Ann Oper Res
DOI 10.1007/s10479-012-1303-2

Robust generalized eigenvalue classifier with ellipsoidal uncertainty

Petros Xanthopoulos · Mario R. Guarracino · Panos M. Pardalos

Abstract Uncertainty is a concept associated with data acquisition and analysis, usually appearing in the form of noise or measure error, often due to some technological constraint. In supervised learning, uncertainty affects classification accuracy and yields low quality solutions. For this reason, it is essential to develop machine learning algorithms able to handle efficiently data with imprecision. In this paper we study this problem from a robust optimization perspective. We consider a supervised learning algorithm based on generalized eigenvalues and we provide a robust counterpart formulation and solution in case of ellipsoidal uncertainty sets. We demonstrate the performance of the proposed robust scheme on artificial and benchmark datasets from University of California Irvine (UCI) machine learning repository and we compare results against a robust implementation of Support Vector Machines.

Keywords Robust optimization · Generalized eigenvalue classification · Uncertainty

1 Introduction

Uncertainty is a concept closely related to most real life applications that involve data collection and interpretation. Examples can be found in data acquired with biomedical instruments

P. Xanthopoulos
Industrial Engineering and Management Systems Department, University of Central Florida, 4000 Central Florida Blvd., P.O. Box 162993, Orland, FL, USA
e-mail: petrosx@ucf.edu

M.R. Guarracino
High Performance Computing and Networking Institute, National Research Council of Italy, Naples, Italy

P.M. Pardalos (✉)
Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, P.O. Box 116595, Gainesville, FL, USA
e-mail: pardalos@ufl.edu

Published online: 17 January 2013

such as microarrays or spectroscopes. The result of such acquisition protocols is data perturbed by errors. Since experiments are time and resource demanding, there is a widespread interest for analysis techniques that are resilient to errors. The latter are usually referred to as *robust* and they have been studied extensively in different disciplines, such as statistics (Huber and Ronchetti 1981), and more recently in computer science and engineering (Hubert et al. 2005). Under this framework the term *robust* is used to describe methods able to handle data whose underlying distributions slightly differ from the assumed ones (existence of noise and/or outliers).

However in mathematical programming the term *robust* is used under a different context. More specifically the task of finding the optimal solution subject to an "extreme worst case" scenario is termed *Robust Optimization* (RO). The problem formulated in the uncertainty framework is usually referred to as *Robust Counterpart* (RC) of the original problem. Over the last years RO has gained widespread attention especially after some pioneering works of Ben Tal and Nemirovski (1998, 1999, 2000) and El Ghaui (El Ghaoui and Lebret 1997; El Ghaoui et al. 1998). In addition to the collection of theoretical results, the development of RO has been enhanced by recent advances in interior point algorithms (Vandenberghe 2010). For a comprehensive introduction to RO the reader may refer to Ben-Tal et al. (2009), Ben-Tal and Nemirovski (2002), Bertsimas et al. (2011). In data mining, and more specifically in supervised learning, there have been several studies dealing with RC formulation of Support Vector Machines (SVM), originally developed by Vapnik (1999). The SVM solve the two class supervised classification problem by finding the separating hyperplane with the maximum margin, given that the classes are linearly separable. This leads to the minimization of the classifier's norm, which is computed through the solution of a convex quadratic optimization problem with linear constraints. When classes are not linearly separable, a penalty term is introduced to the objective function, making the problem feasible. The first robust algorithm based on this idea is soft margin SVM (Cortes and Vapnik 1995), where variables are introduced to weight the points in model construction. Since minimizing the training error leads to models that may have poor generalization capabilities due to overfitting, regularization methods have been introduced (Evgeniou et al. 2000; Smola et al. 1998). Over the last decade, SVM formulations tolerant to errors have been studied extensively (Song et al. 2002; Trafalis and Gilbert 2006, 2007). To this extent, all those techniques regularize the original SVM formulation with a term that is specifically tailored for a particular type of perturbations. Regarding SVM, only recently the equivalence between robustness and regularization (Xu et al. 2009) has been proved, for data with spherical uncertainty. Apart from SVM there are several other examples of applications of RO in data mining. In Kim and Boyd (2008), Kim et al. (2006) Boyd et al. proposed an RC formulation for Fisher's Linear Discriminant Analysis (LDA) problem, and proved some useful theoretical results on the RC formulation of the minimization of Raleigh Quotients for matrices with special structure. In El Ghaoui and Lebret (1997), and independently in Shivaswamy et al. (2006), a RC formulation of least squares was proposed whereas Xu et al. (2010) prove a theoretical relation between robust least squares and Least Absolute Shrinkage and Selection Operator (LASSO) regression. In D'Aspremont et al. (2004) a RO based formulation for Principal Component Analysis (PCA), for providing sparse solutions, is proposed. For an overview of RO applications in data mining we refer the reader to Caramanis et al. (2011), Xanthopoulos et al. (2012).

In the present paper we examine and extend the RC formulation to the Regularized Generalized Eigenvalue Classification (ReGEC) algorithm, originally proposed by Mangasarian and Wild (2006) and further studied by Guarracino et al. (2007). As pointed out in the literature generalized eigenvalue classifiers perform better when data points of each class are

located on a linear subspace. In addition experimental analysis (Mangasarian and Wild 2006; Guarracino et al. 2007) show that computation of ReGEC hyperplanes is usually much faster than SVM's separating hyperplane especially for the linear case. Here we propose algorithms for immunizing the classifier against a perturbation scenario, and evaluate its performance through different simulated and real datasets.

The rest of this paper is organized as follows. In Sect. 2 we detail the original Regularized Generalized Eigenvalue classification (ReGEC). In Sect. 3 we describe the uncertainty sets under consideration and we present the RC formulation and the proposed robust algorithm, highlighting some open problems. In Sect. 4 we present the computational experiments. In Sect. 5 conclusions are formulated and open problems are discussed.

A word about notation. Matrices are indicated with capital letters and vectors with small letters. Vectors are always column vectors. The transpose of a vector x is x^T, the transpose of a matrix A is A^T. A column vector of ones of arbitrary dimension is denoted with e. $[A \ -e]$ is the matrix A augmented with a column whose elements are all equal to -1.

2 Regularized generalized eigenvalue classification

Given two classes of points, represented by the rows of matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times n}$, each row being a point in the features space $\mathbb{R}^n$, we wish to determine two hyperplanes, one for each class, with the following characteristics: (a) The sum of the distances between each point in the class and the hyperplane to be minimum, and (b) the sum of the distances between the hyperplane and the points of the other class to be maximum. If we denote the hyperplane related to class A with $w_A^T x - \gamma_A = 0$, the problem consists in determining the variables $w_A \in \mathbb{R}^n$ and $\gamma_A \in \mathbb{R}$ such that

$$\min_{w_A, \gamma_A \neq 0} \frac{\|Aw_A - e\gamma_A\|^2}{\|Bw_A - e\gamma_A\|^2}, \tag{1}$$

where e is a column vector of ones of proper dimension. If we let:

$$G = [A \quad -e]^T [A \quad -e], \qquad H = [B \quad -e]^T [B \quad -e], \tag{2}$$

the problem (1) becomes:

$$\min_{z_A \neq 0} \frac{z_A^T G z_A}{z_A^T H z_A}, \tag{3}$$

with $z_A = [w_A^T \quad \gamma_A]^T$. This is a Rayleigh quotient minimization and the global optimum can be found by solving the following generalized eigenvalue problem

$$Gz = \lambda Hz. \tag{4}$$

Since H and G are real and symmetric matrices by construction, if H is positive definite, the optimal solution is attained at the eigenvector $z_A = z_{\min}$ that corresponds to the minimum eigenvalue $\lambda_{\min}$. By following a similar process, we can determine the hyperplane for class B. Given the symmetry of the problem, if z_B is the eigenvector that corresponds to the maximum eigenvalue of the problem (4), then it is the eigenvector corresponding to the minimum eigenvalue of

$$Hz = \lambda Gz. \tag{5}$$

This means that we can compute the hyperplanes for classes A and B through the eigenvectors related to minimum and maximum eigenvalues of problem (4). The model is able to

predict the class label of an unknown sample x_u by assigning it to the class with minimum distance from the corresponding hyperplane

$$class(x_u) = \arg \min_{i \in \{A,B\}} = \frac{|w_i^T x_u - \gamma_i|}{\|w_i\|}. \tag{6}$$

The multi-class generalization is straightforward. For each class $A_i, i = 1, \ldots, k$, a model is built with respect to every other class A_j, $j \neq i$. The $k-1$ models for each class are then merged using their normal vectors. The latter are averaged using Singular Value Decomposition (SVD), which produces a vector with minimum angle with respect to each normal vector. Such a vector defines a single hyperplane, that is used as classification model for the class A_i. The assignment of a test point to a class is done in two steps. First, the points of each class are projected on their respective plane. Then, the test point is assigned to the class of the nearest neighbor projected point. A detailed description of the method and its performance can be found in Irpino et al. (2010).

In a real world setting, where given data come with induced uncertainty, we wish to determine a classification model that would incorporate such prior information. For this reason one needs to explicitly introduce the following two concepts. The *uncertainty set*, containing all of the admissible perturbations, and the *worst case scenario* problem. Then, the RC solution is usually given as a solution of a min-max problem, in which we minimize the classifier objective function for the maximum/worst case scenario caused by the perturbations in the uncertainty set.

In the original paper on generalized eigenvalue classification by Mangasarian and Wild (2006), since H is not always positive definite, Tikhonov regularization is used for solving the generalized eigenvalue problem. The regularization avoids instabilities and numerical difficulties related to possible singularities of the eigenvalues and multiple eigenvectors produced by a rank deficient H. In this framework, a regularization parameter δ is employed, which is adjusted usually through some trial-and-error procedure; then, $G + \delta I$ and $H + \delta I$, where I is the identity matrix, are used one at a time, instead of G and H, in the solution of two distinct eigenvalue problems that provide the two classification models for data points in A and B. The algorithm proves to have better performance compared to the one without regularization.

Guarracino et al. (2007) proposed an alternative regularization framework for this problem. The regularization framework proposed is the following:

$$(G + \delta_1 \bar{H})z = \lambda(H + \delta_2 \bar{G})z, \tag{7}$$

where $\bar{G}$ and $\bar{H}$ are diagonal matrices whose main diagonals have the diagonal elements of G and H and δ_1, δ_2 are the regularization parameters. This alternative regularization approach has shown to yield higher classification accuracy compared to Tikhonov regularization.

Although regularization has been proposed as a method for enhancing the performance of generalized eigenvalue classification, no attempt has been made to establish a connection between regularization and robustness.

3 Problem description

Suppose that we are given the problem (3) with the additional information that the available value of data points for the first class is $\tilde{A} = A + \Delta A$ and $\tilde{B} = B + \Delta B$, where ΔA and ΔB

are the perturbation matrices. We further impose $\Delta A \in \mathcal{U}_{\mathcal{A}}$, $\Delta B \in \mathcal{U}_{\mathcal{B}}$, where $\mathcal{U}_{\mathcal{A}}, \mathcal{U}_{\mathcal{B}}$ are the sets of admissible uncertainties. If we use the following transformation:

$$\tilde{G} = [\tilde{A} \quad -e]^T [\tilde{A} \quad -e], \qquad \tilde{H} = [\tilde{B} \quad -e]^T [\tilde{B} \quad -e], \tag{8}$$

the RC formulation is given by the following min-max problem:

$$\min_{z \neq 0} \max_{\Delta A \in \mathcal{U}_{\mathcal{A}}, \Delta B \in \mathcal{U}_{\mathcal{B}}} \frac{z^T \tilde{G} z}{z^T \tilde{H} z}. \tag{9}$$

For general uncertainty sets $\mathcal{U}_{\mathcal{A}}, \mathcal{U}_{\mathcal{B}}$ the problem is intractable (Verdú and Poor 2002). Indeed, similar min-max optimization problems, with a Rayleigh quotient objective function, arise in other applications like Fisher Linear Discriminant Analysis (LDA), signal processing and finance (Kim and Boyd 2008; Kim et al. 2006). When the uncertainty information is given in the form of norm bounded inequalities involving matrices ΔG and ΔH, then it is possible to associate Tikhonov regularization, as used in Mangasarian and Wild (2006), with the solution to the following robust problem:

$$\min_{z \neq 0} \max_{\|\Delta G\| \leq \eta_1} \frac{z^T (G + \Delta G) z}{z^T H z}. \tag{10}$$

This can be stated through the following theorem:

Theorem 1 *The solution for the following* min-max *problem*:

$$\min_{z \neq 0} \max_{\|\Delta G\| \leq \eta_1, \|\Delta H\| \leq \eta_2} \frac{z^T (G + \Delta G) z}{z^T (H + \Delta H) z} \tag{11}$$

is given by the eigenvector related to the smallest eigenvalue of the following generalized eigenvalue system:

$$(G + \eta_1 I) z = \lambda (H - \eta_2 I) z. \tag{12}$$

Proof The original problem (11) can be written as:

$$\min_{z \neq 0} \frac{\max_{\|\Delta G\| \leq \eta_1} z^T (G + \Delta G) z}{\min_{\|\Delta H\| \leq \eta_2} z^T (H + \Delta H) z}. \tag{13}$$

We now consider the two individual problems for the numerator and the denominator. More specifically one can write the corresponding Karush-Kuhn Tucker (KKT) system for the numerator problem:

$$z z^T + \lambda 2 \Delta G = 0, \tag{14}$$

$$\lambda \big(\|\Delta G\| - \eta_1 \big) = 0. \tag{15}$$

Solving Eq. (14) with respect to ΔG yields:

$$\Delta G = -\frac{1}{2\lambda} z z^T. \tag{16}$$

From Eq. (15) we obtain:

$$\lambda = \pm \frac{\|z\|^2}{2\eta_1}, \tag{17}$$

which gives the final expression for ΔG:

$$\Delta G = \pm \frac{\eta_1}{\|z\|^2} z z^T. \tag{18}$$

These two matrices correspond to the maximum and the minimum of the problem, respectively. Substituting Eq. (18) in the numerator of Eq. (11) we obtain: $\min_{z\neq 0} z^T(G+\eta_1 I)z$. Repeating the same process for the denominator and substituting into the master problem shows that Eq. (11) is equivalent to the following minimization problem:

$$\min_{z\neq 0} \frac{z^T(G+\eta_1 I)z}{z^T(H-\eta_2 I)z}. \tag{19}$$

This problem attains its minimum when z is equal to the eigenvector that corresponds to the smallest eigenvalue of the generalized eigenvalue problem of Eq. (12). □

The regularization employed by Mangasarian et al. (2006) uses $\Delta G = \eta_1 I$, $\Delta H = 0$ for the solution of the first, and $\Delta G = 0$, $\Delta H = \eta_2 I$ for the second problem and therefore it is a form of robustness. Unfortunately, when bounds are given on the norms of ΔG and ΔH, we have no direct way to derive the perturbations introduced on points, namely ΔA and ΔB. This means that the regularization implies robustness, but we cannot relate it to perturbations in training points. The same reasoning applies to ReGEC, If we use $\bar{G}$ instead of $\eta_1 I$ and $\bar{H}$ instead of $\eta_2 I$ in Eq. (7) to regularize numerator and denominator, respectively.

It is worth noting that a similar theorem has been stated in Shahbazpanahi et al. (2003) for a completely different problem in the field of adaptive filter design. Here it is the first time that its relation with robust classification methods has been shown to provide a straightforward connection between regularization and robustness for generalized eigenvalue classifiers.

When the uncertainty information is not given in the above form, e.g. we have information for each specific data point, the solution provided by Eq. (12) gives a very conservative estimate of the original solution. For this reason we propose an alternative algorithmic solution that can take into consideration additional information available for the problem.

3.1 RC under ellipsoidal uncertainty set

Now we will focus on the following uncertainty sets where the perturbation information is explicitly given for each data point in the form of an ellipsoid:

$$\mathcal{U}_{\mathcal{A}} = \left\{\Delta A \in \mathbb{R}^{m\times n}, \Delta A = \left[\delta_i^{(A)}\delta_2^{(A)}\ldots\delta_m^{(A)}\right]^T : \delta_i^{(A)T}\Sigma_i\delta_i^{(A)} \leq 1, i = 1,\ldots,m\right\}, \tag{20}$$

and

$$\mathcal{U}_{\mathcal{B}} = \left\{\Delta B \in \mathbb{R}^{p\times n}, \Delta B = \left[\delta_i^{(B)}\delta_2^{(B)}\ldots\delta_p^{(B)}\right]^T : \delta_i^{(B)T}\Sigma_i\delta_i^{(B)} \leq 1, i = 1,\ldots,p\right\}, \tag{21}$$

where $\delta_i^{(A)}, i = 1,\ldots,m$ and $\delta_i^{(B)}, i = 1,\ldots,p$ are the individual perturbations that occur in each sample and $\Sigma_i \in \mathbb{R}^{n\times n}$ is a positive definite matrix that defines the ellipse's size and rotation. This covers the Euclidean norm case when Σ_i is equal to the unit matrix.

Since the objective functions in numerator and denominator in Eq. (1) are nothing but the sum of distances of the points of each class from the class hyperplane, we can consider the problem of finding the maximum (or minimum) distance from an ellipse's point to the hyperplane defined by $w^T x - \gamma = 0$. Since the distance of a point to a hyperplane is given by $|w^T x - \gamma|/\|w\|$ the problem can be written as:

$$\max \ \left|w^T x - \gamma\right| \tag{22a}$$

$$\text{s.t.} \quad (x - x_c)^T \Sigma (x - x_c) - 1 \leq 0 \tag{22b}$$

where x_c is the ellipse's center. We can consider the two cases of the problem:

$$\max \; w^T x - \gamma \tag{23a}$$
$$\text{s.t.} \quad (x - x_c)^T \Sigma (x - x_c) - 1 \leq 0, \tag{23b}$$
$$w^T x \geq 0 \tag{23c}$$

and

$$\max \; -w^T x + \gamma \tag{24a}$$
$$\text{s.t.} \quad (x - x_c)^T \Sigma (x - x_c) - 1 \leq 0, \tag{24b}$$
$$w^T x \leq 0. \tag{24c}$$

Now, let us consider problem (23a)–(23c). The corresponding KKT system will be:

$$w^T - 2\mu_1 (x - x_c)\Sigma - \mu_2 w^T = 0, \tag{25a}$$
$$(x - x_c)^T \Sigma (x - x_c) - 1 \leq 0. \tag{25b}$$

Also, there exist μ_1, μ_2 such that:

$$\mu_1 \left[(x - x_c)^T \Sigma (x - x_c) - 1 \right] = 0, \tag{26a}$$
$$-\mu_2 w^T x = 0. \tag{26b}$$

From Eq. (26b) we derive $\mu_2 = 0$, because in different case the point would satisfy the equation of the line. If we solve Eq. (25a) with respect to x and substitute in Eq. (25b) we obtain the following expression for μ_1:

$$\mu_1 = \pm \frac{\sqrt{w^T \Sigma^{-1} w}}{2}, \tag{27}$$

which gives the expression for x:

$$x = x_c \pm \frac{\Sigma^{-1} w}{\sqrt{w^T \Sigma^{-1} w}}. \tag{28}$$

The two solutions correspond to the points of the ellipsoid at minimum and maximum distance from the line. If we consider problem (24a)–(24c) we can derive the same result. Since the solution is expressed as a function of w, and thus z, our original problem becomes:

$$\min_{z \neq 0} \max_{\Delta A \in \mathcal{U}_A, \Delta B \in \mathcal{U}_B} \frac{z^T G z}{z^T H z} = \min_{z \neq 0} \frac{z^T H(z) z}{z^T G(z) z}. \tag{29}$$

where

$$G(z) = \left[\tilde{A}(z) \quad -e \right]^T \left[\tilde{A}(z) \quad -e \right], \qquad \tilde{A}(z) = \begin{bmatrix} a_1 \pm \frac{\Sigma_1^{-1} w}{\sqrt{w^T \Sigma_1^{-1} w}} \\ \dots \\ a_m \pm \frac{\Sigma_m^{-1} w}{\sqrt{w^T \Sigma_m^{-1} w}} \end{bmatrix} \tag{30}$$

and

$$H(z) = \left[\tilde{B}(z) \quad -e \right]^T \left[\tilde{B}(z) \quad -e \right], \qquad \tilde{B}(z) = \begin{bmatrix} b_1 \pm \frac{\Sigma_1^{-1} w}{\sqrt{w^T \Sigma_1 w}} \\ \dots \\ b_p \pm \frac{\Sigma_p^{-1} w}{\sqrt{w^T \Sigma_p w}} \end{bmatrix} \tag{31}$$

The latter problem cannot be solved with the corresponding generalized eigenvalue problem, because the matrices depend on z. In addition, its complexity is not known and it is an open problem for future research. For these reasons we use an iterative algorithm for finding the solution. First, we solve the nominal problem and we use this solution as the starting point of the iterative algorithm. Next, starting from the initial solution hyperplanes, we estimate the "worst case" point by Eq. (28). Then, we solve again the problem but this time for the updated points. The process is repeated until the solution hyperplanes converges (becoming smaller than a predetermined number ϵ), or a maximum number of iterations $i_{\max}$ has been reached. In the case of non convergence we chose the solution with better objective function value. So far we do not have a formal proof about the convergence of this proposed algorithmic scheme which is an open problem for future research. An algorithmic description of the iterative procedure is shown in Algorithm 1.

Algorithm 1 Training Robust Iterative ReGEC

$z_1 = [w_1^T \ \gamma_1]^T = \arg\min \frac{\|Aw - e\gamma\|^2}{\|Bw - e\gamma\|^2}$
$A^{(1)} \leftarrow A, B^{(1)} \leftarrow B$
$z_0 \leftarrow$ any value such that $\|z_1 - z_0\| > \epsilon$
$i \leftarrow 1$
while $\|z_i - z_{i-1}\| \leq \epsilon$ or $i \leq i_{\max}$ **do**
  $i \leftarrow i + 1$
  for each row $x_j^{(i)}$ of matrix $A^{(i)}$ **do**
    $x_j^{(i)} \leftarrow \max\left\{x_j^{(1)} + \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}, x_j^{(1)} - \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}\right\}$
  end for
  for each row $x^{(i)}$ of matrix $B^{(i)}$ **do**
    $x_j^{(i)} \leftarrow \min\left\{x_j^{(1)} + \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}, x_j^{(1)} - \frac{\Sigma_j^{-1} w_i}{\sqrt{w_i^T \Sigma_j^{-1} w_i}}\right\}$
  end for
  Form updated matrices $A^{(i)}, B^{(i)}$
  $z_i = [w_{i+1}^T \ \gamma_{i+1}]^T = \arg\min \frac{\|A^{(i)} w - e\gamma\|^2}{\|B^{(i)} w - e\gamma\|^2}$
end while
return $z_i = [w_i^T \ \gamma_i]^T$

A similar process is followed for class B. In case of k classes, Algorithm 1 is applied for all the combinations of classes and then the final hyperplane is obtained through SVD on the matrix containing the $k - 1$ normal vectors. This is the same process followed for the original ReGEC (Irpino et al. 2010). Also, it is worth noting that the testing part of the algorithm is exactly the same as in case of the nominal problem. Thus the labels of the unknown points are decided with the rule described by Eq. (6) where w_i are estimated from Algorithm 1, in case of two classes, and with the training point projection method explained earlier, in case of three or more classes.

3.2 Balancing between robustness and optimality

Robust approaches have been criticized for providing over conservative solutions in the sense that they are optimal only when the worst assumed scenario occurs (Bertsimas and

Sim 2004). In real life applications, it might be more interesting to robustify an algorithm against an average case scenario. For this reason, we propose an adjustable model. Instead of directly using Eq. (28), we can take the convex combination of the nominal data points and the "robust" ones:

$$x^{balanced} = \xi \cdot x_c + (1-\xi)\left(\frac{\Sigma^{-1}w}{\sqrt{w^T\Sigma^{-1}w}} + x_c\right)$$
$$= x_c + (1-\xi)\left(\frac{\Sigma^{-1}w}{\sqrt{w^T\Sigma^{-1}w}}\right), \quad 0 \le \xi \le 1. \tag{32}$$

Parameter ξ determines how close points will be to their nominal values or to their "worst scenario" ones. For computational purposes we can choose ξ by generating several average based scenarios and selecting the value that gives the lowest objective function value for the training data points. We are going to call this method ξ-Robust Regularized Generalized Eigenvalue Classifier (ξ-R-ReGEC).

4 Computational results

4.1 A case study

Now we will illustrate how ξ-R-ReGEC works under ellipsoidal uncertainty. In this example each class is composed of three points in a two dimensions space. Let us assume for all points the ellipsoidal uncertainty is the same and it is described by the matrix $\Sigma = I \cdot [\alpha^{-2}\ \beta^{-2}]^T$. We first consider the simple two class example where each class is represented by:

$$A = \begin{bmatrix} 5.00 & 4.00 & 4.63 \\ 7.11 & 5.36 & 4.42 \end{bmatrix}^T, \qquad B = \begin{bmatrix} 2.82 & 2.00 & 1.00 \\ 1.44 & 7.11 & -0.68 \end{bmatrix}^T. \tag{33}$$

Those points are shown in Fig. 1.

In order to examine the behavior of the robust solution we perform the following test. We compute the nominal solution based on the data values without any perturbation. Then we assume an ellipsoidal perturbation in two dimensions and we compute the robust solution. We create 1000 different realizations (different sets of points perturbed with ellipsoid perturbation) and we compute the objective function value for the robust classifier and the nominal one. This experiment is repeated for increasing values of α and β. The results are shown in Fig. 2.

We note that for small values of ellipse parameters the robust solution is very conservative. This means that it might be optimal for the assumed worst case scenario, but it does not perform well in the average case. As the perturbation increases robust solution tends to have a constant behavior for any realization of the system.

4.2 Experiments

Now we will demonstrate the performance of the algorithm on data sets from UCI repository (Frank and Asuncion 2010). Dataset characteristics are shown in Table 1.

For each run we used hold out cross validation with 50 repetitions. In every repetition 90 % of the samples were used for training and 10 % for testing. At each repetition we train the robust algorithm with the nominal data plus the uncertainty information and we test it on a random realization of the testing dataset, that is nominal values of testing dataset plus

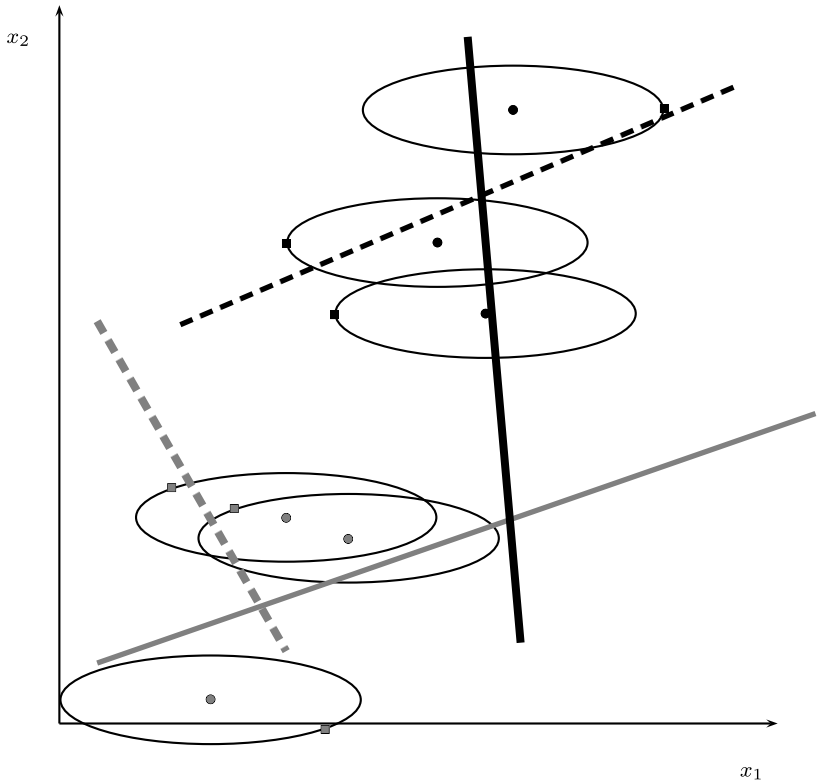

Fig. 1 This is an illustrative example where the *gray class* (class A) consists of $(5, 7.11)$, $(4, 5.36)$, $(4.63, 4.42)$ and the *black class* (class B) consists of $(2.82, 1.44)$, $(2, 1.72)$, $(1, -0.68)$. The original points are denoted with *circles* whereas the points used for the robust solution with *squares*. The Original ReGEC classifications hyperplanes are shown in *solid lines* whereas the robust counterparts are shown in *dashed*

noise that satisfies the ellipsoidal uncertainty constraints. The uncertainty was kept constant and equal to 0.1 for all dimensions (features). For all computational experiments $\epsilon = 10^{-2}$ and $i_{\max} = 10$ were used. For the non fixed dimension the perturbation is set equal to α that is a parameter for our experiments. All data features are initially normalized so that they have 0 mean and unitary standard deviation. All code was developed in MATLAB. The robust SVM solver used for comparison is written in Python and runs under Matlab.

The results are reported in the following Figs. 3, 4, 5 and 6 in form of heat maps. For each considered dataset, we plot the results of ReGEC algorithm in the left panel, and those attained by ξ-R-ReGEC in the right one. Results have been obtained with the above described cross validation technique, for each fixed value of ξ and α. Each tone of gray represents the average accuracy for all cross validation repetitions, as reported in the legend. We notice that, in all considered cases, right panels show higher classification values for larger values of α. This confirms that it is possible to find values of ξ for which the classification model is resilient to perturbations. We notice that some datasets are less sensitive to the value of ξ. In the Pima Indian and Iris datasets, the classification accuracy increases as the value of α increases for $0.4 \leq \xi \leq 0.8$. For Wine and NDC datasets, the robust classifier obtains nearly constant classification accuracy for large values of α and $0 \leq \xi \leq 0.8$.

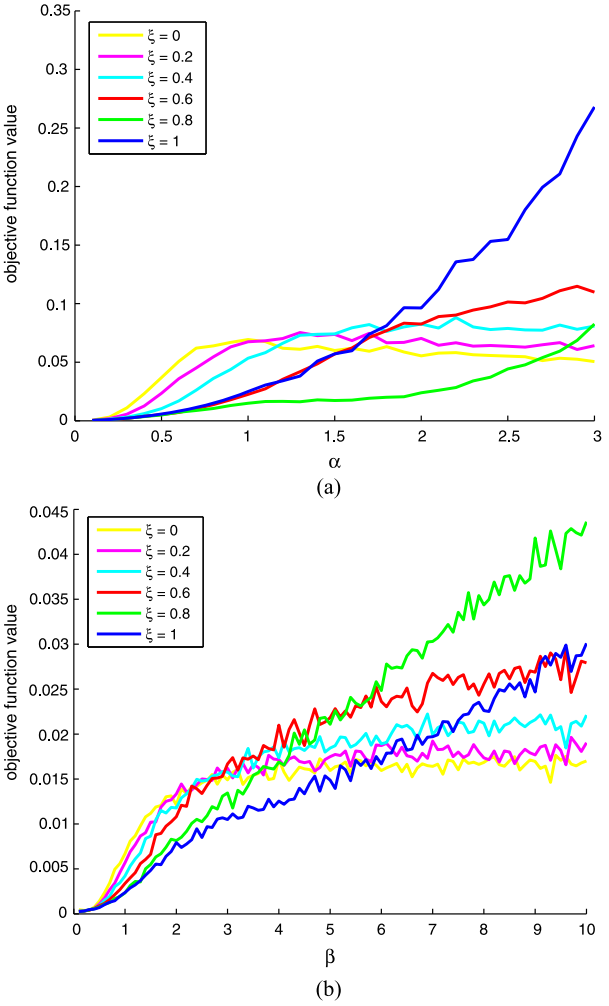

Fig. 2 Objective function value dependence on the ellipse parameters

Table 1 Dataset description

Dataset name	# of points	# of attributes	# of classes
Pima Indian (Smith et al. 1988)	768	8	2
Iris (Fisher et al. 1936)	150	4	3
Wine (Aeberhard et al. 1992a, 1992b)	178	13	3
NDC (Musicant 1998)	300	7	2

Next we compare our proposed algorithm against Robust SVM. For this purpose we used the package CVXOPT (Vandenberghe 2010) and in particular the customized solver for Robust SVM under ellipsoidal uncertainties also described in Andersen et al. (2011). The original formulation of the problem is from Shivaswamy et al. (2006). More precisely, the R-SVM problem is solved as a second order cone quadratic program (QP) with second

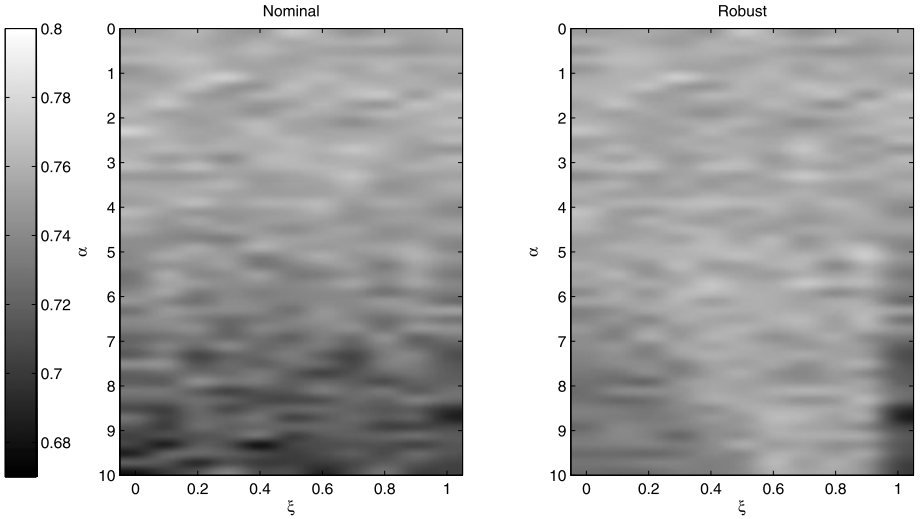

Fig. 3 Analysis for Pima Indian dataset. The *horizontal axis* determines the ξ parameter that balances between robust and nominal solution whereas the *vertical axis* is the perturbation parameter

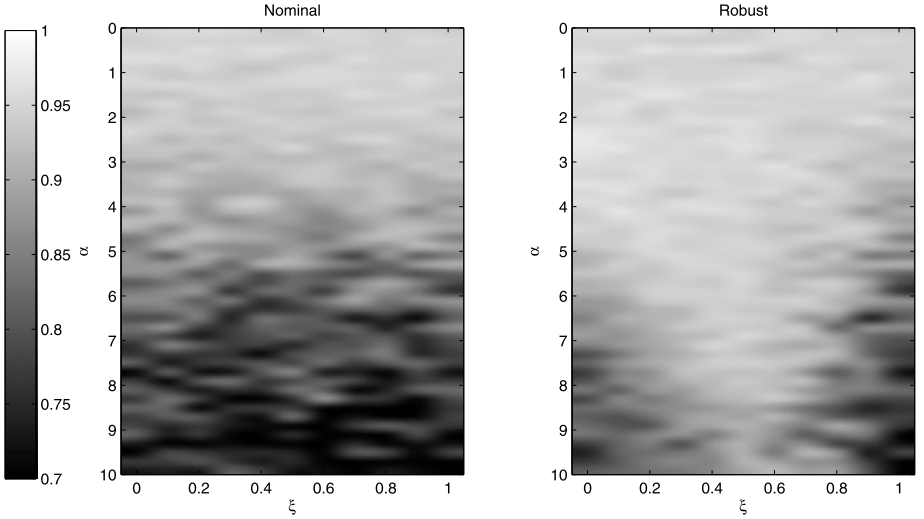

Fig. 4 Analysis for Iris dataset. The *horizontal axis* determines the ξ parameter that balances between robust and nominal solution whereas the *vertical axis* is the perturbation parameter

order cone constraints. For a two class classification problem, the separating hyperplane is given by the solution of the following Second Order Cone Program (SOCP):

$$\min \ \frac{1}{2}\|w\|^2 + \xi e^T \upsilon \tag{34a}$$

$$\text{s.t.} \ \ \text{diag}(d)(Xw - be) \succeq 1 - \upsilon - Eu, \tag{34b}$$

$$u \succeq 0, \tag{34c}$$

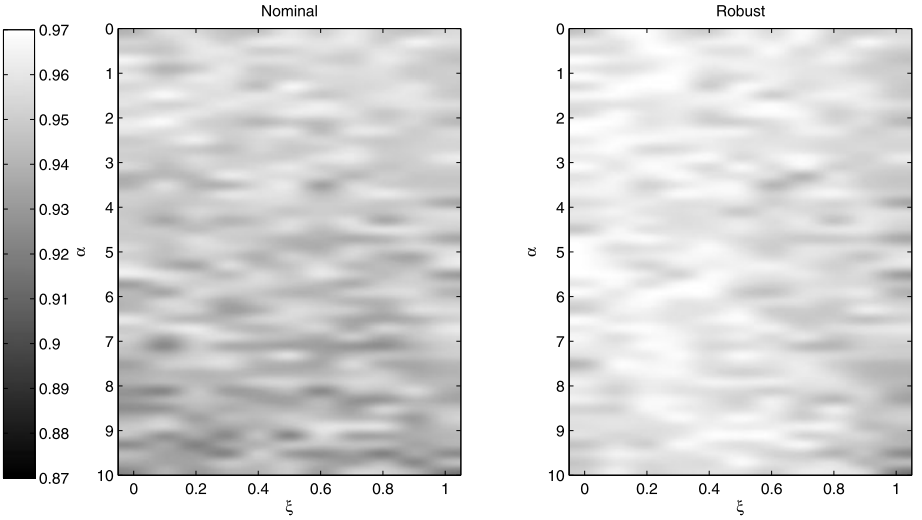

Fig. 5 Analysis for wine dataset. The *horizontal axis* determines the ξ parameter that balances between robust and nominal solution whereas the *vertical axis* is the perturbation parameter

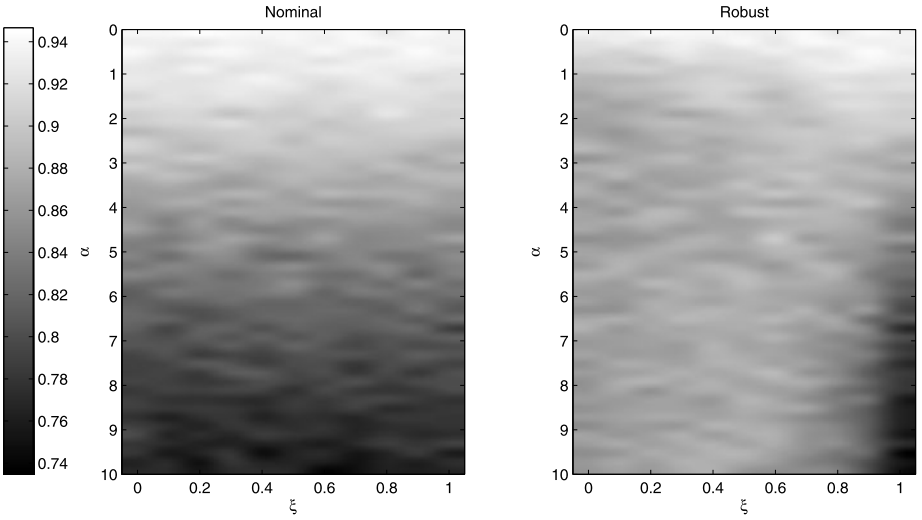

Fig. 6 Analysis for dataset NDC. The *horizontal axis* determines the ξ parameter that balances between robust and nominal solution whereas the *vertical axis* is the perturbation parameter

$$\|\Sigma_j\| \leq u_j, \quad j = 1, \ldots, t, \tag{34d}$$

where the diag(x) is the diagonal matrix that has vector x in the main diagonal, $d \in \{0, 1\}^m$ is the label vector, E is an indicator matrix that associates an ellipse with a corresponding data point, i.e. if $E_{ij} = 1$ means that the ith ellipsoid is associated with the jth data point, and Σ_j is a positive semidefinite matrix that defines the shape of the ellipse. The separation hyperplane is defined by $w, b. \upsilon, u$ are additional decision variables and ξ is a parameter that

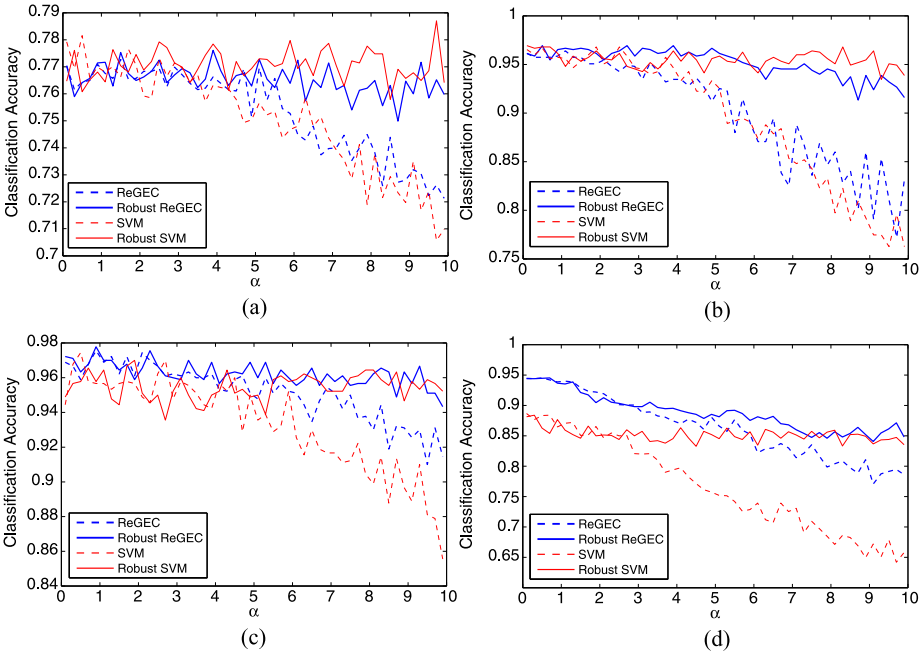

Fig. 7 Comparison graph for (**a**) Pima Indian, (**b**) Iris, (**c**) Wine and (**d**) NDC datasets. *Red color* corresponds to ReGEC whereas *blue* for SVM. *Solid line* corresponds to the robust and *dashed* to nominal formulation (Color figure online)

penalizes the wrong classified samples. Note that for $t = 0$ and $E = 0$ the problem reduces to the original soft margin SVM classifier:

$$\min \ \frac{1}{2}\|w\|^2 + \xi e^T \upsilon \tag{35a}$$

$$\text{s.t.} \ \ \text{diag}(d)(Xw - be) \succeq 1 - \upsilon, \tag{35b}$$

$$u \succeq 0. \tag{35c}$$

The results obtained by ReGEC and SVM for the four datasets of Table 1 are shown in Fig. 7.

In Fig. 7 vertical axes correspond to mean classification accuracies. Like before, the hold out cross validation with 50 repetitions is used. For ReGEC the ξ parameter was adjusted in each run through the training phase and for SVM γ was chosen through a uniform grid search. It is worth mentioning that alterations of γ did not change dramatically the obtained classification accuracies.

In all cases and for both algorithms, as α increases, the accuracy of the two nominal algorithms decreases more rapidly than for RCs. Furthermore, the RCs always obtain higher accuracy values. For three datasets (Pima Indian, Wine and Iris) both robust algorithms perform equally well even for large perturbation values. For NDC, ReGEC is better especially for perturbations less that $\alpha = 5$. In general, both non robust algorithms fail to achieve a high classification accuracy for high perturbation values. We can therefore conclude that the proposed robustification of the algorithm is well suited for problems in which uncertainty on training points can be modeled as belonging to ellipsoid.

5 Conclusions—future work

In this work a robust formulation of ReGEC is introduced, that can handle perturbation in the form of ellipsoidal uncertainties in data. An iterative algorithm is proposed to solve the associated min-max problem. Results on UCI datasets show an increased classification accuracy, when synthetic ellipsoidal perturbations are introduced to training data.

The present work gave rise to a number of additional research questions which still remain open. First, as we outlined in Sect. 3, a computational complexity analysis of the proposed algorithm, together with a formal proof of its convergence, might give new insight in the class of problems that can be solved. Then, in the kernel version of the algorithm, since data are non linearly projected in the feature space, a representation of the ellipsoidal perturbation in the feature space is needed. Further, we note that this embedding is not trivial, as the convex shape of the ellipsoid in the original space is not preserved in the feature space, thus resulting in a much more difficult problem. For this reason, the direction proposed in Pothin and Richard (2006) seems promising and worth investigating. In addition, further uncertainty scenarios need to be explored. Some interesting uncertainty sets include norm bounded uncertainties on the data matrix, i.e. when $\|\Delta A\| \leq \rho$, box constraints, i.e. when $\|\delta_i\|_1 \leq \rho_i$ or polyhedral constraints, i.e. when $A_i \delta_i \leq b_i$. Most of these scenarios have been explored for other classifiers (e.g. SVM) introducing significant performance improvement over the nominal solution schemes. Finally a future extension, which again would be interesting especially in the field of biomedical data classification is to investigate the equivalent formulation for the case of imbalanced datasets and incremental learning (Cifarelli et al. 2007). It is worth noting that although models for robust optimization and imbalanced classification have been developed independently yet today, to the best of authors knowledge, there has not been developed a single classifier combining these two characteristics.

Acknowledgements This project was partially funded by National Science Foundation (N.S.F.) grants and Italian Flagship Project *Interomics* funded by MIUR and CNR.

References

Aeberhard, S., Coomans, D., & De Vel, O. (1992a). *Comparison of classifiers in high dimensional settings*. Tech. Rep. Dept. Math. Statist., James Cook Univ. North Queensland, Australia

Aeberhard, S., Coomans, D., & De Vel, O. (1992b). *The classification performance of RDA*, Cambridge. Tech. Rep. (pp. 92–01). Dept. of Computer Science/Dept. of Mathematics and Statistics, James Cook University of North Queensland

Andersen, M. S., Dahl, J., Liu, Z., & Vandenberghe, L. (2011). Interior-point methods for large-scale cone programming. *Optimization for machine learning*. Cambridge: MIT Press.

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. S. (2009). *Robust optimization*. Princeton: Princeton University Press.

Ben-Tal, A., & Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research*, *23*(4), 769–805.

Ben-Tal, A., & Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters*, *25*(1), 1–14.

Ben-Tal, A., & Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, *88*(3), 411–424.

Ben-Tal, A., & Nemirovski, A. S. (2002). Robust optimization—methodology and applications. *Mathematical Programming*, *92*(3), 453–480.

Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review 53*(3), 464–501.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, *52*(1), 35–53.

Caramanis, C., Mannor, S., & Xu, H. (2011). Robust optimization in machine learning. In S. Sra, S. Nowozin, & S. J. Wright (Eds.), *Optimization for machine learning* (pp. 369–402). Cambridge: MIT Press.

Cifarelli, C., Guarracino, M. R., Seref, O., Cuciniello, S., & Pardalos, P. M. (2007). Incremental classification with generalized eigenvalues. *Journal of Classification*, *24*(2), 205–219.
Cortes, C., & Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.
D'Aspremont, A., Ghaoui, L., Jordan, M., & Lanckriet, G. (2004). A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, *49*(3), 434–448.
El Ghaoui, L., & Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, *18*, 1035–1064.
El Ghaoui, L., Oustry, F., Lebret, H., et al. (1998). Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, *9*, 33–52.
Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization networks and support vector machines. *Advances in Computational Mathematics*, *13*(1), 1–50.
Fisher, R., et al. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.
Frank, A., & Asuncion, A. (2010). *UCI machine learning repository*. http://archive.ics.uci.edu/ml.
Guarracino, M. R., Cifarelli, C., Seref, O., & Pardalos, P. M. (2007). A classification method based on generalized eigenvalue problems. *Optimization Methods & Software*, *22*(1), 73–81.
Huber, P., & Ronchetti, E. (1981). *MyiLibrary: robust statistics* (Vol. 1). Hoboken: Wiley Online Library.
Hubert, M., Rousseeuw, P., & Vanden Branden, K. (2005). ROBPCA: a new approach to robust principal component analysis. *Technometrics*, *47*(1), 64–79.
Irpino, A., Guarracino, M. R., & Verde, R. (2010). Multiclass generalized eigenvalue proximal support vector machines. In *4th IEEE conference on complex, intelligent and software intensive systems (CISIS 2010)*. (pp. 25–32). Los Alamitos: IEEE Computer Society.
Kim, S. J., & Boyd, S. (2008). A minimax theorem with applications to machine learning, signal processing, and finance. *SIAM Journal on Optimization*, *19*(3), 1344–1367.
Kim, S. J., Magnani, A., & Boyd, S. (2006). Robust fisher discriminant analysis. *Advances in Neural Information Processing Systems*, *18*, 659.
Mangasarian, O. L., & Wild, E. W. (2006). Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(1), 69–74.
Musicant, D. R. (1998). *NDC: normally distributed clustered datasets*. http://www.cs.wisc.edu/dmi/svm/ndc/.
Pothin, J., & Richard, C. (2006). *Incorporating prior information into support vector machines in the form of ellipsoidal knowledge sets*. Citeseer.
Shahbazpanahi, S., Gershman, A., Luo, Z., & Wong, K. (2003). Robust adaptive beamforming using worst-case SINR optimization: a new diagonal loading-type solution for general-rank signal models. In *2003 IEEE international conference on acoustics, speech, and signal processing. Proceedings (ICASSP'03)* (Vol. 5). New York: IEEE Press.
Shivaswamy, P., Bhattacharyya, C., & Smola, A. (2006). Second order cone programming approaches for handling missing and uncertain data. *The Journal of Machine Learning Research*, *7*, 1283–1314.
Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Johns Hopkins APL Technical Digest*, *10*, 262–266.
Smola, A. J., Schölkopf, B., & Müller, K. R. (1998). The connection between regularization operators and support vector kernels. *Neural Networks*, *11*(4), 637–649.
Song, Q., Hu, W., & Xie, W. (2002). Robust support vector machine with bullet hole image classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *32*(4), 440–448.
Trafalis, T. B., & Gilbert, R. C. (2006). Robust classification and regression using support vector machines. *European Journal of Operational Research*, *173*(3), 893–909.
Trafalis, T. B., & Gilbert, R. C. (2007). Robust support vector machines for classification and computational issues. *Optimization Methods & Software*, *22*(1), 187–198.
Vandenberghe, L. (2010). *The CVXOPT linear and quadratic cone program solvers*.
Vapnik, V. N. (1999). *The nature of statistical learning theory*. *Information science and statistics*. Berlin: Springer.
Verdú, S., & Poor, H. (2002). On minimax robustness: a general approach and applications. *IEEE Transactions on Information Theory*, *30*(2), 328–340.
Xanthopoulos, P., Pardalos, P. M., & Trafalis, T. B. (2012). *Robust data mining*. New York: Springer.
Xu, H., Caramanis, C., & Mannor, S. (2009). Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, *10*, 1485–1510.
Xu, H., Caramanis, C., & Mannor, S. (2010). Robust regression and lasso. *IEEE Transactions on Information Theory*, *56*(7), 3561–3574.