Comp Sci 525 Computing Project
Breast Cancer Diagnosis via Quadratic Programming

Linear and quadratic programming can be used to solve problems in many applications. In this project, we will use quadratic programming for breast cancer diagnosis.

The project will use the Wisconsin Diagnosis Breast Cancer Database (WDBC) made publicly available by Dr. William H. Wolberg of the Department of Surgery of the University of Wisconsin Medical School, Professor W. Nick Street of the Management Sciences Department of the University of Iowa and Prof. O.L. Mangasarian of the Computer Sciences Department of the University of Wisconsin.

The database is available at various ftp sites. You should use the version that is available in the files wdbc.data and wdbc.names of the public directory ∼`cs525-1/public`. You should read the file wdbc.names.

Each patient in the database had a fine needle aspirate (FNA) taken from her breast. An example of a benign FNA and a malignant FNA are given in Figure 1. After this procedure, a computer program determined 30 (numerical) attributes that "represent" the FNA. Each line in the database gives these 30 values as attributes 3 to 32. The first attribute of each line is a patient identifier, while the second character ("B" or "M") represents the results of further analysis or surgery that determined if the FNA was benign or malignant. You may wish to think of each line in the data as representing a point $x \in \mathrm{R}^{30}$, and the collection of points as being separated into two sets called $\mathcal{B}$ (benign) and $\mathcal{M}$ (malignant).

The idea of the project is to come up with a discriminant function (a separating plane in this case) to determine if an unknown sample is benign or malignant. In order to do this, you will use part of the data in the above database as a "training set" to generate your separating plane and

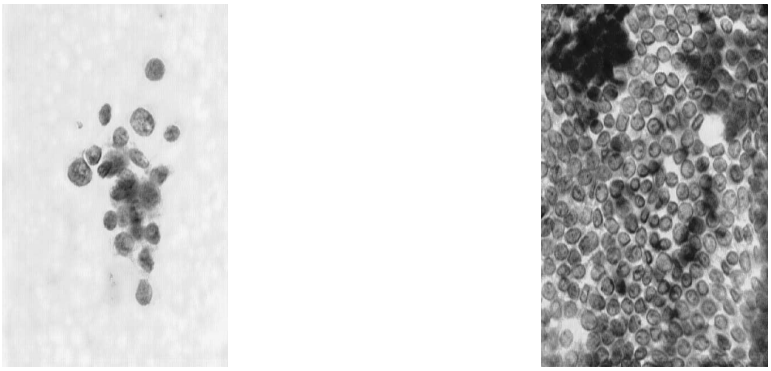

Figure 1: Nuclei of cells of malignant(left) and benign(right) fine needle aspirates taken from patients' breasts

the remaining part as a "testing" set to test your separating plane. The separating plane, to be determined by solving a single linear program in MATLAB is based on a formulation proposed in [1, 4]. Description of this project can also be found in [3]. A description of the linear program that generates the separating plane follows. The separating plane is subsequently used to determine whether new aspirates are benign or malignant.

A linear function f will be constructed which has the property:

$$f(x) > 0 \implies x \in \mathcal{M}, \qquad f(x) \leq 0 \implies x \in \mathcal{B},$$

to the extent possible. This function is given by $f(x) = w'x - \gamma$, and determines a plane $w'x = \gamma$ that separates to the extent possible malignant points from benign ones in R^{30}. It remains to show how to determine $w \in \mathrm{R}^{30}$ and $\gamma \in \mathrm{R}$ from the training data.

It we let the sets of m points $\mathcal{M}$ be represented by a matrix $M \in \mathrm{R}^{m \times n}$ and the set of k points $\mathcal{B}$ be represented by a matrix $B \in \mathrm{R}^{k \times n}$, then the problem becomes one of choosing w and γ to

$$\min_{w,\gamma} \frac{1}{m} \left\| (-Mw + e\gamma + e)_+ \right\|_1 + \frac{1}{k} \left\| (Bw - e\gamma + e)_+ \right\|_1$$

Here e is an appropriately dimensioned vector of all ones, $((z)_+)_i = \max\{z_i, 0\}$, $i = 1, 2, \ldots, m$ and $\|z\|_1 = \sum_{i=1}^{m} |z_i|$ for $z \in \mathrm{R}^m$. This problem approximately minimizes the number of points that are misclassified by choosing w and γ to minimize the sum of the distances to the separating plane whenever a point is on the incorrect side of the plane. The $(\cdot)_+$ and $\|\cdot\|$ functions can be eliminated by using the following linear programming reformulation;

$$\min_{w,\gamma,y,z} \left\{ \frac{1}{m} e'y + \frac{1}{k} e'z \mid Mw - e\gamma + y \geq e, -Bw + e\gamma + z \geq e, y \geq 0, z \geq 0 \right\}$$

If the datasets are separable, then this linear program may have multiple solutions. In order to choose an appropriate solution from these, typically w is chosen to maximize the "separation margin" between the two datasets. It can be shown that the separation margin is given by the reciprocal of $\|w\|$, so we modify the above formulation to add a multiple of the two-norm of w to the objective:

$$\begin{array}{ll} \min_{w,\gamma,y,z} & \left(\frac{1}{m} e'y + \frac{1}{k} e'z\right) + \frac{\mu}{2} w'w \\ \text{s.t.} & Mw - e\gamma + y \geq e, -Bw + e\gamma + z \geq e, y \geq 0, z \geq 0 \end{array}$$

This is the formulation you should attempt to implement and solve. (Alternative, linear programming formulations can be found in [2].)

On the CS departmental unix machines, a standard routine `cplexqp` written in MATLAB is provided for solving quadratic programs of the form

$$\min \left\{ c'x + \frac{1}{2}x'Qx \mid Ax \leq b, Hx = g, \bar{l} \leq x \leq \bar{u} \right\}$$

which can be called by using `[x,obj]=cplexqp(Q,c,A,b,H,g,lb,ub)`. Type `help cplexqp` within MATLAB to get more information about this quadratic programming routine. If the above command gives an error, then review the setup instructions for Matlab for this course since it should be available to everyone in the course.

The project consists of the following four parts:

1. Formulate the problem as a quadratic program. Solve the problem using the matrices M and B as a training set using the first 369 cases of the wdbc.data file (see field 2). The next 100 cases will be used as your tuning set, while the last 100 cases will be used as a testing set as indicated below. An example of an m-file that extracts the training, tuning and testing data as "matrices" from the original data file can be found in the public directory as `getdata.m`. This file also "whitens" the data by subtracting the mean from each feature and dividing through by its standard deviation. Try the value of $\mu = 0.0001$. **Make sure you print out w, γ and the minimum value of the QP.**

2. Test the separating plane on the 100 cases of the tuning set. Report the number of misclassified points on the tuning set. It is probably a good idea if you create an m-file to do this. What is the effect of μ? Now experiment with the values of $\mu = 5e-5, 1e-4, 1.5e-4, \ldots, 5e-4$. What is the best value of μ from this set? (Explain how you break ties). Use this best value of μ for the remainder of the project. What is the "testing" set error and the number of misclassifed points for this choice of μ?

3. Suppose that the oncologist wants to use only 2 of the 30 attributes in his diagnosis. Determine which pair of attributes is most effective in determining a correct diagnosis as follows. Use each of $\binom{30}{2} = 435$ pairs of possible attributes and for each pair determine from the training set

a separating plane in R^2. For each plane use the tuning set with the corresponding pair of attributes to determine the number of misclassified cases. Make sure you print out the number of misclassified points in the tuning set for each pair of attributes using

```
fprintf('atts %2d %2d: misclass %3d\n',i,j, wrong);
```

4. Use the best performing answer from Part 3 above (break ties using any procedure but specify it explicitly), find and print out the number of misclassified points on the testing set and plot all the testing set points on a two dimensional figure using MATLAB's built–in plotting routines Use 'o' for benign points and '+' for malignant points in the plot. Then use MATLAB to draw in the calculated line $w'x = \gamma$. Check to see if the number of misclassified points agrees with the plot and comment. (Note that some points may coalesce, so you may want to randomly perturb the points by a small amount to visualize all these points).

   Hand in your results and m-files.

References

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

[2] 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.

[3] M. C. Ferris and O. L. Mangasarian. Breast cancer diagnosis via linear programming. *IEEE Computational Science and Engineering*, 2:70–71, 1995.

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