

# Core Machine Learning Techniques for Information Retrieval

---

Luo Si

Purdue University

Rong Jin

Michigan State University

Acknowledgement: we appreciate contribution and helpful discussion  
from Prof. Yi Zhang, UCSC

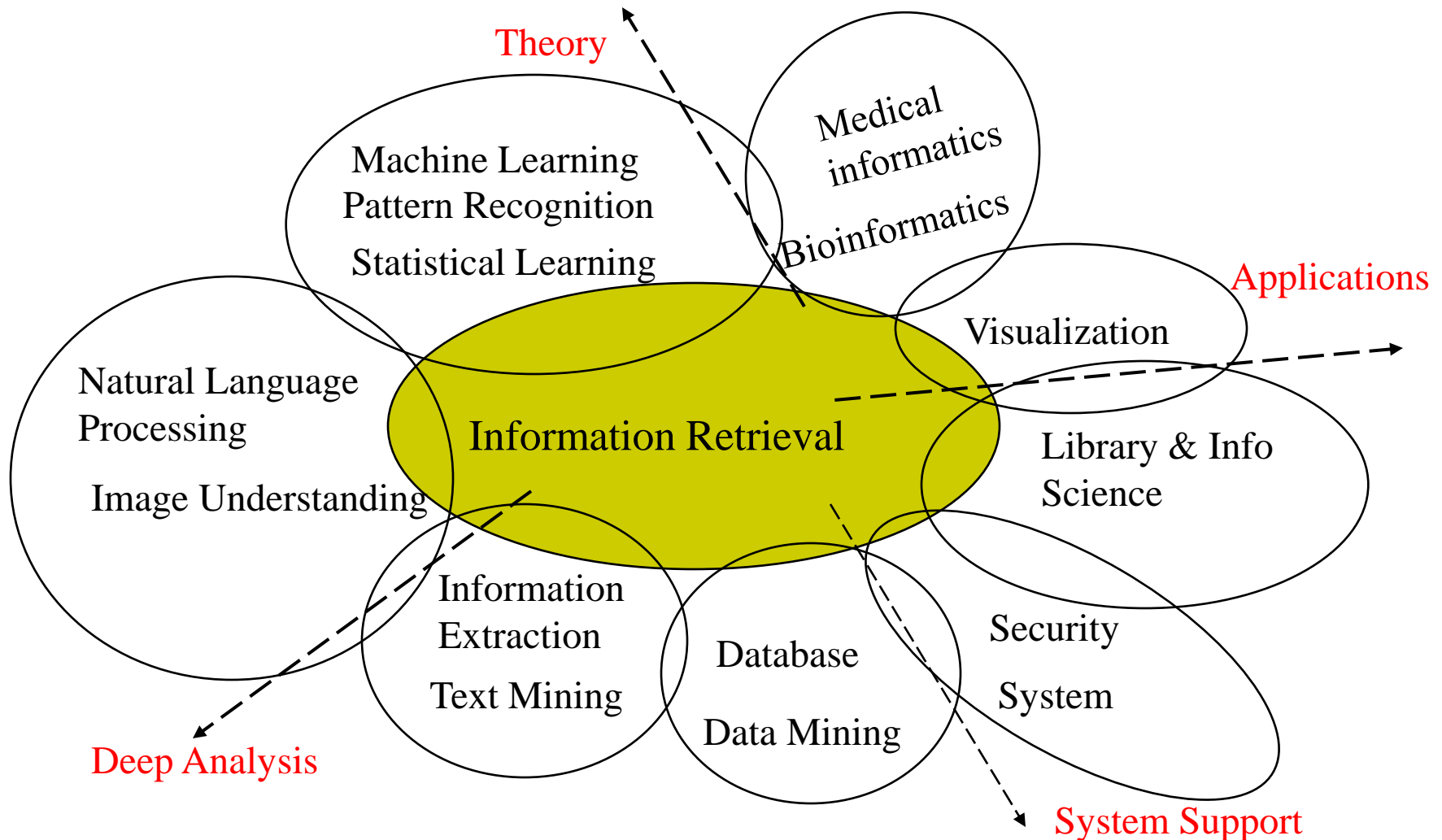


# Outline

---

- **Introduction to Core Concepts in Machine Learning**
  - Core machine learning concepts
  - Overview of learning techniques for IR applications
  - Basic Probability & Statistics
- Basic Optimization Methods
- Machine Learning Applications in IR

# IR and other disciplines





# Core Concepts in Machine Learning

---

- Machine Learning Approach
  - Problem analysis, input/output representation, hypothesis space, constraints (domain knowledge!)
  - Formulate as an optimization problem
  - Solve the problem with optimization methods

# Core Concepts in Machine Learning

---

## □ Data Representation

- Important for IR applications for effectiveness and efficiency
- Common text pre-processing methods (stop-word removing, stemming)
- Doc representation: bag of words against higher-order (e.g., bigram)
- Feature selection against regularization
- Go beyond text representation: page-rank value in search, percentage of capitalized letters for name entity recognition

# Core Concepts in Machine Learning

---

## □ Supervised Learning

- Given training examples with labels, learn a model to predict labels of test examples
- Classification (predict spam email or regular), regression (predict tomorrow's temperature), ...

## □ Unsupervised Learning

- Given training examples without labels, identify hidden structure to summarize/explain data
- Clustering, dimension reduction, density estimation in statistics

# Core Concepts in Machine Learning

---

## □ Generative Model

- Model both input and output data such as language model for IR, mostly in probabilistic models
- Work well with good domain knowledge/assumptions

## □ Discriminative Model

- Model output data given input data such as learning to rank methods
- Work well with enough labeled training data without making many assumptions

# Core Concepts in Machine Learning

---

## □ Bias and Variance

- Simple view: low bias means more accurate on training; low variance means robust performance on different training/test

## □ Trade-off: Bias and Variance

- Just remember training examples with labels; low bias and huge variance
- Always predict same thing: low variance and huge bias
- Need a trade-off between bias and variance



# Outline

---

- **Introduction to Core Concepts in Machine Learning**
  - Core machine learning concepts
  - Overview of learning techniques for IR applications
  - Basic Probability & Statistics
- Basic Optimization Methods
- Machine Learning Applications in IR

# Text Categorization

The image shows a screenshot of the Yahoo! Shopping website. At the top, there is a dark blue header with the "YAHOO! SHOPPING" logo on the left and a search bar on the right. Below the header, there are navigation tabs for "Home", "Clothing", "Electronics", "Computers", and "Home & Garden". A "Shop for:" dropdown menu is visible, followed by a yellow "SEARCH" button and a link to "All deals".

The main content area is divided into sections. The "Shopping Categories" section is organized into a grid:

- Clothing & Accessories**: Women's, Men's, Shoes
- Computers**: Laptops, Desktops, Software
- Electronics**: iPods, Cameras, TVs, Cell Phones
- DVDs, Music & Books**: All Movies, All Music, All Books
- Flowers & Gifts**: Gifts, Roses, Flowers & Plants
- Fragrances & Beauty**: Perfumes, Makeup, Skin Care
- Home & Garden**: Bed & Bath, Kitchen, Furniture, Tools
- Jewelry & Watches**: Diamonds, Engagement, Watches
- Sales & Deals**: Daily Deals, Coupons & Free Shipping
- Sports & Outdoors**: Fitness, Team Sports, Camping
- Toys & Baby**: Dolls, Car Seats, Board Games
- More Categories**: Autos, Grocery, Health, Video Games

Below the categories is a "4th of July Deals" section with a left and right arrow. It features four product images: a smartphone, a grill, a lounge chair, and an American flag.

- Key technologies: supervised learning, semi-supervised learning

# Text Categorization

- Open directory project
  - the largest human-edited directory of the Web
  - Manual classification
  - Over 4 million sites and 590 K categories

The screenshot shows the DMOZ website interface. At the top, there is a green header with the DMOZ logo and the text "open directory project". To the right of the header, it says "In partnership with AOL search". Below the header, there is a search bar with a "Search" button and a link to "advanced" search. The main content area is a grid of category links, each with a sub-link. The categories include: Arts (Movies, Television, Music...), Business (Jobs, Real Estate, Investing...), Computers (Internet, Software, Hardware...), Games (Video Games, RPGs, Gambling...), Health (Fitness, Medicine, Alternative...), Home (Family, Consumers, Cooking...), Kids and Teens (Arts, School Time, Teen Life...), News (Media, Newspapers, Weather...), Recreation (Travel, Food, Outdoors, Humor...), Reference (Maps, Education, Libraries...), Regional (US, Canada, UK, Europe...), Science (Biology, Psychology, Physics...), Shopping (Autos, Clothing, Gifts...), Society (People, Religion, Issues...), Sports (Baseball, Soccer, Basketball...), and World (Deutsch, Español, Français, Italiano, Japanese, Nederlands, Polska, Dansk, Svenska...). At the bottom, there is a "Become an Editor" button and the text "Help build the largest human-edited directory of the web". The footer contains the copyright information "Copyright © 1998-2007 Netscape" and a small green lizard logo.

4,830,584 sites - 75,151 editors - over 590,000 categories

# Text Categorization

## Medical Subject Headings (Categories)

1.  Anatomy [A]
2.  Organisms [B]
3.  Diseases [C]
4.  Chemicals and Drugs [D]
5.  Analytical, Diagnostic and Therapeutic Techniques and
6.  Psychiatry and Psychology [F]
  - o [Behavior and Behavior Mechanisms \[F01\]](#) +
  - o [Psychological Phenomena and Processes \[F02\]](#) +
  - o [Mental Disorders \[F03\]](#) +
  - o [Behavioral Disciplines and Activities \[F04\]](#) +
7.  Biological Sciences [G]

About 26,000  
descriptors in a  
twelve-level hierarchy



NCBI PubMed  
A service of the National Library of Medicine and the National Institutes of Health  
www.pubmed.gov

All Databases PubMed Nucleotide Protein Genome Structure

Search PubMed for Mutation of mutL in hereditary colon cancer Go Clear

I: [Papadopoulos N et al.](#) Mutation of a mutL homolog in...[PMID: 8128251]

PMID- 8128251

OWN - NLM

STAT- MEDLINE

DA - 19940413

TI - Mutation of a mutL homolog in hereditary colon cancer.

PG - 1625-9

AB - Some cases of hereditary nonpolyposis colorectal cancer (HNPCC) are due to alterations in a mutS-related mismatch repair gene. A search of a large database of expressed sequence tags derived from random complementary DNA clones revealed three additional human mismatch repair genes, all related to the bacterial mutL gene. One of these genes (hMLH1) resides on chromosome 3p21, within 1 centimorgan of markers previously linked to cancer susceptibility in HNPCC kindreds. Mutations of hMLH1 that would

MH - Amino Acid Sequence

MH - \*Chromosomes, Human, Pair 3

MH - Codon

MH - Colorectal Neoplasms, Hereditary Nonpolyposis/\*genetics

# Document Clustering

web news images maps blogs wikipedia jobs more »

**SIGIR** Search [advanced preferences](#)

clouds sources sites time remix

**All Results** (206)

- + Iraq (49)
- Provider (2)
- Advocacy Groups Are Mainly Formed To Promote Or Advocate (2)
- + Universität, Trier (5)
- + International forum for the presentation of new research results (6)
- + Sigir'08 (5)
- + Special Interest Group (7)
- + Workshop (32)
- ACM -Sigir 2010 Was Held At Unimail, Geneva (2)
- + International ACM (15)

[more](#) | [all clouds](#)

Top 206 results retrieved for the query **SIGIR** ([details](#))

**Sigil** [See more from Encyclopedia »](#)

Sigil A sign or seal for an occult entity. Sigils, especially those that are the marks of angels, deities, or demons, are often used on amulets and talismans. According to occultists, such signs are like the signatures of gods and other supernatural entities, and the inscribing of such sigils

Search Results

[sigir.com: The Leading Sigir Site on the Net](#)

**sigir.com** has been connecting our visitors with providers of Dating Ads, ... Check out the **sigir.com** ...

[www.sigir.com](#) - [cache] - Gigablast, Yahoo!

[About SIGIR](#)

About **SIGIR** The Office of the Special Inspector General for Iraq Reconstruction (**SIGIR** to the Coalition Provisional Authority Office of ... [www.sigir.mil/about/index.html](#) - Cache [www.sigir.mil/about/index.html](#) - [cache] - Bing, Additional Sources

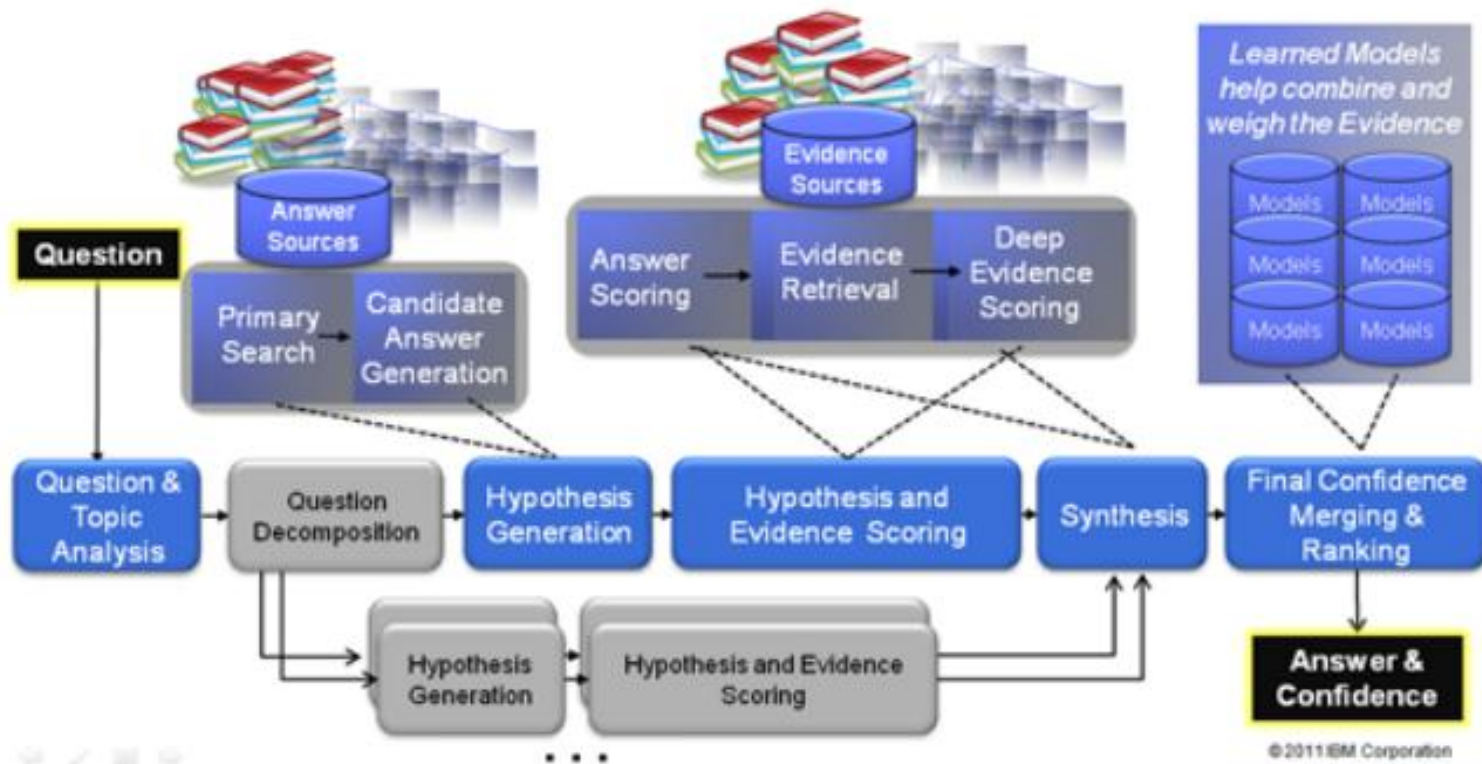
[Sigir 2007](#)

Advocacy groups are mainly formed to promote or advocate a certain thought, idea or project. There are different kinds of advocacy groups that we hear about with ...

[www.sigir2007.org](#) - [cache] - Additional Sources

- Key technologies: unsupervised learning

# Question Answering



- ❑ Classify question; identify answers; match questions and answers
- ❑ Key technologies: supervised learning, structure learning.

# Image Retrieval by Text Query

Image						
Annotation	WATER, PEOPLE, SWIMMERS, POOL	CARS, TRACKS, WALL, FORMULA	SKY, MOUNTAIN, CLOUDS, PARK	FIELD, HORSES, FOALS, MARE	SKY, PLANE, JET, CLOUDS	BIRDS, NEST, GRASS, TREE

- Automatically annotate images with textual words
- Retrieve images with textual queries
- Key technologies: classification
  - Each image is a visual document, each annotated keyword corresponds to a class

# Information Extraction

## Web page: free style text

J2EE Developer - US-MI-Dearborn .:

Reference #: 8040473	Industry: Automotive
Job Type: Contract	Job Length: 4 months
Salary: Commensurate with Experience	
Job Location: Dearborn, MI 48126	

Develop a JAVA/J2EE and Oracle solution for a Supplier Portal. Work will include enhancements to the security map application to enable supplier registration.

Required Skills: IBM Websphere 4.0.3 or 5.0; JAVA; PL/SQL.

Desired Skills: Sun Solaris; Oracle 8i; JavaScript; HTML; XML; LDAP; SDM experience or equivalent; Strong oral and written communication skills; Commitment to high quality deliverables and strong teamwork skills.

The TAC Automotive Group is an Equal Opportunity Employer.

## Relational DB

Title	J2EE Developer
Length	4 month
Salary	....
Location	
Reference	



- Key technologies: structure prediction (e.g., Hidden Markov Model and Markov Random Field)

# Citation/Link Analysis

**Toward General-Purpose Learning for Information Extraction (1998)** ([Make Corrections](#)) ([15 citations](#))  
Dayne Freitag  
Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics

View or download:  
[cmu.edu/afs/cs/project/...lingie.p](http://cmu.edu/afs/cs/project/...lingie.p)  
Cached: [PS.gz](#) [PS](#) [PDF](#) [Image](#) [Up](#)

From: [jprc.com/Artificial\\_Intel...inde](http://jprc.com/Artificial_Intel...inde)  
([Enter author homepages](#))

**CiteSeer** [Home/Search](#) [Bookmark](#) [Context](#) [Related](#)

([Enter summary](#))

Rate this article: 1 2 3 4  
[Comment on this article](#)

**Abstract:** Two trends are evident in the recent evolution of the field of information extraction: a preference for simple, often corpus-driven techniques over linguistically sophisticated ones; and a broadening of the central problem definition to include many non-traditional text domains. This deve calls for information extraction systems which are as retargetable and general as possible. Here, we describe SRV, a learning architecture for information extracti is designed for maximum... ([Update](#))

**Context of citations to this paper:** [More](#)

...multiple strategies for learning to extract text fields from Web pages. **We have developed a number of approaches to this task [18,19,21], including mul strategy learning [20] Integrate statistical bag of words methods into first order learning tasks.** We have begun...

...of information extraction was not addressed by Cohen. **In recent work Freitag proposes an ILP like formalism for information extraction [6], called SRV** informally describes the examples as a set of annotated documents. Without going into the details of his rule...

**Cited by:** [More](#)

Active Learning Selection Strategies for Information Extraction - Aidan Finn Nicholas (2003) ([Correct](#))

Extracting Information from Text - Bagga, Chai, Biermann ([Correct](#))

Learning for Text Categorization and Information Extraction ... - Junker, Sintek, Rinck (1999) ([Correct](#))

**Similar documents (at the sentence level):**

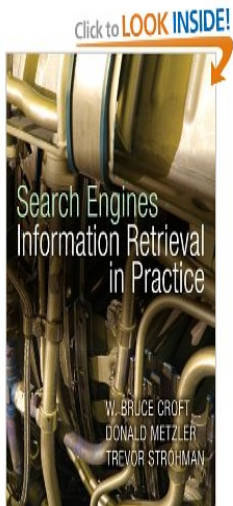
25.2%: [Toward General-Purpose Learning for Information Extraction - Freitag \(1998\)](#) ([Correct](#))

7.6%: [Information Extraction from HTML: Application of a General.. - Freitag \(1998\)](#) ([Correct](#))

6.0%: [Machine Learning for Information Extraction in Informal Domains - Freitag \(1998\)](#) ([Correct](#))

- Key technologies: semi-supervised learning, learning to rank

# Recommender Systems



## Search Engines: Information Retrieval in Practice [Hardcover]

Bruce Croft (Author), Donald Metzler (Author), Trevor Strohman (Author)

★★★★☆ (4 customer reviews) | Like (1)

List Price: \$105.00

Price: **\$82.68** & this item ships for **FREE with Super Saver Shipping**. [

You Save: \$22.32 (21%)

**In Stock.**

Ships from and sold by Amazon.com. Gift-wrap available.

Only 7 left in stock--order soon (more on the way).

Want it delivered **Wednesday, June 29**? Order it in the next 8 hours and 0 minutes, at checkout. [Details](#)

**14 new** from \$70.00 **28 used** from \$40.00



FREE Two-Day Shipping for Students. [Learn more](#)

## Customers Who Bought This Item Also Bought



Information Retrieval: Implementing and Evalu... by Stefan Buettcher

★★★★★ (1)



Introduction to Information Retrieval by Christopher D. Manning

★★★★★ (14)



Building Search Applications: Lucene, LingPipe,... by Manu Konchady

★★★★★ (1)



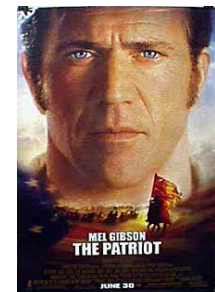
Search User Interfaces by Marti A. Hearst

★★★★★ (3)

\$51.50



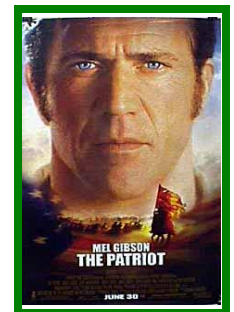
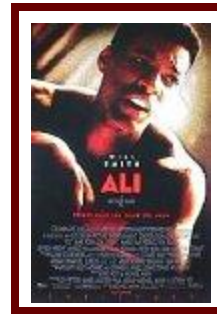
# Recommender Systems



User 1	?	5	3	4	2
User 2	4	1	5	?	5
User 3	5	?	4	2	5
User 4	1	5	3	5	?

- Sparse data problem: a lot of missing values

# Recommender Systems



User 1	?	5	3	4	2
User 2	4	1	5	?	5
User 3	5	?	4	2	5
User 4	1	5	3	5	?

- ❑ Sparse data problem: a lot of missing values
- ❑ Key technologies: clustering, (ordinal) regression, matrix completion



# Outline

---

- **Introduction to Core Concepts in Machine Learning**
  - Core machine learning concepts
  - Overview of learning techniques for IR applications
  - **Basic Probability & Statistics**
- Basic Optimization Methods
- Machine Learning Applications in IR

# Basics of Probability Theory

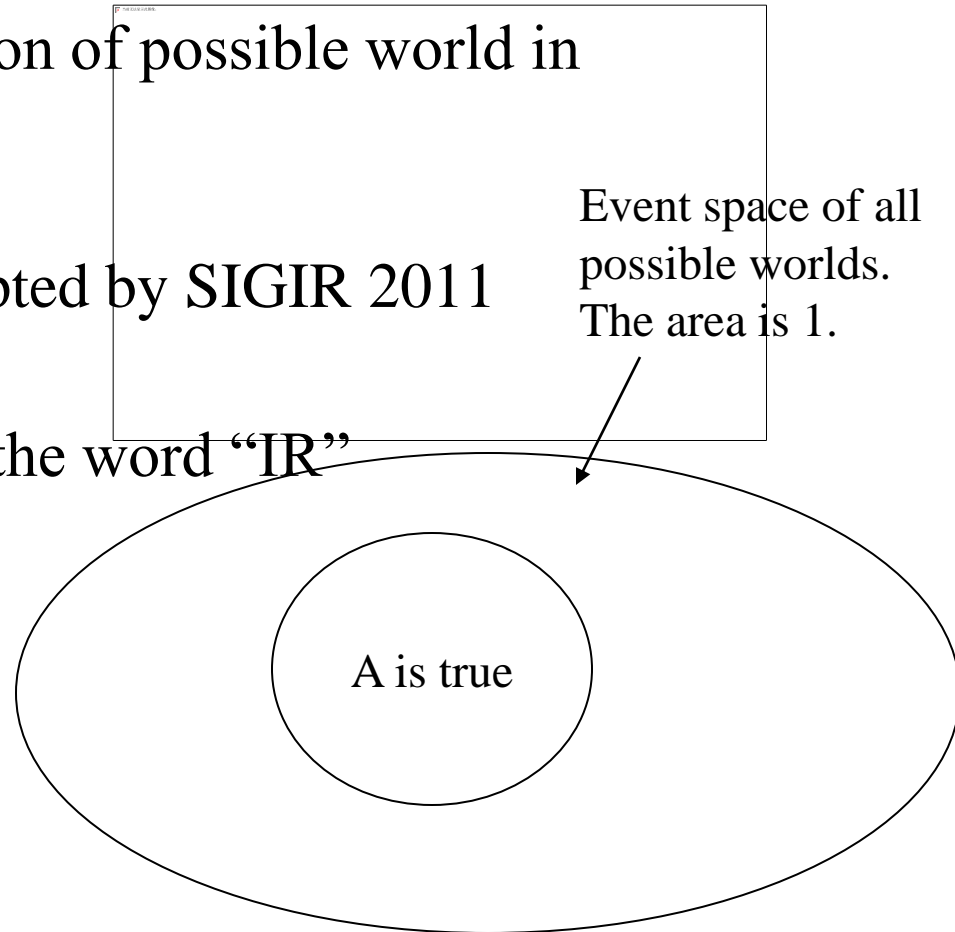
□ Probability  $\Pr(A)$ : “the fraction of possible world in which A is true”

■ Examples

A = Your paper will be accepted by SIGIR 2011

A = It is hot in Beijing

A = One document contains the word “IR”



# Conditional Probability

---

- SIGIR2011 = “a document contains the phrase SIGIR 2011”
- BEIJING = “a document contains the word Beijing”
  
- $P(\text{BEIJING}) = 0.001$
- $P(\text{SIGIR2011}) = 0.000000001$
- $P(\text{BEIJING}|\text{SIGIR2011}) = 1/2$

# Conditional Prob.

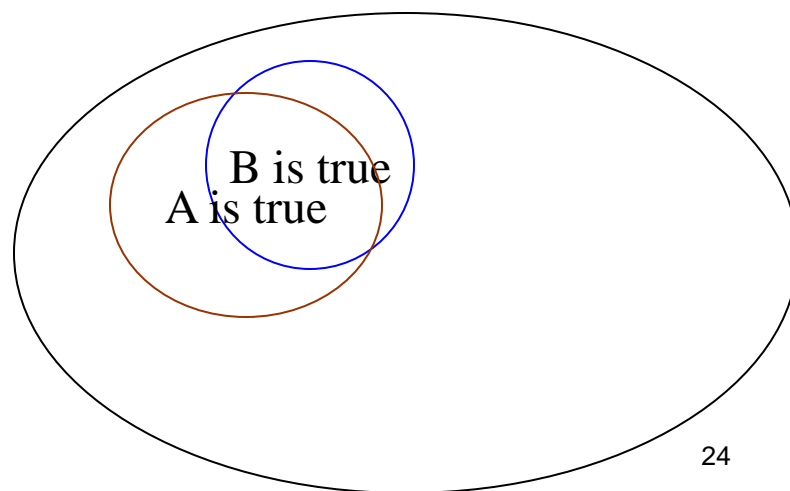
---

Definition

$$\Pr(A|B) = \frac{\Pr(A;B)}{\Pr(B)}$$

Chain rule

$$\Pr(A;B) = \Pr(A|B) \times \Pr(B)$$



# Conditional Prob.

---

Definition

$$\Pr(A|B) = \frac{\Pr(A;B)}{\Pr(B)}$$

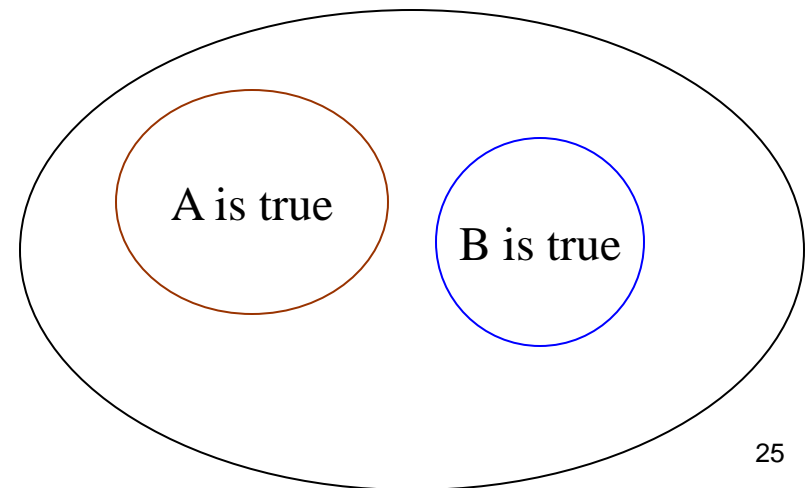
□ Independent variables

$$\Pr(A|B) = \Pr(A)$$

Chain rule

$$\Pr(A;B) = \Pr(A|B) \Pr(B)$$

$$\Pr(A, B) = \Pr(B) \Pr(A)$$



# Conditional Prob.

---

Definition

$$\Pr(A|B) = \frac{\Pr(A;B)}{\Pr(B)}$$

□ Independence

$$\Pr(A|B) = \Pr(A)$$

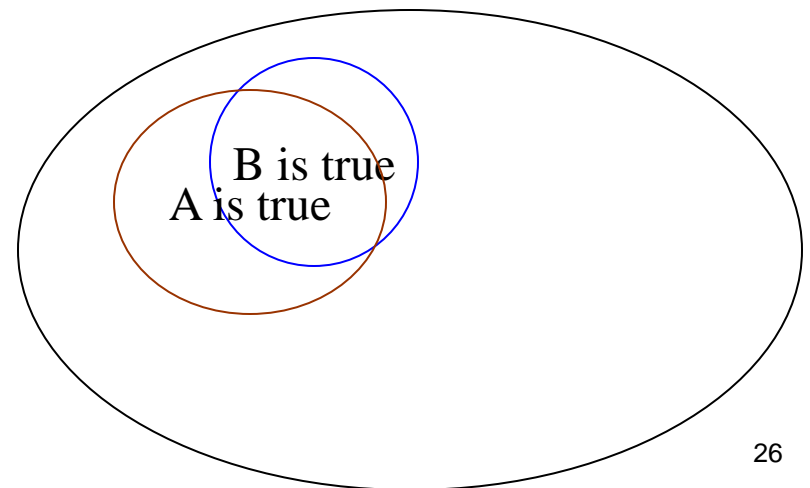
□ Marginal probability

$$\Pr(B) = \sum_{j=1}^n \Pr(B; A = a_j)$$

Chain rule

$$\Pr(A;B) = \Pr(A|B) \Pr(B)$$

$$\Pr(A, B) = \Pr(B) \Pr(A)$$



# Expectation & Variance

---

- **Expectation:** average outcome

$$E(x) = \sum_x xP(X = x)$$

- Examples: average outcome of a dice

$$1/6*1+1/6*2+1/6*3+1/6*4+1/6*5+1/6*6=21/6=3.5$$

- **Variance:** how diverse are the outcomes (deviation from expectation)

$$\text{Var}(x) = \sum_x (x - E(x))^2 P(X = x)$$

- Examples: average outcome of a coin (1 for head, 0 for tail)

$$1/2*(0-1/2)^2 + 1/2*(1-1/2)^2 = 1/2$$

# Common Probability Distribution

## Bernoulli/Binomial

- Model binary outcomes: side of a coin, whether a term appears in a document, whether an email is a spam...
  - Bernoulli: binary outcome (i.e., 0 or 1), with probability  $p$  to be 1

$$\Pr(X = x | p) = p^x (1 - p)^{1-x}; x \in \{0, 1\}; 0 < p < 1$$

**Expectation:**  $p$

**Variance:**  $p(1-p)$

- Binomial:  $n$  outcomes of a binary variable, the probability  $p$  to be 1, what is probability of outcome 1 appear  $x$  times

$$\Pr(X = x | n; p) = \binom{n}{x} p^x (1 - p)^{n-x}; x \in \{0, \dots, n\}; 0 < p < 1$$

**Expectation:**  $np$

**Variance:**  $np(1-p)$

# Common Probability Distribution

## Multinomial (Language Model)

□ Model multiple outcomes: side of a dice; word/topic in documents;

$n$  experiments of a variable with multiple outcomes  $(1, \dots, |V|)$ , with probability  $p_1$  to be outcome 1, ..., the probability  $p_{|V|}$  to be  $|V|$ ,

The probability of outcome 1 appears  $x_1$  times, ...  $|V|$  appears  $x_{|V|}$  times

$$\Pr(X_1 = x_1; \dots; X_{|V|} = x_{|V|}) = \frac{n!}{x_1! \dots x_{|V|}!} p_1^{x_1} \dots p_{|V|}^{x_{|V|}}; \quad \sum_{v=1}^{|V|} x_v = n;$$

$$0 \leq x_v \leq n; \quad \sum_{v=1}^{|V|} p_v = 1$$

**Expectation:**  $E(X_v) = np_v$

**Variance:**  $\text{Var}(X_v) = np_v(1-p_v)$

# Common Probability Distribution

## Multinomial (Language Model)

---

### □ Examples:

Three words in vocabulary (**s**port, **b**asketball, **f**inance) with probabilities as  $(p_s=0.5, p_b=0.4, p_f=0.1)$

A document generated by this model contains 10 words

Questions:

What is the expectation of occurrences of word “sport”?

$$10 * 0.5 = 5$$

What is the probability of generating 5 “sport”, 3 “basketball” and 2 “finance”

$$\frac{10!}{5!3!2!} 0.5^5 0.4^3 0.1^2$$

# Common Probability Distribution

## Gaussian (Normal)

---

□ Model continuous distribution: draw data points close to a specific point

Gaussian distribution: sample points close (measured by  $\sigma$ ) to a point  $\mu$ .

$$\Pr(X = x_j; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x_j - \mu)^2}{2\sigma^2}\right)$$

**Expectation:**  $E(X) = \mu$

**Variance:**  $\text{Var}(X) = \sigma^2$

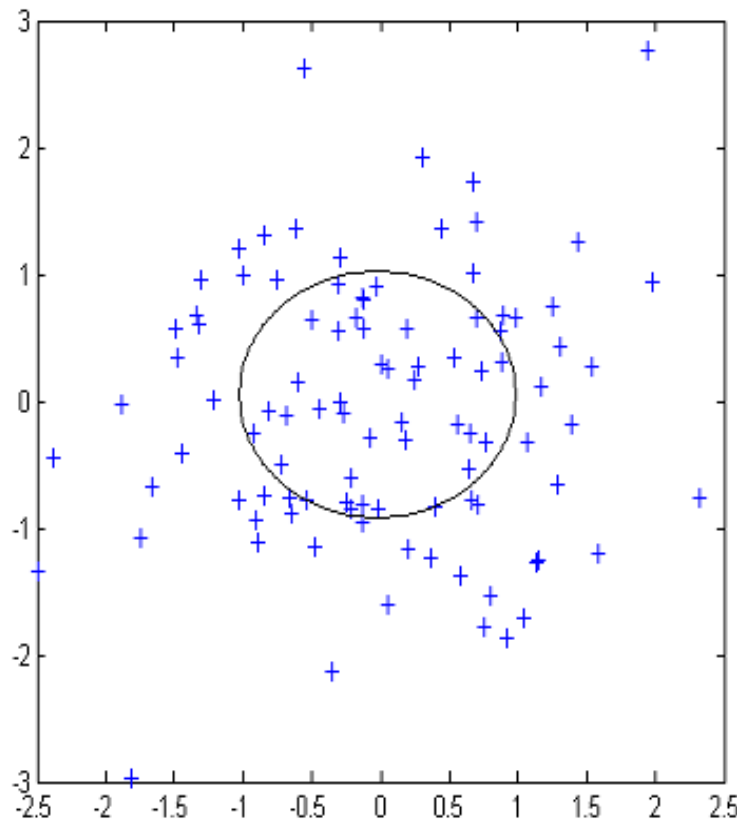
Expectation and variance can be vectors: multivariate Gaussian

# Common Probability Distribution

## Gaussian

- Example

- Gaussian (Normal) distribution with  $\mu = [0 \ 0]$ ,  $\Sigma = [1 \ 0; 0 \ 1]$ ; 100 data points '+' randomly generated by the model



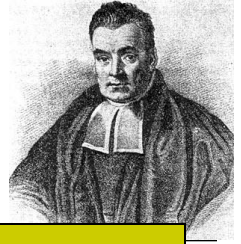
# Probability Distributions

---

- Binomial distributions
- Beta distribution
- Multinomial distributions
- Dirichlet distribution
- Gaussian distributions

Prior/Smoothing LM

# Bayes' Rule

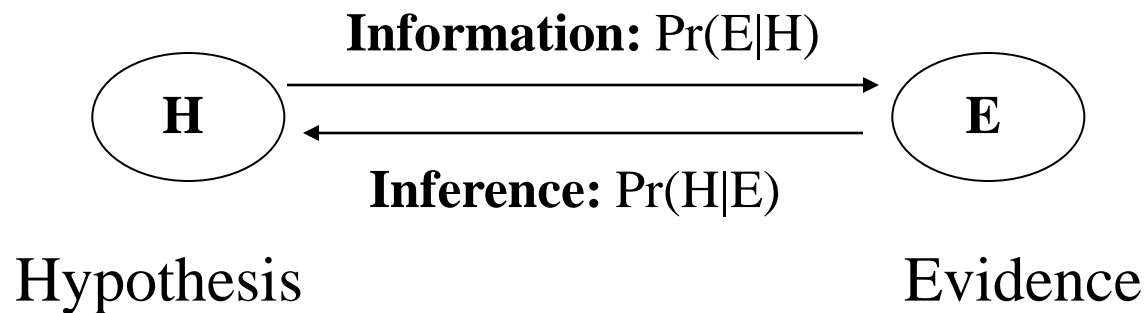


Posterior

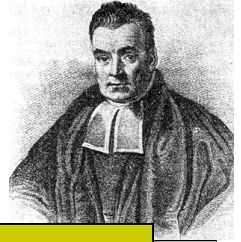
Prior

Likelihood

$$\Pr(H|E) / \Pr(H) \Pr(E|H)$$



# Bayes' Rule

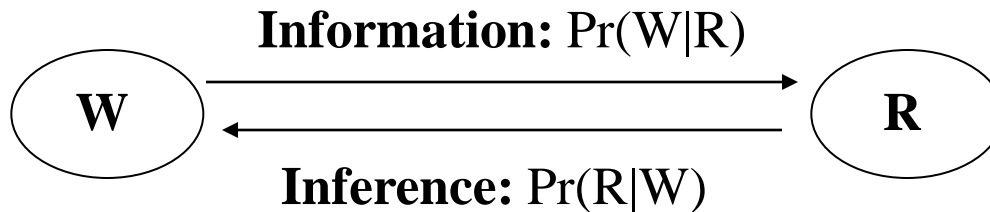


Posterior

Prior

Likelihood

$$\Pr(H|E) / \Pr(H) \Pr(E|H)$$



$\Pr(* *)$	<b>R</b>	$\neg$ <b>R</b>
<b>W</b>	0.7	0.4
$\neg$ <b>W</b>	0.3	0.6

**R**: It rains

**W**: The grass is wet

# Statistical Inference

Posterior

Prior

Likelihood

$$\Pr(H|E) / \Pr(H) \Pr(E|H)$$

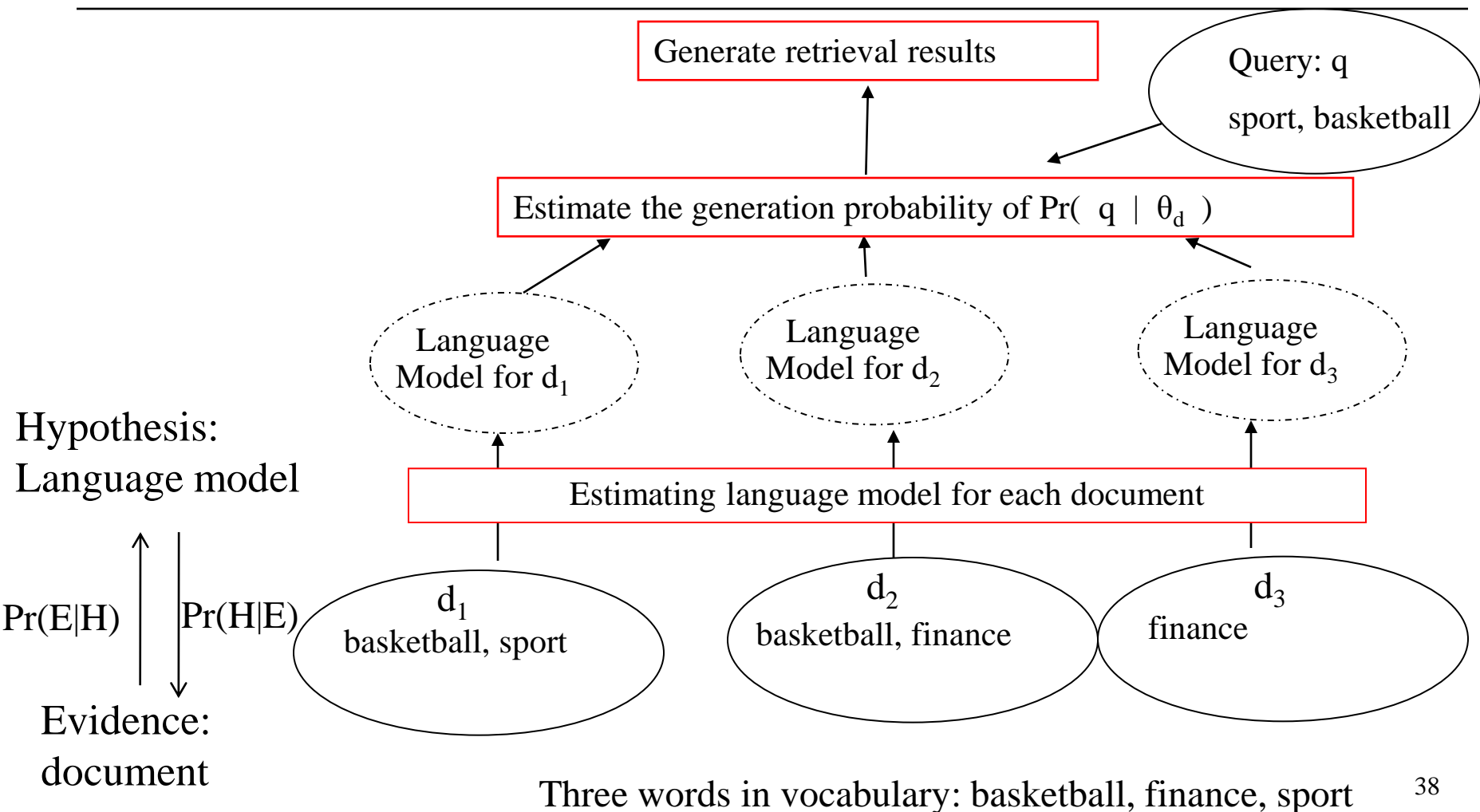
- Learning stage: a parametric model for  $\Pr(E|H)$
- Inference stage: for a given observation  $E$ 
  - Compute  $\Pr(H|E)$  for each hypothesis  $H$
  - Choose the hypothesis with the largest  $\Pr(H|E)$

# Maximum Likelihood Estimation vs. Maximum A Posterior Estimation

---

- Maximum Likelihood Estimation Inference
  - Compute  $\Pr(E|H)$  for each hypothesis  $H$ , ignore prior
  - Choose the hypothesis with the largest  $\Pr(E|H)$
  - May overfit data
  
- Maximum A Posterior Estimation
  - Compute  $\Pr(H|E)$  for each hypothesis  $H$  with prior
  - Use domain knowledge or data from large collection in prior
    - Conjugate prior: enable easy calculation, Dirichlet as conjugate prior for multinomial

# Language Model for IR: Example





# Outline

---

- Introduction to Core Concepts in Machine Learning
- **Basic Optimization Methods**
- Machine Learning Applications in IR

# Introduction to Optimization

---

## □ Optimization

The mathematical discipline for finding maxima and minima of functions, possibly subject to constraints.

Example we have seen:

$$\mu_d^{\alpha} = \operatorname{argmax}_{\mu_d} P(d|\mu_d) = \prod_{v=1}^{|Q|} p_v^{C(w_v;d)}$$

# Basic Optimization Methods

## □ Calculate analytic solution

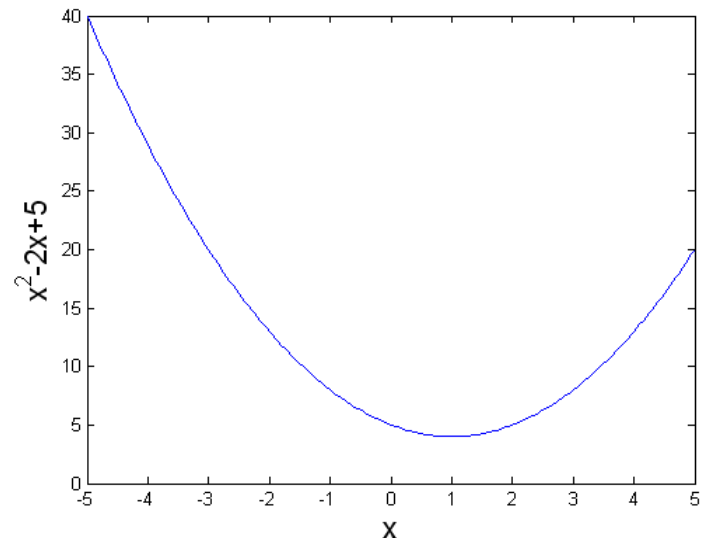
- Calculate the first derivative (with Lagrange multiplier when subjected to constraints)
- Set the above equation to 0 and try to solve the equation
- Check second derivative for positive (minimum) or negative (maximum)

Example:

$$x^* = \arg \min_x f(x) = \arg \min_x (x^2 - 2x + 5)$$

$$f(x)' = 2x - 2 = 0 \Rightarrow x^* = 1$$

$$f(x^*)'' = 2 > 0 \Rightarrow \text{It is minimum}$$

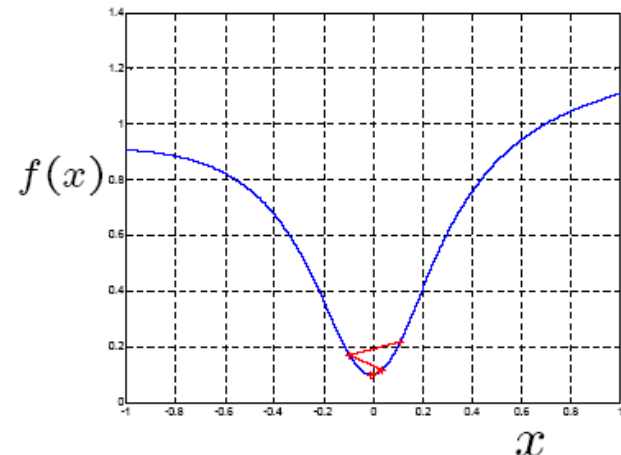


# Basic Optimization Methods

- Optimization with iterative gradient descent method
  - Many optimization problems do not have analytic solutions
  - Iterative method refines solution step by step
  
- Newton method uses information of first derivative and second derivative (exact or approximated) to refine solution

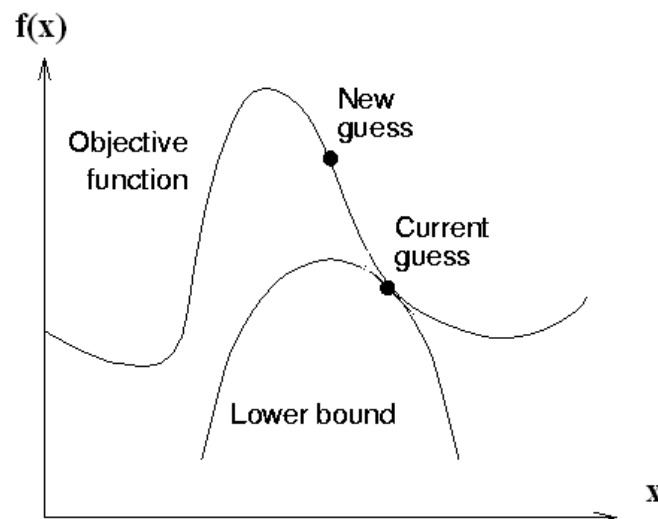
$$x^{(t+1)} = x^{(t)} - \frac{f'(x^{(t)})}{f''(x^{(t)})}$$

New updated solution      Old solution



# Basic Optimization Methods

- Newton method does not automatically guarantee improvement of new solution over old one
- Expectation and Maximization method  
Lower bound method, always make improvement; more elegant, often have good probabilistic interpolation



# Basic Optimization Methods

---

- Examples of Expectation & Maximization:
  - Given two biased dices A and B with known  $(P_A(1), \dots, P_A(6))$  and  $(P_B(1), \dots, P_B(6))$ . Each time, with probability  $\lambda$  draw A, and with probability  $1 - \lambda$  draw B.
  - Observe a sequence  $X = \{x_1, \dots, x_n\}$  and want to estimate  $\lambda^*$
  
- Intuition of Expectation and Maximization
  - If we know which dice is used in each time,  $\lambda^*$  is ratio of times A is used. But the information is hidden
  - **Expectation:** Given current model, estimate posterior probability of which dice is used in each time
  - **Maximization:** Given the posterior probabilities, update model parameter

# Basic Optimization Methods

---

- Expectation: Given current model, estimate posterior probability of which dice is used in each time

$$F_{Ai} = \frac{\lambda^{(t)} P_A(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)} \quad F_{Bi} = \frac{(1 - \lambda^{(t)}) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)} \quad \text{st. } F_{Ai} + F_{Bi} = 1$$

- Maximization: Given the posterior probabilities, update model parameter

$$\lambda^{(t+1)} = \frac{\sum_{i=1}^n F_{Ai}}{n}$$

- Iterate with the above two steps until converge

# Basic Optimization Methods

---

$$\lambda^* = \arg \max_{\lambda} l(X, \lambda)$$

$$= \arg \max_{\lambda} \sum_{i=1}^n \left( \log \left( \lambda P_A(x_i) + (1 - \lambda) P_B(x_i) \right) \right)$$

$$\lambda^* = \arg \max_{\lambda} l(X, \lambda) = \arg \max_{\lambda} \left[ l(X, \lambda) - l(X, \lambda^{(t)}) \right]$$

$$= \arg \max_{\lambda} \sum_{i=1}^n \left( \log \left[ \frac{\lambda P_A(x_i) + (1 - \lambda) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)} \right] \right)$$

$$= \arg \max_{\lambda} \sum_{i=1}^n \left( \log \left[ \frac{\frac{\lambda^{(t)} P_A(x_i)}{\lambda^{(t)} P_A(x_i)} \lambda P_A(x_i) + \frac{(1 - \lambda^{(t)}) P_B(x_i)}{(1 - \lambda^{(t)}) P_B(x_i)} (1 - \lambda) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)} \right] \right)$$

# Basic Optimization Methods

$$\lambda^* = \arg \max_{\lambda} l(X, \lambda) = \arg \max_{\lambda} [l(X, \lambda) - l(X, \lambda^{(t)})]$$

$$= \arg \max_{\lambda} \sum_{i=1}^n \left( \log \left[ \frac{\lambda P_A(x_i) + (1-\lambda)P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1-\lambda^{(t)})P_B(x_i)} \right] \right)$$

$$= \arg \max_{\lambda} \sum_{i=1}^n \left( \log \left[ \frac{\frac{\lambda^{(t)} P_A(x_i)}{\lambda^{(t)} P_A(x_i)} \lambda P_A(x_i) + \frac{(1-\lambda^{(t)}) P_B(x_i)}{(1-\lambda^{(t)}) P_B(x_i)} (1-\lambda) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1-\lambda^{(t)}) P_B(x_i)} \right] \right)$$

Set  $F_{Ai} = \frac{\lambda^{(t)} P_A(x_i)}{\lambda^{(t)} P_A(x_i) + (1-\lambda^{(t)}) P_B(x_i)}$      $F_{Bi} = \frac{(1-\lambda^{(t)}) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1-\lambda^{(t)}) P_B(x_i)}$     *st.*  $F_{Ai} + F_{Bi} = 1$

$$\geq \sum_{i=1}^n (F_{Ai} \log[\lambda P_A(x_i)] + F_{Bi} \log[(1-\lambda)P_B(x_i)]) + Const$$

Logarithm function is concave (Jensen Inequality)

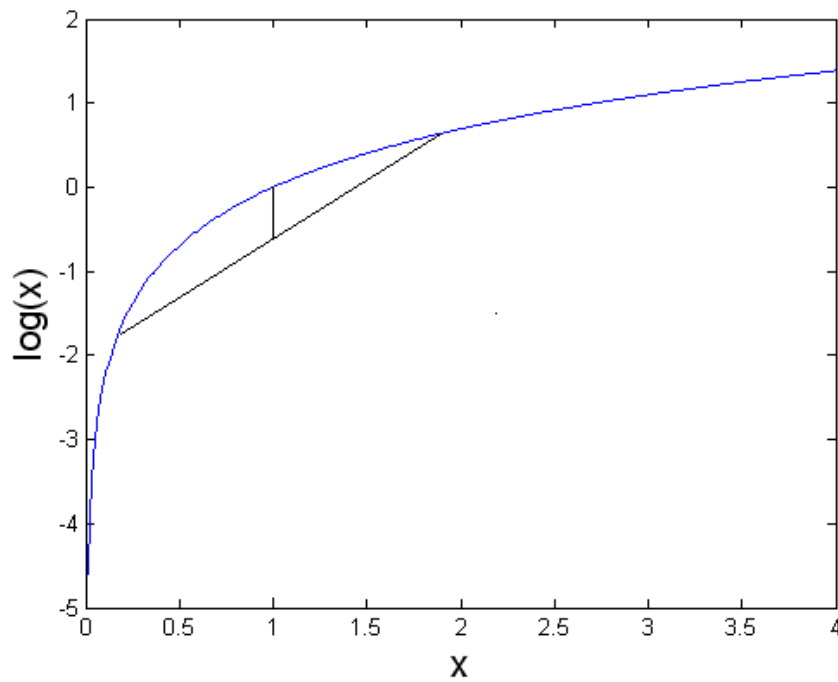
# Basic Optimization Methods

Set  $F_{Ai} = \frac{\lambda^{(t)} P_A(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)}$   $F_{Bi} = \frac{(1 - \lambda^{(t)}) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)}$  *st.*  $F_{Ai} + F_{Bi} = 1$

$$\sum_{i=1}^n \log \left[ \frac{\frac{\lambda^{(t)} P_A(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)} \lambda P_A(x_i) + \frac{(1 - \lambda^{(t)}) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)} (1 - \lambda) P_B(x_i)}{\lambda^{(t)} P_A(x_i) + (1 - \lambda^{(t)}) P_B(x_i)} \right]$$

$$\geq \sum_{i=1}^n (F_{Ai} \log[\lambda P_A(x_i)] + F_{Bi} \log[(1 - \lambda) P_B(x_i)]) + Const$$

Logarithm function is concave (Jensen Inequality)



# Basic Optimization Methods

---

Current solution:  $\lambda^{(t+1)}$  maximizes derived lower bound

$$\lambda^{(t+1)} = \arg \max_{\lambda} g(\lambda) = \arg \max_{\lambda} \sum_{i=1}^n \left( F_{Ai} \log[\lambda P_A(x_i)] + F_{Bi} \log[(1-\lambda)P_B(x_i)] \right)$$

$$g(\lambda)' = \sum_{i=1}^n \left( \frac{F_{Ai}}{\lambda} - \frac{F_{Bi}}{(1-\lambda)} \right) = 0 \quad \Rightarrow \quad \lambda^{(t+1)} = \frac{\sum_{i=1}^n F_{Ai}}{n}$$

# Outline

---

- Introduction to Core Concepts in Machine Learning
- Basic Optimization Methods
- Machine Learning Applications in IR
  - Text categorization
  - Text clustering
  - Collaborative filtering
  - Learning to rank



# Text Categorization (TC): Introduction

---

- Tasks:
  - **Given** some training documents/objects **labeled** with predefined categories
  - **Predict** best categories(s) for **unlabeled** (test) documents/objects
  
- Applications
  - Webpage/document classification
  - Automatic email sorting (spam detection; into different folders)
  - Word sense disambiguation (Java programming vs. Java in Indonesia)
  - Your favorite applications?...

# TC: Learning Technologies

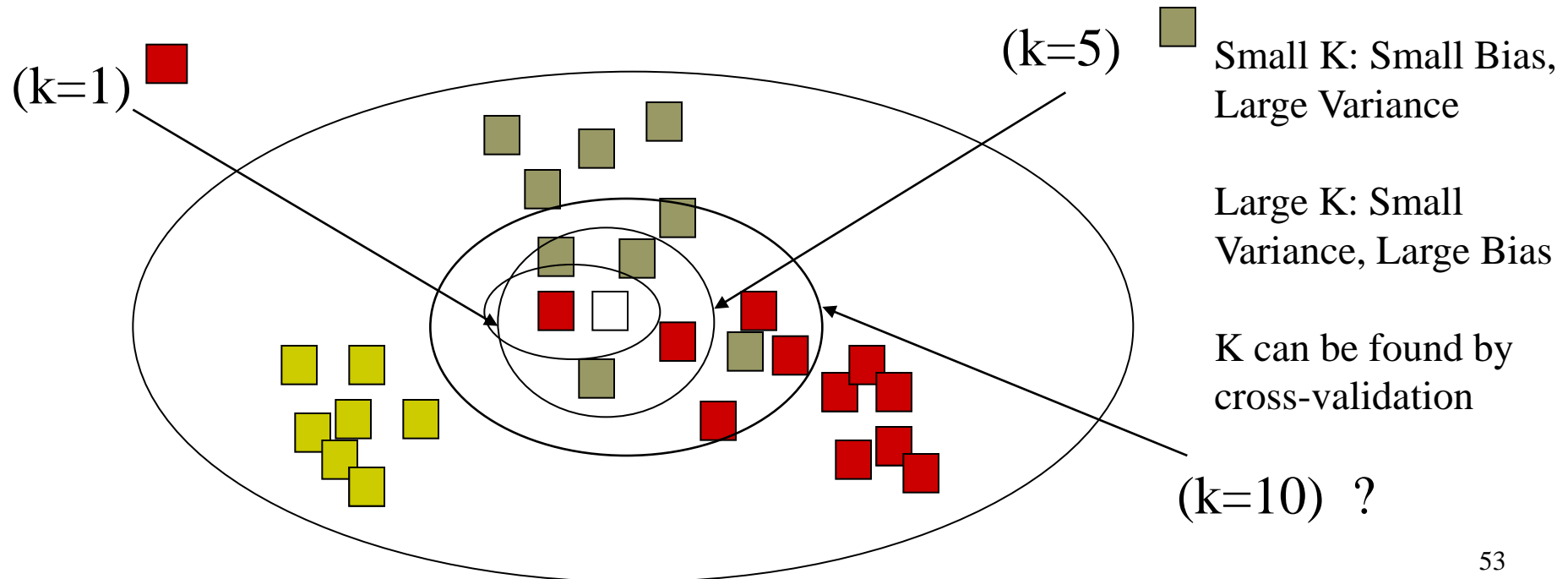
---

- **Nearest neighbor methods** (Yang 1994)
- **Naïve Bayes** (Language Model) (Lewis 1992; McCallum 1998)
- **Regression method** (Fuhr 1991; Yang 1992)
- **Support Vector Machines** (Joachims et al. 98, 05; Hofmann 03)
- Boosting or Bagging (Schapire, Singer, et al. 1998)
- Neural networks (Wiener, Pederson, et al. 1995)
- .....

# TC: K-Nearest Neighbor Classifier

- Also called “Instance-based learning” or “lazy learning”
  - low/no cost in “training”, high cost in online prediction

Idea: find your language by what language your neighbors speak;



# TC: K-Nearest Neighbor Classifier

---

- Often use tf.idf or BM25 document representation and Euclidean distance.
- Theoretical error bound analyzed by Duda & Hart (2000) & Devroye et al (1996). When  $n \rightarrow \infty$  (#docs),  $k \rightarrow \infty$  (#neighbors) and  $k/n \rightarrow \infty$  (ratio of neighbors and total docs), KNN approaches Bayes error.

# TC: Naïve Bayes Classification

---

- Naïve Bayes (NB) Classification (Lewis, 1992)(McCallum, 1998)
  - Generative Model: Model both the input data (i.e., document contents) and output data (i.e., class labels)
  - Make strong assumption of the probabilistic modeling approach
  
- Methodology
  - Similar with the idea of language modeling approaches for information retrieval
  - Train a language model for all the documents in one category

# TC: Naïve Bayes Classification

## □ Methodology

Train a model for all the documents in each category

- Estimate  $P(c)$  for each category as the relative frequency:

$$P(c) = \frac{c_L}{\#Total\_docs} \leftarrow \begin{array}{l} \text{Num of docs in} \\ \text{category } c \end{array}$$

- Estimate multinomial/language model for each category for maximizing generation probability:

$$P^*(w_v | c) = \frac{\sum_{i=1}^{c_L} C(w_v, d_i)}{\sum_{i=1}^{c_L} |d_i|} \leftarrow \begin{array}{l} \text{Count Statistics} \\ \text{Doc length} \end{array}$$

Normalization by count statistics

# TC: Naïve Bayes Classification

## □ Methodology

For each test document, make prediction as

$$c^* = \arg \max_c P(c | d) = \arg \max_c P(c) P(d | c) \text{ Bayes Rule}$$

$$P(d | c) \propto \prod_w \underbrace{P(w | c)}_{\text{Category Model}}^{c(w,d)} \leftarrow \text{Count Statistics}$$

Category Model

$$c^* = \arg \max_c \left\{ P(c) \prod_v P(w_v | c)^{C(w_v, d)} \right\} = \arg \max_c \left\{ \log(P(c)) + \sum_v \log P(w_v | c) C(w_v, d) \right\}$$

Model Parameters: learned by Naïve Bayes for maximizing generation probability, may not maximize classification accuracy

# TC: Logistic Regression Classification

- Directly model category/class of a given document (e.g., for a binary classification problem) :

$$\log \frac{P(c_+ | d)}{P(c_- | d)} = \beta(0) + \sum_v \beta(v) \times C(w_v, d)$$

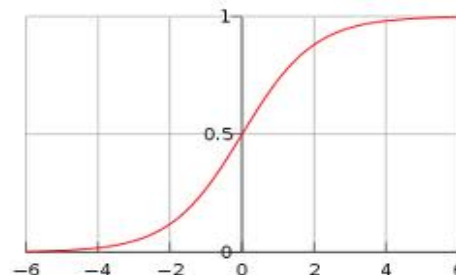
Count Statistics  $\nearrow$

Model Parameters to estimate

$$P(c_+ | d) = \frac{\exp\left(\beta(0) + \sum_v \beta(v) \times C(w_v, d)\right)}{1 + \exp\left(\beta(0) + \sum_v \beta(v) \times C(w_v, d)\right)}$$

Sigmoid/logistic function:

$$\sigma\left(\beta(0) + \sum_v \beta(v) \times C(w_v, d)\right)$$



# TC: Logistic Regression Classification

---

- Model parameter estimation by Maximum Likelihood Estimation:
  - Find model parameters for a category that maximizes conditional classification probability

$$\begin{aligned}\beta^* &= \arg \max_{\beta} \sum_i \left[ \delta(d_i, c_+) \log(P(c_+ | d_i)) + (1 - \delta(d_i, c_+)) \log[1 - P(c_+ | d_i)] \right] \\ &= \arg \max_{\beta} \sum_i \left[ \log \left( \sigma \left( y_i \left( \beta(0) + \sum_v \beta(v) \times C(w_v, d_i) \right) \right) \right) \right]\end{aligned}$$

$y_i=1$  iff the document is in category  $c_+$      $y_i=-1$  iff the document is in category  $c_-$



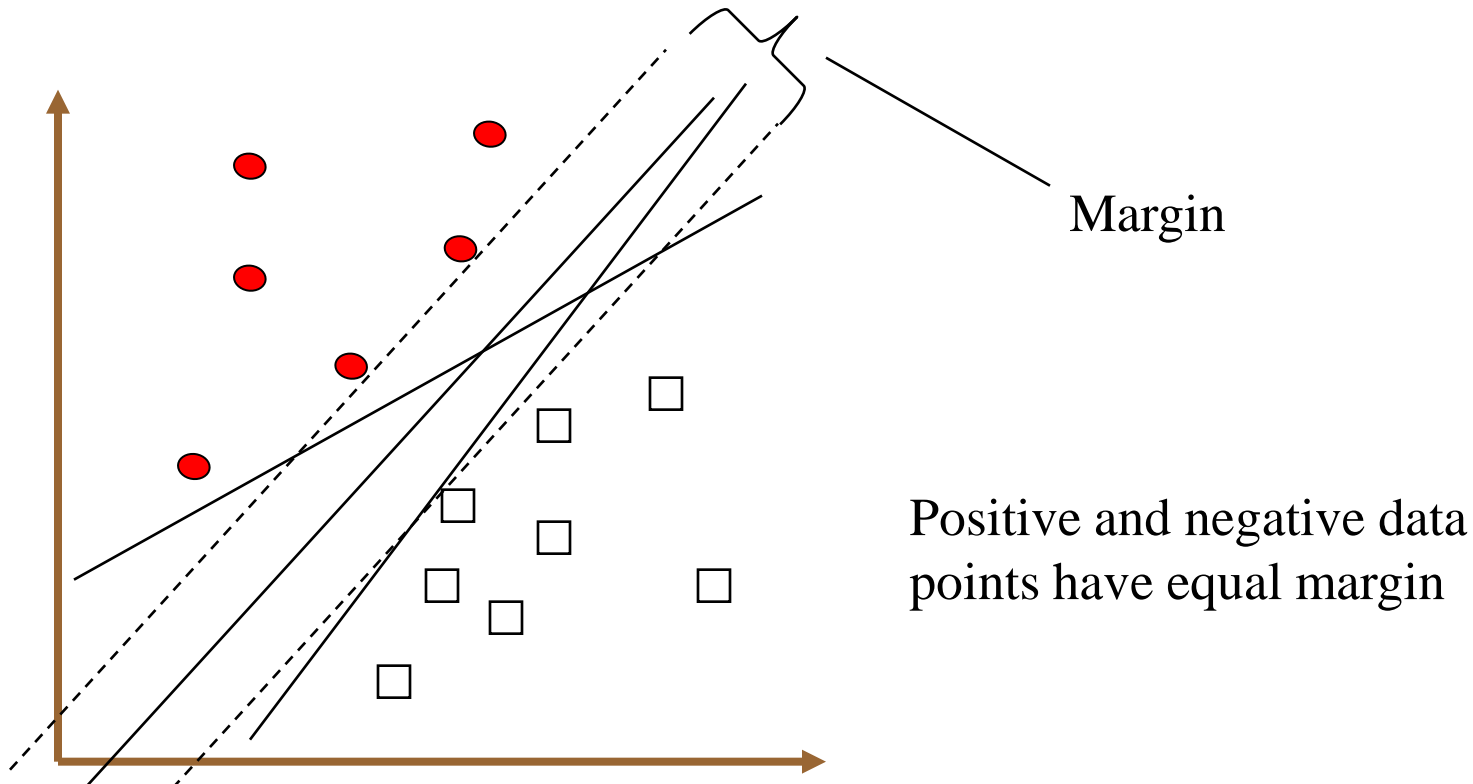
# TC: Logistic Regression Classification

---

- Many optimization methods for solving the above problem
  - Newton method, use first and second derivatives; Quasi-Newton method (BFGS) use first derivative to approximate second
  - Iterative Scaling (similar to EM as a low-bound method)

# TC: Support Vector Machine

- Support Vector Machine (SVM) (Joachims, 1998)
  - Build a classifier that separates data points in an accurate and robust manner with large margin.



# TC: Support Vector Machine

---

## □ Linear SVM

- Optimization problem for model parameters ( $\beta = \{\beta(v)\}$ ) associated with features (words) in input instances (e.g., each  $x_i = \{C(w_v, d_i)\}$ ) for predicting outputs ( $\{y_i\}$  binary classification)

Introduction “slack variables”, slack variables are always positive

$$\begin{cases} \beta^T x_i + b \geq 1 - \xi_i & y_i = 1 \\ \beta^T x_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 \end{cases}$$

Introduce const C to balance error for linear boundary and the margin

$$\frac{1}{2} \|\beta\|^2 + C \sum_i \xi_i$$

# TC: Support Vector Machine

---

Margin  $\rightarrow$  Trade-off Parameter  $\leftarrow$  Error for each instance

$$\max_w \frac{1}{2} \|\beta\|^2 + C \sum_i \xi_i$$

*subject to* :  $y_i (\beta^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$

- Error can be reformulated as  $\xi_i = \max(0, 1 - y_i(\beta^T x_i + b))$ , called hinge loss, an upper bound as classification error as  $I(y_i(\beta^T x_i + b) \leq 0)$
- Often solved in dual-form that puts a weight on each  $x_i$ , which often enables a kernel representation that can model high-dimensional data representation

# TC: Evaluation

---

Contingency Table Per Category (for all docs)

	Truth: True	Truth: False	
Predicted Positive	a	b	a+b
Predicted Negative	c	d	c+d
	a+c	b+d	n=a+b+c+d

a: number of truly positive docs

c: number of false negative docs

n: total number of test documents

b: number of false-positive docs

d: number of truly-negative docs

# TC: Evaluation

---

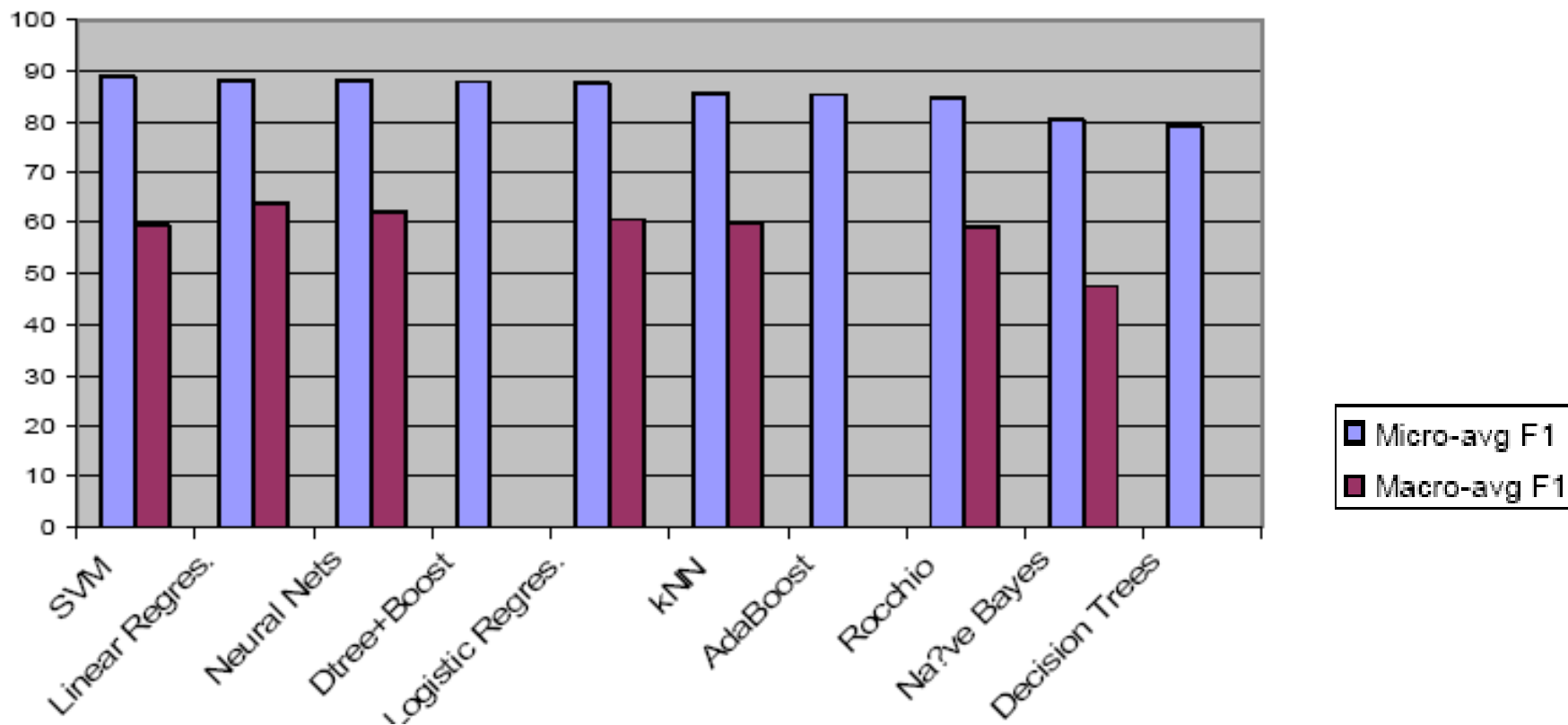
Recall:  $r=a/(a+c)$  truly-positive (percentage of positive docs detected)

Precision:  $p=a/(a+b)$  how accurate is the predicted positive docs

Accuracy:  $(a+d)/n$  how accurate is all the predicted docs

F-measure: 
$$F_{\beta} = \frac{(\beta^2 + 1)pr}{\beta^2 p + r} \quad F_1 = \frac{2pr}{p + r}$$

# TC: Evaluation



Performance of different algorithms on Reuters-21578 corpus: 90 categories, 7769 Training docs, 3019 test docs, (Yang, 1999)



# Outline

---

- Introduction to Core Concepts in Machine Learning
- Basic Optimization Methods
- Machine Learning Applications in IR
  - Text categorization
  - Text clustering
  - Collaborative filtering
  - Learning to rank

# Text Clustering/Topic Modeling:

---

- Tasks:
  - **Given** a set of text documents/objects without label information
  - **Group** the text objects into meaningful clusters
  
- Applications
  - Corpus analysis navigation
  - Organize Web search results
  - Speed up retrieval speed
  - Your favorite applications?...



# Text Clustering: Learning Technologies

---

- **K-means** (Cutting et al. 1994)
- **Mixture model** (Nigam et al., 2000)
- **Probabilistic latent semantic indexing** (Hofmann. 1999)
- **Latent Dirichlet Allocation** (Blei et al. 2003)
- Max margin clustering (Xu et al. 2005)
- .....

# Text Clustering: K-means

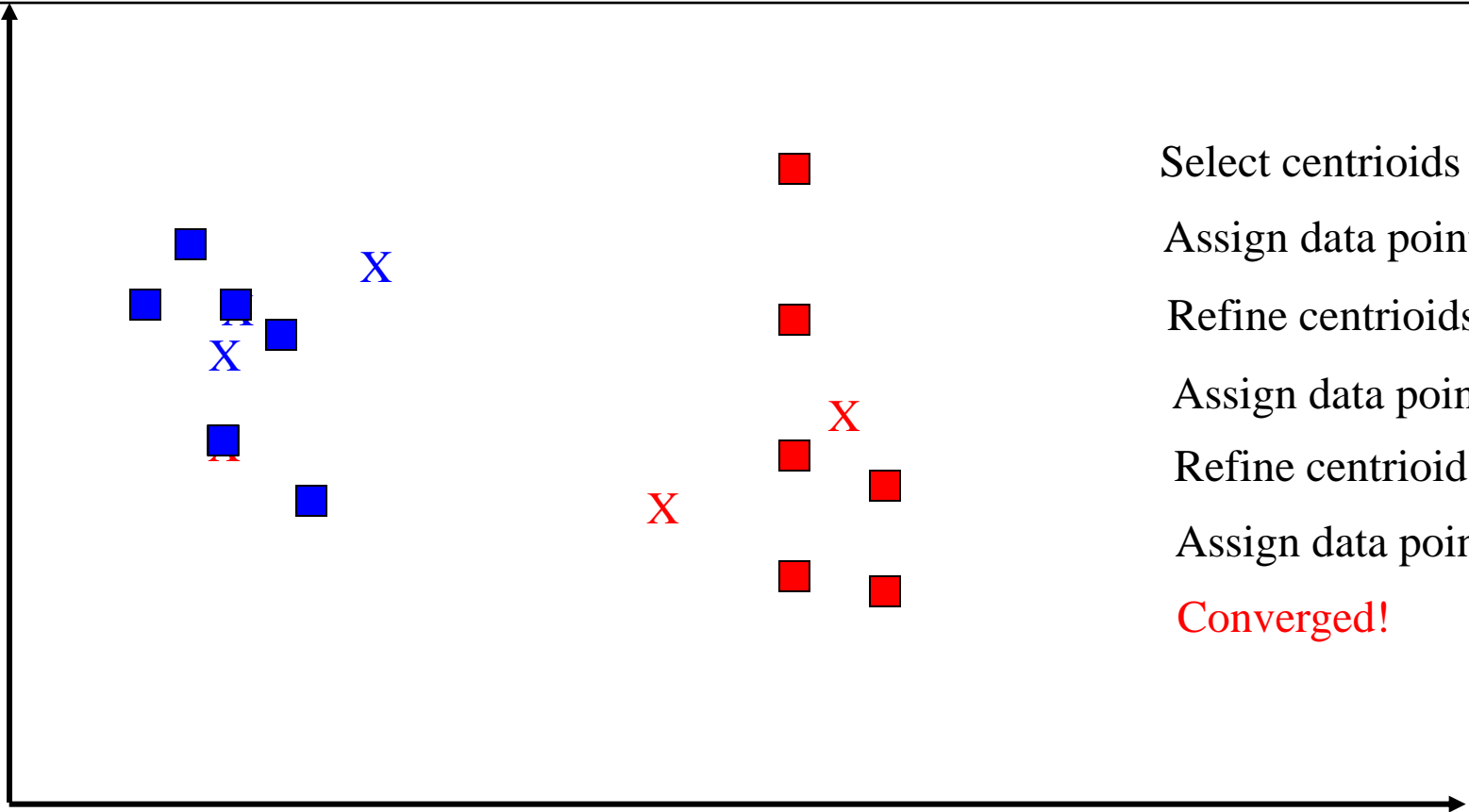
---

- 0. Represent text objects as real-valued vectors (e.g., tf.idf or BM25 document representation and Euclidean distance)
- 1. Select a set of seeds as centroids (center in a cluster).
- 2. Assign all data points to a cluster based on distance
- 3. Refine the centroids with mean representation of data points in clusters

$$\mathbf{r}_{\mu(c)} = \frac{1}{|c|} \sum_{x \in c} d$$

- Repeat 2 and 3 until converge

# K-means clustering example



- Select centroids
- Assign data points
- Refine centroids
- Assign data points
- Refine centroids
- Assign data points
- Converged!**

# Text Clustering: K-means

---

## Observation for K-means

- Easy to implement
- Efficient, used in many real world applications
- Hard cluster, one document only in one cluster, do not allow multiple cluster/topics for a document
- Heuristic choice of representation and distance

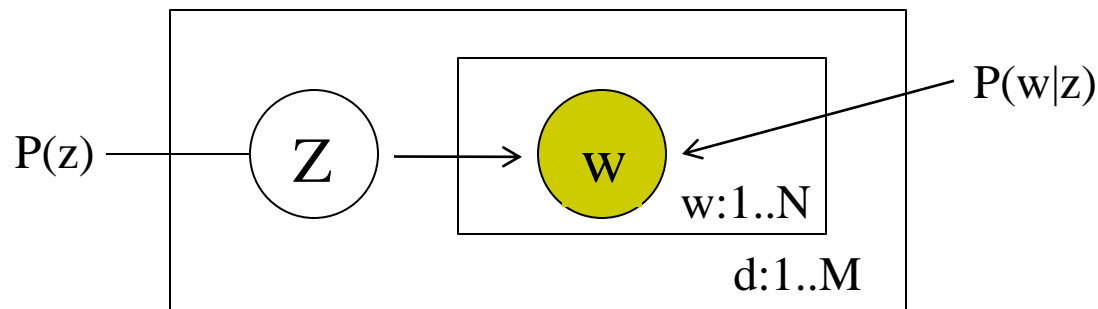
# Text Clustering: Mixture Model

Mixture Model of Unigrams (Nigam, McCallum, et al. 2000)

- There are a set of topics (language model) with weight  $P(z)$

The generation process of a document

- Sample a topic  $z$  with topic weight distribution  $P(z)$
- For each word in the document
  - Generate the word given the topic model  $P(w|z)$  (language model)



# Text Clustering: Mixture Model

- Learn model parameters  $\{P(z)\}$ ,  $\{P(w|z)\}$  by MLE

$$\max_d \prod_z \left[ \sum_z P(z) \prod_{v=1}^{|V|} P(w_v | z)^{C(w_v, d)} \right]$$

- 0. Generate random values of model parameters
- 1. Expectation: For each doc  $d$ , calculate posterior prob  $P(z|d)$

$$P(z | d) = \frac{P_t(z) \prod_{v=1}^{|V|} P_t(w_v | z)^{C(w_v, d)}}{\sum_{z'} P_t(z') \prod_{v=1}^{|V|} P_t(w_v | z')^{C(w_v, d)}}$$

- 2. Maximization: given posterior probs, refine model parameters

$$P_{t+1}(z) = \frac{\sum_d P(z | d)}{M} \quad P_{t+1}(w_v | z) = \frac{\sum_d P(z | d) C(w_v, d)}{\sum_d P(z | d) N}$$

- 3. Repeat 1 and 2 until converge



# Text Clustering: Mixture Model

---

## Observation for Mixture Model

- Soft clustering, one document can be associated with multiple clusters
- More solid generation story
- Given a topic, all words in a document are from the topic.  
Need to allow different words to belong to different topics

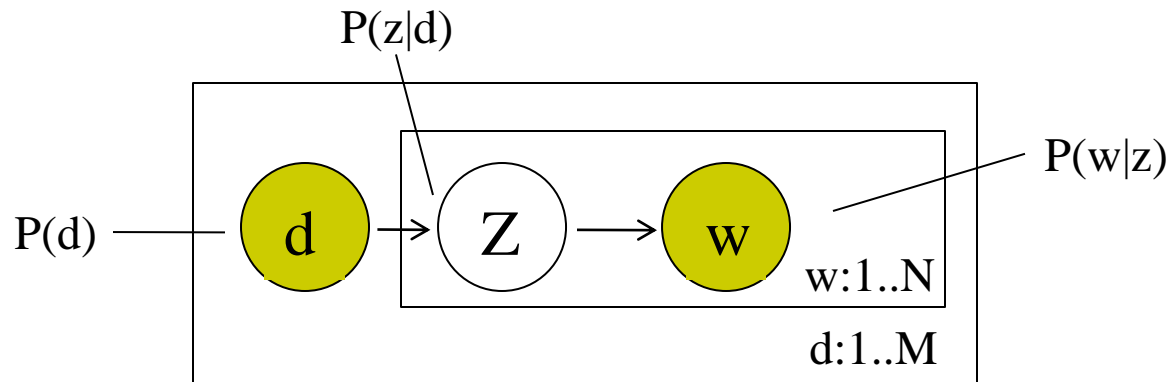
# Text Clustering: Probabilistic Latent Semantic Indexing (PLSI)

Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999)

- Soft clustering, words in a doc can belong to different topics

The generation process of a document

- For each word in the document
  - Choose a topic  $z$  according to a multinomial conditioned on the index  $d$
  - Generate the word by drawing from a multinomial conditioned on  $z$



# Text Clustering: PLSI

- Learn model parameters by MLE

$$\max_d \prod \left[ \prod_{v=1}^{|V|} \left[ \sum_z P(d | z) P(z) P(w_v | z) \right]^{C(w_v, d)} \right]$$

- 0. Generate random values of model parameters
- 1. Expectation: For each word in doc, calculate posterior prob  $P(z|d,w)$

$$P(z | d, w) = \frac{P_t(z)P_t(d | z)P_t(w_v | z)}{\sum_{z'} P_t(z')P_t(d | z')P_t(w_v | z')}$$

- 2. Maximization: given posterior probs, refine model parameters

$$P_{t+1}(z) = \frac{\sum_{d, w_v} C(w_v, d) P(z | d, w_v)}{M * N} \quad P_{t+1}(w_v | z) = \frac{\sum_d C(w_v, d) P(z | d, w_v)}{\sum_{d', w_v'} C(w_v', d') P(z | d', w_v')} \quad P_{t+1}(d | z) = \frac{\sum_{w_v} C(w_v, d) P(z | d, w_v)}{\sum_{d', w_v'} C(w_v', d') P(z | d', w_v')}$$

- 3. Repeat 1 and 2 until converge

# Text Clustering: PLSI

---

## Observation for PLSI

- Soft clustering, allows different words in a doc to belong to different topics, more modeling power!
- Num of model parameters grows with num of docs, may overfit
- Less solid generation story,  $\sum_d P(d | z) = 1$  for all training docs, what about unseen/new documents?
- Fold-in process for unseen/new documents, ..., keep all other model parameters fixed, and tune the topic document prob.

# Text Clustering: Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) (Blei, Ng, et al. 2003)

- Soft clustering, allow words from multiple topics, valid model

Topic mixture proportions of each doc drawn from a common distribution

Dirichlet distribution is a conjugate prior distribution topic mixture (multinomial distribution)

$$P(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K \theta_i^{\alpha_i - 1}; \forall i, \alpha_i \geq 1$$

Topic Mixture  $\downarrow$

Hyper parameter  $\downarrow$

$\uparrow$

Gamma function to ensure a valid prob distribution

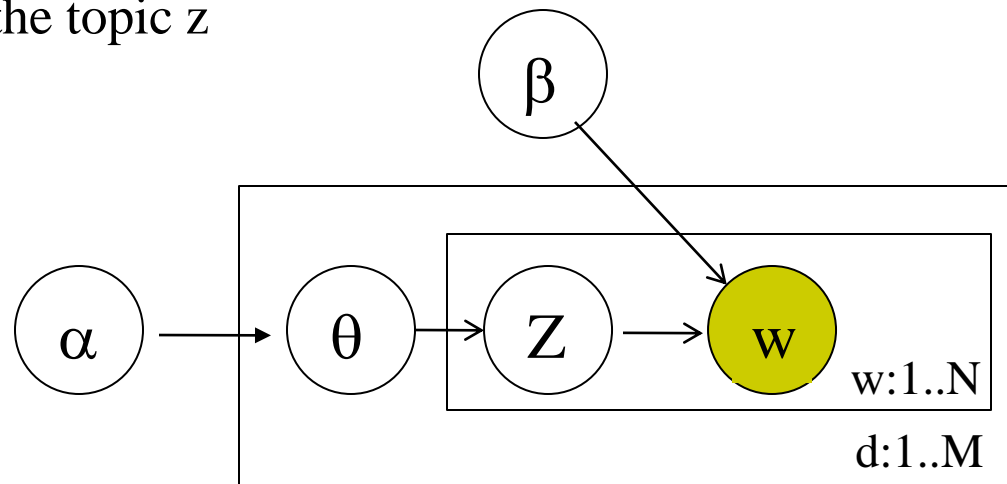
# Text Clustering: Latent Dirichlet Allocation (LDA)

The generation process of a document

Choose  $\theta$  from  $\text{Dirichlet}(\alpha)$

For each word in the document

- Choose a topic  $z$  from  $\text{Multinomial}(\theta)$
- Choose a word  $w$  from  $P(w|z, \beta)$ , a multinomial probability conditioned on the topic  $z$



# Text Clustering: Latent Dirichlet Allocation (LDA)

---

Learn model parameters  $\{\alpha\}$ ,  $\{\beta\}$  by MLE

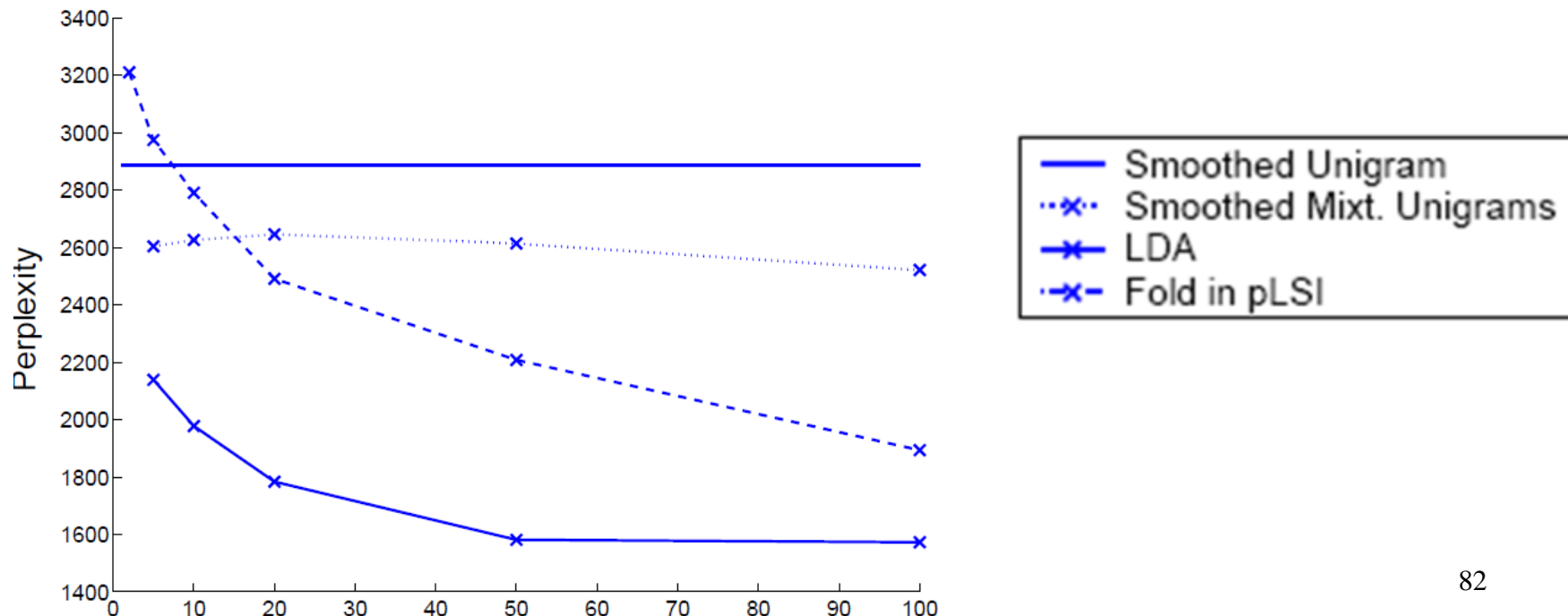
$$\max \prod_{d=1}^M \int \left( P(\theta_d | \alpha) \left( \prod_{v=1}^{|V|} \left[ \sum_z P(z | \theta_d) P(w_v | z, \beta) \right]^{C(w_v, d)} \right) d\theta_d \right)$$

- 0. Generate random values of model parameters
- 1. Expectation: calculate full posterior prob  $p(\theta, z | w, \alpha, \beta)$  is intractable, approx it with two factorized posterior probs (variational inference)
- 2. Maximization: given approx poster prob, refine model parameters
- 3. Repeat 1 and 2 until converge

# Text Clustering: Latent Dirichlet Allocation (LDA)

## Observation for LDA

- Soft clustering, allow different words in a doc belong to different topics, solid model
- Large computation cost





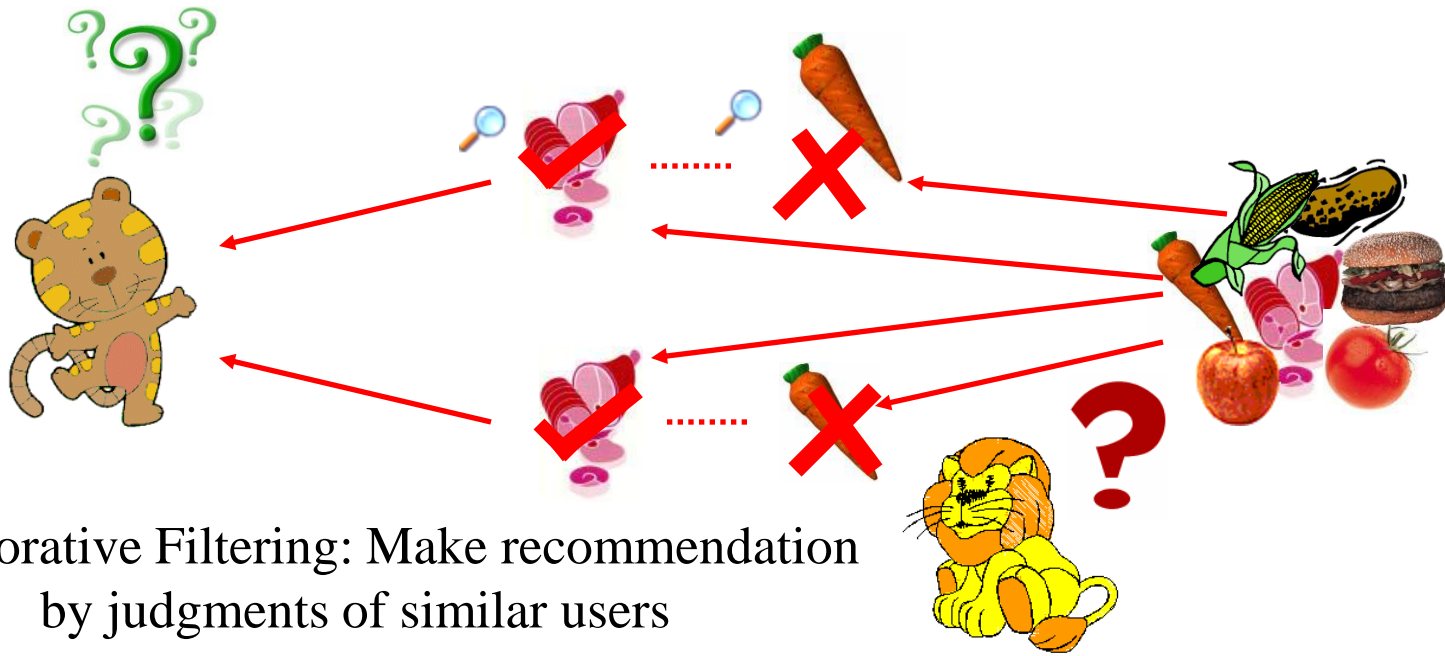
# Outline

---

- Introduction to Core Concepts in Machine Learning
- Basic Optimization Methods
- Machine Learning Applications in IR
  - Text categorization
  - Text clustering
  - Collaborative filtering
  - Learning to rank

# Collaborative Filtering (CF): Introduction

Content-Based Filtering: Recommend by analyzing the content information



Collaborative Filtering: Make recommendation by judgments of similar users

# Collaborative Filtering (CF): Introduction

---

- Tasks:
  - **Given** ratings from training users on objects, a few ratings from a test user on some objects
  - **Predict** the test user's ratings on other objects
- Applications
  - Book recommendation
  - Movie recommendation
  - Friend recommendation
  - Your favorite applications?...

# CF: Learning Technologies

---

- **Pearson correlation coefficient** (Resnick, Iacovou, et al., 1994)
- **Aspect model** (Hofmann & Puzicha, 1999)
- **Flexile mixture model** (Si & Jin, 2003)
- **Netflix methods** (Koren, 2009; Töscher & Jahrer., 2009)
- Personal diagnosis (Pennock, 2000)
- Ordinal regression (Shashua et al., 2002)
- .....

# CF: Formal framework

		Objects: $O_m$				
		$O_1$	$O_2$	$O_3$	$\dots O_j \dots$	$O_M$
Training Users: $U_n$	$U_1$	3	2	4		
	$U_2$		4	1		1
	$\vdots$					
	$U_i$					
	$\vdots$					
	$U_N$	5		2		2
Test User $U^{\text{test}}$		2	3			

$R_{u^{\text{test}}}(O_j) = ?$

What we have:

- Assume there are some ratings by training users
- Test user provides some amount of additional training data

What we do:

- Predict test user's rating based on training information

# CF: Pearson Correlation Coefficient

PCC (Resnick, Iacovou, et al., 1994)

How to determine the similarity between users?

- Measure the similarity in rating patterns between different users

Pearson Correlation Coefficient Similarity

Vector Space Similarity

$$w_{u,u^{test}} = \frac{\sum (R_{u^{test}}(o) - \bar{R}_{u^{test}})(R_u(o) - \bar{R}_u)}{\sqrt{\sum (R_{u^{test}}(o) - \bar{R}_{u^{test}})^2} \sqrt{\sum (R_u(o) - \bar{R}_u)^2}}$$

Average Ratings

$$w_{u,u^{test}} = \frac{\sum R_{u^{test}}(o)R_u(o)}{\sqrt{\sum R_{u^{test}}(o)^2} \sqrt{\sum R_u(o)^2}}$$

Prediction:  $\hat{R}_{u^{test}}(o) = \bar{R}_{u^{test}} + \frac{\sum_u w_{u,u^{test}} (R_u(o) - \bar{R}_u)}{\sum_u |w_{u,u^{test}}|}$

# CF: Pearson Correlation Coefficient

---

Observations with memory based approach

- Can be pretty effective
- Has a large amount of computation online costs

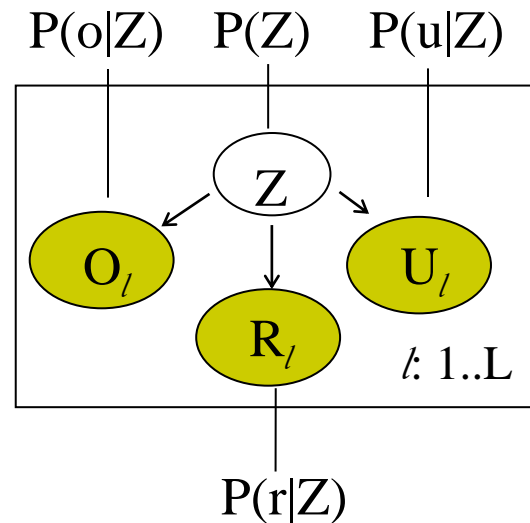
Possible Solution

- Cluster users/items offline, save for online computation cost
- Propose more solid probabilistic modeling method

# CF: Aspect Model (AM)

AM (Hofmann & Puzicha, 1999) models individual preferences as a convex combination of preference factors.

$$P(o_{(l)}, u_{(l)}, r_{(l)}) = \sum_{z \in Z} P(z) P(o_{(l)} | z) P(u_{(l)} | z) P(r_{(l)} | z)$$



# CF: Aspect Model (AM)

- Learn  $\{P(z)\}$ ,  $\{P(o|z)\}$ ,  $\{P(u|z)\}$ ,  $\{P(r|z)\}$  by MLE

$$\max \prod_{l=1}^L \sum_{z \in Z} P(z) P(o_{(l)} | z) P(u_{(l)} | z) P(r_{(l)} | z)$$

- Expectation: For each rating, calculate posterior prob

$$P(z | o_{(l)}, u_{(l)}, r_{(l)}) = \frac{P_t(z) P_t(o_{(l)} | z) P_t(u_{(l)} | z) P_t(r_{(l)} | z)}{\sum_{z' \in Z} P_t(z') P_t(o_{(l)} | z') P_t(u_{(l)} | z') P_t(r_{(l)} | z')}$$

- Maximization: given posterior probs, refine model parameters

$$P_{t+1}(z) = \frac{1}{L} \sum_l P(z | o_{(l)}, u_{(l)}, r_{(l)}) \quad P_{t+1}(o | z) = \frac{\sum_{o_{(l)}=o} P(z | o_{(l)}, u_{(l)}, r_{(l)})}{L \times P(z)}$$

$\{P(u|z)\}$  and  $\{P(r|z)\}$  can be obtained in a similar manner

# CF: Aspect Model (AM)

---

## □ Prediction Procedure

- Fold-In process to calculate joint probabilities

$$P(o, u^{test}, r_{(l)}) = \sum_z P(z) P(o | z) P(u^{test} | z) P(r_{(l)} | z)$$

Fold-in process by EM algorithm

- Calculate expectation for prediction

$$\hat{R}_{u^{test}}(o) = \sum_r r \frac{P(o, u^{test}, r)}{\sum_{r'} P(o, u^{test}, r')}$$

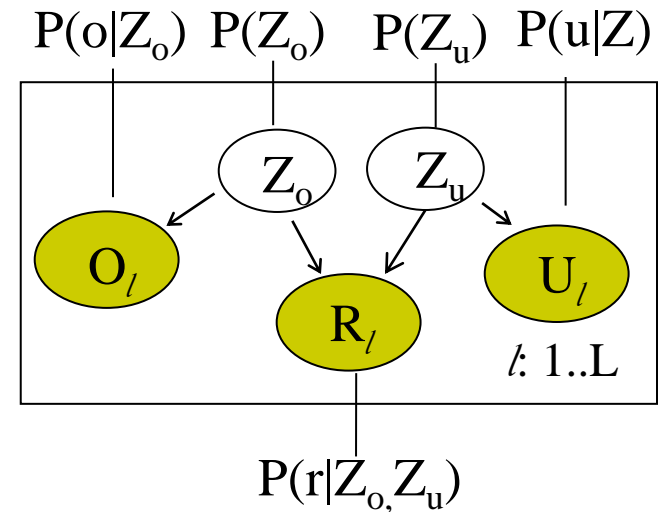
## Observation for AM

- Users and objects are different, may be generated from different groups

# CF: Flexible Mixture Model (FMM)

- Flexible Mixture Model (FMM) (Si & Jin, 2003):  
Cluster users and objects in separate groups

$$P(o_{(l)}, u_{(l)}, r_{(l)})$$
$$= \sum_{Z_o, Z_u} P(Z_o) P(Z_u) P(o_{(l)} | Z_o) P(u_{(l)} | Z_u) P(r_{(l)} | Z_o, Z_u)$$



# CF: Flexible Mixture Model (FMM)

---

- Learn  $\{P(z_o)\}$ ,  $\{P(z_u)\}$ ,  $\{P(o|z_o)\}$ ,  $\{P(u|z_u)\}$ ,  $\{P(r|z_o, z_u)\}$

$$\max \prod_{l=1}^L \sum_{Z_o, Z_u} P(Z_o)P(Z_u)P(o_{(l)} | Z_o)P(u_{(l)} | Z_u)P(r_{(l)} | Z_o, Z_u)$$

- Expectation: For each rating, calculate posterior prob

$$P(z_o, z_u | o_{(l)}, u_{(l)}, r_{(l)}) = \frac{P(Z_o)P(Z_u)P(o_{(l)} | Z_o)P(u_{(l)} | Z_u)P(r_{(l)} | Z_o, Z_u)}{\sum_{Z_o', Z_u'} (P(Z_o')P(Z_u')P(o_{(l)} | Z_o')P(u_{(l)} | Z_u')P(r_{(l)} | Z_o', Z_u'))}$$

- Maximization: given posterior probs, refine model parameters

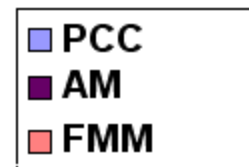
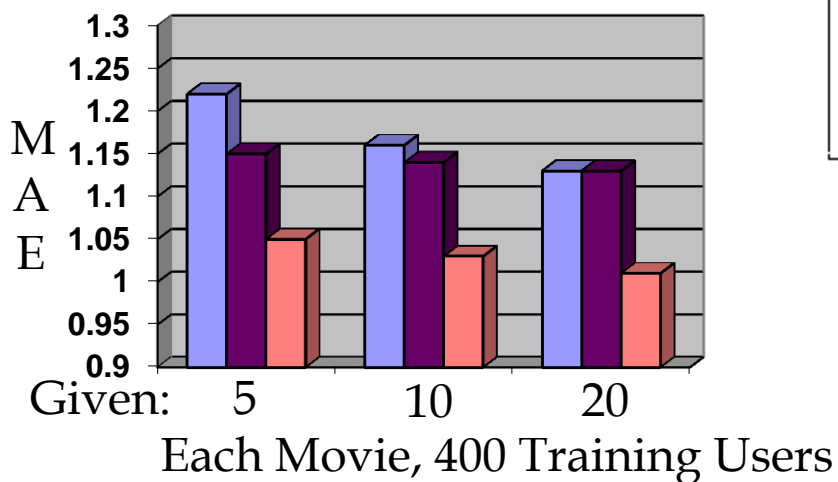
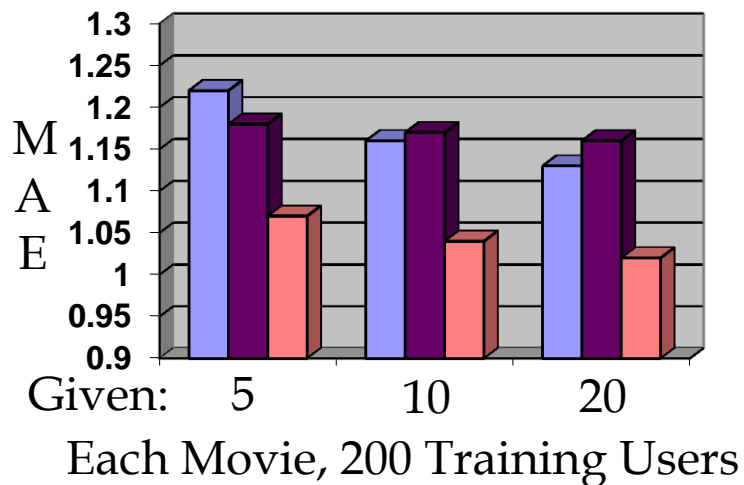
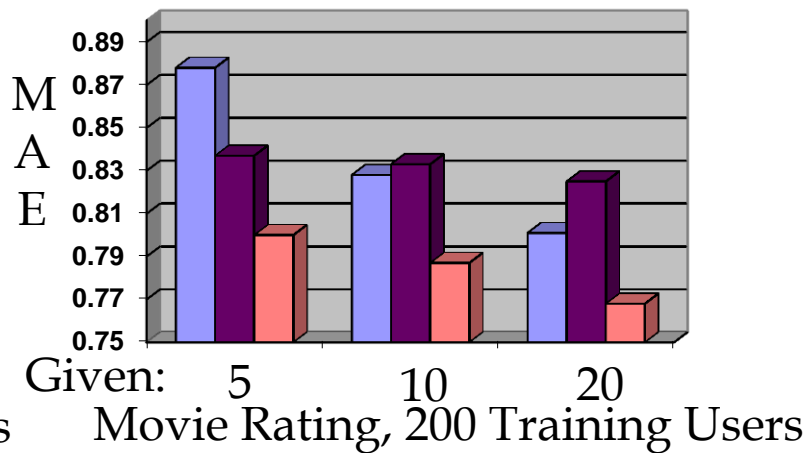
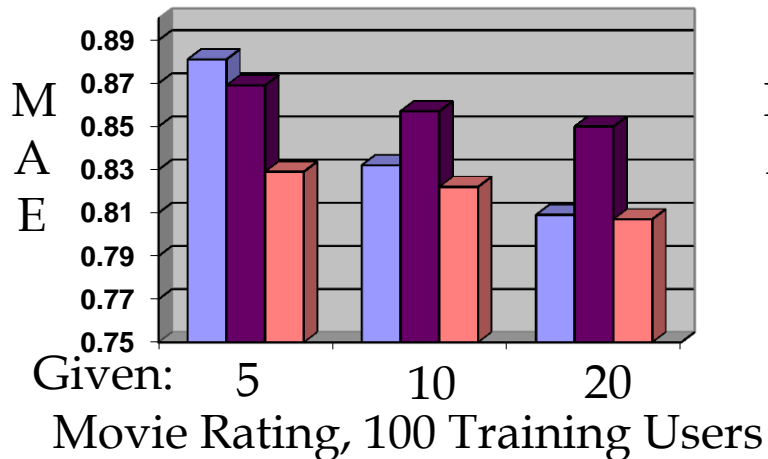
# CF: Evaluation

---

- Two movie datasets: Movie rating has ratings from 500 users on 1000 movies; Each movie has ratings from 2000 users on 1682 movies
- Different configurations of num of training users
- Different configurations of given num of ratings from test users
- Evaluation metric MAE: average absolute deviation of predicted ratings to the actual ratings on movies.

$$MAE = \frac{1}{L_{Test}} \sum_l |r_{(l)} - R_{o(l)}^{\hat{}}(u_{(l)})|$$

# CF: Evaluation



# CF: Netflix methods

---

- Netflix competition on a large dataset with ~100M ratings of ~500K users on ~18K movies with info of rating date.
- Many valuable methods proposed such as nearest neighbor methods, matrix factorization, restricted Boltzmann machine, etc..
- We brief introduce techniques behind BellKor and the winning strategy from the combined team of BellKor's Pragmatic Chaos (Koren, 2009; Töscher & Jahrer, 2009)

# CF: Netflix methods

## Bellkor

- Baseline predictors

$$\hat{R}_u(o) = b_{uo} = \mu + b_\mu + b_o$$

Overall deviation      User deviation      Object deviation

- Matrix factorization

$$\hat{R}_u(o) = b_{uo} + q_o^T \left( p_u(t_{uo}) + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j \right)$$

Object vector      Dynamic user factor      Approx user factor from objects rated

$$p_{uk}(t) = p_{uk} + \alpha_{uk} * dev_u(t) \quad k = 1, \dots, f$$

static user factor      Linear approximation of change over time

# CF: Ensemble learning/blending

---

- A key factor in winning Netflix lies in combining/blending many models/predictors together
  - Simple separate modeling and linear blending:  
Models trained separately and linear blending (combination based on linear regression)
  - Separate modeling and non-linear blending:  
Models trained separately with non-linear blending (e.g., Neural-Network)
  - Joint modeling and blending:  
Models trained sequentially and stopped when blending improvement is the best



# Outline

---

- Introduction to Core Concepts in Machine Learning
- Basic Optimization Methods
- Machine Learning Applications in IR
  - Text categorization
  - Text clustering
  - Collaborative filtering
  - Learning to rank

# Learning to Rank (LTR): Learning Technologies

---

## □ Tasks:

- **Given** lists of items (e.g., documents/webpages for queries) with order. Order can be induced from binary judgments, ordinal or numerical scores.
- **Predict** a desired ranking of new items (e.g., documents for new queries)
- Relatively new, but good progress in theory and in real world Web search

## □ Data representation

- Usually each item represents a vector of features (e.g., tf.idf statistics related with a pair of document and query, page rank values)
- Each item may be associated with a score (0/1 for relevance, multi-valued judgment) that generates order

# LTR: Traditional Learning Approach

---

- Direct application of traditional classification or regression models for ranking problems
  - binary relevance judgments as binary classification; multi-valued judgments as “non-ordered” categories or simple real values
  - The treatment of multi-valued judgments can be suboptimal as it ignores order information among the judgments
  
- Data representation
  - For  $i^{\text{th}}$  query,  $j^{\text{th}}$  document,  $\{f_t(q_i, d_j)\}$  represents a vector of input features
  - Output in either classification representation or regression presentation for different methods

# LTR: Traditional Learning Approach

---

- Least Square Polynomial Retrieval (N. Fuhr, 1989)
  - Use polynomial function with input features  $k^{\text{th}}$  element
  - Output classification representation:  $y_k=(0,\dots,1,\dots,0)$
  - Minimize least square error for learning model

$$\min \sum_{i=1}^Q \sum_{j=1}^M \left| y_j^{(i)} - \sum_t \beta(t) f_t(q_i, d_j) \right|^2$$

- Discriminative Model for IR (Nallapati, 2004)
  - Output binary classification representation as 1/0
  - One model uses max entropy (logistic regression)
  - One model uses support vector machine

# LTR: Traditional Learning Approach

---

- Regression Tree Approach (Kramer, Widmer, et al, 2001)
  - Output in regression representation as real values
  - Utilize a regression tree approach for predicting the output values
  - Post-process to convert predicted output as real values to categorical
  
- Observations with training learning methods
  - Sub-optimal treatment for multi-valued discrete judgments
  - Do not model important order info (e.g., pairwise)



# LTR: Pointwise Ordinal Regression

---

- Consider order information in multi-valued discrete judgments
  - Output classification representation as real values
  - Model parameters contain weights for each feature and real-valued boundaries for multi-valued discrete judgments
  - All the model parameters are learned to maximize correctness on training data

# LTR: Pointwise Ordinal Regression

---

- Pranking with Ranking (Crammer & Singer, 2002)
  - Learn a linear function with weights and real-valued boundaries for multi-valued discrete judgments
  - Online optimization algorithm, whenever makes a wrong prediction on multi-valued judgment, adjust weights and real-valued boundaries
  
- Observations with Pointwise Ordinal Regression
  - Not much different from traditional learning algorithm
  - Do not model important order info (e.g., pairwise)

# LTR: Pairwise Approach

---

- Consider pairwise preference between a pair of documents for a user query
  - Pairwise input representation  $\{f_t(q_i, d_u, d_v)\}$  is often the difference between input representations of two documents.
  - Output classification representation is often binary value for preference
  - Pairwise preference is natural for implicit feedback. For example, ingoring result #1 and viewing result #2 suggest  $\#2 > \#1$
  - Pairwise preference better captures the concept that preference or relevance is defined for documents with a specific query

# LTR: Pairwise Approach

---

- Learning to Order Things (W. Cohen, R. Schapire, et al. NIPS 1999)

- Define pairwise loss function as:

$$\frac{-\sum_{i=1}^q \sum_{d_u \neq d_v} \left[ 1 - \frac{\sum \beta_t f_t(q_i, d_u, d_v)}{\beta_t} \right]}{\sum_{i=1}^q \sum_{d_u \neq d_v} 1}$$

- Use a weighted majority algorithm to learn model parameters  $\{\beta_t\}$
- Prediction, pairwise preference  $\rightarrow$  rank list (total order): hard problem
- A greedy ordering algorithm can be shown to obtain at least half agreement on pairwise preference with an optimal rank

# LTR: Pairwise Approach

---

- RankNet (C. Burges, T. Shaked, et al. ICML 2005)

- Model pairwise preference probability as 
$$P(d_u \succ d_v | q_i) = \frac{\exp\left(\sum_{\beta_t} \beta_t f_t(q_i, d_u, d_v)\right)}{1 + \exp\left(\sum_{\beta_t} \beta_t f_t(q_i, d_u, d_v)\right)}$$

- Use cross entropy loss (i.e., prob difference between true pairwise preference and predicted preference)
- Use neural network with gradient descent to learn model parameters  $\{\beta_t\}$  for minimizing loss

# LTR: Pairwise Approach

## □ Ranking SVM (T. Joachims, KDD 2002)

- Adapt SVM for considering pairwise preference as:

Margin Trade-off Parameter Error for each pairwise preference

$$\max_w \frac{1}{2} \|\beta\|^2 + C \sum_i \sum_{u,v} \xi_{uv}^i$$

*subject to:*  $\sum_{\beta_t} \beta_t f_t(q_i, d_u, d_v) \geq 1 - \xi_{uv}^i, \xi_{uv}^i \geq 0, \text{ if } d_u \mathbf{f} d_v$

## □ Observations with Pointwise Ordinal Regression

- Cannot handle preference/relevance judgment defined on rank lists.



# LTR: Listwise Approach

---

- Consider order information in multi-valued discrete judgments
  - Output representation is permutation of documents for a query
  - Input representation is the union of input representation of individual documents for a query
  - Learning model parameters to approximate desired permutation

# LTR: Pointwise Ordinal Regression

---

- Direct optimization of IR evaluation measure in listwise approach
  - Try to optimize (approximated) IR measurement such as MAP
  - Example: SVM-Struct (Y. Yue, T. Finley, et al, 2007) with MAP margin constraints
  
- Listwise loss minimization
  - Minimize a loss function based on distance of permutations
  - Example: ListMLE (F. Xia, T. Liu, et al. 2008) defines permutation likelihood, find model parameters that maximize permutation likelihood

# Bibliography

---

## □ Introduction to IR

- C. Manning, P. Raghavan, P. Schütze. Introduction to Information Retrieval. Cambridge University Press (2008).
- B. Croft, D. Metzler, T. Strohman. Search Engines: Information Retrieval in Practice. Addison Wesley (2008)
- S. Büttcher; C. L. A. Clarke, G. V. Cormack. Information Retrieval: Implementing and Evaluating Search Engines. MIT Press (2010)
- R Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999

## □ Introduction to ML

- C. M. Bishop. Pattern Recognition and Machine Learning. Springer. 2006

## □ Introduction to Probability & Statistics

- L. Wasserman. All of Statistics. Springer. 2010

## □ Introduction to Optimization

- S. Boyd. & L. Vandenberghe. Convex Optimization. Cambridge University Press. 2004.

# Bibliography

---

## □ Text categorization

- Y. Yang. Expert network: Effective and Efficient Learning from Human Decisions in Text Categorization and retrieval. SIGIR. 1994
- D. D. Lewis. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. SIGIR, 1992.
- A. McCallum. A Comparison of Event Models for Naïve Bayes Text Categorization. AAAI workshop, 1998.
- Fuhr N, Hartmann S, et al. Air/x—A rule-based Multistage Indexing Systems for Large Subject Fields. RAIO. 1991.
- Y. Yang and C. G. Chute. An example-based Mapping Method for Text Categorization and Retrieval. ACM TOIS. 12(3)-252-277, 1994.
- T. Joachims. Text Categorization with Support Vector Machines: Learning with many relevant features. ECML. 1998.
- L. Cai and T. Hofmann. Hierarchical Document Categorization with Support Vector Machines. CIKM. 2004.
- R. E. Schapire, Y. Singer, et al. Boosting and Rocchio Applied to Text Filtering. SIGIR. 1998.
- E. Wiener, J. O. Pedersen, et al. A Neural Network Approach to Topic Spotting. SDAIR, 1995

# Bibliography

---

## □ Text categorization (cont)

- R. O. Duda, P. E. Hart, et al. Pattern Classification (2<sup>nd</sup> Ed). Wiley. 2000.
- L. Devroye, L. Györfi, et al. A Probabilistic Theory of Pattern Recognition. Springer. 1996.
- Y. M. Yang. An evaluation of statistical approaches to text categorization. Information Retrieval, 1999

## □ Text clustering

- D. R. cutting, D. R. Karger, et al. Scatter/Gather: A Cluster-based approach to browsing large document collections. SIGIR, 1992.
- K. Nigam, A. McCallum, et al. Text classification from labeled and unlabeled documents using EM. Machine Learning 39, 2000.
- T. Hofmann. Probabilistic Latent Semantic Indexing. SIGIR 1999
- D. M. Blei, A. Y. Ng, et al. Latent Dirichlet Allocation. Journal of Machine Learning Research. 2003
- L. L. Xu, J. Neufeld, et al. Maximum Margin Clustering. NIPS 2005

# Bibliography

---

## □ Collaborative Filtering

- P. Resnick, N. Iacovou, et al. Grouplens: An Open Architecture for Collaborative Filtering of Netnews. CSCW 1994
- T. Hofmann & J. Puzicha. Latent Class Models for Collaborative Filtering. IJCAI 1999
- L. Si & R. Jin. Flexible Mixture Model for Collaborative Filtering. ICML 2003
- Y. Koren The BellKor Solution to the Netflix Grand Prize. 2009
- A. Töscher & Michale Jahrer. The BigChaos Solution to the Netflix Grand Prize. 2009
- D. M. Pennock, E. Horvitz, et al. Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-based Approach. UAI 2000
- A Shashua & A. Levin. Taxonomy of Large Margin Principle Algorithms for Ordinal Regression Problems. NIPS 2002

# Bibliography

---

## □ Learning to Rank

- Hang Li. Learning to Rank for Information Retrieval and Natural Language Processing. Morgan Claypool Publisher. 2011
- T. Y. Liu. Learning to Rank for Information Retrieval. FnTIR 2009
- N. Fuhr. Optimal Polynomial Retrieval Functions Based on the Probability Ranking Principle. TOIS, 1989
- R. Nallapati. Discriminative Models for Information Retrieval. SIGIR 2004
- S. Kramer, G. Widmer, et al. Prediction of Ordinal Classes Using Regression Tree. Computer & Communication Sciences. 2001
- K. Crammer & Y. Singer, PRanking with ranking, NIPS 2002
- W. W. Cohen, R. E. Schapire, et al. Learning to Order Things. J. Artif Ingell. Res, 1999
- C. J. C. Burges, T. Shaked, et al. Learning to Ranking Using Gradient Descent. ICML 2005
- T. Joachims. Optimizing Search Engine Using Clickthrough data. KDD 2002
- Y. Yue, T. Finley, et al. A Support Vector Method for Optimizing Average Precision. SIGIR 2007
- F. Xia, T. Liu, et al. Listwise Approach to Learning to Rank: Theory and Algorithms. ICML 2008



# Thank You

---

# TC: Naïve Bayes Classification

- Find model parameters for a category that maximizes the generation likelihood  $P(d_{c_1, \dots, d_{c_L}} | c)$

$$P(d_{c_1, \dots, d_{c_L}} | c) \propto \prod_{i=1}^{c_L} \prod_{v=1}^{|V|} P(w_v | c)^{C(w_v, d_i)}$$

$$l(\theta_c) = \sum_{i=1}^{c_L} \prod_{v=1}^{|V|} \log(P(w_v | c)) C(w_v, d_i)$$

Use Lagrange multiplier approach; Set partial derivatives to zero

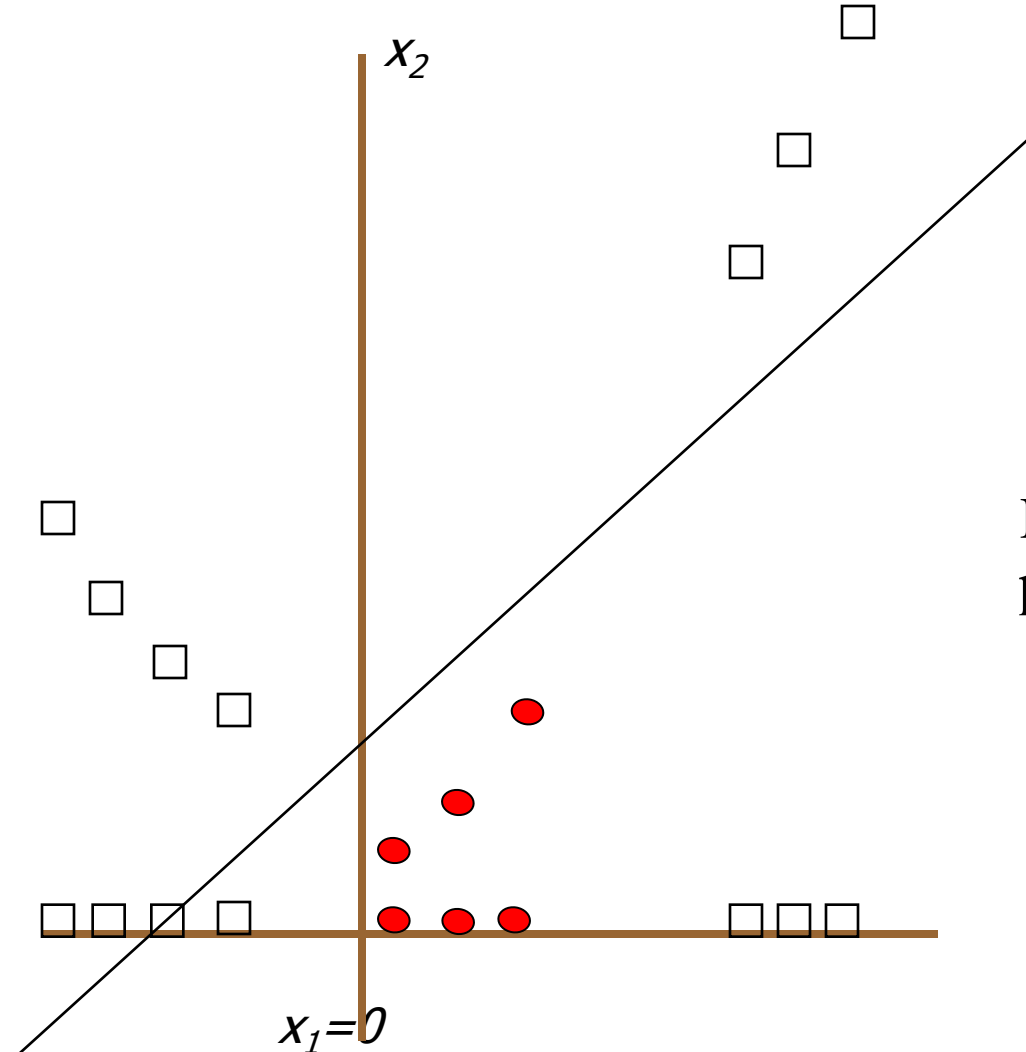
$$l'(\theta_c) = \sum_{i=1}^{c_L} \prod_{v=1}^{|V|} \log(P(w_v | c)) C(w_v, d_i) + \lambda \left( \sum_v p(w_v | c) - 1 \right)$$

$$\frac{\partial l'}{\partial P(w_v | c)} = \frac{\sum_{i=1}^{c_L} C(w_v, d_i)}{P(w_v | c)} + \lambda = 0 \Rightarrow P(w_v | c) = -\frac{\sum_{i=1}^{c_L} C(w_v, d_i)}{\lambda}$$

Since  $\sum_k p_k = 1$ ,  $\lambda = -\sum_{i=1}^{c_L} |d_i|$  So,  $P^*(w_v | c) = \frac{\sum_{i=1}^{c_L} C(w_v, d_i)}{\sum_{i=1}^{c_L} |d_i|}$

Normalization by count statistics

# Non-linear SVM



Key idea: transform  $X_i$  to a higher dimension space