Text Classification

Jason Rennie
jrennie@ai.mit.edu

Text Classification

- Assign text document a label based on content.
- Examples:
  - E-mail filtering
  - Knowedge-base creation
  - E-commerce
  - Question Answering
  - Information Extraction

E-mail Filtering

- Filter e-mail into folders set up by user.
- Aids searching for old e-mails
- Can be used to prioritize incoming e-mails
  - High priority to e-mails concerning your Ph.D. thesis
  - Low priority to “FREE Pre-Built Home Business”

Knowedge-Base Creation

- Company web sites provide large amounts of information about products, marketing contact persons, etc.
- Categorization can be used to find companies' web pages and organize them by industrial sector.
- This information can be sold to, e.g. person who wants to market "Flat Fixer" to tire company.

E-Commerce

- Users locate products in two basic ways: search and browsing.
- Browsing is best when user doesn't know exactly what he/she wants.
- Text classification can be used to organize products into a hierarchy according to description.
- EBay: Classification can be used to ensure that product fits category given by user.

Question Answering

- “When did George Washington die?”
- Search document database for short strings with answer.
- Rank candidates
- Many features (question type, proper nouns, noun overlap, verb overlap, etc)
- Problem: learn if string is the answer based on its feature values.

Information Extraction

- Want to extract information from talk announcements (room, time, date, title, speaker, etc)
- Many features may identify the information (keyword, punctuation, capitalization, numeric tokens, etc.)
- Problem: scan over text of document, filling buckets with desired information.
- Freitag (1998) showed that this approach could identify speaker (63%), location (76%), start time (99%) and end time (96%).

Basics of Text Classification

- Canonical Problem: Set of training documents, $(d_1, \ldots, d_n)$, with labels, $(y_1, \ldots, y_n)$. Set of test documents, $(x_1, \ldots, x_n)$.
- Goal: Assign correct labels to test documents.

Representation

```
From: dyer@spdcc.com (Steve Dyer)
Subject: Re: food-related seizures?

My comments about the Feingold Diet have no relevance to your
daughter’s purported FrostedFlakes-related seizures.  I can’t imagine
why you included it.
```

↓

food	1
seizures	2
diet	1
catering	0
religion	0
⋮	⋮

Representation

- Punctuation is removed, case is ignored, words are separated into tokens. Known as "feature vector" or "bag-of-words" representation.
- Vector length is size of vocabulary. Common vocabulary size is 10,000-100,000. Classification problem is very high dimensional.

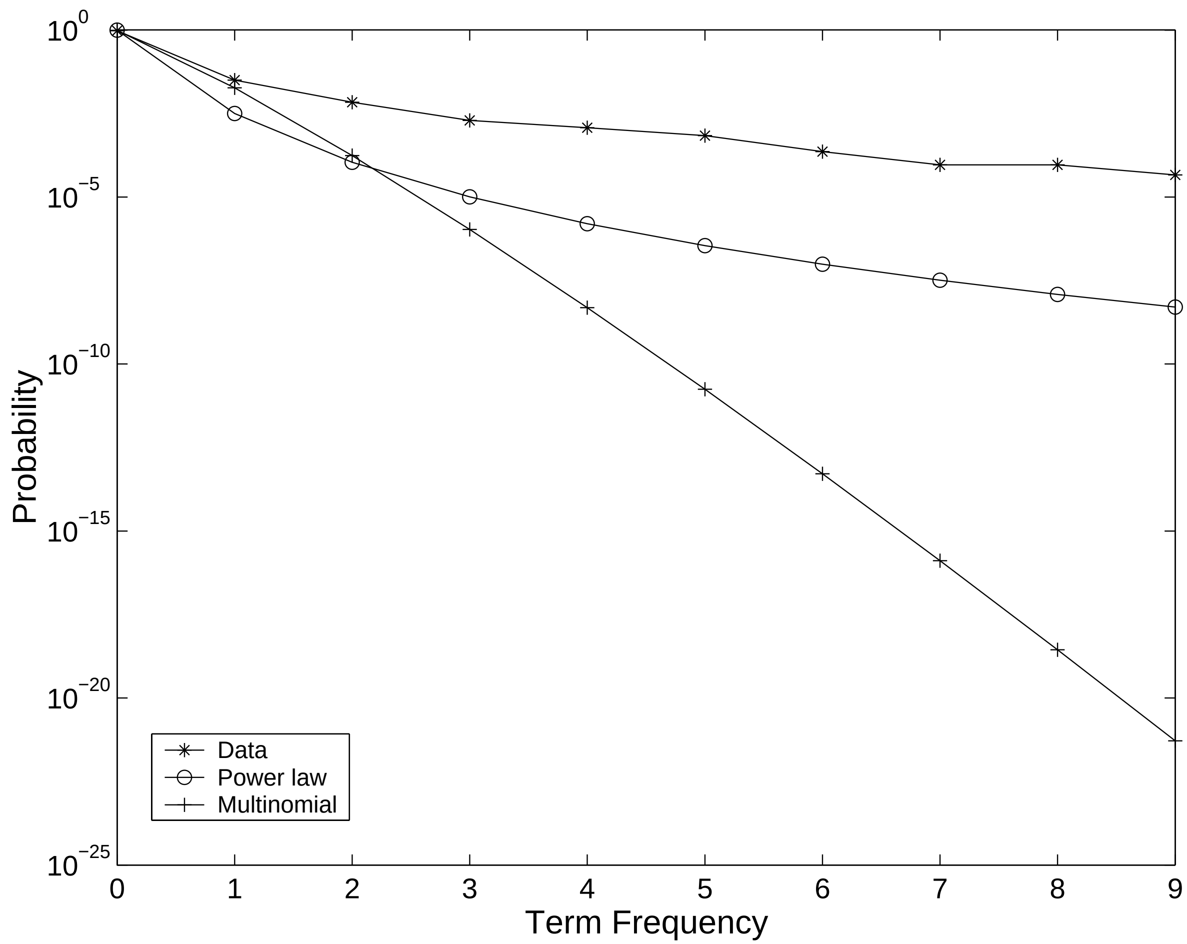
Why is text different?

- Near independence of features
- High dimensionality (often larger vocabulary than # of examples!)
- Importance of speed

Word Vector has Problems

- longer document $\Rightarrow$ larger vector
- words tend to occur a little or a lot
- rare words have same weight as common words

Text is Heavy Tailed



SMART "ltc" Transform

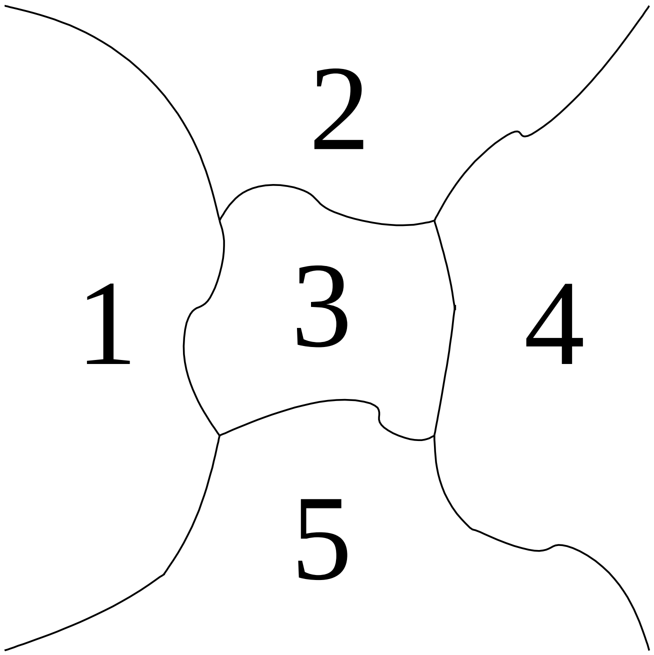
- $\text{new-tf}_i = \log(\text{tf}_i + 1.0)$
  - Corresponds to a power law distribution:
    $p(\text{tf}_i) \propto (1 + \text{tf}_i)^{\log \theta}$
- $\text{new-wt}_i = \text{new-tf}_i * \log \frac{\text{num-docs}}{\text{num-docs-with-term}}$ ("TFIDF")
- $\text{norm-wt}_i = \frac{\text{new-wt}_i}{\sqrt{\sum_i \text{new-wt}_i^2}}$ (unit length vectors)

Types of Classification Problems

- Binary: label each new document as positive or negative.
  Is this a news article Tommy would want to read?
- Multiclass: give one of m labels to each new document.
  Which customer support group should respond to this e-mail?
- Multitopic: assign zero to m topics to each new document.
  Who are good candidates for reviewing this research paper?
- Ranking: rank categories by relevance.
  Help user annotate documents by suggesting good categories.

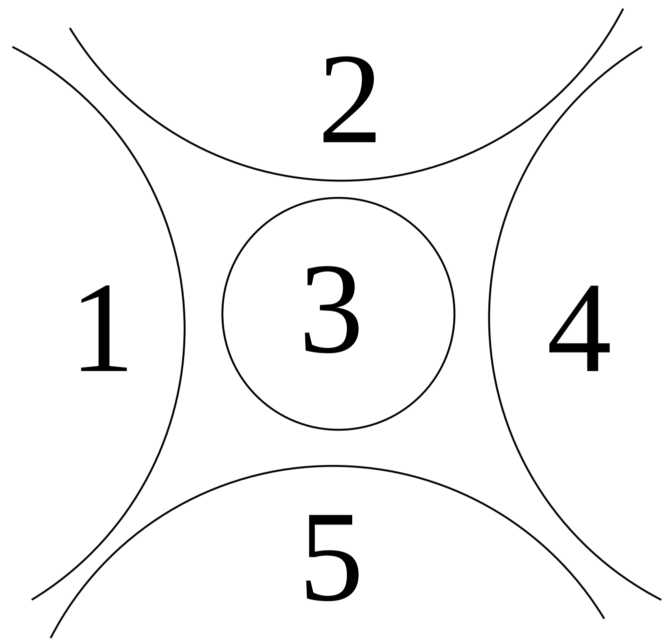
Multiclass Classification

- Decision Theory: minimum error decision boundary lies where density of top two classes are equal.
- Problem: Learning densities is ineffective for classification



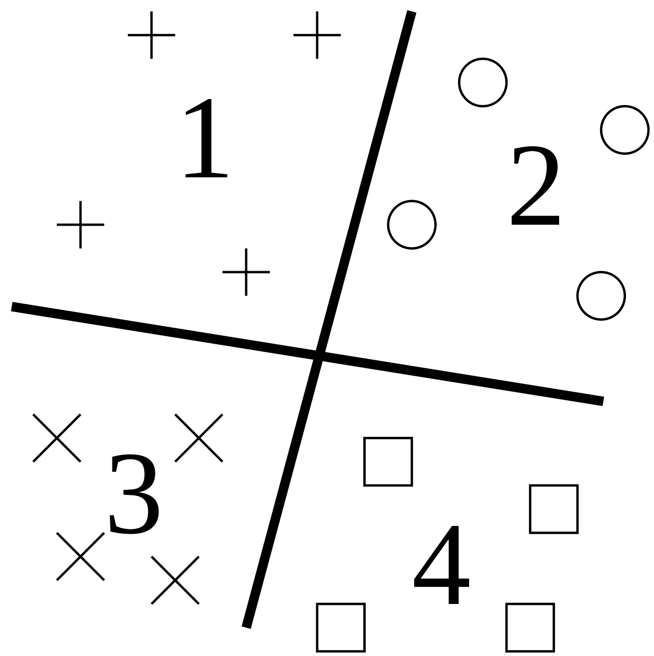
Multiclass Classification

- Simple approach: construct one binary classifier to discriminate each class from the rest.
- Problem: we can't say anything about the middle regions.



Multiclass Classification

- Better approach: construct lots of binary classifiers that, together, approximate the true boundaries.



Error Correcting Output Coding

- Idea: Represent each label as a length l binary code. Learn one binary classifier for each of the l bits in the code.
- For each example, assign label with "closest" code.
- Motivation: errors can be corrected using more bits than are needed to partition labels.

$$\begin{array}{c|ccccccc} 1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ 2 & +1 & -1 & -1 & -1 & +1 & -1 & -1 \\ 3 & -1 & +1 & -1 & -1 & -1 & +1 & -1 \\ 4 & -1 & -1 & +1 & -1 & -1 & -1 & +1 \end{array} \tag{1}$$

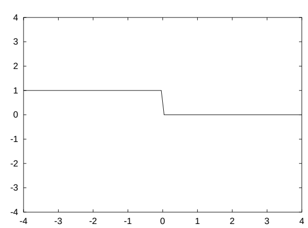
Code matrix

ECOC: The Loss Function

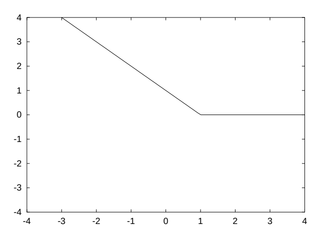
- ECOC works best when margin values are used

$$\hat{H}(x) = \arg \min_{c \in \{1,\ldots,m\}} \sum_{i=1}^{l} g(f_i(x) M_{ci}) \tag{2}$$

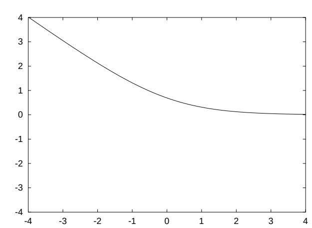
- The loss function (g) is a transform on the outputs:



Hamming



Hinge (SVM)



Logistic

ECOC: Some Results

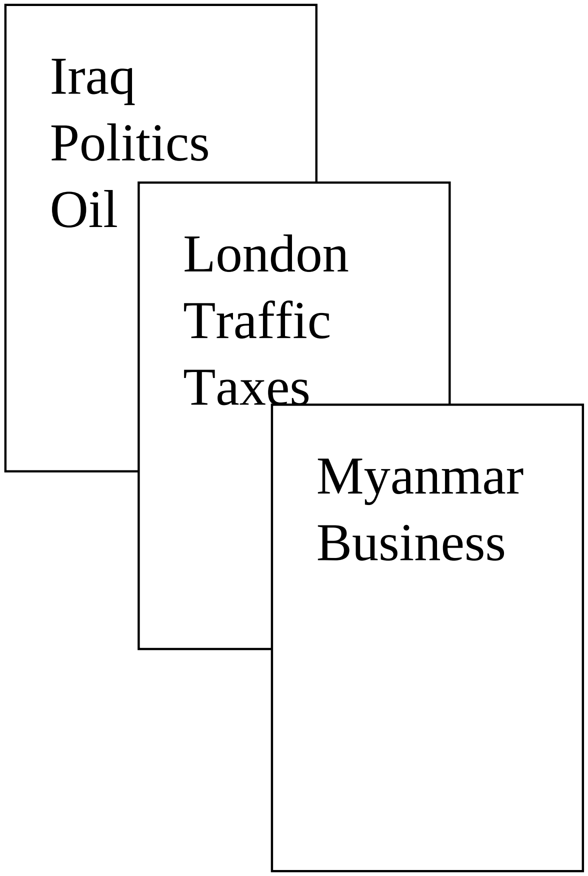
- ECOC works better than using the usual multiclass approach for DTs and NNs. (Dietterich and Bakiri, 1995).
- Loss-based decoding works better than Hamming decoding using SVMs (Allwein et. al., 2000).
- ECOC w/ loss decoding very effective for text classification (Rennie and Rifkin, 2001).

Multiclass Classification: Interesting Questions

- Is a continuous code matrix useful? (Crammer & Singer 2001)
- How do you construct best code matrix? (Crammer & Singer 2000) (Assumes existence of binary classifiers)

Multitopic Classification

- A document may be composed of many different topics.
- Zero or many topics per document.
- “Label” is a bit vector of topic indicators.



Multitopic Classification

- Basic approach: learn a binary classifier for each topic.

Iraq	vs.	Non-Iraq
Politics	vs.	Non-Politics
Oil	vs.	Non-Oil

- Problem: "Iraq" doucument contains other things too.

Multitopic Classification

- How to identify part of document that gives it “Iraq” topic?
- Easier problem: How do we model a multi-topic document?

Multitopic Classification

- If we ignore word order, each word is randomly generated from one of m topic-models.
- Problem becomes: how do we learn model for each topic?
- Ueda and Saito (2003) suggest modeling text as a multinomial and learning the models with an EM-like algorithm.

Iraq
Politics
Oil

Parametric Mixture Model

- Let $\vec{y}$ be a label (bit vector)
- Let $\vec{\theta}_t = (\theta_{t1}, \ldots, \theta_{tV})$ be the model for topic t.
- Let $h_t(\vec{y})$ be the label $\vec{y}$ mixing proportion for topic t.

Model for a document with label $\vec{y}$ is

$$\phi(\vec{y}) = \sum_{t=1}^{m} h_t(\vec{y})\vec{\theta}_t. \tag{3}$$

- Parameters for $\vec{y}$ are a *convex* combination

Parametric Mixture Model

- Simple case (PMM1): Assume $h_t(\vec{y})$ equals $\frac{1}{k}$, k is number of non-zero bits in $\vec{y}$. (convex optimization)

- Harder case (PMM2): Learn $h_t(\vec{y})$ via EM.

- Ueda and Saito: PMM1 works better than NB, SVM, kNN and NN. PMM2 useful in certain cases.

- PMM related to (McCallum 1999) and Latent Dirichlet Analysis (Blei, Ng, Jordan 2002)

Multitopic Classification: Interesting Problems

- Identify region(s) of document corresponding to topic(s)
- Capturing correlation between topics
- Hierarchy of topics (is parent or child more appropriate?)

Ranking

- How do you design a personalized search engine?
- Input: Ranking of documents based on relevance
- Want to learn a function that assigns rankings given a query

Ranking

- Option 1: Label documents rank R or higher "relevant," $R+1$ or lower "not-relevant," train a classifier. Rank based on classifier confidence values.
- Option 2: Train regression algorithm on rank values. Rank based on regression outputs.
- Option 3: Train a ranking algorithm.

Ranking

- A ranking algorithm has same form as classification and regression algorithms.
- Example: $f(x) = \sum w_i x_i$ (linear)
- Difference is training
- Question: What constitutes a mistake?

Ranking: What is a Mistake?

- Classification: mistake if predicted rank, r, greater than R and real rank, r^t less than R (or vice versa)
- Regression: error is difference between predicted value and true rank, $(r - r^t)^2$
- Ranking: mistake if documents are in wrong order

Ranking Loss: Examples

- Let $\{d_1, \ldots, d_n\}$ be a set of documents.
- Let $\{y_1^t, \ldots, y_n^t\}$ be the true ranks.
- Let $\{\hat{y}_1, \ldots, \hat{y}_n\}$ be the predicted ranks.
- Let $e_i = |y_i^t - \hat{y}_i|$.
- Loss $= \sum_i e_i$.

Ranking Loss

- Ranking Loss better suited to a ranking problem
- Crammer and Singer (2002) show that using a ranking loss function works better on text than using the zero-one classification loss.

Review

- “Text Classification” appears in many forms
- Multiclass classification
- Multitopic classification
- Ranking

Tokenization

- First step of text classification is tokenization.

Document $\rightarrow$ Tokenization $\rightarrow$ Stemming $\rightarrow$

Feature Selection $\rightarrow$ Bag of Words

"They just canceled them completely"

canceled	completely	just	them	they

Tokenization

- Tokenization determines the features for the classifier
- A bad classifier with good features can easily outperform a good classifier with bad features
- Very important step!

Tokenization

- Tokenization gets little attention
- Standard methods: separate on whitespace, alphabetic strings, alphanumeric strings.
- Problem: different tokenizations work best for different domains.
- Is there a better way?

Compression for Word Learning

- Can compression help tokenization?
- We want tokens to reflect features that appear in the documents.
- Compression encourages the construction of features that appear more frequently than their individual characters would imply.

Compression for Word Learning: An Idea

- Begin with individual characters as the tokens.
- Allow pairs of tokens to be compressed together.
- De Marcken (1995) did exactly this.
- Creates a hierarchical decomposition of documents.

Compression: Examples

Rank	$-\log p_G(w)$	w	rep(w)
0	4.589	.	*terminal*
1	4.890	,	*terminal*
100	10.333	[two]	[[two]]
101	10.342	[it was]	[[it][was]]
501	12.467	[ized]	[[ize]d]
502	12.469	[ling]	[l[ing]]
15000	16.684	[pakistan]	[[pa]k[ist][an]]
15001	16.684	[creativity]	[[creat][ivity]]
27167	18.006	[[massachusetts][institute of technology]]	

Compression: Hierarchy Example

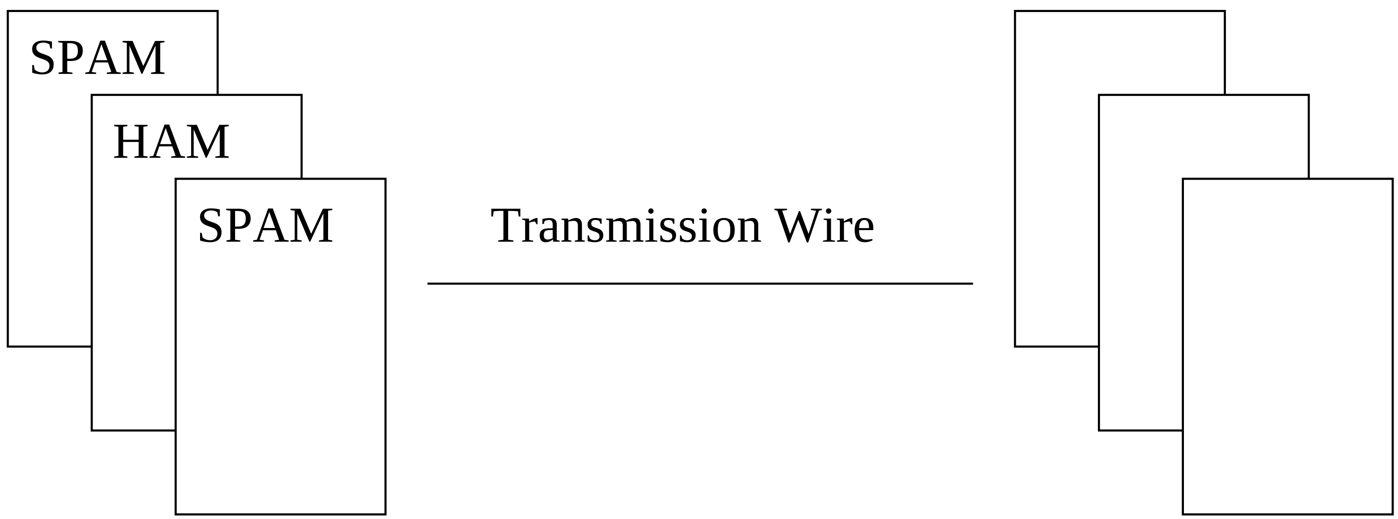
[[f[or]][[t[he]][[[p[ur]][[[po]s]e][of]]]][[[ma[in]][ta[in]]][[in]g]]
[[[in][t[er]]][[n[a[t[i[on]]]]][al]]][[pe][a[ce]]][[an]d][[p[ro]][[mo]t][[in]g]]
[t[he]][[adv[a[n[ce]]]][[[me]n]t]][[of][a[ll]]][[pe][op][le]][[[t[he]]
[[[[un][it]][ed]][[[st[at]]e]s]]][[of][a[me][r[ic]]a]]][[[jo][in]][ed]][in]
[f[o[un]d]][[in]g][[t[he]][[[[un][it]][ed]][[n[a[t[i[on]]]]]s]]]

- Tokens can be taken from any level of the hierarchy—from "ur" to "the united nations."
- Much more useful than collecting all substrings.
- Compression object eliminates numerous meaningless strings.

Classification via Compression

Standard compression problem:

- Want to transmit labels with fewest number of bits.
- Documents can be used as background knowledge.
- What is fewest number of bits needed to transmit labels?



Examples of Learned Features

␣x	comp.os.xwindows
␣windows	comp.os.ms-windows.misc
␣car␣	rec.autos
for␣sale	misc.forsale
␣turk	talk.politics.mideast
486	comp.sys.ibm.pc.hardware
3.1	comp.os.ms-windows.misc
␣␣$	misc.forsale
t␣condition	misc.forsale

String Kernels

- Kernel method
- Documents projected into feature space of substrings
- Requires discount factor (longer strings receive less weight)
- Thought up by Haussler (1999) and Watkins (1999).
- Lodhi et. al. (2001) successfully applied string kernels to text—found they work about as well as substrings.

Summary

- Text classification comes in many different flavors.
- Text presents interesting and unique problems.