Discussion 5

Michael Zhang

Updated February 23, 2017

1 Administrative

- Homework 3 due next Monday.

2 Problem 2

- Remember that 0-1 loss functions for binary classification gives us a nice way to find the decision boundary: set the posteriors equal. (why?)
- After that, we work through some algebra. This was a problem on the Spring 2016 midterm!

3 Linear Regression

- We've focused on classification up until now (with QDA and LDA being the latest techniques). We are now going to move into regression. Rather than predicting a class, we predict a number, which is usually continuous. Some examples of regression problems:
  - given features of a house (bedrooms, location, size, contains swimming pool etc.), predict the value of the house.
  - A company interested in effect of advertising (online, newspaper, radio) on sales
- It can be helpful to see stuff written out as matrices, so I'm going to be explicit. In regression, we have input feature vectors $x_1, x_2, \ldots, x_n$ and dependent scalars $y_1, y_2, \ldots, y_n$. Our goal is to have $y \approx f(x, \theta)$, where θ is a parameter vector (there are nonparametric approaches as well, look up if you are curious). We can express all our data X in a design matrix where each row is a sample:

$$X = \begin{bmatrix} \text{———} x_1^T \text{———} \\ \text{———} x_2^T \text{———} \\ \text{———} x_3^T \text{———} \\ \cdots \\ \text{———} x_n^T \text{———} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_n \end{bmatrix}$$

- There are two steps: we first choose a model and then we fit the data with our model. Note that we have flexibility in choosing our model and a model that is easier to train might not fit the data well.
- In linear regression, we assume that we can fit a model $f(x, \theta) = x^T w \approx y$. We put the bias in the weight vector. Linear models work well on some datasets and are easy to train.
- To measure how well our model is doing, we define a loss that measures how far our predictions are from the ground truth. (What are some possible loss functions? Why is |prediction - actual| usually a better loss function than prediction - actual i.e. why does the absolute value matter? Which loss function punishes more heavily predictions that are far off from the ground truth?)
- In least-squares, we used the squared loss and have the objective $\min_w \sum_{i=1}^n (x_i^T w - y_i)^2$.

- We can also write this as

$$\left\| \begin{bmatrix} \text{———} x_1^T \text{———} \\ \text{———} x_2^T \text{———} \\ \text{———} x_3^T \text{———} \\ \cdots \\ \text{———} x_n^T \text{———} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_d \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_n \end{bmatrix} \right\|_2^2$$

- To see why these are equivalent, think of Xw as a vector of predictions and $Xw - y$ as the vector of differences between our predictions and the ground truth. Taking the square of the 2-norm sums up the square of each entry, which exactly matches the first problem. Writing it in matrix vector form (vectorizing) allows us to more easily use calculus. Our optimization problem is now:

$$\min_w \|Xw - y\|^2$$

- Sometimes we may care more about getting some predictions right than others. This is one motivation for *weighted least-squares*. Our objective is now:

$$\sum_{i=1}^{n} c_i (x_i^T w - y_i)^2$$

  where each c_i represents how much we penalize a mistake on the i-th training example. (What are the c_i in vanilla least squares?)

- Vectorizing this is a bit trickier, but one way is to look at the expansion of the quadratic form $u^T A u = \sum_i \sum_j A_{ij} u_i u_j$. For a diagonal matrix, this reduces to $u^T D u = \sum_i D_{ii} u_i^2$.

- If we go back to vanilla least squares, we can view

$$\|Xw - y\|^2 = (Xw - y)^T \mathbf{I} (Xw - y) = \sum_i \mathbf{1} \times u_i^2$$

  where $u_i = x_i^T w - y_i$. What do we want to use instead of $\mathbf{I}$?

- We define a $n \times n$ diagonal matrix $\mathbf{C}$ with the weights $c_1, c_2, \ldots, c_n$ on its diagonal. Our objective is now:

$$\min_w (Xw - y)^T \mathbf{C} (Xw - y)$$

- We can now take the gradient and set it equal to 0. Some general tips for taking gradients (because you need to be comfortable with them for this class):
  - Those proofs you did in Homework 2 are very useful.
    1. $\nabla_x x^T A x = (A + A^T)x$. When A is symmetric, this is $2Ax$.
    2. $\nabla_x a^T x = a$. $\nabla_x x^T a = a$. (Why does the first imply the second?)
  - Most of the time the trick is to recognize these terms after you've expanded your expression. In our current problem $A = X^T C X$ and we have a $w^T A w$ term.
  - If you get confused, check to make sure shapes match up.

- Final note: To make least squares (and other models) more robust, we often end up adding a regularization term $\lambda |w|^2$ or $\lambda |w|$. This decreases the magnitude of the weight vector in our solution. We'll see more of the motivation for why we do this in future lectures/discussions.

4 Remaining Time

- Talk more about Multivariate Gaussians and other concepts that are unclear on homework.
- Potentially do MLE problem from last discussion (if not done last week).