

CS 294 – Practical Machine Learning  
Term Project  
Robust Classification for Data with Interval  
Uncertainty and Label Errors

Ufuk Topcu  
16848755  
utopcu@berkeley.edu

December 11, 2006

**Abstract**

A robust linear binary classification problem will be considered. Robustness will be for data with interval uncertainty, i.e., data points are unknown but their mean and bounds on their components are known. Convex optimization formulation for the problem is derived and the method is applied to a genomic micro-array data. An extension for this framework will be developed for data with uncertainties due to label errors. For this problem, the convex optimization formulation is derived. The implementation of this method is postponed due to lack of a useful data set.

## 1 Introduction

A linear binary classification problem will be considered. In order to measure the quality of the candidate classifier  $(w, b)$ , an objective function, namely loss function,

$$L(w, b) = \sum_{i=1}^N \phi(y_i(w^T x_i + b))$$

is minimized. The purpose is to minimize the misclassification error. An ideal loss function is a step function, zero for the correctly classified data and one for misclassified data. However, for numerical optimization purposes such a function leads to hard problems. Usually convex upper bounds are used. These upper bounds look like in Fig. (1).

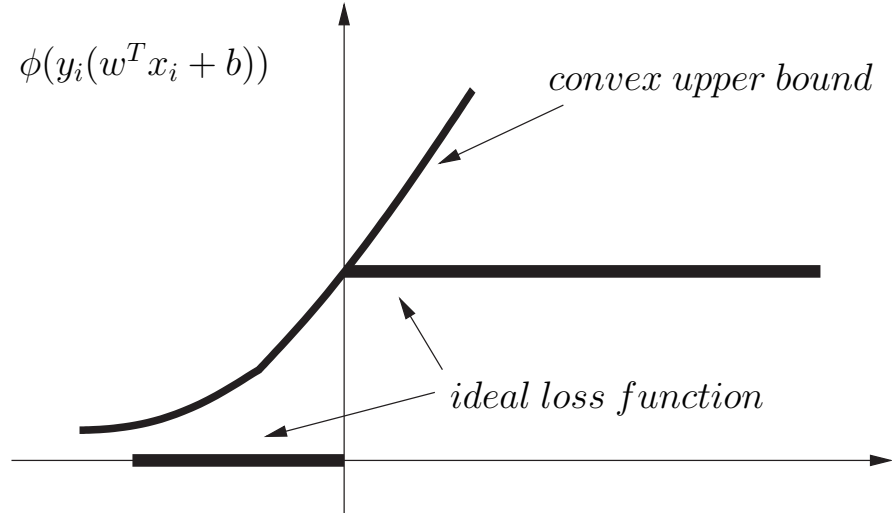


Figure 1: Ideal loss function and convex upper bounds.

Two types of loss function, logistic regression loss function and support vector machine loss function, will be used [1]. These loss functions are also used in [4].

1. *Logistic regression loss function:*

$$L_{LR}(w, b, X, y) := \sum_{i=1}^N \log \left( 1 + e^{-y_i(w^T x_i + b)} \right), \quad (1)$$

2. *The soft-margin support vector machine loss function:*

$$L_{SVM}(w, b, X, y) := \sum_{i=1}^N (1 - y_i(w^T x_i + b))_+ \quad (2)$$

where the function  $(\cdot)_+$  is defined as  $(\cdot)_+ : \mathbb{R} \rightarrow [0, \infty)$  is defined as  $a_+ = \max(a, 0)$  for  $a \in \mathbb{R}$ .

Above,  $x_i$  denotes  $i$ -th data point and  $y_i$  denotes corresponding label,  $y_i = +1$  or  $-1$ ,  $X$  is the matrix whose columns are the data points  $x_i$  and  $y$  is the column vector of labels. The linear classification function is  $w^T x + b$ , where  $w$  and  $b$  are the parameters. For the details of notation and formal problem description see section 2.  $L_{SVM}$  provides an upper bound on the number of misclassification errors. Indeed, for a misclassified data point  $(x_i, y_i)$ ,  $(1 - y_i(w^T x_i + b))_+ > 1$ . In [4], it was stated that  $L_{LR}$  also provides an upper bound to the number of misclassification errors.

Robustness to the uncertainties in the datas  $x_i$  and to the uncertainties in the label vector will be considered. In the former it is assumed that  $x_i$  is not known but only the mean value of each component and a confidence level for each component is known. Next section provides a formal description of the problem of interest. An extension to this problem will be considered, in which data points will assumed to be known and the uncertainty will be due the possible errors in the label vector, more specifically, it will be assumed that  $k$  of  $N$  labels are incorrect. This problem is solved in [4] for the SVM loss function and formulated as linear programming problem. One of the contributions of this project is to extend the derivation to logistic regression loss function.

In all these cases, problems are established as robust optimization problems. Roughly, robust optimization involves optimization of a cost function under "bounded" uncertainties in the problem data, eg. those in  $x_i$  and  $y_i$ . Then, the goal is to optimize the cost function not only for a specific instance of the problem but for all possible instances, eg. for all possible values of  $x_i$  and  $y_i$ . To this end, the worst case cost function is optimized. For more detailed information on robust optimization, see [10, 12, 11].

## 2 Setup

Setup and notation will be parallel to those in [4] as explained in this subsection. Let  $X \in \mathbb{R}^{n \times N}$  and  $y \in \mathbb{R}^N$  denote the matrix of data points and corresponding vector of labels. The vector of labels consists of  $+1$  and  $-1$ , for short  $y \in \{-1, 1\}^N$ . Let  $\rho > 0$  be a real number and  $\Sigma \in \mathbb{R}_+^{n \times N}$ , where  $\mathbb{R}_+$  is the restriction of the real line to the positive reals. The interval matrix model introduced in [4] is based on the following set description of the uncertainty of the data points

$$\mathcal{X}(\rho) := \{Z \in \mathbb{R}^{n \times N} : X - \rho\Sigma \leq Z \leq X + \rho\Sigma\}. \quad (3)$$

Here and later usual inequalities, i.e.,  $\leq, \geq, <, >$ , denote componentwise inequalities. In [4],  $X$  was referred to as nominal data matrix,  $\Sigma$  is called standard error matrix and  $\rho$  is a scaling associated with the standardized error in  $\Sigma$ . In words, data points, entries of  $Z$  are known up to their mean, corresponding entries of  $X$ , and their bounds, corresponding entries of  $\rho\Sigma$ , see Fig. (2). For later reference, denote  $i$ -th column of  $X$ ,  $Z$  and  $\Sigma$  by  $x_i, z_i$  and  $\sigma_i$  respectively and define

$$\sigma := \sum_{i=1}^N \sigma_i.$$

Here we consider linear classifiers in the form

$$\text{classification boundary : } w^T x + b = 0.$$

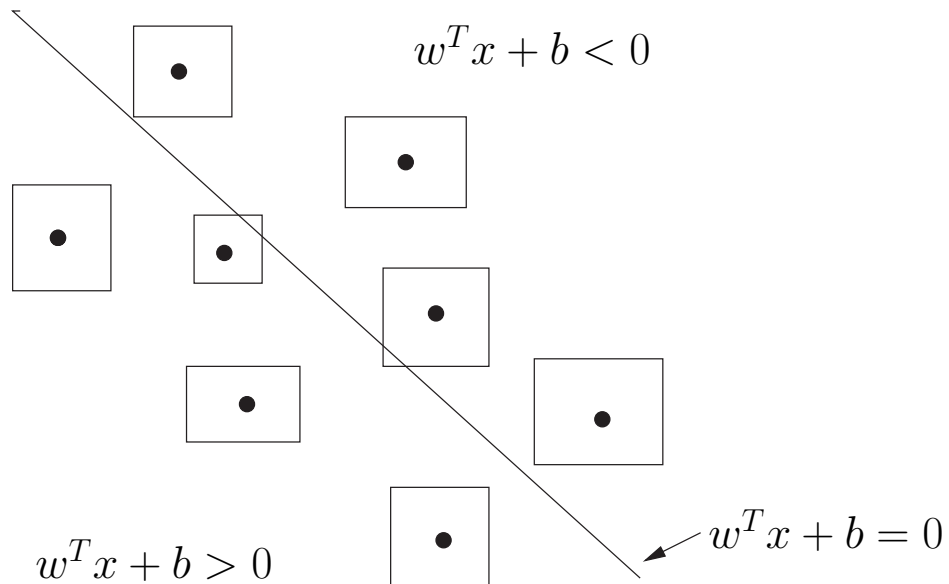


Figure 2: Classification of data with interval uncertainty.

Using this setup the robust classification problem can be stated as

$$\min_{w,b} \max_{Z \in \mathcal{X}(\rho)} L_{SVM \text{ or } LR}(w, b, Z, y).$$

Note that the purpose is not to minimize the loss function for a specific data matrix but for all possible data matrices described by (3).

For the setup of the problem for the classification robust to the uncertainties due to label errors, see the corresponding section.

**Note:** Although, whenever appropriate, it is stated that most of the results derived in this project are either from [4] or parallel to those in there, I would like to make a short summary about the content. Results derived in section 3 are completely from [4] with a exception that the derivation is in a slightly and trivially more general setting (one of the parameters is let to be any positive number instead of 1). By "from [4]", I mean results are stated there. All the results are re-derived and details are filled whenever necessary. Results in section 4.1 are re-derivation of those in [4]. Results in section 4.2 are totally new although the idea is a generalization of that in [4] for the SVM setting.

### 3 Classification robust to interval uncertainties in the data points

#### 3.1 Robust logistic regression

In this section, the following problem is considered

$$\min_{w,b} \max_{Z \in \mathcal{X}(\kappa\rho)} \sum_{i=1}^N \log \left( 1 + e^{-y_i(w^T z_i + b)} \right) + (1 - \kappa)\sigma^T |w|, \quad (4)$$

where  $0 \leq \kappa \leq 1$  and for  $w \in \mathbb{R}^n$   $|w|_i = |w_i|$ . The derivation in this section will be more general than that in [4]. There, the derivation was given for  $\rho = 1$  (the final result was stated for the general case). Here it is developed for positive  $\rho$ . Introduction of the extra term leads to a more general loss function and in fact the worst-case value of this more general loss function provides an upper bound for that straightforwardly obtainable from the logistic regression loss function, i.e.,

$$\max_{Z \in \mathcal{X}(\rho)} \sum_{i=1}^N \log \left( 1 + e^{-y_i(w^T z_i + b)} \right) \leq \max_{Z \in \mathcal{X}(\kappa\rho)} \sum_{i=1}^N \log \left( 1 + e^{-y_i(w^T z_i + b)} \right) + (1 - \kappa)\sigma^T |w|$$

since  $\kappa = 1$  recovers the expression of the left hand side. Actually, allowing non-unity  $\kappa$  provides a relaxation for the regular logistic regression problem.

The problem in (4) is an infinite dimensional optimization problem. However it can reduced to finite dimensional one by noting that

$$\log \left( 1 + e^{-y_i(w^T z_i + b)} \right) \leq \log \left( 1 + e^{-y_i(w^T x_i + b) + \kappa\rho\sigma_i^T |w|} \right).$$

This leads to the problem

$$\min_{w,b} \sum_{i=1}^N \log \left( 1 + e^{-y_i(w^T x_i + b) + \kappa \rho \sigma_i^T |w|} \right) + (1 - \kappa) \sigma^T |w|. \quad (5)$$

By introducing positive vectors  $w_p$  and  $w_n$  such that  $w = w_p - w_n$ , this problem can be written as

$$\min_{w_p \geq 0, w_n \geq 0, b} \sum_{i=1}^N \log \left( 1 + e^{-y_i((w_p - w_n)^T x_i + b) + \kappa \rho \sigma_i^T (w_p + w_n)} \right) + (1 - \kappa) \sigma^T (w_p + w_n). \quad (6)$$

Note that this last problem is a convex optimization problem. It is amenable to interior point algorithms [2]. Next, the dual problem for the problem in (6) will be derived. This formulation will be amenable to the readily available convex optimization solvers [5]. For this some more notation is needed. Let

$$\xi := \begin{pmatrix} w_p \\ w_n \\ b \end{pmatrix}, \quad v := (\kappa - 1) \begin{pmatrix} \sigma \\ \sigma \\ 0 \end{pmatrix}, \quad M := \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \end{pmatrix}$$

$$a_i := \begin{pmatrix} \rho \kappa \sigma_i - y_i x_i \\ \rho \kappa \sigma + y_i x_i \\ -y_i \end{pmatrix} \text{ for } i = 1, \dots, N, \quad A := (a_1 \cdots a_N).$$

Using this notation, the problem in (6) can be written as

$$\max_{\xi, \eta} v^T \xi - \sum_{i=1}^N \log(1 + e^{\eta_i}) \text{ subject to } \eta = A^T \xi, \quad M \xi = 0. \quad (7)$$

Then, the Lagrangian for this constrained optimization problem is

$$L(\xi, \eta, \lambda, \nu) := v^T \xi - \sum_{i=1}^N \log(1 + e^{\eta_i}) + \lambda^T (\eta - A^T \xi) + \nu^T M \xi. \quad (8)$$

Invoking the first order optimality conditions we obtain

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow v - A \lambda + M^T \nu = 0 \Rightarrow v = A \lambda - M^T \nu$$

and

$$\frac{\partial L}{\partial \eta} = 0 \Rightarrow -\frac{e^{\eta_i}}{1 + e^{\eta_i}} + \lambda_i = 0 \Rightarrow \lambda_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}.$$

More conveniently these can be written as

$$1 + e^{\eta_i} = \frac{1}{1 - \lambda_i} \quad \text{and} \quad \eta_i = \log\left(\frac{\lambda_i}{1 - \lambda_i}\right).$$

Noting that the dual variable  $\nu$  drops from the optimal Lagrangian due to cancellations, the dual function can be written as

$$\begin{aligned} g(\lambda) &:= -\sum_{i=1}^N \log\left(\frac{1}{1-\lambda_i}\right) + \sum_{i=1}^N \log\left(\frac{\lambda_i}{1-\lambda_i}\right) \\ &= \lambda^T \log(\lambda) + (\mathbf{1} - \lambda)^T \log(\mathbf{1} - \lambda), \end{aligned} \tag{9}$$

where  $\mathbf{1}$  is the vector of ones of appropriate size, for a vector  $a \in \mathbb{R}_+^d$ ,  $[\log(a)]_i = \log(a_i)$ . Then, the dual optimization problem becomes

$$\begin{aligned} \min_{\lambda \geq 0, \nu \geq 0} \quad & \lambda^T \log \lambda + (\mathbf{1} - \lambda)^T \log(\mathbf{1} - \lambda) \\ \text{s.t.} \quad & \lambda \leq \mathbf{1} \\ & A\lambda = M^T \nu + v. \end{aligned} \tag{10}$$

In [4] the following trick is used to eliminate the dual variable  $\nu$  from the preceding optimization problem: Partition  $\nu$  as  $\nu = (\nu_p, \nu_n)$ . Then, the condition  $A\lambda = M^T \nu + v$  implies

$$\sum_{i=1}^N (\rho\kappa\sigma_i - y_i x_i) \lambda_i = -(1 - \kappa)\sigma + \nu_p, \tag{11}$$

$$\sum_{i=1}^N (\rho\kappa\sigma_i + y_i x_i) \lambda_i = -(1 - \kappa)\sigma + \nu_n. \tag{12}$$

Then, it is easy to show that there exist non-negative vectors  $\nu_p$  and  $\nu_n$  if and only if the following condition holds

$$|XY\lambda| \leq \rho\kappa\Sigma\lambda + (1 - \kappa)\sigma,$$

where  $Y = \mathbf{diag}(y)$ . Finally, the problem in (10) leads to

$$\begin{aligned} \min_{\lambda \geq 0} \quad & \lambda^T \log \lambda + (\mathbf{1} - \lambda)^T \log(\mathbf{1} - \lambda) \\ \text{s.t.} \quad & |XY\lambda| \leq \rho\kappa\Sigma\lambda + (1 - \kappa)\sigma \\ & \lambda \leq \mathbf{1} \\ & y^T \lambda = 0, \end{aligned} \tag{13}$$

where the last constraint directly follows from  $A\lambda = M^T\nu + v$ . This last problem is an entropy problem and it is convex. Moreover, it can directly be solved by using Mosek [5].

Note that the variable  $\xi$  is the dual variable corresponding to the the dual constraint  $A\lambda = M^T\nu + v$ . Therefore  $w_p$  is dual to (11),  $w_n$  is dual to (12), and  $b$  is dual to the dual constraint  $y^T\lambda = 0$ . One property of Mosek is it solves the dual problem (primal problem in this case because we are trying to solve what was called the the dual problem) along with the primal problem (in this case called the dual problem) and outputs the optimal dual variables as well [5]. Hence, Mosek's output also contains the optimal values of the variables of interest, namely  $w$  and  $b$ .

### 3.2 Robust SVM

Robust SVM problem is the minimization of the worst-case SVM loss function, namely,

$$\min_{w,b} \max_{Z \in \mathcal{X}(\rho)} \sum_{i=1}^N (1 - y_i(w^T z_i + b))_+ . \quad (14)$$

Noting that the entries of  $\sigma_i$  are non-negative and the function  $(\cdot)_+$  is monotonic this problem can be written as

$$\min_{w,b} \sum_{i=1}^N (1 - y_i(w^T x_i + b) + \rho \sigma_i^T |w|)_+ . \quad (15)$$

Introducing slack variables  $e_i$ ,  $i = 1, \dots, N$ , the following problem is obtained from the previous one

$$\begin{aligned} \min_{w,b,e} \quad & e^T \mathbf{1} \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - e_i + \rho \sigma_i^T |w| \text{ for } i = 1, \dots, N \\ & e \geq 0 \end{aligned} \quad (16)$$

In order to take care of the terms involving  $|w|$ , introduce two non-negative vectors  $w_p$  and  $w_n$  that decompose  $w$  as  $w = w_p - w_n$ . Using this, the problem in (16) leads to

$$\begin{aligned} \min_{w_p, w_n, b, e} \quad & e^T \mathbf{1} \\ \text{s.t.} \quad & y_i((w_p - w_n)^T x_i + b) \geq 1 - e_i + \rho \sigma_i^T (w_p + w_n) \text{ for } i = 1, \dots, N \\ & e \geq 0, w_p \geq 0, w_n \geq 0. \end{aligned} \quad (17)$$

The problem in (17) is a linear programming problem. As in the logistic regression case, a more general problem is also considered in [4]. This is based on the following observation

$$\begin{aligned} \max_{Z \in \mathcal{X}(\rho)} \quad & \sum_{i=1}^N (1 - y_i(w^T z_i + b))_+ \\ & = \sum_{i=1}^N (1 - y_i(w^T x_i + b) + \rho \sigma_i^T |w|)_+ \\ & \leq \sum_{i=1}^N (1 - y_i(w^T x_i + b) + \kappa \rho \sigma_i^T |w|)_+ + (1 - \kappa) \rho \sigma^T |w|, \end{aligned} \quad (18)$$

where  $0 \leq \kappa \leq 1$ . The equality follows from the monotonicity of the function  $(\cdot)_+$  and the inequality follows from the fact that  $(1 - \kappa) \rho \sigma^T |w| \geq 0$  and hence  $((1 - \kappa) \rho \sigma^T |w|)_+ = (1 - \kappa) \rho \sigma^T |w|$ . To see the latter, let  $\lambda \in [0, 1]$ . Then,  $(\lambda a + (1 - \lambda) b)_+ = \max(\lambda a + (1 - \lambda) b, 0) \leq \lambda \max(a, 0) + (1 - \lambda) \max(b) = \lambda a_+ + (1 - \lambda) b_+$ . Using the upper bound on the worst-case loss function an alternative more general formulation robust SVM problem is obtained as

$$\begin{aligned} \min_{w, b, e} \quad & e^T \mathbf{1} + \rho(1 - \kappa) |w| \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - e_i + \rho \kappa \sigma_i^T |w|, \text{ for } i = 1, \dots, N \\ & e \geq 0 \end{aligned} \quad (19)$$

Again, introduce two non-negative vectors  $w_p$  and  $w_n$  that decompose  $w$  as  $w = w_p - w_n$  to obtain

$$\begin{aligned} \min_{w_p, w_n, b, e} \quad & e^T \mathbf{1} + (1 - \kappa) \sigma^T (w_p + w_n) \\ \text{s.t.} \quad & y_i((w_p - w_n)^T x_i + b) \geq 1 - e_i + \rho \kappa \sigma_i^T (w_p + w_n) \text{ for } i = 1, \dots, N \\ & e \geq 0, w_p \geq 0, w_n \geq 0. \end{aligned} \quad (20)$$

The last optimization problem is a linear programming problem and there are many efficient solvers even for large scale problems. The implementation of this problem in [8] using Mosek will be used in this project.

## 4 Robust Classification with Label Errors

In this section the following optimization problem will be considered

$$\min_{w, b} \max_{z \in \mathcal{Y}(y, k)} L_{SVM \text{ or } LR}(w, b, X, z), \quad (21)$$

where

$$\mathcal{Y}(y, k) := \{z : z_i = (1 - 2\delta_i) y_i, i = 1, \dots, N, \delta \in [0, 1]^N, \mathbf{1}^T \delta \leq k\}.$$

In this model, the uncertainty is in the label vector and we assume that at most  $k$  of  $N$  labels are incorrect. The aim is to obtain a classifier robust to the uncertainties described in (21). This problem was studied in [4] for the SVM loss function and the problem was reduced to a linear program. Here, this linear program will be re-derived. A contribution of this project is deriving a convex formulation for the logistic regression problem. The formulation of these optimization problems follows.

#### 4.1 Label errors with SVM loss function

The formulation in this subsection is from [4]. First the following subproblem is considered.

$$\phi := \max_{z \in \mathcal{Y}(y,k)} \sum_{i=1}^N (1 - \alpha_i z_i)_+,$$

where  $\alpha_i = x_i^T w + b$ . This problem can be written as

$$\begin{aligned} \phi &= \max_{0 \leq t \leq 1} \max_{z \in \mathcal{Y}(y,k)} \sum_{i=1}^N t_i (1 - \alpha_i z_i) \\ &= \max_{0 \leq t \leq 1} \max_{\delta \in [0,1]^N, \mathbf{1}^T \delta \leq k} \sum_{i=1}^N (t_i (1 - \alpha_i y_i) + 2\delta_i t_i y_i \alpha_i) \\ &= \max_{0 \leq t \leq 1} (\mathbf{1}^T (t - \eta) + \max_{\delta \in [0,1]^N, \mathbf{1}^T \delta \leq k} \delta^T \eta), \end{aligned}$$

where  $\eta$  is such that  $\eta_i = t_i \alpha_i y_i$ . Now, focus on the following subproblem

$$\max_{\delta \in [0,1]^N, \mathbf{1}^T \delta \leq k} \delta^T \eta.$$

This is an LP and by LP duality it can be equivalently written as

$$\min_{\lambda_1, \lambda_2, \mu \geq 0} \mathbf{1}^T \lambda + \mu k : \eta_i + \lambda_{1,i} - \lambda_{2,i} + \mu = 0, \text{ for } i = 1, \dots, N,$$

where  $\lambda_1$ ,  $\lambda_2$  and  $\mu$  are dual variables of appropriate size. Note that, there exist  $\lambda_{1,i}, \lambda_{2,i}, \mu \geq 0$  such that the last constraint holds if and only if there exist  $\lambda_{1,i}, \mu \geq 0$  such that  $-\eta_i + \lambda_{1,i} + \mu \geq 0$ . Combining these results, the subproblem can be written as

$$\min_{\mu \geq 0} \mu k + \mathbf{1}^T (\eta - \mu \mathbf{1})_+.$$

Now the optimization problem for the worst case loss function can be written as

$$\phi = \max_{0 \leq t \leq 1} \min_{\mu \geq 0} \sum_{i=1}^N (t_i (1 - \alpha_i y_i) + \mu k + (\alpha_i y_i t_i - \mu)_+).$$

By weak duality [3], the following holds

$$\begin{aligned}\phi &\leq \min_{\mu \geq 0} \max_{0 \leq t \leq 1} \sum_{i=1}^N (t_i(1 - \alpha_i y_i) + \mu k + (\alpha_i y_i t_i - \mu)_+) \\ &= \min_{\mu \geq 0} \mu k N + \sum_{i=1}^N ((1 - \alpha_i y_i) + (\alpha_i y_i - \mu)_+).\end{aligned}$$

Consequently, the robust classification problem can be written as

$$\min_{\mu \geq 0, w, b} \mu k N + \sum_{i=1}^N ((1 - y_i(w^T x_i + b)) + (y_i(w^T x_i + b) - \mu)_+),$$

which is amenable to linear programming solvers. To see this, replace  $y_i(w^T x_i + b) - \mu$  by  $\xi_i$  and  $1 - y_i(w^T x_i + b) + \xi_i$  by  $\zeta_i$  and enforce the extra constraints  $y_i(w^T x_i + b) - \mu \leq \xi_i$ ,  $0 \leq \xi_i$ ,  $1 - y_i(w^T x_i + b) + \xi_i \leq \zeta_i$ , and  $0 \leq \zeta_i$ .

## 4.2 Label errors with logistic regression loss function

The key problem in this derivation is the following problem

$$\max_{\delta \in [0, 1]^N, \mathbf{1}^T \delta \leq k} \sum_{i=1}^N \log(1 + e^{-z_i \alpha_i}),$$

where  $\alpha_i = w^T x_i + b$  and  $z \in \mathcal{Y}(y, k)$  with  $\mathcal{Y}(y, k)$  as defined in the previous section. This problem can be written as

$$\max_{\delta \in [0, 1]^N, \mathbf{1}^T \delta \leq k} \sum_{i=1}^N \log(1 + e^{-y_i \alpha_i + 2\delta_i y_i \alpha_i}).$$

Dualizing with respect to the second constraint, i.e., with respect to  $\mathbf{1}^T \delta \leq k$ , the following series of equivalent formulations are obtained.

$$\min_{\mu \geq 0} \max_{\delta \in [0, 1]^N} \sum_{i=1}^N \log(1 + e^{-y_i \alpha_i + 2\delta_i y_i \alpha_i}) + \mu(k - \mathbf{1}^T \delta)$$

$$\min_{\mu \geq 0} \mu k + \sum_{i=1}^N \max_{\delta_i \in [0, 1]} \log(1 + e^{-y_i \alpha_i + 2\delta_i y_i \alpha_i}) - \mu \delta_i$$

$$\min_{\mu \geq 0} \mu k + \sum_{i=1}^N \max\{\log(1 + e^{-y_i \alpha_i}), \log(1 + e^{y_i \alpha_i}) - \mu\}$$

Finally, the following convex optimization problem is obtained.

$$\begin{aligned}\min_{\mu \geq 0, \lambda, \alpha} & \mu k + \mathbf{1}^T \lambda \\ \text{s.t.} & \lambda_i \geq \log(1 + e^{-y_i \alpha_i}) \\ & \lambda_i \geq \log(1 + e^{y_i \alpha_i}) - \mu.\end{aligned} \tag{22}$$

Using this formulation of the worst case loss function, the robust classification problem with logistic regression loss function and with at most  $k \leq N$  label errors can be formulated as

$$\begin{aligned} \min_{\mu \geq 0, \lambda, w, b} \quad & \mu k + \mathbf{1}^T \lambda \\ \text{s.t.} \quad & \lambda_i \geq \log(1 + e^{-y_i(w^T x_i + b)}) \\ & \lambda_i \geq \log(1 + e^{y_i(w^T x_i + b)}) - \mu. \end{aligned} \quad (23)$$

The problem in (23) is a convex optimization problem since its cost function is linear in the variables and the constraints are convex. Indeed, the function  $\xi \mapsto \log(1 + e^\xi)$  is convex and the terms in the constraints are the composition of linear functions with a convex function; hence, they are convex. This problem is amenable to interior-point methods [2]. However, by some simple manipulation, it can be transferred to a form which can be readily tackled using Mosek. To this end, since the exponential function is monotonic the feasible set of the last problem is the set of  $\mu \geq 0$ ,  $\lambda \geq 0$  (this was implicit in the previous formulation; hence, nothing new),  $w$  and  $b$  satisfying

$$1 \geq e^{-\lambda_i} + e^{-y_i(w^T x_i + b) - \lambda_i}$$

and

$$1 \geq e^{-\lambda_i - \mu} + e^{-y_i(w^T x_i + b) - \lambda_i - \mu}.$$

Also, minimizing  $e^{\mu k + \mathbf{1}^T \lambda}$  instead of  $\mu k + \mathbf{1}^T \lambda$  does not change the optimal value due to the monotonicity of the exponential function. Consequently, the following optimization problem is obtained

$$\begin{aligned} \min_{\mu \geq 0, \lambda \geq 0, w, b} \quad & e^{\mu k + \mathbf{1}^T \lambda} \\ \text{s.t.} \quad & 1 \geq e^{-\lambda_i} + e^{-y_i(w^T x_i + b) - \lambda_i} \\ & 1 \geq e^{-\lambda_i - \mu} + e^{-y_i(w^T x_i + b) - \lambda_i - \mu}. \end{aligned} \quad (24)$$

This last problem is called exponential optimization problem [7] and there are readily available routines in Mosek [5] to solve this problem.

## 5 Implementation, data set and experimentation

For the implementation robust classification with interval uncertainty problem a toolbox written by L. El Ghaoui [8] is used. This toolbox is based on Matlab and uses Mosek [5] as the convex optimization solver. For the implementation

of the robust SVM problem the formulation in (20) with  $\kappa = 1$  is used. The implementation of the robust logistic regression is based on (13) with  $\kappa = 1$ . One of the main reasons to choose Mosek [5] as the convex optimization solver is that it supports the entropy problems as in 13. The implementation for the robust SVM problem can be efficiently performed with any state-of-the-art linear programming solver as well.

Data set used in [4] and also in this project is a genomic micro-array data obtained by Iconix Pharmaceuticals [6]. For a similar application, see [9]. The data is composed of a matrix  $X$  containing the log-ratios of responses of  $n = 8565$  genes to different drugs and a corresponding standard error matrix obtained with 3 replicates for each experiment. There is  $N = 193$  experiments in total. In this project, a label vector to classify a specific drug class class (namely statin class) from all other drugs (classes). Statin class contains 31 points. The underlying biological problem is to be able to separate this class (station class) from all other experiments with as few genes as possible with a good "enough" predictive performance [4].

## 5.1 Results for robust logistic regression loss function

In this implementation a 3-to-2 training-test data ratio is used with 3 random partitions for cross validation. Results are shown in Tables (5.1)-(5.6). Results are averages over the partitions. For the regular logistic regression results are obtained in [8] by implementing a two-class logistic regression model with a  $l_1$ -norm norm in the objective added for sparsity purposes. As seen in the tables, as the uncertainty level increases ( $\rho$  increases), the performance of the robust logistic regression implementation improves, whereas that for the regular logistic regression classifier gets worse. Regular logistic regression tends to predict everything to belong to the negative family as  $\rho$  increases. The performance of robust logistic regression classifier degrades later than that of the regular logistic regression classifier. Of course this improvement is not for free: computation times for robust implementation is higher than the regular implementation. However, this gap does not render the robust implementation impractical. The computation times for the robust implementation are still reasonable (see Table (5.7)). Furthermore, robust implementation results in larger classifiers, i.e., more of the genes are involved.

Table 1: Average values over test data for  $\rho = 0.05$

robust logistic regression			logistic regression		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	3.33	9.67	true positive	8.67	4.33
true negative	0	65	true negative	0.33	64.67

Table 2: Average values over test data for  $\rho = 0.1$

robust logistic regression			logistic regression		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	6.33	6.67	true positive	7.33	5.67
true negative	1	64	true negative	0	65

Table 3: Average values over test data for  $\rho = 0.2$

robust logistic regression			logistic regression		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	8.33	4.67	true positive	6	7
true negative	0	65	true negative	0	65

Table 4: Average values over test data for  $\rho = 0.4$

robust logistic regression			logistic regression		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	10.67	2.33	true positive	3.67	9.33
true negative	0	65	true negative	0	65

Table 5: Average values over test data for  $\rho = 0.7$

robust logistic regression			logistic regression		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	10	3	true positive	0	13
true negative	0.33	64.67	true negative	0	65

Table 6: Average values over test data for  $\rho = 1$

robust logistic regression			logistic regression		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	7.67	5.33	true positive	0	13
true negative	0	65	true negative	0	65

Table 7: Computation times (seconds) and length of the classifier.

$\rho$	robust logistic regression		logistic regression	
	computation time	classifier length	computation time	classifier length
0.05	75	80	42	23
0.1	82	147	43	18
0.2	139	111	38	1
0.4	139	110	38	1
0.7	93	26	25	1
1	72	12	48	1

## 5.2 Results for robust logistic regression loss function

Implementation is done as in the logistic regression case. Results are shown in Tables (5.8)-(5.14). In the table "SVM" stands for the linear, soft-margin support vector machine classifier and "ROB-SVM" stands for the formulation in section 3. In the logistic regression case, the robust classifier was outperforming the regular one as the uncertainty level increases. In SVM case, although the performance of the robust classifier improves as  $\rho$  increases up to some level, it almost never outperforms the regular one. On the negative points robust one does better job. Computation times for the robust classifier are smaller than those for the regular one.

**Summary:** The performance of the robust logistic regression classifier performs better than the regular sparse classifier as the level of uncertainty increases at the expense of longer (but tractable) computationally times. However, the performance of the robust classifier based on SVM loss function performs slightly worse than the regular one. One reason might be that the dimension of the data space is very high (more than 8000 dimensions) compared to the number of experiment (193). One can claim that simply there is not enough data to exploit the features of the robust classifier. On the other hand, one useful extension might be applying some kind of dimensionality reduction before classification. However, this might not be as straightforward as in the common case where the data points are known. At this point, no dimensionality reduction technique exploiting the structure of the data (uncertain) is known to me.

Table 8: Average values over test data for  $\rho = 0.2$ 

ROB-SVM			SVM		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	3	10	true positive	10.33	2.67
true negative	0	65	true negative	0.77	64.33

Table 9: Average values over test data for  $\rho = 0.35$ 

ROB-SVM			SVM		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	10	3	true positive	11.33	1.67
true negative	0	65	true negative	1.33	63.67

Table 10: Average values over test data for  $\rho = 0.5$ 

ROB-SVM			SVM		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	10.33	2.67	true positive	11	2
true negative	0.33	64.67	true negative	1.33	63.67

Table 11: Average values over test data for  $\rho = 0.75$ 

ROB-SVM			SVM		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	11	2	true positive	10	3
true negative	1.33	63.67	true negative	1	64

Table 12: Average values over test data for  $\rho = 1$ 

ROB-SVM			SVM		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	9.67	3.33	true positive	10	3
true negative	1	64	true negative	2	63

Table 13: Average values over test data for  $\rho = 1.5$ 

ROB-SVM			SVM		
	pre. positive	pre. negative		pre. positive	pre. negative
true positive	7.67	5.33	true positive	9.67	3.33
true negative	0.67	64.33	true negative	0.67	64.33

Table 14: Computation times (seconds) and length of the classifier.

$\rho$	ROB-SVM		SVM	
	computation time	classifier length	computation time	classifier length
0.2	75	188	110	44
0.35	79	155	110	31
0.5	72	71	106	27
0.75	116	23	98	16
1	91	13	74	12
1.5	114	5	25	9

## 6 Future work

The following list of extensions seem to be useful for this methodology:

1. Combine two type of uncertainties, namely interval uncertainty for the data points and uncertainties due to errors in the label vector. In [4], it was claimed that this can be done. However, this needs to be worked out.
2. The part on the label errors should be studied more in order to understand most effective formulation for computational purposes and be implemented.
3. In [4], the sparsity properties of the resulting classifier was discussed. A complete understanding of this issue deserves to be developed.
4. Effects of non-unity  $\kappa$  on the implementation and performance are to be studied.

**Acknowledgement:** Prof. Laurent El Ghaoui from EECS department has been very helpful by discussions on robust optimization and robust classification and by providing the his toolbox for implementation part of this project.

## References

- [1] T. Hastie, R. Tibshirani and J. Friedman *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer: New York, 2001.

- [2] Y. Nesterov and A. Nemirovskii *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.
- [3] S. Boyd and L. Vandenberghe *Convex Optimization*. Cambridge University Press, 2004.
- [4] L. El Ghaoui, G.R.G. Lanckriet and G. Natsoulis Robust Classification with Interval Data *Technical report, Report No. UCB/CSD-03-1279* 2003.
- [5] <http://www.mosek.com>.
- [6] <http://www.iconixpharm.com>.
- [7] <http://www.mosek.com/products/4.0/tools/doc/html/tools/tools.html>.
- [8] L. El Ghaoui *A Matlab toolbox for the implementation of robust classification problems with interval uncertainty*. Berkeley, 2003.
- [9] G. Natsoulis, L. El Ghaoui, G.R.G. Lanckriet, A.M. Tolley, F. Leroy, S. Dunlea, B.P. Eynon, C.I. Pearson, S. Tugendreich, and K. Jarnagin *Classification of a large microarray data set: Algorithm comparison and analysis of drug signatures*. Genome Research, 15: 724–736, 2005.
- [10] L. El Ghaoui and H. Lebret *Robust solutions to least-squares problems with uncertain data*. SIAM Journal of Matrix Analysis and Applications, 18(4): 1035–1064, 1997.
- [11] A. Ben-Tal and A. Nemirovski *Robust Optimization - Methodology and Applications*. Mathematical Programming Series B, 92: 453–480, 2002.
- [12] H. Wolkowicz, R. Saigal and L. Vandenberghe *handbook on Semidefinite Programming*. Kluwer Academic Publishers, 2000.