Announcements

- W4 out, due next week Monday
- P4 out, due next week Friday
- Mid-semester survey
Announcements II

- Course contest

- Regular tournaments. Instructions have been posted!
- First week extra credit for top 20, next week top 10, then top 5, then top 3.
- First nightly tournament: tentatively Monday night

P4: Ghostbusters 2.0

- **Plot:** Pacman's grandfather, Grandpac, learned to hunt ghosts for sport.

- He was blinded by his power, but could hear the ghosts’ banging and clanging.

- **Transition Model:** All ghosts move randomly, but are sometimes biased

- **Emission Model:** Pacman knows a “noisy” distance to each ghost

![Noisy distance prob]

<table>
<thead>
<tr>
<th>True distance = 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy distance prob</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>
Today

- Dynamic Bayes Nets (DBNs)
  - [sometimes called temporal Bayes nets]

- Demos:
  - Localization
  - Simultaneous Localization And Mapping (SLAM)

- Start machine learning

Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time \( t \) can condition on those from \( t-1 \)
- Discrete valued dynamic Bayes nets are also HMMs
Exact Inference in DBNs

- Variable elimination applies to dynamic Bayes nets
- Procedure: “unroll” the network for T time steps, then eliminate variables until $P(X_T|e_{1:T})$ is computed

Online belief updates: Eliminate all variables from the previous time step; store factors for current time only

DBN Particle Filters

- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the $t=1$ Bayes net
  - Example particle: $G_1^a = (3,3)$, $G_1^b = (5,3)$
- **Elapse time**: Sample a successor for each particle
  - Example successor: $G_2^a = (2,3)$, $G_2^b = (6,3)$
- **Observe**: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(E_1^a|G_1^a) \times P(E_1^b|G_1^b)$
- **Resample**: Select prior samples (tuples of values) in proportion to their likelihood
DBN Particle Filters

- A particle is a complete sample for a time step
- Initialize: Generate prior samples for the t=1 Bayes net
  - Example particle: \( G_1^a = (3,3) \quad G_1^b = (5,3) \)
- Elapse time: Sample a successor for each particle
  - Example successor: \( G_2^a = (2,3) \quad G_2^b = (6,3) \)
- Observe: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: \( P(E_1^a | G_1^a) \cdot P(E_1^b | G_1^b) \)
- Resample: Select prior samples (tuples of values) in proportion to their likelihood

Trick I to Improve Particle Filtering Performance: Low Variance Resampling

![Figure 4.7](image)

- Advantages:
  - More systematic coverage of space of samples
  - If all samples have same importance weight, no samples are lost
  - Lower computational complexity
Trick II to Improve Particle Filtering Performance: Regularization

- If no or little noise in transitions model, all particles will start to coincide

→ regularization: introduce additional (artificial) noise into the transition model

---

SLAM

- SLAM = Simultaneous Localization And Mapping
  - We do not know the map or our location
  - Our belief state is over maps and positions!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

- [DEMOS]

DP-SLAM, Ron Parr
Robot Localization

- In robot localization:
  - We know the map, but not the robot’s position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store B(X)
  - Particle filtering is a main technique

[Demos]
Global-floor

SLAM

- SLAM = Simultaneous Localization And Mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods
Particle Filter Example

- 3 particles

map of particle 1

map of particle 2

map of particle 3

SLAM

- DEMOS
  - fastslam.avi, visionSlam_heliOffice.wmv
Further readings

- We are done with Part II Probabilistic Reasoning
- To learn more (beyond scope of 188):
  - Koller and Friedman, Probabilistic Graphical Models (CS281A)
  - Thrun, Burgard and Fox, Probabilistic Robotics (CS287)

Part III: Machine Learning

- Up until now: how to reason in a model and how to make optimal decisions
- Machine learning: how to acquire a model on the basis of data / experience
  - Learning parameters (e.g. probabilities)
  - Learning structure (e.g. BN graphs)
  - Learning hidden concepts (e.g. clustering)
Machine Learning Today

- An ML Example: Parameter Estimation
  - Maximum likelihood
  - Smoothing
- Applications
- Main concepts
- Naïve Bayes

Parameter Estimation

- Estimating the distribution of a random variable
- *Elicitation:* ask a human (why is this hard?)
- *Empirically:* use training data (learning!)
  - E.g.: for each outcome x, look at the *empirical rate* of that value:
    \[ P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}} \]
    \[ P_{ML}(r) = 1/3 \]
  - This is the estimate that maximizes the *likelihood of the data*
    \[ L(x, \theta) = \prod_i P_{\theta}(x_i) \]
  - *Issue:* overfitting. E.g., what if only observed 1 jelly bean?
Estimation: Smoothing

- Relative frequencies are the maximum likelihood estimates

\[ \hat{\theta}_{ML} = \arg \max_{\theta} P(X|\theta) \]
\[ = \arg \max_{\theta} \prod_i P_{\theta}(X_i) \]
\[ \Rightarrow P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}} \]

- In Bayesian statistics, we think of the parameters as just another random variable, with its own distribution

\[ \hat{\theta}_{MAP} = \arg \max_{\theta} P(\theta|X) \]
\[ = \arg \max_{\theta} P(X|\theta)P(\theta)/P(X) \]
\[ = \arg \max_{\theta} P(X|\theta)P(\theta) \]

Estimation: Laplace Smoothing

- Laplace’s estimate:
  - Pretend you saw every outcome once more than you actually did

\[ P_{LAP}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]} \]
\[ \Rightarrow P_{ML}(X) = \]
\[ \Rightarrow P_{LAP}(X) = \]

- Can derive this as a MAP estimate with Dirichlet priors (see cs281a)
Estimation: Laplace Smoothing

- **Laplace’s estimate (extended):**
  - Pretend you saw every outcome \( k \) extra times
  \[
P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}
\]
  - What’s Laplace with \( k = 0 \)?
  - \( k \) is the strength of the prior

- **Laplace for conditionals:**
  - Smooth each condition independently
  \[
P_{LAP,k}(x|y) = \frac{c(x, y) + k}{c(y) + k|X|}
\]

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
  - …

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok. I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.
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
  - ...

Other Classification Tasks

- In classification, we predict labels y (classes) for inputs x
- Examples:
  - Spam detection (input: document, classes: spam / ham)
  - OCR (input: images, classes: characters)
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Automatic essay grader (input: document, classes: grades)
  - Fraud detection (input: account activity, classes: fraud / no fraud)
  - Customer service email routing
  - ... many more
- Classification is an important commercial technology!
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 \( x \)
- **Experimentation cycle**
  - Learn parameters (e.g. model probabilities) on training set
  - (Tune hyperparameters on held-out 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
  - Overfitting: fitting the training data very closely, but not generalizing well
  - We’ll investigate overfitting and generalization formally in a few lectures

---

Bayes Nets for Classification

- **One method of classification**:
  - Use a probabilistic model!
  - Features are observed random variables \( F_i \)
  - \( Y \) is the query variable
  - Use probabilistic inference to compute most likely \( Y \)

\[
y = \arg\max_y P(y|f_1\ldots f_n)
\]

- You already know how to do this inference
Simple Classification

- Simple example: two binary features

\[ P(m|s, f) \quad \text{direct estimate} \]

\[ P(m|s, f) = \frac{P(s, f|m)P(m)}{P(s, f)} \quad \text{Bayes estimate (no assumptions)} \]

\[ P(m|s, f) = \frac{P(s|m)P(f|m)P(m)}{P(s, f)} \quad \text{Conditional independence} \]

\[ P(+m, s, f) = P(s|m)P(f|m)P(+m) \]

\[ P(-m, s, f) = P(s|m)P(f|m)P(-m) \]

General Naïve Bayes

- A general *naïve Bayes* model:

\[ P(Y, F_1 \ldots F_n) = \]

\[ P(Y) \prod_i P(F_i|Y) \]

\[ |Y| \text{ parameters} \quad n \times |F| \times |Y| \text{ parameters} \]

- 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
    \[
    P(Y, f_1 \ldots f_n) = \frac{P(y_1, f_1 \ldots f_n)}{P(y_2, f_1 \ldots f_n)} \cdots \frac{P(y_k, f_1 \ldots f_n)}{P(y_k) \prod_i P(f_i|y_i)} + \frac{P(y_1) \prod_i P(f_i|y_1)}{P(y_2) \prod_i P(f_i|y_2)} \cdots \frac{P(y_k) \prod_i P(f_i|y_k)}{P(y_k)}
    \]
  - Step 2: get probability of evidence
  - Step 3: renormalize

General Naïve Bayes

- What do we need in order to use naïve Bayes?
  - 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 \(P(Y|F_1 \ldots F_n)\)
    - 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
A Digit Recognizer

- Input: pixel grids

![Pixel Grid Example]

- 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.
    
    $1 \rightarrow (F_{0,0} = 0 \ F_{0,1} = 0 \ F_{0,2} = 1 \ F_{0,3} = 1 \ F_{0,4} = 0 \ \ldots \ F_{15,15} = 0)$

  - Here: lots of features, each is binary valued

- Naïve Bayes model:
  
  $P(Y|F_{0,0} \ldots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$

- What do we need to learn?
Examples: CPTs

Parameter Estimation

- Estimating distribution of random variables like \( X \) or \( X | Y \)
- **Empirically**: use training data
  - For each outcome \( x \), look at the empirical rate of that value:

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

\[
P_{\text{ML}}(r) = 1/3
\]

- This is the estimate that maximizes the likelihood of the data

\[
L(x, \theta) = \prod_i P_\theta(x_i)
\]

- **Elicitation**: ask a human!
  - Usually need domain experts, and sophisticated ways of eliciting probabilities (e.g. betting games)
  - Trouble calibrating
A Spam Filter

- 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

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, I know this is blatantly OT but I’m beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

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
  \[ P(Y, W_1 \ldots W_n) = P(Y) \prod_i P(W_i | Y) \]

- 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?

Word at position \( i \), not \( i \)th word in the dictionary!
Example: Spam Filtering

- **Model:** \( P(Y, W_1 \ldots W_n) = P(Y) \prod_i P(W_i|Y) \)
- **What are the parameters?**

| \( P(Y) \) | \( P(W|\text{spam}) \) | \( P(W|\text{ham}) \) |
| --- | --- | --- |
| ham : 0.66 | the : 0.0156 | the : 0.0210 |
| spam: 0.33 | to : 0.0153 | to : 0.0133 |
| | and : 0.0115 | of : 0.0119 |
| | of : 0.0095 | 2002: 0.0110 |
| | you : 0.0093 | with: 0.0108 |
| | a : 0.0086 | from: 0.0107 |
| | with: 0.0080 | and : 0.0105 |
| | from: 0.0075 | a : 0.0100 |
| | ... | ... |

- **Where do these tables come from?**

Spam Example

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

\[ P(\text{spam} \mid w) = 98.9 \]
Example: Overfitting

\[
P(\text{features}, Y = 2) \quad P(\text{features}, Y = 3)
\]
\[
P(Y = 2) = 0.1 \
P(Y = 3) = 0.1
\]
\[
P(\text{on} | Y = 2) = 0.8 \
P(\text{on} | Y = 3) = 0.8
\]
\[
P(\text{on} | Y = 2) = 0.1 \
P(\text{on} | Y = 3) = 0.9
\]
\[
P(\text{off} | Y = 2) = 0.1 \
P(\text{off} | Y = 3) = 0.7
\]
\[
P(\text{on} | Y = 2) = 0.01 \
P(\text{on} | Y = 3) = 0.0
\]

2 wins!!

Example: Overfitting

- Posterials determined by relative probabilities (odds ratios):

\[
\frac{P(W|\text{ham})}{P(W|\text{spam})} \quad \frac{P(W|\text{spam})}{P(W|\text{ham})}
\]

| south-west : inf | screens : inf |
| nation : inf | minute : inf |
| morally : inf | guaranteed : inf |
| nicely : inf | $205.00 : inf |
| extent : inf | delivery : inf |
| seriously : inf | signature : inf |

... 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

- Problems with maximum likelihood estimates:
  - If I flip a coin once, and it’s heads, what’s the estimate for P (heads)?
  - What if I flip 10 times with 8 heads?
  - What if I flip 10M times with 8M 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: Smoothing

- Relative frequencies are the maximum likelihood estimates

\[
\theta_{ML} = \arg\max_{\theta} P(X|\theta) \quad \Rightarrow \quad P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}
\]

- In Bayesian statistics, we think of the parameters as just another random variable, with its own distribution

\[
\theta_{MAP} = \arg\max_{\theta} P(\theta|X) = \arg\max_{\theta} \frac{P(X|\theta)P(\theta)}{P(X)} = \arg\max_{\theta} P(X|\theta)P(\theta)
\]

Estimation: Laplace Smoothing

- Laplace's estimate:
  - Pretend you saw every outcome once more than you actually did

\[
P_{LAP}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]} \quad \Rightarrow \quad P_{ML}(X) = \frac{1}{N + \vert X \vert}
\]

- Can derive this as a MAP estimate with Dirichlet priors (see cs281a)
Estimation: Laplace Smoothing

- **Laplace’s estimate (extended):**
  - Pretend you saw every outcome $k$ extra times:
    \[ P_{\text{LAP},k}(x) = \frac{c(x) + k}{N + k|X|} \]
  - **$P_{\text{LAP},0}(X) =$**
  - **$P_{\text{LAP},1}(X) =$**
  - **$P_{\text{LAP},100}(X) =$**
  - What’s Laplace with $k = 0$?
  - $k$ is the strength of the prior

- **Laplace for conditionals:**
  - Smooth each condition:
    \[ P_{\text{LAP},k}(x|y) = \frac{c(x, y) + k}{c(y) + k|X|} \]

Estimation: Linear Interpolation

- **In practice, Laplace often performs poorly for $P(X|Y)$:**
  - When $|X|$ is very large
  - When $|Y|$ is very large

- **Another option: linear interpolation**
  - Also get $P(X)$ from the data
  - Make sure the estimate of $P(X|Y)$ isn’t too different from $P(X)$

\[
P_{\text{LIN}}(x|y) = \alpha \hat{P}(x|y) + (1.0 - \alpha) \hat{P}(x)
\]

- What if $\alpha$ is 0? 1?

- For even better ways to estimate parameters, as well as details of the math see cs281a, cs288
Real NB: Smoothing

- For real classification problems, smoothing is critical
- New odds ratios:

\[
\frac{P(W|\text{ham})}{P(W|\text{spam})} \quad \frac{P(W|\text{spam})}{P(W|\text{ham})}
\]

<table>
<thead>
<tr>
<th>Example</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>helvetica</td>
<td>11.4</td>
<td>verdana</td>
</tr>
<tr>
<td>seems</td>
<td>10.8</td>
<td>Credit</td>
</tr>
<tr>
<td>group</td>
<td>10.2</td>
<td>ORDER</td>
</tr>
<tr>
<td>ago</td>
<td>8.4</td>
<td>&lt;FONT&gt;</td>
</tr>
<tr>
<td>areas</td>
<td>8.3</td>
<td>money</td>
</tr>
</tbody>
</table>
| ...      | ...   | ...     | ...

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: k, α

- 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
Baselines

- **First step: get a baseline**
  - Baselines are very simple “straw man” procedures
  - Help determine how hard the task is
  - Help know what a “good” accuracy is

- **Weak baseline: most frequent label classifier**
  - Gives all test instances whatever label was most common in the training set
  - E.g. for spam filtering, might label everything as ham
  - Accuracy might be very high if the problem is skewed
  - E.g. calling everything “ham” gets 66%, so a classifier that gets 70% isn’t very good…

- For real research, usually use previous work as a (strong) baseline

Confidences from a Classifier

- **The confidence of a probabilistic classifier:**
  - Posterior over the top label
  \[
  \text{confidence}(x) = \max_y P(y|x)
  \]
  - Represents how sure the classifier is of the classification
  - Any probabilistic model will have confidences
  - No guarantee confidence is correct

- **Calibration**
  - Weak calibration: higher confidences mean higher accuracy
  - Strong calibration: confidence predicts accuracy rate
  - What’s the value of calibration?
Precision vs. Recall

- Let’s say we want to classify web pages as homepages or not
  - In a test set of 1K pages, there are 3 homepages
  - Our classifier says they are all non-homepages
  - 99.7 accuracy!
  - Need new measures for rare positive events

- Precision: fraction of guessed positives which were actually positive
- Recall: fraction of actual positives which were guessed as positive

- Say we guess 5 homepages, of which 2 were actually homepages
  - Precision: 2 correct / 5 guessed = 0.4
  - Recall: 2 correct / 3 true = 0.67

- Which is more important in customer support email automation?
- Which is more important in airport face recognition?

Precision vs. Recall

- Precision/recall tradeoff
  - Often, you can trade off precision and recall
  - Only works well with weakly calibrated classifiers

- To summarize the tradeoff:
  - Break-even point: precision value when \( p = r \)
  - F-measure: harmonic mean of \( p \) and \( r \):
    \[
    F_1 = \frac{2}{1/p + 1/r}
    \]
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
Summary Naïve Bayes Classifier

- Bayes rule lets us do diagnostic queries with causal probabilities
- The naïve Bayes assumption takes all features to be independent given the class label
- We can build classifiers out of a naïve Bayes model using training data
- Smoothing estimates is important in real systems
- Classifier confidences are useful, when you can get them