Global Optimization Methods for Extended Fisher Discriminant Analysis

Satoru Iwata
The University of Tokyo
iwata@mist.i.u-tokyo.ac.jp

Yuji Nakatsukasa
The University of Tokyo
nakatsukasa@mist.i.u-tokyo.ac.jp

Akiko Takeda
The University of Tokyo
takeda@mist.i.u-tokyo.ac.jp

Abstract

The Fisher discriminant analysis (FDA) is a common technique for binary classification. A parametrized extension, which we call the extended FDA, has been introduced from the viewpoint of robust optimization. In this work, we first give a new probabilistic interpretation of the extended FDA. We then develop algorithms for solving an optimization problem that arises from the extended FDA: computing the distance between a point and the surface of an ellipsoid. We solve this problem via the KKT points, which we show are obtained by solving a generalized eigenvalue problem. We speed up the algorithm by taking advantage of the matrix structure and proving that a globally optimal solution is a KKT point with the smallest Lagrange multiplier, which can be computed efficiently as the leftmost eigenvalue. Numerical experiments illustrate the efficiency and effectiveness of the extended FDA model combined with our algorithm.

1 Introduction

Various binary classification methods such as the Fisher discriminant analysis (FDA) have been extensively studied in machine learning and statistics. A recent work of Takeda et al. [19] introduced a unified framework for binary classification, which extends the range of hyper-parameters in several existing learning models. In particular, [19] briefly suggested a parametrized extension of FDA, which we call the extended FDA.

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

Our first contribution is a probabilistic interpretation of the extended FDA: It attempts to maximize the margin while ensuring that the worst-case probability of a correct prediction is bounded below. The interpretation implies that the extension of FDA has an effect of allowing the margin to be negative, which may suit well for noisy datasets where distributions of samples in each class are highly overlapping.

The extended FDA involves an optimization problem parametrized by $\kappa \geq 0$. By taking the dual, this problem is turned into computing the minimum distance between a given point and the surface of an ellipsoid obtained from the data. The parameter κ serves as a scale factor of the ellipsoid. When κ is sufficiently small, the given point stays outside the ellipsoid. In this case, computing the minimum distance reduces to a second order cone program (SOCP), which is a convex optimization problem that can be solved efficiently by the interior point method. For a certain value $\kappa = \kappa_0$, the given point hits the surface of the ellipsoid, in which case the extended FDA becomes equivalent to the original FDA. When $\kappa > \kappa_0$, the point stays inside the ellipsoid, and computing the minimum distance is essentially a non-convex optimization problem, which is in general hard to solve. It is nonetheless important to address such cases, because they correspond to the extended FDA that allows for negative margins.

A standard approach to non-convex optimization in practice is to design an efficient local search algorithm that finds a locally optimal solution. If we are lucky, the obtained solution may be globally optimal, or may perform just as well. It should be remarked, however, that there is no guarantee in this approach. An alternative approach is to design an exact algorithm based on the cutting-plane or branch-and-bound method [11]. This approach gives an optimal solution, but the computational cost may be enormous.

Our second contribution is the design of an efficient algorithm that finds a globally optimal solution for the extended FDA, convex or non-convex, by exploiting the structure of the problem. Specifically, we analyze

the Karush-Kuhn-Tucker (KKT) conditions and show that all the possible Lagrange multipliers that yield the KKT points are generalized eigenvalues of a certain matrix pencil. This allows us to use existing and reliable algorithms developed for the generalized eigenvalue problem.

Furthermore, we show that a globally optimal solution corresponds to the leftmost generalized eigenvalue, which is in fact real and can be obtained much faster than the set of all the generalized eigenvalues. Using these facts and by exploiting special properties of the matrix pencil, we develop an algorithm that runs as fast as the diagonalization of the symmetric matrix that defines the ellipsoid.

The point-ellipsoid distance is a well-studied problem (e.g., [13]). In particular, Eberly [7] gives a complete analysis for dimensions $n \leq 3$, but less complete for higher dimensions, for which no explanation is given to prove that the smallest eigenvalue (root) for an algebraic equation corresponds to the the solution. We examine the problem for arbitrary n, showing that the point-ellipsoid distance corresponds to the KKT point with the smallest Lagrange multiplier. Our discussion also allows the ellipsoid matrix to be singular.

Numerical experiments illustrate that the extended FDA combined with our proposed algorithm is much faster than the C-support vector machine (C-SVM) [6], while giving comparable prediction performance for most datasets. The inputs of the extended FDA are simply obtained from the covariance matrix and the mean vector of samples in each class. Crucially they do not depend on the sample size, which often leads to significant speedups. Indeed, our global optimization algorithm requires much less running time than local search algorithms [19], which iteratively solves convex relaxation problems. Our experiments using random matrices show that our algorithm can deal with the extended FDA with feature size as large as 10^4.

The rest of this paper is organized as follows. In Section 2, we review the binary classification problem and outline FDA and its extension. Section 3 derives a probabilistic interpretation for extended FDA. In Section 4, we develop our efficient algorithm for the dual (point-ellipsoid distance) problem, which includes analyses of the KKT points and the algorithm complexity. Numerical experiments are presented in Section 5.

2 Problem Setting

In this section, we describe the problem setting for binary classification. Let $\mathcal{X} \subset \mathbb{R}^n$ be the input domain and $\{+1, -1\}$ be the set of binary labels. The observed training samples are denoted by $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \{+1, -1\}$ for $i = 1, \ldots, m$. We estimate a decision function $f : \mathcal{X} \to \mathbb{R}$ from the training samples. After obtaining f, the label of a new input $\boldsymbol{x}$ is predicted by the classifier $h(\boldsymbol{x}) = \mathrm{sign}(f(\boldsymbol{x}))$, where sign means the sign function, i.e., $\mathrm{sign}(\xi) = 1$ if $\xi \geq 0$ and -1 otherwise. We shall focus on linear classifiers, i.e., $h(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{x}^\top \boldsymbol{w} + b)$, where $\boldsymbol{w}$ ($\in \mathbb{R}^n$) is a vector and b ($\in \mathbb{R}$) is a bias parameter. The discussions in this paper can be directly applied to kernel classifiers by following [16].

2.1 Fisher discriminant analysis (FDA)

Let $\bar{\boldsymbol{x}}_+$ (or $\bar{\boldsymbol{x}}_-$) be the n-dimensional sample mean vector and Σ_+ (or Σ_-) be the sample covariance matrix for each class. Let $A := \Sigma_+ + \Sigma_-$ and $\boldsymbol{c} := \bar{\boldsymbol{x}}_+ - \bar{\boldsymbol{x}}_-$. Suppose for the moment that the matrix A is nonsingular (we will deal with the singular case later in Section 4.5). In FDA, a discriminant hyperplane is computed from A and $\boldsymbol{c}$. The hyperplane is determined from the optimal solution $\boldsymbol{w}^*$ to the problem [8]:

$$\max_{\boldsymbol{w}} \frac{\boldsymbol{c}^\top \boldsymbol{w}}{\|A^{1/2}\boldsymbol{w}\|}. \tag{2.1}$$

The intuition behind (2.1) is to find a direction which maximizes the projected class means (the numerator) while minimizing the class variance in this direction (the denominator). An optimal solution of (2.1) can be obtained by solving a generalized eigenvalue problem: $(\boldsymbol{c}\boldsymbol{c}^\top)\boldsymbol{w} = \lambda A \boldsymbol{w}$.

After obtaining a solution $\boldsymbol{w}^*$ by FDA, we can determine the bias term b by assuming a probabilistic model, such as the normal distribution, for the observed samples. The empirical loss function can also be used for computing b as

$$\min_b \frac{1}{m} \sum_{i=1}^{m} I[y_i(\boldsymbol{x}_i^\top \boldsymbol{w}^* + b) < 0], \tag{2.2}$$

where I is the indicator function evaluating to 1 if the condition enclosed is true, and 0 otherwise.

2.2 Extended FDA

FDA can be extended to the following classification model [19]:

$$\max_{\boldsymbol{w}:\|\boldsymbol{w}\|^2=1} \min_{\boldsymbol{x}\in\mathcal{U}^\kappa} \boldsymbol{x}^\top \boldsymbol{w}, \tag{2.3}$$

where $\mathcal{U}^\kappa$ denotes the ellipsoid defined by

$$\mathcal{U}^\kappa = \{\boldsymbol{c} + A^{1/2}\boldsymbol{u} \mid \|\boldsymbol{u}\| \leq \kappa\} \tag{2.4}$$

with reference to a positive parameter κ and $\|\cdot\|$ denotes the Euclidean norm. The parameter κ controls

the size of the ellipsoid. When κ is large, $\mathcal{U}^\kappa$ tends to include $\mathbf{0}$ in its interior. We call (2.3) with $\mathcal{U}^\kappa$ of (2.4) the "extended FDA". Indeed, when $\mathcal{U}^{\kappa_0}$ contains $\mathbf{0}$ in its boundary, i.e., $\mathbf{0} \in \mathrm{bd}(\mathcal{U}^{\kappa_0})$, the classification model (2.3) with such $\mathcal{U}^{\kappa_0}$ coincides with FDA (see [19]). The optimization problem (2.3) can further express various learning models such as ν-SVM [18] and minimax probability machine [14] by setting $\mathcal{U}^\kappa$ appropriately.

The difficulty of solving (2.3) depends on whether $\mathbf{0} \in \mathcal{U}^\kappa$ or not. Specifically, when $\mathbf{0} \notin \mathcal{U}^\kappa$, the constraint $\|\boldsymbol{w}\|^2 = 1$ can be relaxed to $\|\boldsymbol{w}\|^2 \le 1$ without changing the optimality and (2.3) reduces to a convex problem. On the other hand, if $\mathbf{0}$ is in the interior of $\mathcal{U}^\kappa$, that is, $\mathbf{0} \in \mathrm{int}(\mathcal{U}^\kappa)$, then $\|\boldsymbol{w}\|^2 = 1$ is equivalent to $\|\boldsymbol{w}\|^2 \ge 1$ and (2.3) is a nonconvex problem. We call the former "convex extended FDA" and the latter "nonconvex extended FDA".

When $\mathbf{0} \notin \mathcal{U}^\kappa$, by taking the dual of the inner minimization in (2.3) and replacing the nonconvex constraint by $\|\boldsymbol{w}\|^2 \le 1$, the convex extended FDA reduces to a second order cone program (SOCP):

$$\min_{\boldsymbol{w}} \kappa\|A^{1/2}\boldsymbol{w}\| - \boldsymbol{c}^\top \boldsymbol{w} \quad \text{sub. to} \quad \|\boldsymbol{w}\|^2 \le 1. \tag{2.5}$$

By replacing the Euclidean norm $\|\boldsymbol{w}\|$ with the L_1-norm $\|\boldsymbol{w}\|_1$ for the constraint $\|\boldsymbol{w}\|^2 \le 1$, (2.5) becomes similar to a sparse feature selection model [2] based on FDA.

When $\mathbf{0} \in \mathrm{int}(\mathcal{U}^\kappa)$, nonconvex extended FDA is

$$\min_{\boldsymbol{w}} \kappa\|A^{1/2}\boldsymbol{w}\| - \boldsymbol{c}^\top \boldsymbol{w} \quad \text{sub. to} \quad \|\boldsymbol{w}\|^2 = 1. \tag{2.6}$$

A two-stage algorithm was proposed for (2.3) in [19]. At first, it solves a convex relaxation problem whose constraint is $\|\boldsymbol{w}\|^2 \le 1$. If the optimal solution $\boldsymbol{w}^*$ is not $\mathbf{0}$, then $\boldsymbol{w}^*$ is an optimal solution to (2.3). If $\boldsymbol{w}^* = \mathbf{0}$, then it applies a local search algorithm [17], which iteratively solves relaxation problems of (2.6) with the nonconvex constraint $\|\boldsymbol{w}\|^2 = 1$ replaced by a linearized constraint $\tilde{\boldsymbol{w}}_t{}^\top \boldsymbol{w} = 1$ at a feasible solution $\tilde{\boldsymbol{w}}_t$. The algorithm repeatedly solves SOCPs until it converges.

In Section 4, we design more efficient algorithms that work both in the convex and nonconvex cases.

3 Probabilistic Interpretation for Extended FDA

We give a probabilistic interpretation for (2.3). Here, suppose that $\boldsymbol{x}$ is a vector of random variables following a distribution with prescribed mean vector $\boldsymbol{c}$ and covariance matrix A, but are otherwise arbitrary. The notation $\boldsymbol{x} \sim (\boldsymbol{c}, A)$ refers to the class of distributions that have mean $\boldsymbol{c}$ and covariance A. The definitions of $\boldsymbol{c}$ and A imply that $\boldsymbol{x}$ represents the difference of random vectors of two classes, i.e., inputs $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ for class 1 and class -1.

We start with showing a probabilistic interpretation for convex extended FDA (2.5). To distinguish between the convex and nonconvex cases for (2.3), we can also use the optimal value of (2.6) instead of the set $\mathcal{U}^\kappa$ of (2.4). Note that the optimal value of extended FDA (2.6) is an increasing function of κ. As long as the optimal value is negative, (2.6) can be replaced by (2.5) without altering the problem. Then the extended FDA can be understood as a stochastic programming problem as follows.

Theorem 3.1. *The convex extended FDA* (2.5) *is equivalent to*

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 \quad \text{sub. to} \quad \inf_{\boldsymbol{x}\sim(\boldsymbol{c},A)} \Pr\{\boldsymbol{x}^\top \boldsymbol{w} \ge 1\} \ge \alpha, \tag{3.1}$$

where $\alpha = \frac{\kappa^2}{1+\kappa^2}$.

Proof. The probabilistic constraint in (3.1) is the same as

$$\sup_{\boldsymbol{x}\sim(\boldsymbol{c},A)} \Pr\{\boldsymbol{x}^\top \boldsymbol{w} \le 1\} \le 1 - \alpha,$$

which is transformed into

$$\boldsymbol{c}^\top \boldsymbol{w} \ge \sqrt{\tfrac{\alpha}{1-\alpha}}\|A^{1/2}\boldsymbol{w}\| + 1$$

when $\alpha > 0$ (see, e.g., [14, 2] for this equivalence). We can verify the equivalence between (2.5) and

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 \quad \text{sub. to} \quad \boldsymbol{c}^\top \boldsymbol{w} \ge \kappa\|A^{1/2}\boldsymbol{w}\| + 1 \tag{3.2}$$

in the following way. For an optimal solution $\boldsymbol{w}^*$ of (3.2), $\frac{\boldsymbol{w}^*}{\|\boldsymbol{w}^*\|}$ is optimal for (2.5); otherwise, we can find a better feasible solution for (3.2) than $\boldsymbol{w}^*$. This means that we obtain an optimal solution of (2.5) by solving (3.2). The inverse relation also holds, i.e., for an optimal solution $\widehat{\boldsymbol{w}}^*$ and optimal value $-\widehat{f}^*$ (which is assumed to be negative) of (2.5), $\frac{\widehat{\boldsymbol{w}}^*}{\widehat{f}^*}$ is optimal for (3.2); otherwise, we can find a feasible solution for (2.5) that achieves a smaller objective value than $-\widehat{f}^*$. We thus proved the equivalence between (2.5) and (3.2), which is also equivalent to (3.1). □

Theorem 3.1 suggests that the classifier maximizes the margin $(1/\|\boldsymbol{w}\|)$ under the condition that the worst-case probability of a correct prediction is bounded below by α.

In practice, $\boldsymbol{c}$ and A are obtained from the mean vectors and covariance matrices from the training samples. When the given dataset is almost linearly separable, $\boldsymbol{x}^\top \boldsymbol{w} = (\boldsymbol{x}_+ - \boldsymbol{x}_-)^\top \boldsymbol{w} \ge 1$ will hold with high

probability. Then the prediction accuracy probability α can be set to a large value. However, in general, (3.1) becomes infeasible for extremely large α.

Note that as α of (3.1) gets large, the corresponding κ in (2.5) also gets large. When κ is large enough so that $\mathbf{0} \in \mathcal{U}^\kappa$, our problem is nonconvex as in (2.6). We give a probabilistic interpretation for the nonconvex problem (2.6) with positive optimal value as follows.

Theorem 3.2. *The nonconvex extended FDA* (2.6) *is equivalent to*

$$\max_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 \quad \text{sub. to} \quad \inf_{\boldsymbol{x}\sim(\boldsymbol{c},A)} \Pr\{\boldsymbol{x}^\top \boldsymbol{w} \geq -1\} \geq \alpha, \tag{3.3}$$

where $\alpha = \frac{\kappa^2}{1+\kappa^2}$.

Proof. We can verify the equivalence between (2.6) with a positive optimal value and

$$\max_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 \quad \text{sub. to} \quad \boldsymbol{c}^\top \boldsymbol{w} \geq \kappa \|A^{1/2}\boldsymbol{w}\| - 1 \tag{3.4}$$

analogously to the proof of Theorem 3.1. We can give a probabilistic interpretation for (3.4) as (3.3) by following the discussion of [14]. □

Here $(\boldsymbol{x}_+ - \boldsymbol{x}_-)^\top \boldsymbol{w} \geq -1$ is expected to hold with high probability. This inequality is weakened from the condition $(\boldsymbol{x}_+ - \boldsymbol{x}_-)^\top \boldsymbol{w} \geq 1$ for the convex case by allowing the margin to be negative.

These probabilistic interpretations for both convex and nonconvex cases should be compared with the hard margin SVM:

$$\max_{\boldsymbol{w}} \min_{\boldsymbol{x}_+\in\mathcal{U}_+, \boldsymbol{x}_-\in\mathcal{U}_-} \frac{(\boldsymbol{x}_+ - \boldsymbol{x}_-)^\top \boldsymbol{w}}{\|\boldsymbol{w}\|},$$

where $\mathcal{U}_+$ and $\mathcal{U}_-$ are the convex hulls of training samples with labels $+1$ and -1, respectively. The hard margin SVM is usually defined for the linearly separable case, that is, $\min_{\boldsymbol{x}_+\in\mathcal{U}_+, \boldsymbol{x}_-\in\mathcal{U}_-} (\boldsymbol{x}_+ - \boldsymbol{x}_-)^\top \boldsymbol{w} > 0$. Then the hard margin SVM reduces to

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 \quad \text{sub. to} \quad \boldsymbol{x}^\top \boldsymbol{w} \geq 1, \quad \forall \boldsymbol{x} \in \mathcal{U},$$

using the Minkowski difference $\mathcal{U}$ of $\mathcal{U}_+$ and $\mathcal{U}_-$, i.e., $\mathcal{U} = \{\boldsymbol{x}_+ - \boldsymbol{x}_- \mid \boldsymbol{x}_+ \in \mathcal{U}_+, \boldsymbol{x}_- \in \mathcal{U}_-\}$. This problem is closely related to (3.1). On the other hand, we can think of the extended version for nonlinearly separable datasets by allowing $\min_{\boldsymbol{x}_+\in\mathcal{U}_+, \boldsymbol{x}_-\in\mathcal{U}_-} (\boldsymbol{x}_+ - \boldsymbol{x}_-)^\top \boldsymbol{w} < 0$. Then the extended hard margin SVM reduces to

$$\max_{\boldsymbol{w}} \|\boldsymbol{w}\|^2 \quad \text{sub. to} \quad \boldsymbol{x}^\top \boldsymbol{w} \geq -1, \quad \forall \boldsymbol{x} \in \mathcal{U}.$$

This is related to (3.3).

4 Global Optimization Methods via Generalized Eigenvalues

We consider a dual problem for (2.3) and provide algorithms that are more efficient than existing ones, e.g., two-stage algorithm [19], especially in the nonconvex case. In contrast to existing methods, our algorithm solves the problem (2.3) regardless of whether it is convex or nonconvex.

The following is the dual problem of (2.3) due to Briec [4] (see [19]):

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\| \quad \text{sub. to} \quad \boldsymbol{x} \in \mathrm{bd}(\mathcal{U}^\kappa). \tag{4.1}$$

This asks the nearest distance between a point (the origin) and points on the surface of an ellipsoid, i.e., the point-ellipsoid distance. The set $\mathrm{bd}(\mathcal{U}^\kappa)$ can be expressed by one nonconvex constraint $(\boldsymbol{x}-\boldsymbol{c})^\top A^{-1}(\boldsymbol{x}-\boldsymbol{c}) = \kappa^2$, where A is an $n \times n$ symmetric positive definite matrix. Fortunately there is no duality gap here and the optimal solution $\boldsymbol{w}^*$ of (2.3) can be obtained from the optimal solution $\boldsymbol{x}^*$ of (4.1) by

$$\boldsymbol{w}^* = \frac{A^{-1}(\boldsymbol{c} - \boldsymbol{x}^*)}{\|A^{-1}(\boldsymbol{c} - \boldsymbol{x}^*)\|}.$$

This can be simplified as $\boldsymbol{w}^* = \boldsymbol{x}^*/\|\boldsymbol{x}^*\|$ if $\mathbf{0} \notin \mathcal{U}^\kappa$ and as $\boldsymbol{w}^* = -\boldsymbol{x}^*/\|\boldsymbol{x}^*\|$ if $\mathbf{0} \in \mathrm{int}(\mathcal{U}^\kappa)$. Below we discuss algorithms for solving (4.1).

4.1 Finding all the KKT points

We introduce the Lagrange multiplier λ and attempt to find $\boldsymbol{x} \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ that satisfy

$$(\boldsymbol{x} - \boldsymbol{c})^\top A^{-1}(\boldsymbol{x} - \boldsymbol{c}) = \kappa^2, \tag{4.2}$$

$$A\boldsymbol{x} = \lambda(\boldsymbol{x} - \boldsymbol{c}). \tag{4.3}$$

The first condition (4.2) requires that $\boldsymbol{x}$ is on the surface of the ellipsoid, and (4.3) represents the KKT condition. A feasible solution $\boldsymbol{x}$ is called a KKT point if it satisfies (4.3) for some λ. Note that (4.3) can be rephrased as

$$(\lambda I - A)\boldsymbol{x} = \lambda \boldsymbol{c}. \tag{4.4}$$

In order to find λ that admits $\boldsymbol{x}$ satisfying these conditions, we now introduce a $(2n+1) \times (2n+1)$ matrix pencil whose eigenvalues give the KKT points:

$$M(s) = \begin{bmatrix} \kappa^2 & O & \boldsymbol{c}^\top \\ O & -A & sI - A \\ \boldsymbol{c} & sI - A & O \end{bmatrix}. \tag{4.5}$$

Note that the entries of $M(s)$ are affine functions in s. Since $M(s)$ is clearly a regular matrix pencil, i.e., $\det M(s) \neq 0$ as a polynomial in s, it has $2n+1$ eigenvalues, including one at infinity. The eigenvalues of $M(s)$ are the candidate values of λ:

Lemma 4.1. *If a Lagrange multiplier λ admits a KKT point, then* $\det M(\lambda) = 0$ *holds.*

Proof. By the condition (4.4), $\boldsymbol{c}$ must belong to $\mathrm{Im}\,(\lambda I - A)$. If λ is an eigenvalue of A, this implies $\mathrm{rank}\,[\boldsymbol{c} \quad \lambda I - A] < n$, and hence $M(\lambda)$ is singular.

We now suppose that λ is not an eigenvalue of A. Then applying Gaussian elimination to $M(\lambda)$ we have

$$\begin{aligned}\det M(\lambda) &= \det \begin{bmatrix} \kappa^2 - h(\lambda)^\top A h(\lambda) & h(\lambda)^\top A & O \\ A h(\lambda) & -A & \lambda I - A \\ O & \lambda I - A & O \end{bmatrix} \\ &= (-1)^n \det(\lambda I - A)^2 \{\kappa^2 - h(\lambda)^\top A h(\lambda)\},\end{aligned}$$

where $h(\lambda) = (\lambda I - A)^{-1}\boldsymbol{c}$. The conditions (4.3) and (4.4) imply that $h(\lambda) = A^{-1}(\boldsymbol{x} - \boldsymbol{c})$. Since $\boldsymbol{x}$ satisfies (4.2) as well, we have $\kappa^2 - h(\lambda)^\top A h(\lambda) = 0$, which means $\det M(\lambda) = 0$. □

Thus we can compute all possible λ by solving the generalized eigenvalue problem $\det M(\lambda) = 0$, which yields $2n$ finite (including complex) candidates of λ.

Once the generalized eigenvalue problem is solved, we can obtain the KKT point $\boldsymbol{x}$ from the corresponding eigenvector $\boldsymbol{v} = \begin{bmatrix}\theta & \boldsymbol{y}^\top & \boldsymbol{z}^\top\end{bmatrix}^\top$ for each λ as follows.

If $\theta \neq 0$, putting $\boldsymbol{x} := -\frac{\lambda}{\theta}\boldsymbol{y}$ provides a solution to (4.4). If λ is not an eigenvalue of A, then this solution $\boldsymbol{x}$ satisfies (4.2) as well. If λ is an eigenvalue of A, then the system of linear equations (4.4) is underdetermined. In fact, $\boldsymbol{x} := -\frac{\lambda}{\theta}\boldsymbol{y} + \boldsymbol{u}$ satisfies (4.4) for any eigenvector $\boldsymbol{u}$ of A corresponding to λ. In this case, among all these solutions to (4.4), we find those that satisfy (4.2) as well. Such $\boldsymbol{x}$ may not exist (see (A.4)), in which case there is no KKT point at λ.

We now consider the case $\theta = 0$. If $\boldsymbol{y} \neq 0$, then λ must be an eigenvalue of A and $\boldsymbol{y}$ must be a corresponding eigenvector, which imply $\boldsymbol{y} \notin \mathrm{Im}\,(\lambda I - A)$. The middle block of $M(\lambda)\boldsymbol{v} = 0$, however, shows that $\lambda\boldsymbol{y} = (\lambda I - A)\boldsymbol{z}$, which is a contradiction. Thus $\boldsymbol{y} = 0$ must hold. Then $\boldsymbol{z}$ is an eigenvector of A corresponding to λ. In this case, we solve the system of linear equations (4.4), which is underdetermined. Among all the solutions, we find those that satisfy (4.2) as well.

4.2 Simplification via eigendecomposition

Let $A = QDQ^\top$ be the eigenvalue decomposition of A, where Q is orthogonal, i.e., $Q^\top Q = QQ^\top = I$, and D is the diagonal matrix of eigenvalues. With the aid of $Q_0 := \mathrm{diag}\{1, Q, Q\}$, we have

$$\tilde{M}(s) := Q_0^\top M(s) Q_0 = \begin{bmatrix} \kappa^2 & O & \tilde{\boldsymbol{c}}^\top \\ O & -D & sI - D \\ \tilde{\boldsymbol{c}} & sI - D & O \end{bmatrix}, \tag{4.6}$$

where $\tilde{\boldsymbol{c}} := Q^\top \boldsymbol{c}$. Note that $\tilde{M}(s)$ is highly sparse, which we take advantage of in our algorithm. Denote the ith entry of $\tilde{\boldsymbol{c}}$ by $\tilde{\boldsymbol{c}}_i$. We also denote the ith diagonal entry of D by d_i. Then we have

$$\det \tilde{M}(s) = \kappa^2 \prod_{i=1}^n (s - d_i)^2 - \sum_{i=1}^n \left(d_i \tilde{c}_i^2 \prod_{j \neq i}^n (s - d_j)^2 \right).$$

Since $\det \tilde{M}(s) = \det M(s)$, the generalized eigenvalue problem is equivalent to the algebraic equation

$$\kappa^2 \prod_{i=1}^n (s - d_i)^2 - \sum_{i=1}^n \left(d_i \tilde{c}_i^2 \prod_{j \neq i}^n (s - d_j)^2 \right) = 0 \tag{4.7}$$

for s. This can be solved in a number of ways, including linearization to the companion matrix (e.g., [9]) and the Ehrlich-Aberth method [1]. However, solving algebraic equations is known to be possibly numerically unstable. For $s \neq d_i$, (4.7) is equivalent to

$$\kappa^2 - \sum_{i=1}^n \frac{d_i \tilde{c}_i^2}{(s - d_i)^2} = 0. \tag{4.8}$$

In Section 4.1, we have described how to obtain $\boldsymbol{x}$ from the eigenpair $(\lambda, \boldsymbol{v})$ of $M(s)$. Since in practice computing the whole eigendecomposition is much more expensive than computing just the eigenvalues, we provide another approach for computing the KKT points $\boldsymbol{x}$ from the Lagrange multipliers λ via (4.4). For each λ, solving (4.4) for $\boldsymbol{x}$ requires a solution of a linear system. If a dense linear solver is invoked from scratch, each solution takes $O(n^3)$ operations, resulting in a total of $O(n^4)$ operations. However, cost saving is possible using the eigendecomposition $A = QDQ^\top$ that we have already computed, as we can compute $\boldsymbol{x}$ directly as

$$\boldsymbol{x} = \lambda Q(\lambda I - D)^{-1} Q^\top \boldsymbol{c}. \tag{4.9}$$

The resulting $\boldsymbol{x}$ naturally satisfies (4.2) as well. Since each $\boldsymbol{x}$ can be computed via matrix-vector multiplications in just $O(n^2)$ operations, finding all the corresponding KKT points is reduced to $O(n^3)$ flops.

Separately from the solutions of (4.8), we also examine eigenvalues of $M(s)$ equal to some d_i, and check if they lead to a KKT point. By (4.7), this can happen only if $\tilde{c}_j = 0$ for all j such that $d_i = d_j$. In this case, the system of linear equations (4.4) is solvable and underdetermined. Among all the solutions, we adopt those that satisfy (4.2) as well. More specifically, we assign

$$\tilde{x}_j := \frac{\tilde{c}_j d_j}{d_i - d_j} \quad \text{for } d_j \neq d_i. \tag{4.10}$$

If $\sum_{j, d_j \neq d_i} \frac{(\tilde{x}_j - \tilde{c}_j)^2}{d_j} > \kappa^2$, then we may assert that no feasible solution satisfies the KKT condition with

$\lambda = d_i$. Otherwise, set $\tilde{x}_i$ such that

$$\sum_{j,d_i=d_j} \tilde{x}_j^2 = d_i\big(\kappa^2 - \sum_{j\neq i} \frac{(\tilde{x}_j - \tilde{c}_j)^2}{d_j}\big), \tag{4.11}$$

and the resulting vector $\tilde{\boldsymbol{x}}$ yields a KKT point $\boldsymbol{x} = Q\tilde{\boldsymbol{x}}$. Note that the choice of $\tilde{x}_j$ is not unique, showing there are many points on the surface of the ellipsoid with the same minimum $\|\boldsymbol{x}\|$. This is especially true when there are many different j in the summand of (4.11). See the appendix for an example where this happens.

4.3 A globally optimal solution from the smallest Lagrange multiplier

As we have seen above, there can be as many as $2n$ real values of λ that satisfy (4.2) and (4.3). Among these candidates, we show that an optimal solution of the optimization problem (4.1) corresponds to the smallest real λ that satisfies the KKT conditions.

Eberly [7] implicitly proves this result for $n = 2$, and uses it also for $n \geq 3$ but without a proof. Below we give a proof for arbitrary n. We start with a claim that holds in the generic case $\tilde{c}_n \neq 0$.

Lemma 4.2. *If $\tilde{c}_n \neq 0$, then the solution of* (4.1) *corresponds to the KKT point with Lagrange multiplier $\lambda < d_n$.*

See the appendix for the proof. We next consider the nongeneric case where $\tilde{c}_n = 0$, or more generally $|\tilde{c}_k| > \tilde{c}_{k+1} = \cdots = \tilde{c}_n = 0$, and show that the minimal $\|\boldsymbol{x}\|$ is still attained at the smallest λ that satisfies the KKT conditions.

Theorem 4.1. *The solution of* (4.1) *corresponds to the KKT point with the smallest Lagrange multiplier λ.*

See the appendix for the proof, which shows that if a valid KKT point with $\lambda = d_{k+j}$ exists, then the point with the largest j (smallest d_{k+j}) is a globally optimal solution. However, in the convex case, where the origin is outside the ellipsoid, there is no KKT point with $\lambda = d_{k+j}$. This is because it follows from $\kappa^2 - \sum_{i=1}^n \tilde{c}_i^2/d_i < 0$ that (4.8) has a negative solution.

Eberly [7] implicitly uses Theorem 4.1 to derive a case-dependent method (e.g. depending on whether $\lambda = d_{k+j}$ satisfies the KKT conditions) for the point-ellipsoid distance. Our algorithm does not need such special care as it can find all the KKT points, and the solution is directly obtained from the smallest λ.

4.4 The leftmost eigenvalue is real

We have just seen that to solve the minimization problem (4.1) we generically need to compute the smallest real eigenvalue λ_* of $M(s)$ in (4.5). Here we show that λ_* is in fact the leftmost eigenvalue, i.e., the eigenvalue having the smallest real part (excluding the eigenvalue at infinity).

Proposition 4.2. *The leftmost eigenvalue of $M(s)$ is real.*

See the appendix for the proof. Theorem 4.1 and Proposition 4.2 suggest that to obtain the desired solution we can invoke any algorithm for sparse generalized eigenproblems that computes the leftmost eigenpair. Known algorithms suitable for this task exist, for example the shift-invert Arnoldi process [15]. Recall that $\tilde{M}(s) = Q_0^\top M(s) Q_0$ as in (4.6) is highly sparse. The shift-invert Arnoldi process requires solving linear systems of the form $M(\lambda)\boldsymbol{v} = \boldsymbol{b}$, which may be efficiently solved via solving $\tilde{M}(\lambda)\tilde{\boldsymbol{v}} = \tilde{\boldsymbol{b}}$, where $\tilde{\boldsymbol{b}} = Q_0^\top \boldsymbol{b}$ and $\tilde{\boldsymbol{v}} = Q_0^\top \boldsymbol{v}$, using a sparse direct solver exploiting the sparsity. ARPACK provides this option, and for example, MATLAB's `eigs` with the option `'sr'` does this task. This dramatically reduces the cost compared with computing all the eigenvalues of $M(\lambda)$.

4.5 When the covariance matrix A is singular

Throughout the above discussion we assumed that A is nonsingular. In some cases A can be singular or highly ill-conditioned. Here we discuss how to solve (4.1) when A^{-1} does not exist.

The point-ellipse distance (4.1) is still well-posed, but the points on the boundary of the ellipse $\mathcal{U}^\kappa = \{\boldsymbol{c} + A^{1/2}\boldsymbol{u} \mid \|\boldsymbol{u}\| = \kappa\}$ now has no component in the null space of A.

This can be dealt with by simply forcing $\tilde{x}_i = \tilde{c}_i$ for the corresponding coordinates i with $d_i = 0$. Then we remove the rows and columns corresponding to $d_i = 0$ in $\tilde{M}(s)$, resulting in a smaller matrix pencil, shrunk by twice the number of the zero eigenvalues of A.

4.6 Algorithm summary and complexity

Below are pseudocodes for solving (4.1) in two cases: when only the globally optimal solution is to be computed, and when all the KKT points are required.

Algorithm 4.1 Find a globally optimal solution.

1: Compute the eigendecomposition $A = QDQ^\top$.
2: Compute the leftmost eigenpair $(\lambda, \boldsymbol{v})$ of $\tilde{M}(s)$.
3: Compute $\boldsymbol{x} := -\frac{\lambda}{\theta}\boldsymbol{y}$ if $\theta \neq 0$. Otherwise set $\tilde{x}_i$ via (4.10) and (4.11) if (4.11) is positive, if not compute the next leftmost eigenpair and repeat.

The dominant cost of the above process is in Step 1

where the eigenvalue decomposition of a symmetric matrix is computed, whose cost is about $9n^3$ flops.

We note that Eberly [7] uses a bisection method for computing just the smallest λ that satisfies (4.8). Assuming that bisection finds a good estimate in a constant number of steps, we may assert that finding λ from (4.8) by the bisection step needs $O(n)$ operations. Since the eigenvalue decomposition of A is still necessary, the overall cost is again $9n^3$.

Algorithm 4.2 Enumerate all the KKT points.

1: Compute the eigendecomposition $A = QDQ^\top$.
2: Compute all the eigenvalues λ of $\tilde{M}(s)$.
3: For each real λ, find the corresponding x via (4.9), or (4.10) and (4.11) if (4.11) is positive, if not there is no KKT point at that λ.

The main cost of Algorithm 4.2 is in Step 2, which solves an $2n \times 2n$ generalized eigenvalue problem by the QZ algorithm [10], which requires about $30 \times (2n)^3 = 240n^3$ flops. This is roughly 30 times as expensive as Algorithm 4.1. Note that $\tilde{M}(\lambda)$ is symmetric (although indefinite), so an algorithm that exploits symmetry (which the standard QZ algorithm unfortunately does not) may solve (4.6) faster.

In Algorithm 4.1 we compute $\boldsymbol{x}$ from the eigenvector because `eigs` finds the leftmost eigenpair, including the eigenvector. In contrast, in Algorithm 4.2, it is more efficient to compute just the eigenvalues of $\tilde{M}(\lambda)$ and then find $\boldsymbol{x}$ from (4.9).

To compare with previous algorithms, each iteration of the algorithm in [19] uses the interior point method, which requires at least $O(n^3)$ flops as it solves linear systems of size n repeatedly. Thus our global optimization method is at least as fast as the previous local algorithm, and indeed faster in experiments.

5 Numerical Experiments

Below we present numerical experiments to illustrate the performance of our algorithms and the effectiveness of the extended FDA. We provide three sets of experiments: Computing the point-ellipsoid distance, binary classification with synthetic datasets, and datasets from LIBSVM [5] and the UCI repository [3]. All of our computations were run using MATLAB (R2011b) on a MacBook Pro with 8 GB of RAM and 2.3 GHz Intel Core i7. Our codes are posted at [12].

5.1 Scalability test with random matrices

To examine the performance of Algorithm 4.1 for computing the point-ellipsoid distance in large dimensions,

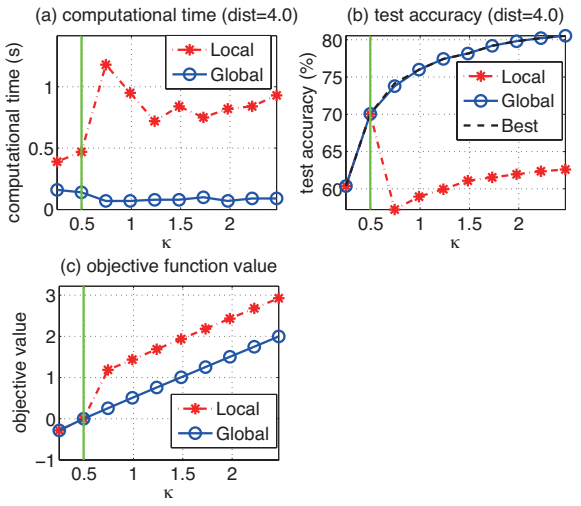

Figure 1: Computational time, test accuracy and the objective function value of (4.1) with respect to parameter κ. The vertical line indicates the threshold κ_0, which corresponds to FDA, and provides the border between convex and nonconvex extended FDA.

we take $A, \boldsymbol{c}$ to be $n \times n$ random matrices and vectors and test the runtime.

n	runtime(s)	runtime(`eig(A)`)/runtime
2000	3.99	0.944
4000	36.43	0.981
6000	115.14	0.986
8000	263.05	0.989
10000	504.54	0.991

Our algorithm can handle n as large as 10^4 on a standard laptop machine, or more if the eigendecomposition of A is available. The dominant runtime is consumed in computing the eigendecomposition of A, and since this step is necessary also in the approach by Eberly, we conclude that our method is nearly the best known in terms of speed.

5.2 Synthetic datasets

Now we show how to generate synthetic data sets. We supposed that the conditional probabilities, $p(\boldsymbol{x}|+1)$ and $p(\boldsymbol{x}|-1)$, were each multivariate normal distributions. The dimension of the input vector $\boldsymbol{x}$ was set to $n = 100$. The mean vector and the variance-covariance matrix of $p(\boldsymbol{x}|+1)$ were defined by the null vector $(0, \ldots, 0)^\top \in \mathbb{R}^n$ and the identity matrix $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$, respectively (i.e., the probability distribution was a multivariate standard normal distribution). For $p(\boldsymbol{x}|-1)$, the variance-covariance matrix was randomly generated so that the eigenvalues were numbers placed at even intervals from 10^{-4} to 10^4. The mean vector of $p(\boldsymbol{x}|-1)$ was defined by $\frac{4}{\sqrt{n}}(1, \ldots, 1)^\top \in \mathbb{R}^n$. Note that the distance between the mean vectors of

Table 1: Performance comparison for UCI datasets

Dataset	n	m	Global		Local		C-SVM	
			acc. [%]	time [s]	acc. [%]	time [s]	acc. [%]	time [s]
fourclass	2	862	**77.96**	0.0086	**77.96**	0.1540	76.22	0.0686
liver-disorders	6	345	**68.69**	0.0134	**68.69**	0.1890	68.41	0.0232
diabetes	8	768	**76.97**	0.0160	76.84	0.2506	76.71	0.1248
breast cancer	10	683	**96.92**	0.0178	**96.92**	0.1812	96.63	0.0584
heart	13	270	82.22	0.0214	82.22	0.2712	**83.33**	0.1886
australian	14	690	**85.80**	0.0258	**85.80**	0.3156	85.51	1.4274
german.numer	24	1000	76.90	0.0934	76.90	0.5670	**78.00**	2.8648
splice	60	1000	79.90	0.0682	79.90	0.8740	**80.10**	10.1138
mushrooms	112	8124	**100.00**	0.5592	-	-	**100.00**	0.5274
a1a	119	1605	**82.74**	0.1016	-	-	82.55	4.6360
w1a	300	2477	**97.78**	0.2764	-	-	96.93	0.2974

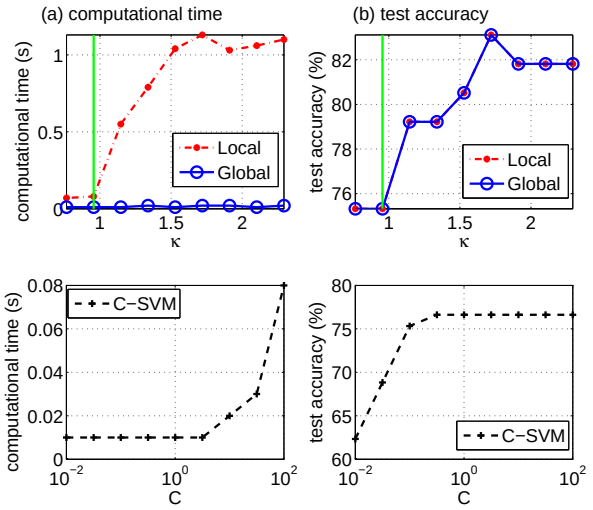

Figure 2: Computational time and test accuracy of the extended FDA (4.1) with respect to parameter κ and those of C-SVM with respect to parameter C for `diabetes`. The vertical line in upper panels indicates the threshold κ_0, which corresponds to FDA. (4.1) reduces to a convex problem on the left-hand side and it remains a nonconvex problem on the right-hand side.

$p(\boldsymbol{x}|+1)$ and $p(\boldsymbol{x}|-1)$ was equal to 4. We assume that the marginal probabilities of the positive and negative labels are the same. The training sample size and test sample size were set to $m = 10^4$ and $\widetilde{m} = 10^4$, respectively. We used sample covariance matrices (Σ_+, Σ_-) and mean vectors $(\bar{\boldsymbol{x}}_+, \bar{\boldsymbol{x}}_-)$ that were computed from training samples of each class as input data for our model. Therefore, the complexity of our algorithm is irrelevant to the sample size.

Figure 1 compares the performance of two algorithms, the local search algorithm [19] (shown as "Local") and our algorithm (shown as "Global"), in terms of computational time and test accuracy. The bias of each classifier was computed by (2.2). "Local" finds a locally optimal solution for (2.6) by repeatedly approximating $\|\boldsymbol{w}\|^2 = 1$ with a linear supporting function. "Best" shows the highest test accuracy among all locally optimal solutions found by enumerating KKT points. Our method can find a globally optimal solution, which leads to better test accuracy than the locally optimal solution found by "Local" and almost equal test accuracy to "Best". Because of large scale dataset ($n = 100$, $m = 10^4$), C-SVM [6] and ν-SVM [18] of LIBSVM [5] could not find optimal solutions within one hour.

5.3 UCI datasets

We present several UCI datasets showing the effectiveness of our proposed method. Table 1 shows the average test accuracy and average computational time which are evaluated by the 10-fold cross validation. We found the best parameter setting from 5 candidates: $\kappa \in \{6, 7, 8, 9, 10\}/8 \times \kappa_0$ using the threshold of convexity, κ_0, for our model (4.1) and $C \in \{10^{-1}, 1, 10, 10^2, 10^3\}$ for C-SVM [6]. The first two candidates for κ reduce (4.1) to convex problems, while (4.1) is essentially nonconvex for the last two candidates. The last three datasets in Table 1 have singular matrices A, which prevent the local search algorithm [19] from converging. Figure 2 compares the performance of three methods (Local, Global and C-SVM) for `diabetes`. The training data consists of randomly chosen samples with size $9m/10$. The test accuracy was evaluated by using the remaining $m/10$ samples. From Table 1 and Figure 2, we can confirm that our model (4.1) achieved comparable prediction performance while being faster than C-SVM for most of the datasets.

References

[1] O. Aberth. Iteration methods for finding all zeros of a polynomial simultaneously. *Math. Comp.*, 27(122):339–344, 1973.

[2] C. Bhattacharyya. Second order cone programming formulations for feature selection. *Journal of Machine Learning Research*, 5:1417–1433, 2004.

[3] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[4] W. Briec. Minimum distance to the complement of a convex set: duality result. *Journal of Optimization Theory and Applications*, 93(2):301–319, 1997.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[7] D. Eberly. Distance from a point to an ellipse, an ellipsoid, or a hyperellipsoid. Available at http://www.geometrictools.com/Documentation/DistancePointToEllipsoid.pdf, 2011.

[8] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[9] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. SIAM, Philadelphia, USA, 2009. Unabridged republication of book first published by Academic Press in 1982.

[10] G. H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 4th edition, 2012.

[11] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, Berlin, third edition, 1995.

[12] S. Iwata, Y. Nakatsukasa, and A. Takeda. Extended Fisher Discriminant Analysis and Point-Ellipsoid Distance. MATLAB File Exchange, http://www.mathworks.com/matlabcentral/file exchange/45364.

[13] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal toolbox. Technical Report EECS-2006-46, EECS, UC Berkeley, 2006.

[14] 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.

[15] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicityly Restorted Arnoldi Methods*, volume 6. SIAM, 1998.

[16] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.R. Müller. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):623–628, 2003.

[17] F. Perez-Cruz, J. Weston, D. J. L. Hermann, and B. Schölkopf. Extension of the ν-SVM range for classification. In *Advances in Learning Theory: Methods, Models and Applications*, pages 179–196, Amsterdam, 2003. IOS Press.

[18] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[19] A. Takeda, H. Mitsugi, and T. Kanamori. A unified classification model based on robust optimization. *Neural Computation*, 25(3):759–804, 2013.