Topic Models (Continued)

Admin Notes

Schedule

Today and next class (group meetings first hour, lecture immediately following)

- Today: Topic model (Pt. II) / Semi-supervised learning (Pt. I)
- 30 Oct.: Semi-supervised learning (Pt. II)

Last four lectures (review, projects, and final)

- 6 Nov.: Sultan's review (1:30-3:15)
  Open session (1:30 – 4:15)
  (Project – Shakarian; Final – Sultan)
- 13, 20 Nov.: Project presentations
- 27 Nov.: No class
- 4 Dec.: Final exam (during normal class time)

Review From Last Time

- Pt. I (Supervised – find a class):
  - Given a document, can we classify it?
  - i.e. Classify the sentiment, etc.

  - Naïve Bayes
    - Features for each word (i.e. TF*IDF)
    - Independence assumption among words
    - For each words, we rely on the value P(w=x | c) which is the probability of the value associated with word w being x given it belong to class c.
    - P(w=x | c) is just the fraction of training samples in class c where the feature value associated with w was x.

Review From Last Time

- Pt. I (Supervised – find a class):
  - Given a document **and a topic**, can we classify it?
  - i.e. Classify the sentiment, etc.

  - Mixture Model
    - Features for each word (i.e. TF*IDF)
    - Independence assumption among words **but conditioned on a specific topic.**
    - For each words, we rely on the value P(w=x | c, θ) which is the probability of the value associated with word w being x given it belong to class c **and is of topic θ.**
    - P(w=x | c , θ) is just the fraction of training samples in class c **in topic** θ where the feature value associated with w was x.
    - Intuition: we have enough samples where P(w=x | c , θ) give me better information than just P(w=x | c)

Review From Last Time

- Pt. II (Unsupervised – find a topic):
  - Given a set of documents can we identify a set of topics for each of them (preferably with a PDF)?

  - SVD
    - Consider a matrix of documents by words
    - This can be decomposed into three matrices using a linear algebra technique called Singular Value Decomposition (SVD)
    - A(n*m) = U(n*n) E(n*m) V(m*m)
      - U's colums are Eigenvectors of AAT
      - V's columns are Eigenvectors of ATA
      - E is a diagonal matrix containing the square roots of the eigenvalues of U or V in descending order
    - Keep only k eigen values from E
      - A(n*m) ≈ U(n*k) E(k*k) V(k*m)
    - U maps terms to topics; V maps topics to documents
    - Not based on a statistical model – just a byproduct of linear algebra (but effective in practice!)

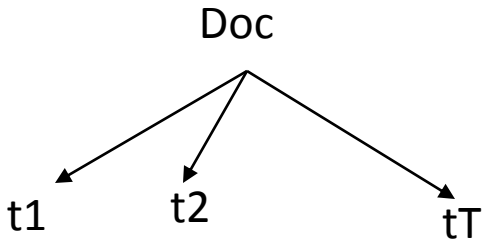
pLSI

Probabilistic Latent Semantic Analysis

- Let us start from what we know
- Remember the random sequence model

$$P(doc) = P(term_1 \mid doc)P(term_2 \mid doc)...P(term_L \mid doc)$$

$$= \prod_{l=1}^{L} P(term_l \mid doc) = \prod_{t=1}^{T} P(term_t \mid doc)^{X(term_t, doc)}$$

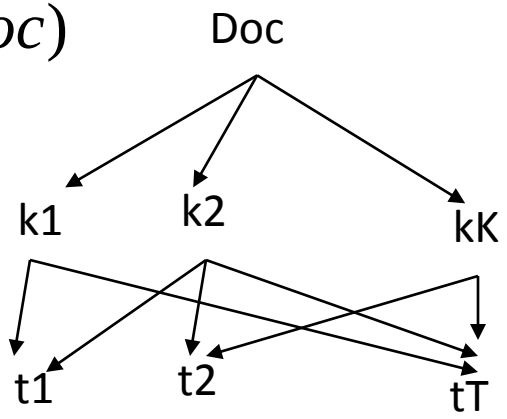


Probabilistic Latent Semantic Analysis

Think: Topic ~ Factor

- Now let us have K topics as well:

$$P(term_t \mid doc) = \sum_{k=1}^{K} P(term_t \mid topic_k) P(topic_k \mid doc)$$



The same, written using shorthands :

$$P(t \mid doc) = \sum_{k=1}^{K} P(t \mid k) P(k \mid doc)$$

So by replacing this, for any doc in the collection ,

$$P(doc) = \prod_{t=1}^{T} \{ \sum_{k=1}^{K} P(t \mid k) P(k \mid doc) \}^{X(t,doc)}$$

Which are the parameters of this model?

Probabilistic Latent Semantic Analysis

- The parameters of this model are:
  P(t|k)
  P(k|doc)
- It is possible to derive the equations for computing these parameters by Maximum Likelihood.
- If we do so, what do we get?
  P(t|k) for all t and k, is a term by topic matrix
  (gives which terms make up a topic)
  P(k|doc) for all k and doc, is a topic by document matrix
  (gives which topics are in a document)

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Deriving the parameter estimation algorithm

- The log likelihood of this model is the log probability of the entire collection:

$$\sum_{d=1}^{N} \log P(d) = \sum_{d=1}^{N} \sum_{t=1}^{T} X(t,d) \log \sum_{k=1}^{K} P(t \mid k) P(k \mid d)$$

which is to be maximised w.r.t. parameters $P(t \mid k)$ and then also $P(k \mid d)$,

subject to the constraint s that $\sum_{t=1}^{T} P(t \mid k) = 1$ and $\sum_{k=1}^{K} P(k \mid d) = 1.$

For those who would enjoy to work it out:

- Lagrangian terms are added to ensure the constraints
- Derivatives are taken wrt the parameters (one of them at a time) and equate these to zero
- Solve the resulting equations. You will get fixed point equations which can be solved iteratively. This is the PLSA algorithm.

Note these steps are the same in deriving the Maximum Likelihood estimate for random sequence models, just the working is a little more tedious.

We skip doing this in the class, we just give the resulting algorithm (see next slide)

The PLSA algorithm

- Inputs: term by document matrix X(t,d), t=1:T, d=1:N and the number K of topics sought
- Initialise arrays P1 and P2 randomly with numbers between [0,1] and normalise them to sum to 1 along rows
- Iterate until convergence

  For d=1 to N, For t =1 to T, For k=1:K

$$P1(t,k) \leftarrow P1(t,k)\sum_{d=1}^{N}\frac{X(t,d)}{\sum_{k=1}^{K}P1(t,k)P2(k,d)}P2(k,d); \quad P1(t,k) \leftarrow \frac{P1(t,k)}{\sum_{t=1}^{T}P1(t,k)}$$

$$P2(k,d) \leftarrow P2(k,d)\sum_{t=1}^{T}\frac{x(t,d)}{\sum_{k=1}^{K}P1(t,k)P2(k,d)}P1(t,k); \quad P2(k,d) \leftarrow \frac{P2(k,d)}{\sum_{k=1}^{K}P2(k,d)}$$

- Output: arrays P1 and P2, which hold the estimated parameters P(t|k) and P(k|d) respectively

Example of topics found from a Science Magazine papers collection

universe	0.0439
galaxies	0.0375
clusters	0.0279
matter	0.0233
galaxy	0.0232
cluster	0.0214
cosmic	0.0137
dark	0.0131
light	0.0109
density	0.01

drug	0.0672
patients	0.0493
drugs	0.0444
clinical	0.0346
treatment	0.028
trials	0.0277
therapy	0.0213
trial	0.0164
disease	0.0157
medical	0.00997

cells	0.0675
stem	0.0478
human	0.0421
cell	0.0309
gene	0.025
tissue	0.0185
cloning	0.0169
transfer	0.0155
blood	0.0113
embryos	0.0111

sequence	0.0818
sequences	0.0493
genome	0.033
dna	0.0257
sequencing	0.0172
map	0.0123
genes	0.0122
chromosome	0.0119
regions	0.0119
human	0.0111

years	0.156
million	0.0556
ago	0.045
time	0.0317
age	0.0243
year	0.024
record	0.0238
early	0.0233
billion	0.0177
history	0.0148

bacteria	0.0983
bacterial	0.0561
resistance	0.0431
coli	0.0381
strains	0.025
microbiol	0.0214
microbial	0.0196
strain	0.0165
salmonella	0.0163
resistant	0.0145

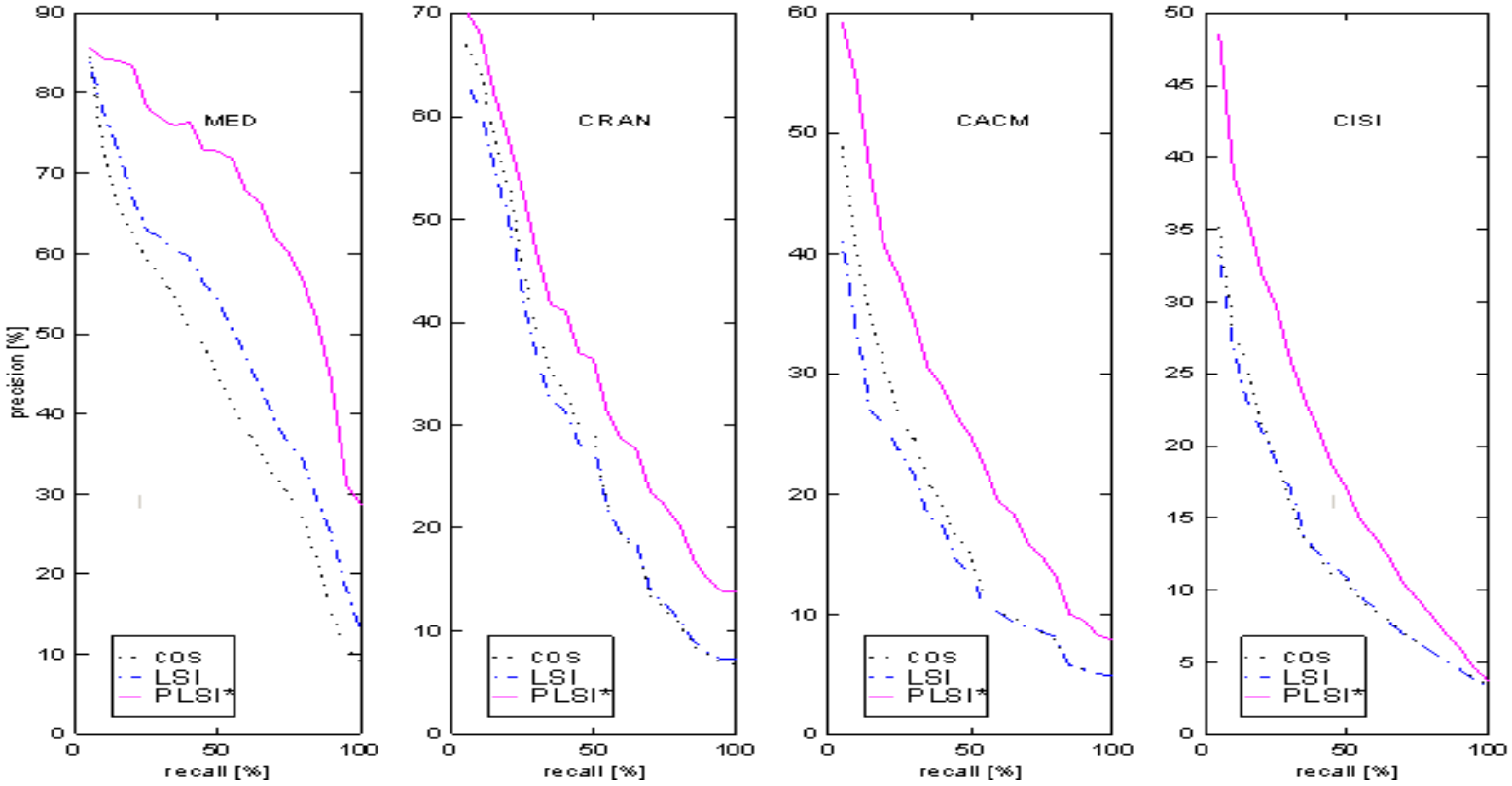
male	0.0558
females	0.0541
female	0.0529
males	0.0477
sex	0.0339
reproductive	0.0172
offspring	0.0168
sexual	0.0166
reproduction	0.0143
eggs	0.0138

theory	0.0811
physics	0.0782
physicists	0.0146
einstein	0.0142
university	0.013
gravity	0.013
black	0.0127
theories	0.01
aps	0.00987
matter	0.00954

immune	0.0909
response	0.0375
system	0.0358
responses	0.0322
antigen	0.0263
antigens	0.0184
immunity	0.0176
immunology	0.0145
antibody	0.014
autoimmune	0.0128

stars	0.0524
star	0.0458
astrophys	0.0237
mass	0.021
disk	0.0173
black	0.0161
gas	0.0149
stellar	0.0127
astron	0.0125
hole	0.00824

The performance of a retrieval system based on this model (PLSI) was found superior to that of both the vector space based similarity (cos) and a non-probabilistic latent semantic indexing (LSI) method. (We skip details here.)



From Th. Hofmann, 2000

Summing up

- Documents can be represented as numeric vectors in the space of words.
- The order of words is lost but the co-occurrences of words may still provide useful insights about the topical content of a collection of documents.
- PLSA is an unsupervised method based on this idea.
- We can use it to find out what topics are there in a collection of documents
- It is also a good basis for information retrieval systems

LDA

Motivations for LDA

- In pLSI, the observed variable d is an index into some training set. There is no natural way for the model to handle previously unseen documents.
- The number of parameters for pLSI grows linearly with M (the number of documents in the training set).
- We would like to be Bayesian about our topic mixture proportions.

Model for Document Generation Under LDA

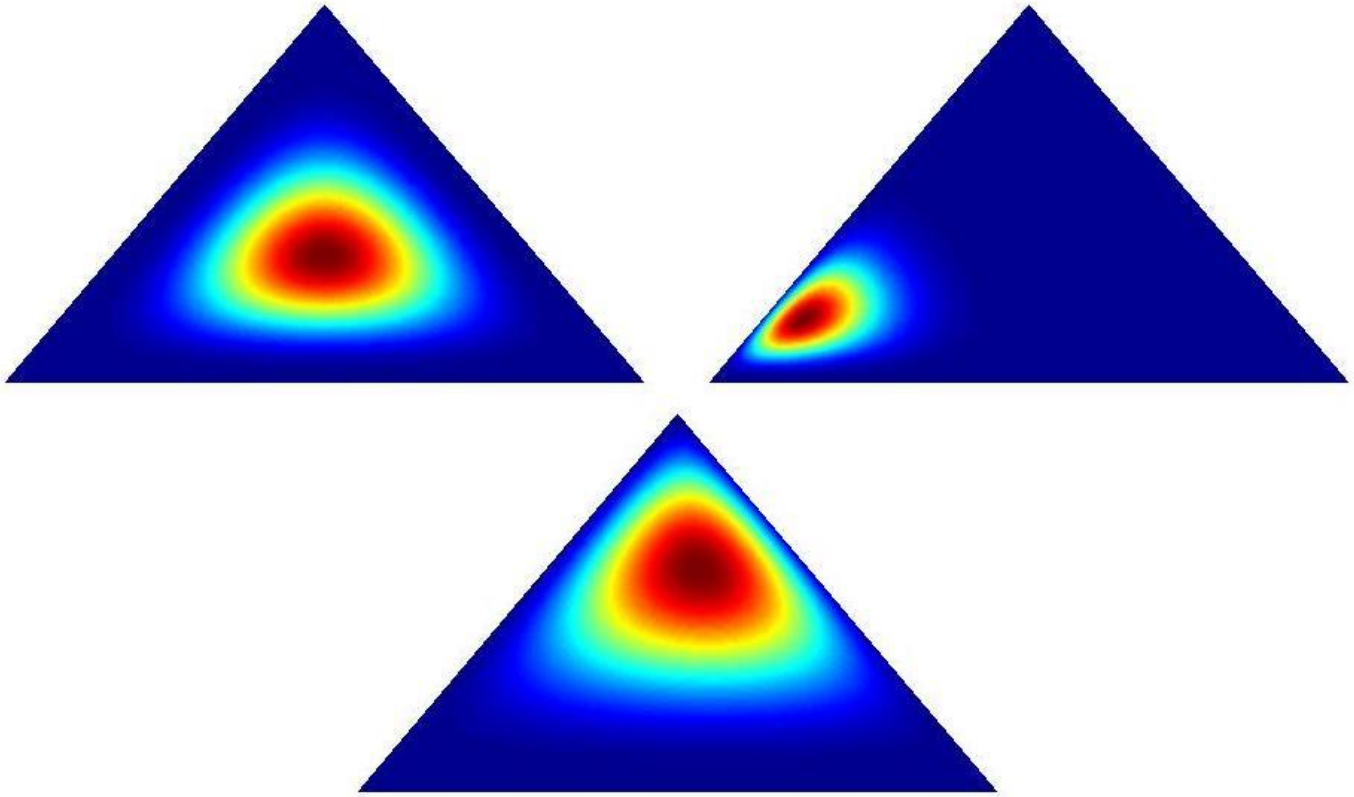
LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus D:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Dirichlet Distributions

- In the LDA model, we would like to say that the *topic mixture proportions* for each document are drawn from some distribution.
- So, we want to put a distribution on multinomials. That is, k-tuples of non-negative numbers that sum to one.
- The space is of all of these multinomials has a nice geometric interpretation as a (k-1)-*simplex*, which is just a generalization of a triangle to (k-1) dimensions.
- Criteria for selecting our prior:
  - It needs to be defined for a (k-1)-simplex.
  - Algebraically speaking, we would like it to play nice with the multinomial distribution.

Dirichlet Examples



Dirichlet Distributions

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

- Useful Facts:
  - This distribution is defined over a (k-1)-simplex. That is, it takes k non-negative arguments which sum to one. Consequently it is a natural distribution to use over multinomial distributions.
  - In fact, the Dirichlet distribution is the conjugate prior to the multinomial distribution. (This means that if our likelihood is multinomial with a Dirichlet prior, then the posterior is also Dirichlet!)
  - The Dirichlet parameter α_i can be thought of as a prior count of the i^{th} class.

Gamma Function

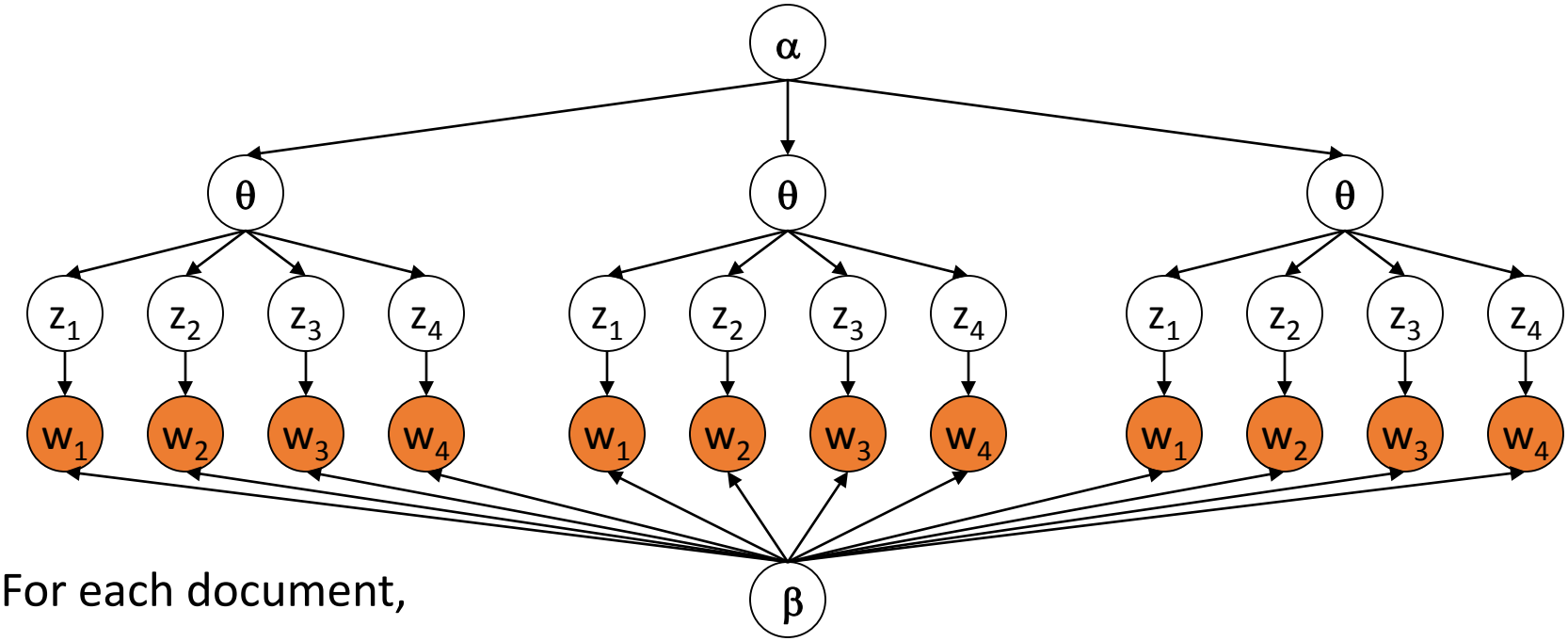
- Intuition: a version of factorial for real numbers

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$$

- Has the following property:

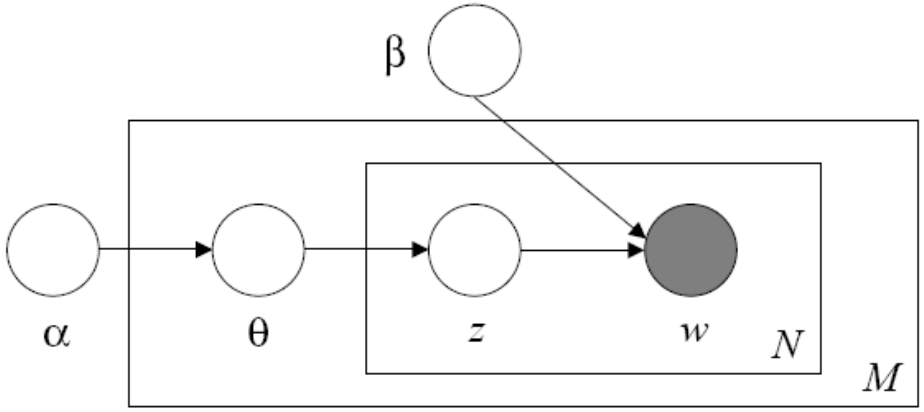
$$\Gamma(1 + x) = x\,\Gamma(x)$$

The LDA Model



- For each document,
- Choose $\theta \sim \text{Dirichlet}(\alpha)$
- For each of the N words wn:
  - Choose a topic $z_n \text{»} \text{ Multinomial}(\theta)$
  - Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

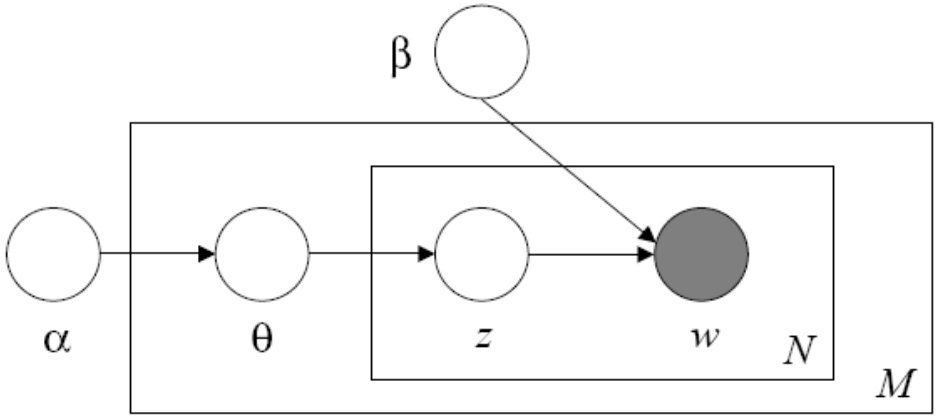
The LDA Model



For each document,

- Choose θ» Dirichlet(α)
- For each of the N words w_n:
  - Choose a topic z_n» Multinomial(θ)
  - Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

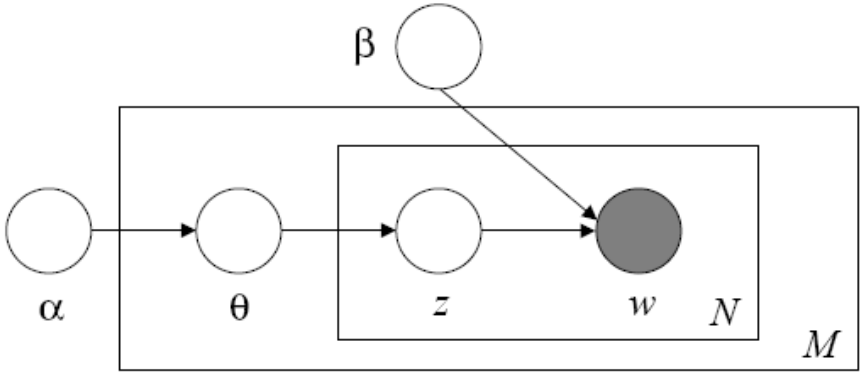
Inference

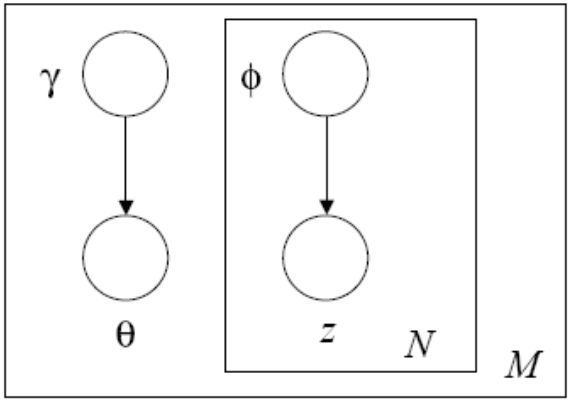


•The inference problem in LDA is to compute the posterior of the hidden variables given a document and corpus parameters α and β. That is, compute $p(\theta,z|w,\alpha,\beta)$.

•Unfortunately, exact inference is intractable, so we turn to alternatives…

Variational Inference





•In variational inference, we consider a simplified graphical model with variational parameters γ, ϕ and minimize the KL Divergence between the variational and posterior distributions.

$$(\gamma^*, \phi^*) = \arg\min_{(\gamma,\phi)} KL(q(\theta, z|\gamma, \phi)||p(\theta, z|w, \alpha, \beta))$$

Parameter Estimation

- Given a corpus of documents, we would like to find the parameters α and β which maximize the likelihood of the observed data.
- Strategy (*Variational EM)*:
  - Lower bound log p(w|α,β) by a function L(γ,ϕ;α,β)
  - Repeat until convergence:
    - Maximize L(γ,ϕ;α,β) with respect to the variational parameters γ,ϕ.
    - Maximize the bound with respect to parameters α and β.

Questions

Semi-Supervised Learning

Outline

- Fully supervised learning (traditional classification)
- Partially (semi-) supervised learning (or classification)
  - Learning with a small set of labeled examples and a large set of unlabeled examples (LU learning)
  - Learning with positive and unlabeled examples (no labeled negative examples) (PU learning).

Learning from a small labeled set and a large unlabeled set

LU learning

Unlabeled Data

- One of the bottlenecks of classification is the labeling of a large set of examples (data records or text documents).
  - Often done manually
  - Time consuming
- Can we label only a small number of examples and make use of a large number of unlabeled examples to learn?
- Possible in many cases.

Why unlabeled data are useful?

- Unlabeled data are usually plentiful, labeled data are expensive.
- Unlabeled data provide information about the joint probability distribution over words and collocations (in texts).

- We will use text classification to study this problem.

Labeled Data

Unlabeled Data

Documents containing “homework” tend to belong to the positive class

DocNo: k ClassLabel: Positive
……
……homework….
…

DocNo: m ClassLabel: Positive
……
……homework….
…

DocNo: n ClassLabel: Positive
……
……homework….
…

DocNo: x (ClassLabel: Positive)
……
……homework….
…lecture….

DocNo: y (ClassLabel: Positive)
……lecture…..
……homework….
…

DocNo: z ClassLabel: Positive
……
……homework….
……lecture….

How to use unlabeled data

- One way is to use the EM algorithm
  - EM: Expectation Maximization
- The EM algorithm is a popular iterative algorithm for maximum likelihood estimation in problems with missing data.
- The EM algorithm consists of two steps,
  - *Expectation* step, i.e., filling in the missing data
  - *Maximization* step – calculate a new maximum *a posteriori* estimate for the parameters.

Incorporating unlabeled Data with EM
(Nigam et al, 2000)

- Basic EM
- Augmented EM with weighted unlabeled data
- Augmented EM with multiple mixture components per class

Algorithm Outline

1. Train a classifier with only the labeled documents.
2. Use it to probabilistically classify the unlabeled documents.
3. Use ALL the documents to train a new classifier.
4. Iterate steps 2 and 3 to convergence.

Basic Algorithm

Algorithm EM(L, U)

1 Learn an initial naïve Bayesian classifier f from only the labeled set L (using Equations (27) and (28) in Chap. 3);
2 **repeat**
  // E-Step
3 **for** each example d_i in U **do**
4 Using the current classifier f to compute $\Pr(c_j|d_i)$ (using Equation (29) in Chap. 3).
5 **end**
  // M-Step
6 learn a new naïve Bayesian classifier f from $L \cup U$ by computing $\Pr(c_j)$ and $\Pr(w_t|c_j)$ (using Equations (27) and (28) in Chap. 3).
7 **until** the classifier parameters stabilize
Return the classifier f from the last iteration.

Fig. 5.1. The EM algorithm with naïve Bayesian classification

Basic EM: E Step & M Step

E Step:

$$\Pr(c_j \mid d_i; \hat{\Theta}) = \frac{\Pr(c_j \mid \hat{\Theta}) \Pr(d_i \mid c_j; \hat{\Theta})}{\Pr(d_i \mid \hat{\Theta})} \tag{29}$$

$$= \frac{\Pr(c_j \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j; \hat{\Theta})}{\sum_{r=1}^{|C|} \Pr(c_r \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r; \hat{\Theta})},$$

M Step:

$$\Pr(w_t \mid c_j; \hat{\Theta}) = \frac{\lambda + \sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}. \tag{27}$$

$$\Pr(c_j \mid \hat{\Theta}) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|}. \tag{28}$$

The problem

- It has been shown that the EM algorithm in Fig. 5.1 works well if the
  - The two mixture model assumptions for a particular data set are true.
- The two mixture model assumptions, however, can cause major problems when they do not hold. In many real-life situations, they may be violated.
- It is often the case that a class (or topic) contains a number of sub-classes (or sub-topics).
  - For example, the class Sports may contain documents about different sub-classes of sports, Baseball, Basketball, Tennis, and Softball.
- Some methods to deal with the problem.

Weighting the influence of unlabeled examples by factor μ

New M step:

$$\Pr(w_t \mid c_j) = \frac{\lambda + \sum_{i=1}^{|D|} \Lambda(i) N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} \Lambda(i) N_{ti} \Pr(c_j \mid d_i)}, \quad (1)$$

where

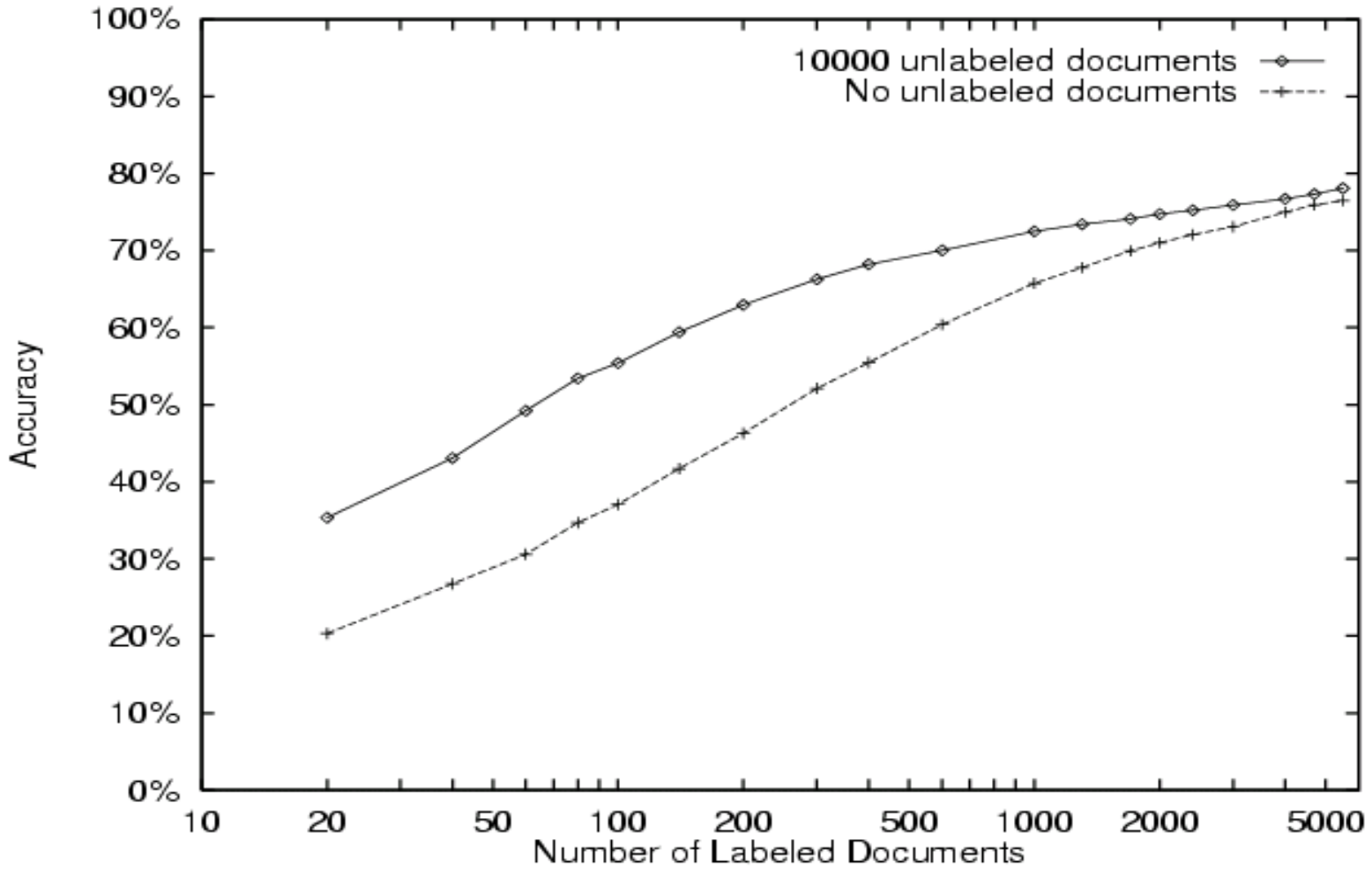
$$\Lambda(i) = \begin{cases} \mu & \text{if } d_i \in U \\ 1 & \text{if } d_i \in L. \end{cases} \quad (2)$$

The prior probability also needs to be weighted.

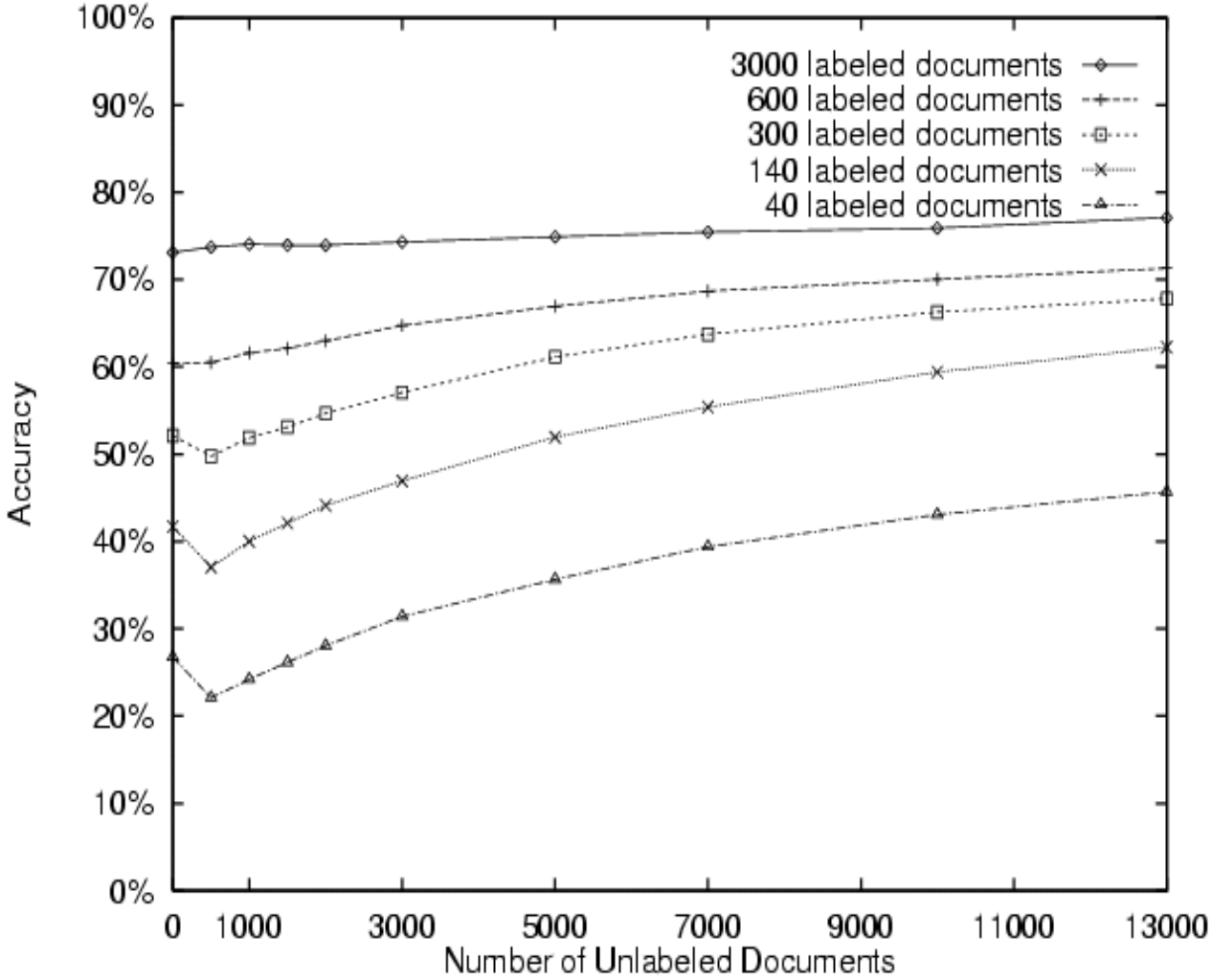
Experimental Evaluation

- Newsgroup postings
  - 20 newsgroups, 1000/group
- Web page classification
  - student, faculty, course, project
  - 4199 web pages
- Reuters newswire articles
  - 12,902 articles
  - 10 main topic categories

20 Newsgroups



20 Newsgroups



Another approach: Co-training

- Again, learning with a small labeled set and a large unlabeled set.
- The attributes describing each example or instance can be partitioned into two subsets. Each of them is sufficient for learning the target function.
  - E.g., hyperlinks and page contents in Web page classification.
- Two classifiers can be learned from the same data.

Co-training Algorithm
[Blum and Mitchell, 1998]

Given: labeled data L,

unlabeled data U

Loop:

Train h1 (e.g., hyperlink classifier) using L

Train h2 (e.g., page classifier) using L

Allow h1 to label *p* positive, *n* negative examples from U

Allow h2 to label *p* positive, *n* negative examples from U

Add these most confident self-labeled examples to L

Co-training: Experimental Results

- begin with 12 labeled web pages (academic course)
- provide 1,000 additional unlabeled web pages
- average error: learning from labeled data 11.1%;
- average error: co-training 5.0%

	Page-base classifier	Link-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

When the generative model is not suitable

- Multiple Mixture Components per Class (M-EM). E.g., a class --- a number of sub-topics or clusters.
- Results of an example using 20 newsgroup data
  - 40 labeled; 2360 unlabeled; 1600 test
  - Accuracy
    - NB 68%
    - EM 59.6%
- Solutions
  - M-EM (Nigam et al, 2000): Cross-validation on the training data to determine the number of components.
  - Partitioned-EM (Cong, et al, 2004): using hierarchical clustering. It does significantly better than M-EM.

Summary

- Using unlabeled data can improve the accuracy of classifier when the data fits the generative model.
- Partitioned EM and the EM classifier based on multiple mixture components model (M-EM) are more suitable for real data when multiple mixture components are in one class.
- Co-training is another effective technique when redundantly sufficient features are available.

Learning from Positive and Unlabeled Examples

PU learning

Learning from Positive & Unlabeled data

- Positive examples: One has a set of examples of a class *P*, and
- Unlabeled set: also has a set *U* of unlabeled (or mixed) examples with instances from *P* and also not from *P* (*negative examples*).
- Build a classifier: Build a classifier to classify the examples in *U* and/or future (test) data.
- Key feature of the problem: no labeled negative training data.
- We call this problem, PU-learning.

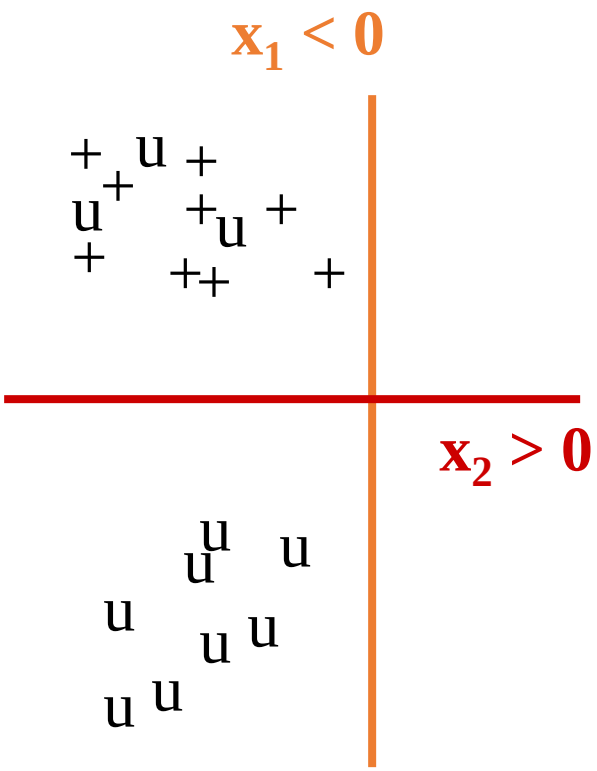
Applications of the problem

- With the growing volume of online texts available through the Web and digital libraries, one often wants to find those documents that are related to one's work or one's interest.
- For example, given a ICML proceedings,
  - find all machine learning papers from AAAI, IJCAI, KDD
  - No labeling of negative examples from each of these collections.
- Similarly, given one's bookmarks (positive documents), identify those documents that are of interest to him/her from Web sources.

Direct Marketing

- Company has database with details of its customer – positive examples, but no information on those who are not their customers, i.e., no negative examples.
- Want to find people who are similar to their customers for marketing
- Buy a database consisting of details of people, some of whom may be potential customers – unlabeled examples.

Are Unlabeled Examples Helpful?



- Function known to be either $x_1 < 0$ or $x_2 > 0$
- Which one is it?

"Not learnable" with only positive examples. However, addition of unlabeled examples makes it learnable.

Theoretical foundations

- (X, Y): X - input vector, $Y \in \{1, -1\}$ - class label.
- f : classification function
- We rewrite the probability of error

$$\Pr[f(X) \neq Y] = \Pr[f(X) = 1 \text{ and } Y = -1] + \Pr[f(X) = -1 \text{ and } Y = 1] \quad (1)$$

We have $\Pr[f(X) = 1 \text{ and } Y = -1]$

$$= \Pr[f(X) = 1] - \Pr[f(X) = 1 \text{ and } Y = 1]$$

$$= \Pr[f(X) = 1] - (\Pr[Y = 1] - \Pr[f(X) = -1 \text{ and } Y = 1]).$$

Plug this into (1), we obtain

$$\Pr[f(X) \neq Y] = \Pr[f(X) = 1] - \Pr[Y = 1] + 2\Pr[f(X) = -1 \mid Y = 1]\Pr[Y = 1] \quad (2)$$

Theoretical foundations (cont)

- $\Pr[f(X) \neq Y] = \Pr[f(X) = 1] - \Pr[Y = 1]$ (2)
  $+ 2\Pr[f(X) = -1 \mid Y = 1] \Pr[Y = 1]$
- Note that $\Pr[Y = 1]$ is constant.
- If we can hold $\Pr[f(X) = -1 \mid Y = 1]$ small, then learning is approximately the same as minimizing $\Pr[f(X) = 1]$.
- Holding $\Pr[f(X) = -1 \mid Y = 1]$ small while minimizing $\Pr[f(X) = 1]$ is approximately the same as
  - minimizing $\Pr_u[f(X) = 1]$
  - while holding $\Pr_P[f(X) = 1] \geq r$ (where r is recall Pr[f(X)=1| Y=1]) which is the same as $(\Pr_p[f(X) = -1] \leq 1 - r)$

  if the set of positive examples P and the set of unlabeled examples U are large enough.
- Theorem 1 and Theorem 2 in [Liu et al 2002] state these formally in the noiseless case and in the noisy case.

Put it simply

- A constrained optimization problem.
- A reasonably good generalization (learning) result can be achieved
  - If the algorithm tries to minimize the number of unlabeled examples labeled as positive
  - subject to the constraint that the fraction of errors on the positive examples is no more than 1-*r*.

An illustration

- Assume a linear classifier. Line 3 is the best solution.

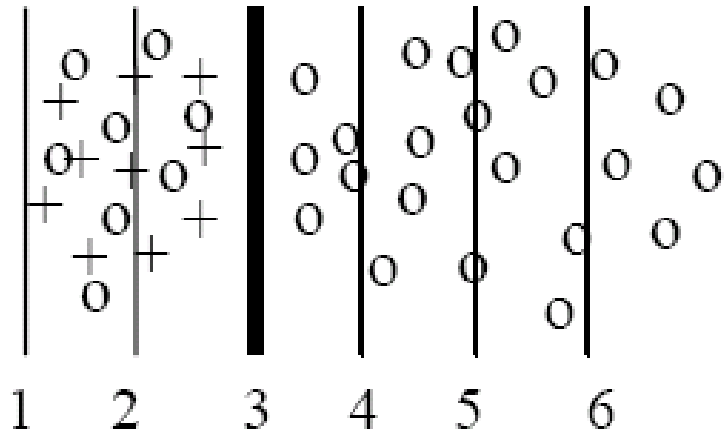


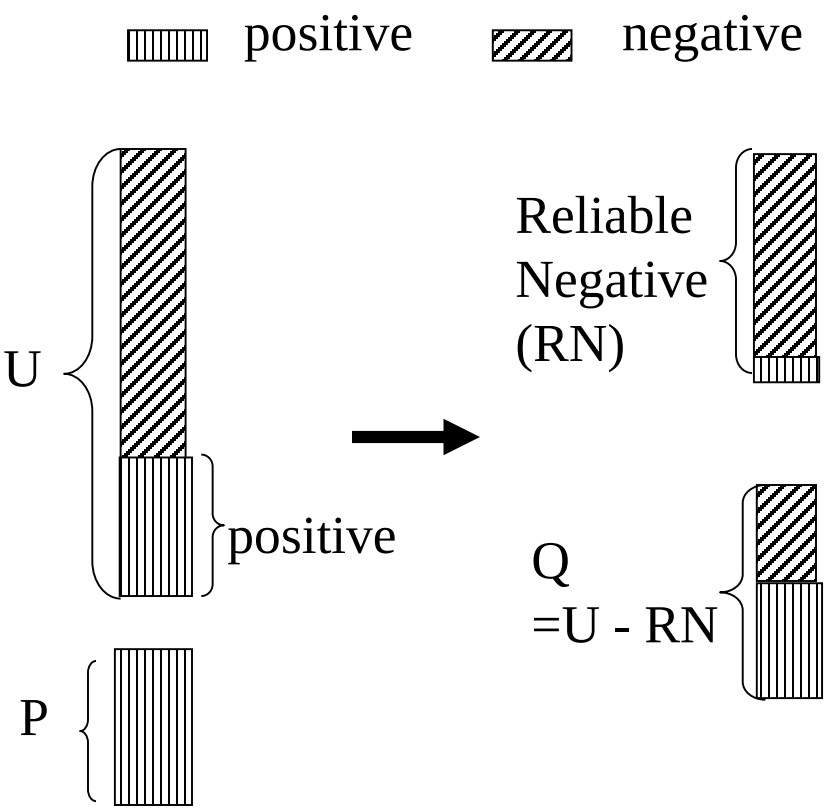
Fig. 5.6. An illustration of the constrained optimization problem

Existing 2-step strategy

- Step 1: Identifying a set of reliable negative documents from the unlabeled set.
  - S-EM [Liu et al, 2002] uses a Spy technique,
  - PEBL [Yu et al, 2002] uses a 1-DNF technique
  - Roc-SVM [Li & Liu, 2003] uses the Rocchio algorithm.
  - ...
- Step 2: Building a sequence of classifiers by iteratively applying a classification algorithm and then selecting a good classifier.
  - S-EM uses the Expectation Maximization (EM) algorithm, with an error based classifier selection mechanism
  - PEBL uses SVM, and gives the classifier at convergence. I.e., no classifier selection.
  - Roc-SVM uses SVM with a heuristic method for selecting the final classifier.

Step 1

Step 2



Using P, RN and Q to build the final classifier iteratively

or

Using only P and RN to build a classifier

Step 1: The Spy technique

- Sample a certain % of positive examples and put them into unlabeled set to act as "spies".
- Run a classification algorithm assuming all unlabeled examples are negative,
  - we will know the behavior of those actual positive examples in the unlabeled set through the "spies".
- We can then extract reliable negative examples from the unlabeled set more accurately.

Step 1: Other methods

- 1-DNF method:
  - Find the set of words W that occur in the positive documents more frequently than in the unlabeled set.
  - Extract those documents from unlabeled set that do not contain any word in W. These documents form the reliable negative documents.
- Rocchio method from information retrieval.
- Naïve Bayesian method.

Step 2: Running EM or SVM iteratively

(1) Running a classification algorithm iteratively

- Run EM using P, RN and Q until it converges, or
- Run SVM iteratively using P, RN and Q until this no document from Q can be classified as negative. RN and Q are updated in each iteration, or
- …

(2) Classifier selection.

Do they follow the theory?

- Yes, heuristic methods because
  - Step 1 tries to find some initial reliable negative examples from the unlabeled set.
  - Step 2 tried to identify more and more negative examples iteratively.
- The two steps together form an iterative strategy of increasing the number of unlabeled examples that are classified as negative while maintaining the positive examples correctly classified.

Can SVM be applied directly?

- Can we use SVM to directly deal with the problem of learning with positive and unlabeled examples, without using two steps?
- Yes, with a little re-formulation.

Support Vector Machines

- Support vector machines (SVM) are linear functions of the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, where $\mathbf{w}$ is the weight vector and $\mathbf{x}$ is the input vector.
- Let the set of training examples be $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ is an input vector and y_i is its class label, $y_i \in \{1, -1\}$.
- To find the linear function:

Minimize: $$\frac{1}{2}\mathbf{w}^T\mathbf{w}$$

Subject to: $$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, n$$

Soft margin SVM

- To deal with cases where there may be no separating hyperplane due to noisy labels of both positive and negative training examples, the soft margin SVM is proposed:

Minimize: $$\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{n}\xi_i$$

Subject to: $$y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, ..., n$$

where $C \geq 0$ is a parameter that controls the amount of training errors allowed.

Biased SVM (noiseless case)

- Assume that the first *k*-1 examples are positive examples (labeled 1), while the rest are unlabeled examples, which we label negative (-1).

Minimize: $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=k}^{n}\xi_i$

Subject to: $\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \geq 1, \quad i = 1, 2, \ldots, k-1$

$-1(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = k, k+1, \ldots, n$

$\xi_i \geq 0,\ i = k, k+1, \ldots, n$

Biased SVM (noisy case)

- If we also allow positive set to have some noisy negative examples, then we have:

Minimize: $$\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C_{+}\sum_{i=1}^{k-1}\xi_i + C_{-}\sum_{i=k}^{n}\xi_i$$

Subject to: $$y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1,2\ldots, n$$

$$\xi_i \geq 0,\ i = 1, 2, \ldots, n.$$

- This turns out to be the same as the asymmetric cost SVM for dealing with unbalanced data. Of course, we have a different motivation.

Estimating performance

- We need to estimate the performance in order to select the parameters.
- Since learning from positive and negative examples often arise in retrieval situations, we use F score as the classification performance measure $F = 2pr / (p+r)$ (p: precision, r: recall).
- To get a high F score, both precision and recall have to be high.
- However, without labeled negative examples, we do not know how to estimate the F score.

A performance criterion

- Performance criteria $pr/\Pr[Y=1]$: It can be estimated directly from the validation set as $r^2/\Pr[f(X) = 1]$
  - Recall $r = \Pr[f(X)=1 \mid Y=1]$
  - Precision $p = \Pr[Y=1 \mid f(X)=1]$

To see this

$$\Pr[f(X)=1 \mid Y=1]\ \Pr[Y=1] = \Pr[Y=1 \mid f(X)=1]\ \Pr[f(X)=1]$$

$$\Leftrightarrow \quad \frac{r}{\Pr[f(X)=1]} = \frac{p}{\Pr[Y=1]}$$ //both side times r

- Behavior similar to the F-score (= $2pr / (p+r)$)

A performance criterion (cont …)

- $r^2/\Pr[f(X) = 1]$
- r can be estimated from positive examples in the validation set.
- $\Pr[f(X) = 1]$ can be obtained using the full validation set.
- This criterion actually reflects the theory very well.

Empirical Evaluation

- Two-step strategy: We implemented a benchmark system, called LPU, which is available at http://www.cs.uic.edu/~liub/LPU/LPU-download.html
  - Step 1:
    - Spy
    - 1-DNF
    - Rocchio
    - Naïve Bayesian (NB)
  - Step 2:
    - EM with classifier selection
    - SVM: Run SVM once.
    - SVM-I: Run SVM iteratively and give converged classifier.
    - SVM-IS: Run SVM iteratively with classifier selection
- Biased-SVM (we used SVM$_{light}$ package)

Table 1: Average F scores on Reuters collection

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Step1	1-DNF	1-DNF		1-DNF		Spy	Spy	Spy	Rocchio	Rocchio	Rocchio		NB	NB	NB	NB	
Step2	EM	SVM	PEBL	SVM-IS	S-EM	SVM	SVM-I	SVM-IS	EM	SVM	SVM-I	Roc-SVM	EM	SVM	SVM-I	SVM-IS	NB
0.1	0.187	0.423	0.001	0.423	0.547	0.329	0.006	0.328	0.644	0.589	0.001	0.589	0.547	0.115	0.006	0.115	0.514
0.2	0.177	0.242	0.071	0.242	0.674	0.507	0.047	0.507	0.631	0.737	0.124	0.737	0.693	0.428	0.077	0.428	0.681
0.3	0.182	0.269	0.250	0.268	0.659	0.733	0.235	0.733	0.623	0.780	0.242	0.780	0.695	0.664	0.235	0.664	0.699
0.4	0.178	0.190	0.582	0.228	0.661	0.782	0.549	0.780	0.617	0.805	0.561	0.784	0.693	0.784	0.557	0.782	0.708
0.5	0.179	0.196	0.742	0.358	0.673	0.807	0.715	0.799	0.614	0.790	0.737	0.799	0.685	0.797	0.721	0.789	0.707
0.6	0.180	0.211	0.810	0.573	0.669	0.833	0.804	0.820	0.597	0.793	0.813	0.811	0.670	0.832	0.808	0.824	0.694
0.7	0.175	0.179	0.824	0.425	0.667	0.843	0.821	0.842	0.585	0.793	0.823	0.834	0.664	0.845	0.822	0.843	0.687
0.8	0.175	0.178	0.868	0.650	0.649	0.861	0.865	0.858	0.575	0.787	0.867	0.864	0.651	0.859	0.865	0.858	0.677
0.9	0.172	0.190	0.860	0.716	0.658	0.859	0.859	0.853	0.580	0.776	0.861	0.861	0.651	0.846	0.858	0.845	0.674

Table 2: Average F scores on 20Newsgroup collection

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Step1	1-DNF	1-DNF		1-DNF		Spy	Spy	Spy	Rocchio	Rocchio	Rocchio		NB	NB	NB	NB	
Step2	EM	SVM	PEBL	SVM-IS	S-EM	SVM	SVM-I	SVM-IS	EM	SVM	SVM-I	Roc-SVM	EM	SVM	SVM-I	SVM-IS	NB
0.1	0.145	0.545	0.039	0.545	0.460	0.097	0.003	0.097	0.557	0.295	0.003	0.295	0.368	0.020	0.003	0.020	0.333
0.2	0.125	0.371	0.074	0.371	0.640	0.408	0.014	0.408	0.670	0.546	0.014	0.546	0.649	0.232	0.013	0.232	0.611
0.3	0.123	0.288	0.201	0.288	0.665	0.625	0.154	0.625	0.673	0.644	0.121	0.644	0.689	0.469	0.120	0.469	0.674
0.4	0.122	0.260	0.342	0.258	0.683	0.684	0.354	0.684	0.671	0.690	0.385	0.682	0.705	0.610	0.354	0.603	0.704
0.5	0.121	0.248	0.563	0.306	0.685	0.715	0.560	0.707	0.663	0.716	0.565	0.708	0.702	0.680	0.554	0.672	0.707
0.6	0.123	0.209	0.646	0.419	0.689	0.758	0.674	0.746	0.663	0.747	0.683	0.738	0.701	0.737	0.670	0.724	0.715
0.7	0.119	0.196	0.715	0.563	0.681	0.774	0.731	0.757	0.660	0.754	0.731	0.746	0.699	0.763	0.728	0.749	0.717
0.8	0.124	0.189	0.689	0.508	0.680	0.789	0.760	0.783	0.654	0.761	0.763	0.766	0.688	0.780	0.758	0.774	0.707
0.9	0.123	0.177	0.716	0.577	0.684	0.807	0.797	0.798	0.654	0.775	0.798	0.790	0.691	0.806	0.797	0.798	0.714

Results of Biased SVM

Table 3: Average F scores on the two collections

	γ	Average F score of Biased-SVM	Previous best F score
Reuters	0.3	0.785	0.78
	0.7	0.856	0.845
20Newsgroup	0.3	0.742	0.689
	0.7	0.805	0.774

Summary

- Gave an overview of the theory on learning with positive and unlabeled examples.
- Described the existing two-step strategy for learning.
- Presented an more principled approach to solve the problem based on a biased SVM formulation.
- Presented a performance measure pr/P(Y=1) that can be estimated from data.
- Experimental results using text classification show the superior classification power of Biased-SVM.
- Some more experimental work are being performed to compare Biased-SVM with weighted logistic regression method [Lee & Liu 2003].