

Mining the Web

Corpus, Document and Language Models

Soumen Chakrabarti

September 16, 2019

Outline

- Aka language model (LM)
 - Assign a probability to a passage or document
 - Given a prefix, predict next character, word
 - Used in search, OCR, handwriting recognition, translation, summarization swipe typing, spelling and grammar correction, etc.
- Here we will mostly **limit to document-as-multiset**, not sequence, of words
- The term-document matrix
- Generative models, perplexity, curse of dimensionality
- Multivariate binary, Poisson, multinomial models
- Word burstiness, non-parametric and Dirichlet models

Homogeneous corpus models

Multivariate binary model

- A document *event* is just a bit vector with a 0/1 slot for each term w in the vocabulary W
- An instantiated document vector is written as x
- x_w will denote the bit corresponding to word $w \in W$

$$\begin{aligned}\Pr(x|\phi) &= \prod_{w \in W} \phi_w^{x_w} (1 - \phi_w)^{1-x_w} \\ &= \prod_{w \in x} \phi_w \prod_{w \in W, w \notin x} (1 - \phi_w),\end{aligned}$$

where “ $w \in x$ ” means “word w occurs in document x ”, i.e., $x_w = 1$

- Short documents are discouraged by the model because $|W| \gg \|x\|_1$
- Products make strong independence assumptions and greatly underestimate $\Pr(x|\Phi)$

Sparsity and smoothing

- Training corpus sets up vocabulary W
- What is the probability of a test doc with out-of-vocabulary (OOV) word?
- If $\phi_{\text{oov}} = 0$, test doc probability is also zero
- Smoothing: set aside some probability mass and apportion them among events not seen (often enough) during smoothing
- Suppose you toss a coin 4 times and get 0 heads
- Does that mean $\Pr(\text{head}) = 0$?
- How about you toss it 4 million times and get 0 heads?
- In general, if you toss a coin N times and get K heads, where K can be 0 or N , what is the probability of the next toss being a head?
- Is there a principled way to avoid 0 and 1?
- What is our prior belief about coins?

Prior and posterior estimates of coin bias ⁽¹⁾

- Even after tossing a coin a very large number of times, we do not really know its $\theta = \Pr(\text{head})$
- We can only estimate a **density** $f(\theta)$ over $\theta \in [0, 1]$
- A reasonable “zero-knowledge” **prior belief** is that $f(\theta) = 1$ for $\theta \in [0, 1]$ (the uniform prior)
- I.e., coins with all possible biases are equally likely
- (In reality we probably have more trust in the fairness of coins; i.e., f is peaked at/near $\theta = 1/2$)
- With this prior belief, we toss the coin N times and observe K heads

Prior and posterior estimates of coin bias (2)

- After the observation, our knowledge of the coin turns into a **posterior belief** $g(\theta|K, N)$
- In informal notation, $\Pr(\theta|K, N) = \Pr(K, N|\theta) \Pr(\theta) / \sum_{\theta'} \Pr(K, N|\theta') \Pr(\theta')$ by Bayes rule
- Formally,

$$g(\theta|K, N) = \frac{\binom{N}{K} \theta^K (1 - \theta)^{N-K}}{\binom{N}{K} \int_{\phi=0}^1 \phi^K (1 - \phi)^{N-K} d\phi}$$

Prior and posterior estimates of coin bias (3)

- If we insist on a point estimate based on the posterior, we might ask for its expectation (expected coin bias after observation)

$$\int_{\theta=0}^1 \theta g(\theta|K, N) d\theta = \frac{\int_{\phi=0}^1 \phi \phi^K (1 - \phi)^{N-K} d\phi}{\int_{\phi=0}^1 \phi^K (1 - \phi)^{N-K} d\phi} = \dots = \frac{K + 1}{N + 2} \neq \frac{K}{N}$$

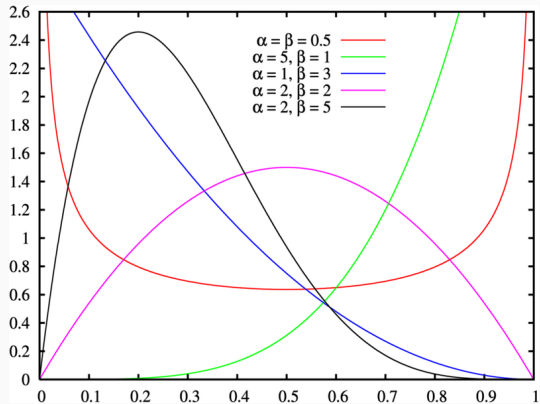
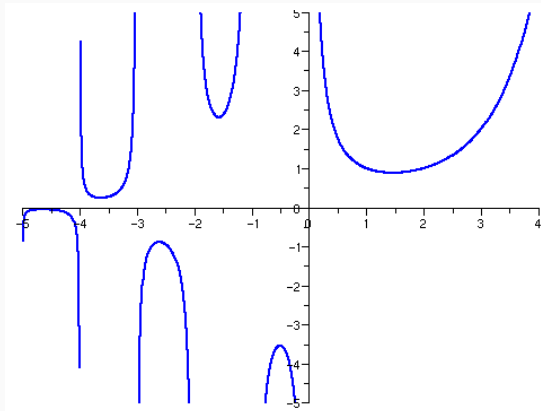
- $(K + 1)/(N + 2)$ is never 0 or 1, but can approach them as $N \rightarrow \infty$
- Due to Laplace, by way of actuarial analysis (if a person is seen alive 10,000 days ...)
- Can generalize from two events (head/tail) to > 2 events (die toss)
- Can adjust to other prior beliefs within convenient density families (most coins tend to be fair / unfair)

Gamma function, beta distribution

- The beta distribution with params α, β has density

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad 0 < x < 1$$

where $B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$



Coin toss posterior distribution

- Extending from just the mean to the whole posterior distribution
- Start with coin having prior density $B(a, b)$ over head probability parameter
- Now toss the coin to get H heads and T tails
- What is the posterior distribution over head probability?

Poisson model ⁽¹⁾

- Now we will model term counts, but continue to assume that word events are independent of other words
- A specific document event x will now be a vector of non-negative word counts, not bits
- The corpus model is expressed through one parameter for each word w : the mean count μ_w of that word in a document
- Assume that word counts are random variables X_w that follow Poisson distributions with means μ_w :

$$\Pr(X_w = z) = \frac{e^{-\mu_w} \mu_w^z}{z!}, \quad \text{for } z = 0, 1, 2, \dots$$

Poisson model (2)

- The probability of invoking the Poisson document generator and getting a count vector x is therefore

$$\begin{aligned}\Pr(x|\mu) &= \prod_{\text{all } w} \Pr(X_w = x_w) = \prod_{\text{all } w} \frac{e^{-\mu_w} \mu_w^{x_w}}{x_w!} \\ &= \exp\left(-\sum_{\text{all } w} \mu_w\right) \prod_{w \in x} \frac{\mu_w^{x_w}}{x_w!},\end{aligned}$$

Multinomial model ⁽¹⁾

- Control document length directly
- Document writer first decides the total term count (including repetitions) of the document x to be generated by drawing a random positive integer L from a suitable distribution $\text{Pr}(\ell)$
- Gets actual length ℓ_x
- Next, the writer gets a die: it has $|W|$ faces, one face for each word in the vocabulary
- When tossed, the face corresponding to word w comes up with probability θ_w
- $\sum_w \theta_w = 1$

Multinomial model (2)

- Author tosses the die ℓ_x times, and writes down the words that come up.
- As in the Poisson model, a document instance is a vector x of word counts, x_w denoting the count of word w , with $\sum_w x_w = \ell_x$
- The document *event* in this case comprises ℓ_x and the set of counts $\{x_w\}$
- The probability of this compound event is given by:

$$\begin{aligned}\Pr(\ell_x, \{x_w\}) &= \Pr(L = \ell_x) \Pr(\{x_w\} | \ell_x, \theta) \\ &= \Pr(L = \ell_x) \binom{\ell_x}{\{x_w\}} \prod_{w \in x} \theta_w^{x_w} = \Pr(L = \ell_x) \ell_x! \prod_{w \in x} \frac{\theta_w^{x_w}}{x_w!}\end{aligned}$$

where $\binom{\ell_x}{\{x_w\}} = \ell_x! / (\prod_w x_w!)$ is the multinomial coefficient

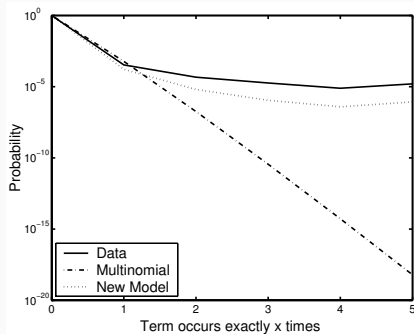
Multinomial model (3)

- **Note!** In the multinomial model, given the document length, word counts are *not* independent
- Smoothing similar to multivariate Bernoulli
- Over the training corpus, vocabulary size is W , see N word slots, of which n_w are the word w
- Then $\theta_w = (n_w + 1)/(N + W)$
- This may place excessive weight on oov word events
- Dial down to $\theta_w = (n_w + \lambda)/(N + \lambda W)$
- Tune λ by splitting the corpus into halves
 - From first half collect counts
 - Find log likelihood of second half for various λ

Modeling word burstiness

- Remember product over terms in all of binary, Poisson, multinomial
- Ordinarily, the word *xylem* is rare and unlikely to appear in a Web page sampled uniformly at random
- But *given* you have seen *xylem* once in a document, you are much more likely to see it again
- Two models of burstiness:
 - Non-parametric word marginals
 - Dirichlet word distributions

Evidence from corpus



- For both Poisson and multinomial models,
 $\Pr(\mathbf{x}) \propto \theta_w^{x_w} \implies \log \Pr(\mathbf{x}) \propto x_w \log \theta_w$
- Most $\log \theta_w \ll 1$ leading to straight line with negative slope (dashed line)
- But observed number of docs with large x_w is much greater than prediction

Word marginals from the exponential family

- Poisson and multinomial distributions belong to the exponential family:

$$\Pr(X_w = x_w | \phi_w) = g(\phi_w) f(x_w) \exp(\phi_w h(x_w))$$

- $g(\phi_w)$ is a normalizing constant equal to $1/(\sum_z f(z) \exp(\phi_w h(z)))$, so that $\sum_{z \geq 0} \Pr(z | \phi_w)$ becomes 1
- In case of Poisson distribution, $f(z) = 1/(z!)$ and $h(z) = z$, which leads to $g(\phi) = \exp(-e^\phi)$ (i.e., let e^ϕ be the “ μ ” used earlier)
- In case of the multinomial distribution, $f(z) = \binom{\ell}{z}$, where ℓ is the observed length of the document and $h(z) = z$, which leads to $g(\phi) = (1 + e^\phi)^{-\ell}$
- $h(z) = z$ means exponentially growing “surprise” on seeing a given word again and again — too extreme
- Would like to fit h (and f and thereby g) from data rather than arbitrarily guess

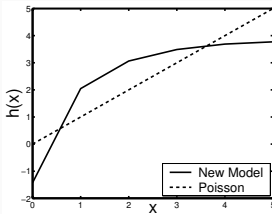
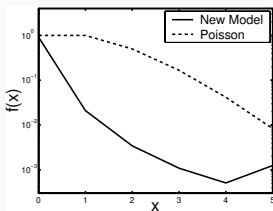
Fitting non-parametric f and h

- From term-document count matrix, build tables for f_z and h_z for all values of $z \in [0, C]$
- N is the total number of documents; $n(w, i)$ is the number of documents that mention w exactly i times; $N = \sum_{i=0}^C n(w, i)$
- Data log likelihood is

$$\begin{aligned}\log Q &= \log \prod_w \prod_{i=0}^C \Pr(i|\phi_w)^{n(w,i)} \\ &= \sum_w \sum_{i=0}^C n(w, i) \log \Pr(i|\phi_w) \\ &= \sum_w \sum_{i=0}^C n(w, i) (\log g(\phi_w) + \log f_i + \phi_w h_i)\end{aligned}$$

Estimating f , h and Φ

- Want to maximize $\log Q$ by searching for f , h , and Φ
- Initialize f , h , Φ using Poisson assumption
- Alternating optimization
- Hold f and h fixed, optimize Φ (global optimum)
- Hold Φ fixed, optimize f and h (local optima possible)



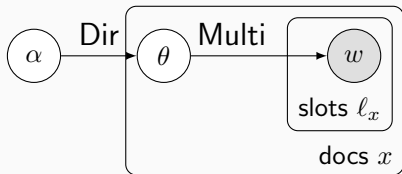
Dirichlet word hypergenerator (1)

- Generalize $B(\alpha, \beta)$ to $\text{Dir}(\alpha_1, \dots, \alpha_w, \dots) = \text{Dir}(\alpha)$:

$$\Pr(\theta|\alpha) = \frac{\Gamma(\sum_w \alpha_w)}{\prod_w \Gamma(\alpha_w)} \prod_w \theta_w^{\alpha_w-1}, \sum_w \theta_w = 1; \theta_w \geq 0 \text{ for all } w.$$

- Two step document generation

$$\Pr(x|\alpha) = \int_{\theta} \Pr(\theta|\alpha) \Pr(x|\theta) d\theta$$



Dirichlet word hypergenerator (2)

- Assuming word independence,

$$\Pr(x|\alpha) = \Pr(\ell_x) \binom{\ell_x}{\{x_w\}} \frac{\Gamma(\sum_w \alpha_w)}{\Gamma(\sum_w (x_w + \alpha_w))} \prod_w \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)}$$

- Given a corpus X , we wish to find α that maximizes the log-likelihood of the corpus

$$\arg \max_{\alpha} \log \Pr(X|\alpha) = \arg \max_{\alpha} \sum_{x \in X} \log \Pr(x|\alpha)$$

Dirichlet word hypergenerator (3)

- Simplify, with $A = \sum_w \alpha_w$ and $\sum_w x_w = \ell_x$:

$$\begin{aligned}\Pr(x|\alpha) &= \Pr(\ell_x)\ell_x! \frac{\Gamma(\sum_w \alpha_w)}{\Gamma(\sum_w (x_w + \alpha_w))} \prod_{\text{all } w} \frac{1}{x_w!} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)} \\ &= \frac{\Gamma(A)}{\Gamma(A + \ell_x)} \prod_{\text{all } w} \frac{1}{x_w!} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)}\end{aligned}$$

Parameter estimation (1)

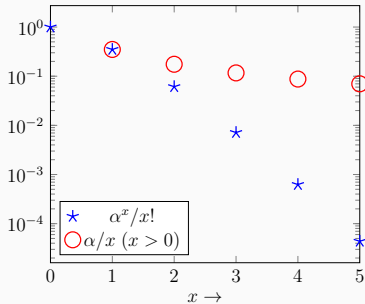
- $\lim_{a \rightarrow 0} \frac{\Gamma(x+a)}{\Gamma(a)} = a\Gamma(x)$, reasonable to assume $\alpha_w \rightarrow 0$ for most w
- Using this, can approximate

$$\begin{aligned}\Pr(x|\alpha) &= \frac{\Gamma(A)}{\Gamma(A + \ell_x)} \prod_w \frac{1}{x_w!} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)} = \prod_{w:x_w=0} 1 \prod_{w:x_w \geq 1} \frac{1}{x_w!} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)} \\ &\approx \prod_{w:x_w \geq 1} \alpha_w \frac{\Gamma(x_w)}{x_w!} = \prod_{w:x_w \geq 1} \frac{\alpha_w}{x_w}\end{aligned}$$

- Pay for α_w only once, as x_w goes from 0 to ≥ 1
- Thereafter, decrease as roughly $1/x_w$

Parameter estimation (2)

- Contrast with Poisson and multinomial: $\alpha_w^{x_w}$



- Let $\Psi(z) = d/dz(\log \Gamma(z))$ be the **digamma function**
- Shape similar to $\log z$ for positive z

Parameter estimation (3)

- Suppose $y = z + c$; then $dy = dz$ and
$$\frac{d}{dz} \log \Gamma(z + c) = \frac{d}{dy} \log \Gamma(y) \times \frac{dy}{dz} = \Psi(y) \times 1 = \Psi(z + c)$$
- Recall $\log \Pr(x|\alpha) = \blacksquare + \log \frac{\Gamma(A)}{\Gamma(A + \ell_x)} + \sum_{w:x_w \geq 1} \log \frac{\alpha_w}{x_w} =$
$$\blacksquare + \log \Gamma(A) - \log \Gamma(A + \ell_x) + \sum_{\text{all } w} \mathbb{I}[x_w \geq 1] (\log \alpha_w - \log x_w)$$
- Gradient $\frac{\partial \log \Pr(x|\alpha)}{\partial \alpha_w} = \Psi(A) - \Psi(A + \ell_x) + \frac{1}{\alpha_w} \mathbb{I}[x_w \geq 1]$
- Sum over all docs $x \in X$ and set gradient to zero to get
$$\alpha_w = \frac{\sum_{x \in X} \mathbb{I}[x_w \geq 1]}{\sum_{x \in X} \Psi(A + \ell_x) - |X| \Psi(A)}$$

Parameter estimation ⁽⁴⁾

- Sum over all w to get

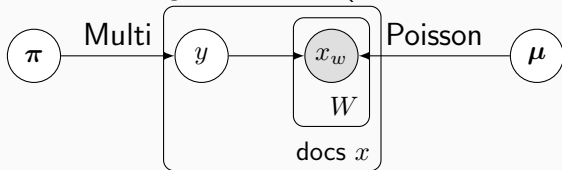
$$A = \frac{\sum_{x,w} \mathbb{I}[x_w \geq 1]}{\sum_x \Psi(A + \ell_x) - |X| \Psi(A)}$$

- Solve $A = f(A)$ using Newton's method
- Now recover all α_w
- Does this do better than Poisson and multinomial?
- Fit α_w s on train corpus
- Find log likelihood of held-out portion of corpus

Multi-topic corpus models

Probabilistic multi-topic models (1)

- To write a document, author first picks topic from a multinomial distribution over K topics $\text{Multi}(\pi_1, \dots, \pi_K)$
- Each topic y is associated with (say) a Poisson word generator with means μ_{yw} for word w ; overall model matrix $\boldsymbol{\mu} \in \mathbb{R}_+^{K \times W}$.
- Now author generates x_w (count of word w in doc x) by sampling $\text{Poisson}(\mu_{yw})$



- Goal: Given corpus (term-document count matrix) \mathbf{X} , estimate (K and) $\boldsymbol{\pi}, \boldsymbol{\mu}$

Expectation maximization

- Assume K is magically known for now
- Want $\arg \max_{\pi, \mu} \log \Pr(X|\pi, \mu)$
- For a single document, $\Pr(x|\pi, \mu) = \sum_{y=1}^K \pi_y \Pr(x|\mu_y)$, the latter probability following the Poisson distribution
- Because samples in X are iid, we can decompose
$$\log \Pr(X|\pi, \mu) = \sum_{x \in X} \log \Pr(x|\pi, \mu) = \sum_{x \in X} \log \left(\sum_y \pi_y \Pr(x|\mu_y) \right)$$
- The sum inside the log is a problem for optimization
- Let's focus on one x and consider $\log \left(\sum_y \pi_y \Pr(x|\mu_y) \right)$
- Write as $\log \left(\sum_y q(y) \frac{\pi_y \Pr(x|\mu_y)}{q(y)} \right)$ where $q(y)$ is some distribution dependent on (our fixed) x

Lower bounding the objective

- If $q(y)$ is a multinomial distribution summing to 1, then
 $\log(\sum_y q(y) f(y)) \geq \sum_y q(y) \log f(y)$ (Jensen's inequality)
- Design q to maximize the lower bound on the rhs, assuming we have fixed current estimates/guesses π^g, μ^g

$$\max_q \sum_y q(y) \log(\pi_y^g \Pr(x|\mu_y^g)) - q(y) \log q(y)$$

subject to $\sum_y q(y) = 1$

- Standard Lagrangian optimization gives

$$q_x^g(y) \propto \pi_y^g \Pr(x|\mu_y^g), \quad \text{or} \quad q_x^g(y) = \frac{\pi_y^g \Pr(x|\mu_y^g)}{\sum_k \pi_k^g \Pr(x|\mu_k^g)}$$

- This is just $\Pr^g(y|x)$, the posterior probability that x was generated from topic y given current parameter estimates

Completing the optimization

- Now put together all $x \in X$ and write a lower bound to the objective, with π, μ variable and q fixed for each x :

$$\max_{\pi, \mu} \sum_{x \in X} \sum_y q_x^g(y) \log(\pi_y \Pr(x|\mu_y))$$

(terms not involving π and μ have been dropped)

- Subject to $\sum_y \pi_y = 1$, and in general other conditions may apply on μ
- Again, standard Lagrangian optimization gives $\pi_y^* \propto \sum_{x \in X} q_x^g(y)$
- This is just the fractional count of documents in topic y
- μ can be optimized similarly, depending on the parametric form of $\Pr(x|\mu_k)$

► HW

Shortcoming of the simple mixture model

- There is uncertainty in what topic y creates a document x
- Before seeing x this is given by prior distribution π ; after seeing x this is $\Pr(y|x, \pi, \mu)$
- But the assumption is that exactly one topic generates a document
- Goes back to the use of EM in EE, e.g., in handwritten character recognition, you couldn't have written a '3' and a '8' simultaneously
- Documents are different: you can write one simultaneously about topics *cricket* and *politics*

“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.

A soft-OR (fuzzy logic) model

- Topic y causes word w to occur with a causation measure $\gamma_{yw} \in [0, 1]$ (but don't necessarily interpret as a probability)
- The extent to which topic y is activated while writing document x is $a_{xy} \in [0, 1]$

- The belief that word w will occur in the document x is then

$$b_{xw} = 1 - \prod_y (1 - a_{xy} \gamma_{yw})$$

- The goodness of fitting a document x is

$$\begin{aligned} g(x) &= \log \left(\prod_{w \in x} b_{xw} \prod_{w \notin x} (1 - b_{xw}) \right) \\ &= \sum_w \log(x_w b_{xw} + (1 - x_w)(1 - b_{xw})), \end{aligned}$$

assuming a binary $x_w \in \{0, 1\}$ model

- Can perform hill-climbing for γ, a (but slow and unreliable)

A dyadic aspect model

- So far we have thought of words w as (random) symbols and documents x as collections of symbols
- Now we will think of the document as a symbol as well, and call it d , i.e., we have random variables D and W
- Think of the joint random event (d, w) that happened some number of times recorded in the term-document matrix X
- Topic c determines which document d you compose, and what words w you use

$$\Pr(d, w) = \sum_c \Pr(c, d, w) = \sum_c \Pr(c) \Pr(d, w|c)$$

- Further simplification: D and W conditionally independent given C

$$\Pr(d, w) \approx \sum_c \Pr(c) \Pr(d|c) \Pr(w|c)$$

Extending EM to aspect model

- Let $n(d, w)$ be the nonnegative integer in row w , column d of X , the term-document matrix
- The EM update equations look like this [▶ HW](#)

$$\Pr(c|d, w) = \frac{\Pr(c, d, w)}{\Pr(d, w)} = \frac{\Pr(c) \Pr(d, w|c)}{\sum_{\gamma} \Pr(\gamma, d, w)} = \frac{\Pr(c) \Pr(d|c) \Pr(w|c)}{\sum_{\gamma} \Pr(\gamma) \Pr(d|\gamma) \Pr(w|\gamma)}$$

$$\Pr(c) = \frac{\sum_{d,w} n(d, w) \Pr(c|d, w)}{\sum_{\gamma} \sum_{d,w} n(d, w) \Pr(\gamma|d, w)}$$

$$\Pr(d|c) = \frac{\sum_t n(d, w) \Pr(c|d, w)}{\sum_{\delta} \sum_w n(\delta, w) \Pr(c|\delta, w)}$$

$$\Pr(w|c) = \frac{\sum_d n(d, w) \Pr(c|d, w)}{\sum_{\tau} \sum_d n(d, \tau) \Pr(c|d, \tau)}$$

Critique of the aspect model

- Transductive, not predictive model: need all documents in advance
- Cannot naturally evaluate the probability of a new document not in training corpus
- Number of parameters is number of topics K plus $K |W|$ plus $K |X|$: scales linearly with corpus size
- Local optima can be a problem

How to use aspect model in search

- From fixed corpus, aspect model estimates $\Pr(c)$, $\Pr(d|c)$, $\Pr(w|c)$
- Query q is a new “document” not seen with the corpus
- Need to **fold in** the new query as a document

$$\Pr(c|q, w) = \frac{\Pr(c) \Pr(q|c) \Pr(w|c)}{\sum_{\gamma} \Pr(\gamma) \Pr(q|\gamma) \Pr(w|\gamma)}$$

$$\Pr(q|c) = \frac{\sum_w n(q, w) \Pr(c|q, w)}{\sum_w n(q, w) \Pr(c|q, w) + \sum_d \sum_w n(d, w) \Pr(c|d, w)}$$

- EM for every query!
- $\Pr(q|c)$ leads to $\Pr(c|q)$, which is a kind of “projection” of q on to topic space
- Can use $\sum_c \Pr(c|q) \Pr(c|d)$ or $\sum_c \Pr(c) \Pr(d|c) \Pr(q|c)$ as similarity between q and d

Roadmap

- All corpus produced by one topic (unrealistic, poor data fit)
- Each doc generated from one (latent, uncertain) topic
- Aspect model: decompose doc-word matrix as convex combination of “per-topic layers”
- Similar to writing doc-word matrix as product of three matrices
- Related to a bipartite doc-word graph induced by the doc-word matrix
- Similar words occur in similar documents; similar documents mention similar words

Bridging the syntax gap

- Synonymy and polysemy
- Need to match documents to queries without any shared word
- Practical approach: pseudo-relevance feedback (PRF)
 - Process query from user to get top hits
 - Assume these are relevant
 - Extract keywords from these documents
 - Pad query (perhaps with smaller weight)
 - Process padded query
 - Return merged result lists
- Why stop at two queries?
- How to set magic weights?

Word-document random walks ⁽¹⁾

- Corpus as bipartite graph: word layer, document layer
- Doc node d connects to word node w if w appears in d
- Random walk with absorption:
 1. Start the walk at node v initialized to w
 2. Repeat the following sub-steps: With probability $1 - \alpha$ terminate the walk at v , and with the remaining probability α execute these half-steps:
 - 2.1 From word node v , walk to a random document node d containing word v
 - 2.2 From document node d walk to a random word node $v' \in d$Now set $v \leftarrow v'$ and loop.
- Let there be m words and n documents

Word-document random walks (2)

- Starting with the m -node word layer, walking over to the n -node document layer can be expressed with a $m \times n$ matrix A , where $A_{wd} = \Pr(d|w)$
- Each row of A adds up to 1 by design
- Once we are at the document layer, the transition back to the word layer can be represented with a $n \times m$ matrix B , where $B_{dw} = \Pr(w|d)$
- Each row of B adds up to 1 by design
- In general $B \neq A'$
- The overall transition from words back to words is then represented by the matrix product $C = AB$, where C is $m \times m$
- Rows of C add up to one as well

Word-document random walks (3)

- Starting from word w , the probability that the process stops at word q after k steps is given by

$$(1 - \alpha)\alpha^k(C^k)_{wq}$$

where $(C^k)_{wq}$ is the (w, q) -entry of the matrix C^k

- Summing over all possible non-negative k , we get

$$\begin{aligned} t(q|w) &= (1 - \alpha)(\mathbb{I} + \alpha C + \cdots + \alpha^k C^k + \cdots)_{wq} \\ &= (1 - \alpha)(\mathbb{I} - \alpha C)_{wq}^{-1} \end{aligned}$$

► HW

- For $0 < \alpha < 1$, because rows of C add up to 1, $(\mathbb{I} - \alpha C)^{-1}$ will always exist
- Parameter $\alpha \in (0, 1)$ controls the amount of diffusion

Word-document random walks ⁽⁴⁾

$w = \text{ebolavirus}$, **Web corpus**: virus, ebola, hoax, viruses, outbreak, fever, disease, haemorrhagic, gabon, infected, aids, security, monkeys, hiv, zaire

$w = \text{starwars}$, **Web corpus**: star, wars, rpg, trek, starwars, movie, episode, movies, war, character, tv, film, fan, reviews, jedi

$w = \text{starwars}$, **TREC corpus**: star, wars, soviet, weapons, photo, army, armed, film, show, nations, strategic, tv, sunday, bush, series

- Starting at given w , top-scoring qs make eminent sense
- Depends on corpus, naturally

Matrix factorization

- Recall model $\Pr(d, w) = \sum_c \Pr(c) \Pr(d|c) \Pr(w|c)$
- If $X \in \mathbb{R}^{D \times W}$ is the corpus matrix, this suggests **factoring** it as $X = U\Sigma V^\top$ where ...
- $\Pr(d|c) \rightarrow U \in \mathbb{R}^{D \times C}$ gives the topic decomposition/projection of each doc
- $\Pr(w|c) \rightarrow V \in \mathbb{R}^{W \times C}$ gives the word model of every topic
- $\Sigma = \text{diag}(\Pr(c)) \in \mathbb{R}^{C \times C}$ is a diagonal matrix of cluster priors
- In the aspect model, all U, Σ, V were non-negative
- (and various slices added up to 1)
- Suppose we remove these constraints
- A natural loss to minimize would be $\|X - U\Sigma V^\top\|_F^2$, the square of the Frobenius error (sum of squares of elementwise errors)

Eigensystem of word-document matrix

- Mild change of notation: m terms, n documents, term-document matrix $A \in \mathbb{R}^{m \times n}$
- $C = AA^\top$ is symmetric
- Term-document bipartite walk starting with initial 'presence' vector $x \in \mathbb{R}^m$ results in $x, x(AA^\top), x(AA^\top)^2, x(AA^\top)^3$, etc.
- Normalize (say) $\|x\|_2$ to 1 after every iteration
- Power method, finds dominant (left) eigenvector $q_{\cdot 1}$ of C , with $q_{\cdot 1}C = \mu_1 q_{\cdot 1}$
- There are m (row) eigenvectors that can be stacked and written as $Q^\top C = MQ^\top$, or $CQ = QM$
- If the eigenvectors of C (columns of Q) are linearly independent, Q has an inverse, $\therefore CQQ^{-1} = C = QMQ^{-1} = QMQ^\top$

Singular value decomposition

- Similarly $D = A^\top A \in \mathbb{R}^{n \times n}$ has an eigen-decomposition $DR = R\Lambda$
- $A^\top A r_{\cdot j} = \lambda_j r_{\cdot j}$, $r_{\cdot j} \in \mathbb{R}^{n \times 1}$
- ▶ HW $\lambda_j \geq 0$, so let $\sigma_j = \sqrt{\lambda_j}$ and $u_{\cdot j} = \frac{A r_{\cdot j}}{\sigma_j} \in \mathbb{R}^{m \times 1}$
- ▶ HW $u_{\cdot j}^\top u_{\cdot j} = 1$, $u_{\cdot j}^\top u_{\cdot k} = 0$ for $k \neq j$
- “Fill out” U to a $m \times m$ matrix \underline{U} with orthonormal columns
- Let $V = R$
- ▶ HW What is $\underline{U}^\top A V = \Sigma \in \mathbb{R}^{m \times n}$?
- This leads to the decomposition $A = \underline{U} \Sigma V^\top$
- Implications for text search

Comparison with Principal Component Analysis

- Mean of row i is $\mu_i = \sum_{k=1}^n A_{ik}$
- Covariance of rows i and j is $\frac{1}{n-1} \sum_{k=1}^n (A_{ik} - \mu_i)(A_{jk} - \mu_j)$
- Subtract μ_i from every element of i th row of A to get matrix B
- Covariance can be written as $\frac{1}{n-1} B^\top B$
- PCA finds eigen system of B and projects terms to space spanned by first 2–3 eigenvectors
- Related to plotting the first 2–3 columns of U in SVD
- Mean-shifting destroys sparsity

How to use SVD/LSI in search

- Document-term matrix $A_{m \times n}$ decomposed as $U\Sigma V^T$
- Each row of V gives a r -dimensional representation \hat{d} of a doc d originally in n -dim space
- Query q is an $m \times 1$ vector in document space
- Project to “LSI space” using

$$\hat{q} = \Sigma_{r \times r}^{-1} U'_{r \times m} q_{m \times 1}$$

- Now \hat{q} and each \hat{d} are comparable
- ⊖ \hat{q}, \hat{d} are not sparse in r -dim
- ⊕ “Fill” achieves bridging across syntax gap

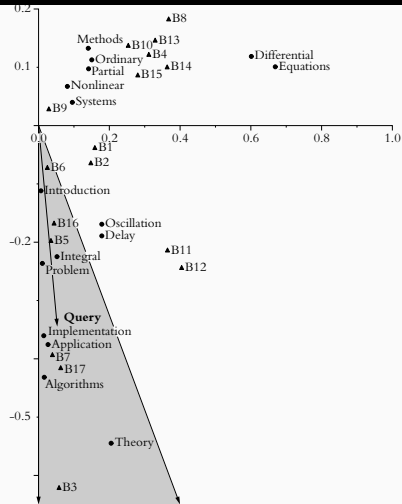
Also see: Holger Bast, Debapriyo Majumdar: Why spectral retrieval works. SIGIR 2005: 11–18.

SVD/LSI example: Corpus

Label	Titles
B1	A Course on <u>Integral Equations</u>
B2	Attractors for <u>Semigroups</u> and <u>Evolution Equations</u>
B3	Automatic Differentiation of <u>Algorithms: Theory, Implementation, and Application</u>
B4	Geometrical Aspects of <u>Partial Differential Equations</u>
B5	Ideals, Varieties, and <u>Algorithms</u> –An <u>Introduction</u> to Computational Algebraic Geometry and Commutative Algebra
B6	<u>Introduction</u> to Hamiltonian Dynamical <u>Systems</u> and the <i>N</i> -Body <u>Problem</u>
B7	Knapsack Problems: <u>Algorithms</u> and Computer Implementations
B8	Methods of Solving Singular <u>Systems</u> of <u>Ordinary Differential Equations</u>
B9	<u>Nonlinear Systems</u>
B10	<u>Ordinary Differential Equations</u>
B11	<u>Oscillation Theory</u> for Neutral <u>Differential Equations</u> with <u>Delay</u>
B12	<u>Oscillation Theory</u> of <u>Delay Differential Equations</u>
B13	Pseudodifferential Operators and <u>Nonlinear Partial Differential Equations</u>
B14	Sync <u>Methods</u> for Quadrature and <u>Differential Equations</u>
B15	Stability of Stochastic <u>Differential Equations</u> with Respect to Semi-Martingales
B16	The Boundary <u>Integral</u> Approach to Static and Dynamic Contact <u>Problems</u>
B17	The Double Mellin–Barnes Type <u>Integrals</u> and Their <u>Applications</u> to <u>Convolution Theory</u>

(a)

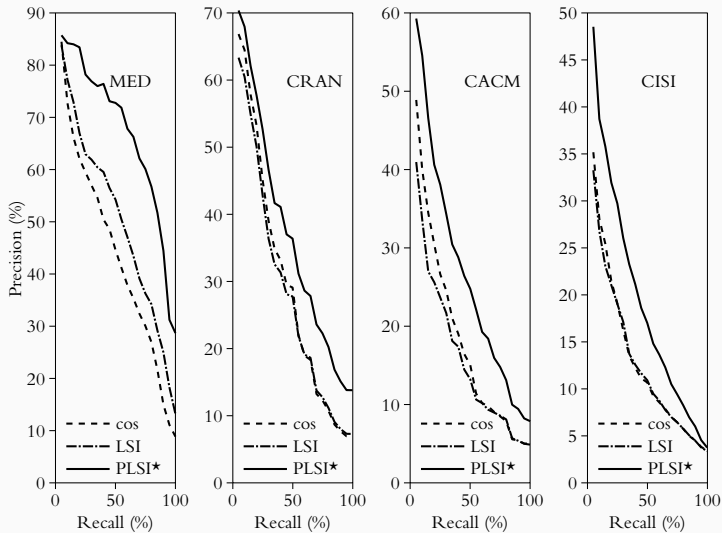
SVD/LSI example: Embedding



(b)

- Columns of U and V plotted in same “LSI space”
- Ordinary and partial drawn close together
- Implementation and application
- Introduction neither here nor there

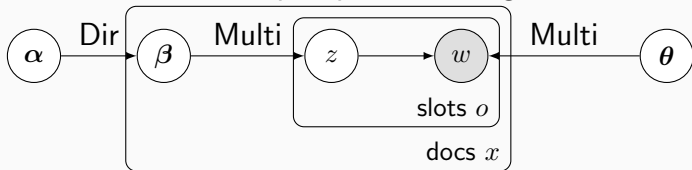
Cosine vs. SVD/LSI vs. PLSI



Nonnegative matrix factorization

A three-level generative model

- Author tosses a Dirichlet hypergenerator $\text{Dir}(\alpha)$ to get β
- This induces multinomial topic generator $\text{Multi}(\beta)$
- Let the word at token offset o in the document x be x_o
- Now, for each word x_o in the document, the author tosses $\text{Multi}(\beta)$ to get a topic z
- And then tosses a topic-specific word generator with parameters $(\theta_{z,w})$



- Sometimes even θ generated from another Dirichlet
- Only $O(kW)$ model parameters, does not scale with $|X|$

LDA document generation probabilities (1)

- If β were given for a document, it would be easy to write down the probability of the document:

$$\Pr(x|\beta, \theta) = \prod_{o=1}^O \sum_{z_o=1}^k \Pr(z_o|\beta) \Pr(x_o|\theta_{z_o}) = \prod_o \sum_{z_o=1}^k \beta_{z_o} \theta_{z_o, x_o}$$

- Because we do not know β , we must sum the above over all possible β s:

$$\begin{aligned} \Pr(x|\alpha, \theta) &= \int_{\beta} \Pr(\beta|\alpha) \left(\prod_o \sum_{z_o=1}^k \Pr(x_o|\theta_{z_o}) \right) d\beta \\ &\dots = \frac{\Gamma(\sum_y \alpha_y)}{\prod_y \Gamma(\alpha_y)} \int_{\beta} \left(\prod_{y=1}^k \beta_y^{\alpha_y-1} \right) \left(\prod_o \sum_{z_o=1}^k \beta_{z_o} \theta_{z_o, x_o} \right) d\beta \end{aligned}$$

LDA document generation probabilities (2)

- Given a corpus X of documents drawn iid, $\Pr(X|\alpha, \theta)$ is

$$\prod_{x \in X} \left[\frac{\Gamma(\sum_y \alpha_y)}{\prod_y \Gamma(\alpha_y)} \int_{\beta} \left(\prod_{y=1}^k \beta_y^{\alpha_y - 1} \right) \left(\prod_o \sum_{z_o=1}^k \beta_{z_o} \theta_{z_o, x_o} \right) d\beta \right]$$

- Given X , find $\arg \max_{\alpha, \theta} \log \Pr(X|\alpha, \theta)$
- If β were given for a document, it would be easy to write down the probability of the document:

$$\Pr(x|\beta, \theta) = \prod_{o=1}^O \sum_{z_o=1}^k \Pr(z_o|\beta) \Pr(x_o|\theta_{z_o}) = \prod_o \sum_{z_o=1}^k \beta_{z_o} \theta_{z_o, x_o}$$

LDA document generation probabilities (3)

- Because we do not know β , we must sum the above over all possible β s:

$$\begin{aligned}\Pr(x|\alpha, \theta) &= \int_{\beta} \Pr(\beta|\alpha) \left(\prod_o \sum_{z_o=1}^k \Pr(x_o|\theta_{z_o}) \right) d\beta \\ \dots &= \frac{\Gamma(\sum_y \alpha_y)}{\prod_y \Gamma(\alpha_y)} \int_{\beta} \left(\prod_{y=1}^k \beta_y^{\alpha_y-1} \right) \left(\prod_o \sum_{z_o=1}^k \beta_{z_o} \theta_{z_o, x_o} \right) d\beta\end{aligned}$$

- Given a corpus X of documents drawn iid, $\Pr(X|\alpha, \theta)$ is

$$\prod_{x \in X} \left[\frac{\Gamma(\sum_y \alpha_y)}{\prod_y \Gamma(\alpha_y)} \int_{\beta} \left(\prod_{y=1}^k \beta_y^{\alpha_y-1} \right) \left(\prod_o \sum_{z_o=1}^k \beta_{z_o} \theta_{z_o, x_o} \right) d\beta \right]$$

- Vector of z_o values over all O positions written as $\vec{z} \in \{1, \dots, k\}^O$

LDA model estimation from corpus

- Given X , find $\arg \max_{\alpha, \theta} \log \Pr(X|\alpha, \theta)$
- Two popular approaches
 - Extend EM to two latent variables β, \vec{z}
 - Use Gibb's sampling

EM for Dirichlet topic mixture (1)

$$\begin{aligned}\log \int_{\beta} \sum_{\vec{z}} p(\beta, \vec{z}, x | \alpha, \theta) d\beta &= \log \int_{\beta} \sum_{\vec{z}} q(\beta, \vec{z}) \frac{p(\beta, \vec{z}, x | \alpha, \theta)}{q(\beta, \vec{z})} d\beta \\ &\geq \int_{\beta} \sum_{\vec{z}} q(\beta, \vec{z}) \log p(\beta, \vec{z}, x | \alpha, \theta) d\beta - \int_{\beta} \sum_{\vec{z}} q(\beta, \vec{z}) \log q(\beta, \vec{z}) d\beta\end{aligned}$$

where $\int_{\beta} \sum_{\vec{z}} q(\beta, \vec{z}) = 1$

We will choose $q(\beta, \vec{z})$ to maximize the rhs

Model $q(\beta, \vec{z})$ as a product of simpler distributions $\text{Dir}(\beta | \gamma) \prod_o \text{Multi}(z_o | \phi[o])$

Here $\phi[o] \in \Delta^k$, the unit simplex over k topics

EM for Dirichlet topic mixture (2)

For each given x , we have to solve (E-step)

$$\max_{\gamma, \{\phi[o]\}} \int_{\beta} \sum_{\vec{z}} q(\beta, \vec{z}) \log p(\beta, \vec{z}, x | \alpha, \theta) d\beta - \int_{\beta} \sum_{\vec{z}} q(\beta, \vec{z}) \log q(\beta, \vec{z}) d\beta \quad (\text{E})$$

Remember the optimization variables are ‘personalized’ to the specific doc x , so we might call the weights $\gamma_x, \{\phi_x[o] : o = 1, \dots, O\}$

► HW Show that optimal choices are

$$\begin{aligned} \phi_x[o, z] &\propto \theta_{z, x_o} \exp \left(\Psi(\gamma_z) - \Psi \left(\sum_y \gamma_y \right) \right) \\ \gamma_x[z] &= \alpha_z + \sum_o \phi_x[o, z] \end{aligned}$$

Here α and γ are values from the previous iteration

EM for Dirichlet topic mixture (3)

Replace $\gamma_x, \{\phi_x[o] : o = 1, \dots, O\}$ with optimal values in (E), sum over all $x \in X$, and maximize over α and γ (M-step)

► HW Show that (locally) optimal updates are defined by

$$\theta_{z,w} \propto \sum_{x \in X} \sum_o \phi_x[o, z] \mathbb{I}[x_o = w] \quad \forall z \in [k], w \in [W]$$

$$\frac{\partial L}{\partial \alpha_z} = |X| \left(\Psi \left(\sum_y \alpha_y \right) - \Psi(\alpha_z) \right) + \sum_{x \in X} \left(\Psi(\gamma_x[z]) - \Psi \left(\sum_y \gamma_x[y] \right) \right) \quad \forall z \in [k]$$

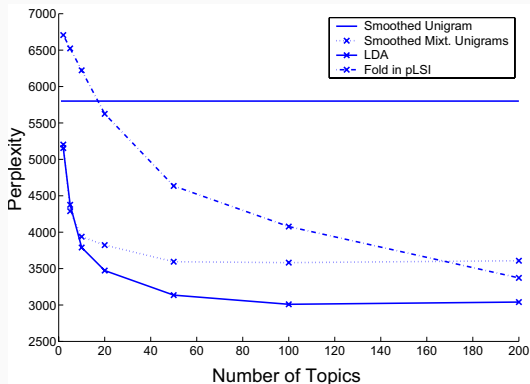
(closed form for $\theta_{z,w}$ and gradient update for α_z)

Somewhat complicated, and inflexible if model is tweaked

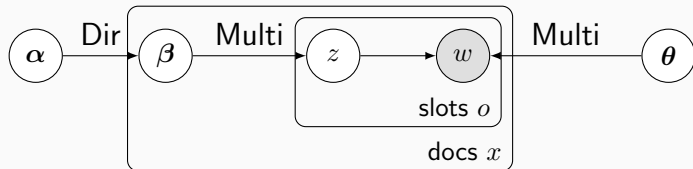
LDA evaluation

- How to measure if LDA fits corpus better than other models?
- **Perplexity**: How surprised are you seeing a new document, armed with an estimated model Θ ?

$$\text{perplexity}(X) = \exp \left(\frac{\sum_{x \in X} -\log \Pr_{\Theta}(x)}{|X|O} \right)$$



“Collapsed Gibbs sampling” or “heat bath” approach



- If all $z_{x,o}, w_{x,o}$ are fixed, can sample θ (pick most likely values)
- If all β_x are fixed, can estimate α (ditto)
- If α and all $z_{x,o}$ are fixed, can estimate β
- If all β_x and θ are fixed, can (re)sample z , say one $z_{x,o}$

Iterative scheme to update all latent variables, starting with some initial values

Co-clustering and cross-associations

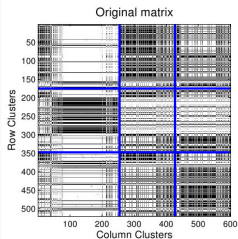
- Binary term-document matrix $\{0, 1\}^{m \times n}$
- m terms, one per row; n docs, one per column

Hypothesis about data generation:

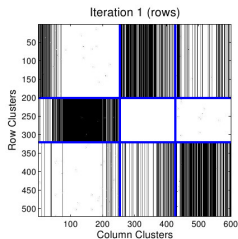
- Start with a $k \times \ell$ matrix of probabilities $p_{i,j}(0) = 1 - p_{i,j}(1)$
- Fix two groupings $\mu : \{1, 2, \dots, m\} \rightarrow \{1, 2, \dots, k\}$ and $\nu : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, \ell\}$, with typically $k \ll m$ and $\ell \ll n$
- Let $A_{(i,j)}$ be elements $A(q, r)$ such that $\mu(q) = i$ and $\nu(r) = j$
- Toss coin with head probability $p_{i,j}(1)$ to fill each of $n(i, j)$ cells in $A_{(i,j)}$

The reverse problem: Given A , find $k, \ell, \mu, \nu, p_{i,j}$

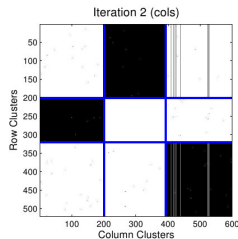
Another view of the reverse problem



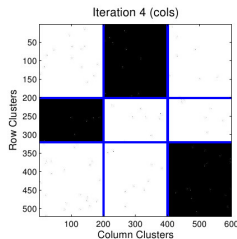
(a) Original groups



(b) Row shifts (Step 2)



(c) Column shifts (Step 4)



(d) Column shifts (Step 4)

- Given matrix A
- Permute the rows and columns
- Until a block structure emerges
- Where each block is well-explained by a single coin

Cost of compressing a block

Iterative reassignment

Applications of corpus modeling

Applications of corpus modeling

- Already seen how SVD and aspect models bridge gap between distinct but similar words
- Probabilistic relevance ranking
 - Each doc D defines a word generating distribution θ_D
 - Query Q is generated from this distribution
 - $\Pr(Q|\theta_D)$ indicates relevance of doc D to query Q
- Clustering and scatter/gather
- Balancing relevance and diversity/novelty

Ponte and Croft proposal

- Score document D wrt a query Q (each interpreted as a set or multiset of words) by estimating $\Pr(Q|D)$
- Multivariate binary model

$$\Pr(Q|\theta_D) = \prod_{q \in Q} \Pr(q|D) \prod_{q \notin Q} (1 - \Pr(q|D))$$

(penalty for dropping terms in D ?)

- Multinomial model

$$\Pr(Q|\theta_D) \propto \prod_{q \in Q} \Pr(x_q|D)$$

Smoothing

- Any word in Q not in D implies D disqualified completely
- Word not in D but in the collection/corpus \mathcal{C}

$$\Pr(w|D) = (1 - \lambda) \frac{c(w, D)}{|D|} + \lambda \Pr(w|\mathcal{C})$$

- Dirichlet prior/smoothing

$$\Pr(w|D) = \frac{c(w, D) + \mu \Pr(w|\mathcal{C})}{|D| + \mu}$$

- Bayesian scoring

$$\Pr(Q|D) = \int \Pr(Q|\theta_D) \Pr(\theta_D|D) d\theta_D$$

Assuming term independence and using conjugate priors makes this tractable

KL divergence scoring

- Score of doc D wrt query Q

$$\begin{aligned} -\text{KL}(\theta_Q \parallel \theta_D) &= - \sum_{w \in V} \text{Pr}(w|\theta_Q) \log \frac{\text{Pr}(w|\theta_Q)}{\text{Pr}(w|\theta_D)} \\ &\propto_Q \sum_{w \in V} \text{Pr}(w|\theta_Q) \log \text{Pr}(w|\theta_D) \end{aligned}$$

- New headache: estimate θ_Q (from very short Q)
- Use relevance feedback?

Text search as translation

- Long-standing goal of Information Retrieval: return documents with words *related to* query words, without damaging precision
- If q ranges over words in query Q , and w ranges over all words in the corpus vocabulary, we can write

$$\Pr(Q|D) = \prod_{q \in Q} \sum_w t(q|w) \Pr(w|\theta_D)$$

assuming conditional independence between query words

- $t(q|w)$ is the probability that a corpus w gets “translated” into query word q (e.g., $q = \text{random}$ and $w = \text{probability}$)
- One possibility is to use word embeddings from SVD or word2vec etc., and define $t(q|w) \propto \exp(\vec{q} \cdot \vec{w})$

Text clustering example

<input type="checkbox"/> Cluster 1 Size: 8	key army war francis spangle banner air song scott word poem british
<ul style="list-style-type: none"><input type="radio"/> Star-Spangled Banner, The<input type="radio"/> Key, Francis Scott<input type="radio"/> Fort McHenry<input type="radio"/> Arnold, Henry Harley<input type="radio"/> National Anthem	
<input type="checkbox"/> Cluster 2 Size: 68	film play career win television role record award york popular stage p
<ul style="list-style-type: none"><input type="radio"/> Burstyn, Ellen<input type="radio"/> Stanwyck, Barbara<input type="radio"/> Berle, Milton<input type="radio"/> Zukor, Adolph<input type="radio"/> Broadway, The	
<input type="checkbox"/> Cluster 3 Size: 97	bright magnitude cluster constellation line type contain period spectr
<ul style="list-style-type: none"><input type="radio"/> star<input type="radio"/> Galaxy, The<input type="radio"/> extragalactic systems<input type="radio"/> interstellar matter<input type="radio"/> cluster, star	
<input type="checkbox"/> Cluster 4 Size: 67	astronomer observatory astronomy position measure celestial telescop
<ul style="list-style-type: none"><input type="radio"/> astronomy and astrophysics<input type="radio"/> astrometry<input type="radio"/> Agena<input type="radio"/> astronomical catalogs and atlases<input type="radio"/> Herschel, Sir William	
<input type="checkbox"/> Cluster 5 Size: 10	family species flower animal arm plant shape leaf brittle tube foot hor
<ul style="list-style-type: none"><input type="radio"/> blazing star<input type="radio"/> brittle star<input type="radio"/> bishop's cap<input type="radio"/> feather star	

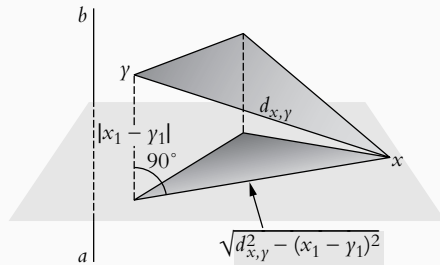
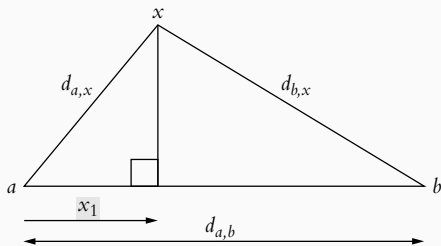
Intrinsic and extrinsic representations

- We have large assumed that an entity (document) has some intrinsic representation
- I.e., exists independent of other entities
- In some applications, intrinsic features not available or sufficient
- Extrinsic judgment of similarity or distances between entities available
- E.g., metric distance measure
- Goal: embed entities in a low-dimensional geometric space (for visualization, say)

Multidimensional scaling (MDS)

- Find a “direction” on which the “projection” of entities are well-separated
- “Project” entities to this direction/line to get first “coordinate”
- “Project” entities to “hyperplane perpendicular to line”
- Recurse in one fewer dimension

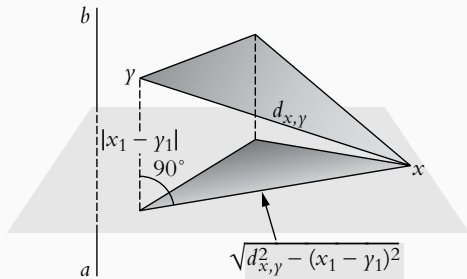
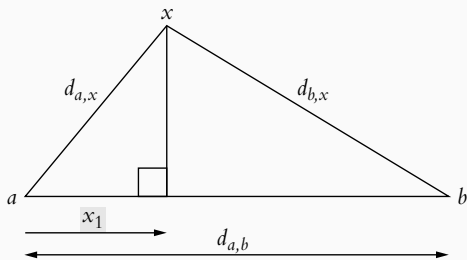
Multidimensional scaling (MDS)



- Initial direction heuristic: find farthest point pair (or some approximation of that) — points a, b with “distance” $d_{a,b}$
- For any other point x we know $d_{a,x}$ and $d_{b,x}$
- Using cosine rule, get

$$d_{b,x}^2 = d_{a,x}^2 + d_{a,b}^2 - 2x_1 d_{a,b} \quad \implies \quad x_1 = \frac{d_{a,x}^2 + d_{a,b}^2 - d_{b,x}^2}{2 d_{a,b}}$$

Multidimensional scaling (MDS)



- Projection to hyperplane perpendicular to pivot line
- Consider points x and y with distance $d_{x,y}$, first coordinates x_1 and y_1 , and projections x' , y' on the hyperplane
- By the Pythagorean theorem, the new distance d' on the hyperplane is

$$d'_{x',y'} = \sqrt{d_{x,y}^2 - (x_1 - y_1)^2}$$

k -Means and self-organizing maps

- Representative vector μ_c with each cluster c
- Cluster represented as a point in 2d space
- Cluster c has neighborhood $N(c)$
- Proximity function $h(\gamma, c)$, which tells us how close a γ is to c ; $h(c, c) = 1$
- If document d is closest to c_d , the update contribution from d should apply not only to c_d but to all clusters $\gamma \in N(c_d)$

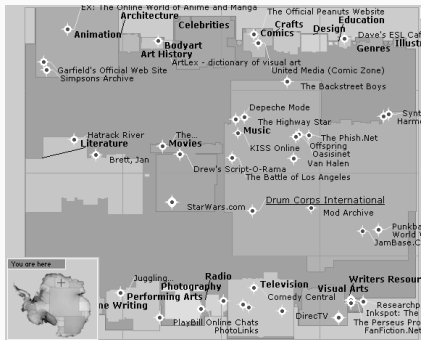
$$\mu_\gamma \leftarrow \mu_\gamma + \eta h(\gamma, c_d)(d - \mu_\gamma)$$

- η is a learning rate to stabilize μ s

SOM example



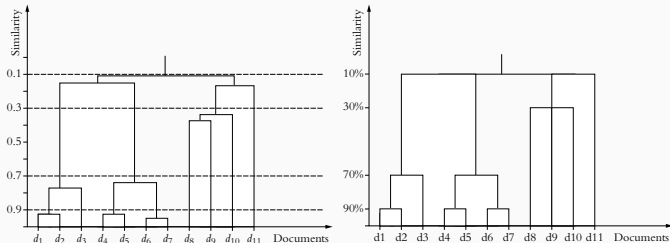
(a)



(b)

- Broad topics settle into contiguous regions

Bottom-up agglomerative clustering



- 1: let each document d be in a singleton group $\{d\}$
- 2: let G be the set of groups
- 3: **while** $|G| > 1$ **do**
- 4: choose $\Gamma, \Delta \in G$ according to some measure of similarity $s(\Gamma, \Delta)$
- 5: remove Γ and Δ from G
- 6: let $\Phi = \Gamma \cup \Delta$
- 7: insert Φ into G

Cluster merge strategies

- Merit for merging Γ and Δ
- Self-similarity of $\Gamma \cup \Delta$

$$s(\Phi) = \frac{1}{\binom{|\Phi|}{2}} \sum_{d_1, d_2 \in \Phi} s(d_1, d_2) = \frac{2}{|\Phi| (|\Phi| - 1)} \sum_{d_1, d_2 \in \Phi} s(d_1, d_2)$$

- TFIDF cosine measure is commonly used for interdocument similarity $s(d_1, d_2)$
- Maintain “unnormalized group profile vector” $p(\Phi) = \sum_{d \in \Phi} \vec{d}$ (vector sum) and number of documents

$$s(\Phi) = \frac{\langle p(\Phi), p(\Phi) \rangle - |\Phi|}{|\Phi| (|\Phi| - 1)}$$

$$p(\Gamma \cup \Delta) = \langle p(\Gamma), p(\Gamma) \rangle + \langle p(\Delta), p(\Delta) \rangle + 2 \langle p(\Gamma), p(\Delta) \rangle$$

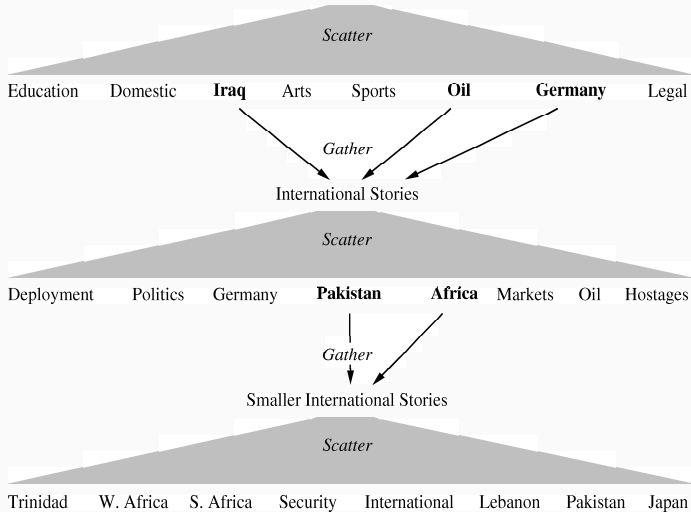
- $O(n^2 \log n)$ time with some assumptions

Scatter-gather approach

- Built around hierarchical agglomerative clustering algo
- Philosophy: need both TOC and index in a book
- Start with query results and cluster them
- User picks one or more clusters
- Recluster the union of chosen clusters
- May reveal “orthogonal dimension” of similarity
- Rinse and repeat

Scatter-gather example

New York Times News Service, August 1990



Redundancy, diversity, marginal relevance

- Vector space model: two documents similar to each other are both relevant or irrelevant wrt a query
- I.e., their scores and ranks should be similar
- You get only 10 slots, don't waste on similar docs
- **Marginal relevance** of a doc given what user has already seen
- Already seen \approx above it in the ranked list (not really)
- Two classes of approaches
 - Use conventional retrieval, then cluster top responses by similarity, and present exemplars from clusters
 - Directly optimize response list for marginal relevance

Hedging our bets: Max marginal relevance

- PRF assumes that documents similar to each other are equally relevant or irrelevant to a query
- And that top hits are good and have relevant words useful for padding
- What if the first hit is terribly wrong?
- What if the top 3 are all terrible?
- Let Q be the query, R a universe of documents selected for reranking, $S \subset R$ a subset already selected, D a document, sim_1 and sim_2 two suitable similarity functions, $\lambda \in [0, 1]$ a magic parameter

$$\arg \max_{D_i \in R \setminus S} \lambda \text{sim}_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} \text{sim}_2(D_i, D_j)$$

- I.e., avoid large sim_2 to any document already chosen
- Ad hoc, but works, especially for non-redundant multi-document summarization

Subtopic/aspect retrieval

- Already chosen docs $1, \dots, i-1$
- With **reference** language models $\theta_1, \dots, \theta_{i-1}$
- Want to choose next doc i with model θ_i
- Simplify: old model θ_O , new model θ_N
- If novelty were the only issue, we might wish to maximize $\text{KL}(\theta_N \parallel \theta_O)$
- Another option is a mixture model with two components

Reference component: $\Pr(w_i | \theta_O)$

Background component: $\Pr(w_i | \theta_B)$ where θ_B may be from a large background corpus

$$\ell(\lambda | d) = \sum_i \log((1 - \lambda) \Pr(w_i | \theta_O) + \lambda \Pr(w_i | \theta_B))$$

- How nice is the optimization?
- Large λ means more novel