CS260, Winter 2017
Problem Set 4: Linear Regression and Perceptron
Due 2/1/2017

Ray Zhang

Jan 21, 2017

1 Q1

(a) Problem 1a

Solution: Solution to problem 1a

The log-likelihood of the data is a function of β and σ_n for all $n = 1, 2, 3, ..., N$.

For a single data point:

- $y_n = x_n^T\beta + \epsilon_n$.

- $\epsilon_n \sim N(0, \sigma_n)$.

- $P(y_n|x_n) = N(x_n^T\beta, \sigma_n)$.

Log probability of a single data point:

$logP(y_n|x_n) = -\frac{1}{2}log2\pi - log\sigma_n - \frac{(y_n - x_n^T\beta)^2}{2\sigma_n^2}$

If we express this likelihood function as a function of x_n, β and σ_n, then we have, for all data:

$\mathcal{L}(\beta|\sigma, x) = -\sum_{n=1}^{N} \frac{1}{2}log2\pi + log\sigma_n + \frac{(y_n - x_n^T\beta)^2}{2\sigma_n^2}$

(b) Problem 1b

Solution: Solution to problem 1b

Maximizing $\mathcal{L}(\beta|\sigma, x)$ is the same as minimizing the negative log probability, $-\mathcal{L}(\beta|\sigma, x)$:

$argmin_\beta - \mathcal{L}(\beta|\sigma, x)$

To find minima or maxima, we take the gradient of β:

$\nabla_\beta(\sum_{n=1}^{N}\frac{1}{2}log2\pi + log\sigma_n + \frac{(y_n - x_n^T\beta)^2}{2\sigma_n^2})$

$= \sum_{n=1}^{N}\frac{2(y_n - x_n^T\beta)}{2\sigma_n^2}x_n$

$= \sum_{n=1}^{N}\frac{y_n - x_n^T\beta}{\sigma_n^2}x_n$

$= \sum_{n=1}^{N}a_n x_n$

$= X^T a = 0$

where $a = \Sigma^{-1}(y - X\beta)$, where $\Sigma_{ij} = 0$ if $i \neq j$, and σ_n^2 if $i = j = n$.

Substituting:

$X^T\Sigma^{-1}(y - X\beta) = 0$

$= X^T\Sigma^{-1}y - X^T\Sigma^{-1}X\beta$

$X^T\Sigma^{-1}X\beta = X^T\Sigma^{-1}y$

$\beta = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y$, which is what we wanted.

2 Q2

(a) Problem 2a

Solution: Solution to problem 2a

We want to regularize with the following:

$(\beta_i - \beta_{i+1})^2$, which is an L2 regularization on the size of adjacent values in β.

Combining this with the original L2 regularization in our cost function:

$\mathcal{L}(\beta, \lambda_1, \lambda_2) = ||y - X\beta||^2 + \lambda_1\beta^T\beta + \lambda_2(\beta - \beta_{shifted})^T(\beta - \beta_{shifted})$

To construct $\beta_{shifted}$, we will use a shift matrix: U, which stands for upper shift, where $U_{ij} = 1$ if $i + 1 = j$, and 0 otherwise.

To express it in differentiable form:

$\mathcal{L}(\beta, \lambda_1, \lambda_2) = (y - X\beta)^T(y - X\beta) + \lambda_1\beta^T\beta + \lambda_2(\beta - U\beta)^T(\beta - U\beta)$

(b) Problem 2b

Solution: Solution to problem 2b

To find the closed form, we solve for β:

$y^Ty - y^TX\beta - \beta^TX^Ty + \beta^TX^TX\beta + \lambda_1\beta^T\beta + \lambda 2(\beta^T\beta - \beta^TU\beta - \beta^TU^T\beta - \beta^TU^TU\beta)$

$= y^Ty - 2y^TX\beta + \beta^TX^TX\beta + \lambda_1\beta^T\beta + \lambda 2(\beta^T\beta - 2\beta^TU\beta + \beta^TU^TU\beta)$

$\nabla_\beta\mathcal{L} = -2y^TX + 2X^TX\beta + 2\lambda_1\beta + \lambda_2(2\beta - 2U\beta + 2U^TU\beta) = 0$

$y^TX = X^TX\beta + \lambda_1\beta + \lambda_2\beta - \lambda_2U\beta + \lambda_2U^TU\beta$

$y^TX = (X^TX + \lambda_1I + \lambda_2(I - U + U^TU))\beta$

$\beta = (X^TX + \lambda_1I + \lambda_2(I - U + U^TU))^{-1}y^TX.$

3 Q3

We will use a lagrangian to make sure that the optimization will be inside of the subspace spanned by $A\beta = b$:

$\mathcal{L}(\beta, \lambda) = ||X\beta - y||^2 + \lambda(A\beta - b)$

Expanding this gives us:

$\mathcal{L}(\beta, \lambda) = y^Ty - 2X^Ty\beta + \beta^TX^TX\beta + \lambda A\beta - \lambda b$

Differentiating:

$\nabla_\beta\mathcal{L} = -2X^Ty + 2(X^TX)\beta + \lambda A = 0$

$X^Ty = X^TX\beta + \frac{1}{2}\lambda A$

$\beta = (X^TX)^{-1}(X^Ty - \frac{\lambda}{2}A)$

Plugging β back into the constraint:

$A\beta = b$

$A(X^TX)^{-1}(X^Ty - \frac{\lambda}{2}A) = b$

$A(X^TX)^{-1}X^Ty - A(X^TX)^{-1}\frac{\lambda}{2}A = b$

Solving for λ gives us:

$\frac{\lambda}{2} = (A(X^TX)^{-1}A)^{-1}(A(X^TX)^{-1}X^Ty - b)$

$\lambda = 2(A(X^TX)^{-1}A)^{-1}(A(X^TX)^{-1}X^Ty - b)$

Plugging back to solve for β, we get:

$\beta = (X^TX)^{-1}(X^Ty - \frac{\lambda}{2}A)$

$\beta = (X^TX)^{-1}(X^Ty - (A(X^TX)^{-1}A)^{-1}(A(X^TX)^{-1}X^Ty - b)A)$

4 Q4

Originally, our update rule was too aggressive:

$sign(w^Tx_n) \neq y_n$, then update using $w^{t+1} = w^t + y_nx_n$.

Our point was to move the new prediction on the other side of the hyperplane, such that:

$y_nw^Tx_n < 0$, update to $y_n(w + y_nx_n)^Tx_n = y_nw^Tx_n + y_n^2x_n^Tx_n$.

However, because it's too aggressive, we often get an update value where $y_n(w^{t+1})^Tx_n > 0$. We will change our update rule like following:

$y_nw^Tx_n = \epsilon$, where $\epsilon < 0$.

We want to update such that: $w^{t+1} = w^t + v$ for some v.

$y_n(w + v)^Tx_n = 0 = \epsilon - \epsilon$

Thus, $y_nv^Tx_n = -\epsilon$. That will be our lagrangian constraint.

We want to minimize $||v||^2$ so that the update is minimized in terms of harshness, but we still move the hyperplane to the correct boundaries.

Thus, our lagrangian is:

$\mathcal{L}(w, \lambda) = ||v||^2 + \lambda(y_nv^Tx_n + \epsilon)$

$\nabla_vL = 2v + \lambda(y_nx_n) = 0$

Thus:

$v = \frac{\lambda(y_nx_n)}{2}$

Plugging into the constraint:

$y_nv^Tx_n + \epsilon = 0$

$y_n\frac{\lambda(y_nx_n)}{2} + \epsilon = 0$

$y_n\frac{\lambda(y_nx_n)}{2} = -\epsilon = -(y_nw^Tx_n)$

$\lambda = \frac{-2y_nw^Tx_n}{y_n^2x_n^Tx_n}$

Plugging back into v:

$$v = \frac{\lambda(y_n x_n)}{2} = \frac{\frac{-2y_n w^T x_n}{y_n^2 x_n^T x_n}(y_n x_n)}{2}$$

$$v = \frac{-w^T x_n}{x_n^T x_n} x_n$$

$$v = -proj_{x_n} w$$

Thus, the conclusive result is to update by some v, where v is the projection of w onto x_n. Plugging back in to the update rule:

$$w^{t+1} = w^t - proj_{x_n} w$$

$$y_n (w^{t+1})^T x_n = y_n w^T x_n - y_n (proj_{x_n} w)^T x_n$$

$$= y_n w^T x_n + y_n \frac{-w^T x_n}{x_n^T x_n} x_n^T x_n$$

$$= y_n w^T x_n - y_n w^T x_n = 0$$

This makes intuitive sense, since we added by a positive term to push the projection onto the positive side of the hyperplane originally, here we are also pushing it by a positive term.

So the final update rule is:

$w^{t+1} = w^t - proj_{x_n} w$ if $sign(w^T x_n) \neq y_n$