Announcements

- Project 4 due Thursday
- Contest up since last night.
  - Nightly tournaments starting 11pm.

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)

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
  - ...  

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
  - $P(M)$
  - $P(S|M)$
  - $P(F|M)$
  - $P(F|S)$
  - $P(F)$

### General Naïve Bayes
- A general naïve Bayes model:
  $$ P(Y, F_1 \ldots F_n) = P(Y) \prod_i P(F_i|Y) $$
  - $|Y|$ parameters
  - $n \times |F|$ 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
    $$ P(Y, f_1 \ldots f_n) = \frac{P(y_1, f_1 \ldots f_n) \cdots P(y_n, f_1 \ldots f_n)}{P(f_1 \ldots f_n)} $$
  - Step 2: get probability of evidence
  - Step 3: renormalize
    $$ P(Y|f_1 \ldots f_n) $$

### 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

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

- Here: lots of features, each is binary valued

Naïve Bayes model:

\[
P(Y|F_{0,0}, F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)
\]

- 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:

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

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

- 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

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 prob \( P(W|C) \)

  - Why make this assumption?
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 to : 0.0153 and : 0.0115 of : 0.0095 you : 0.0093 with : 0.0086 from : 0.0107 | **ham** : 0.33 |
| spam : 0.33 | the : 0.0210 to : 0.0133 and : 0.0119 of : 0.0110 you : 0.0108 with : 0.0105 from : 0.0100 | spam : 0.33 |

- 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 |

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

Example: Overfitting

- Posterior determined by relative probabilities (odds ratios):

| word | \( P(w|\text{spam}) \) | \( P(w|\text{ham}) \) |
|------|-----------------|-----------------|
| south-west : inf | minute : inf | nation : inf |
| morally : inf | guaranteed : inf | $205.00 : inf |
| nicely : inf | delivery : inf | signature : inf |
| seriously : 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.

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

- In Bayesian statistics, we think of the parameters as just another random variable, with its own distribution
  \[ \theta_{MAP} = \arg \max_{\theta} P(\theta|X) \]
  \[ = \arg \max_{\theta} P(X|\theta)P(\theta)/P(X) \]

Estimation: Laplace Smoothing

- Laplace’s estimate:
  \[ P_{LAP}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]} \]
  \[ P_{ML}(X) = \frac{c(x) + 1}{N + |X|} \]

- Can derive this as a MAP estimate with Dirichlet priors (see cs281a)

Estimation: Laplace Smoothing (extended)

- Laplace’s estimate (extended):
  \[ P_{LAP}(x) = \frac{c(x) + k}{N + k|X|} \]
  \[ P_{LAP}(X) = \frac{c(x) + 1}{N + |X|} \]

- What’s Laplace with \( k = 0 \)?
  - \( k \) is the strength of the prior

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