Naïve Bayes

Example: Spam Filter

- **Input:** an email
- **Output:** spam/ham
- **Setup:**
  - Get a large collection of example emails, each labeled as "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 on being stuck in the corner, but when I plugged it in, 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

- **Classification:** given inputs x, predict labels c
- **Examples:**
  - Spam detection (input: document, classes: spam / ham)
  - OCR (input: images, classes: characters)
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Automatic essay grading (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!
Model-Based Classification

- Model-based approach
  - Build a model (e.g., Bayes’ net) where both the label and features are random variables
  - Instantiate any observed features
  - Query for the distribution of the label conditioned on the features

- Challenges
  - What structure should the BN have?
  - How should we learn its parameters?

Naïve Bayes for Digits

- Naïve Bayes: Assume all features are independent effects of the label

General Naïve Bayes

- A general Naïve Bayes model:

  \[
  P(Y, F_1 \ldots F_n) = P(Y) \prod_{i=1}^{n} P(F_i|Y)
  \]

  \[
  \begin{align*}
  |Y| & \text{ parameters} \\
  |Y| \times |F| \times |Y| & \text{ values} \\
  & \text{n x |F| x |Y| parameters}
  \end{align*}
  \]

  - We only have to specify how each feature depends on the class
  - Total number of parameters is linear in \( n \)
  - Model is very simplistic, but often works anyway

Inference for Naïve Bayes

- Goal: compute posterior distribution over label variable \( Y \)

  \[
  P(Y|F_1 \ldots F_n) = \frac{P(Y) \prod_{i=1}^{n} P(F_i|Y)}{P(F_1 \ldots F_n)}
  \]

  \[
  \begin{align*}
  & \text{Step 1: get joint probability of label and evidence for each label} \\
  & \text{Step 2: sum to get probability of evidence} \\
  & \text{Step 3: normalize by dividing Step 1 by Step 2}
  \end{align*}
  \]

General Naïve Bayes

- What do we need in order to use Naïve Bayes?

  - Inference method (we just saw this part)
    - Start with a bunch of probabilities \( P(Y) \) 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 counts: we’ll look at this soon

Example: Conditional Probabilities
A Spam Filter

- **Naïve Bayes spam filter**
- **Data:**
  - Collection of emails, labeled spam or ham
  - Note: someone has to hand label all this data!
- **Classifiers**
  - Learn on the training set
  - Test it on new emails

**Naïve Bayes for Text**

- **Bag-of-words Naïve Bayes:**
  - **Features:** \( W_i \) is the word at position \( i \)
  - As before: predict label conditioned on feature variables (spam vs. ham)
  - As before: assume features are conditionally independent given label
  - New: each \( W_i \) is identically distributed

**Training and Testing**

- **Data:** labeled instances, e.g. emails labeled 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
    - What is a classifier which does well on training data but not on validation data, and how to avoid it?
Generalization and Overfitting

Example: 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

Parameter Estimation
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:
    \[
    \hat{p}_M(x) = \frac{\text{count}(x)}{\text{total samples}}
    \]
  - This is the estimate that maximizes the likelihood of the data
    \[
    L(x, \theta) = \prod_i p(x_i)
    \]

Smoothing

Maximum Likelihood?

- Relative frequencies are the maximum likelihood estimates
  \[
  \hat{\theta}_{ML} = \arg \max_{\theta} P(X|\theta)
  \]
  \[
  = \arg \max_{\theta} \prod_i p(x_i)
  \]
- Another option is to consider the most likely parameter value given the data
  \[
  \hat{\theta}_{MAP} = \arg \max_{\theta} P(\theta|X)
  \]
  \[
  = \arg \max_{\theta} P(X|\theta)P(\theta)/P(X)
  \]
  \[
  = \arg \max_{\theta} P(X|\theta)P(\theta)
  \]

Unseen Events

Laplace Smoothing

- Laplace’s estimate:
  - Pretend you saw every outcome once more than you actually did
    \[
    \hat{p}_{LAP}(x) = \frac{c(x) + 1}{\sum_i [c(x) + 1]} \quad \hat{p}_{ML}(x) = \frac{c(x) + 1}{N + |X|}
    \]
  - Can derive this estimate with Dirichlet prior (see cs281a)

Laplace Smoothing

- Laplace’s estimate (extended):
  - Pretend you saw every outcome \( k \) extra times
    \[
    \hat{p}_{LAP^k}(x) = \frac{c(x) + k}{N + k|X|}
    \]
  - What’s Laplace with \( k = 0? \)
  - \( k \) is the strength of the prior
  - Laplace for conditionals:
    - Smooth each condition independently:
      \[
      \hat{p}_{LAP^k|p}(x|p) = \frac{c(x, p) + k}{c(p) + k|X|}
      \]
      \[
      \hat{P}_{LAP^k}(X) = \]
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 the empirical P(X) from the data
  - Make sure the estimate of P(X|Y) isn't too different from the empirical P(X)
  
  \[ P_{\text{LIN}}(x|y) = \alpha \hat{P}(x|y) + (1 - \alpha) \hat{P}(x) \]

- What if \( \alpha \) is 0 or 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:

|          | P(W|ham) | P(W|spam) | P(F|ham) |
|----------|---------|-----------|---------|
| helvetica| 11.4    | 28.9      |         |
| seems    | 10.5    | 28.4      |         |
| group    | 10.2    | 27.2      |         |
| shop     | 8.4     | 26.3      |         |
| areas    | 8.3     |           |         |
| ...      |         |           |         |
| credit   | 28.8    | 28.4      |         |
| order    | 27.2    |           |         |
| money    | 26.3    |           |         |
| ...      |         |           |         |

Do these make more sense?

Tuning

- We've got two kinds of unknowns
  - Parameters: the probabilities P(X|Y), P(Y)
  - Hyperparameters: e.g. the amount / type of smoothing to do, \( \alpha \)

- What should we learn where?
  - Learn parameters from training data
  - 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

Tuning on Held-Out Data

Features

- Examples of errors

Dear [Customer],

Omnipage Pro has partnered with ScanSoft to offer you the latest version of OmniPage Pro for just $99.99 - the regular price is $499! We recommend this offer as authorized by ScanSoft, in genuine and valid. You can re-certify your downloads if necessary.

To receive the $30 Amazon.com promotional certificate, click through to http://amazon.com/apparel and see the prominent link for the $30 offer. All details are there.

If you’d rather not receive future offers from us, please click here.

Thank you for your interest in our products and services.

Best regards,

GlobalSCAPE Customer Service
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

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

Summary

- 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

Precision vs. Recall

- Let’s say we want to classify web pages as homepages or not
  - In a test set of 16 pages, there are 3 homepages
  - Our classifier says they are all non-homepages
    - 16 correct / 16 guessed = 1.0 accuracy
    - Need new measures for rare positive events
- Precision: fraction of guessed positives which were actually positive
  - Precision: 2 correct / 5 guessed = 0.4
- Recall: fraction of actual positives which were guessed as positive
  - Say we guess 5 homepages, of which 2 were actually homepages
    - 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 \( r \):
    \[ F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}} \]