CS 189: Introduction to Machine Learning - Discussion 4

1. Intercepts in Linear Regression

   In the traditional linear regression scenario, where we model y with a line, or,

   $$\hat{y} = \vec{w}^T \vec{x}$$

   we aim to estimate $\vec{w}$. However, this model forces the lines to cross the origin (plug in $\vec{x} = \vec{0}$), severely limiting the power of the model. A typical solution to this problem is that the weight vector is extended by 1 and each input vector $\vec{x}$ has a 1 added at the beginning. This effectively adds an intercept term the model and allows for any line to be created.

   But that's boring! Let's come up with a solution to the intercept issue.

   Let's say we're stuck with the technique that fits only lines that go through the origin. We could shift the data to center it at zero, fit a line, and then shift our zero-centered line to where it's supposed to be. To center our data around the origin, we can subtract the mean of the x's and the mean of the y's from the data.

   a) Given that $\bar{x}$ and $\bar{y}$ are the means of our data, find the new model for a line predicting y from $\vec{x}$.

   Solution: We shift our data by the means, centering it at 0. Then train a linear standard linear regression model on it. What is the new intercept term?

   $$y_{centered} = \vec{w}^T \vec{x}_{centered}$$

   $$y - \bar{y} = \vec{w}^T (\vec{x} - \bar{x})$$

   $$y = \vec{w}^T \vec{x} + (\bar{y} - \vec{w}^T \bar{x})$$

   The intercept is $\bar{y} - \vec{w}^T \bar{x}$.

   Another approach to the intercept term would be to just model an intercept in our equation and estimate it from the data. The model would now look like this:

   $$\hat{y} = \vec{w}^T \vec{x} + w_0$$

   b) Find the MLE estimate of w_0. You should get the same answer as before. Assume that y is modeled with a line plus an intercept and Gaussian noise, i.e.,

   $$y \sim \mathcal{N}(\vec{w}^T \vec{x} + w_0, \sigma^2)$$

Solution: Given that y is modeled with a line plus an intercept and Gaussian noise, i.e.,

$$y \sim \mathcal{N}(\vec{w}^T\vec{x} + w_0, \sigma^2)$$

We have n training samples, $(x_1, x_2, x_3, \ldots, x_n)$. Performing MLE:

$$L(\vec{w}, w_0|y_1, \ldots, y_n, x_1, \ldots, x_n) = P(y_1|x_1, \vec{w}, w_0)P(y_2|x_2, \vec{w}, w_0)\cdots P(y_n|x_n, \vec{w}, w_0)$$

$$L(\vec{w}, w_0|x_1, \ldots, x_n) = \prod_{i=1}^{n} P(y_i|x_i, \vec{w}, w_0)$$

$$L(\vec{w}, w_0|x_1, \ldots, x_n) = (\frac{1}{\sqrt{(2\pi\sigma^2)}})^n \exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - (\vec{w}^T\vec{x_i} + w_0))^2)$$

$$l(\vec{w}, w_0|x_1, \ldots, x_n) = n\ln(\frac{1}{\sqrt{(2\pi\sigma^2)}}) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - (\vec{w}^T\vec{x_i} + w_0))^2$$

We want to maximize this the log likelihood, or equivalently minimize the following quantity:

$$\min(\frac{1}{2}\sum_{i=1}^{n}(y_i - (\vec{w}^T\vec{x_i} + w_0))^2)$$

$$\min(\frac{1}{2}\sum_{i=1}^{n}(y_i^2 - 2y_i(\vec{w}^T\vec{x_i} + w_0) + (\vec{w}^T\vec{x_i})^2 + 2w_0\vec{w}^T\vec{x_i} + w_0^2))$$

Taking the gradient with respect to w_0 gives us:

$$\frac{1}{2}\sum_{i=1}^{n}(-2y_i + 2w_0 + 2\vec{w}^T\vec{x_i}) = 0$$

$$\hat{w_0} = \frac{\sum_{i=1}^{n} y_i}{n} - \frac{\sum_{i=1}^{n} \hat{w}^T\vec{x_i}}{n}$$

2. Linearly Separable Data with Logistic Regression

Show (or explain) that for a linearly separable data set, the maximum likelihood solution for the logistic regression model is obtained by finding a vector β whose decision boundary $\beta^T x = 0$ separates the classes, and taking the magnitude of β to be infinity. **Note**: Remember that as mentioned in lecture, doing maximum-likelihood on logistic regression is same as minimizing cross-entropy loss (see lecture-6, slides-21,22). In lecture, we explored the cross-entropy loss-minimization perspective to logistic regression. This question will make you explore the likelihood perspective.

Solution:

Because the data is linearly separable, it is possible to find a hyperplane with unit normal vector β such that each halfspace induced by this hyperplane contain all samples of one class.

Consider all points on the half space defined by $\beta^T x \geq 0$. Without loss of generality, let's say that all these points come from class 1, while the points such that $\beta^T x < 0$ come from class 0. For some point x_{c_1} in class 1,

$$P(y = 1|x_{c_1}) = \mu_1 = \frac{1}{1 + exp(-\beta^T x_{c_1})} > 0.5$$

because $\beta^T x_{c_1} \geq 0$. Likewise, for a point x_{c_0} in class 0,

$$P(y = 0|x_{c_0}) = 1 - P(y = 1|x_{c_0}) = 1 - \mu_1 > 0.5$$

since $\beta^T x_{c_0} < 0$. Now, when we inspect the likelihood of the data, given by

$$L(\beta|D) = \prod_{i=1}^{n} \mu_i^{y_i}(1 - \mu_i)^{1-y_i} = \prod_{i \in C_1} \mu_i \prod_{j \in C_0} (1 - \mu_j)$$

we see that if we take some arbitrary $k > 1$ and scale the unit vector β by k, our likelihood will increase, since all of the individual probabilities in the likelihood will increase. In fact, we can set $k = \infty$, which will maximize our likelihood. This will render the sigmoid function to be infinitely steep at $\beta^T x_i = 0$ (making it a step function). $P(y = y_i|x_i) = 1$ for all x_i, and the likelihood will be 1. Obviously this is severely overfitting the data, and regularization for this problem would help us avoid that issue.

3. LMS algorithm

   Derive the LMS algorithm for a given dataset contaning m example pairs $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i \in \mathbb{R}^n$ and $y \in \mathbb{R}$. The model is assumed to be linear, i.e., $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The aim is to minimize the squared loss over all examples and obtain the Stochastic Gradient Descent update equation.

Solution: The L2 error function is described as

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2 \\ &= \sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - y_i)^2 \end{aligned} \tag{1}$$

We want to minimize this loss. Note that it is convex function in $\mathbf{w}$, so stationary point is indeed the minima. Derivative is

$$\frac{\partial\mathcal{L}(\mathbf{w})}{\partial\mathbf{w}} = 2\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - y_i)\mathbf{x}_i$$

Now we write the update equations. Let η be the step-size.
Steepest Gradient Descent: Each gradient step is over full dataset.

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - 2\eta\sum_{i=1}^{n}(\mathbf{w}^{(t)T}\mathbf{x}_i - y_i)\mathbf{x}_i \tag{2}$$

Stochastic Gradient Descent: Each gradient step uses a single example. This is also called LMS (Least Mean Square) algorithm.

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - 2\eta(\mathbf{w}^{(t)T}\mathbf{x}_i - y_i)\mathbf{x}_i \tag{3}$$

Note 1: In above update equation, you can as well have the ,mapping $\Phi(\mathbf{x})$ in place of $\mathbf{x}$. That will be the solution for $f(\mathbf{x}) = \mathbf{w}^T\Phi(\mathbf{x})$. This is what was discussed in (lecture 8, slide 25).

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - 2\eta(\mathbf{w}^{(t)T}\Phi(\mathbf{x}_i) - y_i)\Phi(\mathbf{x}_i) \tag{4}$$

Note 2: If we add L2 regularizer (i.e. $\gamma \mathbf{w}^T \mathbf{w}$) to the loss function in Equation (1), then the update equation (4) for $\mathbf{w}$ will look as follows (lecture 8, slide 22):

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - 2\eta(\mathbf{w}^{(t)T}\Phi(\mathbf{x}_i) - y_i)\Phi(\mathbf{x}_i) - \eta\gamma\mathbf{w}^{(t)} \tag{5}$$

Note 3: If we consider the dual kernel for of loss function in Equation (1), then the update equation in Equation (5) can be written as follows (lecture 4, slide 25) and (lecture 8, slide 24):

$$\mathbf{w} = \sum_i \alpha_i \Phi(\mathbf{x}_i)$$

(From Equation (5))

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - 2\eta \underbrace{(\mathbf{w}^{(t)T}\Phi(\mathbf{x}_i) - y_i)}_{\text{update in } \alpha_i} \Phi(\mathbf{x}_i) - \eta\gamma \sum_i \underbrace{\alpha_i}\, \Phi(\mathbf{x}_i)$$

Update equation for i^{th} example pair $(\mathbf{x}_i, y_i)$ (time t),

$$\mathbf{w}^{(t)} = \sum_i \alpha_i^{(t)} \Phi(\mathbf{x}_i)$$

$$\alpha_i^{(t+1)} = \alpha_i^{(t)} - 2\eta(\mathbf{w}^{(t)T}\Phi(\mathbf{x}_i) - y_i) - \eta\gamma\alpha_i^{(t)} \quad \text{(For example } i\text{)}$$

$$\alpha_h^{(t+1)} = \alpha_h^{(t)} - \eta\gamma\alpha_h^{(t)} \quad \text{(For other examples } h \in \{1, 2 \ldots, m\}/i\text{)}$$

4. Linear Algebra

   a) Let A be a square matrix. Show that we can write A as the sum of a symmetric matrix B and an antisymmetric matrix C:

   $$A = B + C$$

   (where $B = B^\top$ and $C = -C^\top$).

   b) Show that if C is antisymmetric, then $x^\top C x = 0$ for all nonzero x.

   c) Show that the inverse of a symmetric matrix is symmetric.

d) Explain why covariance matrices of multivariate Gaussians can be assumed to be symmetric without loss of generality.

Solution:

a) Choose $B = \frac{A+A^\top}{2}$ and $C = \frac{A-A^\top}{2}$. Notice that if A is symmetric, then $A = B$ and $C = 0$.

b) The expression $x^\top Cx$ is a scalar, so $x^\top Cx = (x^\top Cx)^\top = x^\top C^\top x = -x^\top Cx$ (where the last step uses the fact that C is antisymmetric). Thus $x^\top Cx = 0$.

c) Let B be a symmetric matrix. Then $(B^{-1})^\top = (B^\top)^{-1} = B^{-1}$, so B^{-1} is symmetric.

d) For a random vector $\mathbf{x} = [x_1 x_2 \dots x_n]^T$, covariance between each pair is defined as

$$cov(x_i, x_j) = \mathbb{E}[(x_i - \mu_i)(x_j - \mu_j)]$$

and thus covariance matrix Σ is defined as

$$\Sigma_{ij} = cov(x_i, x_j) = \mathbb{E}[(x_i - \mu_i)(x_j - \mu_j)] = \mathbb{E}[(x_j - \mu_j)(x_i - \mu_i)] = cov(x_j, x_i) = \Sigma_{ji}$$

Thus Σ is symmetric matrix.