Week 3

Agenda

1. Async Review with Quizzes
2. Feature engineering Part 1
3. Project 1
4. Case Study: Spam filtering
5. Breakout on Graham article
6. Back to Async Review

Quizzes

What makes naive Bayes "naive"?

- It usually doesn't work very well.
- The assumption that the classes are independent.
- The assumption that the features are independent given the class.

Feature selection is important because

- It can remove poorly estimated features.
- It keeps only the features that give the optimal performance.

Summing log probabilities is equivalent to multiplying probabilities.

- True
- False

A perfectly calibrated classifier is ____% accurate on examples where the posterior probability is 85%.

What is the Laplace (with k=1) smoothed estimate for P(sun) given this data: domain: {sun,rain,wind} observations: [sun,rain,rain,wind,sun,sun]

In our one-feature spam classifier, we have made no assumptions of independence.

- True
- False

Naive Bayes Review

Bayes's Rule

- Update our belief about X, given evidence E.

$$\begin{aligned} P(X|E) &= P(X, E) / P(E) \text{ (apply the definition)} \\ &= P(E, X) / P(E) \text{ (reorder the variables)} \\ &= P(X) P(E|X) / P(E) \text{ (apply the definition)} \end{aligned}$$

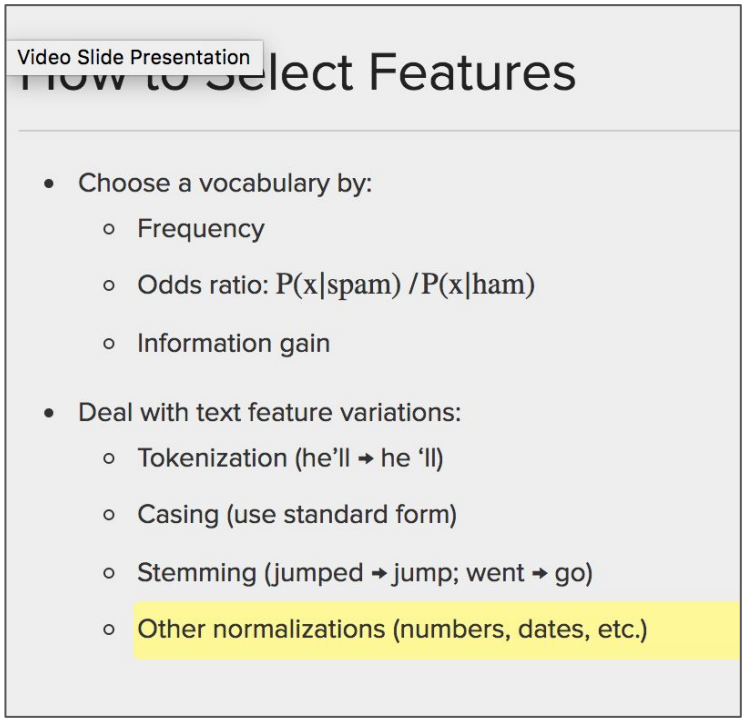
- Terminology:
  - **Prior**: $P(X)$
  - **Posterior**: $P(X|E)$
  - **Likelihood**: $P(E|X)$
- Why is Bayes's rule helpful?
  - Often one conditional is easier to come by than the other.

1. Think how NB training step works: How long does it take to train? Think relative to the size of training data. Can training be parallelized?
2. How fast is the trained model at making predictions?
   a. How does that compare to KNN?
3. An online learner is one that can update its parameters incrementally given new examples.
   a. Why can Naive Bayes be thought of as an 'online model'?
   b. Is KNN an 'online model'?

Case Study: Spam classification

- Summarize the problem
- How would you approach it?

Feature Engineering



1. What might make a feature a 'good' feature?
   a. For house price prediction?
   b. For mortgage approval classification?
   c. Restaurant recommendation?
2. What is feature selection?
   a. Why does it matter with Naive Bayes?
3. How was spam classified before "always on" Internet?

Project 1

Examples

```
>>> X = [[0], [1], [2], [3]]
>>> y = [0, 0, 1, 1]
>>> from sklearn.neighbors import KNeighborsClassifier
>>> neigh = KNeighborsClassifier(n_neighbors=3)
>>> neigh.fit(X, y)
KNeighborsClassifier(...)
>>> print(neigh.predict([[1.1]]))
[0]
>>> print(neigh.predict_proba([[0.9]]))
[[0.66666667 0.33333333]]
```

Methods

`fit`(self, X, y)	Fit the model using X as training data and y as target values
`get_params`(self[, deep])	Get parameters for this estimator.
`kneighbors`(self[, X, n_neighbors, ...])	Finds the K-neighbors of a point.
`kneighbors_graph`(self[, X, n_neighbors, mode])	Computes the (weighted) graph of k-Neighbors for points in X
`predict`(self, X)	Predict the class labels for the provided data.
`predict_proba`(self, X)	Return probability estimates for the test data X.
`score`(self, X, y[, sample_weight])	Return the mean accuracy on the given test data and labels.
`set_params`(self, **params)	Set the parameters of this estimator.

Graham Article

(http://www.paulgraham.com/spam.html)

1. What are the classes? What are the features? What is the evaluation?
2. What are some of the engineering 'hacks' Graham makes?
3. How fast would his model be to train? Predict? Retrain with addition of one email to training data? (general idea, not exact "numbers")

> I don't know why I avoided trying the statistical approach for so long. I think it was because I got addicted to trying to identify spam features myself, as if I were playing some kind of competitive game with the spammers. (Nonhackers don't often realize this, but most hackers are very competitive.) When I did try statistical analysis, I found immediately that it was much cleverer than I had been. It discovered, of course, that terms like "virtumundo" and "teens" were good indicators of spam. But it also discovered that "per" and "FL" and "ff0000" are good indicators of spam. In fact, "ff0000" (html for bright red) turns out to be as good an indicator of spam as any pornographic term.

Graham Article

(http://www.paulgraham.com/spam.html)

> Here's a sketch of how I do statistical filtering. I start with one corpus of spam and one of nonspam mail. At the moment each one has about 4000 messages in it. I scan the entire text, including headers and embedded html and javascript, of each message in each corpus. I currently consider alphanumeric characters, dashes, apostrophes, and dollar signs to be part of tokens, and everything else to be a token separator. (There is probably room for improvement here.) I ignore tokens that are all digits, and I also ignore html comments, not even considering them as token separators.
>
> I count the number of times each token (ignoring case, currently) occurs in each corpus. At this stage I end up with two large hash tables, one for each corpus, mapping tokens to number of occurrences.

1. What are stemming and tokenization?
2. What do stemming and tokenization accomplish from a machine learning perspective?
3. Why do you think Naive Bayes is a popular (baseline) choice for doing text classification?
4. Is spam filtering (necessarily) text classification? (i.e. you build a spam filter without using text as features)?
5. What might be some non-textual features that provide evidence of spam?

Spam Classification

Naive Bayes for Spam

- Here's our model:
  - Wi is the word at position i in the input.
  - $P(Y|X) \sim P(Y)\,P(W_1|Y)\,P(W_2|Y) \ldots P(W_n|Y)$
- "Bag of Words" (BOW) assumption:
  - Usually, each feature has its own distribution: $P(F_i|Y)$
  - Here, each position has the same distribution: $P(W|Y)$
  - Keeps the number of parameters manageable.

Spam Classification Example

Feature	P(f|spam)	P(f|ham)	Total Spam	Total Ham
(prior)	0.4000	0.6000	−0.92	−0.51
dear	0.0013	0.0009	−7.56	−7.52
sir	0.0023	0.0004	−13.64	−15.35
,	0.0220	0.0241	−17.45	−19.07
first	0.0018	0.0023	−23.77	−25.15
I	0.0062	0.0119	−28.86	−29.57
must	0.0034	0.0028	−34.54	−35.45
solicit	0.0007	0.0002	−41.08	−43.97

- We use log probabilities to prevent underflow.
- $P(\text{spam}|X) = 0.95$
- $P(\text{ham}|X) = 0.05$
- $\text{prediction} = \text{argmax}_y \log P(y) + \sum_i \log P(F_i|Y)$

1. Why do you think Naive Bayes might be so closely associated with text classification? What about NB makes it well suited to working with text based features?
2. Where do the training labels come from in a commercial spam filter?

Generative Modeling

Generative Story for Naive Bayes

- Naive Bayes is a generative model:
  - $P(Y|X) \sim P(Y)P(W_1|Y)P(W_2|Y)...P(W_n|Y)$
- To generate a document:
  - Pick a class spam/ham according to $P(Y)$.
  - Repeat until you have enough words:
    - Pick a word according to $P(W|Y)$.
- Note: Not all models are generative.
  - E.g., logistic regression: not a generative model; it models posterior distribution $P(Y|X)$ directly.

1. What does it mean to be a generative model? Why isn't KNN a generative model?
2. If you generate emails with a NB spam detector, what might those emails look like?

Exercise Laplace Smoothing

The next three slides contain examples for different k.

LaPlace Smoothing Single Variable Distribution

Given data: a1, a2, a1, a2, a3, a1, a3, a2

Domain: A = {a1, a2, a3}

Wanted: estimate of P(A) with Laplace smoothing with k = 1

Try to complete the following slides yourself! Stuck? Come to office hours again :)

examples taken from Pieter Abbeel

LaPlace Smoothing Single Variable Distribution

Given data: a1, a2, a1, a2, a3, a1, a3, a2

Domain: A = {a1, a2, a3}

Wanted: estimate of P(A) with Laplace smoothing with k = 1

P(A=a1) = (ML) ⅜ -> LP	with k = 1 -> = (3+1) / (8+3*1) = 4/11 -> domain size = 3, k=1, so 3*1
P(A=a2) = ⅜	with k = 1 -> = (3+1) / (8+3*1) = 4/11
P(A=a3) = 2/8	with k = 1 -> = (2+1) / (8+3*1) = 3/11

examples taken from Pieter Abbeel

LaPlace Smoothing Single Variable Distribution

Given data: a1, a2, a1, a2, a3, a1, a3, a2

Domain: A = {a1, a2, a3}

Wanted: estimate of P(A) with Laplace smoothing with k = 2

LaPlace Smoothing Conditional Distribution

Given data: (a1, b1), (a2, b2), (a1, b3), (a1, b3), (a2, b2), (a1, b2), (a1, b1)

Domain: A = {a1, a2}, B = {b1, b2, b3}

Wanted: estimate of P(B|A) with Laplace smoothing with k = 2

A=a1: (a1, b1), (a1, b3), (a1,b3),(a1,b2), (a1,b1), (a1,b1),(a1,b1),(a1,b2),(a1,b2),(a1,b3),(a1,b3)
A=a2: (a1,b1), (a1,b3),...

P(b1|A=a1) = 4/11, P(b2|A=a1) = 3/11, P(b3|A1) = 4/11
P(b2|A=a1) = ...