Advanced Multimedia

Text Classification
Tamara Berg
The Dream

- It’d be great if machines could
  - Process our email (usefully)
  - Translate languages accurately
  - Help us manage, summarize, and aggregate information
  - Use speech as a UI (when needed)
  - Talk to us / listen to us

- But they can’t:
  - Language is complex, ambiguous, flexible, and subtle
  - Good solutions need linguistics and machine learning knowledge

- So:
What is NLP?

- **Fundamental goal:** *deep understand of broad language*
  - Not just string processing or keyword matching!

- **End systems that we want to build:**
  - Ambitious: speech recognition, machine translation, information extraction, dialog interfaces, question answering…
  - Modest: spelling correction, text categorization…

Slide from Dan Klein
What is NLP?

- Fundamental goal: *deep* understand of *broad* language
  - Not just string processing or keyword matching!

- End systems that we want to build:
  - Ambitious: speech recognition, machine translation, information extraction, dialog interfaces, question answering…
  - Modest: spelling correction, *text categorization*…

Slide from Dan Klein
What does categorization/classification mean?
Example: Spam Filter

- Input: email
- Output: spam/ham

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidencial 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: 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

---

**Sample Emails:**

- **Email 1:**
  - 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. ...

- **Email 2:**
  - TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.
  - 99 MILLION EMAIL ADDRESSES FOR ONLY $99

- **Email 3:**
  - 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.

---

Slide from Dan Klein
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

---

- **Examples:**

  **Example A:**
  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. 

  **Example B:**
  TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.
  99 MILLION EMAIL ADDRESSES FOR ONLY $99

  **Example C:**
  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: 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.
Topic classification

classes:
- UK
  - congestion
  - London
- China
  - Olympics
  - Beijing
- poultry
  - feed
  - chicken
- coffee
  - roasting
  - beans
- elections
  - recount
  - votes
- sports
  - diamond
  - baseball

training set:
- Parliament
  - Big Ben
- tourism
  - Great Wall
- Mao
  - communist
- bird flu
  - turkey
- Kenya
  - harvest

subject areas:
- elections
  - recount
  - votes
- sports
  - diamond
  - baseball

test set:
- first
  - private
- Chinese
  - airline

$\gamma(d') = China$
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
  - ...

http://yann.lecun.com/exdb/mnist/index.html
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!
Applications of text classification in IR

- Language identification (classes: English vs. French etc.)
- The automatic detection of spam pages (spam vs. nonspam, example: googel.org)
- The automatic detection of sexually explicit content (sexually explicit vs. not)
- Sentiment detection: is a movie or product review positive or negative (positive vs. negative)
- Topic-specific or vertical search – restrict search to a “vertical” like “related to health” (relevant to vertical vs. not)
- Machine-learned ranking function in ad hoc retrieval (relevant vs. nonrelevant)
- Semantic Web: Automatically add semantic tags for non-tagged text (e.g., for each paragraph: relevant to a vertical like health or not)
Classification methods: 1. Manual

- Manual classification was used by Yahoo in the beginning of the web. Also: ODP, PubMed
- Very accurate if job is done by experts
- Consistent when the problem size and team is small
- Manual classification is difficult and expensive to scale.
- We need automatic methods for classification.
Classification methods: 2. Rule-based

- There are “IDE” type development environments for writing very complex rules efficiently. (e.g., Verity)
- Often: Boolean combinations (as in Google Alerts)
- Accuracy is very high if a rule has been carefully refined over time by a subject expert.
- Building and maintaining rule-based classification systems is expensive.
Classification methods: 3. Statistical/Probabilistic

- Machine Learning - how to select a model on the basis of data / experience
  - Learning parameters (e.g. probabilities)
  - Learning structure (e.g. dependencies)
  - Learning hidden concepts (e.g. clustering)
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
Classifiers

• Today we’ll talk about 2 simple kinds of classifiers
  – Nearest Neighbor Classifier
  – Naïve Bayes Classifier
Classifiers

• Today we’ll talk about 2 simple kinds of classifiers
  – Nearest Neighbor Classifier
  – Naïve Bayes Classifier
The paper bag is a remarkable contrivance. It serves us constantly and inconspicuously. It folds flat, yet opens into a structure that can stand open upon the table while we eat our sandwiches from it and chat with friends.

If we take the bag apart, we find it's made from a single paper cylinder. One end of the cylinder has been folded into a complex 3-dimensional pattern and finished off with a bit of paste. It would be, and once was, costly to make, because each fragile cylinder had to be folded manually into that hardy sack.
The paper bag is a remarkable contrivance. It serves us constantly and inconspicuously. It folds flat, yet opens into a structure that can stand open upon the table while we eat our sandwiches from it and chat with friends.

If we take the bag apart, we find it's made from a single paper cylinder. One end of the cylinder has been folded into a complex 3-dimensional pattern and finished off with a bit of paste. It would be, and once was, costly to make, because each fragile cylinder had to be folded manually into that hardy sack.
Example

• Doc1 = “the quick brown fox jumped”
• Doc2 = “brown quick jumped fox the”
Example

- Doc1 = “the quick brown fox jumped”
- Doc2 = “brown quick jumped fox the”

Would a bag of words model represent these two documents differently?
Document Vectors

• Documents are represented as “bags of words”
• Represented as vectors when used computationally
  • Each vector holds a place for every term in the collection
  • Therefore, most vectors are sparse
Document Vectors

• Documents are represented as “bags of words”
• Represented as vectors when used computationally
  • Each vector holds a place for every term in the collection
  • Therefore, most vectors are sparse

Lexicon – the vocabulary set that you consider to be valid words in your documents.
  Usually stemmed (e.g. running->run)
Document Vectors:
One location for each word.

<table>
<thead>
<tr>
<th></th>
<th>nova</th>
<th>galaxy</th>
<th>heat</th>
<th>h’wood</th>
<th>film</th>
<th>role</th>
<th>diet</th>
<th>fur</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A

Slide from Mitch Marcus
Document Vectors:
One location for each word.

<table>
<thead>
<tr>
<th></th>
<th>nova</th>
<th>galaxy</th>
<th>heat</th>
<th>h’wood</th>
<th>film</th>
<th>role</th>
<th>diet</th>
<th>fur</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

“Nova” occurs 10 times in text A
“Galaxy” occurs 5 times in text A
“Heat” occurs 3 times in text A
(Blank means 0 occurrences.)
## Document Vectors

<table>
<thead>
<tr>
<th></th>
<th>nova</th>
<th>galaxy</th>
<th>heat</th>
<th>h'wood</th>
<th>film</th>
<th>role</th>
<th>diet</th>
<th>fur</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>10</td>
<td>8</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>9</td>
<td>10</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>6</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Document ids

Slide from Mitch Marcus
Vector Space Model

• **Documents are represented as** *vectors* **in term space**
  • Terms are usually stems
  • Documents represented by vectors of terms

• **A vector distance measures similarity between documents**
  • Document similarity is based on length and direction of their vectors
  • Terms in a vector can be “weighted” in many ways
# Document Vectors

<table>
<thead>
<tr>
<th>Document ids</th>
<th>nova</th>
<th>galaxy</th>
<th>heat</th>
<th>h’wood</th>
<th>film</th>
<th>role</th>
<th>diet</th>
<th>fur</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>10</td>
<td>8</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td>9</td>
<td>10</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>6</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Similarity between documents

\[ A = \begin{bmatrix} 10 & 5 & 3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \]
\[ G = \begin{bmatrix} 5 & 0 & 7 & 0 & 0 & 9 & 0 & 0 \end{bmatrix}; \]
\[ E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 10 & 10 & 0 \end{bmatrix}; \]
Similarity between documents

A = [10 5 3 0 0 0 0 0 0];
G = [ 5 0 7 0 0 9 0 0 0];
E = [ 0 0 0 0 0 10 10 0 0];

Treat the vectors as binary = number of words in common.

Sb(A,G) = ?
Sb(A,E) = ?
Sb(G,E) = ?

Which pair of documents are the most similar?
Similarity between documents

\[ A = [10 \ 5 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0] ; \]
\[ G = [5 \ 0 \ 7 \ 0 \ 0 \ 9 \ 0 \ 0] ; \]
\[ E = [0 \ 0 \ 0 \ 0 \ 10 \ 10 \ 0 \ 0] ; \]

Sum of Squared Distances (SSD) = \[ \sum_{i=1}^{n} (X_i - Y_i)^2 \]

SSD(A,G) = ?
SSD(A,E) = ?
SSD(G,E) = ?
Similarity between documents

\[ A = [10 \ 5 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0] ; \]
\[ G = [5 \ 0 \ 7 \ 0 \ 0 \ 9 \ 0 \ 0] ; \]
\[ E = [0 \ 0 \ 0 \ 0 \ 10 \ 10 \ 0] ; \]

Angle between vectors: \( \cos(\theta) = \frac{a \cdot b}{\|a\| \|b\|} \)

Dot Product: \( a \cdot b = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \)

Length (Euclidean norm): \( \|a\| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} \)
Some words give more information than others

- Does the fact that two documents both contain the word “the” tell us anything? How about “and”? 

[http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic/stop_words](http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic/stop_words)
Some words give more information than others

- Does the fact that two documents both contain the word “the” tell us anything? How about “and”? Stop words (noise words): Words that are probably not useful for processing. Filtered out before natural language is applied.
- Other words can be more or less informative.

No definitive list but might include things like:
http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
Vector Space Model

• **Documents are represented as vectors in term space**
  • Terms are usually stems
  • Documents represented by vectors of terms
• **A vector distance measures similarity between documents**
  • Document similarity is based on length and direction of their vectors
  • Terms in a vector can be “weighted” in many ways
Assigning Weights to Terms

1. Binary Weights
2. Raw term frequency
3. \textit{tf x idf}
   - Want to weight terms highly if they are 
     - frequent in relevant documents ... BUT
     - infrequent in the collection as a whole
TF x IDF Weights

- **tf x idf measure:**
  - Term Frequency (tf) – how often a term appears in a document
  - Inverse Document Frequency (idf) -- a way to deal with terms that are frequent across many documents

- **Goal:** Assign a tf * idf weight to each term in each document
TF x IDF Calculation

\[ w_{ik} = tf_{ik} \times \log(N / n_k) \]

\( T_k = \text{term } k \text{ in document } D_i \)
TF x IDF Calculation

\[ w_{ik} = tf_{ik} \times \log\left( \frac{N}{n_k} \right) \]

\( T_k = \text{term } k \text{ in document } D_i \)
\( tf_{ik} = \text{frequency of term } T_k \text{ in document } D_i \)
TF x IDF Calculation

\[ w_{ik} = tf_{ik} \times \log(\frac{N}{n_k}) \]

- \( T_k \) = term \( k \) in document \( D_i \)
- \( tf_{ik} \) = frequency of term \( T_k \) in document \( D_i \)
- \( idf_k \) = inverse document frequency of term \( T_k \) in \( C \)
- \( N \) = total number of documents in the collection \( C \)
- \( n_k \) = the number of documents in \( C \) that contain \( T_k \)
- \( idf_k = \log\left(\frac{N}{n_k}\right) \)
Inverse Document Frequency

- IDF provides high values for rare words and low values for common words

\[
\log\left(\frac{10000}{10000}\right) = 0
\]

\[
\log\left(\frac{10000}{5000}\right) = 0.301
\]

\[
\log\left(\frac{10000}{20}\right) = 2.698
\]

\[
\log\left(\frac{10000}{1}\right) = 4
\]

For a collection of 10000 documents
TF x IDF Normalization

- Normalize the term weights (so longer documents are not unfairly given more weight)
  - The longer the document, the more likely it is for a given term to appear in it, and the more often a given term is likely to appear in it. So, we want to reduce the importance attached to a term appearing in a document based on the length of the document.

\[ w_{ik} = \frac{tf_{ik} \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log(N / n_k)]^2}} \]
Pair-wise Document Similarity

Documents now represented as vectors of TFxIDF weights

\[ D_1 = w_{11}, w_{12}, \ldots, w_{1n} \]
\[ D_2 = w_{21}, w_{22}, \ldots, w_{2n} \]

<table>
<thead>
<tr>
<th></th>
<th>nova</th>
<th>galaxy</th>
<th>heat</th>
<th>h’wood</th>
<th>film</th>
<th>role</th>
<th>diet</th>
<th>fur</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pair-wise Document Similarity

Documents now represented as vectors of TFxIDF weights

\[ D_1 = w_{11}, w_{12}, \ldots, w_{1n} \]

\[ D_2 = w_{21}, w_{22}, \ldots, w_{2n} \]

Similarity can be computed as usual on these new weight vectors (e.g. \( \cos(\theta) \) here)

\[
sim(D_1, D_2) = \frac{\sum_{i=1}^{n} w_{1i} \cdot w_{2i}}{\sqrt{\sum_{i=1}^{n} (w_{1i})^2 \cdot \sum_{i=1}^{n} (w_{2i})^2}}
\]
Classes in the vector space

Here the vector space is illustrated as having 2 dimensions. How many dimensions would the data actually live in?
Query document – which class should you label it with?

Classes in the vector space

Should the document * be assigned to China, UK or Kenya?
Classification by Nearest Neighbor

Classes in the vector space

Should the document ★ be assigned to China, UK or Kenya?

Classify the test document as the class of the document “nearest” to the query document (use vector similarity to find most similar doc)
Classification by kNN

Classify the test document as the majority class of the k documents “nearest” to the query document.
kNN classification

- kNN = $k$ nearest neighbors
**kNN classification**

- kNN = \( k \) nearest neighbors
- For \( k = 1 \) (1NN), we assign each test document to the class of its **nearest neighbor** in the training set.
kNN classification

- kNN = \( k \) nearest neighbors
- For \( k = 1 \) (1NN), we assign each test document to the class of its nearest neighbor in the training set.
- 1NN is not very robust – one document can be mislabeled or atypical.
kNN classification

- kNN = $k$ nearest neighbors
- For $k = 1$ (1NN), we assign each test document to the class of its nearest neighbor in the training set.
- 1NN is not very robust – one document can be mislabeled or atypical.
- For $k > 1$, we assign each test document to the majority class of its $k$ nearest neighbors in the training set.
kNN classification

- kNN = $k$ nearest neighbors
- For $k = 1$ (1NN), we assign each test document to the class of its nearest neighbor in the training set.
- 1NN is not very robust – one document can be mislabeled or atypical.
- For $k > 1$, we assign each test document to the majority class of its $k$ nearest neighbors in the training set.

Rationale of kNN: contiguity hypothesis
- We expect a test document $d$ to have the same label as the training documents located in the local region surrounding $d$. 
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
Classification by kNN

Classes in the vector space

What are the features? What's the training data? Testing data? Parameters?

Should the document \( \star \) be assigned to China, UK or Kenya?
Classifiers

• Today we’ll talk about 2 simple kinds of classifiers
  – Nearest Neighbor Classifier
  – Naïve Bayes Classifier

Nearest neighbor treats all features equally whether or not they matter for the classification task (even tfxIDF weights are independent of document class). Naïve Bayes lets us learn which features are most indicative for classification.
Probabilistic Models

- A probabilistic model is a joint distribution over a set of variables

\[ P(X_1, X_2, \ldots X_n) \]
A random variable is some aspect of the world about which we (may) have uncertainty.

Random variables can be:
- Binary (e.g. \{true,false\}, \{spam/ham\}),
- Take on a discrete set of values (e.g. \{Spring, Summer, Fall, Winter\}),
- Or be continuous (e.g. [0 1]).
Probabilistic Models

- A probabilistic model is a joint distribution over a set of variables

\[ P(X_1, X_2, \ldots X_n) \]

- A joint distribution over a set of random variables: \( X_1, X_2, \ldots X_n \) specifies a real number for each assignment (or outcome):

\[ P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \]
Probabilistic Models

- A probabilistic model is a joint distribution over a set of variables
  \[ P(X_1, X_2, \ldots X_n) \]

- Given a joint distribution, we can reason about unobserved variables given observations (evidence)

- General form of a query:
  \[ P(X_q|x_{e1}, \ldots x_{e_k}) \]

  Stuff you care about  \[ \rightarrow \]  Stuff you already know
Probabilistic Models

- Models are descriptions of how (a portion of) the world works

- Models are always simplifications
  - May not account for every variable
  - May not account for all interactions between variables

- What do we do with probabilistic models?
  - We (or our agents) need to reason about unknown variables, given evidence
  - Example: explanation (diagnostic reasoning)
  - Example: prediction (causal reasoning)
  - Example: value of information
Graphical Model Notation

- **Nodes:** variables (with domains)
  - Can be assigned (observed) or unassigned (unobserved)

- **Arcs:** interactions

  Arrows encode relationships between variables (causality):

  ![Graphical Model Diagram]

  - Cavity
  - Toothache
  - Catch
Example: Coin Flips

- $N$ independent coin flips

- No interactions between variables: absolute independence
Example: Traffic

- Variables:
  - R: It rains
  - T: There is traffic

- Model 1: independence

- Model 2: rain causes traffic

- Why is an agent using model 2 better?
General Naïve Bayes

- A general *naive Bayes* model:

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

- Y class (e.g. spam/ham)
- \(F_1, F_2, \ldots, F_n\) Features (e.g. words)

- We only specify how each feature depends on the class
General Naïve Bayes

What do we need in order to use naïve Bayes?

- $P(Y)$, 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 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

- Example 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.
Example: Spam Filtering

- Model: $P(C,W_1 \ldots W_n) = P(C) \prod_i P(W_i|C)$
- What are the parameters?
Example: Spam Filtering

- **Model:** 
  \[ P(C, W_1 \ldots W_n) = P(C) \prod_i P(W_i|C) \]

- **What are the parameters?**

  \[
  \begin{array}{|c|}
  \hline
  P(C) \\
  \hline
  \text{ham} : 0.66 \\
  \text{spam} : 0.33 \\
  \hline
  \end{array}
  \quad
  \begin{array}{|c|}
  \hline
  P(W|\text{spam}) \\
  \hline
  \text{the} : 0.0156 \\
  \text{to} : 0.0153 \\
  \text{and} : 0.0115 \\
  \text{of} : 0.0095 \\
  \text{you} : 0.0093 \\
  \text{a} : 0.0086 \\
  \text{with} : 0.0080 \\
  \text{from} : 0.0075 \\
  \ldots \\
  \hline
  \end{array}
  \quad
  \begin{array}{|c|}
  \hline
  P(W|\text{ham}) \\
  \hline
  \text{the} : 0.0210 \\
  \text{to} : 0.0133 \\
  \text{of} : 0.0119 \\
  \text{2002} : 0.0110 \\
  \text{with} : 0.0108 \\
  \text{from} : 0.0107 \\
  \text{and} : 0.0105 \\
  \text{a} : 0.0100 \\
  \ldots \\
  \hline
  \end{array}
  \]

- **Where do these tables come from?**
Example: Spam Filtering

- **Model:**
  \[ P(C, W_1 \ldots W_n) = P(C) \prod_{i} P(W_i|C) \]

- **What are the parameters?**

  \[
  \begin{align*}
  P(C) & \\
  \text{ham : 0.66} & \\
  \text{spam: 0.33} & \\
  \\
  P(W|\text{spam}) & \\
  \text{the : 0.0156} & \\
  \text{to : 0.0153} & \\
  \text{and : 0.0115} & \\
  \text{of : 0.0095} & \\
  \text{you : 0.0093} & \\
  \text{a : 0.0086} & \\
  \text{with: 0.0080} & \\
  \text{from: 0.0075} & \\
  \ldots & \\
  \\
  P(W|\text{ham}) & \\
  \text{the : 0.0210} & \\
  \text{to : 0.0133} & \\
  \text{of : 0.0119} & \\
  \text{2002: 0.0110} & \\
  \text{with: 0.0108} & \\
  \text{from: 0.0107} & \\
  \text{and : 0.0105} & \\
  \text{a : 0.0100} & \\
  \ldots & \\
  \end{align*}
  \]

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

  Percentage of documents in training set labeled as spam/ham
Example: Spam Filtering

- Model: \( P(C, W_1 \ldots W_n) = P(C) \prod_i P(W_i | C) \)

- What are the parameters?

| \( P(C) \) | \( P(W|\text{spam}) \) | \( P(W|\text{ham}) \) |
|-----------|-----------------|-----------------|
| 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       |
|           | ...             | ...             |

In the documents labeled as spam, occurrence percentage of each word (e.g. # times “the” occurred/# total words).

- Where do these tables come from?
Example: Spam Filtering

- Model: 
  \[ P(C, W_1 \ldots W_n) = P(C) \prod_i P(W_i|C) \]

- What are the parameters?

\[
\begin{array}{|c|c|}
\hline
P(C) & P(W|\text{spam}) & P(W|\text{ham}) \\
\hline
\text{ham: 0.66} & \text{the: 0.0156} & \text{the: 0.0210} \\
\text{spam: 0.33} & \text{to: 0.0153} & \text{to: 0.0133} \\
\hline
\end{array}
\]

In the documents labeled as ham, occurrence percentage of each word (e.g. # times “the” occurred/# total words).

- Where do these tables come from?
Classification

The class that maximizes:

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

\[
= \arg\max_C P(C) \prod_i P(W_i | C)
\]

How would you do this?
In Practice:

- Multiplying lots of small probabilities can result in floating point underflow.
- Since \( \log(xy) = \log(x) + \log(y) \), we can sum \( \log \) probabilities instead of multiplying probabilities.
- Since \( \log \) is a monotonic function, the class with the highest score does not change.
- So what we usually compute in practice is:

\[
C_{map} = \arg \max_C \log(P(C)) + \sum_i \log(P(W_i | C))
\]
Naïve Bayes

• Is this a “bag of words” model?
Measuring Performance

• **Classification accuracy**: What % of messages were classified correctly?
• **Is this what we care about?**
Measuring Performance

• **Classification accuracy:** What % of messages were classified correctly?

• **Is this what we care about?**

<table>
<thead>
<tr>
<th></th>
<th>Overall accuracy</th>
<th>Accuracy on spam</th>
<th>Accuracy on ham</th>
</tr>
</thead>
<tbody>
<tr>
<td>System 1</td>
<td>95%</td>
<td>99.99%</td>
<td>90%</td>
</tr>
<tr>
<td>System 2</td>
<td>95%</td>
<td>90%</td>
<td>99.99%</td>
</tr>
</tbody>
</table>

Which system do you prefer?
Measuring Performance

- Precision = good messages kept / all messages kept
- Recall = good messages kept / all good messages

Trade off precision vs. recall by setting threshold
Measure the curve on annotated data
Choose a threshold where user is comfortable
Measuring Performance

**Precision vs. Recall of Good (non-spam) Email**

- **High threshold**: all we keep is good, but we don’t keep much
- **Low threshold**: keep all the good stuff, but a lot of the bad too
- **Point where precision = recall** (often reported)

- OK for search engines (maybe)
- Would prefer to be here!
- OK for spam filtering and legal search
Naive Bayes is Not So Naive

- **Naïve Bayes:** First and Second place in KDD-CUP 97 competition, among 16 (then) state of the art algorithms
  
  Goal: Financial services industry direct mail response prediction model: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.

- **Robust to Irrelevant Features**
  
  Irrelevant Features cancel each other without affecting results

- **Very good in Domains with many equally important features**

- **A good dependable baseline for text classification (but not the best)!**

- **Optimal if the Independence Assumptions hold:** If assumed independence is correct, then it is the Bayes Optimal Classifier for problem

- **Very Fast:** Learning with one pass over the data; testing linear in the number of attributes, and document collection size

- **Low Storage requirements**