CS 391D Data Mining: A Mathematical Perspective Fall 2009

Solutions to Homework 3

Lecturer: Inderjit Dhillon Date Due: October 27, 2009

Keywords: *Classification, Logistic Regression, Perceptron, Support Vector Machines*

1. (4 points) Note that $\frac{d\sigma(\boldsymbol{w}^T\boldsymbol{x}_n)}{d\boldsymbol{w}} = \sigma(\boldsymbol{w}^T\boldsymbol{x}_n)(1-\sigma(\boldsymbol{w}^T\boldsymbol{x}_n))\boldsymbol{x}_n$. Thus,

$$\frac{d\log\sigma(\boldsymbol{w}^T\boldsymbol{x}_n)}{d\boldsymbol{w}} = \frac{1}{\sigma(\boldsymbol{w}^T\boldsymbol{x}_n)}\frac{d\sigma(\boldsymbol{w}^T\boldsymbol{x}_n)}{d\boldsymbol{w}} = (1-\sigma(\boldsymbol{w}^T\boldsymbol{x}_n))\boldsymbol{x}_n, \tag{1}$$

$$\frac{d\log(1-\sigma(\boldsymbol{w}^T\boldsymbol{x}_n))}{d\boldsymbol{w}} = \frac{-1}{1-\sigma(\boldsymbol{w}^T\boldsymbol{x}_n)}\frac{d\sigma(\boldsymbol{w}^T\boldsymbol{x}_n)}{d\boldsymbol{w}} = -\sigma(\boldsymbol{w}^T\boldsymbol{x}_n)\boldsymbol{x}_n. \tag{2}$$

Using (1) and (2),

$$\bigtriangledown_w E(\boldsymbol{w}) = \sum_n (z_n - y_n)\boldsymbol{x}_n,$$

where $z_n = \sigma(\boldsymbol{w}^T\boldsymbol{x}_n)$. Now, Hessian of $E(\boldsymbol{w})$ w.r.t $\boldsymbol{w}$ is given by:

$$H = \bigtriangledown_w^2 E(\boldsymbol{w}) = \sum_n z_n(1-z_n)\boldsymbol{x}_n\boldsymbol{x}_n^T$$

Note that $0 < z_n < 1$. Thus, for any u

$$\boldsymbol{u}^T H\boldsymbol{u} = \sum_n z_n(1-z_n)(\boldsymbol{u}^T\boldsymbol{x}_n)^2 \geq 0. \tag{3}$$

Hence $H \succeq 0$, which implies $E(w)$ is a convex function. Note that the equality in (3) holds if and only if $\boldsymbol{u} = 0$ or $\boldsymbol{u} \in$ Null-Space(X), where $X = [x_1, x_2, \ldots, x_N]$. Thus, if $dim($Null-Space$(X)) = 0$, H is strictly positive definite matrix. Hence $E(\boldsymbol{w})$ is a strictly convex function and has a unique minimum.

2. (6 points)

   (a) (2 points) Using the update for $\boldsymbol{w}_t$,

$$\boldsymbol{w}^{*T}\boldsymbol{w}_t = \boldsymbol{w}^{*T}\boldsymbol{w}_{t-1} + y_{t-1}(\boldsymbol{w}^{*T}\boldsymbol{x}_{t-1}).$$

   Since $\boldsymbol{w}^*$ is a separating hyperplane with margin γ, $y_{t-1}(\boldsymbol{w}^{*T}\boldsymbol{x}_{t-1}) \geq \gamma, \forall t$. Thus,

$$\boldsymbol{w}^{*T}\boldsymbol{w}_t \geq \boldsymbol{w}^{*T}\boldsymbol{w}_{t-1} + \gamma.$$

   Hence, using induction it can be easily seen that $\boldsymbol{w}^{*T}\boldsymbol{w}_t \geq t\gamma$. Note that $\boldsymbol{w}_0 = \boldsymbol{0}$, hence base case is satisfied trivially. In fact, assuming $\boldsymbol{w}_0 = \boldsymbol{0}$ is not necessary in completing the proof. Without loss of generality, we can always find a $\boldsymbol{w}_0$ such that $\boldsymbol{w}^{*T}\boldsymbol{w}_0 \geq 0$: first randomly initialize $\boldsymbol{w}_0$; then change the sign of $\boldsymbol{w}_0$ if the number of misclassified instances are more than half of the total number of training instances by the initial separating hyperplane.

(b) (2 points) Using the update for $\boldsymbol{w}_t$:

$$\begin{aligned}\|\boldsymbol{w}_t\|^2 &= \|\boldsymbol{w}_{t-1} + y_{t-1}\boldsymbol{x}_{t-1}\|^2, \\ &= \|\boldsymbol{w}_{t-1}\|^2 + 2y_{t-1}\boldsymbol{w}_{t-1}^T\boldsymbol{x}_{t-1} + y_{t-1}^2\|\boldsymbol{x}_{t-1}\|^2, \\ &\leq \|\boldsymbol{w}_{t-1}\|^2 + \|\boldsymbol{x}_{t-1}\|^2, \\ &\leq \|\boldsymbol{w}_{t-1}\|^2 + R^2,\end{aligned}$$

since $y_{t-1}\boldsymbol{w}_{t-1}^T\boldsymbol{x}_{t-1} \leq 0$ whenever an update is made, $y_t^2 = 1$, and $\|\boldsymbol{x}_t\| \leq R\ \forall t$.
Hence, using induction it can be easily shown that $\|\boldsymbol{w}_t\|^2 \leq tR^2$. Again note that base case holds trivially as $\boldsymbol{w}_0 = \mathbf{0}$.

(c) (2 points) Using (a) and (b):

$$1 \geq \frac{\boldsymbol{w}^{*T}\boldsymbol{w}_t}{\|\boldsymbol{w}^*\|\|\boldsymbol{w}_t\|} \geq \frac{t\gamma}{\|\boldsymbol{w}^*\|\sqrt{t}R}$$

Thus, $t \leq \frac{R^2\|\boldsymbol{w}^*\|^2}{\gamma^2}$. This implies that the Perceptron algorithm converges to a separating hyperplane $\boldsymbol{w}^*$ in at most $\frac{R^2\|\boldsymbol{w}^*\|^2}{\gamma^2}$ steps.

3. (6 points)

(a) (2 points) The sample code for least squares regression is given below.

```
function [W] = linreg(X,y)
% Solve a linear regression for classifcation
%
% Input:
%    X: N by d matrix
%    y: N by 1 vector
%
% Output:
%    W: d+1 by K matrix (K: number of classes)
%

K = length(unique(y));
I = eye(K);
Y = I(y,:);
Xhat = [ones(size(y)) X];
W = Xhat'*Xhat\(Xhat'*Y);

end % linreg function
```

Figure 1 shows the plot of data points according to the predicted labels by least squares regression, from which we observe that the region of input space assigned to the center class is very small and most of the points from that class are misclassified.

(b) (4 points) The sample code for logistic regression is given below. Note that since the Hessian matrix H is positive semi-definite and hence rank deficient we can use the technique introduced in homework 1 to compute the inverse. In the sample code, the pinv Matlab function is used.

```
function [W] = logreg(X,y)
% Solve a logistic regression for classification
```

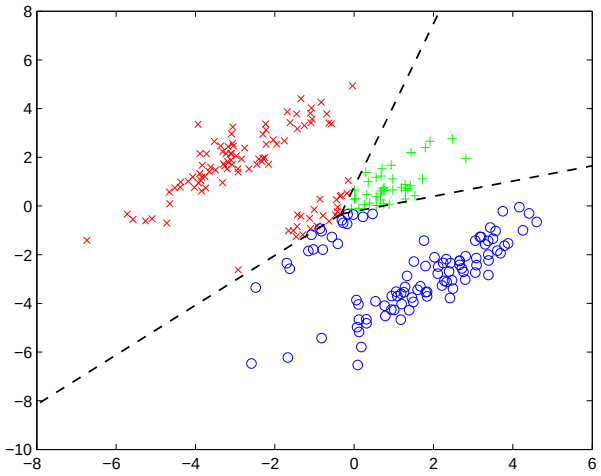

Figure 1: Prediction by least squares regression.

```
%
% Input:
%    X: N by d matrix
%    y: N by 1 vector
%
% Output:
%    W: d+1 by K matrix (K: number of classes)
%

K = length(unique(y));
I = eye(K);
Y = I(y,:);
Xhat = [ones(size(y)) X];
M = size(Xhat,2);

W = zeros(M,K);
maxi = 100; eta = 1e-8;
for t = 1:maxi
    Z = exp(Xhat*W);
    Z = diag(sparse(1./sum(Z,2)))*Z;
    obj = -sum(sum(Y.*log(Z)));
    if t > 1
        if abs(old_obj-obj) <= eta
            break;
        end
    end
    old_obj = obj;
    fprintf('iter %d: obj %f\n',t,obj);
    gW = Xhat'*(Z-Y);
```

```
    H = zeros(K*M);
    for i = 1:K
        for j = 1:K
            Zb = Z(:,i).*(I(i,j)-Z(:,j));
            H((i-1)*M+1:i*M,(j-1)*M+1:j*M) = Xhat'*diag(Zb)*Xhat;
        end
    end
    W(:) = W(:)-pinv(H)*gW(:);
end

end % logreg function
```

Figure 2 shows the plot of data points according to the predicted labels by logistic regression, from which we observe that logistic regression can get a much better classification results on the training data.

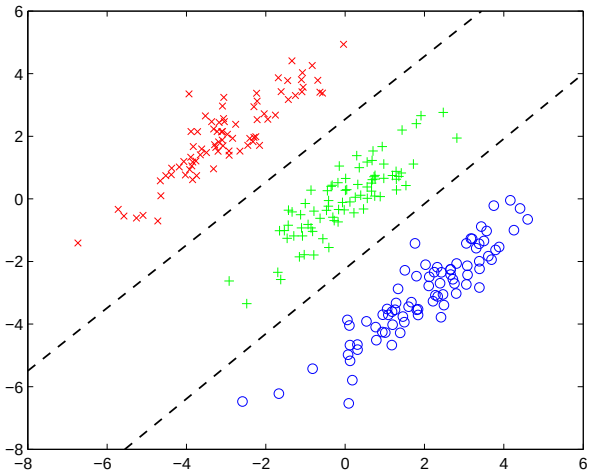

Figure 2: Prediction by logistic regression.

4. (10 points)

   (a) (2 points) As only α_1 and α_2 are updated and $y_1^2 = 1$,

$$\sum_i y_i\alpha_i = 0 \Rightarrow y_1\alpha_1 + y_2\alpha_2 = y_1\overline{\alpha}_1 + y_2\overline{\alpha}_2 \Rightarrow \overline{\alpha}_1 = \alpha_1 + y_2y_1\alpha_2 - y_2y_1\overline{\alpha}_2 \geq 0. \tag{4}$$

   If $y_1 = y_2$, $\alpha_1 + \alpha_2 \geq \overline{\alpha}_2$. If $y_1 \neq y_2$, $\alpha_1 - \alpha_2 + \overline{\alpha}_2 \geq 0 \Rightarrow \overline{\alpha}_2 \geq \alpha_2 - \alpha_1$.

   (b) (2 points) From (4) it is clear that $\alpha_1 + y_2y_1\alpha_2 = \gamma$. Now,

$$W(\alpha_1, \alpha_2) = \alpha_1 + \alpha_2 - \frac{1}{2}\left(\alpha_1^2K_{11} + 2s\alpha_1\alpha_2K_{12} + \alpha_1y_1v_1 + \alpha_2^2K_{22} + \alpha_2y_2v_2\right) + \text{constant}.$$

   Substituting $\alpha_1 = \gamma - s\alpha_2$ and simplifying, we get:

$$\begin{aligned} W(\alpha_2) \quad = \quad & \gamma - s\alpha_2 + \alpha_2 - \frac{1}{2}K_{11}(\gamma - s\alpha_2)^2 - \frac{1}{2}K_{22}\alpha_2^2 \\ & -sK_{12}(\gamma - s\alpha_2)\alpha_2 - y_1(\gamma - s\alpha_2)v_1 - y_2\alpha_2v_2 + \text{constant} \end{aligned}$$

(c) (3 points) Setting $\frac{dW}{d\alpha_2} = 0$, we get:

$$\overline{\alpha}_2 = \frac{1}{d_{12}} \left(s(K_{11} - K_{12})\gamma + y_2(v_1 - v_2) + 1 - s \right)$$

Check that $\frac{d^2W}{d\alpha_2^2} = -d_{12}$ for the above given α_2. Now K is positive definite, hence $d_{12} > 0$.

(d) (3 points) Note that $\gamma = \alpha_1 + s\alpha_2$. Substituting value of γ into the expression for $\overline{\alpha}_2$ given above and simplifying, we get:

$$\bar{\alpha}_2 = \alpha_2 + \frac{y_2(E_1 - E_2)}{d_{12}},$$

where $E_i = f(\mathbf{x}_i) - y_i = (\sum_{j=1}^{N} \alpha_j y_j K_{ij} + w_0) - y_i$.
Note that $\overline{\alpha}_2$ given by above update need not satisfy the constraints. So, we clip $\overline{\alpha}_2$ so as to obtain a feasible dual solution.
After the update $\overline{\alpha}_2$ should be greater than 0. Thus,

$$\overline{\alpha}_2 = \max(0, \overline{\alpha}_2).$$

Let $y_1 = y_2$. Using part (a), $\overline{\alpha}_2 \leq \alpha_1 + \alpha_2$. Thus $\overline{\alpha}_2 = \min(\alpha_1 + \alpha_2, \overline{\alpha}_2) = \max(0, \min(\alpha_1 + \alpha_2, \overline{\alpha}_2))$.
Now, let $y_1 \neq y_2$. Using part (a), $\overline{\alpha}_2 \geq \alpha_2 - \alpha_1$. Thus $\overline{\alpha}_2 = \max(\overline{\alpha}_2, \alpha_2 - \alpha_1) = \max(0, \overline{\alpha}_2, \alpha_2 - \alpha_1)$.
Using part (b), $\overline{\alpha}_1 + s\overline{\alpha}_2 = \alpha_1 + s\alpha_2$. This implies $\overline{\alpha}_1 = \alpha_1 + s(\alpha_2 - \overline{\alpha}_2)$. Note that this update gives feasible $\overline{\alpha}_1$.