CSE 4/587
Data Intensive Computing

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Day 18
Logistic (Logit) Regression

Announcements and Feedback

- Read Chapter 5 from Doing Data Science
- Grading for midterm is complete, grades just need to be uploaded to UBLearns
- Grading for Phase 1 is underway

Recap from Last Class

- Naive Bayes is a classifier based on Bayes Law
    - **Bayes Law:** $P(H \mid E) = P(E \mid H) * P(H) / P(E)$
    - Aggregate the results of ***many simple rules*** (Bayes Law) into a *probabilistic result*
    - Can be used with many many features (high dimensionality)
    - Assumes independent features, and reduces features to 0 or 1 *(this is why it is called naive)*
    - Laplace Smoothing introduces α and β as parameters of the model

More on Classifiers

Example Questions and Answers

● "Will someone click on this ad?"	0 or 1 (no or yes)
● "What number is this (image recognition)?"	0, 1, 2, 3, etc
● "What is this news article about?"	"Sports"
● "Is this spam?"	0 or 1
● "Is this pill good for headaches?"	0 or 1

Answering these questions can be done with classifiers

Choosing a Classifier?

1. What classifier should you use?
2. Which optimization method should you use for that classifier?
3. Which loss function should you minimize?
4. Which features of your data should you use?
5. Which evaluation metric should you use?

Choosing a Classifier?

1. **What classifier should you use?**
2. Which optimization method should you use for that classifier?
3. Which loss function should you minimize?
4. Which features of your data should you use?
5. Which evaluation metric should you use?

Choosing a Classifier?

Why not just try them all and see what performs best?

The real-world has constraints…

- Time constraints (these models take time to run)
- Your understanding (we can't be experts on every model)
- Interpretability (how do you understand/explain the models decisions)
- Scalability (how long to train? how long to score? how much memory?)

Choosing a Classifier?

- **First and foremost, you must understand your problem domain**
  - ie if you are working with election data, understand how election results are computed, etc
  - Which algorithm works best will be problem dependent **AND** question dependent (what is the question you are asking about your domain)
- **Rule of thumb:** simpler models are often easier to interpret but not as powerful (ie decision tree vs random forest)

Feature Engineering

Concern: bad feature selection will lead to bad classification, regardless of classifier choice

Observation: Much of data science is about understanding the data sets and the domain well enough that you can extract meaningful features

From Wikipedia:

"...a **feature** is an individual measurable property or characteristic of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, **classification** and regression."

Binary vs Multi-Class

- Binary classification is about 0,1 (no,yes) outputs
    - ie: is an email spam, is this patient sick, will this person click this ad, etc
- Multi-class problems have more than two labels. The standard solution is to train one binary classifier for each label, which classifies an input as having that label or not (binary).

Multi-Class Example

Article classification example to the right →

Classes are WORLD, BIZ, USA, SPAM, SPORT

```
*** Classifying instance: biz-01.html
P(WORLD|biz-01.html) = 0.085106382978723
P(BIZ|biz-01.html) = 0.765957446808511
P(USA|biz-01.html) = 0.063829787234043
P(SPAM|biz-01.html) = 0.042553191489362
P(SPORT|biz-01.html) = 0.042553191489362
Classified biz-01.html as BIZ

*** Classifying instance: sport-01.html
P(WORLD|sport-01.html) = 0.121212121212121
P(BIZ|sport-01.html) = 0.181818181818182
P(USA|sport-01.html) = 0.090909090909091
P(SPAM|sport-01.html) = 0.060606060606061
P(SPORT|sport-01.html) = 0.545454545454546
Classified sport-01.html as SPORT

*** Classifying instance: usa-01.html
P(WORLD|usa-01.html) = 0.235294117647059
P(BIZ|usa-01.html) = 0.352941176470588
P(USA|usa-01.html) = 0.176470588235294
P(SPAM|usa-01.html) = 0.117647058823529
P(SPORT|usa-01.html) = 0.117647058823529
Classified usa-01.html as BIZ

*** Classifying instance: world-01.html
P(WORLD|world-01.html) = 0.805970149253731
P(BIZ|world-01.html) = 0.089552238805970
P(USA|world-01.html) = 0.044776119402985
P(SPAM|world-01.html) = 0.029850746268657
P(SPORT|world-01.html) = 0.029850746268657
Classified world-01.html as WORLD
```

Multi-Class Example

Article classification example to the right →

Classes are WORLD, BIZ, USA, SPAM, SPORT

```
*** Classifying instance: biz-01.html
P(WORLD|biz-01.html) = 0.085106382978723
P(BIZ|biz-01.html) = 0.765957446808511
P(USA|biz-01.html) = 0.063829787234043
P(SPAM|biz-01.html) = 0.042553191489362
P(SPORT|biz-01.html) = 0.042553191489362
Classified biz-01.html as BIZ

*** Classifying instance: sport-01.html
P(WORLD|sport-01.html) = 0.121212121212121
P(BIZ|sport-01.html) = 0.181818181818182
P(USA|sport-01.html) = 0.090909090909091
P(SPAM|sport-01.html) = 0.060606060606061
P(SPORT|sport-01.html) = 0.545454545454546
Classified sport-01.html as SPORT

*** Classifying instance: usa-01.html
P(WORLD|usa-01.html) = 0.235294117647059
P(BIZ|usa-01.html) = 0.352941176470588
P(USA|usa-01.html) = 0.176470588235294
P(SPAM|usa-01.html) = 0.117647058823529
P(SPORT|usa-01.html) = 0.117647058823529
Classified usa-01.html as BIZ

*** Classifying instance: world-01.html
P(WORLD|world-01.html) = 0.805970149253731
P(BIZ|world-01.html) = 0.089552238805970
P(USA|world-01.html) = 0.044776119402985
P(SPAM|world-01.html) = 0.029850746268657
P(SPORT|world-01.html) = 0.029850746268657
Classified world-01.html as WORLD
```

Run a binary classifier for each class, ie Naive Bayes.

Multi-Class Example

Article classification example to the right →

Classes are WORLD, BIZ, USA, SPAM, SPORT

```
*** Classifying instance: biz-01.html
P(WORLD|biz-01.html) = 0.085106382978723
P(BIZ|biz-01.html) = 0.765957446808511
P(USA|biz-01.html) = 0.063829787234043
P(SPAM|biz-01.html) = 0.042553191489362
P(SPORT|biz-01.html) = 0.042553191489362
Classified biz-01.html as BIZ

*** Classifying instance: sport-01.html
P(WORLD|sport-01.html) = 0.121212121212121
P(BIZ|sport-01.html) = 0.181818181818182
P(USA|sport-01.html) = 0.090909090909091
P(SPAM|sport-01.html) = 0.060606060606061
P(SPORT|sport-01.html) = 0.545454545454546
Classified sport-01.html as SPORT

*** Classifying instance: usa-01.html
P(WORLD|usa-01.html) = 0.235294117647059
P(BIZ|usa-01.html) = 0.352941176470588
P(USA|usa-01.html) = 0.176470588235294
P(SPAM|usa-01.html) = 0.117647058823529
P(SPORT|usa-01.html) = 0.117647058823529
Classified usa-01.html as BIZ

*** Classifying instance: world-01.html
P(WORLD|world-01.html) = 0.805970149253731
P(BIZ|world-01.html) = 0.089552238805970
P(USA|world-01.html) = 0.044776119402985
P(SPAM|world-01.html) = 0.029850746268657
P(SPORT|world-01.html) = 0.029850746268657
Classified world-01.html as WORLD
```

Run a binary classifier for each class, ie Naive Bayes.

BIZ was the class that has the highest probability

Logistic Regression

What is it?

- Statistical model
- An approach for calculating the odds of an event happening vs other possibilities
- Discriminative classification vs Naive Bayes generative classification

Why are we studying it?

- To use it for classification!
- [1] Evidence suggests it performs better than Naive Bayes for large datasets
- Can model non-independent features

[1] http://robotics.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf

Motivating Example - Ad Clicks

- Given a list of websites a particular user visits, can you determine whether or not they will click a particular ad (ie a shoe ad)
  - We do not care about the content of the website, can just convert URL to some hash value or index
  - For each user, can create a vector, where each entry corresponds to a website, and has value 1 if the user visited the website, or 0 otherwise
  - Our training data is now a matrix where each row corresponds to a user, and there is an extra column with a 1 if they clicked the ad, or 0 otherwise

Motivating Example - Ad Clicks

click	url1	url2	url3	url4	url5
1	0	0	0	1	0
1	0	1	1	0	1
0	1	0	0	1	0
1	0	0	0	0	0
1	1	0	1	0	1

Motivating Example - Ad Clicks

click	url1	url2	url3	url4	url5
1	0	0	0	1	0
1	0	1	1	0	1
0	1	0	0	1	0
1	0	0	0	0	0
1	1	0	1	0	1

Goal: Predict the probability of a click, based on URLs visited

Motivating Example - Ad Clicks

This looks just like our formulation for Naive Bayes…

Naive Bayes could work for this problem as well

click	url1	url2	url3	url4	url5
1	0	0	0	1	0
1	0	1	1	0	1
0	1	0	0	1	0
1	0	0	0	0	0
1	1	0	1	0	1

Goal: Predict the probability of a click, based on URLs visited

Logistic Regression - The Math

Odds Ratio: $\frac{p}{1-p}$

The ***logit*** function is the basic building block of Logistic Regression

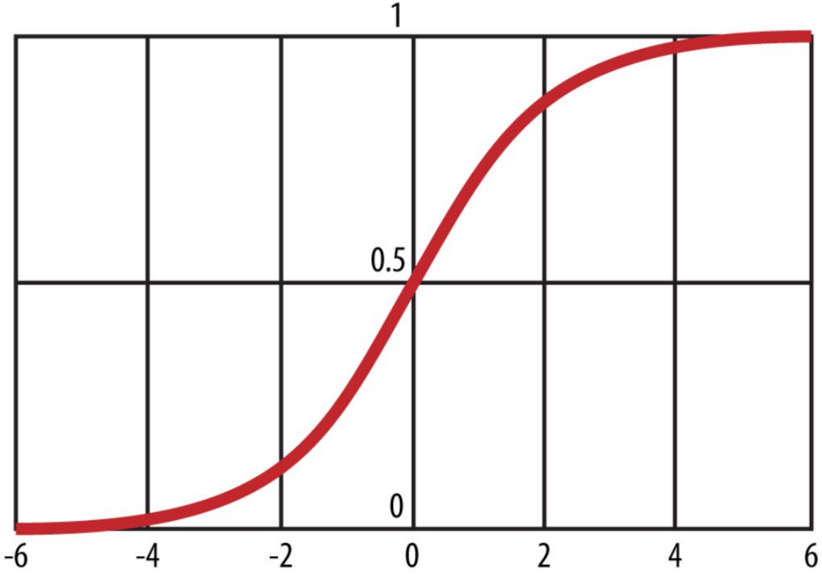
$$logit(p) = log(\frac{p}{1-p}) = log(p) - log(1-p)$$

It takes an ***x*** value in the range [0,1] (ie a probability) and transforms it to ***y*** values ranging across all real numbers

Inverse of logit function

The inverse of the logit function therefore takes a real number, and maps it to a result in the range [0,1]

$$logit^{-1}(t) = \frac{e^t}{1+e^t}$$

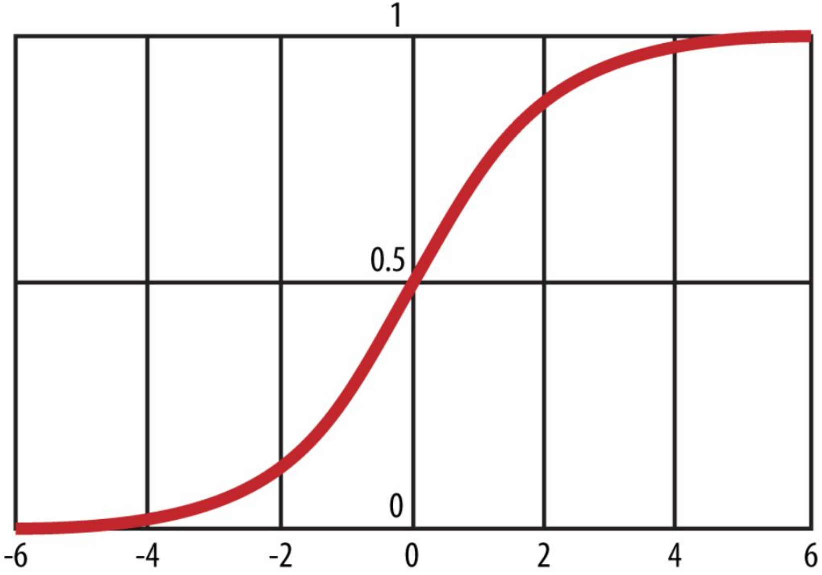


Inverse of logit function

The inverse of the logit function therefore takes a real number, and maps it to a result in the range [0,1]

$$logit^{-1}(t) = \frac{e^t}{1+e^t}$$

We can use this to get a probability



Classification with Logit

$$P(c_i|x_i) = [logit^{-1}(\alpha + \beta^\tau x_i)]^{c_i} \cdot [1 - logit^{-1}(\alpha + \beta^\tau x_i)]^{1-c_i}$$

Classification with Logit

$$P(c_i|x_i) = [logit^{-1}(\alpha + \beta^\tau x_i)]^{c_i} \cdot [1 - logit^{-1}(\alpha + \beta^\tau x_i)]^{1-c_i}$$

$\boldsymbol{x_i}$ is the vector of features for user ***i***
1s for URLs visited, 0s for the rest

$\boldsymbol{c_i}$ is the class (or label) for user ***i***
1 for clicked, 0 for did not click

Classification with Logit

$$P(c_i|x_i) = [logit^{-1}(\alpha + \beta^\tau x_i)]^{c_i} \cdot [1 - logit^{-1}(\alpha + \beta^\tau x_i)]^{1-c_i}$$

If $\boldsymbol{c_i}$ **= 0**, the first term is canceled, if **1** the second term is canceled

Classification with Logit

$$P(c_i = 1 | x_i) = logit^{-1}(\alpha + \beta^\tau x_i)$$

To convert this to a linear function, we can take the log of the odds ratio

$$log\left(\frac{P(c_i = 1|x_i)}{1 - P(c_i = 1|x_i)}\right) = \alpha + \beta^\tau x_i$$

Classification with Logit

$$log\left(\frac{P(c_i = 1|x_i)}{1 - P(c_i = 1|x_i)}\right) = \alpha + \beta^\tau x_i$$

This can be rewritten as...

$$logit(P(c_i = 1|x_i)) = \alpha + \beta^\tau x_i$$

Classification with Logit

$$logit(P(c_i = 1|x_i)) = \alpha + \beta^T x_i$$

Now we have a model we can fit to find α, β

Fitting our Model

- As with linear regression, the math behind fitting a model is outside the scope of the class
- SciKit Learn has methods for fitting Python data using logistic regression

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Another Example

Given one possible logistic regression equation for lung cancer odds:

logit(p) = -4.48 + 0.11 x AGE + 1.16 x SMOKING

With this model, 40 year olds that smoke have a logit(p) value of 1.08

We can perform a back transformation to get the actual probability

Or look it up in a table to find out for logit(p) = 1.08, p = 0.75

Reference: https://www.medcalc.org/manual/logistic-regression.php