Text Categorization using Classical ML

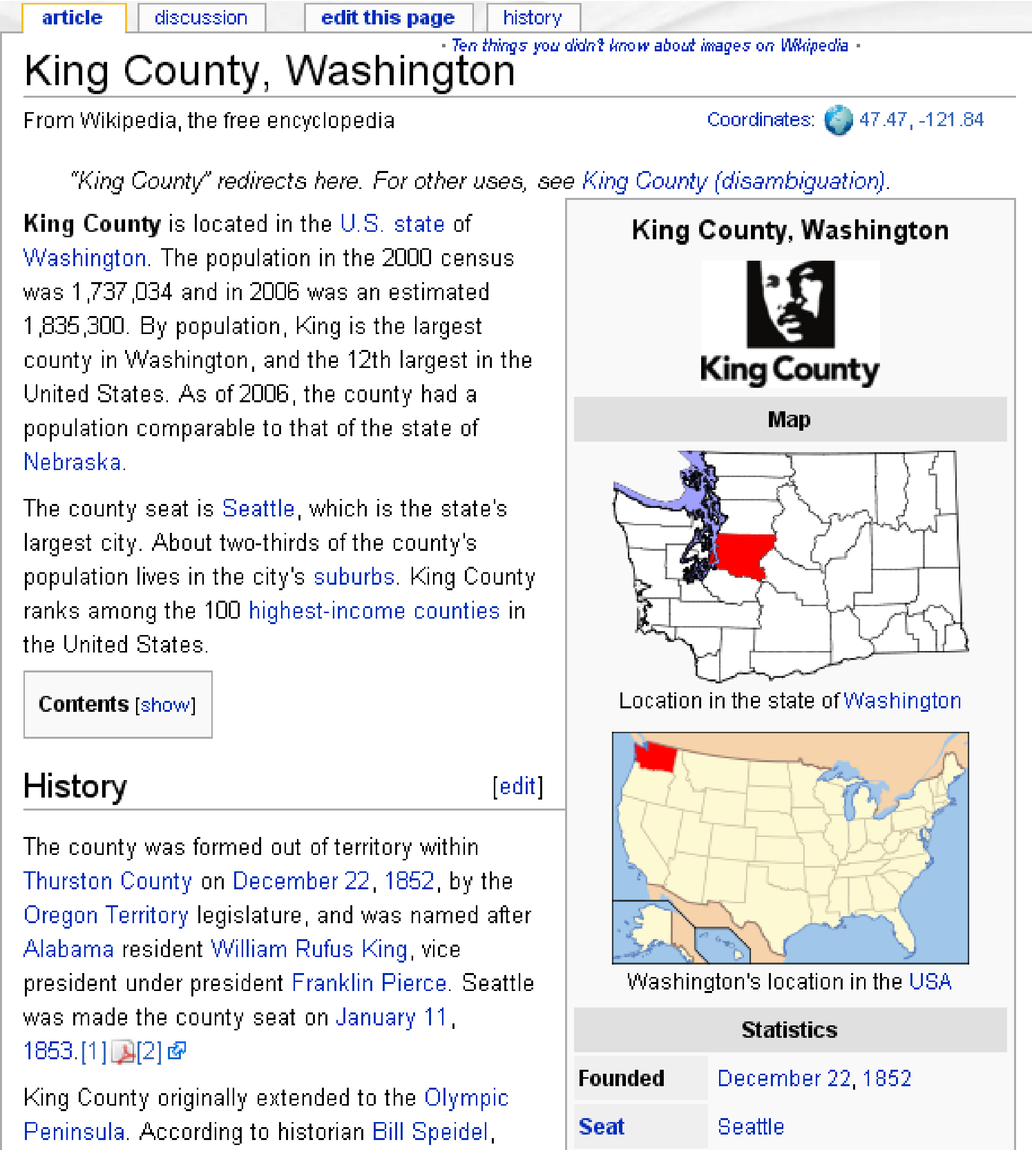
Mausam

(based on slides of Dan Weld, Dan Jurafsky, Prabhakar Raghavan, Hinrich Schutze, Guillaume Obozinski, David D. Lewis, Fei Xia, Michael Collins, Emily Fox, Alexander Ihler, Dan Jurafsky, Dan Klein, Chris Manning, Ray Mooney, Mark Schmidt, Dan Weld, Alex Yates, Luke Zettlemoyer)

Categorization

- Given:
  - A **description of an instance**, $x \in X$, where X is the *instance language* or *instance space*.
  - A **fixed set of categories**: $C = \{c_1, c_2, \ldots c_n\}$
- Determine:
  - The **category of *x***: $c(x) \in C$, where $c(x)$ is a categorization function whose domain is X and whose range is C.

County *vs.* Country?



article | discussion | edit this page | history

· Ten things you didn't know about images on Wikipedia ·

King County, Washington

From Wikipedia, the free encyclopedia

Coordinates: 47.47, -121.84

"King County" redirects here. For other uses, see King County (disambiguation).

King County is located in the U.S. state of Washington. The population in the 2000 census was 1,737,034 and in 2006 was an estimated 1,835,300. By population, King is the largest county in Washington, and the 12th largest in the United States. As of 2006, the county had a population comparable to that of the state of Nebraska.

The county seat is Seattle, which is the state's largest city. About two-thirds of the county's population lives in the city's suburbs. King County ranks among the 100 highest-income counties in the United States.

Contents [show]

History [edit]

The county was formed out of territory within Thurston County on December 22, 1852, by the Oregon Territory legislature, and was named after Alabama resident William Rufus King, vice president under president Franklin Pierce. Seattle was made the county seat on January 11, 1853.[1][2]

King County originally extended to the Olympic Peninsula. According to historian Bill Speidel,

King County, Washington

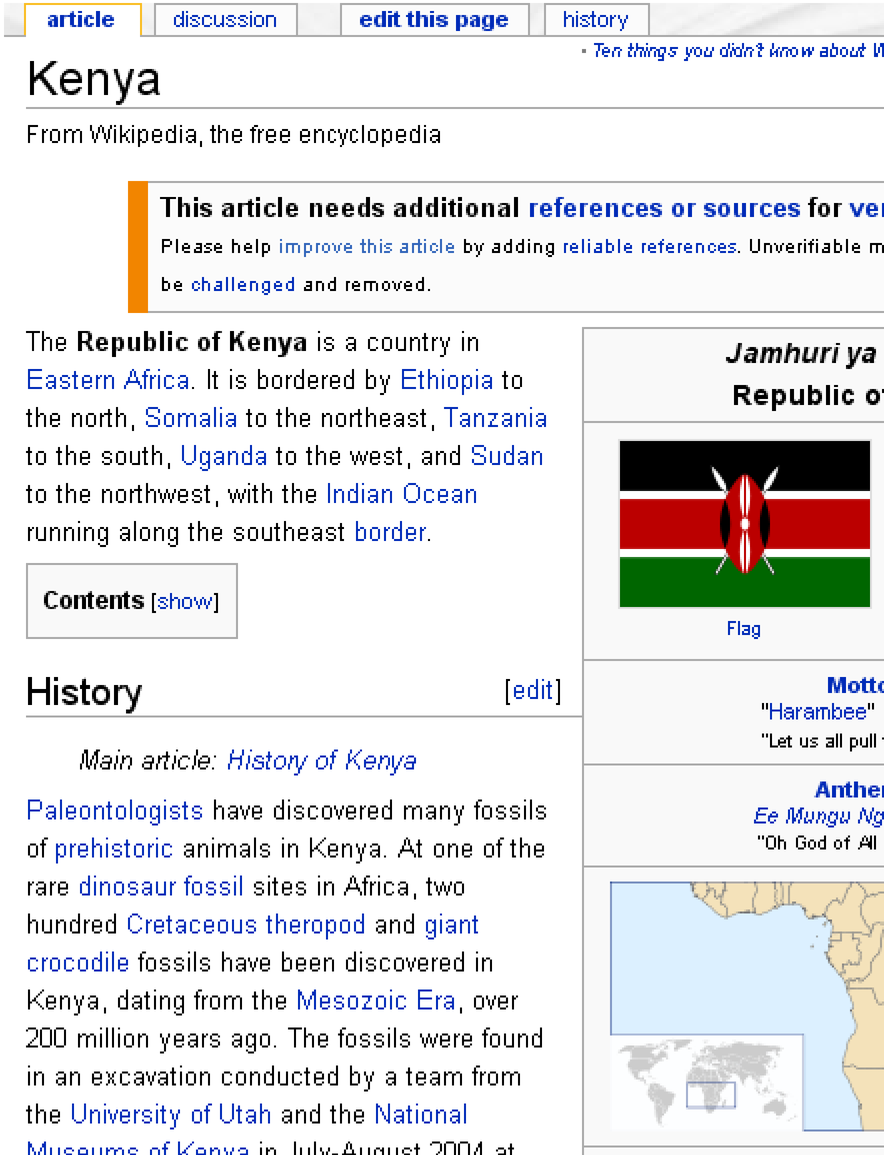
King County

Map

Location in the state of Washington

Washington's location in the USA

Statistics	
Founded	December 22, 1852
Seat	Seattle



article | discussion | edit this page | history

· Ten things you didn't know about W

Kenya

From Wikipedia, the free encyclopedia

This article needs additional references or sources for ver

Please help improve this article by adding reliable references. Unverifiable m be challenged and removed.

The **Republic of Kenya** is a country in Eastern Africa. It is bordered by Ethiopia to the north, Somalia to the northeast, Tanzania to the south, Uganda to the west, and Sudan to the northwest, with the Indian Ocean running along the southeast border.

Contents [show]

History [edit]

Main article: History of Kenya

Paleontologists have discovered many fossils of prehistoric animals in Kenya. At one of the rare dinosaur fossil sites in Africa, two hundred Cretaceous theropod and giant crocodile fossils have been discovered in Kenya, dating from the Mesozoic Era, over 200 million years ago. The fossils were found in an excavation conducted by a team from the University of Utah and the National Museums of Kenya in July-August 2004 at

Jamhuri ya

Republic o

Flag

Motto

"Harambee"

"Let us all pull

Anthem

Ee Mungu Ng

"Oh God of All

Male or female author?

- The main aim of this article is to propose an exercise in stylistic analysis which can be employed in the teaching of English language. It details the design and results of a workshop activity on narrative carried out with undergraduates in a university department of English. The methods proposed are intended to enable students to obtain insights into aspects of cohesion and narrative structure: insights, it is suggested, which are not as readily obtainable through mor

- My aim in th pproach to utterance int nding of what some of these so-called apposition markers indicate. It will be argued that the decision to put something in other words is essentially a decision about style, a point which is, perhaps, anticipated by Burton-Roberts when he describes loose apposition as a rhetorical device. However, he does not justify this suggestion by giving the criteria for classifying a mode of expression as a rhetorical device.

Female writers use

more first person/second person pronouns

more gender laiden third person pronouns

(overall more personalization)

S. Argamon, M. Koppel, J. Fine, A. R. Shimoni, 2003. “Gender, Genre, and Writing Style in Formal Written Texts,” Text, volume 23, number 3, pp. 321–346

Positive or negative movie review?

👎 • unbelievably disappointing

👍 • Full of zany characters and richly applied satire, and some great plot twists

👍 • this is the greatest screwball comedy ever filmed

👎 • It was pathetic. The worst part about it was the boxing scenes.

What is the subject of this article?

MEDLINE Article

ELSEVIER

Available online at www.sciencedirect.com

Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

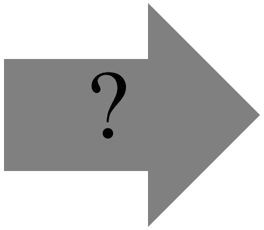
Susanne Gahl,[a] Lise Menn,[b] Gail Ramsberger,[b] Daniel S. Jurafsky,[b] Elizabeth Elder,[b] Molly Rewega,[b] and L. Holland Audrey[c]

Abstract

[illegible]

1. Introduction

[illegible]



MeSH Subject Category Hierarchy

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

Text Classification

- Assigning documents to a fixed set of categories, *e.g.*
- Web pages
  - Yahoo-like classification
  - Assigning subject categories, topics, or genres
- Email messages
  - Spam filtering
  - Prioritizing
  - Folderizing
- Blogs/Letters/Books
  - Authorship identification
  - Age/gender identification
- Reviews/Social media
  - Language Identification
  - Sentiment analysis
  - …

Classification Methods: Hand-coded rules

- Rules based on combinations of words or other features
  - spam: black-list-address OR (“dollars” AND “have been selected”)
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive
- MACHINE LEARNING

Bayesian Methods

- Learning and classification methods based on probability theory.
  - Bayes theorem plays a critical role in probabilistic learning and classification.
  - Uses *prior* probability of each category given no information about an item.
- Categorization produces a ***posterior*** probability distribution over the possible categories given a description of an item.

The bag of words representation

$\gamma($

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

$)=c$

👍

👎

The bag of words representation

$\gamma($

I **love** this movie! It's **sweet,** but with **satirical** humor. The dialogue is **great** and the adventure scenes are **fun…** It manages to be **whimsical** and **romantic** while **laughing** at the conventions of the fairy tale genre. I would **recommend** it to just about anyone. I've seen it **several** times, and I'm always **happy** to see it **again** whenever I have a friend who hasn't seen it yet.

$)=c$

👍

👎

The bag of words representation: using a subset of words

$\gamma($

x **love** xxxxxxxxxxxxxxxx **sweet**
xxxxxxx **satirical** xxxxxxxxxx
xxxxxxxxxxx **great** xxxxxxx
xxxxxxxxxxxxxxxxxxx **fun** xxxx
xxxxxxxxxxxxx **whimsical** xxxx
romantic xxxx **laughing**
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx **recommend** xxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx **several** xxxxxxxxxxxxxxxxx
xxxxx **happy** xxxxxxxxx **again**
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx

$)=c$

👍

👎

The bag of words representation

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

$)=c$

👍

👎

Bayes' Rule Applied to Documents and Classes

- For a document *d* and a class *c*

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Naïve Bayes Classifier (I)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Dropping the denominator

Naïve Bayes Classifier (II)

$$c_{MAP} = \underset{c \square C}{\operatorname{argmax}} P(d \mid c) P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \quad ., x_n \mid c) P(c)$$

Document d represented as features x1..xn

Naïve Bayes Classifier (IV)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \quad ., x_n \mid c) P(c)$$

O(|*X*|^n•|*C*|) parameters

How often does this class occur?

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

Multinomial Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \square C}{\operatorname{argmax}} P(c_j) \underset{x \square X}{\square} P(x \mid c)$$

Multinomial Naïve Bayes Independence Assumptions

$$P(x_1, x_2, \quad ., x_n \mid c)$$

- **Bag of Words assumption**: Assume position doesn’t matter
- **Conditional Independence**: Assume the feature probabilities $P(x_i|c_j)$ are independent given the class *c.*

$$P(x_1, \quad ., x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Problem with Maximum Likelihood

Sec.13.3

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive** (***thumbs-up)***?

$$\hat{P}(\text{"fantastic"} \,|\text{positive}) \;=\; \frac{count(\text{"fantastic", positive})}{\square_{w \square V} \; count(w,\text{positive})} \;=\; 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \text{argmax}_c \, \hat{P}(c) \square_i \, \hat{P}(x_i \,|\, c)$$

Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\underset{w\square V}{\square} \left(count(w, c)\right) + 1)}$$

$$= \frac{count(w_i, c) + 1}{\square\, \underset{w\square V}{\square}\, count(w, c) \,\square + |V|}$$

Easy to Implement

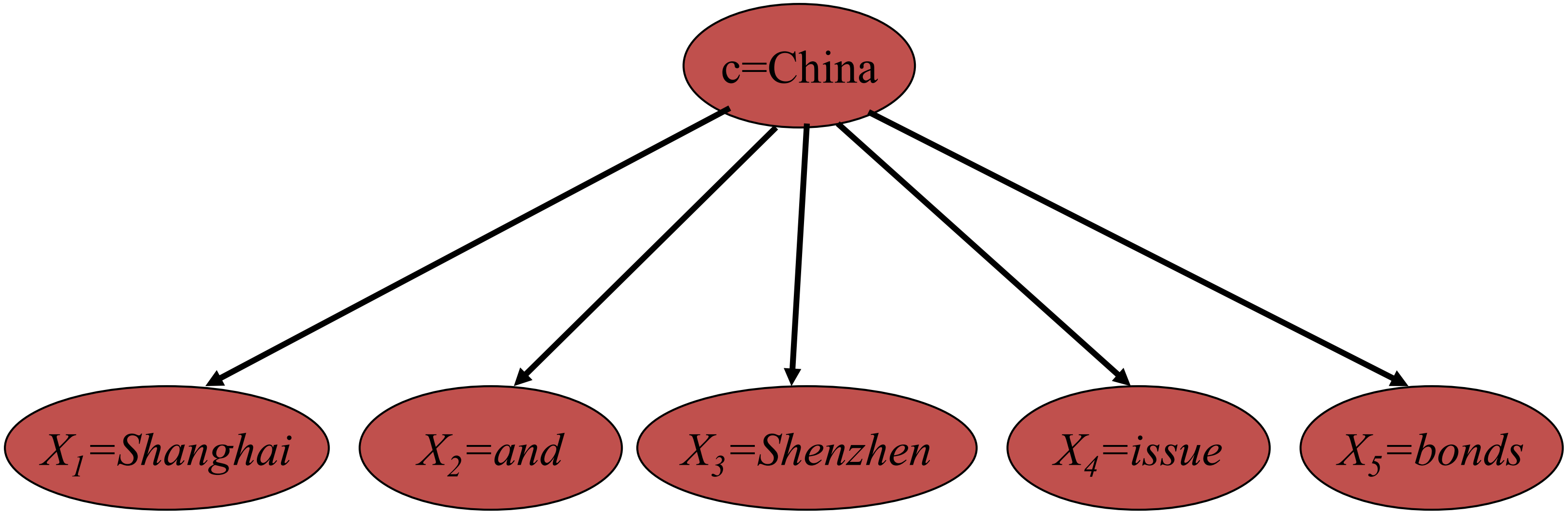
- But…

- If you do… it probably won’t work…

Probabilities: Important Detail!

- We are multiplying lots of small numbers
  Danger of underflow!
  - $0.5^{57} = 7$ E -18
- Solution? Use logs and add!
  - $p_1 * p_2 = e^{\log(p1)+\log(p2)}$
  - Always keep in log form

Generative Model for Multinomial Naïve Bayes



Advantages

- Simple to implement
  - No numerical optimization, matrix algebra, etc
- Efficient to train and use
  - Easy to update with new data
  - Fast to apply
- Binary/multi-class
- Good in domains with many equally important features
  - Decision Trees suffer from fragmentation in such cases – especially if little data
- Comparatively good effectiveness with small training sets
- A good dependable baseline for text classification
  - But we will see other classifiers that give better accuracy

Disadvantages

- Independence assumption wrong
  - Absurd estimates of class probabilities
    - Output probabilities close to 0 or 1
  - Thresholds must be tuned; not set analytically
- Generative model
  - Generally lower effectiveness than discriminative techniques

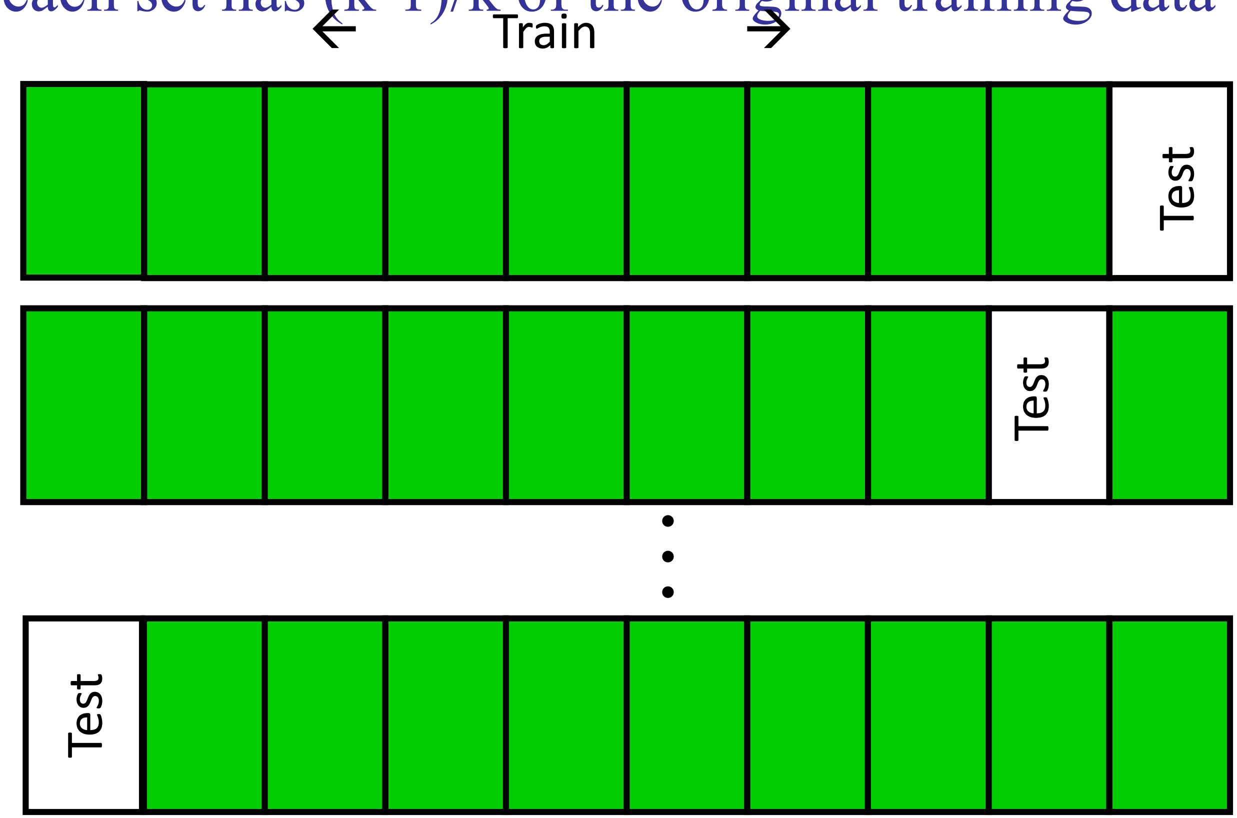
Experimental Evaluation

Question: How do we estimate the performance of classifier on unseen data?

- Can’t just at accuracy on training data – this will yield an over optimistic estimate of performance
- Solution: Cross-validation
- Note: this is sometimes called estimating how well the classifier will generalize

Evaluation: Cross Validation

- Partition examples into *k* disjoint sets
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



Cross-Validation (2)

- Leave-one-out
  - Use if < 100 examples (rough estimate)
  - Hold out one example, train on remaining examples

- 10-fold
  - If have 100-1000's of examples

Joint vs. Conditional Models

- We have some data $\{(d, c)\}$ of paired observations d and hidden classes c.
- Joint (generative) models place probabilities over both observed data and the hidden stuff (generate the observed data from hidden stuff):
  - All the classic Stat-NLP models:
    - *n*-gram models, Naive Bayes classifiers, hidden Markov models, probabilistic context-free grammars, IBM machine translation alignment models

Joint vs. Conditional Models

- Discriminative (conditional) models take the data as given, and put a probability over hidden structure given the data:
  - Logistic regression, conditional loglinear or maximum entropy models, conditional random fields
  - Also, SVMs, (averaged) perceptron, etc. are discriminative classifiers (but not directly probabilistic)

Conditional vs. Joint Likelihood

- A *joint* model gives probabilities $P(d,c)$ and tries to maximize this joint likelihood.
  - It turns out to be trivial to choose weights: just relative frequencies.
- A *conditional* model gives probabilities $P(c|d)$. It takes the data as given and models only the conditional probability of the class.
  - We seek to maximize conditional likelihood.
  - Harder to do (as we'll see…)
  - More closely related to classification error.

Text Categorization with Word Features

Data
BUSINESS: Stocks hit a yearly low …

Label: BUSINESS
Features
{…, stocks, hit, a, yearly, low, …}

(Zhang and Oles 2001)

- Features are presence of each word in a document and the document class (they do feature selection to use reliable indicator words)
- Tests on classic Reuters data set (and others)
  - Naïve Bayes: 77.0% F_1
  - Logistic regression: 86.4%
  - Support vector machine: 86.5%

Feature-Based Linear Classifiers

- Linear classifiers at classification time:
  - Linear function from feature sets $\{\phi_i\}$ to classes $\{y\}$.
  - Assign a weight w_i to each feature ϕ_i.
  - We consider each class for an observed datum x
  - For a pair (x,y), features vote with their weights:
    - vote(y) = $\Sigma w_i \phi_i(x,y)$
    - Choose the class y which maximizes $\Sigma w_i \phi_i(x,y)$

Features for Multi-Class Problems

- $\phi_i(x,y) = 1$ *if* $\phi_i(x) = 1$ *and* $label(x) = y$

  $= 0$ *otherwise*

Assign a weight for each feature $\phi_i(x,y)$*, i.e., a different weight for each prediction y*

For a pair (x,y), features vote with their weights:

- vote(y) = $\Sigma w_i \phi_i(x,y)$
- Choose the class y which maximizes $\Sigma w_i \phi_i(x,y)$
- This can be written in linear algebra notation as $W^T X$ and it will yield a |X|x|Y| matrix with a score for each (x,y)

“all models are wrong
some are useful!”

-- *George Box*

Exponential Models (log-linear, maxent, Logistic, Gibbs)

- Model: use the scores as probabilities:

$$p(y|x;w) = \frac{\exp\left(w \cdot \phi(x,y)\right)}{\sum_{y'} \exp\left(w \cdot \phi(x,y')\right)}$$

← Make positive

← Normalize

- Learning: maximize the (log) conditional likelihood of training data

$$\{(x_i, y_i)\}_{i=1}^{n}$$

$$L(w) = \sum_{i=1}^{n} \log p(y_i|x_i;w) \qquad w^* = \arg\max_{w} L(w)$$

- Prediction: output argmax$_y$ p(y|x;w)

Derivative of Log-linear Model

$$p(y|x;w) = \frac{\exp\left(w \cdot \phi(x,y)\right)}{\sum_{y'} \exp\left(w \cdot \phi(x,y')\right)}$$

- Unfortunately, argmax$_w$ L(w) doesn't have a close formed solution
- We will have to differentiate and use gradient ascent

$$L(w) = \sum_{i=1}^{n} \log p(y_i|x_i;w)$$

$$L(w) = \sum_{i=1}^{n} \left(w \cdot \varphi(x_i,y_i) - \log \sum_{y} \exp(w \cdot \varphi(x_i,y)) \right)$$

$$\frac{\partial L(w)}{\partial w_{jk}} = \sum_{i=1}^{n} \left(\varphi_{jk}\,(x_i,y_i) - \mathrm{p}(\mathrm{k}|x_i;\mathrm{w})\varphi_{jk}(x_i,\mathrm{k}) \right)$$

Total count of feature j in candidates with class k

Expected count of feature j in predicted candidates of class k

Proof (Conditional Likelihood Derivative)

- Recall

$$\boxed{p(y|x;w) = \frac{\exp\left(w \cdot \phi(x,y)\right)}{\sum_{y'} \exp\left(w \cdot \phi(x,y')\right)}}$$

$$P(Y \mid X, w) = \prod_{(x,y)\in D} p(y \mid x, w)$$

- We can separate this into two components:

$$\log P(Y|X,w) = \sum_{i=1}^{n} (w \cdot \varphi(x_i,y_i)) - \sum_{i=1}^{n} \left(\log \sum_{y} \exp(w \cdot \varphi(x_i,y)) \right)$$

- The derivative is the difference between the derivatives of each component

$$\log P(Y \mid X, w) = N(w) - D(w)$$

Proof: Numerator

$$\frac{\partial N(w)}{\partial w_{jk}} = \frac{\partial \sum_{i=1}^{n}\left(\sum_{l}\left(w_{ly_i}\varphi_{ly_i}(x_i,y_i)\right)\right)}{\partial w_{jk}}$$

$$= \sum_{i=1}^{n} \frac{\partial\left(\sum_{l}\left(w_{ly_i}\varphi_{ly_i}(x_i,y_i)\right)\right)}{\partial w_{jk}}$$

$$= \sum_{i=1}^{n} \varphi_{jk}\,(x_i,y_i)$$

Derivative of the numerator is:

the empirical count of feature j with class k

Note: $\varphi_{jk}\,(x_i,y_i)$=0 if y≠ k

Proof: Denominator

$$\frac{\partial D(w)}{\partial w_{jk}} = \frac{\partial \sum_{i=1}^{n} \log \sum_y \exp(\sum_l (w_{ly} \varphi_{ly}\ (x_i,y))}{\partial w_{jk}}$$

$$= \sum_{i=1}^{n} \frac{1}{\sum_{y'} \exp(\sum_l (w_{ly'} \varphi_{ly'}\ (x_i,y')))} \frac{\partial \sum_y \exp(\sum_l (w_{ly} \varphi_{ly}\ (x_i,y)))}{\partial w_{jk}}$$

$$= \sum_{i=1}^{n} \frac{1}{\sum_{y'} \exp(\sum_l (w_{ly'} \varphi_{ly'}\ (x_i,y')))} \sum_y \frac{\exp(\sum_l (w_{ly} \varphi_{ly}\ (x_i,y\ ))}{1} \frac{\partial \sum_l (w_{ly} \varphi_{ly} (x_i,y\ ))}{\partial w_{jk}}$$

$$= \sum_{i=1}^{n} \sum_{y} \frac{\exp(\sum_l (w_{ly} \varphi_{ly}\ (x_i,y)))}{\sum_{y'} \exp(\sum_l (w_{ly'} \varphi_{ly'}\ (x_i,y')))} \varphi_{jk}\ (x_i,y)$$

$$= \sum_{i=1}^{n} \sum_{y} P(y|x_i; w)\ \ \varphi_{jk}\ (x_i,y)$$

$$= \sum_{i=1}^{n} \mathrm{p}(\mathrm{k}|x_i; \mathrm{w}) \varphi_{jk} (x_i,\mathrm{k})$$

= expected count of
feature j predicted with class k

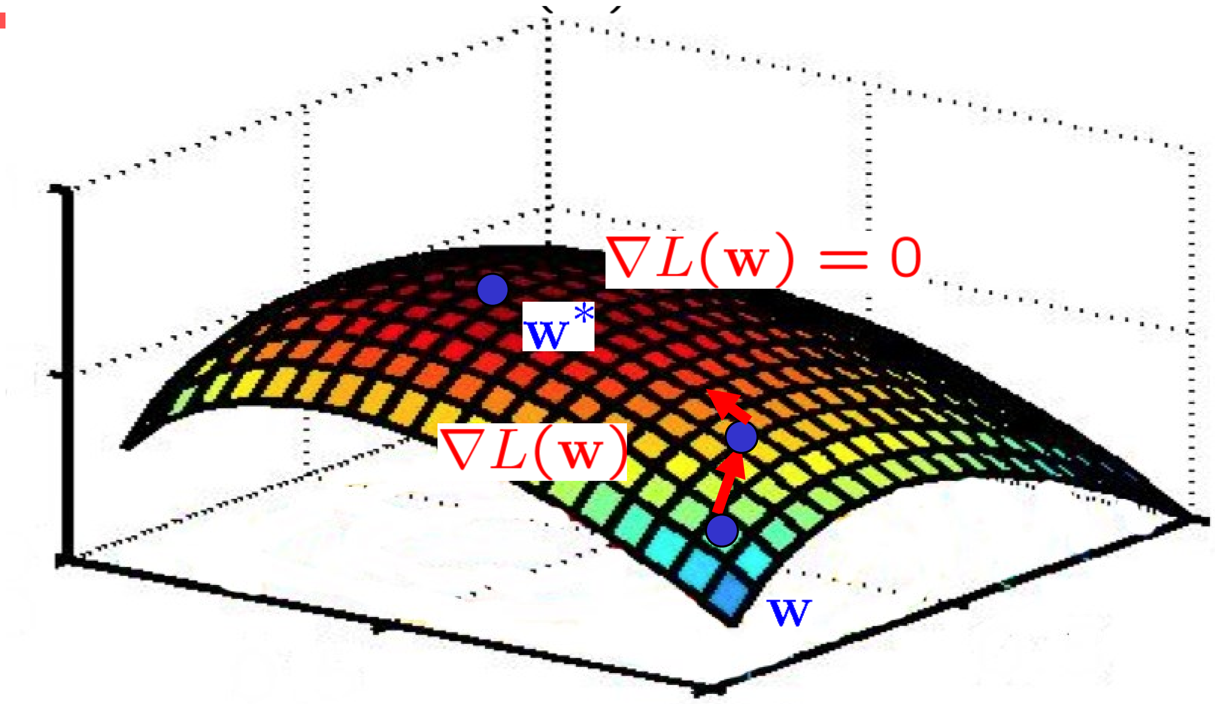
Proof (concluded)

$$\frac{\partial P(Y|X;w)}{\partial w_{jk}} = actualcount(\varphi_{jk}) - \mathrm{predictedcount}(\varphi_{jk})$$

- The optimum parameters are the ones for which each feature's predicted expectation equals its empirical expectation. The optimum distribution is:
  - Always unique (but parameters may not be unique)
  - Always exists (if feature counts are from actual data).
- These models are also called maximum entropy models because we find the model has the maximum entropy while satisfying the constraints:

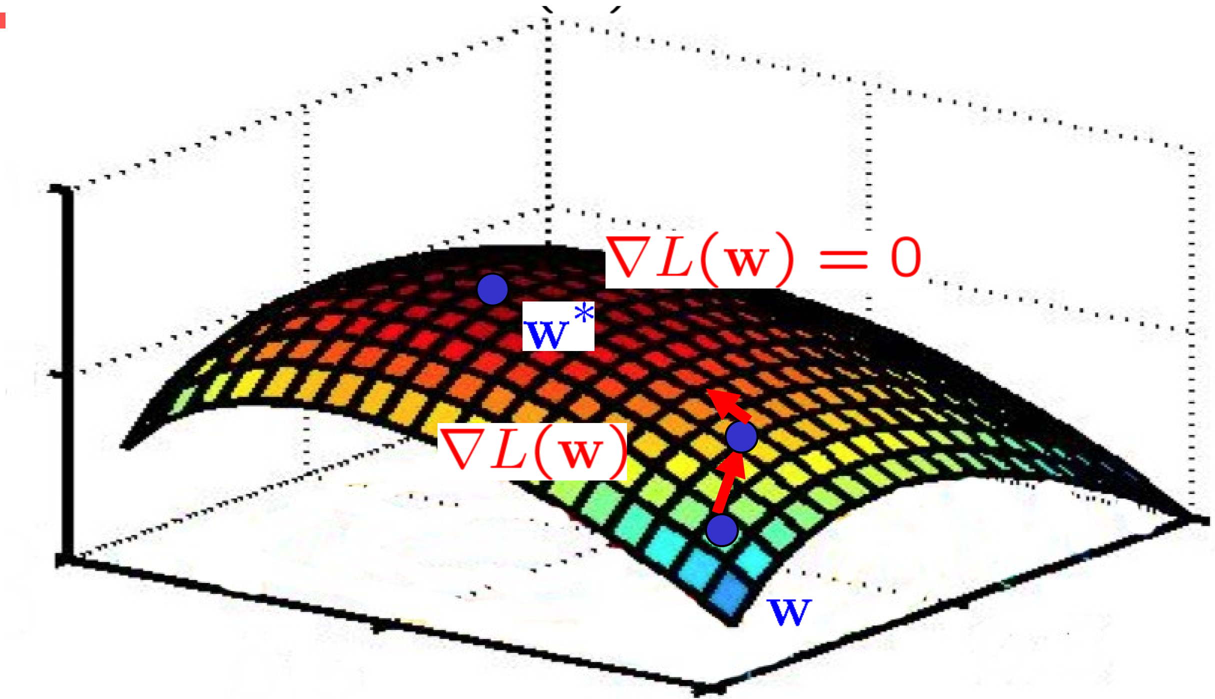
$$E_p(\phi_i) = E_{\tilde{p}}(\phi_i), \forall i$$

Unconstrained Optimization



- Basic idea: move uphill from current guess
- Gradient ascent / descent follows the gradient incrementally
- At local optimum, derivative vector is zero
- Will converge if step sizes are small enough, but not efficient
- All we need is to be able to evaluate the function and its derivative

Unconstrained Optimization



- For convex functions, a local optimum will be global
- Basic gradient ascent isn't very efficient, but there are simple enhancements which take into account previous gradients: conjugate gradient, L-BFGS
- There are special-purpose optimization techniques for maxent, like iterative scaling, but they aren't better

What About Overfitting?

- For Naïve Bayes, we were worried about zero counts in MLE estimates
  - Can that happen here?

- Regularization (smoothing) for Log-linear models
  - Instead, we worry about large feature weights
  - Add a regularization term to the likelihood to push weights towards zero

$$L(w) = \sum_{i=1}^{n} \log p(y_i | x_i; w) - \frac{\lambda}{2} ||w||^2$$

Derivative for Regularized Maximum Entropy

- Unfortunately, argmax$_w$ L(w) still doesn't have a close formed solution
- We will have to differentiate and use gradient ascent

$$L(w) = \sum_{i=1}^{n} \left(w \cdot \phi(x_i, y_i) - \log \sum_{y} \exp(w \cdot \phi(x_i, y)) \right) - \frac{\lambda}{2}||w||^2$$

$$\frac{\partial}{\partial w_j} L(w) = \sum_{i=1}^{n} \left(\phi_j(x_i, y_i) - \sum_{y} p(y|x_i; w)\phi_j(x_i, y) \right) - \lambda w_j$$

Total count of feature j in correct candidates

Expected count of feature j in predicted candidates

Big weights are bad

L1 and L2 Regularization

L2 Regularization for Log-linear models

- Instead, we worry about large feature weights
- Add a regularization term to the likelihood to push weights towards zero

$$L(w) = \sum_{i=1}^{n} \log p(y_i|x_i; w) - \frac{\lambda}{2}||w||^2$$

Regularization Constant

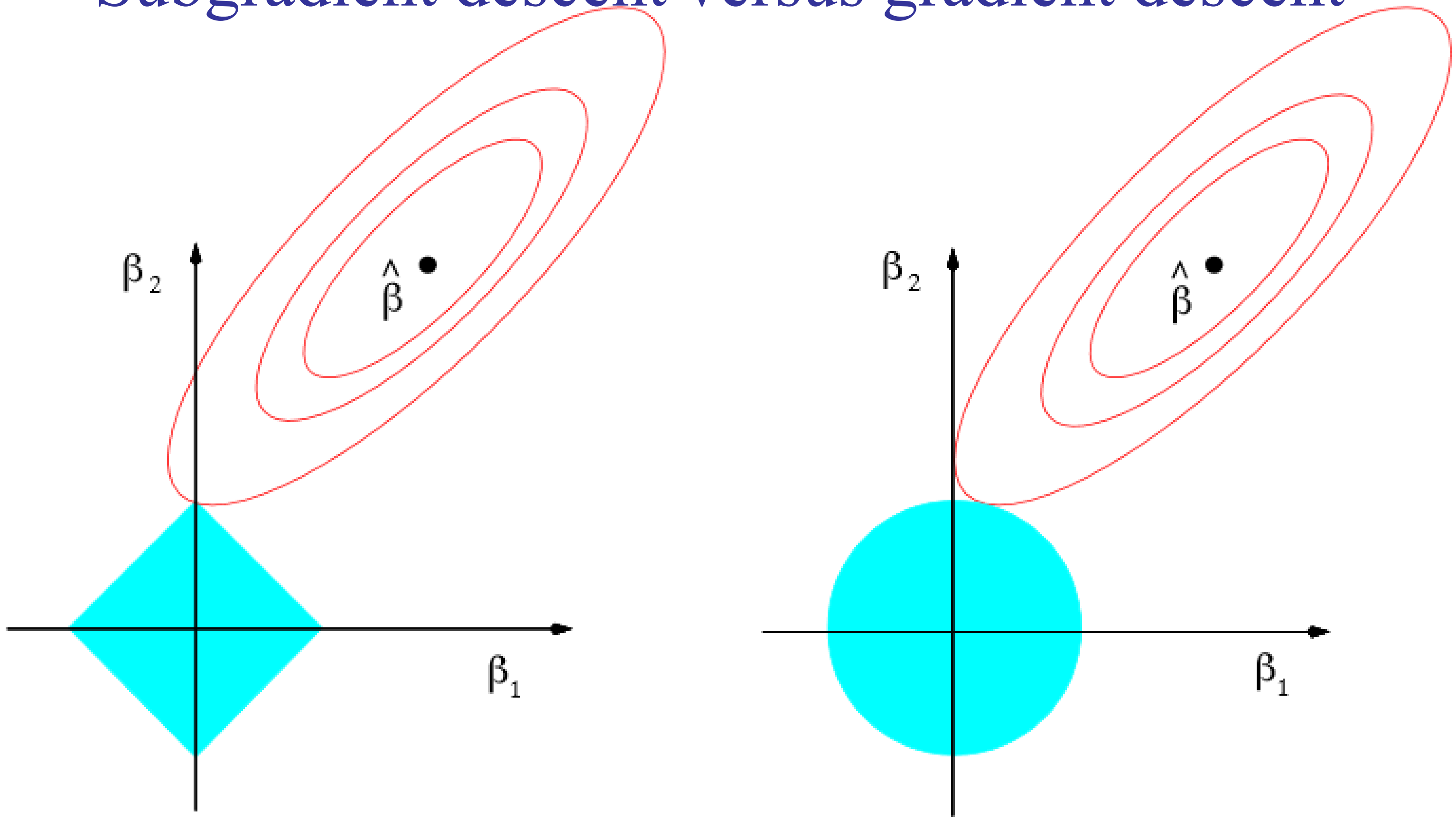
L1 Regularization for Log-linear models

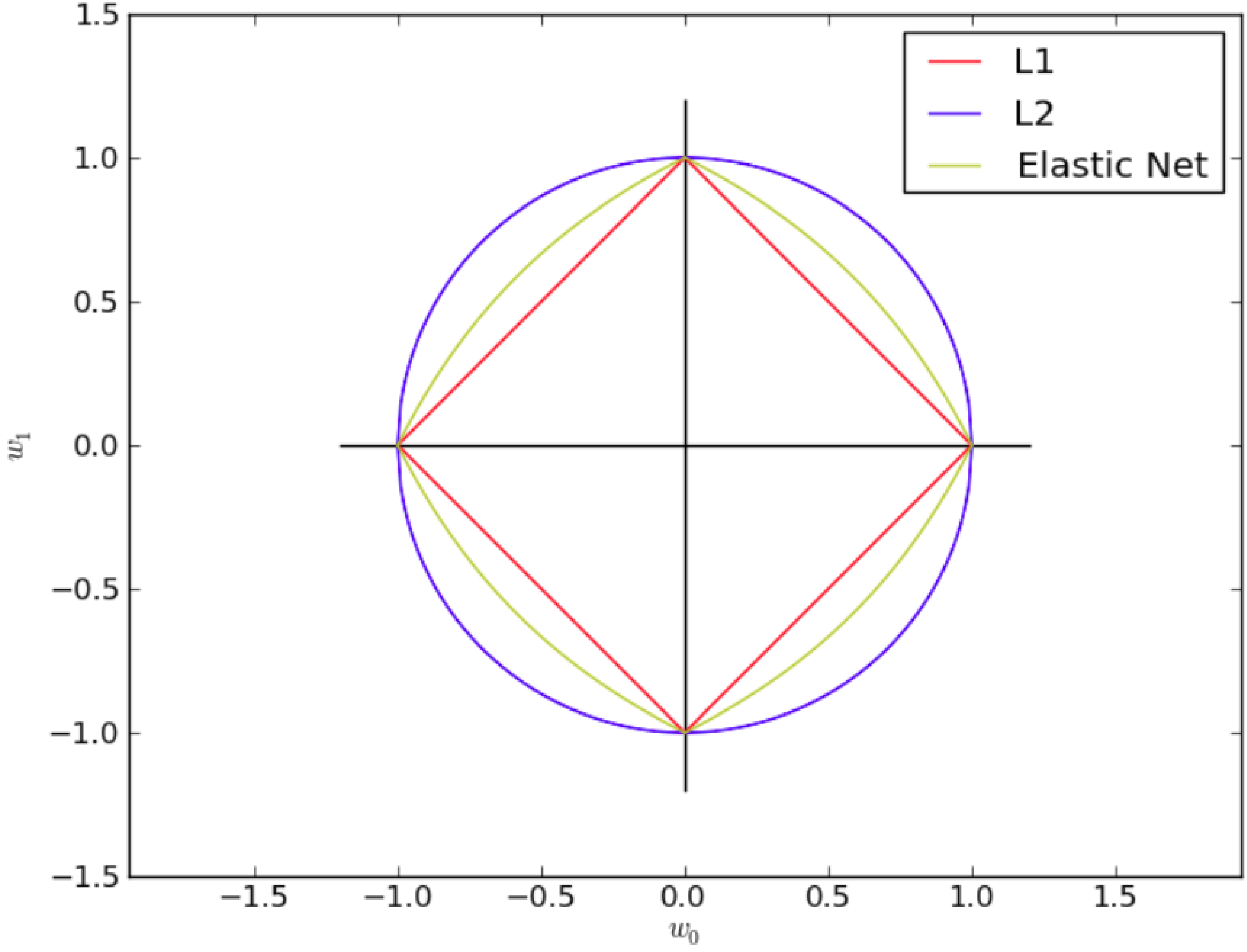
- Instead, we worry about number of active features
- Add a regularization term to the likelihood to push weights to zero
- For L1 regularization, we need to compute subgradients.

$$L(w) = \sum_{i=1}^{n} \log p(y_i|x_i; w) - \lambda||w||$$

L1 vs L2

- Optimizing L1 harder
  - Discontinuous objective function
  - Subgradient descent versus gradient descent




1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
w_1
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
w_0
L1
L2
Elastic Net

How to pick weights?

- Goal: choose “best” vector w given training data
  - For now, we mean “best for classification”
- The ideal: the weights which have greatest test set accuracy / F1 / whatever
  - But, don’ t have the test set
  - Must compute weights from training set
- Maybe we want weights which give best training set accuracy?
  - May not (does not) generalize to test set
  - Easy to overfit
- Use devset

Diving Deeper into Feature Engineering

Construct Better Features

- Key to machine learning is having good features
- In gen 2 ML, large effort devoted to constructing appropriate features
- Ideas??

Issues in document representation

Cooper’s concordance of Wordsworth was published in 1911. The applications of full-text retrieval are legion: they include résumé scanning, litigation support and searching published journals on-line.

- *Cooper’s vs. Cooper vs. Coopers.*
- *Full-text vs. full text vs. {full, text} vs. fulltext.*
- *résumé vs. resume.*

slide from Raghavan, Schütze, Larson

Punctuation

- *Ne'er*: use language-specific, handcrafted "locale" to normalize.
- *State-of-the-art*: break up hyphenated sequence.
- *U.S.A.* vs. *USA*
- *a.out*

slide from Raghavan, Schütze,

Numbers

- 3/12/91
- Mar. 12, 1991
- 55 B.C.
- B-52
- 100.2.86.144
  - Generally, don't represent as text
  - Creation dates for docs

slide from Raghavan, Schütze, Larson

Possible Feature Ideas

- Look at capitalization (may indicated a proper noun)

- Look for commonly occurring sequences
  - E.g. New York, New York City
  - Limit to 2-3 consecutive words
  - Keep all that meet minimum threshold (e.g. occur at least 5 or 10 times in corpus)

Case folding

- Reduce all letters to lower case
- Exception: upper case in mid-sentence
  - *e.g.,* ***General Motors***
  - ***Fed*** vs. ***fed***
  - ***SAIL*** vs. ***sail***

slide from Raghavan, Schütze, Larson

Thesauri and Soundex

- Handle synonyms and spelling variations
  - – Hand-constructed equivalence classes
    - e.g., ***car*** = ***automobile***

slide from Raghavan, Schütze,

Spell Correction

- Look for all words within (say) edit distance 3 (Insert/Delete/Replace) at query time
  - *e.g., **arfiticial inteligence***
- Spell correction is expensive and slows the processing significantly
  - Invoke only when index returns zero matches?

slide from Raghavan, Schütze,

Stemming

- Are there different index terms?
  - retrieve, retrieving, retrieval, retrieved, retrieves…
- Stemming algorithm:
  - (retrieve, retrieving, retrieval, retrieved, retrieves) ⇨ retriev
  - Strips prefixes of suffixes (-s, -ed, -ly, -ness)
  - Morphological stemming
- Problems: sand / sander & wand / wander

Features

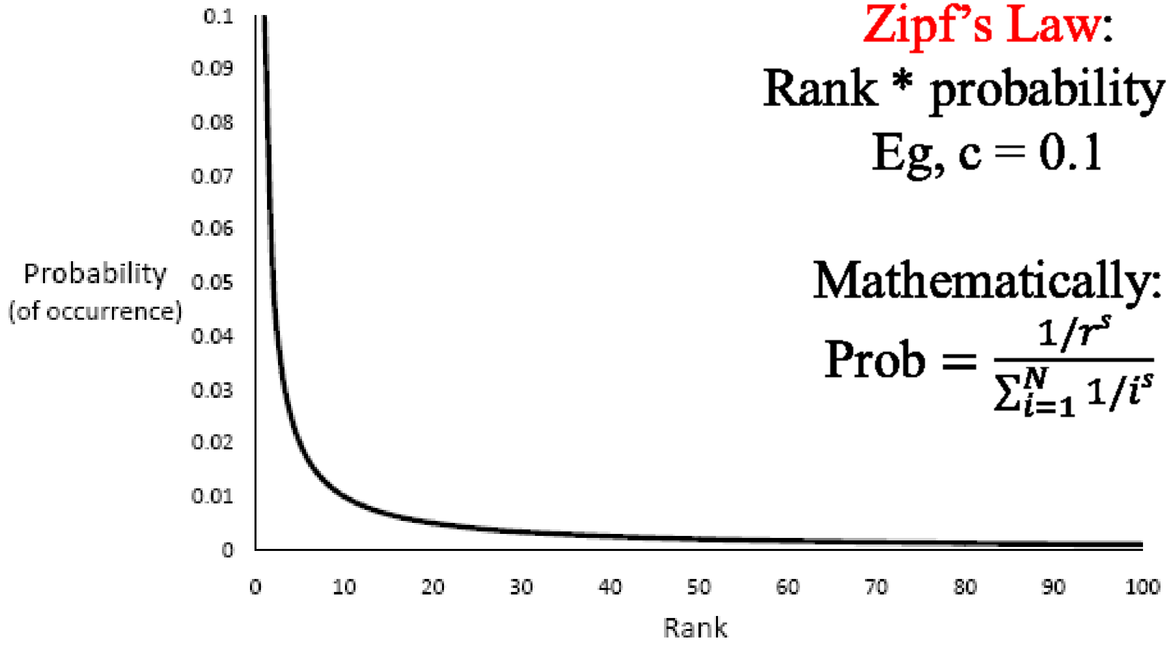
Sec. 15.3.2

- Domain-specific features and weights: *very* important in real performance

- Upweighting: Counting a word as if it occurred twice:
  - title words (Cohen & Singer 1996)
  - first sentence of each paragraph (Murata, 1999)
  - In sentences that contain title words (Ko *et al*, 2002)

Properties of Text

- Word frequencies - skewed distribution
- `The' and `of' account for 10% of all words
- Six most common words account for 40%



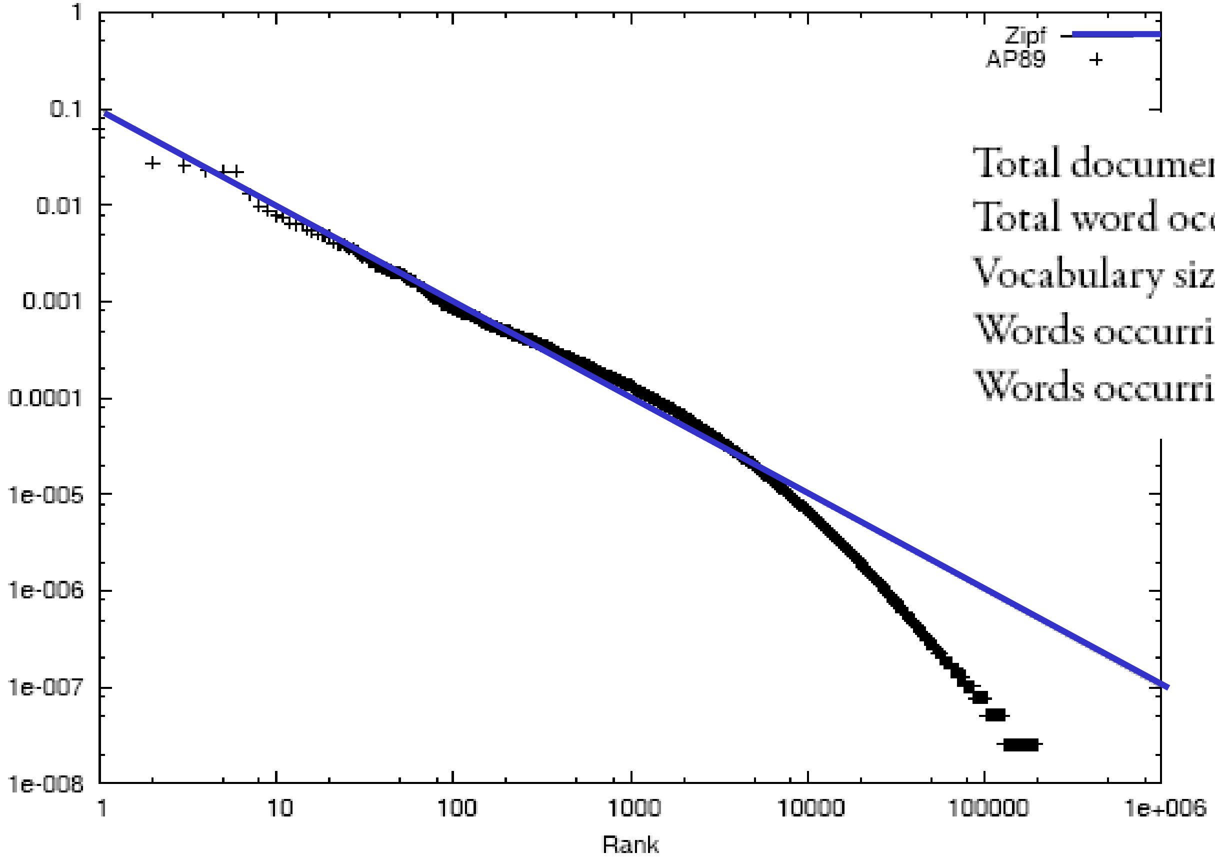
Zipf's Law:
Rank * probability = c
Eg, c = 0.1

Mathematically:

$$\text{Prob} = \frac{1/r^s}{\sum_{i=1}^{N} 1/i^s}$$

From [Croft, Metzler & Strohman 2010]

Associate Press Corpus `AP89’



Total documents	84,678
Total word occurrences	39,749,179
Vocabulary size	198,763
Words occurring > 1000 times	4,169
Words occurring once	70,064

From [Croft, Metzler & Strohman 2010]

Middle Ground

- Very common words → bad features
  - Language-based stop list:

    words that bear little meaning

    20-500 words

    http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
  - Subject-dependent stop lists

- Very rare words *also* bad features

Drop words appearing less than k times / corpus

Word Frequency

- Which word is more indicative of document similarity?
  - ‘book,’ or ‘Rumplestiltskin’?
  - Need to consider “document frequency”--- how frequently the word appears in doc collection.

- Which doc is a better match for the query “Kangaroo”?
  - One with a single mention of Kangaroos… or a doc that mentions it 10 times?
  - Need to consider “term frequency”--- how many times the word appears in the current document.

TF x IDF

$$w_{ik} = tf_{ik} * \log(N / n_k)$$

$T_k = term\ k\ in\ document\ D_i$

$tf_{ik} = frequency\ of\ term\ T_k\ in\ document\ D_i$

$idf_k = inverse\ document\ frequency\ of\ term\ T_k\ in\ C$

$$idf_k = \log\left(\frac{N}{n_k}\right)$$

$N = total\ number\ of\ documents\ in\ the\ collection\ C$

$n_k = the\ number\ of\ documents\ in\ C\ that\ contain\ T_k$

Inverse Document Frequency

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

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

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

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

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

- Add 1 to avoid 0.

TF-IDF normalization

- Normalize the term weights
  - so longer docs not given more weight (fairness)
  - force all values to fall within a certain range: [0, 1]

$$w_{ik} = \frac{tf_{ik}(1 + \log(N / n_k))}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [1 + \log(N / n_k)]^2}}$$

Evaluation in Multi-class Problems

Evaluation: Classic Reuters-21578 Data Set

Sec. 15.2.4

- Most (over)used data set, 21,578 docs (each 90 types, 200 tokens)
- 9603 training, 3299 test articles (ModApte/Lewis split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Average document (with at least one category) has 1.24 classes
- Only about 10 out of 118 categories are large

Common categories (#train, #test)

- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)

- Trade (369,119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)

Reuters Text Categorization data set (**Reuters-21578**) document

Sec. 15.2.4

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE>   CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter

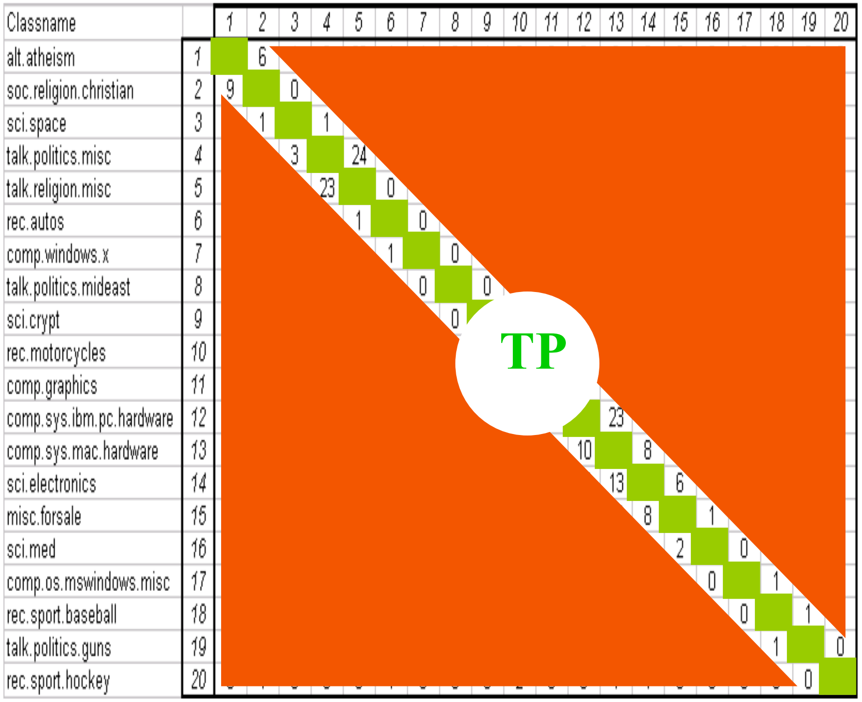
&#3;</BODY></TEXT></REUTERS>

Precision & Recall

Two class situation

		Predicted	
Actual		“P”	“N”
	P	TP	FN
	N	FP	TN

Multi-class situation:



Classname		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
alt.atheism	1		6																		
soc.religion.christian	2	9		0																	
sci.space	3		1		1																
talk.politics.misc	4			3		24															
talk.religion.misc	5				23		0														
rec.autos	6					1		0													
comp.windows.x	7						1		0												
talk.politics.mideast	8							0		0											
sci.crypt	9								0												
rec.motorcycles	10																				
comp.graphics	11																				
comp.sys.ibm.pc.hardware	12													23							
comp.sys.mac.hardware	13												10		8						
sci.electronics	14													13		6					
misc.forsale	15														8		1				
sci.med	16															2		0			
comp.os.mswindows.misc	17																0		1		
rec.sport.baseball	18																	0		1	
talk.politics.guns	19																		1		0
rec.sport.hockey	20																			0	

Precision = TP/(TP+FP)
Recall = TP/(TP+FN)
F-measure = 2pr/(p+r)

Micro-- vs. Macro--Averaging

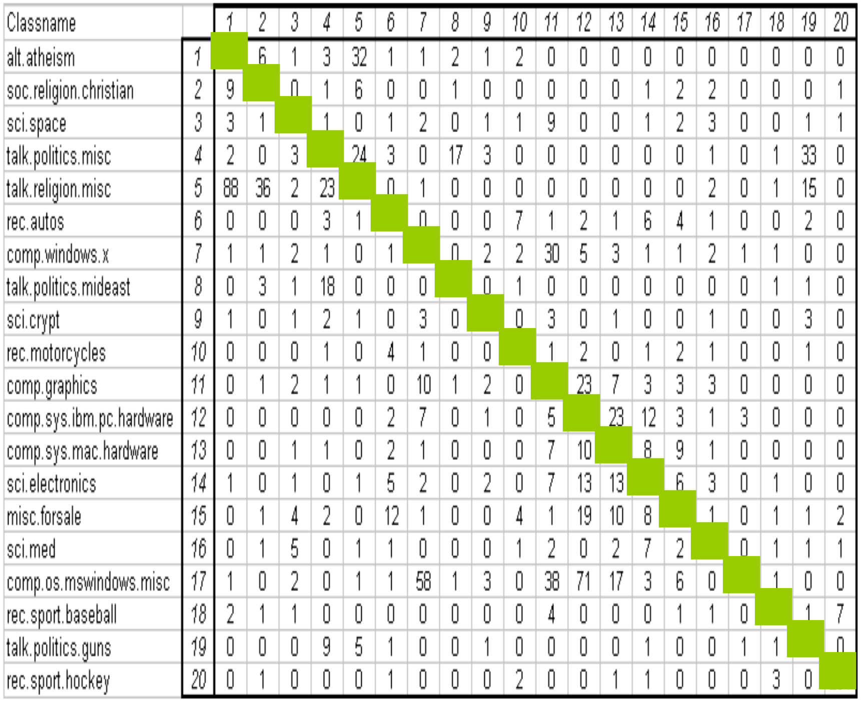
- If we have more than one class, how do we combine multiple performance measures into one quantity?

- Macroaveraging
  - Compute performance for each class, then average.

- Microaveraging
  - Collect decisions for all classes, compute contingency table, evaluate

Precision & Recall

Multi-class Multi-label situation:



Classname		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
alt.atheism	1	🟩	6	1	3	32	1	1	2	1	2	0	0	0	0	0	0	0	0	0	0
soc.religion.christian	2	9	🟩	0	1	6	0	0	1	0	0	0	0	0	1	2	2	0	0	0	1
sci.space	3	3	1	🟩	1	0	1	2	0	1	1	9	0	0	1	2	3	0	0	1	1
talk.politics.misc	4	2	0	3	🟩	24	3	0	17	3	0	0	0	0	0	0	1	0	1	33	0
talk.religion.misc	5	88	36	2	23	🟩	0	1	0	0	0	0	0	0	0	0	2	0	1	15	0
rec.autos	6	0	0	0	3	1	🟩	0	0	0	7	1	2	1	6	4	1	0	0	2	0
comp.windows.x	7	1	1	2	1	0	1	🟩	0	2	2	30	5	3	1	1	2	1	1	0	0
talk.politics.mideast	8	0	3	1	18	0	0	0	🟩	0	1	0	0	0	0	0	0	0	1	1	0
sci.crypt	9	1	0	1	2	1	0	3	0	🟩	0	3	0	1	0	0	1	0	0	3	0
rec.motorcycles	10	0	0	0	1	0	4	1	0	0	🟩	1	2	0	1	2	1	0	0	1	0
comp.graphics	11	0	1	2	1	1	0	10	1	2	0	🟩	23	7	3	3	3	0	0	0	0
comp.sys.ibm.pc.hardware	12	0	0	0	0	0	2	7	0	1	0	5	🟩	23	12	3	1	3	0	0	0
comp.sys.mac.hardware	13	0	0	1	1	0	2	1	0	0	0	7	10	🟩	8	9	1	0	0	0	0
sci.electronics	14	1	0	1	0	1	5	2	0	2	0	7	13	13	🟩	6	3	0	1	0	0
misc.forsale	15	0	1	4	2	0	12	1	0	0	4	1	19	10	8	🟩	1	0	1	1	2
sci.med	16	0	1	5	0	1	1	0	0	0	1	2	0	2	7	2	🟩	0	1	1	1
comp.os.mswindows.misc	17	1	0	2	0	1	1	58	1	3	0	38	71	17	3	6	0	🟩	1	0	0
rec.sport.baseball	18	2	1	1	0	0	0	0	0	0	0	4	0	0	0	1	1	0	🟩	1	7
talk.politics.guns	19	0	0	0	9	5	1	0	0	1	0	0	0	0	1	0	0	1	1	🟩	0
rec.sport.hockey	20	0	1	0	0	0	1	0	0	0	2	0	0	1	1	0	0	0	3	0	🟩

Aggregate

Average Macro Precision = $\Sigma p_i/N$

Average Macro Recall = $\Sigma r_i/N$

Average Macro F-measure = $2p_M r_M/(p_M+r_M)$

Average Micro Precision = $\Sigma TP_i/ \Sigma_i Col_i$

Average Micro Recall = $\Sigma TP_i/ \Sigma_i Row_i$

Average Micro F-measure = $2p_\mu r_\mu /(p_\mu+r_\mu)$

Precision(class i) = $TP_i/(TP_i+FP_i)$

Recall(class i) = $TP_i/(TP_i+FN_i)$

F-measure(class i) = $2p_i r_i/(p_i+r_i)$

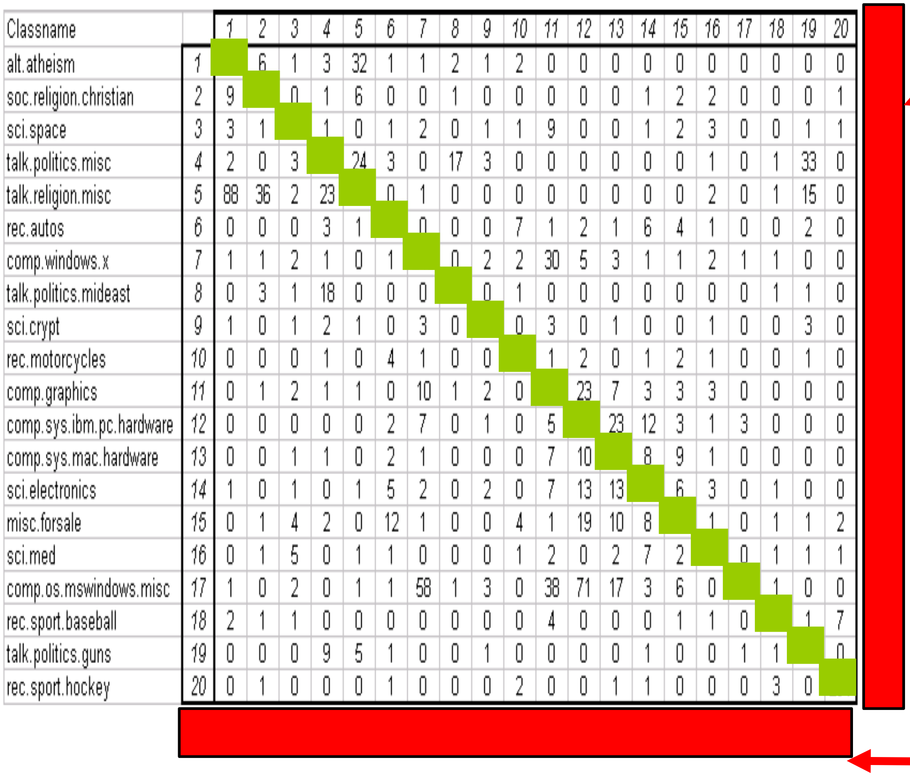
Precision(class 1) = $251/(Column_1)$

Recall(class 1) = $251/(Row_1)$

F-measure(class 1)) = $2p_i r_i/(p_i+r_i)$

Precision & Recall

Multi-class situation:



Classname		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
alt.atheism	1	🟩	6	1	3	32	1	1	2	1	2	0	0	0	0	0	0	0	0	0	0
soc.religion.christian	2	9	🟩	0	1	6	0	0	1	0	0	0	0	0	1	2	2	0	0	0	1
sci.space	3	3	1	🟩	1	0	1	2	0	1	1	9	0	0	1	2	3	0	0	1	1
talk.politics.misc	4	2	0	3	🟩	24	3	0	17	3	0	0	0	0	0	0	1	0	1	33	0
talk.religion.misc	5	88	36	2	23	🟩	0	1	0	0	0	0	0	0	0	0	2	0	1	15	0
rec.autos	6	0	0	0	3	1	🟩	0	0	0	7	1	2	1	6	4	1	0	0	2	0
comp.windows.x	7	1	1	2	1	0	1	🟩	0	2	2	30	5	3	1	1	2	1	1	0	0
talk.politics.mideast	8	0	3	1	18	0	0	0	🟩	0	1	0	0	0	0	0	0	0	1	1	0
sci.crypt	9	1	0	1	2	1	0	3	0	🟩	0	3	0	1	0	0	1	0	0	3	0
rec.motorcycles	10	0	0	0	1	0	4	1	0	0	🟩	1	2	0	1	2	1	0	0	1	0
comp.graphics	11	0	1	2	1	1	0	10	1	2	0	🟩	23	7	3	3	3	0	0	0	0
comp.sys.ibm.pc.hardware	12	0	0	0	0	0	2	7	0	1	0	5	🟩	23	12	3	1	3	0	0	0
comp.sys.mac.hardware	13	0	0	1	1	0	2	1	0	0	0	7	10	🟩	8	9	1	0	0	0	0
sci.electronics	14	1	0	1	0	1	5	2	0	2	0	7	13	13	🟩	6	3	0	1	0	0
misc.forsale	15	0	1	4	2	0	12	1	0	0	4	1	19	10	8	🟩	1	0	1	1	2
sci.med	16	0	1	5	0	1	1	0	0	0	1	2	0	2	7	2	🟩	0	1	1	1
comp.os.mswindows.misc	17	1	0	2	0	1	1	58	1	3	0	38	71	17	3	6	0	🟩	1	0	0
rec.sport.baseball	18	2	1	1	0	0	0	0	0	0	0	4	0	0	0	1	1	0	🟩	1	7
talk.politics.guns	19	0	0	0	9	5	1	0	0	1	0	0	0	0	1	0	0	1	1	🟩	0
rec.sport.hockey	20	0	1	0	0	0	1	0	0	0	2	0	0	1	1	0	0	0	3	0	🟩

Missed predictions

Classifier hallucinations

Aggregate

Average Macro Precision = $\Sigma p_i/N$

Average Macro Recall = $\Sigma r_i/N$

Average Macro F-measure = $2p_M r_M/(p_M+r_M)$

Average Micro Precision = $\Sigma TP_i / \Sigma_i Col_i$

Average Micro Recall = $\Sigma TP_i / \Sigma_i Row_i$

Average Micro F-measure = $2p_\mu r_\mu/(p_\mu+r_\mu)$

Aren't μ prec and μ recall the same?

Precision(class i) = $TP_i/(TP_i+FP_i)$

Recall(class i) = $TP_i/(TP_i+FN_i)$

F-measure(class i) = $2p_i r_i/(p_i+r_i)$

Precision(class 1) = $251/(\text{Column}_1)$

Recall(class 1) = $251/(\text{Row}_1)$

F-measure(class 1)) = $2p_i r_i/(p_i+r_i)$