \times \bar{\mathbf{Y}} \text{ s.t. } r_i(x) \leq r_j(x) \right| && (3) \\ &= \frac{1}{|\mathbf{Y}| \, |\bar{\mathbf{Y}}|} \sum_{k \in \mathbf{Y}} |\{l \in \{1, .., Q\} \text{ s.t. } r_l(x) \geq r_k(x)\}| - |\{l \in \mathbf{Y} \text{ s.t. } r_l(x) \geq r_k(x)\}| && (4) \end{aligned}$$

It represents the average fraction of pairs that are not correctly ordered. For ranking systems, this loss is natural and is related to the *precision* which is a common error measure in Information Retrieval (we take here the definition of [14]):

$$\text{precision}(f, x, \mathbf{Y}) = \frac{1}{|\mathbf{Y}|} \sum_{k \in \mathbf{Y}} \frac{|\{l \in \mathbf{Y} \text{ s.t. } r_l(x) \geq r_k(x)\}|}{|\{l \in \{1, .., Q\} \text{ s.t. } r_l(x) \geq r_k(x)\}|}$$

from which a loss can be directly deduced. All these loss functions have values between 0 and 1 and have been discussed in [14]. Good systems should have a high precision and a low Hamming or Ranking Loss.

For multi-label linear models, we need to define a way of minimizing the empirical error measured by the appropriate cost function and at the same time to control the complexity of the resulting model. This will be done by introducing notions of margin and regularization as has been done for the two-class case in the definition of SVMs. All our reasoning is based on the idea that having a large margin plus a regularization method leads to good systems. It is an *a priori* as another. In the bayesian framework as well as in classical VC-theory, the prior assumptions that either the probabilistic model is well defined or that minimizing a bound on the generalization error provides systems with low generalization error, are also *a priori* that can be discussed. The latter are certainly more theoretically motivated and we believe that a rigorous theoretical analysis such as [2, 6] about what complexity measure is appropriate for a multi-label system should be done. We postpone this analysis to another study and focus our attention on the definition of new systems, how to implement them and what experimental results they provide.

3 Binary Approach

We assume here that the function f is computed as in (2). That means that the decision boundaries are defined by the hyperplanes $\langle w_k, x\rangle + b_k = 0$. By analogy with the two-class case, we define the margin of f on $(x, \mathbf{Y})$ to be the signed distance between x and the decision boundary defined as $\{x \text{ s.t. } \exists k, \langle w_k, x\rangle + b_k = 0\}$. It is equal to:

$$\min_k y_k \frac{\langle w_k, x\rangle + b_k}{\|w_k\|}$$

where y_k has been defined in the introduction and is a binary element equal to $+1$ if label k is in $\mathbf{Y}$, -1 otherwise. For a learning set S, the margin can also be defined as to be:

$$\min_{(x,\mathbf{Y})\in S} \min_k y_k \frac{\langle w_k, x\rangle + b_k}{\|w_k\|}$$

Assuming that the Hamming Loss on the learning set S is zero, the large margin paradigm we follow consists in maximizing the margin. By normalizing the parameters (w_k, b_k) such that:

$$\forall (x, \mathbf{Y}) \in S, \;\; y_k(\langle w_k, x\rangle + b_k) \geq 1$$

with an equality for at least one x, the margin on $S = \{(x_i, \mathbf{Y}_i)\}_{i=1,..,Q}$ is equal to $\min_k \frac{1}{\|w_k\|}$. Here, $\mathbf{Y}_i$ is identified with its binary representation: $(y_{i1}, .., y_{iQ}) \in \{-1, +1\}^Q$. Maximizing the margin or minimizing its inverse yields to the following problem:

Problem 1

$$\min_{w_k, b_k} \max_k \|w_k\|^2$$

$$\textit{subject to :} \quad y_{ik}\left(\langle w_k, x_i\rangle + b_k\right) \geq 1,$$

At this point, we have assumed that the Hamming Loss was zero which is unlikely to be the case in general. For our concern, the Hamming Loss can be computed as:

$$HL(f, x, \mathbf{Y}) = \frac{1}{Q}\sum_{k=1}^{Q} \theta\left(-y_{ik}(\langle w_k, x_i\rangle + b_k)\right)$$

where $\theta(t) = 0$ for $t \leq 0$ and $\theta(t) = 1$ for $t > 0$. We can thus generalize the previous problem by combining the minimization of the Hamming Loss and the maximization of the margin:

Problem 2

$$\min_{w_k, b_k} \left(\max_k \|w_k\|^2\right) + C\sum_{i=1}^{m}\frac{1}{Q}\sum_{k=1}^{Q}\theta\left(-1 + \xi_{ik}\right)$$

$$\textit{subject to :} \quad y_{ik}\left(\langle w_k, x_i\rangle + b_k\right) \geq 1 - \xi_{ik},$$

$$\xi_{ik} \geq 0$$

This problem is non-convex and is difficult to solve (it inherits from the NP-Hardness of the related two-class problems which is a classical SVM with threshold margin). We consider a simpler problem by upper bounding the $\theta(-1 + t)$ function by the linear one. We have then:

Problem 3

$$\min_{w_k, b_k} \left(\max_k \|w_k\|^2\right) + \frac{C}{Q}\sum_{i=1}^{m}\sum_{k=1}^{Q}\xi_{ik}$$

$$\textit{subject to :} \quad y_{ik}\left(\langle w_k, x_i\rangle + b_k\right) \geq 1 - \xi_{ik},$$
$$\xi_{ik} \geq 0$$

This problem is a convex problem with linear constraints and it can be solved by using a gradient descent or other classical optimization method. Note that when the constant C is infinite which corresponds to the case where the minimization of the Hamming Loss is actually the main objective function, then the system is completely decoupled and the optimization can be done on each parameter (w_k, b_k) independently. Within this framework, the binary approach developed in [9] appears as to be the same as ours when $C = \infty$. For finite C, the optimization can not be done so easily, except if we approximate the maximum over the w_k's by:

$$\max_k \|w_k\|^2 \leq \sum_{k=1}^{Q} \|w_k\|^2 \leq Q \max_k \|w_k\|^2$$

In that case, the problem becomes completely separated in the sense that optimizing with respect to w_k does not influence the optimization with respect to w_l for $l \neq k$. Note that the optimization procedure is the same as the one-against-the-rest approach developed in the multi-class setting.

In problem 3, we have replaced the Hamming Loss by its linear approximation (here computed on $(x, \mathbf{Y})$):

$$AHL((w_k, b_k)_{k=1,..,Q}, x, \mathbf{Y}) = \frac{1}{Q} \sum_{k=1}^{Q} |1 - y_k(\langle w_k, x \rangle + b_k)|_+$$

where $|.|_+$ is a function from $\mathbb{R}$ to $\mathbb{R}_+$ that is the identity on $\mathbb{R}_+$ and that equals zero on $\mathbb{R}_-$. The linear system we just defined is then designed to minimize the AHL function rather than the Hamming distance between the binary representation of the label sets and the function f defined in (2). During learning, it tends to put the output $f(x)$ close to the targets $\mathbf{Y}$ in terms of the distance derived from the AHL. When a new input x is given, the output of f should then be computed via:

$$f(x) = \text{argmin}_{\mathbf{Y} \in \mathcal{Y}} \sum_{k=1}^{Q} |1 - y_k(\langle w_k, x \rangle + b_k)|_+ \tag{5}$$

where $\mathcal{Y}$ contains all label sets that are potential outputs. It turns out that, when all label sets are acceptable outputs ($\mathcal{Y} = \{-1, +1\}^Q$), the previous equation rewrites as (2). In some cases where all label sets are not allowed ($\mathcal{Y} \subsetneq \{-1, +1\}^Q$), both computation are different (see figure 1) and $f(x)$ should be calculated as previously rather than with (2). Such cases arise when for instance a Error Correcting Output Code (ECOC) is used to solve multi-class problems. In [5] is presented a method based on error correcting code which consists in transforming a multi-class problem into a multi-label one and in using the fact that not all label sets are allowed. When the learning system outputs an impossible label set, the choice of the correcting code makes it possible to find a potentially correct label set whose Hamming distance is the closest to the output. If the system derived from problem 3 is used with the ECOC approach, considering that the Hamming Loss is not the quantity that is minimized, the computation of the output should be done by minimizing the Hamming distance via equation (5). It is possible to rewrite it as:

$$f(x) = \text{argmin}_{\mathbf{Y} \in \mathcal{Y}} \sum_{k=1}^{Q} |y_k - \sigma(\langle w_k, x \rangle + b_k)| \tag{6}$$

where σ is the linear function thresholded at -1 (resp. $+1$) with value -1 (resp. $+1$).

4 Ranking Approach

4.1 Motivation

As previously noticed in the introduction, the binary approach is not appropriate for problems where correlation between labels exist. To illustrate this point consider figure 2. There are only three labels. One of them (label 1) is present for all points in the learning set. The binary approach leads to a system that will fail to separate, for instance, points with label 3 from points of label sets not containing 3,

Labels	1	2	3	4	5
$\mathbf{Y}_1$	1	1	1	1	1
$\mathbf{Y}_2$	-1	-1	-1	-1	-1
Real output $(\langle w_k, x\rangle + b_k)_{k=1,..,5}$	1	-0.05	-0.05	-0.05	-0.05

Figure 1: Assume we have 5 labels and there are only two possible label sets: one denoted by $\mathbf{Y}_1$ with binary representation $(1, .., 1)$ and the other one $\mathbf{Y}_2 = (-1, .., -1)$. The Hamming distance between the output and $\mathbf{Y}_1$ is 4, although it is 1 for $\mathbf{Y}_2$. Recall that the Hamming distance is computed on the signed vector of the output. The AHL between the output and $\mathbf{Y}_1$ is 4.2 although it is 5.8 for $\mathbf{Y}_2$. If the final output is computed with the Hamming distance, then $\mathbf{Y}_2$ is chosen. If it is computed via the AHL, then $\mathbf{Y}_1$ is chosen.

that is, on points of label 1 and 2. We see then that the expressible power of a binary system can be quite low when simple configurations occur. If we consider the ranking approach, one can imagine the following solution: $w_1 = 0$, $b_1 = \infty$, (w_2, b_2) is the hyperplane separating class 2 from class 3, and $(w_3, b_3) = -(w_2, b_2)$. By taking the number of labels at point x to be $s(x) = \langle w, x\rangle + b$ where $w = (-1, 1)$ and $b = 0$, we have a simple multi-label system that separates all the regions exactly.

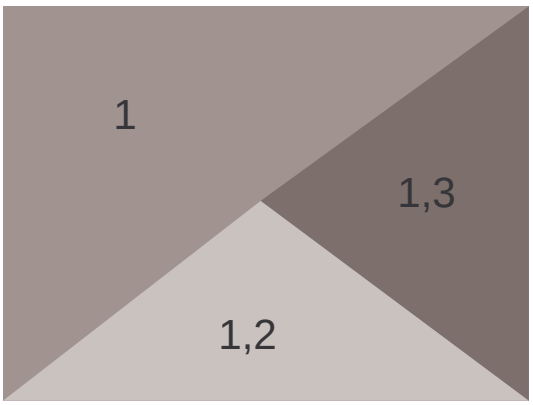

Figure 2: Three labels and three regions in the input space. The upper left region is labelled with 1. The bottom right region is partitioned into two sub-regions with labels $1, 2$ or $1, 3$.

To make this point more concrete we sampled 50 points uniformly on $[0, 1]^2$ and solved all optimization problems with $C = \infty$. On the learning set the Hamming Loss for the binary approach was 0.08.

4.2 Kernel method for Ranking

Our goal in this part is to define a linear model that minimize the Ranking Loss while having a low complexity. The notion of complexity we will use here is the margin. For systems that rank the values of $\langle w_k, x\rangle + b_k$, the decision boundaries for x are defined by the hyperplanes whose equations are $\langle w_k - w_l, x\rangle + b_k - b_l = 0$, where k belongs to the label sets of x and l does not. So, the margin of $(x, \mathbf{Y})$ can be expressed as:

$$\min_{k\in\mathbf{Y}, l\in\bar{\mathbf{Y}}} \frac{\langle w_k - w_l, x\rangle + b_k - b_l}{||w_k - w_l||}$$

Considering that all the data in the learning set S are well ranked, we can normalize the parameters w_k's such that:

$$\langle w_k - w_l, x\rangle + b_k - b_l \geq 1$$

with equality for some x part of S, and $(k,l) \in \mathbf{Y} \times \bar{\mathbf{Y}}$. Maximizing the margin on the whole learning set can then be done via the following problem:

Problem 4

$$\max_{w_j, j=1,..,Q} \min_{(x,\mathbf{Y}) \in S} \min_{k \in \mathbf{Y}, l \in \bar{\mathbf{Y}}} \frac{1}{\|w_k - w_l\|^2}$$

$$\textit{subject to :} \qquad \langle w_k - w_l, x_i \rangle + b_k - b_l \geq 1, \;\; (k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$$

In the case where the problem is not ill-conditioned (two labels are always co-occurring), the objective function can be replaced by: $\max_{w_j} \min_{k,l} \frac{1}{\|w_k - w_l\|^2}$. The previous problem can then be recast as:

Problem 5

$$\min_{w_j, j=1,..,Q} \max_{k,l} \|w_k - w_l\|^2$$

$$\textit{subject to :} \qquad \langle w_k - w_l, x_i \rangle + b_k - b_l \geq 1, \;\; (k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$$

In order to get a simpler optimization procedure, we approximate this maximum by the sum as we did for the binary approach. This leads us finally to the following problem (in the case the learning set can be ranked exactly):

Problem 6

$$\min_{w_j, j=1,..,Q} \sum_{k,l=1}^{Q} \|w_k - w_l\|^2$$

$$\textit{subject to :} \qquad \langle w_k - w_l, x_i \rangle + b_k - b_l \geq 1, \;\; (k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$$

Note that a shift in the parameters w_k does not change the ranking. We can thus require that $\sum_{k=1}^{Q} w_k = 0$ and add this constraint. In that case, the previous problem is equivalent to:

Problem 7

$$\min_{w_j, j=1,..,Q} \sum_{k=1}^{Q} \|w_k\|^2$$

$$\textit{subject to :} \qquad \langle w_k - w_l, x_i \rangle + b_k - b_l \geq 1, \;\; (k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$$

To see this, note that when $\sum_{k=1}^{Q} w_k = 0$, we have $\sum_{k,l=1}^{Q} \|w_k - w_l\|^2 \propto \sum_{k=1}^{Q} \|w_k\|^2$, and that the solution of the latter problem does not change if we add the constraint $\sum_{k=1}^{Q} w_k = 0$.

To generalize our calculations in the case where the learning set can not be ranked exactly, we follow the same reasoning as for the binary case: the ultimate goal would be to minimize the margin and at one and a same time to minimize the Ranking Loss. The latter can be expressed quite directly by extending the constraints of the previous problems. Indeed, if we have $\langle w_k - w_l, x_i \rangle + b_k - b_l \geq 1 - \xi_{ikl}$ for $(k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$, then the Ranking Loss on the learning set S is:

$$\sum_{i=1}^{m} \frac{1}{|\mathbf{Y}_i||\bar{\mathbf{Y}}_i|} \sum_{(k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i} \theta(-1 + \xi_{ikl})$$

Once again, we approximate the functions $\theta(-1 + \xi_{ikl})$ by only ξ_{ikl}, and this gives the final quadratic optimization problem:

Problem 8

$$\min_{w_j, j=1,..,Q} \sum_{k=1}^{Q} \|w_k\|^2 + C \sum_{i=1}^{m} \frac{1}{|\mathbf{Y}_i||\bar{\mathbf{Y}}_i|} \sum_{(k,l)\in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i} \xi_{ikl}$$

$$\textit{subject to :} \qquad \langle w_k - w_l, x_i \rangle + b_k - b_l \geq 1 - \xi_{ikl}, \;\; (k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$$

$$\xi_{ikl} \geq 0$$

In the case where the label sets $\mathbf{Y}_i$ have all a size of 1, we find the same optimization problem as has been derived for multi-class Support Vector Machines [19]. For this reason, we call the solution of this problem a ranking Support Vector Machine (Rank-SVM).

4.3 Set size prediction

So far we have only developed a ranking system. To obtain a complete multi-label system we need to design a set size predictor $s(x)$. A natural way of doing this is to look for inspiration from the binary approach. The latter can indeed be interpreted as a ranking system whose ranks are derived from the real values $(f_1, .., f_Q)$. The predictor of the set size is then quite simple: $s(x) = |\{f_k(x) > 0\}|$ is the number of f_k that are greater than 0. The function $s(x)$ is computed from a threshold value that differentiates labels in the target set from others. For the ranking system introduced in the previous section we generalize this idea by designing a function $s(x) = |\{f_k(x) > t(x)\}|$. The remaining problem now is to choose $t(x)$ which is done by solving a learning problem. The training data are composed by the $(f_1(x_i), .., f_Q(x_i))$ given by the ranking system, and by the target values defined by:

$$t(x_i) = \text{argmin}_t \left|\{k \in \mathbf{Y} \text{ s.t. } f_k(x_i) \leq t\}\right| + \left|\left\{k \in \bar{\mathbf{Y}} \text{ s.t. } f_k(x_i) \geq t\right\}\right|$$

When the minimum is not unique and the optimal values are a segment, we choose the middle of this segment. We refer to this method of predicting the set size as the *threshold based* method.

Note that we could have followed a much simpler scheme to build the function $s(x)$. A naive method would be to consider the set size prediction as a regression problem on the original training data with the targets $(|\mathbf{Y}_i|)_{i=1,..,m}$ and to use any regression learning system. This however does not provide a satisfactory solution mainly because it does not take into account how the ranking is performed. In particular, when there are some errors in the ranking, it does not learn how to compensate these errors although the threshold based approach tries to learn the best threshold with respect to these errors.

4.4 Sum up

Since the model we consider is built from two different sub-systems, let us sum up what we described. We decompose our multi-label model into two parts. Both of them are based on dot products and linear models.

- The first one ranks the labels and is obtained via problem 8. The result is a set of parameters $(w_k, b_k)_{k=1,..,Q}$. The ranking is done on the outputs $r_k(x) = \langle w_k, x \rangle + b_k$ for $k = 1, .., Q$.
- The second model predicts a threshold $t(x)$ and all integer k such that $r_k(x) > t(x)$ are considered to belong to the label set.

5 Categorical regression

As we discussed in the introduction, multi-label problems are very general: many known classification tasks can be cast into multi-label problems. As an example, we develop in this section what we called the categorical regression problem. It can be thought of as the problem of minimizing the loss:

$$CR(h, x, y) = \frac{1}{Q} |h(x) - y|$$

where y is the label of x (here there is only one label for a x) and h is a multi-classifier (i.e. function from $\mathcal{X}$ into $\{1, .., Q\}$). The name regression comes from the fact that this cost function is a ℓ_1 cost function as in regression and categorical comes from the fact that $h(x)$ is discrete. Such a setting arises when the mistakes are ordered. For instance, one could imagine a situation where predicting label 1 although the true label is 4 is worse than predicting label 3. Consider the task of predicting the quality of a web page from labels given by individuals. The latter give an opinion like *very bad*, *bad*, *neutral*, *good* or *very good* web page, and the point is not only to have few mistakes, but to be able to minimize the difference between the prediction and the true opinion. A system that outputs "very bad" although the right answer is "very good" is worse than a system giving the "good" answer. The other way works the same: a "very good" output although it is "very bad" is worse than "bad". Such a categorical regression is in a way related to ordinal regression, the difference being no order on the targets are assumed here, only order on the mistakes.

We propose to deal with categorical regression via the multi-label approach. The interest is actually that it gives a natural way of solving the categorical regression problem when the right setting is used. Rather than coding the labels as integers, let us encode them as binary vector of $\{+1, -1\}$ components:

$$\text{label } i = \left(\underbrace{1, .., 1}_{i \text{ times}}, -1, .., -1 \right) \tag{7}$$

With this encoding the loss CR defined previously can be expressed in terms of the Hamming Loss:

$$CR(h, x, y) = HL(\tilde{h}, x, \mathbf{Y})$$

where $\tilde{h}$ is the function h when its output is encoded as (7), and $\mathbf{Y}$ is the encoded label corresponding to y. We have seen that the minimization of the Hamming Loss leads naturally to the binary approach for the multi-label problem. We can thus build the same system as the one defined in the binary approach section, that is, a linear system composed of many two-class sub-systems that learn how to recognize when the components of the label associated to x is $+1$ or -1. All possible label sets are not allowed here. As for the ECOC approach discussed previously, the function s is thus computed via (6) where $\mathcal{Y}$ contains labels $1, .., Q$ encoded as (7).

We see that interpreting the categorical regression problem as a multi-label one allows to design very easily a learning system that tends to minimize the right cost function while having a controlled complexity via a regularized learning method.

6 Experiments

6.1 Implementation considerations

Solving a constrained quadratic problem as those we introduced in previous sections requires an amount of memory that is quadratic in terms of the learning set size and it is generally solved in $O(m^3)$ computational steps where we have put into the O the number of labels. Such a complexity is too high to apply these methods in many real datasets. To circumvent this limitation, we propose to use a linearization method known as Franke and Wolfe's method [7] in conjunction with a predictor-corrector logarithmic barrier procedure inspired by [1]. Details are described in appendix A with all the calculations relative to the implementation. The memory cost of the method becomes then $O(mQQ_{max})$ where $Q_{max} = \max_i |\mathbf{Y}_i|$ is the maximum number of labels. In many applications, Q is much larger than Q_{max}. The time cost of each iteration is $O(m^2 Q)$.

6.2 Prostate Cancer Dataset

The first problem we consider comes from a real application in Medical Biology. The dataset is formed by 67 examples in a space of 7129 features and of 7 labels. The inputs are results of Micro-array experiments on different tissues coming from patients with different form of Prostate Cancer. The seven

Label	Description	Number of points in the class
1	Peripheral Zone	9
2	Central Zone	3
3	Dysplasia (Cancer precursor stage)	3
4	Stroma	1
5	Benign Prostate Hyperplasia	18
6	G3 (Cancer)	13
7	G4 (Cancer)	27

Figure 3: Description of the labels for the Prostate Cancer dataset.

labels are described in table 3 and represent either the position of the tissue in the body (peripheral zone, etc...) or the degree of malignity of the disease (G3-G4 Cancer).

The label sets have a maximum size of 2 and only 7 points are related to more than one label. Table 3 shows the distribution of the labels. To assess the quality of the classifier, we computed a leave-one-out estimate of the different losses we introduced before for the ranking approach as well as for the binary approach with linear models and a constant $C = \infty$. For the ranking loss and the precision, the binary approach was considered as a ranking approach by interpreting the real values $(f_1, .., f_Q)$ as the ranking values $(r_1, .., r_Q)$. Table 4 reports the results.

	Precision	Ranking Loss	Hamming Loss	One-error
Ranking Approach	0.795	0.128	0.100	0.328
Binary Approach	0.783	0.123	0.094	0.343

Figure 4: Leave-one-out estimate of different losses for the Prostate Cancer dataset seen as a multi-label problem (7 labels).

Since the goal in this particular application is to discriminate the malign cells from the benign ones, we also applied the multi-labelled approach but for labels 5-7. This reduces the number of labels to 3 and the number of points in the learning set to 55. With the same setting as previously, we obtained the results reported in table 5.

	Precision	Ranking Loss	Hamming Loss	One-error
Ranking Approach	0.841	0.200	0.224	0.291
Binary Approach	0.842	0.200	0.188	0.273

Figure 5: Leave-one-out estimate of different losses for the Prostate Cancer dataset seen as a multi-label problem (3 labels).

All the results are quite similar and can not be considered as significantly different from one to another. The partial conclusion of these experiments is then that defining a ranking approach does not improve that much the performance compared to a simple binary approach. Given the nature of the problem we have considered, such a conclusion seems quite natural. If we return back to the main motivation underlying the definition of the ranking approach, we see that it has been introduced to compensate the relative lack of expressible power of the binary approach: in one simple example, we have showed that the latter cannot provide a satisfying discrimination between the label sets although it is possible with our direct approach. In the experiments we have just done, the number of feature is enormous compared to the number of training points. In such situation, any learning system is able to learn.

In some sense, we also believe that this experiment with very few data points shows that the ranking based system does not overfit compare to the binary one. Such a statement as well as an exhaustive

experimental study remains to be done and will be the subject of a forthcoming paper.

The problem with 3 labels we just considered is very close to a pure multi-class problem: except for 3 points, the sizes of the label sets are 1 (i.e. $Q_{max} = 1$). To see whether adding these three points helps or not, we performed the same experiments as before but with a linear multi-class SVM without these three points. As a point of comparison, we also run a linear one-against-the-rest SVM on this problem. Results are reported in table 6 and should be compared to the one-error of the previous multi-label systems. We then seen that there is not a significant difference to state whether one method is better than another one. We did the same experiment for the problem with 7 labels where only 7 points are really multi-labelled. We removed them from the learning set which had then 60 learning points. Only the one-against-the-rest SVM seems to be significantly worse than the other methods on this particular problem.

	Error for 3 labels	Error for 7 labels
Multi-class SVM	0.29	0.32
One-against-the-rest SVM	0.31	0.40

Figure 6: Leave-one-out estimate of different losses for the Prostate Cancer dataset seen as a mutli-class problem. Here, the learning set size is 52 (resp. 60) for the problem with 3 (resp. 7) labels.

No clear partial conclusion can be deduced from this experiments: all multi-label systems seem to perform equally. This can be explained by the nature of the problem which is very close to a multi-class one. In next section we consider a different problem with many multi-labelled points.

6.3 Yeast Genome Dataset

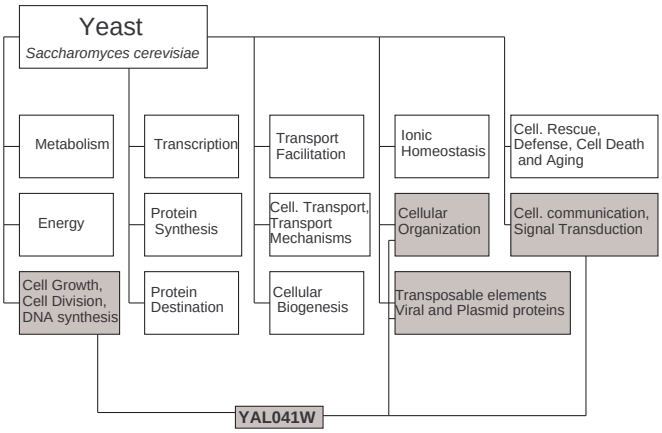

Figure 7: First level of the hierarchy of the gene functional classes. There are 14 groups. One gene, for instance the gene YAL041W can belong to different groups (shaded in grey on the figure).

The genome of the Yeast Saccharomyces cerevisiae has been completely analyzed via Micro-array experiments and is available from the MIPS database [11]. It provides the researcher with the expression level of many genes during the diauxic shift, the mitotic cell division cycle, sporulation and temperature reduction shocks. Functional classes of many genes (around 3300) have been already determined and classified into a hierarchy of function. The first level is depicted in figure 7. Many genes can belong to many classes. The problem we try to solve in this section is to predict the first level of the functional hierarchy by using the results of the Micro-array experiments. We used the same set as in [12]: each input is the concatenation of the expression level and the phylogenetic profile of a gene. We have a dataset of 2417 points of 103 features. We have considered here only genes whose functional class is

already determined. The problem is typically a multi-label problem with 14 labels. In the training set, the maximum size for a label set is 11. To assess the quality of the system, we randomly divided the dataset into a learning set of size 1500 and a test set of size 917 and consider the same split for all methods.

We have assessed the performance of three methods: the binary approach used in conjunction with two-class polynomial SVMs, the ranking approach with different polynomial kernels and Boostexter. The latter is a multi-label method derived from Boosting and is presented in [14]. We have used the standard package that consists in boosting basic stumps and stopped at iteration 1000.

We assessed the quality of these methods from two perspectives. First as a ranking system with the Ranking Loss and the precision. In that case, for the binary approach, the real outputs of the two-class SVMs were used as ranking values. Second, the methods were compared as multi-label systems using the Hamming Loss. To measure the Hamming Loss with Boostexter we used a threshold based $s(x)$ function in combination with the ranking given by the algorithm.

	Rank-SVM				Binary-SVM			
degree	2	3	4	5	2	3	4	5
Precision	**0.703**	**0.740**	**0.746**	**0.762**	0.692	0.721	0.714	0.753
Ranking Loss	**0.227**	**0.191**	**0.190**	**0.175**	0.241	0.212	0.196	0.184
Hamming Loss	**0.238**	**0.217**	**0.209**	**0.201**	0.247	0.224	0.211	0.207
one-error	**0.334**	**0.262**	**0.255**	**0.232**	0.341	0.306	0.267	0.250

Figure 8: Polynomials of degree 2-5. Loss functions for the rank-SVM and the binary approach based on two-class SVMs. Considering the size of the problem, two values different from less than 0.01 are not significantly different. Bold values represent superior performance comparing classifiers with the same kernel.

For rank-SVMs and for two-class SVMs in the binary approach we choose polynomial kernels of degrees two to nine (experiments on two-class problems using the Yeast data in [12] already showed that polynomial kernels were appropriate for this task). Boostexter was used with the standard stump weak learner and was stopped after 1000 iterations. Results are reported in tables 8, 9 and 10.

	Rank-SVM				Binary-SVM			
degree	6	7	8	9	6	7	8	9
Precision	**0.765**	**0.770**	**0.773**	0.769	0.760	0.765	0.770	0.769
Ranking Loss	**0.170**	**0.166**	**0.163**	**0.163**	0.176	0.170	0.165	0.164
Hamming Loss	**0.199**	**0.198**	0.196	0.197	0.200	0.199	**0.195**	**0.195**
one-error	0.232	**0.223**	**0.217**	**0.225**	0.232	0.227	0.218	0.226

Figure 9: Polynomials of degree 6-9. Loss functions for the rank-SVM and the binary approach based on two-class SVMs. Considering the size of the problem, two values different from less than 0.01 are not significantly different. Bold values represent superior performance comparing classifiers with the same kernel.

	Boostexter (1000 iterations)
Precision	0.717
Ranking Loss	0.298
Hamming Loss	0.237
one-error	0.302

Figure 10: Loss functions for Boostexter. Note that these results are worse than with the binary approach or with rank-SVM.

Note that Boostexter performs quite poorly on this dataset compared to SVM-based approaches. This may be due to the simple decision function realized by Boostexter. One of the main advantages of the SVM-based approaches is the ability to incorporate priori knowledge into the kernel and control complexity via the kernel and regularization. We believe this may also be possible with Boostexter but we are not aware of any work in this area.

To compare the binary and the rank-SVM we put in bold the best results for each kernel. For all kernels and for almost all losses, the combination ranking based SVM approach is better than the binary one. In terms of the Ranking Loss, the difference is significantly in favor of the rank-SVM. It is consistent with the fact that this system tends to minimize this particular loss function. It is worth noticing that when the kernel becomes more and more complex the difference between rank-SVM and the binary method disappears.

7 Feature Selection for multi-label problems

The problem of feature selection has already been addressed by different ways in the kernel machine community [18, 8, 17]. Choosing a set of features that gives a good generalization error is a central problem for the analysis of Micro-array data. The latter are indeed formed by many inputs whose features generally represents the genes of a tissue. Choosing a subset of features is then the same as choosing a subset of genes which may provide a lot of indications for the biologist. In a previous report [17], we have shown how to apply the minimization of the ℓ_0 norm of a linear system to do feature selection. The results concern mainly two-class and multi-class classification. Here, we apply the same technic but to multi-labelled problems. Note that the latter are omnipresent in biological and medical applications.

As in [17], we consider the following multiplicative update rule technic:

1. Inputs: Learning set $S = ((x_i, \mathbf{Y}_i))_{i=1,..,m}$.
2. Start with $\left(w_1^0, b_1^0, .., w_Q^0, b_Q^0\right) = (0, .., 0)$.
3. Assume $(w_1^t, b_1^t, .., w_Q^t, b_Q^t)$ is given. At iteration $t+1$, solve the problem:

   Problem 9
   $$\min_{w_k, b_k} \sum_{k=1}^{m} \|w_k\|^2$$
   $$\textit{subject to :} \qquad \left\langle w_k * w_k^t, x_i \right\rangle - \left\langle w_l * w_l^t, x_i \right\rangle + b_k - b_l \geq 1, \quad \textit{for } (k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i$$

   where $w * w^t$ is the component-wise multiplication between w and w^t.
4. Let w be the solution. Set: $w^{t+1} = w * w^t$.
5. Unless $w^{t+1} = w^t$, go back to step 2.
6. Output w^{t+1}.

This technic is an approximation scheme to minimize the number of non-zero components of $(w_1, .., w_Q)$ while keeping the constraints of problem 9 satisfied. It is proved that it converges [17]. If the multi-label problems is actually a ranking problem, then this feature selection method is appropriate. For real multi-label problems, a step is still missing: the computation of the label sets size $s(x)$. As previously, it can be done with the threshold based approach.

Experiments about this feature selection technic applied to biological data will be the subject of a next report.

8 Discussion

So far, we have only discussed about linear systems. Despite their attractive property such as simplicity and easiness of interpretation, they may appear as a restrictive system when complex problems have

to be solved. This apparent drawback vanishes when only dot-products are involved. In that case, the kernel trick [15, 4] can be used and transform a linear model into a potentially highly non-linear one. To do so, it suffices to replace all the dot products $\langle x_i, x_j \rangle$ by $k(x_i, x_j)$ where k is a *kernel*. Examples of such kernels are the gaussian kernel $\exp\left(-\|x_i - x_j\|^2/\sigma^2\right)$ or the polynomial one $\left(\langle x_i, x_j \rangle + 1\right)^d$ for $d \in \mathbb{N}\backslash\{0\}$. If these dot products are replaced, we see that the other dot products $\langle w_k, x_i \rangle$ can also be computed via equation (12) (in appendix) and by linearity of the dot product. Thus, the rank-SVM we introduced can also be defined for different dot product computed as kernels. Note that such a reasoning can not be held for the feature selection procedure that highly relies on the linearity of the model.

To assess the quality of the rank-SVM a large scale experimental study should be developed. We have begun here some experiments that show that the rank-SVM can lead to improvements compare to other systems but we have not identified a significant gap that could justify its systematic use. We believe however that the improvements can be better when exact learning is not requires. In such case, the binary approach suffers indeed from an important drawback: there is no correlation between the errors. So the minimization of the two-class SVM errors will not lead necessarily to an overall multi-label system with a small error.

One of the major drawback of our method is the space complexity: it requires in worst case a amount of memory proportional to mQ^2, which can become unaffordable when Q is large. Consider for instance, the Yeast Genome Dataset. It contains thousands of genes and around 190 labels. The number of variables one has to handle is then of the order of the million, which becomes quite large. Hopefully, in many situation, the labels are structured in a hierarchy where each node concerns a dozen of labels rather than a hundreds. For the Yeast, such a hierarchy can be seen at the address `http://www.mips.biochem.mpg.de/proj/yeast/`. It becomes then possible to solve the learning task by defining for each node of the hierarchy a multi-label system that classifies the input into one or many of its sub-classes. The global system is then a decision tree where, at each node, many decisions corresponding to many labels are taken.

9 Conclusion

In this paper, we have defined a whole system to deal with multi-label problems. The main contribution is the definition of a ranking based SVM that extend the use of the latter to many problems in the area of Bioinformatics and Text Mining. We have not done any test about the latter application domain. Our primary goal was to define a multi-label system and not only a ranking one. However, assessing the performance of our ranking system with respect to existing methods is important and will be the subject of a forthcoming paper. One of the consequence of our approach is the definition of a feature selection method for ranking problems: it gives a subset of the features that are compatible with the constraints imposed by the learning set. Such a feature selection method is of particular interest in the field of bioinformatics as one is often interested in interpretability of a multi-label decision rule. For example one could be interested in a small set of genes which is discriminative in a multi-condition physical disorder.

References

[1] A. Altman and J. Gondzio. Hopdm - a higher order primal-dual method for large scale linear programming. *European Journal of Operational Research*, 66(1):159–160, 1993.

[2] P.L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44:525–536, 1998.

[3] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, 1992. ACM.

[4] N. Cristianini and J. Shawe-Taylor. *Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[5] T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[6] A. Elisseeff, Y. Guermeur, and H. Paugam-Moisy. Margin error and generalization capabilities of multi-class discriminant systems. Technical Report 99-051, NeuroCOLT2, 1999.

[7] M. Franke and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly 3*, pages 95–110, 1956.

[8] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *BIOWulf Technical Report*, 2000.

[9] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[10] A. McCallum. Multi-label text classification with a mixture model trained by em. *AAAI'99 Workshop on Text Learning.*, 1999.

[11] H.W. Mewes, K. Albermann, K. Heumann, S. Liebl, and F. Pfeiffer. Mips: A database for protein sequences, homology data and yeast genome information. *Nucleic Acid Research*, 25:28–30, 1997.

[12] P. Pavlidis, J. Weston, J. Cai, and W.N. Grundy. Combining microarray expression data and phylogenetic profiles to learn functional categories using support vector machines. In *RECOMB*, pages 242–248, 2001.

[13] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3), 1999.

[14] R.E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[15] B. Schölkopf and A. Smola. *Learning with kernels*. MIT Press, Cambridge MA, 2001. *To appear*.

[16] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, N.Y., 1998.

[17] J. Weston, A. Elisseeff, and B. Schölkopf. Use of the ℓ_0-norm with linear models and kernel methods. *BIOWulf Technical Report*, 2001.

[18] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. *Advances in Neural Information Processing Systems*, 13, 2000.

[19] J. Weston and C. Watkins. Multi-class support vector machines. Technical Report 98-04, Royal Holloway, University of London, 1998.

A Solution of the optimization problem

A.1 Computation of the dual

To solve problem 8, we introduce the dual variables $\alpha_{ikl} \geq 0$ related to the constraints:

$$\langle w_k - w_l, x_i \rangle + b_k - b_l - 1 + \xi_{ikl} \geq 0$$

and the variables $\eta_{ikl} \geq 0$ related to the constraints $\xi_{ikl} \geq 0$. The lagrangian can the be computed:

$$L = \frac{1}{2} \sum_{k=1}^{Q} \|w_k\|^2 + C \sum_{i=1}^{m} \frac{1}{|\mathbf{Y}_i||\bar{\mathbf{Y}}_i|} \sum_{(k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i} \xi_{ikl} - \dots$$
$$\dots \sum_{i=1}^{m} \sum_{(k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i} \alpha_{ikl} \left(\langle w_k - w_l, x_i \rangle + b_k - b_l - 1 + \xi_{ikl} \right) - \sum_{i=1}^{m} \sum_{(k,l) \in \mathbf{Y}_i \times \bar{\mathbf{Y}}_i} \eta_{ikl} \quad (8)$$

Setting $\partial_{b_k} L = 0$ at the optimum yields:

$$\sum_{i=1}^{m} \sum_{(j,l)\in(\mathbf{Y}_i,\bar{\mathbf{Y}}_i)} c_{ijl}\alpha_{ijl} = 0 \tag{9}$$

with

$$c_{ijl} = \begin{cases} 0 & \text{if } j \neq k \text{ and } l \neq k \\ +1 & \text{if } j = k \\ -1 & \text{if } l = k \end{cases} \tag{10}$$

Note that c_{ijl} depends on k. We have dropped the index k to avoid too many indexing. Setting $\partial_{\xi_{ikl}} L = 0$ yields:

$$\frac{C}{|\mathbf{Y}_i||\bar{\mathbf{Y}}_i|} = \alpha_{ikl} + \eta_{ikl} \tag{11}$$

At last, setting $\partial_{w_k} L = 0$ yields:

$$w_k = \sum_{i=1}^{m} \left(\sum_{(j,l)\in(\mathbf{Y}_i,\bar{\mathbf{Y}}_i)} c_{ijl}\alpha_{ijl} \right) x_i \tag{12}$$

where c_{ijl} is related to index k via equation (9). For the sake of notation, let us define:

$$\beta_{ki} = \sum_{(j,l)\in(\mathbf{Y}_i,\bar{\mathbf{Y}}_i)} c_{ijl}\alpha_{ijl} \tag{13}$$

Then, we have: $w_k = \sum_{i=1}^{m} \beta_{ki} x_i$.

The dual of problem 8 can then be expressed. In order to have as simple notation as possible, we will express it with both variables β_{ki} and α_{ikl}. We have:

Problem 10

$$\max_{\alpha_{ikl}} W(\alpha) = -\frac{1}{2} \sum_{k=1}^{Q} \sum_{h,i=1}^{m} \beta_{kh}\beta_{ki} \langle x_h, x_i \rangle + \sum_{i=1}^{m} \sum_{(k,l)\in(\mathbf{Y}_i,\bar{\mathbf{Y}}_i)} \alpha_{ikl}$$

$$\textit{subject to :} \qquad \alpha_{ikl} \in [0, \frac{C}{C_i}]$$

$$\sum_{i=1}^{m} \sum_{(j,l)\in(\mathbf{Y}_i,\bar{\mathbf{Y}}_i)} c_{ijl}\alpha_{ijl} = 0, \quad \textit{for } k = 1,..,Q$$

The first box constraints are derived according to (11), by using the fact that $\eta_{ikl} \geq 0$. The solution is a set of variables α_{ikl} from which w_k, $k = 1,..,Q$ can be computed via (12). The bias b_k, $k = 1,..,Q$ are derived by using the Karush-Kuhn-Tucker conditions:

$$\alpha_{ikl} \left(\langle w_k - w_l, x_i \rangle + b_k - b_l - 1 + \xi_{ikl} \right) = 0 \tag{14}$$

$$(C - \alpha_{ikl})\xi_{ikl} = 0 \tag{15}$$

For indices (i,k,l) such that $\alpha_{ikl} \in (0, C)$, we have:

$$\langle w_k - w_l, x_i \rangle + b_k - b_l = 1$$

Since w_k and w_l are already known, this equation can be used to compute the differences between b_k and b_l. Note that if a primal-dual method is used to solve the dual 10, the variables b_k, $k = 1,..,Q$ are directly derived as the dual of the constraints $\sum_{i=1}^{m} \sum_{(j,l)\in(\mathbf{Y}_i,\bar{\mathbf{Y}}_i)} c_{ijl}\alpha_{ijl} = 0$.

The dual problem has an advantage compared to its primal counterpart. First, it contains only Q equality constraints and many box constraints. The latter are quite easy to handle in many optimization

methods such as the one we will describe in next section. The other equality constraints are more general but are not so many such that they can be handled also when the number of variables is large. The main problem here concerns the objective function: it is a quadratic one that cannot be solved directly except by storing its whole Hessian. In the case when the number of points is large as well as the number of classes, the number of variables may be too important to take a direct approach. For that reason, we propose to follow an approximation scheme from Franke and Wolfe [7].

A.2 Franke and Wolfe's method applied to the dual problem

The method we propose to apply is an iterative procedure that is designed to solve the following kind of problems:

Problem 11
$$\max_{\alpha} g(\alpha)$$
$$\textit{subject to :} \quad \langle a_k, \alpha \rangle = 0, \quad \textit{for } k = 1, .., Q$$
$$\alpha_j \in [0, C]$$

where the vectors a_i have the same dimensionality as the vector α. The procedure is as follows:

1. start with $\alpha^0 = (0, .., 0)$
2. Assume at iteration p, α^p is given. Solve the linear problem:

   Problem 12
   $$\max_{\alpha} \langle \nabla g(\alpha^p), \alpha \rangle$$
   $$\textit{subject to :} \quad \langle a_k, \alpha \rangle = 0, \quad \textit{for } k = 1, .., Q$$
   $$\alpha_j \in [0, C]$$

   Let α^* be the optimum.
3. Compute $\lambda \in [0, 1]$ such that $g(\alpha^p + \lambda(\alpha^* - \alpha^p))$ is minimum. Let λ^* be this value.
4. Set $\alpha^{p+1} = \alpha^p + \lambda^*(\alpha^* - \alpha^p)$.
5. End the procedure if $\lambda = 0$ or if $\alpha^{p+1} - \alpha^p$ has a norm lower than a fixed threshold.

The idea of this procedure is to transform a difficult quadratic problem into many simple linear problems. It is proved in [7] that such a procedure converges.

To apply it to our problem, we need to compute the gradient of the dual objective function. For that purpose, let us introduce new vectors:
$$v_k^p = \sum_{i=1}^{m} \beta_{ki}^p x_i$$
where β^p are computed from (13) in terms of the α_{ikl}^p. The latter are the current values of the parameter we optimize with the Franke and Wolfe's method. At the optimum, we will have $w_k = v_k$. At iteration p, the objective function can thus be expressed as:
$$\underbrace{-\frac{1}{2}\sum_{k=1}^{Q} \|v_k^p\|^2}_{=I} + \underbrace{\sum_{i=1}^{m} \sum_{(k,l) \in (\mathbf{Y}_i, \bar{\mathbf{Y}}_i)} \alpha_{ikl}}_{=J}$$

Franke and Wolfe Method applied on problem 10

1. Start with $\alpha = (0, .., 0) \in \mathbb{R}^{s_\alpha}$, where $s_\alpha = \sum_{i=1}^{m} |\mathbf{Y}_i| \left|\bar{\mathbf{Y}}_i\right|$.

2. Set: $c_{ijl}^k = \begin{cases} 0 & \text{if } j \neq k \text{ and } l \neq k \\ +1 & \text{if } j = k \\ -1 & \text{if } l = k \end{cases}$

3. For $k = 1, .., Q$, and $i = 1, .., m$, compute: $\beta_{ki} = \sum_{(j,l)\in(\mathbf{Y}_i, \bar{\mathbf{Y}}_i)} c_{ijl}^k \alpha_{ijl}$

4. For $k = 1, .., Q$, and $j = 1, .., m$, compute: $\langle w_k, x_j \rangle = \sum_{i=1}^{m} \beta_{ki} \langle x_i, x_j \rangle$

5. Set $g_{ikl} = (\langle w_k, x_i \rangle - \langle w_l, x_i \rangle) - 1$

6. Solve:$\min_{\alpha^{\text{new}}} \langle g, \alpha^{\text{new}} \rangle$ with the constraints:

$$\alpha_{ijl}^{\text{new}} \in \left[0, \frac{C}{C_i}\right]$$

and

$$\sum_{i=1}^{m} \sum_{(j,l)\in(\mathbf{Y}_i, \bar{\mathbf{Y}}_i)} c_{ijl}^k \alpha_{ijl} = 0, \quad \text{for } k = 1, .., Q.$$

7. Find $\lambda \in \mathbb{R}$ such that:
$W(\alpha + \lambda\alpha^{\text{new}})$ be maximum and $\alpha + \lambda\alpha^{\text{new}}$ satisfies the previous constraints.

8. Set $\alpha = \alpha + \lambda\alpha^{\text{new}}$.

9. Unless convergence, go back to **3.**

Figure 11: Algorithm for solving problem 10. It takes the learning set $S = \{(x_i, \mathbf{Y}_i)\}_{i=1,..,m}$ and the constant C as inputs, and it outputs the vector α. The computation of the b_k, $k = 1, .., Q$ is done via the Karush-Kuhn-Tucker conditions as explained previously.

The J part is quite direct to differentiate. The I part can be differentiated as:

$$\frac{\partial I}{\partial \alpha_{ikl}} = -\sum_{j=1}^{Q} \left\langle \nabla_{v_j^p} I, \nabla_{\alpha_{ikl}} v_j^p \right\rangle \tag{16}$$

$$= -\langle v_k^p, x_i \rangle + \langle v_l^p, x_i \rangle \tag{17}$$

The computation of the gradient of the objective function can thus be done directly as soon as the vectors v_k^p are given. They are expressed in terms of the α_{ikl}^p and only dot products between them and the x_i's are necessary here. That means that this procedure can work as well for kernels.

Denoting by $W(\alpha)$ the objective function of problem 10, the final algorithm can then be written as in figure 11. To have the non-linear version of this algorithm, just replace the dot products $\langle x_i, x_j \rangle$ by kernels $k(x_i, x_j)$.