Text Analysis and Topic Models

Wray Buntine
National ICT Australia (NICTA)
Australian National University (ANU)

MLSS, Oct., 2010

Part II

Topic Models

Outline

Outline

We look at previous related *latent variable* models, and consider the background to the current techniques.

1. Introducing Topic Models
   - History
   - Dirichlets
   - Text and Topics
2. Algorithms
3. Issues

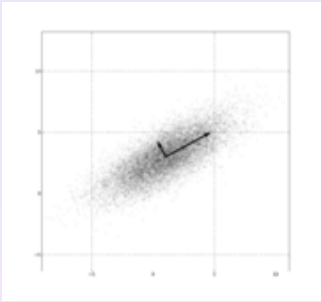
Motivation

- Web industry players are exploring the use of topic models for text. *e.g.*, Microsoft, Yahoo, various startups.
- Large amounts of text in different context available (blogs, news, corporate, Wikipedia, language, ...).
- How do we adapt and scale topic models for the larger web-scale, or at least enterprise-scale?
  - Current processing performance is of the order of one million documents on a desktop in a day.
- How do we handle text, to get sorts of topics we want?
  - May wish to preprocess to extract names and key words.
- How do we embed topic models so they function usefully inside a search engine or document analysis platform?
  - Can we automatically "label" topics?

Outline

Principal Components Analysis

Principal Components Analysis (PCA), dimensionality reduction tool, invented by Karl Pearson in 1901.



- Convert to zero mean, then find eigenvectors and eigenvalues of the data in $\Re^D$ and throw away the smaller dimensions. Has many other names and variations.
- Theory involves modelling data with a multivariate Gaussian.
- Variant is Latent Semantic Indexing (LSI), intended for text in IR, but of mixed benefit, and difficult to interpret.

Gaussian and least squares models fail for the smaller counts data we are considering. Need Poisson or multinomial modelling instead.

Independent Components Analysis

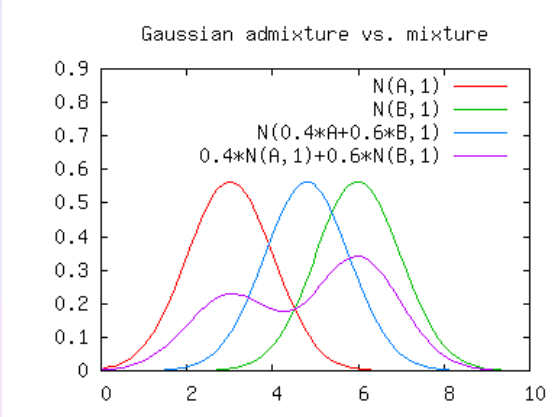
Independent Components Analysis (ICA), invented by Herault and Jutten in 1986, on image and signal data.

- Used widely for *blind source separation*. *e.g.*, given stereo recording of two speakers mixed together in a busy room, split signal into "speaker 1", "speaker 2," and "background."
- Theory assumes data is "low noise," non-Gaussian, and not highly discrete.
  - *e.g.* integers in range 0-256 are approximately continuous, but not so for range 0-3.
- Hence, often PCA is used initially to clean data (to make it low noise).

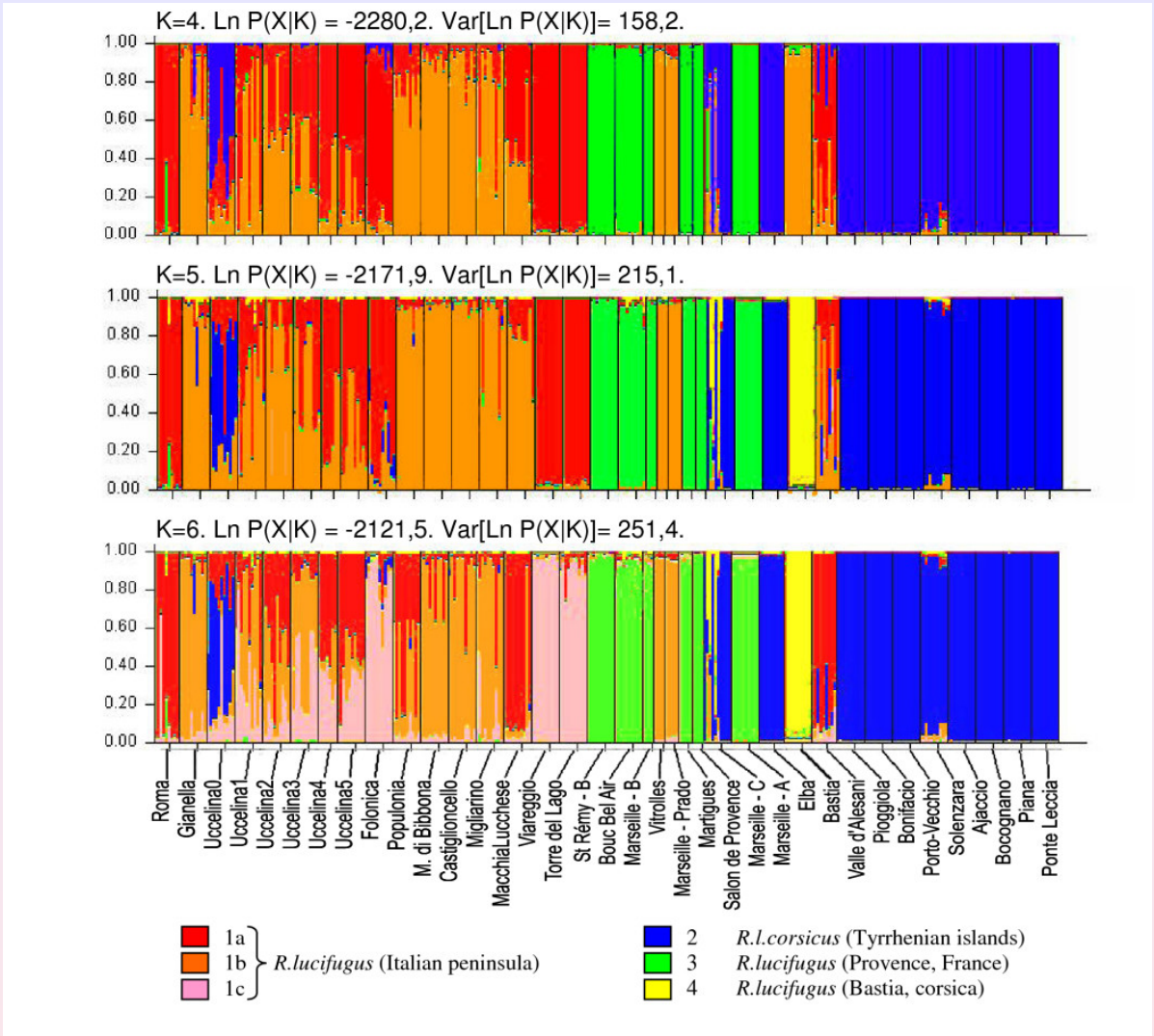
Not necessarily suitable for sparse and small-valued counts data we are considering.

Admixture Models

General technique in statistics where the population parameters are mixed. *i.e.*, each individual is not one or the other, but is rather a mixture.



- Boring for a Gaussian, interesting for a multinomial.
- Widely used in Genetics, see use of Pritchard, Stephens and Donnelly's `Structure` program in journals like *Science*, *PNAS*, *Nature*, *Genetics*, *etc.*
- Statistically, more difficult to model since EM algorithm not always directly applicable. Techniques used from '80's onwards, often based on Gibbs sampling.



"From speciation to introgressive hybridization: the phylogeographic structure of an island subspecies of termite, *Reticulitermes lucifugus corsicus*" Lefebvre, Châline, Limousin, Dupont, Bagnères, *BMC Evolutionary Biology* 2008, **8**:38, doi:10.1186/1471-2148-8-38

Discrete Topic Models

- Soft clustering, "grade of membership", Woodbury & Manton, 1982.
- Hidden facets in image interpretation, Non-negative Matrix Factorization (NMF), Seung and Lee, 1999.
- Probabilistic Latent Semantic Analysis (PLSI), topics in text, Hofmann, 1999.
- Admixture modelling, fully Bayesian, population structure from genotype data, Pritchard, Stephens and Donnelly, 2000.
- Latent Dirichlet Allocation (LDA) Blei, Ng and Jordan, 2001. Variant of Pritchard *et al.* Introduced mean-field algorithm.
- Collapsed Gibbs sampler, Griffiths and Steyvers, 2004.
- Gamma-Poisson model (GaP), Canny 2004 (extension of NMF).

... variants, extensions, adaptations, ..., 2001-2010

Outline

Binomials and Betas

Binomial distribution: if a Boolean variable has a probability p of turning up `true`, and N independent trials are done, then the probability of n `true`'s is, $n \sim \text{binomial}(p, N)$,

$$p(n \mid p, N, \text{binomial}) \;=\; C_n^N p^n (1-p)^{N-n} \;.$$

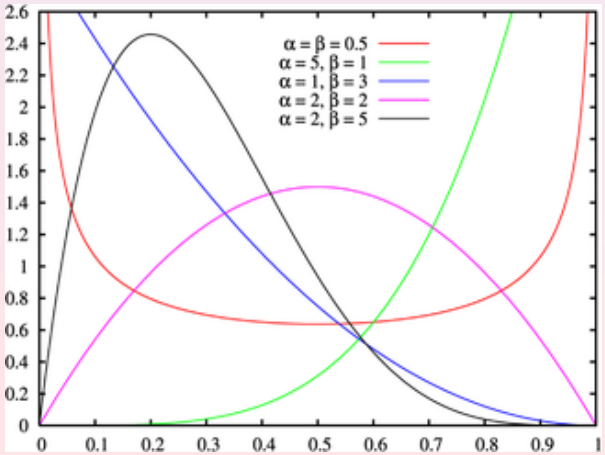
Beta distribution: a mathematically convenient *prior distribution* for the binomial parameter p takes the form of a Beta distribution, $p \;\sim\; \text{Beta}(\alpha, \beta)$.

$$p(p \mid \alpha, \beta, \text{Beta}) \;=\; \frac{1}{Beta(\alpha, \beta)} p^{\alpha-1}(1-p)^{\beta-1} \;.$$

The normalising constant

$$Beta(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta) \;.$$

The *posterior distribution* with n `true`'s out of N independent trials is now $p \mid N, n \;\sim\; \text{Beta}(\alpha + n, \beta + N - n)$.



Multinomials and Dirichlets (generalising from 2-D)

Multinomial distribution: if a discrete variable has a probability p_d of turning up d for $d = 1, ..., D$, and N independent trials are done, and sorting and counting the outcomes yields n_d counts of d's. This yields the vector distribution $\vec{n} \sim \text{multinomial}_D(\vec{p}, N)$,

$$p(\vec{n} \,|\, N, D, \vec{p}, \text{multinomial}) \;=\; C^N_{\vec{n}} \prod_{d=1}^{D} p_d^{n_d} \;.$$

NB. when $N = 1$, have $d \sim \text{discrete}_D(\vec{p})$.

Dirichlet distribution: a mathematically convenient *prior distribution* for the multinomial parameters $\vec{p}$ takes the form of a Dirichlet distribution, $\vec{p} \;\sim\; \text{Dirichlet}_D\,(\vec{\alpha})$.

$$p(\vec{p} \,|\, D, \vec{\alpha}, \text{Dirichlet}) \;=\; \frac{1}{Beta(\vec{\alpha})} \prod_{d=1}^{D} p_d^{\alpha_d - 1} \;.$$

The normalising constant $Beta(\vec{\alpha}) \;=\; \prod_{d=1}^{D} \Gamma(\alpha_d) / \Gamma\left(\sum_{d=1}^{D} \alpha_d\right)$.

Example Multinomial: Mythology

NOUNS					
mythology	0.03337	God	0.02048	name	0.014747
goddess	0.012911	spirit	0.012639	legend	0.0087992
myth	0.0070882	demons	0.006807	Sun	0.0060099
Temple	0.0054717	deity	0.0054247	Bull	0.0051629
Dragon	0.0051379	Maya	0.0051243	King	0.00512
Sea	0.0049453	Norse	0.0044707	horse	0.0044592
symbol	0.0042196	animals	0.0040112	fire	0.0039879
hero	0.0038755	Romans	0.0038696	Apollo	0.0037588
VERBS					
called	0.034078	said	0.031081	see	0.029521
given	0.0269	associated	0.024591	according	0.021724
represented	0.020964	known	0.018896	could	0.017499
made	0.016952	depicted	0.01524	appeared	0.014662
ADJECTIVES					
Greek	0.091163	ancient	0.055393	great	0.02853
Egyptian	0.028071	Roman	0.025783	sacred	0.020446

Probability of a "mythology" word given by the following table. Note smaller probabilities not shown.

Outline

Bag of words to represent text

A page out of Dr. Zeuss's *The Cat in The Hat*:

> *So, as fast as I could, I went after my net. And I said, "With my net I can bet them I bet, I bet, with my net, I can get those Things yet!"*

In the *bag of words* representation as *word* (*count*):

> *after(1) and(1) as(2) bet(3) can(2) could(1) fast(1) get(1) I(7) my(3) net(3) said(1) so(1) them(1) things(1) those(1) went(1) with(2) yet(1) .*

Notes:

- For the Reuters RCV1 collection from 2000: $I \approx 800k$ documents, $J \approx 400k$ different words (excluding those occurring few times), $S \approx 300M$ words total.
- Represent as sparse matrix/vector form with integer entries.
- Compresses to about 2 bytes per token (*e.g.* $2S$ bytes) total storage.

Admixture vs. mixture modelling

A page out of Dr. Zeuss's *The Cat in The Hat*.
In the *mixture model* for multinomials, the page is assigned to one class, so each word is generated from the same fixed class multinomial.

> *So, as fast as I could, I went after my net. And I said, "With my net I can bet them I bet, I bet, with my net, I can get those Things yet!"* **[class=k]**

For each word position l, have word $j_l \sim \text{discrete}_J(\vec{\theta}_k)$.

In the *admixture model*, the page is assigned to an admixture of several multinomials. This can be expanded to mean that each word is generated from a mixture of classes.

> *So* **[class=2]**, *as* **[class=1]** *fast* **[class=2]** *as* **[class=1]** *I* **[class=3]** *could* **[class=1]**, *I* **[class=3]** *went* **[class=3]** *after* **[class=2]** *my* **[class=1]** *...*

For each word position l, have word $j_l \sim \text{discrete}_J(\mathbf{\Theta}\vec{m})$; alternatively, $k_l \sim \text{discrete}(\vec{m})$ and $j_l \sim \text{discrete}_J(\vec{\theta}_{k_l})$.

Viewing Topics at the Word Level

(Blei, Ng, and Jordan, 2003)

"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.

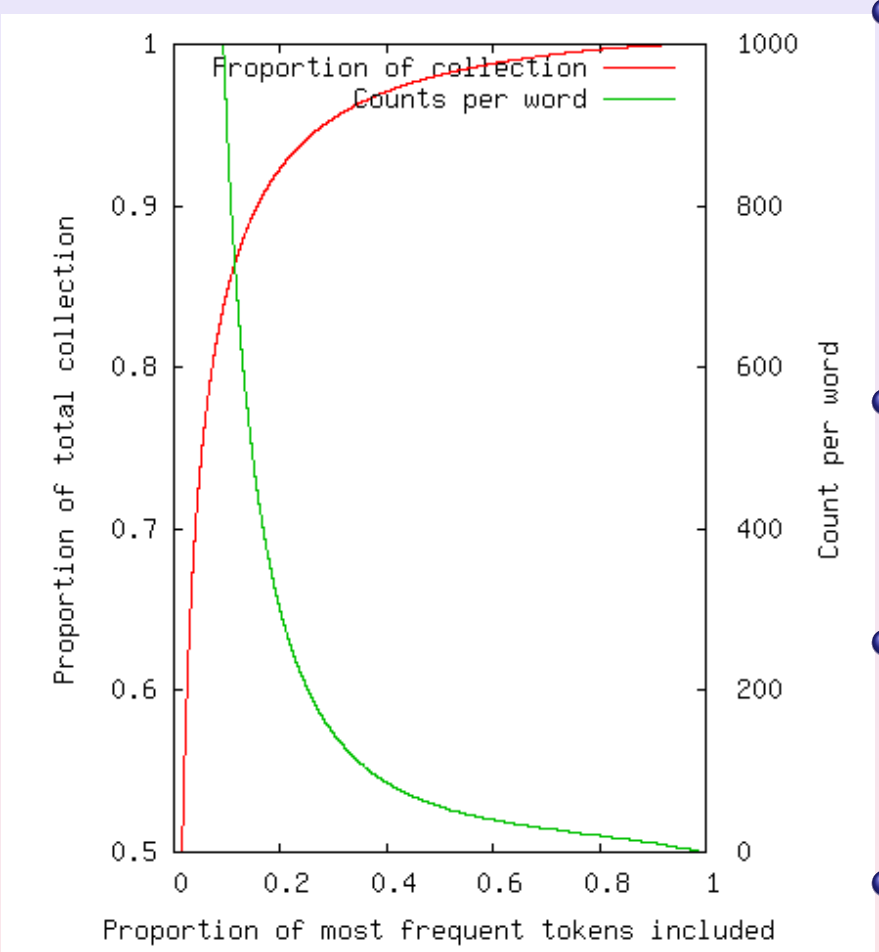
Figure 8: An example article from the AP corpus. Each color codes a different factor from which the word is putatively generated.

Discrete Topic Models, cont.

- Most are essentially a version of *independent* component analysis (ICA) where a scoring function is used that is more sensitive to discrete data.
- NMF with K-L metric, GaP, LDA and PLSI are all variations of one another if we ignore hyperparameters, and the statistical and optimisation methods used.

  NB. Bregman divergence variations also exist.
- Different kinds of algorithms used: variational EM, Gibbs sampling, collapsed Gibbs sampling, ...
- Lots of extensions exist:
  - using N-th (usually 2nd) order Markov models on words,
  - hierarchical extensions,
  - correlated topics, *e.g.*, Pachinko,
  - sparse matrices,
  - supervised,
  - time dependent or otherwise conditional topics.

Document-word tradeoffs



Data from NY Times collection from UCI.

- Deleting about 50% of the most infrequent words from the dictionary decreases the collection size by only about 3%.
- We can train on a subset of the dictionary as a way of boot-strapping.
- Shows that compression of various word matrices and vectors can be significant.
- Should also ignore words occurring in, say, 30% or more of documents as "stop" words.

Issues in text representation

- The basic semantic units in text are not just words but, sometimes, collocations or names.
  - *e.g.*, "New York Times", "George Bush"
  - most common are single words.
  - occasionally collocations words are *not* contiguous.
- Web pages full of "cruft", HTML junk, adverts, company fluff, navigation aids, boilerplate, ...
- Different "styles" of topics exists:

  genre: *e.g.*, product page, blog, news, corporate info.,

  library style categorisation: as done by Dewey Decimal, and DMOZ

  opinion and sentiment: *e.g.*, positive, anti-Microsoft, "green", ...

Outline

We cover basic theory and algorithms of topic models.

1. Introducing Topic Models
2. Algorithms
   - Theory
   - Variational EM
   - Collapsed Gibbs
3. Issues

Outline

Basic Model

- Everything tokenised, so have I documents, J words in the dictionary, K different topics/components.
- *Topic by word* matrix of proportions, $\mathbf{\Theta}$ of dimension $J \times K$. **NB.** $\sum_j \theta_{k,j} = 1$.
- Topic proportions per document i given by $\vec{m}_i$, a *latent or hidden* variable.
- Sample the topics proportions as

$$\vec{m}_i \sim \text{Dirichlet}_K(\vec{\alpha})$$

- Words in a document i generated independently, according to J-dimensional vector $\vec{m}_i^\dagger \mathbf{\Theta}$. For each position in sequence $l = 1, ..., L_i$

$$j_{i,l} \,|\, \mathbf{\Theta}, \vec{m}_i \sim \text{discrete}_J \left(\sum_k m_{i,k} \theta_{k,j_{i,l}} \right) .$$

Variations

- Dirichlet-discrete model (sometimes known as LDA):

$$\begin{aligned} \vec{m}_i &\sim \text{Dirichlet}_K(\vec{\alpha}) && \text{for each } i \\ j_{i,l} \,|\, \mathbf{\Theta}, \vec{m}_i &\sim \text{discrete}_J\left(\sum_k m_{i,k}\theta_{k,j_{i,l}}\right) && \text{for each } i, l \end{aligned}$$

- Gamma-Poisson model:

$$\begin{aligned} m_{i,k} &\sim \text{Gamma}(\alpha_k, \beta_k) && \text{for each } i, k \\ w_{i,j} \,|\, \mathbf{\Theta}, \vec{m}_i &\sim \text{Poisson}_J\left(\sum_k m_{i,k}\theta_{k,j}\right) && \text{for each } i, j \end{aligned}$$

  where $w_{i,j}$ is the count of words j in document i.

- Gamma-Poisson model is a probabilistic version of Non-negative Matrix Factorisation (NMF) when using KL minimisation.
- Dirichlet-discrete model becomes Gamma-Poisson model if we add a Poisson distribution on document lengths L_i.

Sampling Model

The basic model can be summarised:

1. *Prior:* For each topic k, sample the topic multinomial parameters $\vec{\theta}_k$ for $k = 1, ..., K$

$$\vec{\theta}_k \sim \text{Dirichlet}_J(\vec{\gamma})$$

2. *Likelihood:* For each document i,
   1. we'll sample the topics proportions,

$$\vec{m}_i \sim \text{Dirichlet}_K(\vec{\alpha})$$

   2. we'll sample the words $\vec{j}_i$, given document length L_i. For words in sequence $l = 1, ..., L_i$

$$p(j_{i,l} \,|\, \mathbf{\Theta}, \vec{m}_i) = \sum_k m_{i,k} \theta_{k,j_{i,l}} \; .$$

NB. usually, $\vec{\alpha}$ and possibly $\vec{\gamma}$ should be estimated as well.
NB. step 2.2 shows the admixture of $\mathbf{\Theta}$ using proportions $\vec{m}_i$.

Sampling Model, alternate

The basic model can be summarised:

1. *Prior:* ..... (as before)
2. *Likelihood:* For each document i,
   1. we'll sample the topics proportions,

$$\vec{m}_i \sim \text{Dirichlet}_K(\vec{\alpha})$$

   2. we'll sample the words $\vec{j_i}$, given document length L_i. For words in sequence $l = 1, ..., L_i$
      1. we'll sample the topic $k_{i,l}$

$$p(k_{i,l} \,|\, \vec{m}_i) = m_{i,k_{i,l}} \ ,$$

      2. then we'll sample the word $j_{i,l}$ given topic

$$p(j_{i,l} \,|\, \mathbf{\Theta}, k_{i,l}) = \theta_{k_{i,l},j_{i,l}} \ .$$

Outline

Variational EM Algorithm: Rough Outline

Seeks to maximise the likelihood, $p(\vec{j}_i, \vec{m}_i \text{ for } i \in Docs \,|\boldsymbol{\Theta}, \vec{\alpha})$, where $\vec{j}_i$ are the words for a document, $Z_K()$ is Dirichlet normaliser:

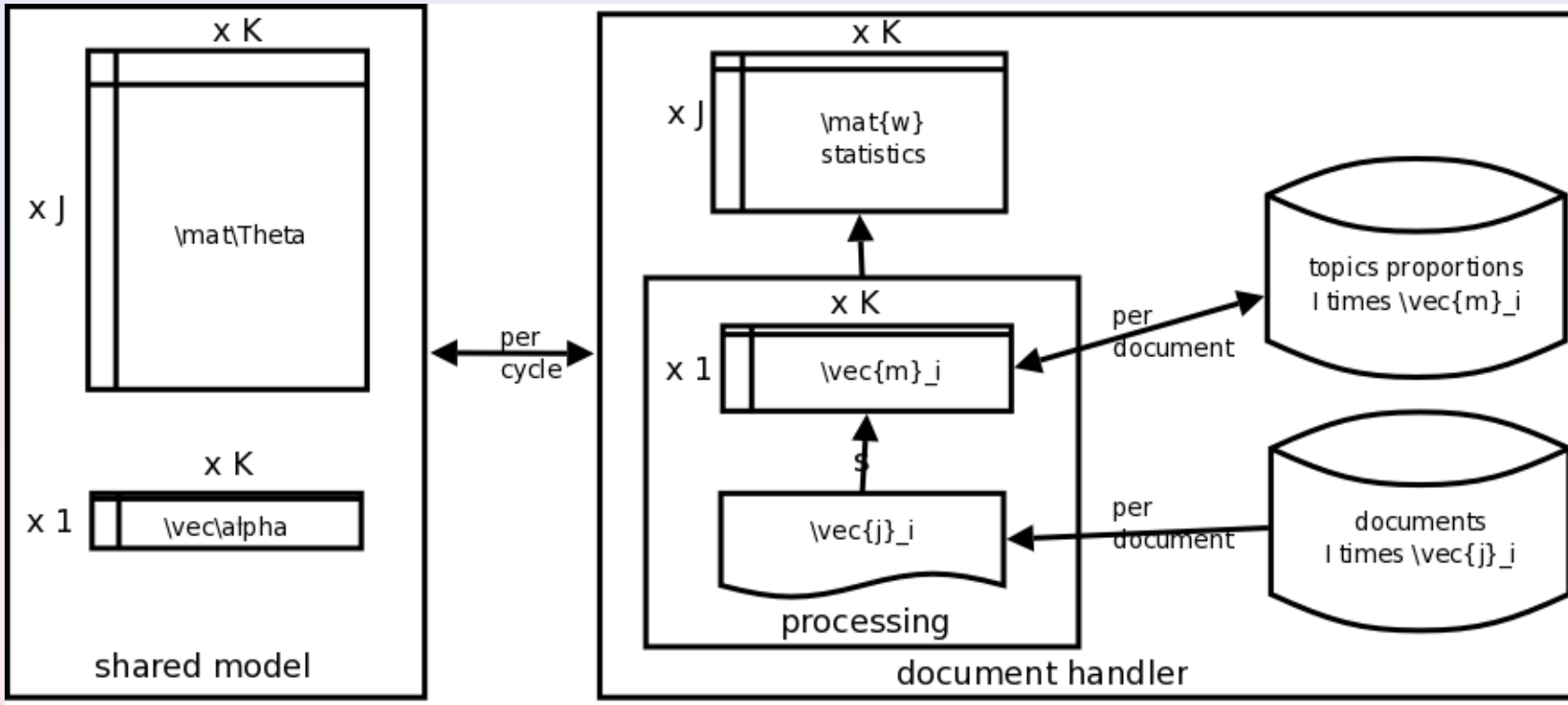
$$\prod_{i \in Docs} \left(\frac{1}{Z_K(\vec{\alpha})} \prod_{k \in Topics} m_{i,k}^{\alpha_k - 1} \right) \left(\prod_{l \in 1,\ldots,L_i} \sum_{k \in Topics} m_{i,k} \theta_{k,j_{i,l}} \right) .$$

Typically consists of a few hundred/thousand cycles in the form

1. For each document i, re-estimate/improve values for $\vec{m}_i$. While doing this, collect sufficient statistics on $\boldsymbol{\Theta}$.
2. Re-assign values for $\boldsymbol{\Theta}$ based on statistics collected in step (1), which uses the factored approximation

$$\prod_i \prod_l \prod_k \theta_{k,j_{i,l}}^{m_{i,k}} .$$

Parallel Variational EM



Parallel Variational EM, notes

- Distribute documents to different document handlers.
- The documents $\vec{j}_i$, and the document proportions $\vec{m}_i$ can be streamed, so are not a significant memory cost.
- $\vec{m}_i$ will need to be compressed when K is large.
- Need to communicate $\mathbf{\Theta}$ and $\vec{\alpha}$ with each major cycle: collect statistics, then distribute update.
- Communication should be done with efficient primitives: *logarithmic add* or *incremental update* to collect statistics, and *broadcast* to send out the updates. These are standard in MPI. *i.e.*, MPI sufficient, no need for map-reduce!
- $\mathbf{\Theta}$ compressible by factor of 2-20, when represented as total counts.

Outline

Collapsed Gibbs Algorithm: Derivation

Take the likelihood, $p(\vec{j}_i, \vec{m}_i \text{ for } i \in Docs \,|\boldsymbol{\Theta}, \vec{\alpha})$:

$$\prod_{i \in Docs} \left(\frac{1}{Z_K(\vec{\alpha})} \prod_{k \in Topics} m_{i,k}^{\alpha_k - 1} \right) \left(\prod_{l \in 1,\ldots,L_i} \sum_{k \in Topics} m_{i,k} \theta_{k,j_{i,l}} \right) .$$

Introduce the topics per word, $p(\vec{j}_i, \vec{k}_i, \vec{m}_i \text{ for } i \in Docs \,|\boldsymbol{\Theta}, \vec{\alpha})$

$$\prod_{i \in Docs} \left(\frac{1}{Z_K(\vec{\alpha})} \prod_{k \in Topics} m_{i,k}^{\alpha_k - 1} \right) \left(\prod_{l \in 1,\ldots,L_i} m_{i,k_{i,l}} \theta_{k_{i,l},j_{i,l}} \right) .$$

Collect terms in $\boldsymbol{\Theta}$ and $\vec{m}_i$, with statistics $\vec{W}$ and $\vec{C}_i$,

$$W_{k,j} = \text{count of word } j \text{ in topic } k = \sum_{i,l} 1_{k_{i,l}=k} 1_{j_{i,l}=j}$$

$$C_{i,k} = \text{count of topic } k \text{ in doc } i = \sum_{l} 1_{k_{i,l}=k}$$

Collapsed Gibbs Algorithm: Derivation

Integrate/marginalise $\vec{m}_i$, giving $p(\vec{j}_i, \vec{k}_i \text{ for } i \in Docs \,|\mathbf{\Theta}, \vec{\alpha})$

$$\prod_{k \in Topics} \prod_{j \in Words} \theta_{k,j}^{W_{k,j}} \prod_{i \in Docs} \frac{Z_K(\vec{\alpha} + \vec{C}_i)}{Z_K(\vec{\alpha})} .$$

Finally, integrate/marginalise $\mathbf{\Theta}$ (by adding prior for $\mathbf{\Theta}$ of $\vec{\gamma}$)
$p(\vec{j}_i, \vec{k}_i \text{ for } i \in Docs \,|\vec{\alpha}, \vec{\gamma})$

$$\prod_{k \in Topics} \frac{Z_J(\vec{\gamma} + \vec{W}_k)}{Z_J(\vec{\gamma})} \prod_{i \in Docs} \frac{Z_K(\vec{\alpha} + \vec{C}_i)}{Z_K(\vec{\alpha})} .$$

Two Likelihoods

$p(\vec{j}_i, \vec{m}_i \text{ for } i \in Docs \,|\mathbf{\Theta}, \vec{\alpha})$, based on topic proportions $\vec{m}_i$ for documents i,

$$\prod_{i \in Docs} \left(\frac{1}{Z_K(\vec{\alpha})} \prod_{k \in Topics} m_{i,k}^{\alpha_k - 1} \right) \left(\prod_{l \in WordSequence} \sum_{k \in Topics} m_{i,k} \theta_{k,j_{i,l}} \right)$$

Versus

$p(\vec{j}_i, \vec{k}_i \text{ for } i \in Docs \,|\mathbf{\Theta}, \vec{\alpha})$, based on topic assignments $\vec{k}_i$ for documents i (represented by statistics $\vec{W}$ (topic by word) and $\vec{C}_i$ (topics) for documents i):

$$\prod_{k \in Topics} \prod_{j \in Words} \theta_{k,j}^{W_{k,j}} \prod_{i \in Docs} \frac{Z_K(\vec{\alpha} + \vec{C}_i)}{Z_K(\vec{\alpha})} \ ,$$

Collapsed Gibbs Algorithm: Rough Outline

Probability $p(\vec{k}_i \text{ for } i \in Docs \,|\vec{j}_i \text{ for } i \in Docs, \vec{\alpha}, \vec{\gamma})$

$$\propto \prod_{k \in Topics} \left(\frac{\Gamma\left(\sum_j \gamma_j\right)}{\Gamma\left(\sum_j \gamma_j + W_{k,j}\right)} \prod_{j \in Words} \frac{\Gamma(\gamma_j + W_{k,j})}{\Gamma(\gamma_j)} \right)$$

$$\prod_{i \in Docs} \left(\frac{\Gamma\left(\sum_k \alpha_k\right)}{\Gamma\left(\sum_k \alpha_k + C_{i,k}\right)} \prod_{k \in Topics} \frac{\Gamma(\alpha_k + C_{i,k})}{\Gamma(\alpha_k)} \right)$$

Change the topic assignment for one word, $k_{i,l}$, gives simple product formula for a Gibbs update on $k_{i,l}$. See Griffiths and Steyvers 2004.

$$p(k_{i,l} = k \,|\, j_{i,l} = j, \vec{W}, \vec{C}, \vec{\alpha}, \vec{\gamma}) \;\propto\; (C_{i,k} + \alpha_k) \frac{W_{k,j} + \gamma_j}{\sum_j (W_{k,j} + \gamma_j)}$$

where $\vec{C}_i$ is the topic totals for document i, and $\vec{W}_k$ is the word totals for topic k.

Collapsed Gibbs Algorithm: Rough Outline, cont.

The formula for a Gibbs update on $k_{i,l}$:

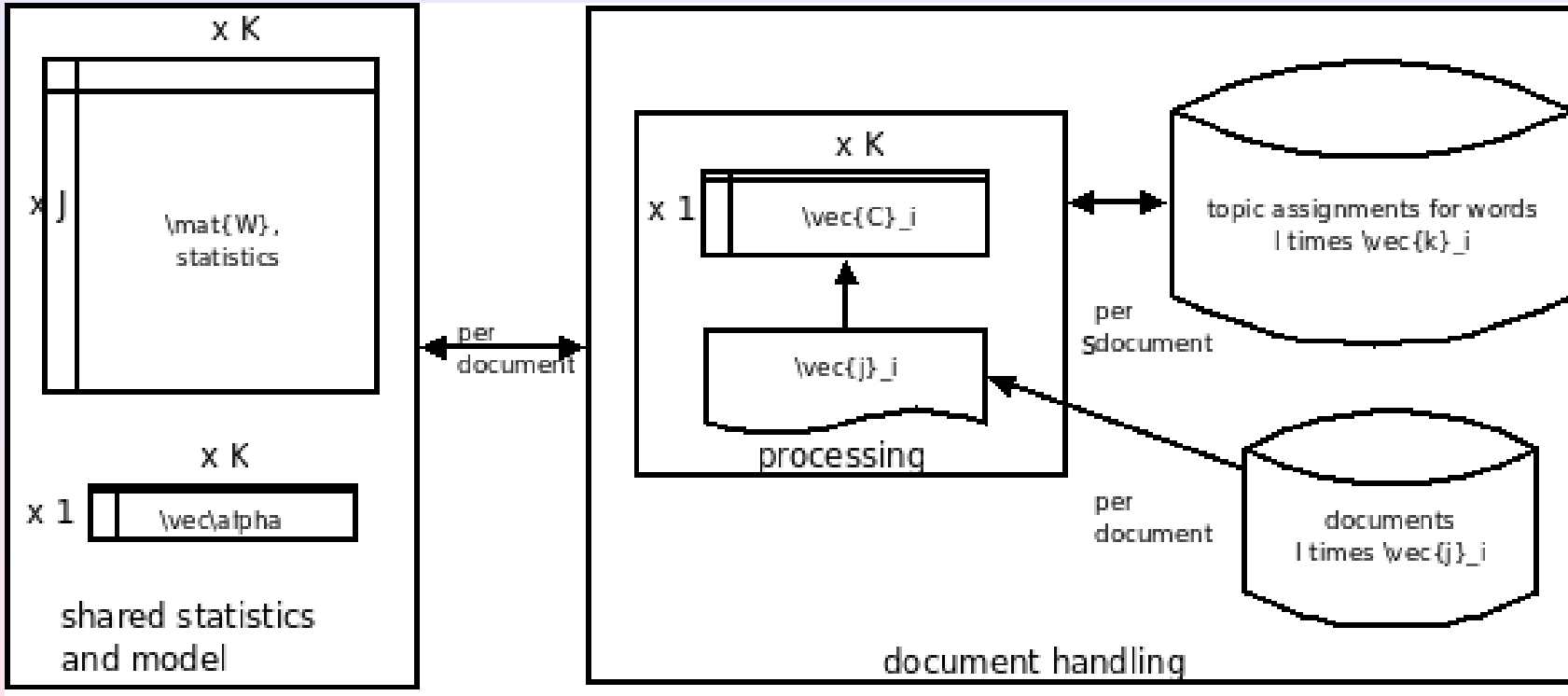
$$p(k_{i,l} = k \,|\, j_{i,l} = j, \vec{W}, \vec{C}, \vec{\alpha}, \vec{\gamma}) \;\propto\; (C_{i,k} + \alpha_k) \frac{W_{k,j} + \gamma_j}{\sum_j (W_{k,j} + \gamma_j)}$$

where $\vec{C}_i$ is the topic totals for document i, and $\vec{W}$ is the topic totals by word.

Algorithm consists of thousand cycles in the form:

1. For each document i,
   1. Recompute topic totals $\vec{C}_i$ from topic assignments $\vec{k}_i$.
   2. For sequence $l = 1, ..., L_i$ in document, with word $j_{i,l}$,
      1. re-sample the topic assignment $k_{i,l}$ using statistics $\vec{W}$ and $\vec{C}_i$.
      2. update $\vec{W}$ and $\vec{C}_i$ as you go.

Parallel Collapsed Gibbs EM



Outline

Getting Independent Test Scores

Probability of a new document of length L given by a vector of word indices $\vec{j}$, $p(\vec{j} \mid \mathbf{\Theta}, \vec{\alpha}, L)$

$$= \int_{\vec{m} \in \text{"unit } K \text{ simplex"}} \frac{1}{Z_K(\vec{\alpha})} \left(\prod_k m_k^{\alpha_k - 1} \right) \prod_l \left(\sum_{k_l} m_{k_l} \theta_{k_l, j_l} \right) \mathrm{d}\vec{m}$$

$$= \sum_{\vec{k}} \frac{Z_K(\vec{\alpha} + \vec{C})}{Z_K(\vec{\alpha})} \prod_{l \in WordSequence} \theta_{k_l, j_l} \ .$$

where $\vec{C}$ is vector of topic counts per $\vec{k}$.

- Same issue as clustering, no real metric, so use likelihood scores?
- No easy exact formula for likelihood since probabilities are in terms of latent variables $\vec{m}$ or $\vec{k}$ with intractible integration/summation.
- Closed form summation can be done for words in document $L = 15$ and components $K < 5$.

Left to Right Sampling

Decompose probability of a new document as

$$p(\vec{j} \,|\, \mathbf{\Theta}, \vec{\alpha}, L) \;=\; \prod_{l=1}^{L} p(j_l \,|\, j_1, \ldots, j_{l-1}, \mathbf{\Theta}, \vec{\alpha}) \;.$$

Using vector samples
$(k_1, \ldots, k_{l-1}) \sim p(k_1, \ldots, k_{l-1} \,|\, j_1, \ldots, j_{l-1}, \mathbf{\Theta}, \vec{\alpha})$ generated by Gibbs, each term $p(j_l \,|\, j_1, \ldots, j_{l-1}, \mathbf{\Theta}, \vec{\alpha})$ is estimated seperately as

$$\approx \;\frac{1}{|\text{Sample}|} \sum_{(k_1,\ldots,k_{l-1}) \in \text{Sample}} p(j_l \,|\, j_1, \ldots, j_{l-1}, k_1, \ldots, k_{l-1}, \mathbf{\Theta}, \vec{\alpha})$$

where

$$\begin{aligned} & p(j_l \,|\, j_1, \ldots, j_{l-1}, k_1, \ldots, k_{l-1}, \mathbf{\Theta}, \vec{\alpha}) \\ &= \sum_k \theta_{k,j_l} p(k_l = k \,|\, k_1, \ldots, k_{l-1}, \mathbf{\Theta}, \vec{\alpha}) \;. \end{aligned}$$

Left-to-Right Sampling, cont.

- Is a "corrected" version of the Left-to-right Filter of Wallach (PhD thesis, 2008).
- Provably converges to the true value.
- Moderately slow, but only a smaller number of samples (*e.g.*, 20) seem necessary when averaging the likelihoods over a large test set.
- See Buntine (ACML 2009) for a faster method.

Outline

Dealing with Text

- Usually we delete all stop words.
- Also delete rare words, whose total occurrence in the collection is less than 50 or 10, say.
- We might also delete all words whose total occurrence in the collection is, 1.5 or 2 times or more the number of documents.
  NB. most will be stop words.
- It is advisable, if possible, to do named entity recognition and collocation detection as well, and use the names and collocations as single tokens.
- With text, "junk" topics often appear.
- We *really need* a joint topic and part-of-speech model.

Outline

Compression of Counts

- Word counts per topic appear in the statistics for $\mathbf{\Theta}$ and $\vec{m}_i$, denoted $\vec{W}$ and $\vec{C}_i$.
- Word counts can be sparse for less frequent words. If $K > 1000$ might have only 2-10% being non-zero.
- Compression *without* standard libraries (`bzip`, `zlib`, etc.) can be done be representing non-zero values, and by doing difference encoding and variable-byte-length integers.
- Compression is necessary for handling large vocabularies and large topic sizes (*e.g.*, $K > 1000$).
- Also, want major loops to be over *non-zero elements* of sparse array (for $\vec{W}$ or $\vec{C}_i$), *not* over $k = 0, ..., K-1$.

Pre-computation

The Gibbs update on $k_{i,l} = 0, 1, ..., K-1$ is given by

$$p(k_{i,l} = k \,|\, j_{i,l} = j, \vec{W}, \vec{C}, \vec{\alpha}, \vec{\gamma}) \;\propto\; (C_{i,k} + \alpha_k) \frac{W_{k,j} + \gamma_j}{\sum_j (W_{k,j} + \gamma_j)}$$

- Pre-compute $\frac{C_{i,k_{i,l}} + \alpha_{k_{i,l}}}{\sum_j \left(W_{k_{i,l},j} + \gamma_j \right)}$ *once* at document start, then update in $O(1)$ steps as each word is resampled. **NB.** possible since term independent of j.
- This lets one do $O(1)$ sampling at initialisation when $W_{k,j}$ is roughly balanced in k.

Outline

Sparse Topics, problem

- For the topic "Ancient Mythology" we don't care about probabilities for the words "laser" or "Brisbane".
- For the topic "Football", we don't car about probabilities for the word "immuno-suppression".
- Yet the standard model insists on probabilities for every word in every topic, no mater how irrelevant.
- Instead we want word inclusion to be sparse on a per-topic basis.

Sparse Topics, theory

Consider the collapsed Gibbs formula:

$$\prod_{k \in Topics} \frac{Z_J(\vec{\gamma} + \vec{W}_k)}{Z_K(\vec{\gamma})} \prod_{i \in Docs} \frac{Z_K(\vec{\alpha} + \vec{C}_i)}{Z_K(\vec{\alpha})} .$$

Lets make each word j on or off for each topic k, with probability ρ_j. When the counts for a word in a topic, $W_{j,k} = 0$, then the word might be off! Use the same Dirichlets otherwise.

$$\prod_{k \in Topics} \prod_{j \in Words_K} \rho_j \prod_{j \in \overline{Words_K}} (1 - \rho_j)$$

$$\prod_{k \in Topics} \frac{Z_{J_k}(\vec{\gamma}_k + \vec{W}_k)}{Z_K(\vec{\gamma}_k)} \prod_{i \in Docs} \frac{Z_K(\vec{\alpha} + \vec{C}_i)}{Z_K(\vec{\alpha})} .$$

This leads to a slightly more complex Gibbs update. Initialisation and search more problematic.

Outline

Supervised Topics

- Original Dirichlet-discrete model:

$$\begin{aligned} \vec{m}_i &\sim \text{Dirichlet}_K(\vec{\alpha}) && \text{for each } i \\ j_{i,l} \,|\, \boldsymbol{\Theta}, \vec{m}_i &\sim \text{discrete}_J\left(\sum_k m_{i,k}\theta_{k,j_{i,l}}\right) && \text{for each } i, l \end{aligned}$$

- Supervised topic model:

$$\begin{aligned} \vec{m}_i &\sim \text{multiLogisticRegression}(\vec{x}_i, \vec{\alpha}) && \text{for each } i \\ j_{i,l} \,|\, \boldsymbol{\Theta}, \vec{m}_i &\sim \text{discrete}_J\left(\sum_k m_{i,k}\theta_{k,j_{i,l}}\right) && \text{for each } i, l \end{aligned}$$

  where $\vec{x}_i$ are any input variables for the i-th document.

- The $\vec{x}_i$ may be time, 2-D location, ...
- Collapsed Gibbs no longer works, but variational inference still OK, with some tricks.
- See Blei and McAuliffe 2007 for details.