Optimization Methods and Software
Vol. 00, No. 00, Month 200x, 1–12

Linear and nonlinear discrimination via the analytic center cutting plane method

Olivier Péton [†] & Nidhi Sawhney [‡] & Jean–Philippe Vial [‡]

(Received 00 Month 200x; In final form 00 Month 200x)

In this paper we address the problem of discriminating data belonging to different classes. We examine both linear and quadratic formulations of the discrimination function in order to classify data instances into two classes. The problem is modelled as the unconstrained minimization of a non-differentiable function, and solved by using a homogeneous version of the Analytic Center Cutting Plane Method. We give numerical results on some classical benchmark problems from the machine learning literature.

Keywords: linear and nonlinear separation; machine learning; mathematical programming; ACCPM.

1 Introduction

As pointed out by Bradley et al. in [1], Data Mining and Knowledge Discovery in Databases has interesting applications for several disciplines, which include statistics, databases, pattern recognition, artificial intelligence, optimization, visualization, high performance and parallel computing. One classical problem is that of predictive modelling, where the goal is to determine relationships between independent attributes and a designated dependent attribute or outcome class. Concrete applications are reported in many different areas, for e.g. cancer diagnosis, human genome construction, pattern recognition, bank/insurance, and more. The problem of predictive modelling, in turn gives rise to the problem of determining a discrimination function which can be used as a classifier to separate data based on the outcome class.

A reasonable measure of efficiency of such a classifier (or separator) is the number of misclassified instances. Unfortunately, minimizing this measure turns out to be an NP-complete problem [2]. As a surrogate to this approach, it has been proposed to use a continuous measure of misclassification [1] and rely on powerful convex optimization schemes to compute a

[†] École des Mines de Nantes, 4 rue Alfred Kastler, F–44307 Nantes Cedex 3, France, olivier.peton@emn.fr
[‡] Hec/Logilab, University of Geneva, 40 bd du Pont d'Arve, CH–1211 Geneva 4, Switzerland, nidhi.sawhney@hec.unige.ch, vial@hec.unige.ch

separator. A linear programming formulation for the discrimination of two classes was proposed by Mangasarian [3] as early as 1965. Subsequent works proposed alternative formulations and variants. In this paper, we choose to adopt the same error-minimization optimization model as in [4] for the discrimination problem with two classes. The objective function of our problem is the weighted sum of a continuous misclassification measure. This function is convex and non-differentiable. The problem is thus a continuous optimization one.

Mangasarian [5] proposes a linear program with equilibrium constraint (LPEC) in order to minimize the number of misclassified instances. In this work, we prefer to keep the unconstrained formulation, considering it as a continuous relaxation of the discrete problem. At each iteration of our algorithm, we calculate the number of misclassified instances. The solution with the least number of misclassified error is kept as the best solution. Hence, the best solution is not necessarily found at the end of the iterative procedure, even though the continuous misclassification measure is roughly monotone decreasing.

The paper is organized as follows. Section 2 introduces the problem formulation and its properties. Section 3 is devoted to the minimization algorithm. We use a homogeneous version of the Analytic Center Cutting Plane Method (ACCPM) [6], due to Nesterov and Vial [7] and implemented in [8]. This homogeneous version is referred to as h-ACCPM. We briefly recall the main steps of the algorithm, and explain why it is particularly well fitted for the discrimination problem. Section 4 presents our numerical experiments on classical benchmark problems taken from the UCI Machine Learning Repository [9].

2 The two-category discrimination problem

Let us first address the linear discrimination problem with two categories (or classes). This problem caught the attention of researchers in the 1960s, and has seen renewed interest in the early 1990s with the explosive growth in the use of databases.

We consider a set of instances (or points) $\mathcal{A} = \{\boldsymbol{a}_i \in \mathbb{R}^n, i = 1, 2, \cdots, m\}$, defined by n numerical attributes (or features). Each instance also has binary label (positive or negative) associated with it, partitioning the set into two categories.

The separation problem consists of determining a discriminating function $f : \mathbb{R}^n \to \mathbb{R}$ such that $f(\boldsymbol{a}_i) > 0$ for all points $\boldsymbol{a}_i$ with positive label and $f(\boldsymbol{a}_i) < 0$ for all points $\boldsymbol{a}_i$ with negative label. The underlying objective is to use this separation function for correctly predicting the class of previously unseen instances.

2.1 *Continuous formulation of the linear separation problem*

Consider the partition $\mathcal{A}_1 \cup \mathcal{A}_2$ of the set $\mathcal{A}$, we wish to find a vector $\boldsymbol{\omega} \in \mathbb{R}^n$ and a scalar $\gamma \in \mathbb{R}$ such that the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n \ : \ \boldsymbol{\omega}^T\boldsymbol{x} = \gamma\}$ separates instances from the two classes $\mathcal{A}_1$ and $\mathcal{A}_2$ as correctly as possible. For typographical convenience, we write $(\boldsymbol{\omega}, \gamma)$ instead of $\binom{\boldsymbol{\omega}}{\gamma}$.

Let $\mathbf{A}_1$ and $\mathbf{A}_2$ denote the matrices of points in $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. Any separating hyperplane such that

$$\mathbf{A}_1\boldsymbol{\omega} > \boldsymbol{e}\gamma, \tag{1}$$

$$\mathbf{A}_2\boldsymbol{\omega} < \boldsymbol{e}\gamma, \tag{2}$$

where $\boldsymbol{e}$ is a vector of ones of appropriate dimension, is called a separating hyperplane. Upon normalization, this system is equivalent to

$$\mathbf{A}_1\boldsymbol{\omega} - \boldsymbol{e}(\gamma + \nu) \geq 0, \tag{3}$$

$$-\mathbf{A}_2\boldsymbol{\omega} + \boldsymbol{e}(\gamma - \nu) \geq 0, \tag{4}$$

where the value ν is called the separation margin.

The system (3)-(4) can be satisfied if and only if the classes $\mathcal{A}_1$ and $\mathcal{A}_2$ are separable. In such a case, there exist an infinite number of hyperplanes that separate the two classes; then one looks for the hyperplane that separates the two classes with the largest margin ν. A composite function that involves $\boldsymbol{\omega}, \gamma$ and ν can also be used.

In the general case where it may not be possible to find a hyperplane with the required separation property, we measure the misclassification errors in the following manner, let

$$\begin{aligned} \boldsymbol{e}_i^1 &= \max(-\boldsymbol{\omega}^T\boldsymbol{a}_i + \gamma + \nu, 0), \ \forall \boldsymbol{a}_i \in \mathcal{A}_1, \\ \boldsymbol{e}_i^2 &= \ \max(\boldsymbol{\omega}^T\boldsymbol{a}_i - \gamma + \nu, 0), \ \ \forall \boldsymbol{a}_i \in \mathcal{A}_2. \end{aligned}$$

The linear separation problem can be formulated as the following minimization problem in $\mathbb{R}^{n+1}$:

$$\min_{(\boldsymbol{\omega},\gamma)\in\mathbb{R}^n\times\mathbb{R}} F(\boldsymbol{\omega}, \gamma) = \frac{1}{|\mathcal{A}_1|}\sum_i \boldsymbol{e}_i^1 + \frac{1}{|\mathcal{A}_2|}\sum_i \boldsymbol{e}_i^2 \tag{5}$$

Problem (5) is convex but not differentiable. It is homogeneous of degree 0 in ($\boldsymbol{\omega}$, γ). Hence, any positive value of ν yields the same misclassification errors, after an appropriate scaling of $\boldsymbol{\omega}$ and γ.

The discrimination may generate two types or errors: negative instances classified as positive, and positive instances classified as negative. In some cases (for example medical diagnostic), these errors do not have the same consequences. Different weights may be given to false positives and false negatives.

2.2 *Quadratic separation*

Linear separation is a simple but not always a realistic model to encompass all real-life separation problems. One can well imagine instances where linear separation might not suffice to obtain a good classification. A finer definition of the discriminating function f may result in an improved quality of the separation. But on the other hand, the following reasons discourage investigation of highly complex forms of separating functions:

(i) Except for very special well-known cases, there is no reason why *a priori* information about the family of discriminating would be available,
(ii) Using very refined discriminating functions involves determining a large number of parameters, making the optimization problem (5) increasingly difficult.
(iii) As in statistics, more parameters are likely to make the separator less robust when applied to unseen data.

Keeping that in mind, we look at a simple refinement of the linear discrimination function. We consider the following formulation for the discrimination function between two classes, that aims at separating $\mathcal{A}_1$ from $\mathcal{A}_2$ by a quadratic form

$$Q(\boldsymbol{x}) = \langle \boldsymbol{\Omega x}, \boldsymbol{x} \rangle + \langle \boldsymbol{\omega}, \boldsymbol{x} \rangle + \gamma. \tag{6}$$

Without loss of generality, we suppose that $\boldsymbol{\Omega}$ is an upper-triangular matrix in $\mathbb{R}^n$, $\boldsymbol{\omega}$ is a real vector of $\mathbb{R}^n$, and γ a scalar. Since this formulation contains the linear case as a special case, it may lead to a more efficient separation. The obvious drawback is the increase in the problem dimension, which is now $\frac{n(n+1)}{2} + n + 1$. This results in a severe limitation of the possible applications. However, in real cases only a few attributes values expressed by $\boldsymbol{\Omega}$ are correlated. If we have *a priori* information about the structure of matrix $\boldsymbol{\Omega}$ (or possibly correlated attributes), we can set most values $\boldsymbol{\Omega}_{ij}$ to zero, reducing the number of variables and making the problem tractable.

Considering a separation margin ν, we want to find $\boldsymbol{\Omega}$, $\boldsymbol{\omega}$ and γ such that

$$\begin{aligned} Q(\boldsymbol{x}) &\geq \nu, \quad \forall \boldsymbol{x} \in \mathcal{A}_1, \\ Q(\boldsymbol{x}) &\leq -\nu, \quad \forall \boldsymbol{x} \in \mathcal{A}_2. \end{aligned}$$

The misclassification errors are then,

$$\boldsymbol{e}_i^1 = \max(\nu - Q(\boldsymbol{a}_i), 0), \forall \boldsymbol{a}_i \in \mathcal{A}_1,$$
$$\boldsymbol{e}_i^2 = \max(\nu + Q(\boldsymbol{a}_i), 0), \forall \boldsymbol{a}_i \in \mathcal{A}_2.$$

As in the linear case, the individual errors $\boldsymbol{e}_i^1$, $\boldsymbol{e}_i^2$ are point-wise maximum of a linear function in $(\boldsymbol{\Omega}, \boldsymbol{\omega}, \gamma)$ and 0. Thus, the minimization is essentially of the same nature but the variables have a higher dimension.

3 The Homogeneous Analytic Center Cutting Plane Method

We now describe our approach for solving the separation problem, starting with brief theoretical background of cutting plane methods. This section then goes on to discuss the motivation behind our approach, and the application of the algorithm to the specific problem.

3.1 *h-ACCPM Algorithm*

Let $f(\boldsymbol{u})$ denote a convex function to be minimized. In our problem of interest, $\boldsymbol{u} = (\boldsymbol{\omega}, \gamma)$ and f is the sum of the violations. We use $f'(\boldsymbol{u})$ to denote the derivative or an element of the subgradient set of f. Finally, let $\boldsymbol{u}^*$ be an optimal point. By convexity of f and optimality of $\boldsymbol{u}^*$,

$$\langle f'(\boldsymbol{u}), \boldsymbol{u}^* - \boldsymbol{u} \rangle \leq f(\boldsymbol{u}^*) - f(\boldsymbol{u}) \leq 0.$$

Therefore the *cut* $\langle f'(\boldsymbol{u}), \boldsymbol{u} - \boldsymbol{u}^* \rangle \geq 0$ is valid at each $\boldsymbol{u}$. A cutting plane method uses this inequality to build a sequence of *query* points $\boldsymbol{u}^1, \boldsymbol{u}^2, \ldots$, and a sequence of *cuts* $\langle f'(\boldsymbol{u}^k), \boldsymbol{u}^k - \boldsymbol{u} \rangle \geq 0$, with the property that

$$\boldsymbol{u}^{k+1} \in \{\boldsymbol{u} \mid \langle f'(\boldsymbol{u}^j), \boldsymbol{u}^j - \boldsymbol{u} \rangle \geq 0,\ j = 1, \ldots k\}.$$

The sets in which the query points are selected form a nested sequence whose intersection contains all optimal solutions $\boldsymbol{u}^*$. The generic cutting plane algorithm is sketched below

Algorithm 1
`Basic iteration`

`(i) Select a query point` $\boldsymbol{u}^{k+1} \in \{\boldsymbol{u} \mid f'(\boldsymbol{u}^j)^T(\boldsymbol{u}^j - \boldsymbol{u}) \geq 0,\ \ j = 1, \ldots k\}$.
`(ii) Compute` $f'(\boldsymbol{u}^{k+1})$ `(oracle).`
`(iii) Test convergence.`

More information is needed to implement steps 1 and 3. We discuss step 2 in section 3.3. Hopefully, the convergence test should guarantee $f(\boldsymbol{u}^{k+1}) \leq f(\boldsymbol{u}^*) + \epsilon$, when it is activated.

The Homogeneous Analytic Center Cutting Plane Method is one such method with nice convergence properties. The method applies to an homogeneous oracle, that is an oracle mapping $h(\boldsymbol{v})$ defined on a cone K, which at each $\boldsymbol{v} \in K$ satisfies the following properties:

(i) $\langle h(\boldsymbol{v}), \boldsymbol{v} - \boldsymbol{v}^* \rangle \geq 0, \ \forall \boldsymbol{v}^* \in V^*$.
(ii) $h(t\boldsymbol{v}) = h(\boldsymbol{v}), \ \forall t > 0$.
(iii) $\langle h(\boldsymbol{v}), \boldsymbol{v} \rangle = 0$.

The set $V^* \subset K$ is the solution set. The mapping $f'(\boldsymbol{u})$ associated with a general convex function $f(\boldsymbol{u})$ does not satisfy the last two hypotheses. A suitable mapping h is constructed via an embedding of the original space into a cone in $\mathbb{R}^{n+1}$. The embedding is as follows. It maps any point $\boldsymbol{u} \in \mathbb{R}^n$ on the ray $\{\boldsymbol{v} = (\boldsymbol{u}t, t) \in \mathbb{R}^{n+1} \mid t > 0\}$. If we denote $\boldsymbol{v}_{-n}$ the subvector of the n first components of $\boldsymbol{v} \in \mathbb{R}^{n+1}$ with $\boldsymbol{v}_{n+1} > 0$, the projection

$$\boldsymbol{v} \to \boldsymbol{u} = \frac{1}{\boldsymbol{v}_{n+1}} \boldsymbol{v}_{-n}$$

yields the generator $\boldsymbol{u}$ of the ray in $\mathbb{R}^{n+1}$. The cut in $\mathbb{R}^n$ translates into a cut in the embedding space

$$\langle h(\boldsymbol{v}), \boldsymbol{v} - \boldsymbol{v}^* \rangle \geq 0$$

where $\boldsymbol{v}^*$ is any point on the ray generated by $\boldsymbol{u}^*$ and

$$h(\boldsymbol{v}) = (f'(\boldsymbol{u})^T, -\langle f'(\boldsymbol{u}), \boldsymbol{u} \rangle)^T. \tag{7}$$

It is easy to verify that $\langle h(\boldsymbol{v}), \boldsymbol{v} \rangle = 0$. Hence,

$$\begin{aligned} \langle h(\boldsymbol{v}), \boldsymbol{v} - \boldsymbol{v}^* \rangle &= -\langle f'(\frac{\boldsymbol{v}_{-n}}{\boldsymbol{v}_{n+1}}), \boldsymbol{v}^*_{-n} \rangle + \langle f'(\frac{\boldsymbol{v}_{-n}}{\boldsymbol{v}_{n+1}}), \frac{\boldsymbol{v}_{-n}}{\boldsymbol{v}_{n+1}} \rangle \boldsymbol{v}^*_{n+1} \\ &= \langle f'(\boldsymbol{u}), \boldsymbol{u} - \boldsymbol{u}^* \rangle \boldsymbol{v}^*_{n+1}. \end{aligned}$$

Since $\boldsymbol{v}^*_{n+1} > 0$, the two inequalities $\langle f'(\boldsymbol{u}), \boldsymbol{u} - \boldsymbol{u}^* \rangle \geq 0$ and $\langle h(\boldsymbol{v}), \boldsymbol{v} - \boldsymbol{v}^* \rangle \geq 0$ are equivalent.

At step 1, the Homogeneous Analytic Center Cutting Plane Method chooses as query point the minimizer of the problem

$$\boldsymbol{v}^{k+1} = \arg\min F_k(\boldsymbol{v}),$$

where F_k is recursively defined by

1. $F_0(\boldsymbol{v}) = \frac{1}{2}||\boldsymbol{v}||^2 + F(\boldsymbol{v})$.

2. $F_k(\boldsymbol{v}) = F_{k-1}(\boldsymbol{v}) - \log\langle h(\boldsymbol{v}^k), \boldsymbol{v}^k - \boldsymbol{v}\rangle$.

The function $F(\boldsymbol{v})$ is a self-concordant barrier for the cone K. (In general, $F(\boldsymbol{v}) = -\log \boldsymbol{v}_{n+1}$.)

The homogeneous cutting plane method is proved to produce a solution $f(\boldsymbol{u}^{k+1}) \leq f(\boldsymbol{u}^*) + \epsilon$ with a number of iterations k at most proportional to $1/\epsilon^2$, but the complexity is independent of n. It is thus pseudo-polynomial. For a detailed convergence analysis, we refer to [7] and [10]. The iterates of the Homogeneous Analytic Center Cutting Plane Method always satisfy $\boldsymbol{v}^k_{n+1} > 0$.

3.2 *Why use the homogeneous cutting plane method?*

Any optimization scheme for convex unconstrained problems could be used to minimize (5). Our decision to use ACCPM has been guided by the following reasons:

(i) The homogeneous version of ACCPM has been designed especially for problems arising in cones, and thus applies naturally to the discrimination problems. The only special requirement is to have a homogeneous oracle. This is done by embedding the cutting plane given by the oracle into a higher dimensional space, as done in equation (7).

(ii) For separation margin $\nu = 0$, one of the main difficulties that is encountered in many mathematical programming approaches is to discard the null solution $\boldsymbol{\omega} = \mathbf{0}$ and $\gamma = 0$. By definition, interior point methods based on central points never take the null solution into consideration. In the particular case of h-ACCPM, the origin lies on the boundary of the domain of the augmented barrier function $F_k(\boldsymbol{v})$ which goes to $+\infty$ as $\boldsymbol{v}$ goes to $\mathbf{0}$. Hence, this barrier prevents the iterates from getting too close to the origin.

This property is useful since it enables us to meet the requirement stated in [4]: do not resort to extraneous constraint to eliminate the null solution.

3.3 *Oracles for linear and quadratic separations*

We now look at the computation of step 2 of the cutting plane algorithm in section 3.1. The parameters of the separator are the unknowns of our problem.

As noted before, the high number of decision variables ($\frac{n(n+1)}{2} + n + 1$) in the full quadratic model may be a severe limitation. One straightforward improvement is to suppose that only some of the $\frac{n(n+1)}{2}$ variables are significant. For instance, we consider the intermediate case where $\boldsymbol{\Omega}$ is a diagonal matrix.

Hence, we consider three types of separations in the space of attributes:

(i) Linear separation.
(ii) Quadratic separation with a diagonal matrix $\mathbf{\Omega}$.
(iii) Quadratic separation with triangular matrix $\mathbf{\Omega}$.

The misclassification error function has the desirable property of being a convex piecewise linear in the unknown parameters, for all the three formulations.

In all cases, the matrix $\mathbf{\Omega}$ can be considered as a collection of individual variables $\mathbf{\Omega}_{ij}$. Hence, the set of decision variables is viewed as a vector of $\frac{n(n+1)}{2} + n + 1$ components in the case of triangular matrix $\mathbf{\Omega}$, and $2n + 1$ components in the case of diagonal matrix $\mathbf{\Omega}$.

Let $\mathcal{A}'_1$ and $\mathcal{A}'_2$ define the set of misclassified points for $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. Hence,

$$\mathcal{A}'_1 = \{i : \boldsymbol{a}_i \in \mathcal{A}_1, \boldsymbol{e}_i^1 > 0\},$$
$$\mathcal{A}'_2 = \{i : \boldsymbol{a}_i \in \mathcal{A}_2, \boldsymbol{e}_i^2 > 0\}.$$

3.3.1 ***Linear separation oracle.*** In the linear case, we have $\mathbf{\Omega} = \mathbf{0}$. Differentiating (5) in $(\boldsymbol{\omega}, \gamma)$ yields $(-\boldsymbol{a}_i, 1)$ for the points of $\mathcal{A}_1$, and $(\boldsymbol{a}_i, -1)$ for the points of $\mathcal{A}'_2$. Thus, for any feasible point $(\boldsymbol{\omega}, \gamma)$, the oracle output is defined by $\boldsymbol{g} = \boldsymbol{g}_1 + \boldsymbol{g}_2$, where

$$\boldsymbol{g}_1 = \tfrac{1}{|\mathcal{A}_1|} \textstyle\sum_{i \in \mathcal{A}'_1} (-\boldsymbol{a}_i, 1),$$
$$\boldsymbol{g}_2 = \tfrac{1}{|\mathcal{A}_2|} \textstyle\sum_{i \in \mathcal{A}'_2} (\boldsymbol{a}_i, -1).$$

3.3.2 ***Diagonal quadratic separation.*** In the case of a diagonal matrix $\mathbf{\Omega}$, the gradient transforms to

$$\boldsymbol{g}_1 = \tfrac{1}{|\mathcal{A}_1|} \sum_{i \in \mathcal{A}'_1} (-\boldsymbol{a}_i \otimes \boldsymbol{a}_i, -\boldsymbol{a}_i, -1),$$
$$\boldsymbol{g}_2 = \tfrac{1}{|\mathcal{A}_2|} \sum_{i \in \mathcal{A}'_2} (\boldsymbol{a}_i \otimes \boldsymbol{a}_i, \boldsymbol{a}_i, 1),$$

where $\otimes$ denotes the component-wise multiplication operator for vectors.

3.3.3 ***General quadratic separation.*** For $i \in \mathcal{A}$, let $\boldsymbol{u}_i$ represent the elements of the upper-triangular matrix $\boldsymbol{a}_i \boldsymbol{a}_i^T$ listed as a vector. Differentiating

the objective function in $(\boldsymbol{\Omega}, \boldsymbol{\omega}, \gamma)$ yields $(-\boldsymbol{u}_i, -\boldsymbol{a}_i, -1)$ for a point of $\mathcal{A}'_1$, and $(\boldsymbol{u}_i, \boldsymbol{a}_i, 1)$ for a point of $\mathcal{A}'_2$. Hence, the gradient vector is given by $\boldsymbol{g} = \boldsymbol{g}_1 + \boldsymbol{g}_2$, where

$$\boldsymbol{g}_1 = \frac{1}{|\mathcal{A}_1|} \sum_{i \in \mathcal{A}'_1} (-\boldsymbol{u}_i, -\boldsymbol{a}_i, -1),$$

$$\boldsymbol{g}_2 = \frac{1}{|\mathcal{A}_2|} \sum_{i \in \mathcal{A}'_2} (\boldsymbol{u}_i, \boldsymbol{a}_i, 1).$$

4 Numerical experiments

4.1 *Datasets and simulation procedure*

We report on numerical results on data sets originating from the UCI machine learning database [9], and follow the same simulation procedure as in [11]. Table 1 summarizes the information about our selected data sets. The third column gives the number of attributes (total) and whether they are binary, categorical (categ) or numeric (num). The class distribution gives the percentage of instances with positive and negative label.

Problem	Instances	binary	categ.	num.	total	class distribution (%)
`breastCancer`	683			9	9	34.48 / 65.52
`cards`	690	4	5	6	15	44.49 / 55.51
`chess`	3196	35	1		36	47.78 / 52.22
`credit`	1000			24	24	29.89 / 70.11
`heartDisease`	297	3	5	5	13	45.54 / 54.46
`hepatitis`	155	13		6	19	48.70 / 51.30
`houseVotes84`	435	16			16	38.62 / 61.38
`ionosphere`	351	1		32	33	35.90 / 64.10
`liver`	345			6	6	42.58 / 57.42
`monks3`	554	2	4	6		47.60 / 52.40
`musk`	476			166	166	42.99 / 57.00
`pima`	768			8	8	34.90 / 65.10
`promotergenes`	106		57		57	50.00 / 50.00
`sonar`	208			60	60	46.64 / 53.36
`spambase`	4601			57	57	39.40 / 60.60
`tictactoe`	958		9		9	34.66 / 65.34
`titanic`	2201	2	1		3	32.30 / 67.70
`WDBC`	569			30	30	37.26 / 62.74
`WPBC`	194			30	30	23.71 / 76.29

Table 1. Two-category datasets

Compared to more general methods, the h-ACCPM is quite restrictive, in the sense that it only accepts complete numerical data. Hence, some preprocessing was necessary. Converting categorical or qualitative data into numerical data was sometimes straightforward: for example yes/no features became 1/0 or 1/-1 variables. When a few values were missing, we simply removed the corresponding instances. In the special case where missing values were concentrated in the same features, we removed the corresponding features. The main idea was to avoid removing too many instances in order to have a fair comparison with other methods. For example, we transformed yes/no features with missing values into 1/0/-1 variables, 0 representing the missing value.

All variables have been scaled to zero mean and unit variance to prevent numeric problems. Compared to other techniques (for example neural networks), this does not seem to have great influence on our results.

We first considered the problem as a pure minimization problem, and ran h-ACCPM in order to minimize the misclassification error over the complete data sets. In the separation problem, the true objective is not the error function (distance to target of misclassified points) but the cardinality of the set of misclassified points. We use h-ACCPM to produce (hopefully) interesting query points. The best classification rate is not necessarily achieved at points with the smaller total error. Indeed, outliers may tilt the separation in the wrong way and induce many misclassifications, though with individual small errors. In practice, we let the algorithm run for a fixed number of iterations. We record the number of misclassified points at each iteration. The output of the run is the separator which achieved the best classification rate. This provides us with a separation which is optimal with respect to the existing data sets, but gives no insights about its ability to classify unseen data.

Therefore, we performed a ten-fold cross-validation. For each data set, a model is built on a training set containing 90% of the data, and the classifier is evaluated on a testing set composed of the remaining 10%. We repeat the operation 10 times with non-overlapping testing sets, so that every instance occurs in the testing set exactly once. Training and testing set are defined randomly.

We repeat the ten-fold cross-validation 10 times with different partitioning of the data into training and testing sets. The following statistics are reported:

- Average value of the misclassification rates,
- Median value of the misclassification rate over the 10 cross-validation results,
- Interquartile range (IQR) of the 10-fold cross-validation results. IQR is the difference between the first and third quartile. Exactly half of the data is comprised in this range.

4.1.1 ***Results.*** At the time of this submission, we mostly have results for the linear case. The experiments for the other formulations are still in progress.

Table 2 gives the mean and median errors and the interquartile range (IQR) for the cross-validation tests. The formulation used for the corresponding result is listed in column 5, where L refers to Linear, QD to Quadratic Diagonal and QT to Quadratic Triangular formulation.

dataset	mean error	median error	IQR	Formulation
`breastCancer`	0.76	0.73	0.24	L
`cards`	7.44	7.57	0.07	L
`chess`	1.51	1.50	0.09	L
`credit`	19.65	19.85	0.70	L
`heartDisease`	5.89	5.88	1.00	L
`hepatitis`	2.69	2.90	2.19	L
`houseVotes84`	0.51	0.46	0.35	L
`ionosphere`	2.59	2.56	0.29	QD
`liver`	21.17	21.46	1.17	QT
`monks3`	0.36	0.36	0.00	QT
`musk`	7.23	7.22	0.24	L
`pima`	16.31	16.34	0.71	L
`promotergenes`	2.53	2.36	1.41	L
`sonar`	8.43	8.43	0.52	L
`spambase`	6.74	6.75	0.16	L
`tictactoe`	0.07	0.00	0.07	QT
`titanic`	20.63	20.65	0.14	QT
`WDBC`	0.35	0.35	0.26	L
`WPBC`	6.07	6.16	0.97	L

Table 2. Results for separation problem using h-ACCPM

The linear separation can be considered as a special case of quadratic separation with a null diagonal matrix $\mathbf{\Omega}$. In a similar way, a diagonal matrix is a special case of triangular matrix. Thus, the error rates should decrease as the models become more complex. This is not always the case, mainly for two reasons. First, the models involve more and more variables, so that the computation effort to get good solutions increases. Since we stop the process after a given number of iterations, we do not always achieve stabilization in the quadratic case. Second, it appears to be more difficult to tune the different parameters of the algorithm for quadratic separations.

However, for some datasets, like `monks3`, the quadratic separation causes significant improvement. With the linear formulation this testcase has an average error of nearly 15%. For the datasets with many attributes, it appears

necessary to determine first which attributes really contribute to an efficient separation, so that the number of variables would be kept reasonable. Feature selection seems to be a good way to improve the present results.

5 Conclusion

In this paper we solved a continuous optimization formulation for the discrimination problem with two classes of instances. We showed that using the Homogeneous Analytic Center Cutting Plane Method enables us to avoid the undesired null solution without adding extraneous constraints. We solved 19 benchmark problems taken from the UCI machine learning repository and improved the best known results on some of them.

Since the problem dimension increases when quadratic (or more sophisticated) separations are calculated, our approach would probably benefit from feature selection. The method generalizes to multicategory discrimination problems. Implementing one-versus-rest and one-versus-one approaches is a natural extension of this work.

References

[1] P. S. Bradley and U. M. Fayyad and O. L. Mangasarian, 1999, Mathematical programming for data mining: formulations and challenges, *INFORMS Journal on Computing*, **11**(3), 217–238.
[2] Ch. Chen and O. L. Mangasarian, 1996, Hybrid Misclassification Minimization, *Advances in Computational Mathematics*, **5**(2), 127–136.
[3] O.L. Mangasarian, 1965, Linear and Non-linear Separation of Patterns by linear programming, *Operations Research*, **13**, 444–452.
[4] K.P. Bennett, O.L. Mangasarian, 1992, Robust Linear Programming Discrimination of Two Linearly Inseparable Sets, *Optimization Methods and Software*, **1**, 23–34.
[5] O. L. Mangasarian, 1994, Misclassification Minimization, *Journal of Global Optimization*, **5**(4), 309–323.
[6] J. L. Goffin, A. Haurie, and J.-Ph. Vial, 1992, Decomposition and nondifferentiable optimization with the projective algorithm, *Management Science*, **37**, 284–302.
[7] Yu. Nesterov and J.-Ph. Vial, 1999, Homogeneous Analytic Center Cutting Plane Methods for Convex Problems and Variational Inequalities, *SIAM Journal on Optimization*, **9**(3), 707–728.
[8] O. Péton, 2002, The homogeneous analytic center cutting plane method, PhD thesis, University of Geneva.
[9] C.L. Blake and C.J. Merz. UCI Repository of machine learning databases. Irvine, CA, 1998, http://www.ics.uci.edu/ mlearn/MLRepository.html
[10] Yu. Nesterov, O. Péton and J.-Ph Vial, 1999, Homogeneous analytic center cutting plane methods with approximate centers, *Optimization Methods and Software*, **11/12**, 243–273.
[11] D. Meyer, F. Leisch, and K. Hornik, 2003, The support vector machine under test, *Neurocomputing*, **55**, 169–186.