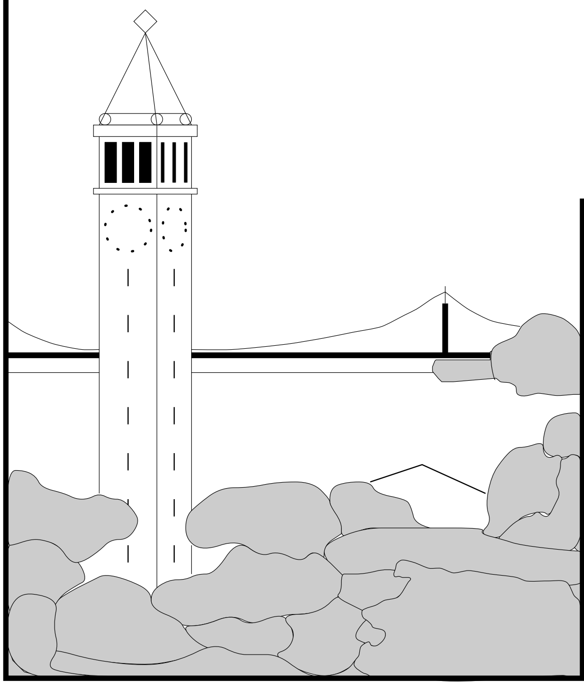
Robust Classification with Interval Data

Laurent El Ghaoui
elghaoui@eecs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Gert R.G. Lanckriet
gert@cs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Georges Natsoulis
gnatsoulis@iconixpharm.com
Iconix Pharmaceuticals, Inc.
325 East Middlefield Road, Mountain View, CA 94043



Report No. UCB/CSD-03-1279

October, 2003

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Robust Classification with Interval Data

Laurent El Ghaoui
elghaoui@eecs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Gert R.G. Lanckriet
gert@cs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Georges Natsoulis
gnatsoulis@iconixpharm.com
Iconix Pharmaceuticals, Inc.
325 East Middlefield Road, Mountain View, CA 94043

October, 2003

Abstract

We consider a binary, linear classification problem in which the data points are assumed to be unknown, but bounded within given hyper-rectangles, i.e., the covariates are bounded within intervals explicitly given for each data point separately. We address the problem of designing a *robust* classifier in this setting by minimizing the worst-case value of a given loss function, over all possible choices of the data in these multi-dimensional intervals. We examine in detail the application of this methodology to three specific loss functions, arising in support vector machines, in logistic regression and in minimax probability machines. We show that in each case, the resulting problem is amenable to efficient interior-point algorithms for convex optimization. The methods tend to produce sparse classifiers, i.e., they induce many zero coefficients in the resulting weight vectors, and we provide some theoretical grounds for this property. After presenting possible extensions of this framework to handle label errors and other uncertainty models, we discuss in some detail our implementation, which exploits the potential sparsity or a more general property referred to as regularity, of the input matrices.

1 Introduction

In several practical classification problems, data points are only provided approximately, i.e., often their covariates are only specified up to given intervals of confidence. For example, when collecting genomic micro-array data, experiments are usually noisy and often a number of replicates of the same experiment are available. This enables us to represent every data point by the smallest hyper-rectangle that encloses all corresponding replicates. Mathematically these uncertainty regions can be specified by a *nominal* data matrix and a second matrix of the same size containing the corresponding *standard errors*, the bounds within which every covariate or feature of every data point is known to lie. This leads to a so-called *interval matrix model* for the data.

In this report, we address a binary, linear classification problem based on an interval matrix uncertainty model for the data. We develop a *robust* methodology, where we minimize the worst-case value of a loss function, over all possible realizations of the data within given interval bounds. We will show how this worst-case loss function can be upper-bounded by a weighted l_1-norm regularization of the original loss function, explaining the implicit regularization effect within this approach of robust classification.

We consider in detail three specific choices of a loss function. The first, the Hinge loss, is used in the context of soft-margin support vector machines [5, 13]. The second loss function is the negative log likelihood function used in logistic regression (see, e.g., [8]). The third loss function is used in the context of minimax probability machines (MPM), which were recently introduced in [9]. For each case, we will show that the robust methodology leads to problems that are directly amenable to efficient (polynomial-time) convex optimization interior-point algorithms. These optimization problems range from linear programming (LP), to second-order cone programming (a generalization of LP which handles l_2-norm bounds) and constrained maximum entropy. For more on interior-point methods for convex optimization, we refer to [12, 1, 11], and the forthcoming excellent book [3].

The extensive connections between mathematical programming, in particular linear and quadratic programming, and classification, have been successfully explored by a number of authors [2, 5, 13, 10, 4, 7]. Our work is in this line, but with an emphasis on exploiting the specific unknown-but-bounded type of information, that describes data within this interval matrix model. As a result, we end up using more general types of convex optimization algorithms. Our work can be placed on the perspective of a growing concern in optimization for robustness with respect to input data uncertainty (see, e.g., [6], [14, ch. 6]).

2 Setup and Main Results

2.1 Problem setup

The linear learning methods we will describe in the following sections will handle data as uncertain observations, defined within a specific uncertainty model, rather than assume a countable set of well-defined data points. In order to deal with this approach, we define the following setup. Let X denote a $n \times N$ matrix of N *nominal* data points $x_i \in \mathbf{R}^n$, with corresponding label vector $y \in \{-1, +1\}^N$. Let Σ be a $n \times N$ matrix of positive numbers, with columns $\sigma_i, i = 1, \ldots, N$. Finally, let $\rho \geq 0$. Together, X, Σ and ρ describe an *interval matrix model* for a $n \times N$ data matrix Z, via the hyper-rectangle (in the space of $n \times N$ data matrices)

$$\mathcal{X}(\rho) = \left\{ Z \in \mathbf{R}^{n \times N} \ : \ X - \rho\Sigma \leq Z \leq X + \rho\Sigma \right\},$$

where inequalities are understood componentwise. This hyper-rectangle in the space of data matrices corresponds to considering N hyper-rectangles of dimension n in the input space $\mathbf{R}^n$, each of them defining an uncertainty region for each of the N uncertain data points $z_i, i = 1, \ldots, N$. The matrix X is referred to as the *nominal matrix*, while the *standard error matrix* Σ reflects the amplitude of the uncertainty (e.g., measurement errors in microarray experiments) for every covariate. Notice how this uncertainty model considers the uncertainty on the different covariates as independent. Although more specialized approaches are possible — e.g., modelling correlations between uncertainties by assuming a hyper-ellipsoidal rather than a hyper-rectangular uncertainty region —, this uncertainty model already accounts for significant uncertainty information and will lead to sparse classifiers as we will see, which is an important advantage. In Subsection 6.3 we will address how ellipsoidal uncertainty models can be dealt with.

The scalar ρ is a global measure of uncertainty. For clarity, we sometimes drop the dependence of $\mathcal{X}$ on ρ, using by default $\rho = 1$. In that case, the "standard error matrix" Σ reflects the absolute uncertainty for every feature.

For $\epsilon = \pm 1$, we denote by I_ϵ the set of indices for class ϵ, by N_ϵ its cardinality, and set $c_\epsilon = 1/\sqrt{N_\epsilon}$. Define X_ϵ (respectively Σ_ϵ) as the matrix whose i-th column is x_i (respectively σ_i), where i ranges I_ϵ (the order is irrelevant). So X_+, Σ_+ are matrices of size $n \times N_+$, where N_+ is the size of the positive class, and likewise for X_-, Σ_- which corresponds to the negative class.

Based on the data given within this uncertainty model — the training data —, we seek a linear classification rule based on the sign of $w^T x + b$, where $w \in \mathbf{R}^n/\{0\}$ is the weight vector characterizing the classification hyperplane, b is a scalar, and x is a new data point to be classified. To measure the performance of the classifier on the uncertain training set we introduce the *robust loss function* $\mathcal{L}$, which depends on the classifier parameters w, b, the uncertain training set $\mathcal{X}(\rho)$ as well as on the label vector y. This function is defined in terms of a classical, *non-robust loss function* L, which assumes well-defined data points Z, without uncertainty:

$$\mathcal{L}(w, b, \mathcal{X}(\rho), y) = \max_{Z \in \mathcal{X}(\rho)} L(w, b, Z, y). \tag{1}$$

The robust loss function can be interpreted as a worst-case loss function, across all possible values of the data points permitted by our interval uncertainty model. We then consider the following problem, referred to as *robust classification with interval data:*

$$\min_{w,b} \max_{Z \in \mathcal{X}(\rho)} L(w, b, Z, y). \tag{2}$$

Note that, in order to protect the classifier against overfitting on the training data, a regularization term is usually added to the loss term. We will see that the robust loss function can be upper-bounded by a weighted l_1-norm regularization of the original loss function, which explains an implicit, indirect regularization effect for robust classification.

2.2 Three specific loss functions

We consider three specific choices of a loss function. The non-robust loss function used in soft-margin support vector machines (SVMs) [5], known as the linear Hinge loss, is given by

$$L_{\mathrm{SVM}}(w, b, Z, y) = \sum_{i=1}^{N} (1 - y_i(w^T z_i + b))_+, \tag{3}$$

where s_+ denotes the positive part of a scalar s. Augmented with a regularization term that accounts for the complexity of the class of linear maximal margin classification functions, the above loss function provides an upper bound on the number of expected future misclassification errors.

The logistic regression loss function [8] is given by

$$L_{\mathrm{LR}}(w, b, Z, y) = \sum_{i=1}^{N} \log\left(1 + e^{-y_i(w^T z_i + b)}\right). \tag{4}$$

This loss function has a specific interpretation: it is the negative logarithm of the likelihood of the labels y given the data Z, corresponding to a parametric model for the distribution of the label vector and the data.

Finally, we consider a perhaps less classical loss function, which was recently introduced in the context of minimax probability machines (MPMs) [9]:

$$L_{\mathrm{MPM}}(w, Z, y) = \frac{\sqrt{w^T \Gamma_+ w} + \sqrt{w^T \Gamma_- w}}{|w^T(\hat{z}_+ - \hat{z}_-)|}, \tag{5}$$

where $\hat{z}_\pm$ and $\Gamma_\pm$ denote the (empirical) mean and covariance matrix for each class. We will return to the motivation for this loss function in Section 5.

The first two loss functions above are convex in w, b, while the homogeneous MPM loss is convex on a hyperplane defined by $w^T(\hat{z}_+ - \hat{z}_-) = 1$, to which we can restrict the search without loss of generality. We refer to the three methods as *robust linear programming SVM (ROBLP)*, *robust logistic regression (ROBLR)* and *robust minimax probability machine (ROBMPM).*

2.3 Main results

Convexity and complexity. The robust problem inherits the convexity properties of the non-robust counterpart, in that convexity (with respect to the hyperplane parameters w, b) of the loss function L implies that of the worst-case loss function $\mathcal{L}$. For general loss functions, the worst-case loss counterpart may be substantially harder to compute, let alone minimize; however, for all the three loss functions specified above, the robust counterpart is a convex optimization problem that can be solved using polynomial-time interior-point methods for a class of convex optimization problems [12, 3].

Robustness. The method directly handles uncertainty Δx in the data points x. It can be extended to be robust against possible implementation errors Δw in the weight vector w, e.g., when realizing the classifier on a machine with finite precision. This proves useful in a feature selection context: after designing a classifier that is (optimally) robust with respect to component-wise implementation errors Δw_i, one can intentionally apply effective implementation errors Δw_i^{eff} to zero out a (potentially large) number of non-zero coefficients in w. The resulting classifier is specifically trained to be robust against these changes. As this results in a sparse weight vector, this is equivalent to explicit feature selection in a linear classification framework. The connection between data uncertainty and implementation errors, and its relationship with sparsity of the classifier vector, is elaborated upon in the specific case of the SVM loss function. The framework is also extended to handle errors in the labels.

Link with l_1-norm regularization. For all the three loss function considered here, we can approximate from above the worst-case loss function by a weighted l_1-norm regularization of the original loss function. Specifically, we show that

$$\mathcal{L}(w, b, \mathcal{X}(\rho), y) = \max_{Z \in \mathcal{X}(\rho)} L(w, b, Z, y) \leq L(w, b, X, y) + \rho \sigma^T |w|, \tag{6}$$

where $|w|$ denotes the vector with components $|w_i|$, and σ is a vector with non-negative components that depends on the error matrix Σ. The above bound is useful because it helps understand why the robust classification method tends to produce sparse classifiers, as l_1-norm regularization is known to have this effect [4]. It also brings a principled way to choose the regularization weights σ in the context of weighted l_1-norm regularization, should one use this more standard approach to classification. A more general statement:

$$\mathcal{L}(w,b) = \max_{Z \in \mathcal{X}(\rho)} L(w,b,Z,y) \leq \max_{Z \in \mathcal{X}(\kappa\rho)} L(w,b,Z,y) + (1-\kappa)\rho\sigma^T|w|, \tag{7}$$

where $\kappa \in [0,1]$, allows to choose the amount of effort devoted to regularization, via the weighted l_1-norm term in (7), compared to that devoted to robustness. We obviously recover the "pure" robust methodology by setting $\kappa = 1$, and the "pure" weighted l_1-norm regularization approach with $\kappa = 0$.

Sparsity-preserving implementation. Our implementation of the methods, discussed in Section 7, exploits the sparsity of the problem. Specifically, if the nominal matrix X is sparse, and the corresponding matrix Σ has a more general property we refer to as regularity, then it is possible to exploit this fact in the algorithm. This feature results in dramatic speed-ups for large-scale problems where the input data is first made sparse or regular by a filtering operation.

2.4 Outline

We describe the three robust classification methods in Sections 3, 4 and 5. In each case, we derive the related bound (6) based on weighted l_1-norm regularization. We examine several variations on the original theme in Section 6, including the sparsity induced by robustness with respect to errors that are imposed on the classifier coefficients. Furthermore, errors in labels and ellipsoidal uncertainty models are addressed. In Section 7, we discuss our implementation of the methods, which exploits structure such as sparsity, or regularity, of the input data.

3 Robust LP

In this section, we consider the problem of *robust linear programming SVM* (ROBLP) with interval data:

$$\min_{w,b} \max_{Z \in \mathcal{X}(\rho)} \sum_{i=1}^{N} (1 - y_i(w^T z_i + b))_+ \tag{8}$$

where y is a vector of ± 1 labels.

3.1 LP formulation

For the loss function (3), we have the worst-case equivalent

$$\begin{aligned} \mathcal{L}_{\text{SVM}}(w,b) &= \max_{Z \in \mathcal{X}(\rho)} \sum_{i=1}^{N} (1 - y_i(w^T z_i + b))_+ \\ &= \sum_{i=1}^{N} (1 - y_i(w^T x_i + b) + \rho\sigma_i^T|w|)_+. \end{aligned} \tag{9}$$

Hence ROBLP can indeed be implemented as a linear program (LP):

$$\min_{w,b}\ e^T\mathbf{1}\ :\ y_i(w^Tx_i+b)\geq 1-e_i+\rho\sigma_i^T|w|,\ \ e_i\geq 0,\ \ i=1,\ldots,N. \tag{10}$$

We can obtain an upper bound on the worst-case loss function, by exploiting the convexity of the max function. This results in

$$\mathcal{L}_{\text{SVM}}(w,b)\leq\sum_{i=1}^N(1-y_i(w^Tx_i+b))_++\rho\sigma^T|w|,$$

where

$$\sigma:=\sum_{i=1}^N\sigma_i \tag{11}$$

is the sum of errors across all data points. The more general bound (7) is expressed as

$$\mathcal{L}_{\text{SVM}}(w,b)\leq\sum_{i=1}^N(1-y_i(w^Tx_i+b)+\kappa\rho\sigma_i^T|w|)_++(1-\kappa)\rho\sigma^T|w|.$$

Minimizing the above upper bound on the worst-case loss function also leads to an LP:

$$\min_{w,b}\ e^T\mathbf{1}+\rho(1-\kappa)\sigma^T|w|\ :\ y_i(w^Tx_i+b)\geq 1+\rho\kappa\sigma_i^T|w|-e_i,\ \ e_i\geq 0,\ \ i=1,\ldots,N. \tag{12}$$

The above is a generalization of the LP-SVM proposed in [4], namely

$$\min_{w,b}\ Ce^T\mathbf{1}+\|w\|_1\ :\ y_i(w^Tx_i+b)\geq 1-e_i,\ \ i=1,\ldots,N, \tag{13}$$

where the correspondence with the regularization parameter is $C=1/\rho$ and σ is assumed to be all ones, while $\kappa=0$.

For later reference, we note that the dual to the LP (12) is

$$\psi=\min_{\lambda}\ \lambda^T\mathbf{1}\quad:\quad|XY\lambda|\leq\kappa\Sigma\lambda+(1-\kappa)\sigma,\ \ 0\leq\lambda\leq\mathbf{1},\ \ y^T\lambda=0, \tag{14}$$

where $Y=\mathbf{diag}\,(y)$.

3.2 Geometric interpretation

The above problems have an interesting geometric interpretation. For simplicity, we set $\kappa=1$. For the slack variables e set to zero, the constraints in (10) express that the hyperplane defined by (w,b) perfectly separates the data points, irrespective of their values in hyper-rectangles (of shape determined by Σ).

The upper bound can be understood as follows. Assume that we set $\kappa=0$, and the slack variable e to zero in the problem (12). The resulting problem is

$$\min_{w,b}\ \sigma^T|w|\ :\ y_i(w^Tx_i+b)\geq 1,\ \ i=1,\ldots,N. \tag{15}$$

The above corresponds to the following setup. We assume that the data points lie in the hyper-rectangles described by the uncertainty set $\mathcal{X}(\rho)$. Now we seek to maximize the level of uncertainty

ρ, while maintaining a perfect separation, irrespective of the data values in $\mathcal{X}(\rho)$. This leads to the problem

$$\max_{w,b} \rho \ : \ y_i(w^T x_i + b) \geq \rho \sigma^T |w|, \ \ i = 1, \ldots, N. \tag{16}$$

By homogeneity, we can always set $\sigma^T|w| = 1/\rho$, and obtain the LP (15). The classical l_1-norm SVM (13) corresponds to the a robust methodology, in the case when the uncertainty around data points is assumed to be the same across all samples, and has a square shape.

4 Robust LR

In this section, we consider the problem of *robust logistic regression with interval data*. For reasons clarified later, we consider here the more general problem of minimizing the loss function given by (7), where $\kappa \in [0, 1]$ is given. We will show later that it provides an upper bound to the original worst-case loss function arising in problem (2), which corresponds to the choice $\kappa = 1$. Our problem is

$$\min_{w,b} \max_{Z \in \mathcal{X}(\kappa)} \sum_{i=1}^{N} \log\left(1 + e^{-y_i(w^T z_i + b)}\right) + (1 - \kappa)\sigma^T|w|, \tag{17}$$

where, for simplicity, we have set $\rho = 1$, and σ is defined in (11).

4.1 Primal problem

Problem (17) is obviously convex, since it involves the minimization in w, b of a convex function — i.e., the point-wise maximum of convex functions in w, b is convex in w, b. Due to the monotonicity of the terms arising in the loss function, we can eliminate the inner maximization, and obtain:

$$\min_{w,b} \sum_{i=1}^{N} \log\left(1 + e^{-y_i(w^T x_i + b) + \kappa \sigma_i^T |w|}\right) + (1 - \kappa)\sigma^T|w|. \tag{18}$$

Using monotonicity again, we can formulate the above as the convex problem

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

where w_p (respectively w_n) stands for the positive (respectively negative) part of the vector w. The above can be interpreted as an ordinary (regularized) logistic regression problem with additional sign constraints on the classifier.

Note that the corresponding "worst-case" value of the data vector has components

$$x_{\text{wc}}(j) = \begin{cases} x(j) - \sigma(j)\text{sign}(w(j)) & \text{if } y(j) = -1, \\ x(j) + \sigma(j)\text{sign}(w(j)) & \text{otherwise,} \end{cases} \tag{19}$$

where w is optimal for the above problem.

The above problem can be addressed with standard interior-point techniques for convex minimization. However we may further reduce the problem to a maximum entropy problem for which off-the-shelf codes exist [11]. This transformation is valid only when $\kappa < 1$, and this is our motivation for examining the problem above in lieu of the "pure" robust methodology of problem (2).

4.2 Dual problem

As expected, the dual of the above problem has an interesting interpretation in terms of maximum entropy. Define the vectors

$$\xi = \begin{pmatrix} w_p \\ w_n \\ b \end{pmatrix}, \quad v = (\kappa - 1) \begin{pmatrix} \sigma \\ \sigma \\ 0 \end{pmatrix}, \quad a_i = \begin{pmatrix} \kappa\sigma_i - y_i x_i \\ \kappa\sigma_i + y_i x_i \\ -y_i \end{pmatrix}, \quad 1 \le i \le N,$$

and the matrices

$$M = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \end{pmatrix}, \quad A = \begin{pmatrix} a_1 & \cdots & a_N \end{pmatrix}. \tag{20}$$

Then the problem writes

$$\max_{\xi, f} \; v^T \xi - \sum_{i=1}^N \log(1 + e^{f_i}) \; : \; f = A^T \xi, \;\; M\xi \ge 0. \tag{21}$$

Introduce the Lagrangean

$$\mathcal{L}(\xi, f, \lambda, \mu) = v^T \xi - \sum_{i=1}^N \log(1 + e^{f_i}) + \lambda^T (f - A^T \xi) + \mu^T M \xi.$$

The optimality conditions yield

$$v = A\lambda - M^T \mu, \;\; \lambda_i = \frac{e^{f_i}}{1 + e^{f_i}}, \;\; i = 1, \ldots, N.$$

We obtain the dual problem

$$\min_{\lambda, \mu} \; \lambda^T \log \lambda + (\mathbf{1} - \lambda)^T \log(\mathbf{1} - \lambda) \; : \; 0 \le \lambda \le \mathbf{1}, \;\; A\lambda = M^T \mu + v, \;\; \mu \ge 0. \tag{22}$$

Partitioning μ in $\mu = (\mu_+, \mu_-)$, the condition $A\lambda = M^T \mu + v$ expresses as

$$\sum_i (\kappa\sigma_i - y_i x_i)\lambda_i = -(1 - \kappa)\sigma + \mu_+, \tag{23}$$

$$\sum_i (\kappa\sigma_i + y_i x_i)\lambda_i = -(1 - \kappa)\sigma + \mu_-, \tag{24}$$

$$\sum_i y_i \lambda_i = 0. \tag{25}$$

The existence of non-negative vectors μ_+, μ_- such that the above first two conditions hold is equivalent to

$$|XY\lambda| \le \kappa\Sigma\lambda + (1 - \kappa)\sigma, \tag{26}$$

where $Y = \mathbf{diag}\,(y)$. The above can be readily recast as linear inequalities in λ. We obtain the dual problem

$$\psi = \min_{\lambda} \; \lambda^T \log \lambda + (\mathbf{1} - \lambda)^T \log(\mathbf{1} - \lambda) \;\; : \;\; \begin{array}{l} |XY\lambda| \le \kappa\Sigma\lambda + (1 - \kappa)\sigma, \\ 0 \le \lambda \le \mathbf{1}, \;\; y^T \lambda = 0. \end{array} \tag{27}$$

When $\kappa \in [0, 1[$, and σ is componentwise positive, the above problem is strictly feasible. Hence the primal-dual gap is zero in that case.

The above is a maximum entropy problem, directly amenable to efficient interior-point methods (see, e.g., [11]). We can recover the optimal values of the variables w, b by noting that ξ is the variable dual to the equality constraint $A\lambda = M^T\mu + v$, which means that w_p is dual to (23), w_n to (24), and b to (25).

Note that when $\Sigma = 0$ and $\kappa = 1$, the above problem reduces to the maximum entropy problem with exact "moment matching" constraints

$$\min_{\lambda} \lambda^T \log \lambda + (\mathbf{1} - \lambda)^T \log(\mathbf{1} - \lambda) \; : \; XY\lambda = 0, \;\; 0 \le \lambda \le \mathbf{1}, \;\; y^T\lambda = 0,$$

which is of course the dual of the standard logistic regression problem. Here, we observe that the constraint (26) is equivalent to the fact that there exists an exact moment match for at least one data matrix within the uncertainty set $\mathcal{X}$, precisely:

$$\exists Z \in \mathcal{X}, \;\; ZY\lambda = 0, \;\; y^T\lambda = 0. \tag{28}$$

The corresponding data matrix can be interpreted as the "worst-case" realization of the data, that is, it yields the worst-case value of the likelihood function evaluated at the optimal w, b. This worst-case data matrix can be inferred from the expressions (19), yielding

$$X_{\rm wc} = X + \Sigma \cdot (u\mathbf{1}^T),$$

where $u(j) = -y(j)\text{sign}(w(j))$, and $\cdot$ is interpreted as the componentwise product.

When $\Sigma = 0$ but $k < 1$, the moment matching constraints have the form $|XY\lambda| \le \sigma$, which corresponds to the dual to the ordinary LR problem with weighted l_1-norm regularization.

4.3 Upper bounds and approximations

The loss function considered above

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

is an upper bound on the worst-case loss function (which corresponds to $\kappa = 1$) and a lower bound on the weighted l_1-norm regularized LR loss (corresponding to $\kappa = 0$).

The latter bound can be derived from the dual formulation, at least when $\kappa < 1$. Indeed, the following constraints of problem (27):

$$|XY\lambda| \le \kappa\Sigma\lambda + (1 - \kappa)\sigma$$

imply

$$|XY\lambda| \le \sigma,$$

which proves that the weighted l_1-norm regularized LR problem yields an upper bound on problem (17).

Another insight provided by the dual formulation stems from comparing the dual robust LP (14) and its LR counterpart (27). We observe that the two problems share the same feasible region, while the objective in the LP case can be interpreted as a (crude) approximation to that of the LR case. A much better approximation is obtained with a quadratic, leading to the QP

$$\psi = \min_{\lambda} \ \lambda^T(\mathbf{1} - \lambda) \quad : \quad \begin{array}{l} |XY\lambda| \le \kappa \Sigma \lambda + (1-\kappa)\sigma, \\ 0 \le \lambda \le \mathbf{1}, \ \ y^T\lambda = 0. \end{array} \tag{29}$$

5 Robust MPM

In this section, we consider the *robust MPM problem with interval data:*

$$\min_{w,b} \max_{Z \in \mathcal{X}} \frac{\sqrt{w^T\Gamma_+ w} + \sqrt{w^T\Gamma_- w}}{|w^T(\hat{x}_+ - \hat{x}_-)|}, \tag{30}$$

where $\hat{x}_\pm$ and $\Gamma_\pm$ stand for empirical estimates for the class means and covariance matrices. We assume that the uncertainty in the points is smoothed out when computing the class averages $\hat{x}_+, \hat{x}_-$, so the latter information is assumed exact.

We first motivate the choice of this loss function by a brief description of the MPM method. For further details we refer the reader to [9].

5.1 The MPM problem

The minimax probability machine (MPM) introduced in [9] is a binary classification method that uses class averages to control the misclassification error. Let x_+ and x_- denote random vectors in the binary classification problem, respectively modelling data from each of two classes, with means and covariance matrices given by $(\hat{x}_+, \Gamma_+)$ and $(\hat{x}_-, \Gamma_-)$, with $x_+, \hat{x}_+, x_-, \hat{x}_- \in \mathbb{R}^n$, $\Gamma_+, \Gamma_- \in \mathbb{R}^{n \times n}$, and Γ_+, Γ_- both positive semidefinite.

The MPM method determines a hyperplane $\mathcal{H}(w,b) = \{z \mid w^T z = b\}$, where $w \in \mathbb{R}^n \backslash \{0\}$ and $b \in \mathbb{R}$, which separates the two classes of points with maximal probability with respect to all distributions having these means and covariance matrices. This reduces to:

$$\max_{\alpha,w,b} \ \alpha \quad \text{s.t.} \quad \begin{array}{l} \displaystyle\inf_{x_+ \sim (\hat{x}_+, \Gamma_+)} \mathbf{Pr}\{w^T x_+ \ge b\} \ge \alpha \\ \displaystyle\inf_{x_- \sim (\hat{x}_-, \Gamma_-)} \mathbf{Pr}\{w^T x_- \le b\} \ge \alpha, \end{array} \tag{31}$$

where the notation $x_+ \sim (\hat{x}_+, \Gamma_+)$ refers to the class of distributions that have prescribed mean $\hat{x}_+$ and covariance Γ_+, but are otherwise arbitrary; likewise for x_-. For the problem to have a solution we assume that $\hat{x}_+ \neq \hat{x}_-$.

Future points z for which $a^T z \ge b$ are then classified as belonging to the class associated with x_+, otherwise they are classified as belonging to the class associated with x_-. In formulation (31) the term $1 - \alpha$ is an upper bound on the worst-case (maximum) misclassification error, and our classifier minimizes this maximum error.

In [9], it is shown that a pair (w, b) is feasible for the problem if and only if

$$w^T\hat{x}_- + \kappa(\alpha)\sqrt{w^T\Gamma_- w} \le b \le w^T\hat{x}_+ - \kappa(\alpha)\sqrt{w^T\Gamma_+ w}, \tag{32}$$

where $\kappa(\alpha) = \sqrt{\alpha/(1-\alpha)}$.

By eliminating the variable b, we obtain that the MPM problem corresponds to minimizing over w the loss function

$$L_{\mathrm{MPM}}(w, X, y) = \frac{\sqrt{w^T \Gamma_+ w} + \sqrt{w^T \Gamma_- w}}{|w^T (\hat{x}_+ - \hat{x}_-)|}.$$

By homogeneity, we can reduce the MPM problem to computing

$$\phi := \min_w \ \|\Gamma_+{}^{1/2} w\|_2 + \|\Gamma_-{}^{1/2} w\|_2 \ : \ w^T (\hat{x}_+ - \hat{x}_-) = 1.$$

The above is a second-order cone program (SOCP) [3], and can be efficiently solved using interior-point methods for conic programming. If w_* is optimal for the above problem, the optimal lower bound on the misclassification error is then

$$1 - \alpha_* = \frac{\phi^2}{1 + \phi^2}.$$

while an optimal intercept b is recovered as

$$b_* = w_*^T \hat{x}_+ - (1/\phi) \sqrt{w_*^T \Gamma_+ w_*} = w_*^T \hat{x}_- + (1/\phi) \sqrt{w_*^T \Gamma_- w_*}.$$

In practice, the MPM method uses empirical estimates for the means and covariance matrices for each class. The effect of estimation errors are discussed in some detail in [9].

To solve the problem, the original implementation as proposed in [9] requires to form estimates for class covariance matrices and means. Empirical estimates are used (with appropriate shrinkage factors to handle estimation errors), which costs $O(Nn^2)$, where N is the total number of points. Then the factors $\Sigma_\pm^{1/2}$ are formed, and the SOCP problem is solved, at additional cost $O(n^3)$. The cost of the whole implementation is thus $O(n^3 + Nn^2)$.

The paper [9] also develops a kernel version of the MPM. If we use a linear kernel we can solve the problem by first forming the Gram matrix of data points (at cost $O(nN^2)$), then solving an MPM problem with N variables, which leads to a total complexity of $O(N^3 + nN^2)$.

5.2 A robust model

In this section, we consider a variation of the MPM problem, where the input data matrix is unknown-but-bounded in the interval matrix $\mathcal{X}$. This corresponds to problem (31), where the constraints should hold irrespective of our choice of the data matrix in $\mathcal{X}$. For simplicity, we assume that the uncertainty is smoothed out in the class averages $\hat{x}_\pm$, so that they are known exactly.

We first show that this robust MPM problem can be formulated as (30). We start with the basic constraint (32), and seek to guarantee that it holds for every data matrix $Z \in \mathcal{X}$. We note that, if Γ_+, Γ_- are empirical estimates of the covariance matrices of the classes, then

$$\|\Gamma_\pm^{1/2} w\|_2 = c_\pm \sqrt{\sum_{i \in I_\pm} [w^T (z_i - \hat{x}_\pm)]^2},$$

where $c_\pm = 1/\sqrt{N_\pm}$. This leads to

$$w^T \hat{x}_- + c_- \kappa(\alpha) c_- \sigma_-(w) \leq b \leq w^T \hat{x}_+ - \kappa(\alpha) c_+ \sigma_+(w), \tag{33}$$

where

$$\sigma_\pm(w) := \max_{Z\in\mathcal{X}} \sqrt{\sum_{i\in I_\pm} [w^T(z_i - \hat{x}_\pm)]^2} = \sqrt{\sum_{i\in I_\pm} [w^T(x_i - \hat{x}_\pm) + \sigma_i^T|w|]^2},$$

where we have used our assumption that $\hat{x}_\pm$ are known exactly. Maximizing α subject to (33) thus reduces to

$$\phi := \min_w \; c_+\sigma_+(w) + c_-\sigma_-(w) \;:\; w^T(\hat{x}_+ - \hat{x}_-) = 1, \tag{34}$$

which, by homogeneity, corresponds to minimizing the worst-case loss function, as in (30).

If w_* is optimal for the above problem, the optimal lower bound on the worst-case misclassification error is $\phi^2/(1+\phi^2)$, while an optimal intercept b is recovered as

$$b_* = w_*^T\hat{x}_+ - (1/\phi)c_-\sigma_-(w) = w_*^T\hat{x}_- + (1/\phi)c_+\sigma_+(w).$$

5.3 Upper bound

We can introduce an upper bound on the robust problem as follows:

$$\min_w c_+\sqrt{\sum_{i\in I_+}[w^T(x_i - \hat{x}_+)]^2} + c_-\sqrt{\sum_{i\in I_-}[w^T(x_i - \hat{x}_-)]^2} + \sigma^T|w| \;:\; w^T(\hat{x}_+ - \hat{x}_-) = 1, \tag{35}$$

where $\sigma = \tilde{\sigma}_+ + \tilde{\sigma}_-$, with

$$\tilde{\sigma}_\pm := \frac{1}{\sqrt{N_\pm}} \sum_{i\in I_\pm} \sigma_i \tag{36}$$

are related to the class averages of the errors.

In the above approximation, we see that the robustness imposes an l_1-norm regularization term. This term tends to produce sparse classifiers. The cost of solving the regularized problem is roughly the same as that of the original MPM problem.

As before we may generalize the bound above using a parameter $\kappa \in [0,1]$, based on the inequalities

$$\sigma_\pm(w) = \max_{Z\in\mathcal{X}} \sqrt{\sum_{i\in I_\pm}[w^T(z_i - \hat{x}_\pm)]^2} \le \sqrt{\sum_{i\in I_\pm}[w^T(x_i - \hat{x}_\pm) + \kappa\sigma_i^T|w|]^2} + (1-\kappa)\tilde{\sigma}_\pm^T|w|.$$

6 Extensions and Variations

6.1 Implementation errors and sparsity of classifier

The robust approach can be interpreted as a way to alleviate errors stemming not from the data but from the classifier itself. We illustrate this point with the SVM loss function; the other two cases can be treated similarly.

Specifically, consider the problem

$$\min_{w,b} \max_{\|\Delta w\|_\infty \le \delta} L_{\rm SVM}(w + \Delta w, b, X, y).$$

In the above problem, the data matrix is fixed to its "nominal" value X. The errors on w may originate when zeroing out small components, and δ is an absolute measure of these errors. In this sense, a larger δ ensures that more components of w can be safely zeroed out, resulting in a sparser vector.

The worst-case loss function is then

$$\mathcal{L}_{\rm SVM}(w,b) = \sum_{i=1}^N (1 - y_i(w^T x_i + b) + \delta\|x_i\|_1)_+,$$

which can be minimized via LP. We note that we can view δ as a variable, and minimize a combination of the above loss function and a function that decreases with δ. This results in a trade-off between the number of (worst-case) misclassification errors and the sparsity of the classifier.

Alternatively, we can assume that the process of zeroing out coefficients is based on relative rather than absolute size, corresponding to a absolute threshold δ that depends on w. Consider for example the case when $\delta = \kappa\|w\|_1$, where $\kappa \geq 0$ is fixed. The corresponding worst-case loss function can then be interpreted as one of the type (9), where $\rho = \kappa$ and the error matrix has columns set to $\sigma_i = \|x_i\|_1$.

Finally, we note that it is possible to control both implementation and data errors. The corresponding robust problem

$$\min_{w,b} \max_{\|\Delta w\|_\infty \leq \delta,\, Z \in \mathcal{X}(\rho)} L_{\rm SVM}(w + \Delta w, b, Z, y),$$

can be represented as an LP, but at the expense of introducing a large number (Nn to be exact) of new variables. An upper bound on the worst-case loss function is obtained by maximizing over Δw independently in the linear and the norm term in (9), resulting in

$$\mathcal{L}_{\rm SVM}(w,b) = \sum_{i=1}^N (1 - y_i(w^T x_i + b) + \rho\sigma_i^T(|w| + \delta\mathbf{1}) + \delta\|x_i\|_1)_+.$$

6.2 Label errors

We may to some extent handle noise in the labels in a worst-case fashion, as follows. We assume that the data matrix X is fixed, while the label vector (composed of say, ± 1's) is uncertain. Our uncertainty model is that only a few labels are subject to errors; we assume that a fixed number k of labels, where $0 \leq k \leq N$ is given, are subject to a change in sign. Here, k is a bound on the number of label errors. In this section, we consider the SVM loss function only, and examine the problem

$$\min_{w,b} \max_{z \in \mathcal{Y}(y,k)} L_{\rm SVM}(w,b,X,z). \tag{37}$$

where

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

describes the sign uncertainty in the label vector.

Consider first the following sub-problem. Let $\alpha, y \in \mathbf{R}^N$ and $k \leq N$ be given. Define

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

(Later, our task will be to minimize ϕ subject to the constraint $\alpha = X^T w + b$.) We have

$$\begin{aligned}
\phi &= \max_{0 \le t \le \mathbf{1}} \max_{z \in \mathcal{Y}} \sum_{i=1}^N t_i(1 - \alpha_i z_i) \\
&= \max_{0 \le t \le \mathbf{1}} \max_{\delta \in \Delta} \sum_{i=1}^N (t_i(1 - \alpha_i y_i) + 2\delta_i t_i y_i \alpha_i) \\
&= \max_{0 \le t \le \mathbf{1}} \left(\mathbf{1}^T (t - \alpha(y,t)) + \max_{\delta \in \Delta} \delta^T \alpha(y,t) \right),
\end{aligned}$$

where $\Delta = \left\{ \delta \in [0\,1]^N \ : \ \mathbf{1}^T \delta \le k \right\}$, and $(\alpha(y,t))_i = t_i y_i \alpha_i$, $i = 1, \ldots, N$.

Using LP duality, we have for fixed t

$$\max_{\delta \in \Delta} \delta^T \alpha(y,t) = \min_{\lambda \ge 0} \lambda k + \mathbf{1}^T (\alpha(y,t) - \lambda)_+.$$

Hence, we can express the sub-problem as

$$\phi = \max_{0 \le t \le \mathbf{1}} \min_{\lambda \ge 0} \sum_{i=1}^N (t_i(1 - \alpha_i y_i) + \lambda k + (\alpha_i y_i t_i - \lambda)_+).$$

Exploiting weak duality, we obtain

$$\begin{aligned}
\phi &\le \min_{\lambda \ge 0} \max_{0 \le t \le \mathbf{1}} \sum_{i=1}^N (t_i(1 - \alpha_i y_i) + \lambda k + (\alpha_i y_i t_i - \lambda)_+) \\
&= \min_{\lambda \ge 0} \sum_{i=1}^N \max_{0 \le u \le 1} (u(1 - \alpha_i y_i) + \lambda k + (\alpha_i y_i u - \lambda)_+) \\
&= \min_{\lambda \ge 0} \sum_{i=1}^N \max_{u = 0,1} (u(1 - \alpha_i y_i) + \lambda k + (\alpha_i y_i u - \lambda)_+) \\
&= \min_{\lambda \ge 0} \lambda k N + \sum_{i=1}^N ((1 - \alpha_i y_i) + (\alpha_i y_i - \lambda)_+)_+ .
\end{aligned}$$

Finally, problem (37) admits the following upper bound:

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

which is amenable to an LP format. Note that if $k = 0$, that is, there are no errors in the labels, we recover the original problem.

It is possible to develop bounds for a robust classification problem where data, labels, and classifier coefficients are all subject to uncertainty.

6.3 Ellipsoidal uncertainty models

We may address other types of uncertainty models than the interval matrix. Consider the case when the interval uncertainty set $\mathcal{X}$ is replaced with a product of ellipsoids. Specifically, set

$$\mathcal{X} = \left\{ Z = [z_1, \ldots, z_N] \in \mathbf{R}^{n \times N} \ : \ (z_i - x_i)^T \Gamma_i^{-1} (z_i - x_i) \leq 1, \ \ i = 1, \ldots, N \right\}$$

where the matrix $X = [x_1, \ldots, x_N]$ and the positive-definite matrices Γ_i, $i = 1, \ldots, N$, are given.

For the above uncertainty models, we have the following worst-case representations. The SVM worst-case loss function has the form

$$\mathcal{L}_{\mathrm{SVM}}(w, b) = \sum_{i=1}^{N} (1 - y_i(w^T x_i + b) + \|\Gamma_i^{1/2} w\|_2)_+,$$

and an upper bound takes the form

$$\mathcal{L}_{\mathrm{SVM}}(w, b) = \sum_{i=1}^{N} \Big((1 - y_i(w^T x_i + b))_+ + \|\Gamma_i^{1/2} w\|_2 \Big).$$

Both corresponding classification problems can be handled using second-order cone programming algorithms [3], which have polynomial-time complexity. The upper bound is again a regularization of the original loss function, with weights that depend directly on the parameters of the uncertainty model $\mathcal{X}$. The model involves a perhaps non-classical sum of l_2-norms. The algorithm is greatly simplified by assuming the error matrices Γ_i are all equal. Both the robust and the upper bound problem have geometric interpretations in terms of classification of ellipsoids. When all matrices are set to multiples of the identity, the problem reduces to a variant to the classical SVM, with an l_2-norm term instead of a squared l_2-norm in the objective.

The logistic regression loss function leads to the worst-case loss

$$\mathcal{L}_{\mathrm{LR}}(w, b) = \sum_{i=1}^{N} \Big(\log(1 + e^{w^T x_i + b + (1 - 2y_i)\|\Gamma_i^{1/2} w\|_2} - y_i(w^T x_i + b - \|\Gamma_i^{1/2} w\|_2) \Big),$$

which is a convex function of w, b, amenable to the separable convex optimization solver in [11].

Finally, the worst-case MPM (in the case the averages $\hat{x}_\pm$ are exact, and when $w^T(\hat{x}_+ - \hat{x}_-) = 1$) leads to

$$\mathcal{L}_{\mathrm{MPM}}(w, b) = c_+ \sqrt{\sum_{i \in I_+} [w^T(x_i - \hat{x}_+) + \|\Gamma_i^{1/2} w\|_2]^2} + c_- \sqrt{\sum_{i \in I_-} [w^T(x_i - \hat{x}_-) + \|\Gamma_i^{1/2} w\|_2]^2},$$

which is amenable to second-order cone minimization. The upper bound involves a sum of l_2-norms, similar to the one in the SVM model.

7 Implementation

In this section we discuss our implementation of the robust classification models. We have attempted at exploiting as much as possible the potential sparsity of the input, since the optimization algorithms in the MOSEK toolbox [11] do exploit the potential sparsity of the matrix that defines equality or linear inequality constraints.

7.1 A basic implementation

We first discuss an implementation based on MOSEK's matlab toolbox [11] that exploits the potential sparsity of both input data matrices X, Σ.

The algorithm ROBLP implements a variant of the linear programming problem (10), in the form

$$\min_{w_n, w_p, b, e} \ e^T \mathbf{1} + (1-\kappa)\sigma^T(w_p + w_n) \ : \ \begin{array}{l} y_i((w_p - w_n)^T x_i + b) \geq 1 - e_i + \rho\kappa\sigma_i^T(w_p + w_n), \\ e_i \geq 0, \ \ i = 1, \ldots, N, \\ w_p \geq 0, \ \ w_n \geq 0, \end{array} \tag{38}$$

where $w = w_p - w_n$ and $\sigma = \Sigma\mathbf{1}$. The above has $2n + N + 1$ variables and N constraints, without counting the sign constraints on the variables themselves, which MOSEK handles separately. The algorithm SPLP corresponds to the "pure" weighted l_1-regularization, obtained by setting $\kappa = 0$.

The algorithm ROBLR implements the maximum entropy problem (27), via

$$\psi = \min_{\lambda, \mu} \ \lambda^T \log \lambda + \mu^T \log \mu \ : \ \begin{array}{l} \lambda + \mu = \mathbf{1}, \ \ \lambda \geq 0, \ \ \lambda^T \mathbf{1} = N_+, \\ -(\rho\kappa\Sigma\lambda + (1-\kappa)\rho\sigma) \leq XY\lambda \leq \rho\kappa\Sigma\lambda + (1-\kappa)\rho\sigma, \end{array}$$

where $\sigma = \Sigma\mathbf{1}$, as for ROBLP. The above has $2N$ variables and $2n + N + 1$ linear inequality constraints. The code SPLR corresponds to the case $\kappa = 0$.

Finally, ROBMPM implements the second-order cone program (34), via

$$\min_{w_p, w_n, t_\pm, u_\pm, s_\pm} \ c_+ t_+ + c_- t_- \ : \ \begin{array}{l} t_\pm \geq \|u_\pm\|_2, \\ u_\pm = (X_\pm + \rho\Sigma_\pm)^T w_p - (X_\pm - \rho\Sigma_\pm)^T w_n - s_\pm \mathbf{1}, \\ s_\pm = \hat{x}_\pm^T(w_p - w_n), \ \ s_+ - s_- = 1, \\ w_p \geq 0, \ \ w_n \geq 0, \end{array}$$

which has $2n + 2N + 4$ variables and $N + 3$ equality constraints. The code SPMPM implements (35), via

$$\min_{w_p, w_n, t_\pm, u_\pm, s_\pm} \ c_+ t_+ + c_- t_- + \sigma^T(w_p + w_n) \ : \ \begin{array}{l} t_\pm \geq \|u_\pm\|_2, \\ u_\pm = X_\pm^T w_p - X_\pm^T w_n - s_\pm \mathbf{1}, \\ s_\pm = \hat{x}_\pm^T(w_p - w_n), \ \ s_+ - s_- = 1 \\ w_p \geq 0, \ \ w_n \geq 0, \end{array}$$

where σ is a weighted sum of class averages of the σ_i's (see (36)). Note that our implementation avoids forming the covariance matrices and exploits potential sparsity of the input matrices, contrarily to the original implementation of MPM.

7.2 Exploiting structure

In many applications, X often contains a lot of very small elements. It is natural to set those small elements to zero, according to some filtering rule. One possible rule is to set $X(i,j)$ to zero if

$$|X(i,j)| \leq \epsilon\Sigma(i,j),$$

where ϵ is a pre-defined relative threshold level. More sophisticated rules are of course possible, but the above is attractive in the light of our interval uncertainty model. Precisely, if the nominal matrix is filtered with a given threshold level ϵ, then the filtered matrix is guaranteed to be contained in the interval matrix model $\mathcal{X}(\rho)$, provided $\epsilon \leq \rho$.

Sparsity of the nominal matrix can be directly exploited by the l_1-norm regularization formulations (algorithms SPLP, SPLR and SPMPM). In contrast, the matrix Σ is almost never sparse, and the robust classification algorithms ROBLP, ROBLR and ROBMPM cannot directly exploit the sparsity of the nominal matrix X when Σ is dense.

To address this issue, we consider the property of regularity. We say that a matrix is regular if many of its elements are equal, or, more generally, if it is a rank-one modification of a sparse matrix. In our context, we assume that the standard error matrix Σ has the form

$$\Sigma = \sigma_{\text{avg}} \mathbf{1}^T + \delta\Sigma,$$

where $\sigma_{\text{avg}} \geq 0$ can be interpreted as an average of standard errors across experiments, and $\delta\Sigma$ is a sparse matrix. Of course, sparsity is a special case of regularity, with $\sigma_{\text{avg}} = 0$.

When X is sparse and Σ regular, we can modify the basic implementations of ROBLP, ROBLR and ROBMPM so as to exploit both properties. For example, consider the ROBLP formulation (38). Introducing a new variable u and associated constraint $u \geq \sigma_{\text{avg}}^T |w|$, we obtain

$$\begin{aligned} \min_{w_n, w_p, b, e, u} \; z^T v \quad : \quad & y_i((w_p - w_n)^T x_i + b) \geq 1 - e_i + \rho u + \rho(\delta\sigma_i)^T (w_p + w_n), \\ & e_i \geq 0, \quad i = 1, \ldots, N, \\ & w_p \geq 0, \quad w_n \geq 0, \quad u \geq \sigma_{\text{avg}}^T (w_p + w_n). \end{aligned}$$

The above formulation amounts to adding one column and one row to the original constraint matrix, and preserves its sparsity.

Similar results hold for the other two loss functions. Also, the idea can be extended to the case when X is not sparse but regular.

8 Concluding Remarks

We considered a robust, binary, linear classification problem in which the input data is unknown but bounded within hyper-rectangles, i.e., multi-dimensional intervals. By duality, the interval bounds naturally lead to the presence of weighted l_1-norms in the constraints imposed on the classifier coefficients; these terms induce sparsity of the classifier vector. Thus, robustness and sparsity go together. The convexity, monotonicity and separability properties of the loss function all play an important role of making the robust problem amenable to efficient algorithms for finite-dimensional convex optimization. Our implementation exploits potential sparsity, or more generally, regularity, of the input matrices. A next step is to analyze the different methods presented here through experiments on real-world data, as well as investigate how well they perform compared to each other and to previously published methods. This is the subject of research that is currently going on.

References

[1] E. D. Andersen and A. D. Andersen. The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In H. Frenk, C. Roos, T. Terlaky,

and S. Zhang, editors, *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers, 2000.

[2] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.

[3] S.P. Boyd and L. Vandenberghe. *Convex Optimization.* TBA, 2003. Available online at `http://www.stanford.edu/boyd/cvxbook.html`.

[4] P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 13(1):1–10, 2000.

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

[6] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.*, 18(4):1035–1064, October 1997.

[7] Glenn Fung and O. L. Mangasarian. Data selection for support vector machine classifiers. In *Knowledge Discovery and Data Mining*, pages 64–70, 2000.

[8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer-Verlag, 2001.

[9] G.R.G. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[10] O. L. Mangasarian and David R. Musicant. Robust linear and support vector regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):950–955, 2000.

[11] MOSEK ApS, `http://www.mosek.com/`. *MOSEK Optimization Software Manual.*

[12] Yu. Nesterov and A. Nemirovsky. *Interior point polynomial methods in convex programming: Theory and applications.* SIAM, Philadelphia, PA, 1994.

[13] B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[14] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook on Semidefinite Programming.* Kluwer Academic Publishers, 2000.