Machine Learning via Polyhedral Concave Minimization

O. L. Mangasarian*

Mathematical Programming Technical Report 95-20
November 1995

Dedicated to Klaus Ritter on the Occasion of his Sixtieth Birthday

Abstract

Two fundamental problems of machine learning, misclassification minimization [10, 24, 18] and feature selection, [25, 29, 14] are formulated as the minimization of a concave function on a polyhedral set. Other formulations of these problems utilize linear programs with equilibrium constraints [18, 1, 4, 3] which are generally intractable. In contrast, for the proposed concave minimization formulation, a successive linearization algorithm without stepsize terminates after a maximum average of 7 linear programs on problems with as many as 4192 points in 14-dimensional space. The algorithm terminates at a stationary point or a global solution to the problem. Preliminary numerical results indicate that the proposed approach is quite effective and more efficient than other approaches.

1 Introduction

We shall consider the following two fundamental problems of machine learning:

Problem 1.1 ***Misclassification Minimization*** *[24, 18] Given two finite point sets $\mathcal{A}$ and $\mathcal{B}$ in the n-dimensional real space R^n, construct a plane that minimizes the number of points of $\mathcal{A}$ falling in one of the closed halfspaces determined by the plane and the number of points of $\mathcal{B}$ falling in the other closed halfspace.*

Problem 1.2 ***Feature Selection*** *[4, 3] Given two finite point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n, select a sufficiently small number of dimensions of R^n such that a plane, constructed in the smaller dimensional space, optimizes some separation criterion between the sets $\mathcal{A}$ and $\mathcal{B}$.*

We immediately note that the misclassification minimization problem is NP-complete [6, Proposition 2]. But, effective methods for its solution have been proposed in [18] and implemented in [1]. An approximate technique [6] has also been implemented. The formulation that we propose in this work terminates in a finite number of linear programs (typically less than seven) at a vertex solution or stationary point of the problem.

We outline the contents of the paper now. In Section 2 we give a precise mathematical formulation of the misclassification minimization and feature selection problems and indicate how they can be set up as linear programs with equilibrium constraints and indicate some of the difficulties

*Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706, email: *olvi@cs.wisc.edu.* This material is based on research supported by Air Force Office of Scientific Research Grant F49620-94-1-0036 and National Science Foundation Grants CCR-9322479.

attendant this formulation. We then introduce in Section 3 a simple concave exponential approximation of the step function, similar to the classical sigmoid function of neural networks [28, 11, 17], but with the significant difference of concavity of the proposed approximation which is not shared by the sigmoid function. This concavity is possible, because the step function is applied here to nonnegative variables. This leads to a finite successive linearization algorithm (SLA) without a stepsize procedure that is described in Section 4 of the paper. Section 5 gives very encouraging results on numerical tests on the misclassification minimization and feature selection problems. Section 6 gives a concluding summary of the paper.

A word about our notation now. For a vector x in the n-dimensional real space R^n, x_+ will denote the vector in R^n with components $(x_+)_i := \max\{x_i, 0\}$, $i = 1, \ldots, n$. Similarly x_* will denote the vector in R^n with components $(x_*)_i := (x_i)_*$, $i = 1, \ldots, n$, where $(\cdot)_*$ is the step function defined as one for positive x_i and zero otherwise, while $|x|$ will denote a vector of absolute values of components of x. The base of the natural logarithm will be denoted by ε and for $y \in R^m$, ε^{-y} will denote a vector in R^m with component ε^{-y_i}, $i = 1, \ldots, m$. The norm $\|\cdot\|_p$ will denote the p norm, $1 \leqq p \leqq \infty$, while $A \in R^{m\times n}$ will signify a real $m \times n$ matrix. For such a matrix, A^T will denote the transpose, and A_i will denote row i. For two vectors x and y in R^n, $x \perp y$ will denote $x^T y = 0$. A vector of ones in a real space of arbitrary dimension will be denoted by e. The notation $\arg\min_{x\in S} f(x)$ will denote the set of minimizers of $f(x)$ on the set S. Similarly $\arg\text{vertex}\min_{x\in S} f(x)$ will denote the set of vertex minimizers of $f(x)$ on the polyhedral set S. By a separating plane, with respect to two given point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n, we shall mean a plane that attempts to separate R^n into two half spaces such that each open halfspace contains points mostly of $\mathcal{A}$ or $\mathcal{B}$. The symbol ":=" defines a quantity appearing on its left by a quantity appearing on its right. For $f : R^n \longrightarrow R$ which is differentiable at x, the notation $\nabla f(x)$ will represent the $1 \times n$ gradient vector. R^n_+ will denote the nonnegative orthant.

2 The Misclassification Minimization and Feature Selection Problems

We consider two nonempty finite point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n consisting of m and k points respectively that are represented by the matrices $A \in R^{m\times n}$ and $B \in R^{k\times n}$. The objective of both problems here is to construct a separating plane:

$$P := \{x \mid x \in R^n,\ x^T w = \gamma\}, \tag{1}$$

where $w \in R^n$, $\gamma \in R$, such that some error criterion is minimized. Thus in the exceptional case when the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ do not intersect, a single linear program [2] will generate a plane P that strictly separates the sets $\mathcal{A}$ and $\mathcal{B}$ as follows:

$$Aw \geqq e\gamma + e,\ \ Bw \leqq e\gamma - e \tag{2}$$

Our concern here is with the usually occurring case when *no* plane P exists satisfying (2). A desirable objective for such a case [24, 18, 1, 6] is to minimize the number of points of $\mathcal{A}$ lying in the complement of the closed halfspace reserved for it, that is, minimize the number of elements of $\mathcal{A}$ in:

$$\{x \mid x^T w < \gamma + 1\}, \tag{3}$$

as well as the number of points of $\mathcal{B}$ lying in the complement of the closed halfspace reserved for it, that is, minimize the number of elements of $\mathcal{B}$ in:

$$\{x \mid x^T w > \gamma - 1\} \tag{4}$$

Thus, if we introduce the nonnegative slack variables $y \in R^m$ and $z \in R^k$ and make use of the step function $(\cdot)_*$, the misclassification minimization problem can be stated as follows:

$$\min_{w,\gamma,y,z} \{e^T y_* + e^T z_* \mid y \geqq -Aw + e\gamma + e,\ y \geqq 0,\ z \geqq Bw - e\gamma + e,\ z \geqq 0\} \tag{5}$$

Note that without the step function $(\cdot)_*$ in (5), the problem becomes a linear program (essentially the robust linear program [2, Equation (2.11)], but without averaging over m and k), in which case y and z of (5) become:

$$y = (-Aw + e\gamma + e)_+,\ z = (Bw - e\gamma + e)_+ \tag{6}$$

Thus, problem (5) *without* the step function $(\cdot)_*$ is equivalent to:

$$\min_{w,\gamma} \left\| \begin{pmatrix} -Aw + e\gamma + e \\ Bw - e\gamma + e \end{pmatrix}_+ \right\|_1 \tag{7}$$

The objective of (7) measures sums of *distances* (assuming each row of A and B has unit 2-norm) of points of $\mathcal{A}$ in the open halfspace (3) from the plane $x^T w = \gamma + 1$ as well as points of $\mathcal{B}$ in the open halfspace (4) from the plane $x^T w = \gamma - 1$. By contrast the objective of problem (5) is to *count* the points of $\mathcal{A}$ contained in the open halfspace (3) and the points of $\mathcal{B}$ contained in the open halfspace (4) and attempt to minimize the totality of such points. To show indeed that problem (5) minimizes the total number of misclassified points we state the following simple lemma.

Lemma 2.1 *Let $a \in R^m$. Then*

$$r \in \arg\min_r \{e^T r_* \mid r \geqq a,\ r \geqq 0\} \Rightarrow r_* = a_* \tag{8}$$

Proof If r is a solution of the indicated minimization problem then for $i = 1, \ldots, m$:

$$(r_i)_* = \left\langle \begin{array}{l} 0 \text{ if } a_i \leqq 0 \\ 1 \text{ if } a_i > 0 \end{array} \right\rangle = (a_i)_*$$

□

By using this lemma on problem (5) we obtain the following proposition, which shows that any solution of (5) (and we will show in Proposition 2.4 below that (5) is always solvable) generates a plane that minimizes the number of misclassified points, that is points of $\mathcal{A}$ in (3) and points of $\mathcal{B}$ in (4).

Proposition 2.2 *Let $(\bar{w}, \bar{\gamma}, \bar{y}, \bar{z})$ solve (5), then*

$$e^T \bar{y}_* + e^T \bar{z}_* = \min_{w,\gamma} e^T(-Aw + e\gamma + e)_* + e^T(Bw - e\gamma + e)_* \tag{9}$$

Proof For a fixed (w, γ), let

$$(y(w,\gamma),\, z(w,\gamma)) \in \arg\min_{y,z} \left\{ e^T y_* + e^T z_* \;\middle|\; \begin{array}{l} y \geqq -Aw + e\gamma + e,\ y \geqq 0 \\ z \geqq Bw - e\gamma + e,\ z \geqq 0 \end{array} \right\} \tag{10}$$

By Lemma 2.1 we have that

$$\begin{array}{r} (y(w,\gamma))_* = (-Aw + e\gamma + e)_* \\ (z(w,\gamma))_* = (Bw - e\gamma + e)_* \end{array} \tag{11}$$

Since $(\bar{w}, \bar{\gamma}, \bar{y}, \bar{z})$ solves (5) we have by (10)-(11) that

$$e^T \bar{y}_* + e^T \bar{z}_* = \min_{w,\gamma} e^T(-Aw + e\gamma + e)_* + e^T(Bw - e\gamma + e)_* \tag{12}$$

□

To establish the existence of solution to problem (5) and to relate it to a linear program with equilibrium constraints (LPEC) [18, 19, 16, 15], we state the following lemma.

Lemma 2.3 *Let $a \in R^m$. Then*

$$r = a_*,\ u = a_+ \Leftrightarrow (r, u) = \arg\min_{r,u} \{e^T r \mid 0 \leq r \perp u - a \geqq 0,\ 0 \leqq u \perp -r + e \geqq 0\} \tag{13}$$

Proof The constraints of the minimization problem constitute the Karush-Kuhn-Tucker conditions for the dual linear programs:

$$\max_r \{a^T r \mid 0 \leq r \leqq e\},\ \min_u \{e^T u \mid u \geqq a,\ u \geqq 0\} \tag{14}$$

which are solved by:

$$r_i = \left\langle \begin{array}{l} 0 \text{ for } a_i < 0 \\ r_i \in [0,1] \text{ for } a_i = 0 \\ 1 \text{ for } a_i > 0 \end{array} \right. ,\quad u = a_+ \tag{15}$$

The objective function $e^T r$ minimized in (13) renders the solution r of (15) unique by making $r_i = 0$ for $a_i = 0$, thus giving $r = a_*$. □

By using Lemma 2.3, problem (5) can be written in the following equivalent form as an LPEC:

$$\min_{w,\gamma,y,z,r,u,s,v} \left\{ e^T r + e^T s \;\middle|\; \begin{array}{ccc} 0 \leqq r \perp u - y \geqq 0 & & 0 \leqq s \perp v - z \geqq 0 \\ 0 \leqq u \perp -r + e \geqq 0 & & 0 \leqq v \perp -s + e \geqq 0 \\ y \geqq -Aw + e\gamma + e & ; & z \geqq Bw - e\gamma + e \\ y \geqq 0 & & z \geqq 0 \end{array} \right\} \tag{16}$$

Since the nonempty (take $w = 0,\ \gamma = 0,\ y = e,\ z = e,\ r = e,\ s = e,\ u = e,\ v = e$) feasible region of (16) is the union of a finite number of polyhedral sets over which the linear objective function $e^T r + e^T s$ is bounded below by zero, it follows that $e^T r + e^T s$ attains a minimum on each of these polyhedral sets. The minimum of these minima is a solution of (16). Since (16) is equivalent to (5), we have the following.

Proposition 2.4 *The misclassification minimization problem (5) has a solution.*

We turn our attention to our second problem, the feature selection problem. The problem again is to separate the finite point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n, *but* with the additional requirement of using as few of the dimensions of R^n as possible. If we take as our point of departure the robust linear program [2, Equation (2.11)], which is very effective in discriminating between sets arising from real world problems [21], and motivate our formulation by the perturbation results of linear programming [20] to suppress as many of the coefficients w of the separating plane $\{x \mid x^T w = \gamma\}$ as possible, we obtain the following problem for a suitably chosen $\lambda \in [0, 1]$:

$$\min_{w,\gamma,y,z} \left\{ (1-\lambda)\left(\frac{e^T y}{m} + \frac{e^T z}{k}\right) + \lambda e^T v_* \;\middle|\; \begin{array}{c} Aw - e\gamma + y \geqq e,\ -Bw + e\gamma + z \geqq e,\ y \geqq 0,\ z \geqq 0 \\ -v \leqq w \leqq v \end{array} \right\} \qquad (17)$$

For $\lambda = 0$, we obtain the robust linear program of [2]. For $\lambda = 1$, all components of w are suppressed yielding no useful result. For λ sufficiently small, the program (17) selects those solutions of the robust linear program, that is (17) with $\lambda = 0$, that minimize $e^T \mid w \mid_*$. This in effect suppresses as many components of w as possible. Computationally, one obviously varies λ until some "best" value of (w, γ) is obtained as evinced by a cross-validating procedure [30].

By using an identical technique to that used to establish the existence of a solution to problem (5), we can similarly replace problem (17) by an LPEC and establish existence of a solution to it. We thus can state the following result.

Proposition 2.5 *The feature selection problem (17) has a solution to each* $\lambda \in [0, 1]$.

We turn our attention now to algorithmic considerations by first approximating the step function $(\cdot)_*$, which appears in both problems (5) and (17), by a smooth concave approximation.

3 Concave Approximation of the Step Function

One of the most common and useful approximations in neural networks [28, 11] is the *sigmoid* function approximation of the step function ζ_* defined as

$$s(\zeta, \alpha) := \frac{1}{1 + \varepsilon^{-\alpha\zeta}}\ ,\ \alpha > 0 \qquad (18)$$

Here ε is the base of the natural logarithm. For moderate values of α, the sigmoid is a very adequate approximation of the step function ζ_*. A shortcoming of the sigmoid is that it is neither convex nor concave. This prevents us from invoking some of the fundamental properties of these functions. In the two applications of this paper, it turns out that the variables to which the step function is applied are nonnegative: y and z in problem (5) and v in problem (17). Consequently, we propose the following simpler concave approximation of the step function for nonnegative variables

$$t(\zeta, \alpha) := 1 - \varepsilon^{-\alpha\zeta} \quad \alpha > 0,\ \zeta \geqq 0 \qquad (19)$$

Two important consequences of this simpler concave approximation of the step function are: first, an existence proof to both the smooth concave approximation of the misclassification minimization problem (5) as well as to the smooth concave approximation of the feature selection problem (17) (Proposition 3.1 below), and second, a finite termination theorem (Theorem 4.2 below) for the successive linearization algorithm (SLA Algorithm 4.1 below). We now state the smooth approximations of the misclassification minimization and the feature selection problems.

3.1 Smooth Concave Misclassification Minimization Problem (5) Let $\alpha > 0$.

$$\min_{w,\gamma,y,z} \{m + k - e^T\varepsilon^{-\alpha y} - e^T\varepsilon^{-\alpha z} \mid y \geqq -Aw + e\gamma + e,\ y \geqq 0,\ \ z \geqq Bw - e\gamma + e,\ z \geqq 0\} \tag{20}$$

We note immediately that the concave objective function is bounded below by zero on the set $R^{n+1} \times R_+^{m+k}$ which contains the feasible region. Furthermore, this lower bound is attained by $y = 0,\ z = 0$, and some infeasible (w, γ) in general. The zero minimum is attained at a feasible point if and only if the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ do not intersect. Otherwise the minimized objective of (20) approximates from below (for moderate values of α) the smallest number of misclassified points by any plane $x^Tw = \gamma$.

3.2 Smooth Concave Feature Selection Problem (17) Let $\lambda \in [0, 1]$ and $\alpha > 0$.

$$\min_{w,\gamma,y,z} \left\{ (1-\lambda)\left(\frac{e^Ty}{m} + \frac{e^Tz}{k}\right) + \lambda(n - e^T\varepsilon^{-\alpha v}) \left| \begin{array}{c} Aw - e\gamma + y \geqq e, -Bw + e\gamma + z \geqq e, y \geqq 0, z \geqq 0 \\ -v \leqq w \leqq v \end{array} \right. \right\} \tag{21}$$

Again for this problem, the concave objective function is bounded below by zero on the feasible region. For various values of the parameter $\lambda \in [0, 1]$, emphasis of separation by the plane $x^Tw = \gamma$ is balanced against suppression of as many coefficients of w as possible, with the term $(n - e^T\varepsilon^{-\alpha v})$ giving an approximation (from below) to the number of nonzero coefficients of w.

By making use of [27, Corollary 32.3.3] which implies that a concave function, bounded from below on a nonempty polyhedral set, attains its minimum on that set, we can state the following existence results for the two smooth problems above.

Proposition 3.1 *The smooth concave misclassification minimization problem (20) and the smooth concave feature selection problem (21) have solutions.*

We turn our attention to algorithmic considerations.

4 Successive Linearization of Polyhedral Concave Programs

By replacing the variables (w, γ) by the nonnegative variables (w^1, γ^1, ζ^1) using the standard transformation $w = w^1 - e\zeta^1$, $\gamma = \gamma^1 - \zeta^1$, the smooth problems (20) and (21) can be transformed to the following concave minimization problem:

$$\min_x \{f(x) \mid Ax \leqq b,\ x \geqq 0\}, \tag{22}$$

where $f \colon R^\ell \to R$, is a differentiable, concave function bounded below on the nonempty polyhedral feasible region of (22), $A \in R^{p\times\ell}$ and $b \in R^p$. By [27, Corollary 32.3.4] it follows that f attains its minimum at a vertex of the feasible region of (22). We now prescribe a simple finite successive linearization algorithm (essentially a Frank-Wolfe algorithm [9] without a stepsize) for solving (22) that appears to give good computational results. (See Section 5.) Other more complex computational schemes for this problem are given in [13, 12].

4.1 Successive Linearization Algorithm (SLA) Start with a random $x^0 \in R^n$. Having x^i determine x^{i+1} as follows:

$$
\begin{array}{l}
x^{i+1} \in \text{arg vertex min}_{x \in X} \ \nabla f(x^i)(x - x^i) \\
X = \{x \mid Ax \leqq b, \ x \geqq 0\}
\end{array}
\tag{23}
$$

Stop if $x^i \in X$ and $\nabla f(x^i)(x^{i+1} - x^i) = 0$.
Comment: The condition $x^i \in X$ takes care of the possibility that x^0 may not be in X.

We show below that this is a finite algorithm which generates a strictly decreasing finite sequence $\{f(x^i)\}$, $i = 1, 2, \ldots, \bar{i}$, which terminates at an $x^{\bar{i}}$ that is a stationary point that may also be a global minimum solution.
Remark: SLA may be started from many different random starting points. This was not necessary in the present applications.

4.2 SLA Finite Termination Theorem Let f be a differentiable concave function on R^n that is bounded below on X. The SLA generates a finite sequence of iterates $\{x^1, x^2, \ldots, x^{\bar{i}}\}$ of strictly decreasing objective function values: $f(x^1) > f(x^2) > \ldots > f(x^{\bar{i}})$, such that $x^{\bar{i}}$ satisfies the minimum principle necessary optimality conditions

$$
\nabla f(x^{\bar{i}})(x - x^{\bar{i}}) \geqq 0, \quad \forall x \in X. \tag{24}
$$

Proof We first show that SLA is well defined. By the concavity of f and its boundedness from below on X, we have that

$$
-\infty < \inf_{x \in X} f(x) - f(x^i) \leqq f(x) - f(x^i) \leqq \nabla f(x^i)(x - x^i), \ \forall x \in X.
$$

It follows for any $x^i \in R^n$, even for an infeasible x^i such as x^0, that $\nabla f(x^i)(x - x^i)$ is bounded below on X. Hence the linear program (23) is solvable and has a vertex solution x^{i+1}. It follows for $i = 1, 2, \ldots,$ that

$$
\forall x \in X\!: \ \nabla f(x^i)(x - x^i) \geqq \min_{x \in X} \nabla f(x^i)(x - x^i) = \nabla f(x^i)(x^{i+1} - x^i) \left\langle \begin{array}{ll} < 0 & \text{(a)} \\ = 0 & \text{(b)} \end{array} \right. \tag{25}
$$

We note immediately that because $x^i \in X$ for $i = 1, 2, \ldots,$ it follows that $\nabla f(x^i)(x^{i+1} - x^i) \leqq 0$. Hence only two cases, (a) or (b), can occur, as indicated above. When case (a) above occurs, the algorithm does not stop at iteration i, and we have from the concavity of f and the strict inequality of case (a) that:

$$
f(x^{i+1}) \leqq f(x^i) + \nabla f(x^i)(x^{i+1} - x^i) < f(x^i)
$$

Hence $f(x^{i+1}) < f(x^i)$, for $i = 1, 2, \ldots$. When case (b) occurs we then have that:

$$
\forall x \in X\!: \ \nabla f(x^i)(x - x^i) \geqq 0, \tag{26}
$$

and the algorithm terminates (provided $x^i \in X$, which may not be the case if $x^i = x^0 \notin X$), and set $\bar{i} = i$. The point $x^{\bar{i}}$ thus satisfies the minimum principle necessary optimality conditions (26) with $x^{\bar{i}} = x^i$, and $x^{\bar{i}}$ may be a global solution. Furthermore, since X has a finite number of vertices, $\{f(x^i)\}$ is strictly decreasing and $f(x)$ is bounded below on X, it follows that case (b) must occur after a finite number of steps. $\square$

We turn our attention to some computational results.

5 Numerical Tests

The proposed approach was tested numerically on publicly available databases from the University of California Repository of Machine Learning Databases [22] as well as the Star/Galaxy database collected by Odewahn [26]. For all the numerical results reported, the value of α used in the concave

Data Set	m k n	Percent of Correctly Classified Points Time Seconds SPARCstation 20 Average No. of LPs over 10 Runs	
		PMM	SLA
WBC Prognosis	28 119 32	95.92 10.65	93.2 0.86 3.0
WBCD	239 443 9	98.57 24.65	97.6 9.47 5.7
Cleveland Heart	216 81 14	91.43 17.46	89.3 2.69 4.3
Ionosphere	225 126 34	98.42 27.26	97.0 10.30 4.0
Liver Disorders	145 200 6	74.85 18.51	71.4 1.09 5.5
Pima Diabetes	268 500 8	80.55 51.40	78.3 14.33 6.5
Star/Galaxy(Dim)	2082 2110 14	96.52 1122.70	96.1 779.89 5.9
Star/Galaxy(Bright)	1505 957 14	99.89 266.13	99.8 69.48 3.2
Tic Tac Toe	626 332 9	69.12 46.45	66.6 6.44 3.3
Votes	168 267 16	98.82 14.76	96.9 1.56 3.4
Total Times		1599.97	896.11

Table 1: Comparison of Successive Linearization Algorithm (SLA) Algorithm 4.1 for the Smooth Misclassification Minimization Problem (20) with the Parametric Minimization Method (PMM) [18, 1]. SLA was coded in GAMS [5] utilizing the CPLEX solver [7]. PMM was coded was coded in AMPL [8] utilizing the MINOS LP solver [23].

exponential approximation $t(\zeta, \alpha) = 1 - \varepsilon^{-\alpha\zeta}$ to the step function ζ_*, was five. This value of α allows $t(\zeta, \alpha)$ to capture the essence of of the step function ζ_* with sufficient smoothness to make the proposed algorithm work effectively without overflow or underflow.

The first test consisted in applying the SLA 4.1 to the smooth misclassification minimization problem (20). For this problem ten databases were used from the Irvine repository and the Star/Galaxy database. Table 1 gives the percent of correctly separated points as well as CPU times using an average of ten SLA runs on the smooth misclassification minimization problem (20). These quantities are compared with those of a parametric minimization method (PMM) applied to an LPEC associated with the misclassification minimization [18, 1]. Table 1 shows that the much simpler SLA algorithm obtained a separation that was almost as good as the parametric method for solving the LPEC at considerably less computing cost. Each problem was solved using no more than a maximum average of 7 LPs over ten runs. Average of solution times of the SLA over all problems run was 56% of the average PMM solution times.

Our second test consisted of solving the smooth concave feature selection problem (21) by SLA 4.1. The test problem consisted of the Wisconsin Breast Cancer Database WBCD tested in the above set of tests, with one modification. Two new random features, uniformly distributed on the interval $[0, 10]$ were added to the problem, so that the problem space was R^{11} instead of the original R^9. With $\lambda = 0.05$ in problem (21), and by solving 6 successive linear programs, the SLA was able to suppress the effect of the random components x_{10} and x_{11} by setting w_{10} and w_{11} equal to zero, as well as some other components: w_3, w_4, w_5, w_7, and w_9. The resulting separation in R^4 correctly separated 97.1% of the points, which is almost as good as the 97.6% correctness obtained above without the feature selection option by solving the misclassification minimization problem (20) in R^9. This indicates that, for this problem, the stationary point obtained by the SLA algorithm in R^4 for the smooth feature selection problem (21) is almost as good as the stationary point obtained in R^9 for the smooth misclassification minimization problem (20). The key observation however, is that the feature selection approach proposed here, not only gets rid of extraneous random features, but also of unimportant features in the original problem.

6 Conclusion

We have formulated two important problems of machine learning: misclassification minimization and feature selection as the minimization of a simple concave function on a polyhedral set that is always solvable. A successive linearization algorithm that requires the solution of a few LPs in each instance appears to be a very effective method of solution.

Acknowledgement

I am indebted to my Ph.D. student Paul S. Bradley for the numerical testing of the proposed algorithm.

References

[1] K. P. Bennett and E. J. Bredensteiner. A parametric optimization method for machine learning. Department of Mathematical Sciences Math Report No. 217, Rensselaer Polytechnic Institute, Troy, NY 12180, 1994. ORSA Journal on Computing, submitted.

[2] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[3] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. Technical report, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1995. To appear.

[4] E. J. Bredensteiner and K. P. Bennett. Feature minimization within decision trees. Department of Mathematical Sciences Math Report No. 218, Rensselaer Polytechnic Institute, Troy, NY 12180, 1995.

[5] A. Brooke, D. Kendrick, and A. Meeraus. *GAMS: A User's Guide.* The Scientific Press, South San Francisco, CA, 1988.

[6] Chunhui Chen and O. L. Mangasarian. Hybrid misclassification minimization. Technical Report 95-05, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, February 1995. Advances in Computational Mathematics, to appear. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-05.ps.Z.

[7] CPLEX Optimization Inc., Incline Village, Nevada. *Using the CPLEX(TM) Linear Optimizer and CPLEX(TM) Mixed Integer Optimizer (Version 2.0)*, 1992.

[8] R. Fourer, D. Gay, and B. Kernighan. *AMPL.* The Scientific Press, South San Francisco, California, 1993.

[9] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[10] David Heath. *A geometric Framework for Machine Learning.* PhD thesis, Department of Computer Science, Johns Hopkins University–Baltimore, Maryland, 1992.

[11] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, California, 1991.

[12] R. Horst, P. Pardalos, and N. V. Thoai. *Introduction to Global Optimization.* Kluwer Academic Publishers, Dodrecht, Netherlands, 1995.

[13] R. Horst and H. Tuy. *Global Optimization.* Springer–Verlag, Berlin, 1993. Second, Revised Edition.

[14] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, San Mateo, CA, 1994. Morgan Kaufmann.

[15] Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints.* Cambridge University Press, Cambridge, England, 1996.

[16] Z.-Q. Luo, J.-S. Pang, D. Ralph, and S.-Q. Wu. Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints. Technical Report 275, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Hamilton, Ontario L8S 4K1, Canada, 1993. Mathematical Programming, to appear.

[17] O. L. Mangasarian. Mathematical programming in neural networks. *ORSA Journal on Computing*, 5(4):349–360, 1993.

[18] O. L. Mangasarian. Misclassification minimization. *Journal of Global Optimization*, 5:309–323, 1994.

[19] O. L. Mangasarian. The ill-posed linear complementarity problem. Technical Report 95-15, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, August 1995. Proceedings of the International Conference on Complementarity Problems, Johns Hopkins University, November 1-4, 1995, SIAM Publishers, Philadelphia, PA, submitted.

[20] O. L. Mangasarian and R. R. Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, November 1979.

[21] O. L. Mangasarian, W. Nick Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, July-August 1995.

[22] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, California, http://www.ics.uci.edu/AI/ML/MLDBRepository.html., 1992.

[23] B. A. Murtagh and M. A. Saunders. MINOS 5.0 user's guide. Technical Report SOL 83.20, Stanford University, December 1983. MINOS 5.4 Release Notes, December 1992.

[24] S. Murthy, S. Kasif, S. Salzberg, and R. Beigel. OC1: Randomized induction of oblique decision trees. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 322–327, Cambridge, MA 02142, 1993. The AAAI Press/The MIT Press.

[25] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26(9):917–922, September 1977.

[26] S. Odewahn, E. Stockwell, R. Pennington, R. Hummphreys, and W. Zumach. Automated star/galaxy discrimination with neural networks. *Astronomical Journal*, 103(1):318–331, 1992.

[27] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[28] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing*. MIT Press, Cambridge, Massachusetts, 1986.

[29] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[30] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36:111–147, 1974.