Breast Tumor Susceptibility to Chemotherapy via Support Vector Machines *

Glenn M. Fung †and Olvi L. Mangasarian‡

Abstract

Support vector machines (SVMs), utilizing RNA signature measurements, were used to generate a classifier to distinguish breast cancer patients that are partial-responders to chemotherapy treatment, from patients that are nonresponders. Partial responders are patients whose tumors were reduced by at least 50%. A stand-alone linear-programming-based SVM algorithm was used to separate the partial-responders from the nonresponders. A novel aspect of the classification approach utilized here is that each patient is represented by multiple points (*replicates*) in the 25-dimensional input space of RNA signature measurements. Replicates for all patients except those for one patient, were used as a training set. The *average* of the replicates for the patient left out was then used to test the *leave one out correctness (looc)*. The *looc* for a group of 35 patients, with 9 partial-responders and 26 nonresponders was 94.2%, in an input space of 5 RNA measurements extracted from an original space of 25 RNA measurements.

Keywords Support vector machines, breast cancer, chemotherapy, DNA macroarrays

1 Introduction

Support vector machines (SVMs) [19, 4] constitute the method of choice for classification problems. SVMs have been applied to a great variety of real world problems including breast cancer diagnosis [17] and prognosis [12, 13]. In this work we apply a recently developed fast Newton algorithm [8] that generates an SVM classifier for the problem of identifying a class of breast cancer patients that may benefit from chemotherapy. The input space for these patients consists of points in a 25-dimensional space of DNA macroarray measurements of RNA signatures [2]. Since most patients in this dataset have three sets of

*Research supported by NSF Grant CCR-0138308.

† *gfung@cs.wisc.edu.* Computer Sciences Department, University of Wisconsin, Madison, WI 53706.

‡ *olvi@cs.wisc.edu.* Computer Sciences Department, University of Wisconsin, Madison, WI 53706, and Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.

25-dimensional DNA macroarray measurements, referred to collectively as *replicates*, the *average* of these replicates is typically used to represent such a patient. In our present approach however, we have treated the convex hull of points representing the replicates for each patient as a “knowledge set”, similar to the polyhedral knowledge sets of [9, 10]. Our linear classifier then attempts to place each convex hull of points representing a nonresponder or a partial-responder in the appropriate halfspace generated by the linear classifier.

This work is organized as follows. In Section 2 we briefly describe the general classification problem and our Newton method for solving it. Section 3 describes the chemotherapy problem that we are concerned with. Section 4 details how we generated and evaluated our SVM classifier. Section 5 concludes the paper.

A word about our notation. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. For a vector x in the n-dimensional real space R^n, the *plus function* x_+ is defined as $(x_+)_i = \max\{0, x_i\}$, $i = 1, \dots, n$, while x_* denotes the subgradient of x_+ which is the step function defined as $(x_*)_i = 1$ if $x_i > 0$, $(x_*)_i = 0$ if $x_i < 0$, and $(x_*)_i \in [0, 1]$ if $x_i = 0$, $i = 1, \dots, n$. Thus, $(x_*)_i$ is any value in the interval $[0, 1]$, when $x_i = 0$, and we typically take $(x_*)_i = 0.5$ in this case. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$, the 2-norm of x will be denoted by $\|x\|_2$. The 1-norm and ∞-norm will be denoted by $\|\cdot\|_1$ and $\|\cdot\|_\infty$ respectively. For a matrix $A \in R^{m\times n}$, A_i is the ith row of A which is a row vector in R^n. A column vector of ones of arbitrary dimension will be denoted by e and the identity matrix of arbitrary order will be denoted by I. If f is a real valued function defined on the n-dimensional real space R^n, the gradient of f at x is denoted by $\nabla f(x)$ which is a column vector in R^n. The $n \times n$ matrix of second partial derivatives of f at x is denoted by $\nabla^2 f(x)$. For a piecewise quadratic function such as, $f(x) = \frac{1}{2}||(Ax - b)_+||^2 + \frac{1}{2}x'Px$, where $A \in R^{m\times n}$, $P \in R^{n\times n}$, $P = P'$, P positive semidefinite and $b \in R^m$, the ordinary Hessian does not exist because its gradient, the $n\times 1$ vector $\nabla f(x) = A'(Ax-b)_+ + Px$, is not differentiable. However, one can define its **generalized Hessian** [11, 6, 16] which is the $n \times n$ symmetric positive semidefinite matrix:

$$\partial^2 f(x) = A' diag(Ax - b)_* A + P, \tag{1}$$

where $diag(Ax - b)_*$ denotes an $m \times m$ diagonal matrix with diagonal elements $(A_i x - b_i)_*$, $i = 1, \dots, m$. The generalized Hessian (1) has many of the properties of the regular Hessian [11, 6, 16] in relation to $f(x)$.

2 The Linear SVM Classifier and the Newton Algorithm

We describe in this section the fundamental classification problem that leads to a linear programming formulation. We consider the problem of classifying m points in the n-dimensional real space R^n, represented by the $m \times n$ matrix A, according to membership of each point A_i in the classes +1 or -1 as specified by a given $m \times m$ diagonal matrix D with ones or minus ones along its diagonal. For this problem, a standard support vector machine with a linear classifier [15, 3] is given by the following mathematical program for some $\nu > 0$:

$$\begin{array}{rrcl} \min\limits_{(w,\gamma,y)} & \nu e'y + \|w\|_1 & & \\ \text{s.t.} & D(Aw - e\gamma) + y & \geq & e \\ & y & \geq & 0, \end{array} \tag{2}$$

where $\|\cdot\|_1$ denotes the 1-norm. As depicted in Figure 1, w is the normal to the bounding planes:

$$\begin{array}{rcrcl} x'w & - & \gamma & = & +1 \\ x'w & - & \gamma & = & -1, \end{array} \tag{3}$$

and γ determines their location relative to the origin. The first plane above bounds the class $+1$ points and the second plane bounds the class -1 points when the two classes are strictly linearly separable, that is when the slack variable $y = 0$. The linear separating surface is the plane

$$x'w = \gamma, \tag{4}$$

midway between the bounding planes (3). If the classes are linearly inseparable then the two planes bound the two classes with a "soft margin" determined by a nonnegative slack variable y, that is:

$$\begin{array}{ll} x'w - \gamma + y_i \geq +1, & \text{for } x' = A_i \text{ and } D_{ii} = +1. \\ x'w - \gamma - y_i \leq -1, & \text{for } x' = A_i \text{ and } D_{ii} = -1. \end{array} \tag{5}$$

The first term $\nu e'y$ in the objective function of (2) minimizes the 1-norm of the slack variable y. The second term in the objective function of (2), which is twice the reciprocal of the ∞-norm distance $\frac{2}{\|w\|_1}$ [14] between the two bounding planes of (3) in the n-dimensional space of $w \in R^n$ for a *fixed* γ, maximizes that distance, often called the "margin". Figure 1 depicts the points represented by A, the bounding planes (3) with margin $\frac{2}{\|w\|_1}$, and the separating plane (4) which separates $A+$, the points represented by rows of A with $D_{ii} = +1$, from $A-$, the points represented by rows of A with $D_{ii} = -1$.

The SVM formulation (2) which is equivalent to the linear program (6), replaces the quadratic term $\frac{1}{2}w'w$ in a conventional SVM [4, 19] in the objective function by the term $\|w\|_1$. Empirical evidence [3] indicates that the 1-norm formulation has the advantage of generating very sparse solutions. This results in the normal w to the separating plane $x'w = \gamma$ having many zero components, which implies that many input space features do not play a role in determining the linear classifier. This makes this approach suitable for feature selection in classification problems. We note that in addition to the conventional interpretation of smaller ν as emphasizing a larger margin between the bounding planes (3), a smaller ν here also results in a sparse solution. This 1-norm formulation

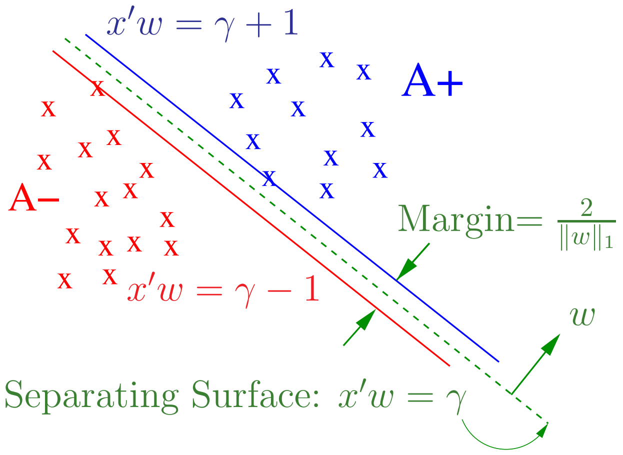

Figure 1: **The bounding planes $x'w = \gamma \pm 1$ with margin $\frac{2}{\|w\|_1}$, and the plane $x'w = \gamma$ separating $A+$, the points represented by rows of A with $D_{ii} = +1$, from $A-$, the points represented by rows of A with $D_{ii} = -1$.**

leads to the linear programming problem:

$$\begin{array}{rrcl} \displaystyle\min_{(p,q,\gamma,y)} & \nu e'y + e'(p+q) & & \\ \text{s.t.} & D(A(p-q) - e\gamma) + y & \geq & e \\ & p, q, y & \geq & 0, \end{array} \tag{6}$$

where the following substitution for w has been made:

$$w = p - q, \ \ p \geq 0, \ q \geq 0, \tag{7}$$

This is a different and a simpler linear program from previous linear programming SVM formulations [15, 3]. The dual of the linear program (6) is the following:

$$\begin{array}{rrcccl} & & \displaystyle\max_{u \in R^m} & e'u & & \\ \text{s.t.} & -e & \leq & A'Du & \leq & e, \\ & & & -e'Du & = & 0, \\ & & & u & \leq & \nu e, \\ & & & u & \geq & 0. \end{array} \tag{8}$$

The asymptotic exterior penalty problem [7, 1] for this dual linear program is the following nonnegatively constrained minimization problem:

$$\begin{array}{rl} \displaystyle\min_{u \geq 0} & -\epsilon e'u + \frac{1}{2}\|(A'Du - e)_+\|^2 + \\ \frac{1}{2}\|(-A'Du - e)_+\|^2 & +\frac{1}{2}\|e'Du\|^2 + \frac{1}{2}\|(u - \nu e)_+\|^2. \end{array} \tag{9}$$

Here, ϵ is a positive penalty parameter that needs to approach zero for standard penalty function methods [1] for solving the dual linear program (8). *However*, it has been established in [8] that an exact solution to the primal linear program (6) or equivalently to (2), can be obtained for finite ϵ as follows.

Proposition 2.1 *[8, Proposition 2.1]***(Equivalence of Least 2-norm LPSVM to Dual Exterior Penalty)** *A solution u to the dual exterior penalty (DEP) problem (9) for $\epsilon \in (0, \bar{\epsilon}]$ for some $\bar{\epsilon} > 0$, provides an* **exact** *least 2-norm solution to primal LPSVM (6) as follows:*

$$\begin{array}{lcl} w = p - q = & = & \frac{1}{\epsilon}((A'Du - e)_+ - (-A'Du - e)_+), \\ \gamma & = & -\frac{1}{\epsilon}e'Du, \\ y & = & \frac{1}{\epsilon}(u - \nu e)_+. \end{array} \tag{10}$$

Based on this proposition all we need to do is solve the dual exterior penalty (9) for a finite value of the penalty parameter ϵ and by incorporating the non-negativity constraint $u \geq 0$ into the objective function of (9) as a penalty term as follows:

$$\begin{array}{ll} \min_{u} f(u) = & -\epsilon e'u + \frac{1}{2}\|(A'Du - e)_+\|^2 \\ & +\frac{1}{2}\|(-A'Du - e)_+\|^2 + \frac{1}{2}\|e'Du\|^2 \\ & +\frac{1}{2}\|(u - \nu e)_+\|^2 + \frac{\alpha}{2}\|(-u)_+\|^2. \end{array} \tag{11}$$

The gradient of this function is given by:

$$\begin{array}{rcl} \nabla f(u) & = & -\epsilon e + DA(A'Du - e)_+ - DA(-A'Du - e)_+ \\ & & +Dee'Du + (u - \nu e)_+ - \alpha(-u)_+, \end{array} \tag{12}$$

and its generalized Hessian as defined by (1) in the Introduction:

$$\begin{array}{rcl} \partial^2 f(u) & = & DA(diag((A'Du - e)_* + (-A'Du - e)_*)A'D \\ & & +Dee'D + diag((u - \nu e)_* + \alpha(-u)_*) \\ & = & DA(diag(|A'Du| - e)_*)A'D \\ & & +Dee'D + diag((u - \nu e)_* + \alpha(-u)_*), \end{array} \tag{13}$$

where the last equality follows from the equality:

$$(a - 1)_* + (-a - 1)_* = (|a| - 1)_*. \tag{14}$$

We are ready now to state our Newton algorithm for solving the primal linear program (6) or equivalently the mathematical program (2).

Algorithm 2.2 LPSVM Newton Algorithm for (9) *Let $f(u)$, $\nabla f(u)$ and $\partial^2 f(u)$ be defined by (11)-(13). Set the parameter values ν, ϵ, δ, tolerance* tol, *α and* imax *(typically: $\epsilon = 10^{-1}$, $tol = 10^{-3}$, $\alpha = 10^3$, $imax = 50$, while ν and δ are set by a tuning procedure described in Section 4). Start with any $u^0 \in R^m$. For $i = 0, 1, \ldots$:*

(I) $u^{i+1} = u^i - \lambda_i(\partial^2 f(u^i) + \delta I)^{-1}\nabla f(u^i) = u^i + \lambda_i d^i$,
where the Armijo stepsize $\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ is such that:

$$f(u^i) - f(u^i + \lambda_i d^i) \geq -\frac{\lambda_i}{4}\nabla f(u^i)'d^i, \tag{15}$$

and d^i is the modified Newton direction:

$$d^i = -(\partial^2 f(u^i) + \delta I)^{-1}\nabla f(u^i). \tag{16}$$

(II) Stop if $\|u^i - u^{i+1}\| \leq tol$ *or* $i = imax$. *Else, set* $i = i + 1$, $\alpha = 2\alpha$ *and go to* (I).

(III) Define the least 2-norm solution of the linear programming SVM (6) by (10) with $u = u^i$.

Convergence of this algorithm was established in [8, Theorem 3.1] together with very encouraging computational results. We shall apply this algorithm to generate our classifier in Section 4.

3 The Chemotherapy Classification Problem

The classification problem considered here is described in detail in [2]. The data for the problem is available at [5]. We briefly outline the problem now.

Thirty-five patients were diagnosed, by a core needle biopsy, as having breast cancer. Chemotherapy was administered to these patients before and after surgery. Of these 35 patients, 9 patients were *partial-responders*, that is their tumor size decreased by 50% or more after four cycles of chemotherapy, and 26 were *nonresponders*, that is their tumor size decreased by less than 50% after four cycles of chemotherapy. For each patient, a 25-gene expression profile was extracted from a macroarray of cancer-associated gene fragments [2].

The novel aspect of this dataset is that each patient is represented by one or more points in R^{25}. Each point represents the transcript of the 25-gene expression. Points associated with a single patient are called *replicates*. Most patients, 25 in this dataset, are represented by three replicates, 8 patients by two replicates and 2 patients by one replicate. Since each patient can be represented by either the convex hull of their replicates or, more typically by the average of these replicates, we decided to use the motivation of knowledge-based support vector machines [9, 10] and treat each convex hull as a knowledge set. Having such a knowledge set be on the correct side of a separating plane in the input space turns out to be equivalent to having all the replicates on that side of the plane. With this motivation in mind, we settled on the following, apparently novel leave-one-out-correctness (*looc*) measure. Leave the replicates representing one patient out, find a separating plane using the replicates of all other patients, test correctness on the *average* of the replicates for the patient left out. The motivation for this approach is a practical one. A patient who has a number of gene expression measurements, is in general classified on the basis of the average of these gene expressions. Thus when a new patient is encountered with a number of gene expression replicates, our classifier will classify that patient based on the average of the replicates with an expected correctness similar to that of the looc of the classifier obtained by our stand-alone LPSVM Newton Algorithm 2.2.

Figure 2 depicts a linear SVM classifier obtained in R^2 for a synthetic case consisting of ten nonresponders and five partial responders, each with various numbers of replicates between one and three. In this example the linear classifier misclassifies the average of the two replicates of the patient left out.

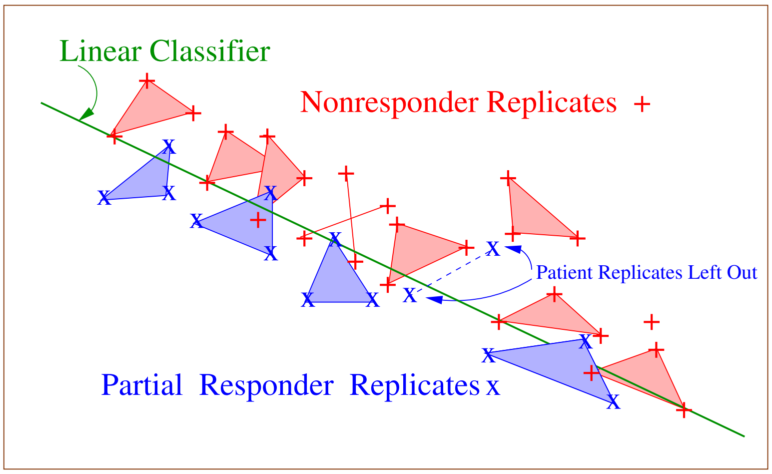

Figure 2: **Linear classifier for convex hulls of synthetic replicates in R^2 based on 10 nonresponder patients and 4 partial responder patients. The linear classifier generated here misclassifies the average of the two replicates for the patient left out who happened to be a partial responder.**

4 Generation and Evaluation of the SVM Classifier

We started our linear classifier computations by applying the LPSVM Algorithm 2.2 to the Chemotherapy Dataset [5] which consists of 93 replicate points in R^{25}. LPSVM immediately suppressed 11 of the 25 features and obtained a linear classifier in R^{14} with a *looc* of 80.0%. In order to improve this *looc* and further suppress features we utilized LPSVM in an incremental greedy approach on the remaining 14 features as follows. We used LPSVM to obtain 14 linear classifiers in R^1, one classifier for each of the 14 gene expression features selected by LPSVM in the first step. We kept the gene expression with the best *looc* classifier. We then used LPSVM to obtain 13 linear classifiers in R^2 by combining the feature we had kept with each of the remaining 13 features. We then kept the classifier in R^2 with the best *looc*. We continued this greedy approach until there was no further improvement in the *looc*, which took place when we went from R^5 to R^6 as depicted in Figure 3. Figure 3 plots the number of leave-one-out misclassifications versus the dimensionality of the input space, that is the number of gene expressions used. The smallest dimension achieving the best leave-one-out-error of 2, was dimension 5. This corresponds to a *looc* 94.2%.

All our computations were performed on the University of Wisconsin Data Mining Institute “locop1” machine, which utilizes a 400 Mhz Pentium II and allows a maximum of 2 Gigabytes of memory for each process. This computer runs a Windows NT Server 4.0, with MATLAB 6 installed [18]. We summarize

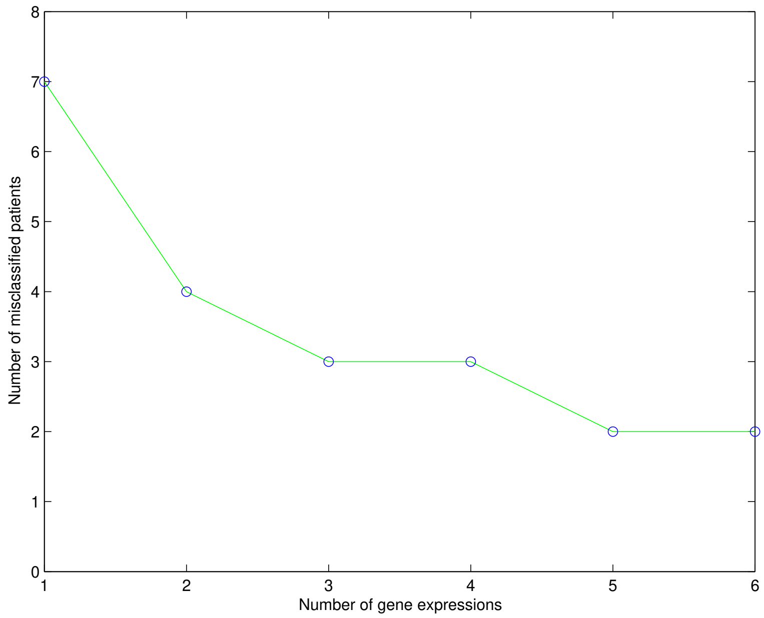

Figure 3: **Leave-one-out-error of the SVM classifier for 35 patients versus number of gene expressions used to generate the classifier.**

now the tuning procedure for the LPSVM Algorithm 2.2.

- A random tuning set consisting of 10% of the training data was chosen and separated from the training set.

- LPSVM was trained on the remaining 90% of the training data using values of ν equal to 2^i with $i = -7, \ldots, 0, \ldots, 7$, and δ equal to 10^j with $j = -3, \ldots, 0, \ldots, 3$. This resulted in a search grid for the pair (ν, δ) consisting of 15×7 points. The remaining parameters of LPSVM were set to the following values: $\epsilon = 10^{-1}, tol = 10^{-3}, \alpha = 10^3, imax = 50$.

- Values of ν and δ that gave the best SVM average replicate correctness on the tuning set were chosen.

- A final SVM was trained using the chosen values of ν, δ and all the training data including the tuning set. The resulting SVM was then tested on the average of the replicates of the one patient left out.

A single run of LPSVM for our problem in R^{25} takes about 0.2 seconds and about 0.09 seconds in R^5. MATLAB codes for LPSVM are available at http://www.cs.wisc.edu/dmi/svm/lpsvm/.

An example of a linear classifier obtained by LPSVM, is depicted in Figure 4 in an R^3 space of three gene expressions. All 93 replicates for the 35 patients were used in this visual illustration. Training set correctness for this example was 94.2%, as measured by the average replicates of two patients, out of 35, that were misclassified.

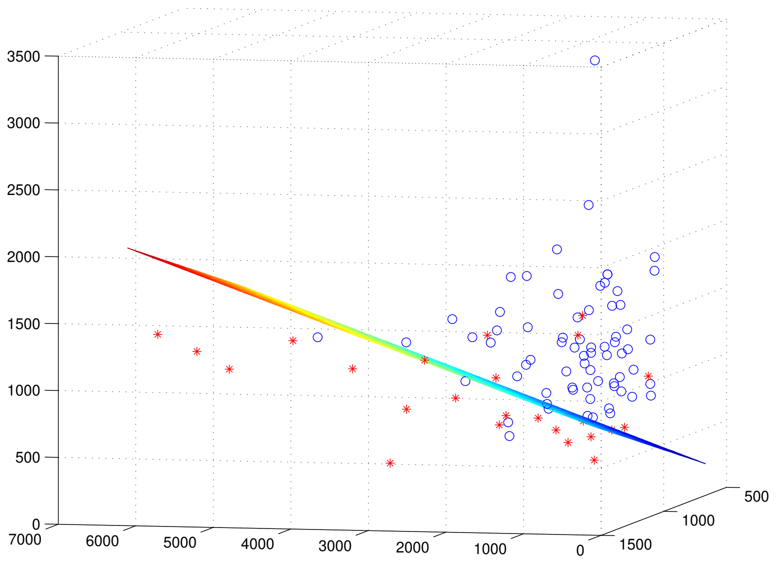

Figure 4: **A linear SVM classifier obtained by LPSVM in an R^3 space of three gene expressions using all 93 replicates for the 35 patients. Training set correctness for this example was 94.2%, as measured by the average replicates of two patients, out of 35, that were misclassified. Nonresponders are represented by circles and partial responders by asterisks.**

5 Conclusion

We have proposed a knowledge-based approach for classifying breast cancer patients as partial-responders or nonresponders to chemotherapy. The support vector machine classifier is based on multiple measurements of 25 gene expressions for each patient. Of these 25 gene expressions, only 5 were needed in order to achieve a 94.2% leave-one-out-correctness by a linear SVM classifier. The classifier was generated in a 5-dimensional space of 5 gene expressions by *all* replicates of all patients except one. The classifier was tested on the average of replicates for the patient left out. The novel approach utilized here is that the convex hull of the replicates, treated as a knowledge set, is required to be on the correct side of the classifier. Furthermore, the stand-alone Newton algorithm employed here, efficiently generates the classifier while suppressing many of the unnecessary features for accurate classification. It is hoped that this combined approach will be useful in other applications where repeated measurements lead to different input space points associated with a single person or event.

Acknowledgments

We are indebted to Aram Mangasarian, ExonHit, Paris, who first pointed out the applicability of SVM classification to this chemotherapy evaluation problem and for insightful suggestions. We are also indebted to Arnaud Zeboulon, ExonHit, for providing us with data for this problem and for helpful technical exchanges regarding this work, and to Laurent Bracco, ExonHit, for technical advice. Data for this problem was obtained in collaboration with the Curie Institute, Paris. Research described in this UW Data Mining Institute Report 03-06, November 2003, was supported by NSF Grant CCR-0138308 and by Microsoft.

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 1999.

[2] L. Bracco, B. M. N. Brinkman, P. de Cremoux, J.-M. Extra, J. Kearsey, H. Magdelénat, S. Pommey, X. Sastre, and A. Zeboulon. Molecular classification of chemotherapy resistant and sensitive breast tumors using an array containing alternative RNA processing events, 2003. Manuscript in preparation.

[3] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, pages 82–90, San Francisco, California, 1998. Morgan Kaufmann. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, MA, 2000.

[5] ExonHit. Chemotherapy Dataset. ExonHit Therapeutics, 65 Boulevard Masséna, 75013 Paris, France. ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/ExonHit/, 2002.

[6] F. Facchinei. Minimization of SC^1 functions and the Maratos effect. *Operations Research Letters*, 17:131–137, 1995.

[7] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York, NY, 1968.

[8] G. Fung and O. L. Mangasarian. A feature selection newton method for support vector machine classification. Technical Report 02-03, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, September 2002. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/02-03.ps. Computational Optimization and Applications, to appear.

[9] G. Fung, O. L. Mangasarian, and J. Shavlik. Knowledge-based support vector machine classifiers. Technical Report 01-09, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, November 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-09.ps, NIPS 2002 Proceedings, to appear.

[10] G. Fung, O. L. Mangasarian, and J. Shavlik. Knowledge-based nonlinear kernel classifiers. Technical Report 03-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 2003. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/02-03.ps. Conference on Learning Theory (COLT 03) and Workshop on Kernel Machines, Washington D.C., August 24-27, 2003. Proceedings edited by M. Warmuth and B. Schölkopf, Springer Verlag, Berlin, 2003, 102-113.

[11] J.-B. Hiriart-Urruty, J. J. Strodiot, and V. H. Nguyen. Generalized hessian matrix and second-order optimality conditions for problems with C^{L1} data. *Applied Mathematics and Optimization*, 11:43–56, 1984.

[12] Y.-J. Lee, O. L. Mangasarian, and W. H. Wolberg. Breast cancer survival and chemotherapy: a support vector machine analysis. Technical Report 99-10, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, December 1999. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Volume 55, 2000, 1-10. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-10.ps.

[13] Y.-J. Lee, O. L. Mangasarian, and W. H. Wolberg. Survival-time classification of breast cancer patients. Technical Report 01-03, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-03.ps. *Computational Optimization and Applications* 25, 2003, 151-166.

[14] O. L. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24:15–23, 1999. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-07r.ps.

[15] O. L. Mangasarian. Generalized support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 135–146, Cambridge, MA, 2000. MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.

[16] O. L. Mangasarian. A finite Newton method for classification problems. Technical Report 01-11, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, December 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-11.ps.*Optimization Methods and Software* 17, 2002, 913-929.

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

[18] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1994-2001. http://www.mathworks.com.

[19] A. Smola and B. Schölkopf. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.