Lecture 20: Dynamic Bayes Nets, Naïve Bayes

Pieter Abbeel – UC Berkeley
Slides adapted from Dan Klein.

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 This Set of Slides

- 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:
    
    $P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}$
    
    $P_{ML}(t) = 1/3$

    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

  $\theta_{ML} = \arg \max_{\theta} P(X|\theta) = \arg \max_{\theta} \prod_i P_\theta(x_i)$

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

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

  $\theta_{MAP} = \arg \max_{\theta} P(\theta|X)$

  $= \arg \max_{\theta} P(X|\theta) P(\theta)/P(X)$

  $= \arg \max_{\theta} P(X|\theta) P(\theta) / P(X)$

Estimation: Laplace Smoothing

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

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

  $P_{ML}(X) = \frac{c(x) + 1}{N + |X|}$

  $P_{LAP}(X) =$

- 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
  - \[ P_{\text{LAP}}(x) = \frac{\nu(x) + k}{N + k|X|} \]
  - What's Laplace with \( k = 0 \)?
  - \( k \) is the strength of the prior

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

**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

**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 on 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

\[ P(m|s, f) = \frac{P(s, f|m)P(m)}{P(s, f)} \]

Bayes estimate (no assumptions)

\[ P(m|s, f) = \frac{P(s|m)P(f|m)P(m)}{P(s, f)} \]

Conditional independence

\[ P(s|m) = \begin{cases} P(+m, s, f) = P(+m)P(f|m)P(+m), \\ P(-m, s, f) = P(-m)P(f|m)P(-m) \end{cases} \]

General Naïve Bayes

- A general naïve Bayes model:

\[ P(Y, F_1 \ldots F_n) = \prod_{i} P(Y|F_i) \]

\[ n \times |F| \times |Y| \] 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

\[ P(Y, F_1 \ldots F_n) = \prod_{i} P(Y|F_i) \prod_{i} P(F_i|Y) \]

- 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 \( F_i \) 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. \( F_{0,0} = 0 \), \( F_{0,1} = 0 \), \( F_{0,2} = 1 \), etc.
  - 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

\[
P(Y) = \begin{bmatrix}
1 & 0.1 \\
2 & 0.1 \\
3 & 0.1 \\
4 & 0.1 \\
5 & 0.1 \\
6 & 0.1 \\
7 & 0.1 \\
8 & 0.1 \\
9 & 0.1 \\
0 & 0.1
\end{bmatrix}
\]

\[
P(W_{Y_1} \mid Y) = \begin{bmatrix}
1 & 0.01 \\
2 & 0.05 \\
3 & 0.20 \\
4 & 0.80 \\
5 & 0.96 \\
6 & 0.90 \\
7 & 0.25 \\
8 & 0.80 \\
9 & 0.80 \\
0 & 0.80
\end{bmatrix}
\]

Parameter Estimation

- Estimating distribution of random variables like \(X\) or \(X \mid Y\)
- **Empirically**: use training data
  - For each outcome \(x\), look at the empirical rate of that value:
    \[
    \hat{p}_n(x) = \frac{\text{count}(x)}{\text{total samples}}
    \]
    - 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**
    \[
    P(Y, W_1 \ldots W_n) = P(Y) \prod_i P(W_i \mid Y)
    \]
  - Tied distributions and bag-of-words:
    - Usually, each variable gets its own conditional probability distribution \(P(F \mid Y)\)
    - In a bag-of-words model
      - Each position is identically distributed
      - All positions share the same conditional probs \(P(W \mid C)\)
      - Why make this assumption?

Spam Example

<table>
<thead>
<tr>
<th>Word</th>
<th>P(spam)</th>
<th>P(ham)</th>
<th>Tot Spam</th>
<th>Tot Ham</th>
</tr>
</thead>
<tbody>
<tr>
<td>(prior)</td>
<td>0.33333</td>
<td>0.66667</td>
<td>-1.1</td>
<td>-0.4</td>
</tr>
</tbody>
</table>

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

P(features, Y = 2) vs P(features, Y = 3)

\[
P(Y = 2) = 0.1 \quad \text{vs} \quad P(Y = 3) = 0.1
\]

\[
P(on|Y = 2) = 0.8 \quad \text{vs} \quad P(on|Y = 3) = 0.8
\]

\[
P(off|Y = 2) = 0.1 \quad \text{vs} \quad P(off|Y = 3) = 0.7
\]

\[
P(on|Y = 2) = 0.01 \quad \text{vs} \quad P(on|Y = 3) = 0.0
\]

2 wins!!

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
  - 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)
    - 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:
  - 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: Laplace Smoothing

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

\[
P_{LAP}(x) = \frac{c(x) + 1}{\sum_{x}(c(x) + 1)} \quad \text{vs} \quad P_{ML}(X) = \frac{c(x) + 1}{N + |X|}
\]
Estimation: Laplace Smoothing

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

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 \( P(X) \)
    \[
    P_{\text{LIN}}(x|y) = \alpha P(x|y) + (1 - \alpha) P(x)
    \]
  - 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:
  \[
  \begin{align*}
  P(W|\text{ham}) & \quad P(W|\text{spam}) \\
  \quad P(\text{ham}) & \quad P(\text{spam})
  \end{align*}
  \]
  
  - Helvetica: 11.4
  - Seems: 10.8
  - Group: 10.2
  - Area: 8.4
  - Area: 8.3
  - ...

  Verdana: 28.8
  Credit: 28.4
  Order: 27.2
  Order: 26.9
  Money: 26.5

  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: \( k, \alpha \)
- 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

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
  \[
  \text{confidence}(x) = \frac{\max_y P(y|x)}{y}
  \]
  - 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?
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