RECAP: THE COURSE SO FAR...
First Lecture - Review

• Admin stuff
• Review – Why to scale, how to count and what to count
  • How: O(…)

2
Why to scale: c. 2001 (Banko & Brill, ACL 2001)

Figure 1. Learning Curves for Confusion Set Disambiguation

Figure 2. Representation Size vs. Training Corpus Size

Task: distinguish pairs of easily-confused words (“affect” vs “effect”) in context
# What to count

<table>
<thead>
<tr>
<th>Operation</th>
<th>~ Time</th>
<th>x/100ns</th>
<th>x/10M ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>random access, RAM</td>
<td>100 ns</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>read 1 Mb sequentially - RAM</td>
<td>250,000 ns</td>
<td>2,500</td>
<td></td>
</tr>
<tr>
<td>random access, disk (seek)</td>
<td>10,000,000 ns</td>
<td>100,000</td>
<td>1</td>
</tr>
<tr>
<td>read 1Mb sequentially - net</td>
<td>10,000,000 ns</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>read 1Mb sequentially - disk</td>
<td>30,000,000 ns</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing data over time]
What to count

- Compilers don’t warn Jeff Dean. Jeff Dean warns compilers.
- ....
- Memory access/instructions are qualitatively different from disk access
- Seeks are qualitatively different from sequential reads on disk
- Cache, disk fetches, etc work best when you stream through data sequentially
- Best case for data processing: stream through the data once in sequential order, as it’s found on disk.
First lecture: review

• Admin stuff
• Review – Why to scale, how to count and what to count

• What sort of computations do we want to do in (large-scale) machine learning programs?
  – Probability
PROBABILITY AND SCALABILITY: LEARNING AND COUNTING
Task: distinguish pairs of easily-confused words ("affect" vs "effect") in context
Why More Data Helps: A Demo

• Data:
  – All 5-grams that appear >= 40 times in a corpus of 1M English books
    • approx 80B words
    • 5-grams: 30Gb compressed, 250-300Gb uncompressed
    • Each 5-gram contains frequency distribution over years
  – Wrote code to compute
    • Pr(A,B,C,D,E | C=affect or C=effect)
    • Pr(any subset of A,…,E | any other fixed values of A,…,E with C=affect V effect)
  – Demo:
    • /Users/wcohen/Documents/code/pyhack/bigml
    • eg: python ngram-query.py data/aeffect-train.txt _ _B effect _ _
Why More Data Helps

• Data:
  – All 5-grams that appear >= 40 times in a corpus of 1M English books
    • approx 80B words
    • 5-grams: 30Gb compressed, 250-300Gb uncompressed
    • Each 5-gram contains frequency distribution over *years*
  – Wrote code to compute
    • \( \text{Pr}(A,B,C,D,E | C=\text{affect or } C=\text{effect}) \)
    • \( \text{Pr}(\text{any subset of } A,…,E | \text{any other fixed values of } A,…,E \text{ with } C=\text{affect or } C=\text{effect}) \)

• Observations [from playing with data]:
  – Mostly *effect* not *affect*
  – Most common word before *affect* is *not*
  – After *not effect* most common word is *a*
  – …
The Joint Distribution

Recipe for making a joint distribution of $M$ variables:

1. Make a truth table listing all combinations of values of your variables (if there are $M$ Boolean variables then the table will have $2^M$ rows).
2. For each combination of values, say how probable it is.
3. If you subscribe to the axioms of probability, those numbers must sum to 1.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.30</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.10</td>
</tr>
</tbody>
</table>
Some of the Joint Distribution

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td>the</td>
<td>effect</td>
<td>of</td>
<td>the</td>
<td>0.00036</td>
</tr>
<tr>
<td>is</td>
<td>the</td>
<td>effect</td>
<td>of</td>
<td>a</td>
<td>0.00034</td>
</tr>
<tr>
<td>.</td>
<td>The</td>
<td>effect</td>
<td>of</td>
<td>this</td>
<td>0.00034</td>
</tr>
<tr>
<td>to</td>
<td>this</td>
<td>effect</td>
<td>:</td>
<td>“</td>
<td>0.00034</td>
</tr>
<tr>
<td>be</td>
<td>the</td>
<td>effect</td>
<td>of</td>
<td>the</td>
<td>…</td>
</tr>
<tr>
<td></td>
<td></td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>not</td>
<td>the</td>
<td>effect</td>
<td>of</td>
<td>any</td>
<td>0.00024</td>
</tr>
<tr>
<td></td>
<td></td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>does</td>
<td>not</td>
<td>affect</td>
<td>the</td>
<td>general</td>
<td>0.00020</td>
</tr>
<tr>
<td>does</td>
<td>not</td>
<td>affect</td>
<td>the</td>
<td>question</td>
<td>0.00020</td>
</tr>
<tr>
<td>any</td>
<td>manner</td>
<td>affect</td>
<td>the</td>
<td>principle</td>
<td>0.00018</td>
</tr>
</tbody>
</table>
An experiment: how useful is the brute-force joint classifier?

• Extracted all affect/effect 5-grams from an old Reuters corpus
  – about 20k documents
  – about 723 n-grams, 661 distinct
  – Financial news, not novels or textbooks

• Tried to predict center word with:
  – $\Pr(C \mid A=a, B=b, D=d, E=e)$
  – then $\Pr(C \mid A, B, D)$
  – then $\Pr(C \mid B, D)$
  – then $\Pr(C \mid B)$
  – then $\Pr(C)$
EXAMPLES

– “The cumulative _ of the” $\rightarrow$ effect (1.0)
– “Go into _ on January” $\rightarrow$ effect (1.0)
– “From cumulative _ of accounting” not present in train data
  • Nor is ““From cumulative _ of _”
  • But “_ cumulative _ of _” $\rightarrow$ effect (1.0)
– “Would not _ Finance Minister” not present
  • But “_ not _ _ _” $\rightarrow$ affect (0.9625)
## Performance summary

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Used</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(C</td>
<td>A,B,D,E)$</td>
<td>101</td>
</tr>
<tr>
<td>$P(C</td>
<td>A,B,D)$</td>
<td>157</td>
</tr>
<tr>
<td>$P(C</td>
<td>B,D)$</td>
<td>163</td>
</tr>
<tr>
<td>$P(C</td>
<td>B)$</td>
<td>244</td>
</tr>
<tr>
<td>$P(C)$</td>
<td>58</td>
<td>31</td>
</tr>
</tbody>
</table>

Is this a useful density estimate?
What Have We Learned?

• Counting’s not enough -?
• Counting goes a long way with big data -?

• Big data can sometimes be made small
  – For a specific task, like this one
  – It’s all in the data preparation -?

• Often density estimation is more important than classification

• Counts are a good? density estimator
Density Estimation

• Our Joint Distribution learner is our first example of something called **Density Estimation**

• A Density Estimator learns a mapping from a set of attributes values to a Probability
Density Estimation

• Compare it against the two other major kinds of models:

  Input Attributes → Classifier → Prediction of categorical output or class
  Input Attributes → Density Estimator → Probability
  Input Attributes → Regressor → Prediction of real-valued output
Density Estimation $\rightarrow$ Classification

To classify $x$

1. Use your estimator to compute $\hat{P}(x,y_1), \ldots, \hat{P}(x,y_k)$

2. Return the class $y^*$ with the highest predicted probability

Ideally is correct with $P(x,y^*) = \hat{P}(x,y^*)/(\hat{P}(x,y_1) + \ldots + \hat{P}(x,y_k))$

Binary case: predict POS if $\hat{P}(x)>0.5$
Classification vs Density Estimation

Classification

Density Estimation
Classification vs density estimation
PROBABILITY AND SCALABILITY: NAÏVE BAYES

Second most scalable learning method in the world?
### Performance ...

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Used</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(C</td>
<td>A,B,D,E)</td>
<td>101</td>
</tr>
<tr>
<td>P(C</td>
<td>A,B,D)</td>
<td>157</td>
</tr>
<tr>
<td>P(C</td>
<td>B,D)</td>
<td>163</td>
</tr>
<tr>
<td>P(C</td>
<td>B)</td>
<td>244</td>
</tr>
<tr>
<td>P(C)</td>
<td>58</td>
<td>31</td>
</tr>
</tbody>
</table>

- Is this good performance?
- If we care about recall, what should we do?
Naïve Density Estimation

What’s an alternative to the joint distribution?

The naïve model generalizes strongly:

Assume that each attribute is distributed independently of any of the other attributes.
Using the Naïve Distribution

• Once you have a Naïve Distribution you can easily compute any row of the joint distribution.
• Suppose $A, B, C$ and $D$ are independently distributed. What is $P(A \land \sim B \land C \land \sim D)$?
Using the Naïve Distribution

• Once you have a Naïve Distribution you can easily compute any row of the joint distribution.

• Suppose A, B, C and D are independently distributed. What is \( P(A \land \neg B \land C \land \neg D) \)?

\[ P(A) \cdot P(\neg B) \cdot P(C) \cdot P(\neg D) \]
Naïve Distribution General Case

• Suppose $X_1, X_2, \ldots, X_d$ are independently distributed.

\[
\Pr(X_1 = x_1, \ldots, X_d = x_d) = \Pr(X_1 = x_1) \cdot \ldots \cdot \Pr(X_d = x_d)
\]

• So if we have a Naïve Distribution we can construct any row of the implied Joint Distribution on demand.

• How do we learn this?
Learning a Naïve Density Estimator

\[ P(X_i = x_i) = \frac{\# \text{records with } X_i = x_i}{\# \text{records}} \quad \text{MLE} \]

\[ P(X_i = x_i) = \frac{\# \text{records with } X_i = x_i + mq}{\# \text{records} + m} \quad \text{Dirichlet (MAP)} \]

Another trivial learning algorithm!
Is this an interesting learning algorithm? No

- For n-grams, what is $\hat{P}(C=\text{effect} \mid A=\text{will})$?
  - In joint: $\hat{P}(C=\text{effect} \mid A=\text{will}) = 0.38$
  - In naïve: $\hat{P}(C=\text{effect} \mid A=\text{will}) = \hat{P}(C=\text{effect}) = \frac{\# [C=\text{effect}]}{\# \text{total Ngrams}} = 0.94 (!)$

- What is $\hat{P}(C=\text{effect} \mid B=\text{no})$?
  - In joint: $\hat{P}(C=\text{effect} \mid B=\text{no}) = 0.999$
  - In naïve: $\hat{P}(C=\text{effect} \mid B=\text{no}) = \hat{P}(C=\text{effect}) = 0.94$
Can we make this interesting? Yes!

• Key ideas:
  – Pick the class variable $Y$
  – Instead of estimating $P(X_1,\ldots,X_n,Y) = P(X_1)^*\ldots*P(X_n)^*Y$,
estimate $P(X_1,\ldots,X_n | Y) = P(X_1 | Y)^*\ldots*P(X_n | Y)$
  – Or, assume $P(X_i | Y) = Pr(X_i | X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n,Y)$
  – Or, that $X_i$ is conditionally independent of every $X_j$, $j\neq i$, given $Y$.

  – How to estimate?

  MLE or MAP
The Naïve Bayes classifier – v1

• Dataset: each example has
  – A unique id $id$
    • Why? For debugging the feature extractor
  – $d$ attributes $X_1,\ldots,X_d$
    • Each $X_i$ takes a discrete value in $\text{dom}(X_i)$
  – One class label $Y$ in $\text{dom}(Y)$

• You have a train dataset and a test dataset

• Assume:
  – the dataset doesn’t fit in memory
  – the model does

stream through it
The Naïve Bayes classifier – v0

- You have a train dataset and a test dataset
- Initialize an “event counter” (hashtable) C
- For each example id, y, x₁,.....,xₙ in train:
  - C("Y=ANY") ++; C("Y=y") ++
  - For j in 1..d:
    • C("Y=y ^ X_j=x_j") ++
- For each example id, y, x₁,.....,xₙ in test:
  - For each y’ in dom(Y):
    • Compute Pr(y’,x₁,.....,xₙ) = \( \prod_{j=1}^{d} \Pr(X_j = x_j | Y = y') \) \( \Pr(Y = y') \)

  = \( \prod_{j=1}^{d} \frac{\Pr(X_j = x_j, Y = y')}{\Pr(Y = y')} \) \( \Pr(Y = y') \)

  - Return the best y’
The Naïve Bayes classifier – v0

• You have a *train* dataset and a *test* dataset
• Initialize an “event counter” (hashtable) C
• For each example id, y, x₁,…..,xₙ in *train*:
  – C(“Y=ANY”) ++;  C(“Y=y”) ++
  – For j in 1..d:
    • C(“Y=y ^ X_j=x_j”) ++
• For each example id, y, x₁,…..,xₙ in *test*:
  – For each y’ in dom(Y):
    • Compute Pr(y’,x₁,…..,xₙ) = \[\prod_{j=1}^{d} \Pr(X_j = x_j \mid Y = y') \Pr(Y = y')\]

\[=\left(\prod_{j=1}^{d} \frac{C(X_j = x_j \land Y = y')}{C(Y = y')} \right) \frac{C(Y = y')}{C(Y = ANY)}\]

  – Return the best y’
The Naïve Bayes classifier – v0

- You have a *train* dataset and a *test* dataset
- Initialize an “event counter” (hashtable) C
- For each example id, y, x₁,...,x_d in train:
  - C(“Y=ANY”) ++;  C(“Y=y”) ++
  - For j in 1..d:
    - C(“Y=y ^ X_j=x_j”) ++
- For each example id, y, x₁,...,x_d in test:
  - For each y’ in dom(Y):
    - Compute $\Pr(y', x_1, ..., x_d) = \left(\prod_{j=1}^{d} \Pr(X_j = x_j | Y = y')\right) \Pr(Y = y')$
  - $= \left(\prod_{j=1}^{d} \frac{C(X_j = x_j \land Y = y')}{\frac{C(Y = y')}{C(Y = ANY)}}\right) \frac{C(Y = y')}{C(Y = ANY)}$
    - This may overfit, so …
  - Return the best y’
The Naïve Bayes classifier – v1

• You have a \textit{train} dataset and a \textit{test} dataset
• Initialize an “event counter” (hashtable) \( C \)
• For each example \( id, y, x_1, \ldots, x_d \) in \textit{train}:
  – \( C(“Y=\text{ANY”}) ++; \) \( C(“Y=y’”) ++ \)
  – For \( j \) in 1..\( d \):
    • \( C(“Y=y \land X_j=x_j”) ++ \)
• For each example \( id, y, x_1, \ldots, x_d \) in \textit{test}:
  – For each \( y’ \) in \( \text{dom}(Y) \):
    • Compute \( \Pr(y’, x_1, \ldots, x_d) = \left( \prod_{j=1}^{d} \Pr(X_j = x_j \mid Y = y’) \right) \Pr(Y = y’) \)

\[
= \left( \prod_{j=1}^{d} \frac{C(X_j = x_j \land Y = y’) + mq_x}{C(Y = y’) + m} \right) \frac{C(Y = y’) + mq_y}{C(Y = \text{ANY}) + m}
\]

where:
\( q_j = 1/|\text{dom}(X_j)| \)
\( q_y = 1/|\text{dom}(Y)| \)
\( mq_x = 1 \)

– Return the best \( y’ \)

This may underflow, so …
The Naïve Bayes classifier – v1

- You have a *train* dataset and a *test* dataset
- Initialize an “event counter” (hashtable) C
- For each example id, y, x₁,...,xₖ in *train*:
  - C(“Y=ANY”) ++;  C(“Y=y”) ++
  - For j in 1..d:
    - C("Y=y ^ X_j=x_j") ++
- For each example id, y, x₁,...,xₖ in *test*:
  - For each y' in dom(Y):
    - Compute log Pr(y',x₁,...,xₖ) =
      \[
      \left( \sum_j \log \frac{C(X_j = x_j \land Y = y') + mq_j}{C(Y = y') + m} \right) + \log \frac{C(Y = y') + mq_j}{C(Y = ANY) + m}
      \]
      - Return the best y'

where:
- \( q_j = 1/| \text{dom}(X_j) | \)
- \( q_y = 1/| \text{dom}(Y) | \)
- \( mq_x=1 \)
The Naïve Bayes classifier – v2

• For text documents, what features do you use?
• One common choice:
  – $X_1 = \text{first word in the document}$
  – $X_2 = \text{second word in the document}$
  – $X_3 = \text{third …}$
  – $X_4 = \text{…}$
  – …

• But: $\Pr(X_{13} = \text{hockey} \mid Y = \text{sports})$ is probably not that different from $\Pr(X_{11} = \text{hockey} \mid Y = \text{sports})$… so instead of treating them as different variables, treat them as different copies of the same variable
The Naïve Bayes classifier – v1

- You have a train dataset and a test dataset
- Initialize an “event counter” (hashtable) C
- For each example id, y, x₁,…..,xₙ in train:
  - C(“Y=ANY”) ++; C(“Y=y”) ++
  - For j in 1..d:
    - C(“Y=y ^ X_j=x_j”) ++
- For each example id, y, x₁,…..,xₙ in test:
  - For each y’ in dom(Y):
    - Compute Pr(y’,x₁,…..,xₙ) = \( \prod_{j=1}^{d} \Pr(X_j = x_j \mid Y = y') \Pr(Y = y') \)
      \[= \left( \prod_{j=1}^{d} \frac{\Pr(X_j = x_j, Y = y')}{\Pr(Y = y')} \right) \Pr(Y = y') \]
  - Return the best y’
The Naïve Bayes classifier – v2

• You have a \textit{train} dataset and a \textit{test} dataset
• Initialize an “event counter” (hashtable) C
• For each example $id, y, x_1, \ldots, x_d$ in \textit{train}:
  – $C(\"Y=\text{ANY}\")$ $\text{++}$; \hspace{0.1cm} $C(\"Y=y\")$ $\text{++}$
  – For $j$ in $1..d$:
    • $C(\Y=y \land X_j=x_j)$ $\text{++}$
• For each example $id, y, x_1, \ldots, x_d$ in \textit{test}:
  – For each $y'$ in $\text{dom}(Y)$:
    • Compute $\Pr(y', x_1, \ldots, x_d) =$ \(\prod_{j=1}^{d} \Pr(X_j = x_j \mid Y = y') \Pr(Y = y')\)
      \[= \left(\prod_{j=1}^{d} \frac{\Pr(X_j = x_j, Y = y')}{\Pr(Y = y')}\right) \Pr(Y = y')\]
  – Return the best $y'$
The Naïve Bayes classifier – v2

• You have a train dataset and a test dataset
• Initialize an “event counter” (hashtable) C
• For each example id, y, x_1, ..., x_d in train:
  – C(“Y=ANY”) ++;  C(“Y=y”) ++
  – For j in 1..d:
    • C(“Y=y ^ X=x_j”) ++
• For each example id, y, x_1, ..., x_d in test:
  – For each y’ in dom(Y):
    • Compute Pr(y’,x_1,....,x_d) = \left( \prod_{j=1}^{d} \Pr(X = x_j \mid Y = y') \right) \Pr(Y = y')

  = \left( \prod_{j=1}^{d} \frac{\Pr(X = x_j,Y = y')}{\Pr(Y = y')} \right) \Pr(Y = y')

  – Return the best y’
The Naïve Bayes classifier – v2

• You have a train dataset and a test dataset
• Initialize an “event counter” (hashtable) C
• For each example id, y, x₁,…..,xₙ in train:
  – C(“Y=ANY”) ++;   C(“Y=y”) ++
  – For j in 1..d:
    • C(“Y=y ^ X=x_j”) ++
• For each example id, y, x₁,…..,xₙ in test:
  – For each y’ in dom(Y):
    • Compute log Pr(y’,x₁,…..,xₙ) =

\[
= \left( \sum_j \log \frac{C(X = x_j \land Y = y') + mq_x}{C(X = ANY \land Y = y') + m} \right) + \log \frac{C(Y = y') + mq_y}{C(Y = ANY) + m}
\]

– Return the best y’
The Naïve Bayes classifier – v2

• You have a train dataset and a test dataset

• To classify documents, these might be:
  – [http://wcohen.com](http://wcohen.com) academic, Faculty Home William W. Cohen Research Professor Machine Learning Department Carnegie Mellon University Member of the Language Technology Institute the joint CMU-Pitt Program in Computational Biology the Lane Center for Computational Biology and the Center for Bioimage Informatics Director of the Undergraduate Minor in Machine Learning Bio Teaching Projects Publications recent all Software Datasets Talks Students Colleagues Blog Contact Info Other Stuff …
  – [http://google.com](http://google.com) commercial Search Images Videos ….
  – ...

• How about for n-grams?
The Naïve Bayes classifier – v2

• You have a train dataset and a test dataset
• To do C-S spelling correction these might be
  – ng1223 effect a_the b_main d_of e_the
  – ng1224 affect a_shows b_not d_mice e_in
  – ....
• I.e., encode event $X_i=w$ with another event $X=i_w$
• Question: are there any differences in behavior from using A,B,C,D?
Complexity of Naïve Bayes

- You have a *train* dataset and a *test* dataset
- Initialize an “event counter” (hashtable) C
- For each example *id, y, x₁,...,xₙ* in *train*:
  - C("Y=ANY") ++; C("Y=y") ++
  - For *j* in 1..*d*:
    - C("Y=y ^ X=x_j") ++
- For each example *id, y, x₁,...,xₙ* in *test*:
  - For each *y'* in *dom(Y)*:
    - Compute \( \log \Pr(y',x₁,...,xₙ) = \)
      \[
      \sum_j \log \frac{C(X = x_j \land Y = y') + mq_x}{C(X = ANY \land Y = y') + m} + \log \frac{C(Y = y') + mq_y}{C(Y = ANY) + m}
      \]
  - Return the best *y'*

Assume hashtable holding all counts fits in memory

Sequential reads

**Complexity:** \( O(n) \), \( n=\text{size of } \text{train} \)

where:
- \( q_j = 1/|V| \)
- \( q_y = 1/|\text{dom}(Y)| \)
- \( mq_x = 1 \)

Sequential reads

**Complexity:** \( O(|\text{dom}(Y)|^{*n'}), \ n'=\text{size of } \text{test} \)
Complexity of Naïve Bayes

• You have a train dataset and a test dataset
• Process:
  – Count events in the train dataset
    • $O(n)$, where $n$ is total size of train
  – Write the counts to disk
    • $O(\min(|\text{dom}(X)| \times |\text{dom}(Y)|, n))$
    • $O(|V|)$, if $V$ is vocabulary and $\text{dom}(Y)$ is small
  – Classify the test dataset
    • $O(|V| + n')$
  – Worst-case memory usage:
    • $O(\min(|\text{dom}(X)| \times |\text{dom}(Y)|, n))$
Naïve Bayes v2

• This is one example of a *streaming classifier*
  – Each example is only read only once
  – You can create a classifier and perform classifications at any point
  – Memory is minimal (<< O(n))
    • Ideally it would be constant
    • Traditionally less than O(sqrt(N))
  – Order doesn’t matter
    • Nice because we may not control the order of examples in real life
      • This is a hard one to get a learning system to have!
• There are few competitive learning methods that as stream-y as naïve Bayes…
Rocchio’s Algorithm
Motivation

• Naïve Bayes is unusual as a learner:
  – Only one pass through data
  – Order doesn’t matter
Rocchio’s algorithm

Rocchio’s algorithm

\[ DF(w) = \# \text{different docs } w \text{ occurs in} \]

\[ TF(w, d) = \# \text{different times } w \text{ occurs in doc } d \]

\[ IDF(w) = \frac{|D|}{DF(w)} \]

\[ u(w, d) = \log(TF(w, d) + 1) \cdot \log(IDF(w)) \]

\[ u(d) = \langle u(w_1, d), \ldots, u(w_{|V|}, d) \rangle \]

\[ u(y) = \alpha \frac{1}{|C_y|} \sum_{d \in C_y} \frac{u(d)}{||u(d)||_2} - \beta \frac{1}{|D - C_y|} \sum_{d' \in D - C_y} \frac{u(d')}{||u(d')||_2} \]

\[ f(d) = \arg \max_y \frac{u(d)}{||u(d)||_2} \cdot \frac{u(y)}{||u(y)||_2} \]

\[ ||u||_2 = \sqrt{\sum_{i} u_i^2} \]

Many variants of these formulae

…as long as \( u(w, d) = 0 \) for words not in \( d \! \)!

Store only non-zeros in \( u(d) \), so size is \( O(|d|) \)

But size of \( u(y) \) is \( O(|n_V|) \)
Charles Bailey was indicted for feloniously stealing on the 29th of December two dressed deer skins valued 20 s the property of Samuel Savage and Richard Savage. Richard Savage is a leather seller at 63 Chiswell Street. My partner's name is Samuel Savage. A few days previous to the 29th of December I looked out seventy skins for an order. These skins being of a bad colour I directed them to be brimstoned to make them of equal colour pale. On the 29th in the afternoon I saw them all smooth on a horse. A few hours afterwards they appeared very much tumbled and one was thrown into the yard and dirtied. I caused them to be brought in the warehouse and counted. There was two gone! Our foreman went to Worship Street and brought Armstrong and Vickery. They searched and found this skin in the prisoner's breeches and the other skin was found in the workshop. Carter, I am foreman to Samuel and Richard Savage. The seventy skins I was with Mr. Savage looking them out. I took them out of the stove and counted them on the horse. On Friday I counted them three times over there were no more than sixty-eight instead of seventy. I went to Worship Street. I brought Mr. Armstrong and Vickery with me. They waited till the men left work and when they came down they were searched and on the prisoner one skin was found.

How do you determine what are the “important” words in a document?
Charles Bailey was indicted for feloniously stealing on the 29th of December two dressed Deer skins value 20 s. the property of Samuel Savage and Richard Savage. Richard Savage I am a Leather seller 63 Chiswell Street. My partner's name is Samuel Savage. A few days previous to the 29th of December I looked out seventy skins for an order these skins being of a bad colour I directed them to be brimstoned to make them of equal colour pale. On the 29th in the afternoon I saw them all smooth on a horse a few hours afterwards they appeared very much tumbled and one was thrown into the yard and dirtied. I caused them to be brought in the warehouse and counted. There was two gone. Our foreman went to Worship Street and brought Armstrong and Vickrey. They searched and found this skin in the prison a window and the skin was sent in the morning.
Charles Bailey was indicted for feloniously stealing on the 29th of December two dressed deer skins value 20 s the property of Samuel Savage and Richard Savage Richard Savage a few days previous to the 29th of December I looked out seventy skins for an order these skins being of a bad colour I directed them to be brimstoned to make them of equal colour pale on the 29th in the afternoon I saw them all smooth on a horse a few hours afterwards they appeared very much tumbled and one was thrown into the yard and dirtied I caused them to be brought in the warehouse and counted there was two gone our foreman went worshiped Ray Armstrong and Vickroy...
Charles Bailey was indicted for feloniously stealing on the 29th of December two dressed deer skins value 20 s the property of Samuel Savage and Richard Savage Richard Savage I am a leather seller 63 Chiswell Street my partner's name is Samuel Savage a few days previous to the 29th of December I looked out seventy skins for an order these skins being of a bad colour I directed them to be brimstoned to make them of equal colour pale on the 29th in the afternoon I saw them all smooth on a horse a few hours afterwards they appeared very much tumbled and one was thrown into the yard and dirtied I caused them to be brought in the warehouse and counted there was two gone our foreman went to worship street and brought Armstrong and Vickrey they searched and found this skin in the prisoner's breeches and the other skin was found in the workshop Carter I am foreman to Samuel and Richard Savage the seventy skins I was with Mr Savage looking them out I took them out of the stove and counted them on the horse and on Friday I counted them

TF-IDF weighting: frequently-appearing rare terms, which are often names and places and other things important to the document
Rocchio results...

Joachim ’98, “A Probabilistic Analysis of the Rocchio Algorithm…”

<table>
<thead>
<tr>
<th></th>
<th>PrTFIDF</th>
<th>BAYES</th>
<th>TFIDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsgroups</td>
<td>91.8</td>
<td>89.6</td>
<td>86.3</td>
</tr>
<tr>
<td>“acq”</td>
<td>88.9</td>
<td>88.5</td>
<td>84.5</td>
</tr>
<tr>
<td>“wheat”</td>
<td>93.9</td>
<td>94.8</td>
<td>90.9</td>
</tr>
<tr>
<td>“crude”</td>
<td>90.2</td>
<td>95.5</td>
<td>85.4</td>
</tr>
<tr>
<td>“earn”</td>
<td>90.5</td>
<td>90.9</td>
<td>90.6</td>
</tr>
<tr>
<td>“cbond”</td>
<td>91.9</td>
<td>90.9</td>
<td>87.7</td>
</tr>
</tbody>
</table>

Table 2: Maximum accuracy in percentages.

Variant TF and IDF formulas

Rocchio’s method (w/ linear TF)
<table>
<thead>
<tr>
<th>System</th>
<th>Type</th>
<th>Results reported by</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORD</td>
<td>(non-learning)</td>
<td>Yang [1999]</td>
</tr>
<tr>
<td>ProbBayes Blm</td>
<td>probabilistic</td>
<td>[Dumais et al. 1998]</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>[Joachims 1998]</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>[Lam et al. 1997]</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>[Lewis 1992a]</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>[Li and Yamanishi 1999]</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>[Li and Yamanishi 1999]</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>[Yang and Liu 1999]</td>
</tr>
<tr>
<td>Nb</td>
<td>probabilistic</td>
<td>.443 (MF$_1$)</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.650</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.752</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.720</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.310</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.290</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.815</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.795</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.773</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.747</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.794</td>
</tr>
<tr>
<td></td>
<td>ProbBayes Blm</td>
<td>.884</td>
</tr>
<tr>
<td>C4.5 Ind</td>
<td>decision trees</td>
<td>[Dumais et al. 1998]</td>
</tr>
<tr>
<td></td>
<td>decision trees</td>
<td>[Joachims 1998]</td>
</tr>
<tr>
<td></td>
<td>decision trees</td>
<td>[Lewis and Ringuette 1994]</td>
</tr>
<tr>
<td></td>
<td>decision trees</td>
<td>[Yang 1999]</td>
</tr>
<tr>
<td></td>
<td>decision trees</td>
<td>[Yang and Liu 1999]</td>
</tr>
<tr>
<td></td>
<td>decision rules</td>
<td>.670</td>
</tr>
<tr>
<td></td>
<td>decision rules</td>
<td>.794</td>
</tr>
<tr>
<td></td>
<td>decision rules</td>
<td>.310</td>
</tr>
<tr>
<td></td>
<td>decision rules</td>
<td>.884</td>
</tr>
<tr>
<td></td>
<td>decision rules</td>
<td>.794</td>
</tr>
<tr>
<td></td>
<td>decision rules</td>
<td>.794</td>
</tr>
<tr>
<td>SLEEPING EXPERTS</td>
<td>decision rules</td>
<td>[Cohen and Singer 1999]</td>
</tr>
<tr>
<td>RIFFER</td>
<td>decision rules</td>
<td>[Cohen and Singer 1999]</td>
</tr>
<tr>
<td>DL-Esc</td>
<td>decision rules</td>
<td>[Cohen and Singer 1999]</td>
</tr>
<tr>
<td>CHARADE</td>
<td>decision rules</td>
<td>[Cohen and Singer 1999]</td>
</tr>
<tr>
<td>CHARADE</td>
<td>decision rules</td>
<td>[Li and Yamanishi 1999]</td>
</tr>
<tr>
<td>CHARADE</td>
<td>decision rules</td>
<td>[Moulinier and Ganaucia 1996]</td>
</tr>
<tr>
<td>CHARADE</td>
<td>decision rules</td>
<td>[Moulinier et al. 1996]</td>
</tr>
<tr>
<td>BalancedWinnow WINNOW-Hoeff</td>
<td>regression</td>
<td>[Yang 1999]</td>
</tr>
<tr>
<td></td>
<td>regression</td>
<td>[Yang and Liu 1999]</td>
</tr>
<tr>
<td></td>
<td>regression</td>
<td>.855</td>
</tr>
<tr>
<td></td>
<td>regression</td>
<td>.810</td>
</tr>
<tr>
<td></td>
<td>regression</td>
<td>.849</td>
</tr>
<tr>
<td></td>
<td>on-line linear</td>
<td>[Dagan et al. 1997]</td>
</tr>
<tr>
<td></td>
<td>on-line linear</td>
<td>[Lam and Ho 1998]</td>
</tr>
<tr>
<td></td>
<td>on-line linear</td>
<td>[Yang 1999]</td>
</tr>
<tr>
<td></td>
<td>on-line linear</td>
<td>[Yang and Liu 1999]</td>
</tr>
<tr>
<td></td>
<td>on-line linear</td>
<td>.747 (M)</td>
</tr>
<tr>
<td></td>
<td>on-line linear</td>
<td>.833 (M)</td>
</tr>
<tr>
<td></td>
<td>on-line linear</td>
<td>.822</td>
</tr>
<tr>
<td>ROCCHIO</td>
<td>batch linear</td>
<td>[Cohen and Singer 1999]</td>
</tr>
<tr>
<td>FINDSIM</td>
<td>batch linear</td>
<td>[Dumais et al. 1998]</td>
</tr>
<tr>
<td>ROCCHIO</td>
<td>batch linear</td>
<td>[Joachims 1998]</td>
</tr>
<tr>
<td>ROCCHIO</td>
<td>batch linear</td>
<td>[Lam and Ho 1998]</td>
</tr>
<tr>
<td>ROCCHIO</td>
<td>batch linear</td>
<td>[Li and Yamanishi 1999]</td>
</tr>
<tr>
<td>CLASSI</td>
<td>neural network</td>
<td>[Ng et al. 1997]</td>
</tr>
<tr>
<td>NNET</td>
<td>neural network</td>
<td>[Yang and Liu 1999]</td>
</tr>
<tr>
<td></td>
<td>neural network</td>
<td>[Wiener et al. 1995]</td>
</tr>
<tr>
<td></td>
<td>neural network</td>
<td>.802</td>
</tr>
<tr>
<td></td>
<td>neural network</td>
<td>.838</td>
</tr>
<tr>
<td></td>
<td>neural network</td>
<td>.820</td>
</tr>
<tr>
<td>GIB-W</td>
<td>example-based</td>
<td>[Lam and Ho 1998]</td>
</tr>
<tr>
<td>k-NN</td>
<td>example-based</td>
<td>[Joachims 1998]</td>
</tr>
<tr>
<td>k-NN</td>
<td>example-based</td>
<td>[Lam and Ho 1998]</td>
</tr>
<tr>
<td>k-NN</td>
<td>example-based</td>
<td>[Yang 1999]</td>
</tr>
<tr>
<td>k-NN</td>
<td>example-based</td>
<td>[Yang and Liu 1999]</td>
</tr>
<tr>
<td></td>
<td>example-based</td>
<td>.690</td>
</tr>
<tr>
<td></td>
<td>example-based</td>
<td>.852</td>
</tr>
<tr>
<td></td>
<td>example-based</td>
<td>.820</td>
</tr>
<tr>
<td></td>
<td>example-based</td>
<td>.856</td>
</tr>
<tr>
<td></td>
<td>example-based</td>
<td>.860</td>
</tr>
<tr>
<td></td>
<td>example-based</td>
<td>.860</td>
</tr>
<tr>
<td></td>
<td>example-based</td>
<td>.820</td>
</tr>
<tr>
<td>SVM LIGHT</td>
<td>SVM</td>
<td>[Dumais et al. 1998]</td>
</tr>
<tr>
<td>SVM LIGHT</td>
<td>SVM</td>
<td>[Joachims 1998]</td>
</tr>
<tr>
<td>SVM LIGHT</td>
<td>SVM</td>
<td>[Li Yamanishi 1999]</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>[Yang and Liu 1999]</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>.870</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>.864</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>.841</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>.859</td>
</tr>
<tr>
<td>AdaBoost.MH</td>
<td>committee</td>
<td>[Schapire and Singer 2000]</td>
</tr>
<tr>
<td></td>
<td>committee</td>
<td>[Weiss et al. 1999]</td>
</tr>
<tr>
<td></td>
<td>committee</td>
<td>.860</td>
</tr>
<tr>
<td></td>
<td>committee</td>
<td>.878</td>
</tr>
<tr>
<td>Bayesian net</td>
<td>[Dumais et al. 1998]</td>
<td>.542 (MF$_1$)</td>
</tr>
<tr>
<td>Bayesian net</td>
<td>[Lam et al. 1997]</td>
<td>.800</td>
</tr>
<tr>
<td>Bayesian net</td>
<td>[Lam et al. 1997]</td>
<td>.850</td>
</tr>
</tbody>
</table>
Rocchio results…

Schapire, Singer, Singhal, “Boosting and Rocchio Applied to Text Filtering”, SIGIR 98

Reuters 21578 – all classes (not just the frequent ones)
A hidden agenda

• Part of machine learning is good grasp of theory
• Part of ML is a good grasp of what hacks tend to work
• These are not always the same
  – Especially in big-data situations

• Catalog of useful tricks so far
  – Brute-force estimation of a joint distribution
  – Naive Bayes
  – Stream-and-sort, request-and-answer patterns
  – BLRT and KL-divergence (and when to use them)
  – TF-IDF weighting – especially IDF
    • it’s often useful even when we don’t understand why
One more Rocchio observation

Rennie et al, ICML 2003, “Tackling the Poor Assumptions of Naïve Bayes Text Classifiers”

<table>
<thead>
<tr>
<th></th>
<th>MNB</th>
<th>TWCNB</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industry Sector</td>
<td>0.582</td>
<td>0.923</td>
<td>0.934</td>
</tr>
<tr>
<td>20 Newsgroups</td>
<td>0.848</td>
<td>0.861</td>
<td>0.862</td>
</tr>
<tr>
<td>Reuters (micro)</td>
<td>0.739</td>
<td>0.844</td>
<td>0.887</td>
</tr>
<tr>
<td>Reuters (macro)</td>
<td>0.270</td>
<td>0.647</td>
<td>0.694</td>
</tr>
</tbody>
</table>

NB + cascade of hacks
One more Rocchio observation

Rennie et al, ICML 2003, “Tackling the Poor Assumptions of Naïve Bayes Text Classifiers”

- TWCNB($\vec{d}, \vec{y}$)

1. $d_{ij} = \log(d_{ij} + 1)$ (TF transform § 4.1)
2. $d_{ij} = d_{ij} \log \frac{\sum_k 1}{\sum_k \delta_{ik}}$ (IDF transform § 4.2)
3. $d_{ij} = \frac{d_{ij}}{\sqrt{\sum_k (d_{kj})^2}}$ (length norm. § 4.3)
4. $\hat{\theta}_{ci} = \frac{\sum_{j:y_j \neq c} d_{ij} + \alpha_i}{\sum_{j:y_j \neq c} \sum_k d_{kj} + \alpha}$ (complement § 3.1)
5. $w_{ci} = \log \hat{\theta}_{ci}$
6. $w_{ci} = \frac{w_{ci}}{\sum_i w_{ci}}$ (weight normalization § 3.2)
7. Let $t = (t_1, \ldots, t_n)$ be a test document; let $t_i$ be the count of word $i$.
8. Label the document according to

$$l(t) = \arg \min_c \sum_i t_i w_{ci}$$

“In tests, we found the length normalization to be most useful, followed by the log transform...these transforms were also applied to the input of SVM”.