Lecture 8: Naive Bayes / Classification
Modeling Social Data, Spring 2017
Columbia University

Anshul Doshi

March 10, 2017

1 Part I - Professor Wiggins

1.1 Classification

Goal: Get as familiar with classification as we are with regression

Learning by example:

- Spam Detection
  - unwanted email
  - How can you detect spam?

Building a theory of 3s

- Trying to predict a categorical value
- no sense of distance
- Another Example
  - Image recognition/OCR
  - 1996 - Optical character recognition
  - Just nee to successfully predict (don't need exact estimates)
  - Like deep learning + atari

1.1.1 What is Classification? - Bananas vs. Oranges

What would Gauss do?

Under Gauss, each banana and orange comes from the normal distribution and we get factors like length and height. We then use prior probabilities on choosing a banana.

What if you don't know all the features or have many covariates?

For example, we can have things like time of purchase or smell, etc... Thus, we want to be able to build a model for high dimensional data.

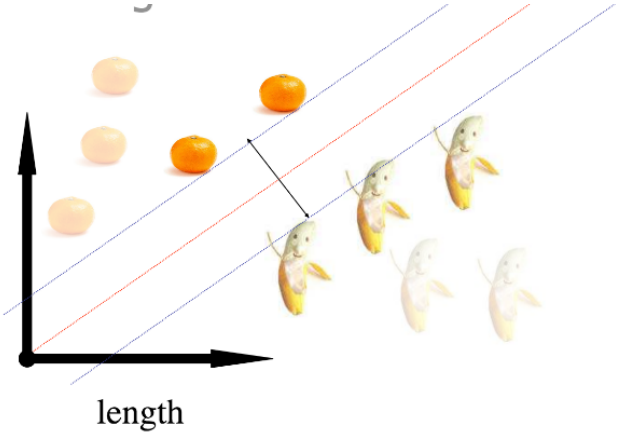

Figure 1: Bananas v Oranges: Margin/Large Deviation Theory

1.1.2 Game Theory

Assume the worst possible distribution. Note that with more features, the harder this is to do.

Large Deviation Theory
This idea of maximum margin, where margin is the distance separating bananas from oranges.

SVMS are a class of models geared toward finding this margin.

1.1.3 Example Application - New York Times

Goal: Want to investigate if there is a link between traffic fatalities and a particular air bag manufacturer - takata

To do this, we want to find "interesting cases" so we want to construct some labels/features that classifies a data entry of a particular incident as interesting or not.

Example phrases that were interesting were "suddenly deployed" or "unexpectedly". This allowed journalist to find cases and this type of computer assisted reporting led to a massive recall of takata airbags.

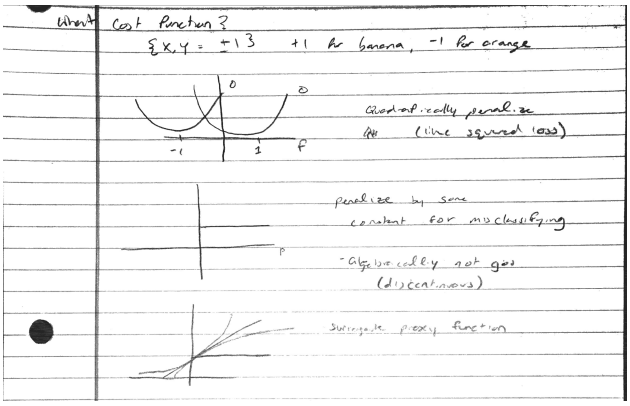

Figure 2: Examples of various cost functions

1.1.4 What is a cost function?

$$x, y = \pm 1 \rightarrow +1 = banana, -1 = orange$$

Misclassification often occurs from having no idea on how to step. Support proxy function gives tweaks on how to step.

1.1.5 MLE for Classification

Binary/Dichotomous/Boolean features + Naive Bayes
Goal: generalize and maintain linearity. Consider the spam problem of recognizing whether a peice of mail is spam.

Example - Bayes Rules Review Assume there you are testing for a disease that has infected 1 percent of the population

99 percent of sick patients test positive and 99 percent of healthy students test negative. Given that a patient tests positive, what is the probability that the patient is sick?

Bayes Theroem

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$$p(sick|+) = \frac{p(+|sick)p(sick)}{p(+)}$$

$$p(+) = p(+|sick)p(sick) + p(+|healthy)p(healthy)$$

$$p(sick|+) = \frac{p(+|sick)p(sick)}{p(+)} = \frac{99}{198} = 50\%$$

How does this relate to spam?
Build a one-word spam classifier:

$$p(spam|word) = \frac{p(word|spam)p(spam)}{p(word)}$$

where we just estimate the probabilities as ratios of counts.

Naive Bayes *Naive* comes from the fact that we are assuming every word appears independent of other words.

Bernoulli likelihood

$$p(x|c) = \prod_j \theta_{jc}^{x_j}(1 - \theta_{jc}^{1-x_{jc}})$$

Take log-likelihood and derivative to get MLE:

$$log(p(c|x)) = \sum_j x_j \log \frac{\theta_{jc}}{1 - \theta_{jc}} + \sum_j \log 1 + \theta_{jc}$$

$$\frac{d}{d\theta} log(p(c|x)) = \frac{\sum_j x_j}{\theta} + \frac{\sum_j 1 - x_j}{1 - \theta}$$

$$\theta_c = \frac{\sum_j x_j}{N}$$

which is just like regression

1.2 Boosting

Assume we have a set of training example. We iteratively add weak rules to our model (example: length ¡ 2, etc...). Each feature we add is meant to minimize a loss function which is determined by the weight of our current x_i. Weight increases if our new rules gets it right and decreases if the new rule gets it wrong.

Questions on boosting:
What happens if we get a feature with a high weight?
Boosting is extremely resistant to overfitting as we minimize test error.

Can a rule become useless? We can use decision trees or boosted trees. There are different types of boosting methods, each useful for different situations.

1.3 Summary

This lecture gives an introduction to classification showing various methods like naive bayes and boosting. We derived how naive bayes uses MLE and saw how boosting uses this iterative approach of minimizing a loss function and adding weights. We also got a brief introduction to SVMs.

2 Part II - Code

2.1 Revisiting Part I

How do we deal with discrete and continuous data?
With regression, we were able to have and equations like

$$p(y|x) = wx$$

This wont work here since $p(y|x)$ can't take any value between 0 and 1.

$$p(y = 1|x) = e^{w_1 x}, p(y = 0|x) = e^{w_2 x}$$

These need to sum to 1 so:

$$p(y = 1|x) = \frac{e^{w_1 x}}{e^{w_1 x} + e^{w_2 x}}$$

We can divide through by $e^{w_2 x}$ such that

$$p(y = 1|x) = \frac{e^{w_3 x}}{e^{w_3 x} + 1}$$

We want to get best w after given some data.
Naive Bayes: All weight are independent so you can estimate them separately
Boosting Doesn't assume this so we need to use whole vector

Naive Bayes
Weights are just MLEs so:

$$\log \frac{p}{1-p} = wx \rightarrow p(y|x) = \frac{e^{w_3 x}}{e^{w_3 x} + 1}$$

$$w_1 x_1 + w_2 x_2 + ...$$

Naive Bayes says we can estimate this by just counting. For example, if we are looking at the keyword "money" and whether its spam:

$$\hat{\theta} = \frac{N_{money,spam}}{N_{money}}$$

$$w = \log \frac{\theta_{money}}{1 - \theta_{money}}$$

2.2 Logistic Regression and Boosting

- Have a cost function for wieght
- Do log-likelihood, derivative $\rightarrow$ No closed form sol.
- Use principle of iteratively adding weights minimizing cost function (boosting)

2.3 Code

2.3.1 Shell Script

Grab enron dataset which has a bunch of spam and ham email. Use the spam formula from before for Naive Bayes:

$$p(spam|word) = \frac{p(word|spam)p(spam)}{p(word)}$$

Running on "enron" will give 0 which is not good as you are betting enron will never be spam. Thus we can add pseudocounts:

$$p_{money} = \frac{p(money|spam) + \alpha}{p_{spam} + \beta}$$

We can use cross validation to get best a and b.

2.3.2 R Code

ELSR dataset for spam. Here we can run naive bayes on training data and create a apriori table. We can then predict on test data and get predicted probabilities for every email. Then we can plot distribution or even bin things and plot a calibration plot.

Note: Naive Bayes is over confident since independence of features and multiplying things we are confident about will lead to overconfidence.

Logistic regression can correct for overconfidence by considering correlation between things.