Robust Linear and Support Vector Regression

Olvi L. Mangasarian and David R. Musicant

Abstract—The robust Huber M-estimator, a differentiable cost function that is quadratic for small errors and linear otherwise, is modeled exactly, in the original primal space of the problem, by an easily solvable simple convex quadratic program for both linear and nonlinear support vector estimators. Previous models were significantly more complex or formulated in the dual space and most involved specialized numerical algorithms for solving the robust Huber linear estimator [3], [6], [12], [13], [14], [23], [28]. Numerical test comparisons with these algorithms indicate the computational effectiveness of the new quadratic programming model for both linear and nonlinear support vector problems. Results are shown on problems with as many as 20,000 data points, with considerably faster running times on larger problems.

Index Terms—Support vector machines, regression, Huber M-estimator, kernel methods.

1 Introduction

We consider the generally unsolvable system of linear equations

$$Ax = b, \tag{1}$$

where A is a given $\ell \times d$ real matrix of ℓ observations and b is a real $\ell \times 1$ vector of corresponding values, all taken from a given dataset. To obtain an approximate solution of (1), one usually minimizes an error residual:

$$\min_x \sum_{i=1}^{\ell} \rho((Ax-b)_i), \tag{2}$$

where typically ρ is the absolute value function or its square. In order to deemphasize outliers and avoid the nondifferentiability of the robust absolute value error residual, a popular residual is the Huber M-estimator cost function [9]:

$$\rho(t) = \begin{cases} \frac{1}{2}t^2, & \text{if } |t| \le \gamma \\ \gamma|t| - \frac{1}{2}\gamma^2, & \text{if } |t| > \gamma, \end{cases} \tag{3}$$

where γ is some positive number. Because the function switches at $|t| = \gamma$ from being quadratic to linear, special methods have been developed for the linear error minimization problem (2) with the Huber M-estimator. A Newton method was proposed in [10], which is possibly nonconvergent because the objective function of (2) is not twice differentiable. Other fairly complex Newton type methods were considered in [28], [6], [14], [12]. In [34], a minmax formulation leading to a mixed complementarity problem was proposed, which subsequently [35] was reduced to our quadratic formulation (23). Relations between the ℓ_1 and Huber estimators, as well as iterative methods, are proposed in [13].

• *The authors are with the Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706. E-mail: {olvi, musicant}@cs.wisc.edu.*

Manuscript received 31 Jan. 2000; revised 29 June 2000; accepted 18 July 2000.
Recommended for acceptance by C. Brodley.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111352.

We outline the content of the paper now. In Section 2, we first give a simple quadratic programming formulation of the Huber M-estimator cost function in one dimension (Lemma 2.1) and then set up the corresponding convex quadratic program (9) for the linear estimator (2) with ρ given by (3) (Proposition 2.2). This explicit convex quadratic programming formulation in the original primal space of the problem is rather simple and easily interpretable, but does not seem to have been given previously in other works that have considered dual formulations of this problem [13], [29], [31]. By using parametric perturbation results of linear programming [17], we show that, for all values of the parameter $\gamma \ge \bar{\gamma}$ for some $\bar{\gamma}$, a Huber linear estimator is just an ordinary least squares estimate (Proposition 2.3). On the other hand, for all sufficiently small values of γ, the Huber estimates depend linearly on γ and converge to a least 1-norm solution (Proposition 2.4). Li and Swetits [13] have studied the dual (17) of the Huber M-estimator quadratic program (9). They show that (17) is a least 2-norm formulation of the dual of the least 1-norm estimator. In addition, they give perturbation results for the solution of (17). Smola [29] also presents a dual formulation of the Huber M-estimator quadratic program [31]. In Section 3, we set up a convex quadratic program (23) for a nonlinear generalized support vector machine [33], [16] which extends the Huber loss function to large classes of nonlinear regression problems. Related but different loss functions were studied in [27], [26], [30], [19] in conjunction with support vector regression. Numerical test results presented in Section 4 show that our direct convex quadratic formulation is considerably faster than earlier proposed methods, such as Huber's Gauss-Seidel method [12], Smola's dual formulation [29], Madsen and Nielsen's Newton type method [14], and Li's conjugate gradient method [12].

A word about our notation. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. The scalar (inner) product of two vectors x and y in the d-dimensional real space R^d will be denoted by $x'y$. For an $\ell \times d$ matrix A, A_i will denote the ith row of A. A column vector of ones of arbitrary dimension will be

denoted by e. For $A \in R^{\ell \times d}$ and $B \in R^{d \times \ell}$, the **kernel** $K(A, B)$ maps $R^{\ell \times d} \times R^{d \times \ell}$ into $R^{\ell \times \ell}$. In particular, if x and y are column vectors in R^d, then $K(x', A')$ is a row vector in R^ℓ, $K(x', y)$ is a real number and $K(A, A')$ is an $\ell \times \ell$ matrix. Note that for our purposes here $K(A, A')$ will be assumed to symmetric, that is, $K(A, A') = K(A, A')'$.

2 Robust Linear Regression as a Convex Quadratic Program

We begin with a very simple lemma that generates the Huber cost function in one dimension as the unconstrained minimum value of a simple convex quadratic function.

Lemma 2.1 (Huber cost as a minimum of a convex quadratic function). *The Huber cost function $\rho(t)$ of (3) is given by:*

$$\rho(t) = \min_{z \in R^1} \ \frac{1}{2}z^2 + \gamma|t - z|, \ \ t \in R^1. \tag{4}$$

Proof. Let $\theta(z, t) = \frac{1}{2}z^2 + \gamma|t - z|$. A subgradient [25] of this convex objective function of (4) with respect to z is given by:

$$\partial\theta(z, t) = \begin{cases} z - \gamma, & \text{when } \ t > z \\ z + \lambda\gamma, \ \ \lambda \in [-1, 1], & \text{when } \ t = z \\ z + \gamma, & \text{when } \ t < z. \end{cases} \tag{5}$$

This subgradient is zero when:

$$\begin{cases} z = \gamma, & \text{for } \ t > z = \gamma \\ z = -\lambda\gamma, & \text{for } \ t = z = -\lambda\gamma, \ \ \lambda \in [-1, 1] \\ z = -\gamma, & \text{for } \ t < z = -\gamma. \end{cases} \tag{6}$$

The zero-subgradient necessary and sufficient optimality condition [25] gives values of z at which the convex objective function of (4) attains a minimum for each $t \in R^1$. Evaluating this objective function at these values of z gives:

$$\begin{cases} \frac{1}{2}\gamma^2 + \gamma(t - \gamma) = \gamma t - \frac{1}{2}\gamma^2, & \text{when } \ t > \gamma \\ \frac{1}{2}t^2, & \text{when } \ t \in [-\gamma, \gamma] \\ \frac{1}{2}\gamma^2 + \gamma(-t - \gamma) = -\gamma t - \frac{1}{2}\gamma^2, & \text{when } \ t < -\gamma, \end{cases} \tag{7}$$

which is equivalent to the definition (3) of $\rho(t)$. □

This lemma allows us immediately to set up the linear estimation problem (2) with the Huber loss function (3) as the following problem:

$$\min_{x \in R^d, z \in R^\ell} \ \frac{1}{2}\|z\|_2^2 + \gamma\|Ax - b - z\|_1. \tag{8}$$

This problem in turn can be reduced to a simple convex quadratic program as follows.

Proposition 2.2 (Huber linear estimator as a convex QP). *A Huber linear estimator $x(\gamma)$ that solves (2) with ρ defined by (3) is given by any solution $(x(\gamma), z(\gamma), t(\gamma))$ of the following convex quadratic program:*

$$\begin{array}{ll} \min\limits_{x \in R^d, z \in R^\ell, t \in R^\ell} & \frac{1}{2}\|z\|_2^2 + \gamma e' t \\ \text{s.t.} & -t \le Ax - b - z \le t. \end{array} \tag{9}$$

Using this formulation, an immediate consequence of a perturbation result of linear programming [17] is the intuitively plausible result when all errors fall within γ, that for each linear estimation problem (1) there exists a positive $\bar{\gamma}$ such that every Huber estimator for all $\gamma \ge \bar{\gamma}$ is identical to the classical least 2-norm estimator. Hence, for all γ larger than this threshold, a classical least-squares solution also solves (1). We state this result as follows.

Proposition 2.3. (Huber M-estimator is a least squares estimator for all large γ). *For a given $A \in R^{\ell \times d}$ and $b \in R^{\ell \times 1}$, there exists a $\bar{\gamma}$ such that for all $\gamma \ge \bar{\gamma}$ a Huber M-estimator obtained by solving either of the equivalent problems (8) or (9) is equivalent to a classical least squares estimator that solves*

$$\min_{x \in R^d} \ \|Ax - b\|_2^2. \tag{10}$$

Proof. Theorem 1 of [17] says that under certain easily satisfiable technical assumptions (which are satisfied here), the solution of the perturbation of any solvable linear program is also a solution of the original linear program itself, as long as the perturbation is sufficiently small. More formally, there exists $\bar{\gamma}$ such that for each $\gamma \ge \bar{\gamma}$, each corresponding Huber estimate solution $(x(\gamma), z(\gamma))$ of (8) is a solution of

$$\min_{x \in R^d, z \in R^\ell} \ \|Ax - b - z\|_1, \tag{11}$$

that is, $z(\gamma) = Ax(\gamma) - b$, which also solves:

$$\min_{z \in R^\ell} \ \|z\|_2^2, \tag{12}$$

that is, $x(\gamma)$ solves (10). □

Perturbing γ in the other direction toward zero is more interesting. In fact, one can show that Huber M-estimators depend linearly on the parameter as the latter converges to zero, as shown in Proposition 2.4, below. However, an example [13, Example 3.8] shows that Huber M-estimators need not be least 1-norm estimators even for arbitrarily small values of the parameter γ. This is in contrast to Proposition 2.3 which states that for all sufficiently large γ Huber M-estimators are least 2-norm estimators.

Proposition 2.4. (Huber M-estimators converge linearly to a least 1-norm solution). *For every sequence $\{\gamma^i\}$ converging to zero, a corresponding subsequence of Huber M-estimators $\{x^{i_j}\}$ such that $\{x^{i_j}, z^{i_j}, t^{i_j}\}$ solve (9) for $\gamma = \gamma^{i_j}$ depends linearly on $\{\gamma^{i_j}\}$, that is for some p and q in R^d*

$$x^{i_j} = p + q\gamma^{i_j}, \ \ \{\gamma^{i_j}\} \to 0. \tag{13}$$

Proof. For $\gamma = \gamma^i$, the Karush-Kuhn-Tucker optimality conditions [15] for (9) are:

$$\begin{aligned} -A'(r-s) &= 0 \\ z+r-s &= 0 \\ r+s &= \gamma^i e \\ 0 \le r,\ \ Ax-z+t-b &\ge 0, \\ 0 \le s,\ \ -Ax+z+t+b &\ge 0, \\ r'(Ax-z+t-b)=0,\ \ s'(-Ax+z+t+b) &= 0.\ (*) \end{aligned} \tag{14}$$

For each γ^i, pick a basic solution [20, Lemma 2.1], [7, Theorem 2.11] which also satisfies the complementarity condition (*). Since there are a finite number of linearly independent columns in the linear system of (14), one set of linearly independent columns is used infinitely often by a basic solution corresponding to the sequence $\{\gamma^i\}$. Choose a subsequence $\{\gamma^{i_j}\}$ corresponding to this repeated basic solution. Hence, there exists a matrix M with $d+2\ell+\ell_1+\ell_2$ rows ($\ell_1, \ell_2 \le \ell$) and linearly independent columns corresponding to basic components y_B (typically nonzero) of $y := (x,\ z,\ t,\ r,\ s)$, such that:

$$My_B = \begin{bmatrix} 0 \\ 0 \\ \gamma^{i_j} e \\ b \\ -b \end{bmatrix}. \tag{15}$$

It follows then that:

$$y_B = (M'M)^{-1}M' \begin{bmatrix} 0 \\ 0 \\ \gamma^{i_j} e \\ b \\ -b \end{bmatrix} \tag{16}$$

from which (13) follows. □

It is interesting to note that Li and Swetits [13] have established a related result, Lipschitz continuity, for the dual of the quadratic Huber M-estimator problem (9) which turns out to be:

$$\min_u \left\{ \frac{\gamma}{2}\|u\|_2^2 + b'u \mid A'u = 0,\ -e \le u \le e \right\}. \tag{17}$$

They show that the solution $u(\gamma)$ of this dual problem, which is a least 2-norm solution of the dual of the least 1-norm problem $\min_x \|Ax-b\|_1$, is Lipschitzian with respect to γ.

Another interesting observation is that if the square of the 2-norm in (17) is replaced by the 1-norm as follows:

$$\min_u \left\{ \gamma\|u\|_1 + b'u \mid A'u = 0,\ -e \le u \le e \right\}, \tag{18}$$

then the linear programming dual of (18) is:

$$-\min_{x,z} \left\{ \|Ax-b-z\|_1 \mid \|z\|_\infty \le \gamma \right\}. \tag{19}$$

The loss function corresponding to this problem is essentially that of [18], [30], which is zero in the interval $[-\gamma, \gamma]$, and otherwise linear. Thus, this loss function replaces the quadratic part of the Huber loss function by zero. Such a loss function was also studied in [32].

Finally, we also note that Smola [29], [31] uses rather different techniques to derive a dual formulation of the quadratic Huber M-estimator problem (9). Smola's formulation can be stated as follows, after removing a suppression term:

$$\begin{array}{ll} \min\limits_{u,v} & \frac{\gamma}{2}(\|u\|_2^2 + \|v\|_2^2) + b'(u-v) \\ \text{s.t.} & A'(u-v) = 0 \\ & 0 \le u, v \le e, \end{array} \tag{20}$$

which is equivalent to the Li-Swetits dual formulation (17) above. It is this formulation of Smola's that we use in our comparative experiments.

3 Robust Nonlinear Support Vector Regression as a Convex Quadratic Program

In order to apply the Huber loss function to nonlinear regression problems using kernel functions [33], [16], we follow an approach similar to that of [27], [26], [19] where loss functions other than the Huber one were utilized. The idea is to implicitly transform the data of the problem from the given *input space* into a higher dimensional *feature space*, and perform linear regression in that space. This then corresponds to a nonlinear regression surface in the original input space. For that purpose, we make a variable transformation in our system of linear equations (1):

$$x = A'\alpha,\ \ \alpha \in R^\ell. \tag{21}$$

Corresponding to this transformation our convex quadratic program of the Huber linear estimator becomes:

$$\begin{array}{ll} \min\limits_{\alpha \in R^\ell, z \in R^\ell, t \in R^\ell} & \frac{1}{2}\|z\|_2^2 + \gamma e't \\ \text{s.t.} & -t \le AA'\alpha - b - z \le t. \end{array} \tag{22}$$

We observe that problem (22) depends only on a matrix consisting of scalar products of different rows in A, i.e. AA'. This immediately leads to the idea of replacing the kernel AA' for a linear estimator by a much more general kernel $K(A,A') : R^{\ell \times d} \times R^{d \times \ell} \to R^{\ell \times \ell}$. Under certain conditions, this kernel is a surrogate for mapping the data into a higher dimensional space and performing the scalar products there. The use of a kernel function, therefore, allows an implicit mapping into a higher dimensional space while saving significant computational costs. This concept is described in much more detail in some of the support vector machine literature [33], [4], [16]. Substituting AA' for $K(A,A')$ leads to the following convex quadratic program for a Huber nonlinear estimator:

$$\begin{array}{ll} \min\limits_{\alpha \in R^\ell, z \in R^\ell, t \in R^\ell} & \frac{1}{2}\|z\|_2^2 + \gamma e't \\ \text{s.t.} & -t \le K(A,A')\alpha - b - z \le t. \end{array} \tag{23}$$

We determine the predicted value for a new data point p by the calculation $K(p',A')\alpha$. As such, the regression surface depends strongly on the training matrix A in precisely the same fashion as a classification or regression surface does in traditional support vector machines. We do point out that the matrix $K(A,A')$ has ℓ by ℓ nonzero elements. If the dataset is large, i.e., ℓ is large, than this problem is

TABLE 1
Comparison of Algorithms for Robust Linear Regression

		$\gamma = 0.1$		$\gamma = 1$		$\gamma = 1.345$	
Dataset	**Algorithm**	**Time (sec)**	**Iters**	**Time (sec)**	**Iters**	**Time (sec)**	**Iters**
CPU Small	Mangasarian/Musicant QP	**7.46**	12.4	**6.26**	10.1	6.37	10.0
	Smola Dual QP	10.63	18.8	7.08	11.9	6.98	11.7
	Huber Gauss-Seidel	50.89	1210.7	6.44	147.9	**5.24**	112.6
	Madsen/Nielsen Newton	160.08	6.2	44.90	4.3	18.31	3.5
	Li Conjugate Gradient	178.01	172.3	92.26	92.3	99.87	102.0
Census 1000	Mangasarian/Musicant QP	**0.58**	11.0	**0.48**	9.4	**0.54**	8.9
	Smola Dual QP	0.93	20.0	0.62	13.0	0.60	12.4
	Huber Gauss-Seidel	7.24	1838.5	1.46	279.8	0.97	217.0
	Madsen/Nielsen Newton	2.02	10.5	1.01	4.9	0.89	4.6
	Li Conjugate Gradient	2.52	24.9	11.68	115.3	5.71	57.1
Census 5000	Mangasarian/Musicant QP	**3.63**	12.9	**2.82**	9.9	**2.82**	9.7
	Smola Dual QP	6.90	22.9	4.46	14.5	4.36	14.1
	Huber Gauss-Seidel	37.74	1745.8	5.30	195.8	3.58	147.8
	Madsen/Nielsen Newton	34.93	9.3	18.96	5.5	14.86	5.2
	Li Conjugate Gradient	11.73	21.8	26.99	51.4	48.02	94.2
Census 10000	Mangasarian/Musicant QP	**7.99**	13.5	**6.42**	10.2	**6.16**	9.9
	Smola Dual QP	16.63	24.9	10.95	15.8	10.51	15.1
	Huber Gauss-Seidel	124.27	2522.6	10.89	215.7	8.24	158.8
	Madsen/Nielsen Newton	132.32	8.7	75.05	5.5	56.35	5.2
	Li Conjugate Gradient	36.19	34.2	92.74	86.7	113.56	105.2
Census 20000	Mangasarian/Musicant QP	**18.48**	14.0	**14.66**	10.5	**16.64**	10.2
	Smola Dual QP	39.21	28.6	26.13	17.7	25.23	16.8
	Huber Gauss-Seidel	305.79	2398.8	27.04	202.4	23.78	154.1
	Madsen/Nielsen Newton	516.76	8.0	318.45	5.4	254.78	5.2
	Li Conjugate Gradient	68.89	31.4	151.94	39.5	231.08	99.9

Best times, in bold type, were achieved by our simple quadratic programming formulation in all cases except one.

significantly larger than the linear formulation, which has ℓ by d nonzero elements in the matrix A. For a large dataset, the linear problem (9) will therefore solve dramatically faster than the nonlinear formulation (23). However, our experimental results in Section 4 show that nonlinear kernels can yield higher test set accuracies. Kernels have also been used in Smola's dual QP method. We note that our description of kernels here follows directly from our formulation (9), and is noticeably different from the approach taken in [31].

4 Numerical Tests and Results

We first show the effectiveness of our formulation (9) of the Huber linear estimator as a QP by comparing it to four other algorithms to solve the same problem, namely:

- Smola's dual QP method [29]
- Huber's Gauss-Seidel method [12]
- Madsen and Nielsen's Newton type method [14]
- Li's conjugate gradient method [12].

In all cases except the last one, a termination tolerance of 10^{-5} was used. Li's conjugate gradient method showed extremely long termination times, so we reduced the tolerance for this algorithm to 10^{-3}. The primary purpose of these experiments is to examine differences in training times. Training and testing set accuracies are not of importance in these experiments, since all these algorithms in theory converge to a solution of the same QP, thus leading to similar training and testing set correctness. Inconsequential differences in accuracies arise in practice due to the fact that each algorithm yields a slightly different approximate solution. Additionally, it is possible that different algorithms could converge to different solutions if there are multiple solutions.

We note that in implementing our algorithm, we used the following equivalent variation of the QP given in (9):

$$\begin{array}{ll} \min\limits_{x\in R^d, z\in R^\ell, r\in R^\ell, s\in R^\ell} & \frac{1}{2}\|z\|_2^2 + \gamma e'(r+s) \\ \text{s.t.} & Ax - b - z = r - s \\ & r, s \geq 0. \end{array} \tag{24}$$

Formulation (24) yielded faster running times than a straightforward implementation of (9).

All experiments were run on the University of Wisconsin Computer Sciences Department Ironsides cluster. This cluster of four Sun Enterprise E6000 machines consists of 16 UltraSPARC II 250 MHz processors and two gigabytes of RAM on each node resulting in a total of 64 processors and eight gigabytes of RAM. We implemented data I/O, cross-validation procedures, and kernel calculations in the MATLAB environment [21], though all algorithms were

TABLE 2
Comparison of Robust Linear and Nonlinear Regression

Dataset	Kernel	ν	Training Accuracy	Testing Accuracy	Time (cpu secs)
CPU	Linear		94.50%	94.06%	0.21
Small	Gaussian	0.5	97.26%	95.90%	44.56
Boston	Linear		85.60%	83.81%	0.18
Housing	Gaussian	1	92.36%	88.15%	36.5

implemented in C++ using a MATLAB executable "mex" file [22]. This methodology allows us to call the algorithms from MATLAB as if they were native functions. All QPs in our algorithm were solved with the state-of-the-art CPLEX solver [11].

To demonstrate the performance of these algorithms, we performed tenfold cross validation of these algorithms on two datasets. The first dataset, Census, is a version of the US Census Bureau "Adult" dataset, which is publicly available from Silicon Graphics' website [1]. This dataset contains nearly 300,000 data points with 11 numeric attributes, and is used for predicting income levels based on census attributes. We ran the algorithms on varying subsets of this dataset. The second dataset, Comp-Activ, was obtained from the Delve website [5]. This dataset contains 8,192 data points and 25 numeric attributes. We implemented the "cpuSmall prototask," which involves using twelve of these attributes to predict what fraction of a CPU's processing time is devoted to a specific mode ("user mode").

We scaled the data by dividing by the following factor τ, suggested by Holland and Welsh in [8]:

$$\tau = 1.48 \cdot \mathrm{med}_i\{|(A\hat{x} - b)_i - \mathrm{med}_j\{(A\hat{x} - b)_j\}|\}, \qquad (25)$$

where $\hat{x}$ is an estimate of the solution x. We determined $\hat{x}$ by taking a small sample of the data and doing a standard least-squares regression. The factor 1.48 provides an approximately unbiased estimate of scale when the error model is Gaussian [8].

We also varied the parameter γ as specified in (3). Holland and Welsh recommend $\gamma = 1.345$, which satisfies certain statistical properties [8]. We have also used $\gamma = 1$ and $\gamma = 0.1$. We note that these smaller values of γ provide a closer approximation to a traditional 1-norm regression problem.

Table 1 shows the comparison of the various algorithms. In nearly all cases, our algorithm shows faster running times than the other algorithms. The differences in running times between our algorithm and the others become more pronounced for the larger versions of the Census dataset. In comparison to the other algorithms, Smola's dual QP also performs reasonably. Huber's Gauss-Seidel method performs well for large γ, but deteriorates significantly for small γ. Madsen and Nielsen's method, while very low in the number of iterations, runs somewhat slowly. Li's conjugate gradient method appears somewhat in the middle, though it should be remembered that a less stringent termination tolerance was used. The experimental success of our method demonstrates the value in representing the Huber regression problem as a simple quadratic program. By doing so, we can leverage the time and effort that has been put into high-powered and well-tweaked software tools like CPLEX to solve the problem. Some of the other approaches require software tailored to fit the problem. We also note that while other lesser-performing solvers might yield slower running times on our formulation than that yielded by CPLEX, the fundamental result here is that our formulation is a general quadratic program that can be used in any solver. As solver technology improves, solution time for our method will continue to improve. This is not the case for the specialized programs required by some of the other algorithms, which are not likely to see the same effort in improving their efficiencies.

Our second set of experiments is designed to show the effectiveness of using a kernel function in performing nonlinear robust regression. We used the Gaussian radial basis kernel [33], [2] in our experiments, namely

$$[K(A, A')]_{i,j} = \exp(-\nu\|A_i - A_j\|_2^2), i, j = 1, \ldots, \ell, \qquad (26)$$

where ν is a small positive parameter. These experiments take significantly longer to run than the first set, as the kernel function enlarges the size of the problem. We therefore ran these experiments for a subset of the CPUSmall task containing only 500 data points, and for the Boston Housing dataset (containing 506 points) available at the UCI machine learning repository [24]. The results of these experiments are shows in Table 2. In both cases, the Gaussian kernel resulted in improved testing set accuracy over the linear one. This demonstrates that robust regression can be used for finding complex nonlinear regression surfaces.

5 Conclusion

A new methodology for solving the Huber M-estimator problem is presented which reduces the problem to a simple quadratic program. This formulation is shown to perform well when compared to other algorithms in the literature. A modification of this quadratic program is introduced to allow kernel functions to be used. These kernel functions are used to find nonlinear regression surfaces which can yield better testing set performances than a purely linear separating surface. Future work includes using chunking methodologies to solve massive versions of these problems, as well as considering parallel solutions to these massive problems.

ACKNOWLEDGMENTS

The research described in this Data Mining Institute Report 99-09, November 1999, was supported by the US National Science Foundation grants CCR-9729842 and CDA-9623632, by Air Force Office of Scientific Research grant F49620-97-1-0326, and by the Microsoft Corporation.

REFERENCES

[1] "Adult Dataset," *U.S. Census Bureau*, publicly available from: www.sgi.com/Technology/mlc/db/.
[2] V. Cherkassky and F. Mulier, *Learning from Data—Concepts, Theory, and Methods.* New York: John Wiley & Sons, 1998.
[3] D.I. Clark and M.R. Osborne, "Finite Algorithms for Huber's M-Estimator," *SIAM J. Scientific and Statistical Computing,* vol. 7, pp. 72-85, 1986.
[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines.* Cambridge: Cambridge Univ. Press, 2000.
[5] "Delve:Data for Evaluating Learning in Valid Experiments," http://www. cs. utoronto. ca/~delve/.
[6] H. Ekblom, "A New Algorithm for the Huber Estimator in Linear Models," *BIT,* vol. 28, pp. 123-132, 1988.
[7] D. Gale, *The Theory of Linear Economic Models.* New York: McGraw-Hill, 1960.
[8] P.W. Holland and R.E. Welsch, "Robust Regression Using Iteratively Reweighted Least Squares," *Comm. Statistics—Theory and Methods,* vol. A6, pp. 813-827, 1977.
[9] P.J. Huber, *Robust Statistics.* New York: John Wiley, 1981.
[10] P.J. Huber and R. Dutter, "Numerical Solution of Robust Regression Problems," *Proc. Symp. Computational Statistics,* G. Brushmann, ed., pp. 165-172, 1974.
[11] *ILOG CPLEX 6. 5 Reference Manual,* ILOG CPLEX Division, Incline Village, Nev., 1999
[12] W. Li, "Numerical Estimates for the Huber M-Estimator Problem," *Approximation Theory VIII,* C.K. Chui and L.L. Schumaker, eds., pp. 325-334, New York: World Scientific Publishing, 1995.
[13] W. Li and J.J. Swetits, "The Linear ℓ_1 Estimator and the Huber M-Estimator," *SIAM J. Optimization,* vol. 8, pp. 457-475, 1998.
[14] K. Madsen and H.B. Nielsen, "Finite Algorithms for Robust Linear Regression," *BIT,* vol. 30, pp. 682-699, 1990.
[15] O.L. Mangasarian, *Nonlinear Programming,* Philadelphia: SIAM, 1994.
[16] O.L. Mangasarian, "Generalized Support Vector Machines," *Advances in Large Margin Classifiers,* A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds., pp. 135-146, Cambridge, Mass: MIT Press, 2000. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.
[17] O.L. Mangasarian and R.R. Meyer, "Nonlinear Perturbation of Linear Programs," *SIAM J. Control and Optimization,* vol. 17, no. 6, pp. 745-752, Nov. 1979.
[18] O.L. Mangasarian and D.R. Musicant, "Data Discrimination via Nonlinear Generalized Support Vector Machines," Technical Report 99-03, Computer Sciences Dept., Univ. Wisconsin, Madison, Mar. 1999. To appear in: *Applications and Algorithms of Complementarity,* M.C. Ferris, O.L. Mangasarian, and J.-S. Pang, eds., Boston: Kluwer Academic Publishers, 2000. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/99-03. ps.
[19] O.L. Mangasarian and D.R. Musicant, "Massive Support Vector Regression," Technical Report 99-02, Data Mining Institute, Computer Sciences Dept., Univ. Wisconsin, Madison, July 1999. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-02.ps.
[20] O.L. Mangasarian and T.-H. Shiau, "Lipschitz Continuity of Solutions of Linear Inequalities, Programs and Complementarity Problems," *SIAM J. Control and Optimization,* vol. 25, no. 3, pp. 583-595, May 1987.
[21] "MATLAB," *User's Guide,* Natick, Mass.: The MathWorks, Inc., 1992.
[22] "MATLAB," *Application Program Interface Guide,* Natick, Mass.: The MathWorks, Inc., 1997.
[23] C. Michelot and M.L. Bougeard, "Duality Results and Proximal Solutions of the Huber M-Estimator Problem," *Applied Math. and Optimization,* vol. 30, pp. 203-221, 1994.
[24] P.M. Murphy and D.W. Aha, "UCI Repository of Machine Learning Databases," 1992. www.ics.uci.edu/~mlearn/MLRepository.html.
[25] B.T. Polyak, *Introduction to Optimization,* Optimization Software, Inc., New York: Publications Division, 1987.
[26] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson, "Support Vector Regression with Automatic Accuracy Control," *Proc. Eighth Int'l Conf. Artificial Neural Networks,* L. Niklasson, M. Boden, and T. Ziemke, eds., pp.111-116,1998. http://svm.first.gmd.de.
[27] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson, "Shrinking the Tube: A New Support Vector Regression Algorithm," technical report, GMD FIRST, Berlin, Germany, 1999. http://svm.first.gmd.de.
[28] D.F. Shanno and D.M. Rocke, "Numerical Methods for Robust Regression: Linear Models." *SIAM J. Scientific and Statistical Computing,* vol. 7, pp. 86-97, 1986.
[29] A. Smola, "Regression Estimation with Support Vector Learning Machines," master's thesis, Technische Universität München, München, Germany, 1996.
[30] A. Smola, B. Schölkopf, and G. Rätsch, "Linear Programs for Automatic Accuracy Control in Regression," technical report, GMD FIRST, Berlin, Germany, 1999. http://svm.first.gmd.de/.
[31] A.J. Smola, "Learning with Kernels," PhD thesis, Technische Universität Berlin, Germany, 1998.
[32] W.N. Street and O.L. Mangasarian, "Improved Generalization via Tolerant Training," *J. Optimization Theory and Applications,* vol. 96, no. 2, pp. 259-279, Feb. 1998. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-11.ps.
[33] V.N. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer, 1995.
[34] S.J. Wright, "Using Complementarity and Optimization Methods in Statistics," *Int'l Conf. Complementarity Problems,* June 1999. Madis
[35] S.J. Wright, "On Reduced Convex QP Formulations of Monotone LCP Problems," Technical Report ANL/MCS-P808-0400, Argonne Nat'l Laboratory, Apr. 2000.

Olvi L. Mangasarian received the PhD degree in applied mathematics from Harvard University and worked for eight years as a mathematician for Shell Oil Company in California before coming to the University of Wisconsin, Madison. He is now the John von Neumann Professor of Mathematics and Computer Sciences and co-director of the Data Mining Institute of the Computer Sciences Department. His main research interests are in mathematical programming, machine learning, and data mining. He is the author of the book, *Nonlinear Programming*, he is the coeditor of four books and associate editor of three journals. His recent papers are available at http://www.cs.wisc.edu/~olvi and http://www.cs.wisc.edu/dmi.

David R. Musicant received BS degrees in both mathematics and physics from Michigan State University, then earned an MA degree in mathematics and an MS degree in computer sciences from the University of Wisconsin-Madison. He recently obtained the PhD degree in computer sciences at the University of Wisconsin, and is currently an assistant professor at Carleton College. He spent three years in the consulting industry as a technical operations research consultant for ZS Associates, and as a senior consultant for Icon InfoSystems, both in Chicago. His research interests involve applying data mining techniques to massive datasets and his recent papers are available at http://www.cs.wisc.edu/~musicant.