Announcements

- Assignments:
  - P3 in glookup
  - W3 (shortened) is up, due 11/23
  - P5 will be out later this week

- Contest status:
  - Rank page!
  - Achievements page!
  - Minor tweaks?

Survey Responses

- Most favorite aspects: projects, demos, lectures
- Least favorite aspects: writtens, sections, exams
- Specific things:
  - Writtens: fewer smaller writtens?
  - Writtens: writtens more like the exams?
  - Sections: positive comments about how, mixed comments about what
  - Sections: handouts merging with writtens?
  - Midterm: “hard” “fair” “long”, compare to previous semesters?
  - Webcast frame rate, “can’t see demos at 1 fps”
  - Readings: mixed, “there are readings?”
  - Office hours: “don’t usually go, but helpful when I do”
  - Grading scales, etc.?

New Proposals

- Change lecture format to mini-vids?
  - Mixed reaction, more negative than positive, worry about whether people would actually watch the prep videos
  - “doesn’t sound like it’d work out very well”
  - “HORRIBLE! Noooo!”
  - “That sounds pretty awesome.”

- Multiple section types?
  - The more the better (28), Only one (29), other answers (15)

Example: Spam Filter

- Input: email
- Output: spam/ham
- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails
- Features: The attributes used to make the ham/spam decision
  - Words: FREE!
  - Text Patterns: $dd, CAPS
  - Non-text: SenderInContacts

Example: Digit Recognition

- Input: images / pixel grids
- Output: a digit 0-9
- Setup:
  - Get a large collection of example images, each labeled with a digit
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future digit images
- Features: The attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape Patterns: NumComponents,AspectRatio,NumLoops
  - ...
A Digit Recognizer

- Input: pixel grids
- Output: a digit 0-9

Naïve Bayes for Digits

- Simple version:
  - One feature \( F_{ij} \) for each grid position \(<i,j>\)
  - Possible feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
  - Each input maps to a feature vector, e.g.
    \[ \vec{f} = (f_{0,0}, f_{1,1}, ... f_{15,15}) \]
  - Here: lots of features, each is binary valued

- Naïve Bayes model:
  \[ P(Y|F_{0,0}, F_{15,15}) = P(Y) \prod_{i,j} P(F_{ij}|Y) \]

- What do we need to learn?

General Naïve Bayes

- A general naïve Bayes model:
  \[ P(Y, F_1 \ldots F_n) = \frac{P(Y)}{\prod_i P(F_i|Y)} \]

- We only specify how each feature depends on the class
- Total number of parameters is linear in \( n \)

Inference for Naïve Bayes

- Goal: compute posterior over causes
- Step 1: get joint probability of causes and evidence
  \[ P(Y, f_1 \ldots f_n) = \frac{P(Y) \prod_i P(F_i|Y)}{P(f_1 \ldots f_n)} \]

- Step 2: get probability of evidence
  \[ P(Y|f_1 \ldots f_n) = \frac{P(Y) \prod_i P(F_i|Y)}{P(f_1 \ldots f_n)} \]

- Step 3: renormalize

General Naïve Bayes

- What do we need in order to use naïve Bayes?
  \[ P(Y) \]
  \[ P(F_{ij}|Y) \]

- Inference (you know this part)
  - Start with a bunch of conditionals, \( P(Y) \) and the \( P(F_i|Y) \) tables
  - Use standard inference to compute \( P(Y|F_{ij}, F_{ik}) \)
  - Nothing new here

- Estimates of local conditional probability tables
  - \( P(Y) \), the prior over labels
  - \( P(F_i|Y) \) for each feature (evidence variable)
  - These probabilities are collectively called the parameters of the model and denoted by \( \theta \)
  - Up until now, we assumed these appeared by magic, but...
  - ...they typically come from training data: we’ll look at this now

Examples: CPTs
Important Concepts

- Data: labeled instances, e.g. emails marked spam/ham
- Training set
- Held-out set
- Test set
- Features: attribute-value pairs which characterize each x

Experiments cycle
- Learn parameters (e.g. model probabilities) on training set
- Compute accuracy of test set
- Very important: never “peek” at the test set!

Evaluation
- Accuracy: fraction of instances predicted correctly
- Overfitting and generalization
- Want a classifier which does well on test data

Naïve Bayes spam filter

Data:
- Collection of emails, labeled spam or ham
- Note: someone has to hand label all this data!

Split into training, held-out, test sets

Classifiers
- Learn on the training set
- (Tune it on a held-out set)
- Test it on new emails

Naïve Bayes for Text

- Bag-of-Words Naïve Bayes:
  - Predict unknown class label (spam vs. ham)
  - Assume evidence features (e.g. the words) are independent
  - Warning: subtly different assumptions than before!

Generative model

\[ P(C, W_1, \ldots, W_n) = P(C) \prod P(W_i|C) \]

Tied distributions and bag-of-words
- Usually, each variable gets its own conditional probability distribution \( P(F|Y) \)
- In a bag-of-words model:
  - Each position is identically distributed
  - All positions share the same conditional probs \( P(W|C) \)
- Why make this assumption?

Spam Example

| Word | P(w|spam) | P(w|ham) | Tot Spam | Tot Ham |
|------|-----------|----------|----------|---------|
| (prior) | 0.33333 | 0.66666 | -1.1 | -0.4 |

\( P(\text{spam} | w) = 98.9 \)

Example: Spam Filtering

Model:

\[ P(C, W_1, \ldots, W_n) = P(C) \prod P(W_i|C) \]

What are the parameters?

| \( P(C) \) | \( P(W|\text{spam}) \) | \( P(W|\text{ham}) \) |
|--|--|--|
| ham : 0.66 | the : 0.0136 | the : 0.0210 |
| spam : 0.33 | to : 0.0153 | to : 0.0138 |
| and : 0.0115 | of : 0.0119 | of : 0.0129 |
| you : 0.0093 | 2002: 0.0110 | 2002: 0.0100 |
| a : 0.0086 | with : 0.0108 | with : 0.0105 |
| from : 0.0075 | a : 0.0107 | a : 0.0100 |
| ... | ... | ... |

Where do these tables come from?

Example: Overfitting

\( P(\text{features}, C = 2) \)

<table>
<thead>
<tr>
<th>( P(C = 2) )</th>
<th>( P(C = 3) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(\text{features}, C = 2) )</td>
<td>( P(\text{features}, C = 3) )</td>
</tr>
<tr>
<td>( P(\text{off}</td>
<td>C = 2) = 0.1 )</td>
</tr>
<tr>
<td>( P(\text{on}</td>
<td>C = 2) = 0.8 )</td>
</tr>
<tr>
<td>( P(\text{on}</td>
<td>C = 2) = 0.1 )</td>
</tr>
<tr>
<td>( P(\text{off}</td>
<td>C = 2) = 0.1 )</td>
</tr>
<tr>
<td>( P(\text{on}</td>
<td>C = 2) = 0.01 )</td>
</tr>
</tbody>
</table>

2 wins!!
Example: Overfitting

- Postiors determined by relative probabilities (odds ratios):

\[
\begin{align*}
P(W|\text{ham}) & \\ P(W|\text{spam}) & \\ \text{south-west} : & \text{inf} \\ \text{nation} : & \text{inf} \\ \text{morally} : & \text{inf} \\ \text{nicely} : & \text{inf} \\ \text{seriously} : & \text{inf} \\ \text{...} & \\ P(W|\text{ham}) & \\ P(W|\text{spam}) & \\ \text{screens} : & \text{inf} \\ \text{minute} : & \text{inf} \\ \text{guaranteed} : & \text{inf} \\ \text{$205.00} : & \text{inf} \\ \text{delivery} : & \text{inf} \\ \text{signature} : & \text{inf} \\ \text{...} & 
\end{align*}
\]

What went wrong here?

Generalization and Overfitting

- Relative frequency parameters will overfit the training data!
  - Just because we never saw a 3 with pixel (15,15) on during training doesn’t mean we won’t see it at test time
  - Unlikely that every occurrence of “minute” is 100% spam
  - Unlikely that every occurrence of “seriously” is 100% ham
  - What about all the words that don’t occur in the training set at all?
  - In general, we can’t go around giving unseen events zero probability

- As an extreme case, imagine using the entire email as the only feature
  - Would get the training data perfect (if deterministic labeling)
  - Wouldn’t generalize at all
  - Just making the bag-of-words assumption gives us some generalization, but isn’t enough

- To generalize better: we need to smooth or regularize the estimates

Estimation: Smoothing

- Maximum likelihood estimates:

\[
P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}
\]

- Problems with maximum likelihood estimates:
  - If I flip a coin once, and it's heads, what’s the estimate for $P(\text{heads})$?
  - What if I flip 10 times with 8 heads?
  - What if I flip 100M times with 80M heads?

- Basic idea:
  - We have some prior expectation about parameters (here, the probability of heads)
  - Given little evidence, we should skew towards our prior
  - Given a lot of evidence, we should listen to the data

Estimation: Laplace Smoothing

- Laplace’s estimate (extended):

\[
P_{\text{LAP}}(x) = \frac{\text{count}(x) + k}{\text{total samples} + k}
\]

- What’s Laplace with $k = 0$?
  - $k$ is the strength of the prior

- Laplace for conditionals:

\[
P_{\text{LAP}}(x|y) = \frac{\text{count}(x, y) + k}{\text{count}(y) + k}
\]

Real NB: Smoothing

- For real classification problems, smoothing is critical

- New odds ratios:

\[
\begin{align*}
P(W|\text{ham}) & \\ P(W|\text{spam}) & \\ \text{helvetica} : & 11.4 \\ \text{seems} : & 10.8 \\ \text{group} : & 10.2 \\ \text{ago} : & 8.4 \\ \text{areas} : & 8.3 \\ \text{...} & \\ \text{verdana} : & 28.8 \\ \text{Credit} : & 28.4 \\ \text{ORDER} : & 27.2 \\ \text{<FONT>} : & 26.9 \\ \text{money} : & 26.5 \\ \text{...} & 
\end{align*}
\]

Do these make more sense?
Tuning on Held-Out Data

- Now we’ve got two kinds of unknowns
  - Parameters: the probabilities \( P(Y|X), P(Y) \)
  - Hyperparameters, like the amount of smoothing to do: \( \alpha \)

- Where to learn?
  - Learn parameters from training data
  - Must tune hyperparameters on different data

- Why?
  - For each value of the hyperparameters, train and test on the held-out data
  - Choose the best value and do a final test on the test data

Errors, and What to Do

- Examples of errors

```
Dear GlobalSCAPE Customer,

GlobalSCAPE has partnered with ScanSoft to offer you the latest version of OmniPage Pro, for just $99.99* - the regular list price is $499! The most common question we’ve received about this offer is - Is this genuine? We would like to assure you that this offer is authorized by ScanSoft, is genuine and valid. You can get the . . .

. . . To receive your $30 Amazon.com promotional certificate, click through to http://www.amazon.com/apparel and see the prominent link for the $30 offer. All details are there. We hope you enjoyed receiving this message. However, if you’d rather not receive future e-mails announcing new store launches, please click . . .
```

What to Do About Errors?

- Need more features – words aren’t enough!
  - Have you emailed the sender before?
  - Have 1K other people just gotten the same email?
  - Is the sending information consistent?
  - Is the email in ALL CAPS?
  - Do inline URLs point where they say they point?
  - Does the email address you by (your) name?

- Can add these information sources as new variables in the NB model

- Next class we’ll talk about classifiers which let you easily add arbitrary features more easily