Lecture 10: Classification, Logistic Regression, and Networks
Modeling Social Data, Spring 2019
Columbia University

George Austin

April 5, 2019

1 Outline

- Classification
- Logistic Regression, with examples in R
  Vowpal Rabbit Example
- Networks
  History
  Structures and Applications

2 Classification

Question: why not solve Classification as a regression problem?

More precisely, say you have a set of response y variables that take values of either 0 or 1, and a set of x values that take any values on the real line.

$$x \in \mathbf{R}$$
$$y \in \{0, 1\}$$

Why not model a linear regression solution using OLS? That is, fit a model like:

$$\hat{y} = w \cdot x$$
$$\text{where } w = \frac{\mathbf{x} \cdot \mathbf{y}}{\mathbf{x} \cdot \mathbf{x}}$$
$$\text{, minmizing the loss function:}$$
$$L = \frac{1}{n} \sum_i \left(y_i - \hat{y_i}\right)^2$$

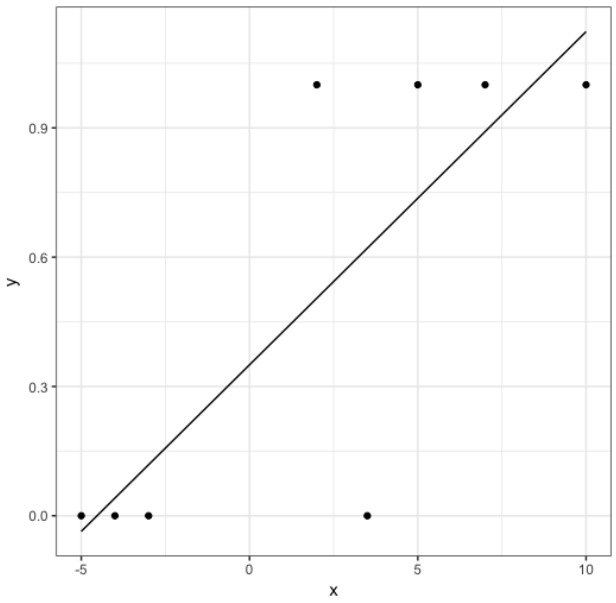
Look at the example plot (on the next page)

We can see that the point at the top right is giving a non zero loss, even though the line is above 1, as so it would predict the right label, but would still give a loss.

One possible fix to this, that is used in practice, is to set up a piece-wise linear model that is 1 if the prediction is greater than 1, and zero if the prediction in less that zero.

$$\hat{y} = 1 \text{ if } w \cdot x > 1$$
$$0 \text{ if } w \cdot x < 0$$
$$w \cdot x \text{ otherwise}$$

Sort of like a piece-wise-linear, jagged version of logistic Regression



3 Logistic Regression

Review: the model make a linear fit to the log odds:

$$\ln(\frac{p}{1-p}) = w \cdot x$$

This minimizes the loss function:

$$L = \prod_i p_i^{y_i} \cdot (1-p_i)^{1-y_i}$$

We then went through an example analyzing the chances of survival for passengers on the Titanic. (See class github page for the R file). The main takeaways are:

- Coefficients returned from logistic regression models can be difficult to interpret, as they relate to log-odds, although it is possible to make the tranformations to regular probabilities
- Instead, it can be much easier to analyze predictions against observations
- As more variables are added into the model, the results are harder to interpret

Next, we looked at machine-learning based models:

- goals are less about model interpretability
- prioritizes ability fro large-scale data processiong
- solely for prediction, quickly

One such machine-learning based approach we looked at was Vowpal Rabbit, which has the following advantages:

- Input formal is very easy, fast (Uses Stochastic Gradient Descent)
- Speed is fast
- It is scalale, can deal with a large number of features
- Feature pairing, progressive validation
- can do logistic regression, but can also do much more!

4 Networks

In our previous models, there has been an assumption of independence between datapoints. With networks, this assumption does not hold, and instead the datapoints are relational. Networks have both mathematical and social applications. First, a brief look at the history of Networks:

- first looked at Granoveter paper, modeling two individuals' number of mutual friends as a function of the strength of their frienship. (stronger friendship = more mutual friends)
- analyzing the success of academic papers. We could see the "long tail" that we've discussed in previous classes... most papers don't get cited much, if at all.
- Next, we looked at Watts and Strogat's Small-World Networks. A regular network sees no random connections between nodes, and a random network has only random connections between nodes. A small-world network is somewhere in between, with some randonm connections and some nonrandom connections. This model has the ability to capture long-ranging ties, i.e. the "6 degrees of separation" phenomenon.
- we saw another plot illustration the connections that blogs have. The plot observed a disconnect between different sides of the political spectrum.

Next, we Looked at types of Networks. There can be many applications for networks, such as modeling social, info, activity, biological or geographical networks. There is no single way to represent networks, but there are usually connections illustrated between nodes. Features of networks can also be directed, weighted...

There are many ways to store network data. Here are a few examples (see the lecture notes for graph visualizations):

1. Edge List
    - Is a list made up of every edge, with each entry containing the two nodes the edges connects
    - simple storage
    - bad for computation, computational complexity is O(number of edges)
2. Adjacency Matrix
    - a square matrix, where each row and column index is a node
    - grid of 1's and 0's. 1 represents a connection between nodes, 0 means no connection
    - this is quick to check edges
    - good for linear algebra
    - matrix is often sparse
3. Adjacency List
    - for each node, there is a list of what other node is connected to
    - this is good for graph traversal
    - if directed graph, needs 2 lists per node

There are many features that can be used to desctibe networks:

- degree - number of connections a node has
- path lenght - shortest past between nodes
- clustering
- components - how many disconnected parts does the network have?