Gibbs Sampling and Variational Methods

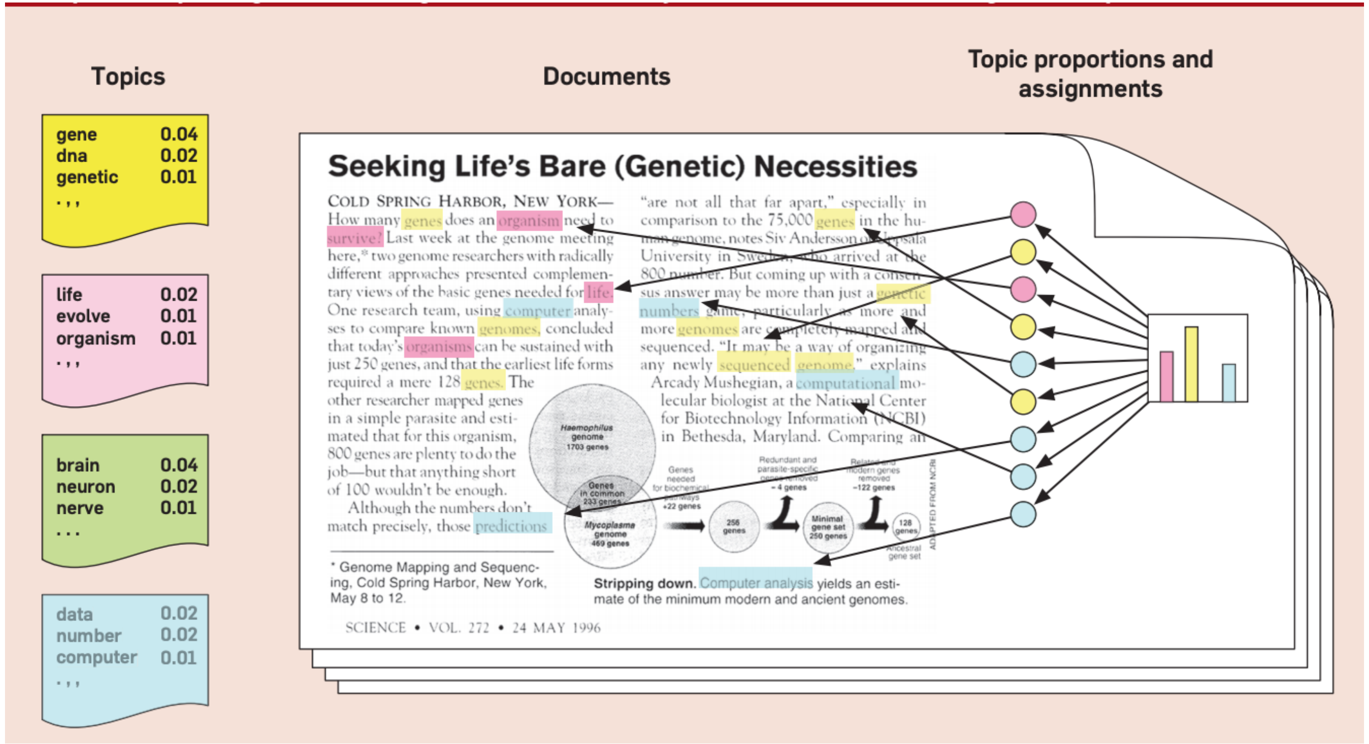
Héctor Corrada Bravo

University of Maryland, College Park, USA

CMSC 644: 2019-04-03

Mixture Models

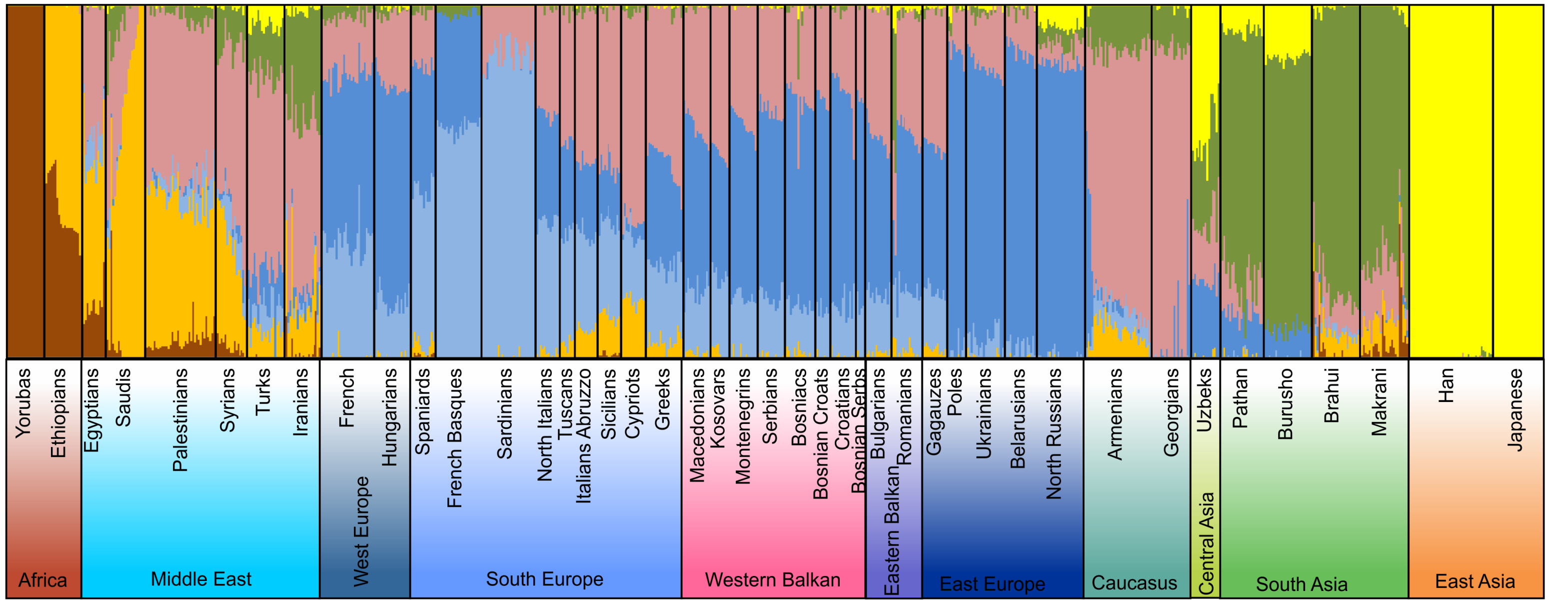
Documents as *mixtures* of topics (Hoffman 1999, Blei et al. 2003)



Blei (2012), Comm. ACM

Mixture Models

More applications: genetics, populations as mixture of ancestral populations



Kovacevic (2014), PLOS One

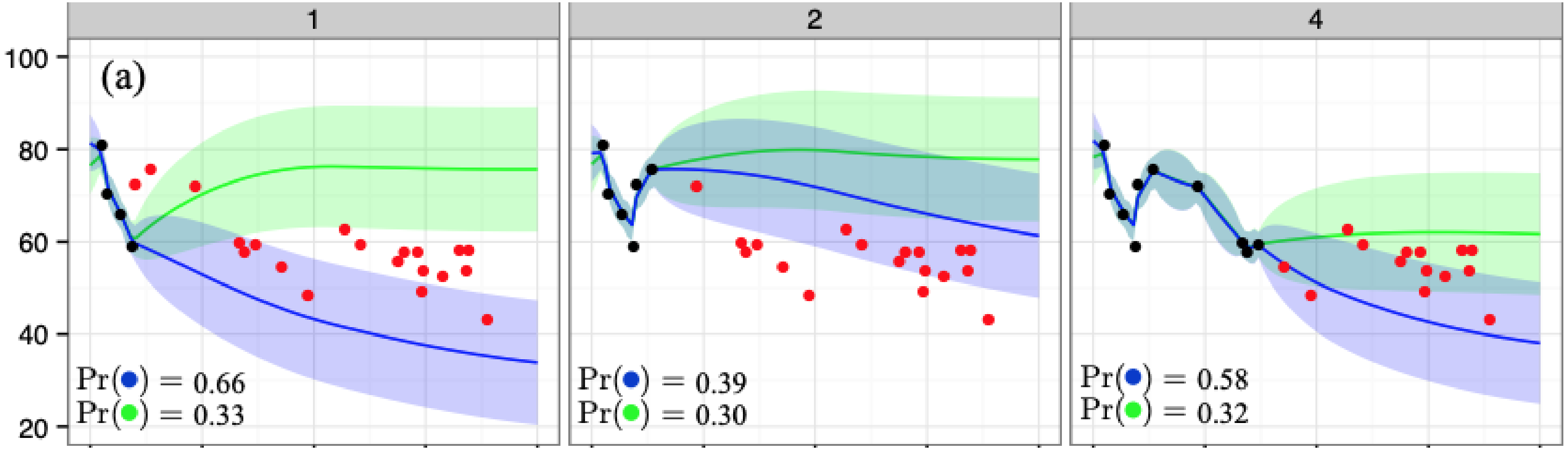
Mixture Models

More applications: clinical subtyping

Glueck, et al. (2017). TVCG

Mixture Models

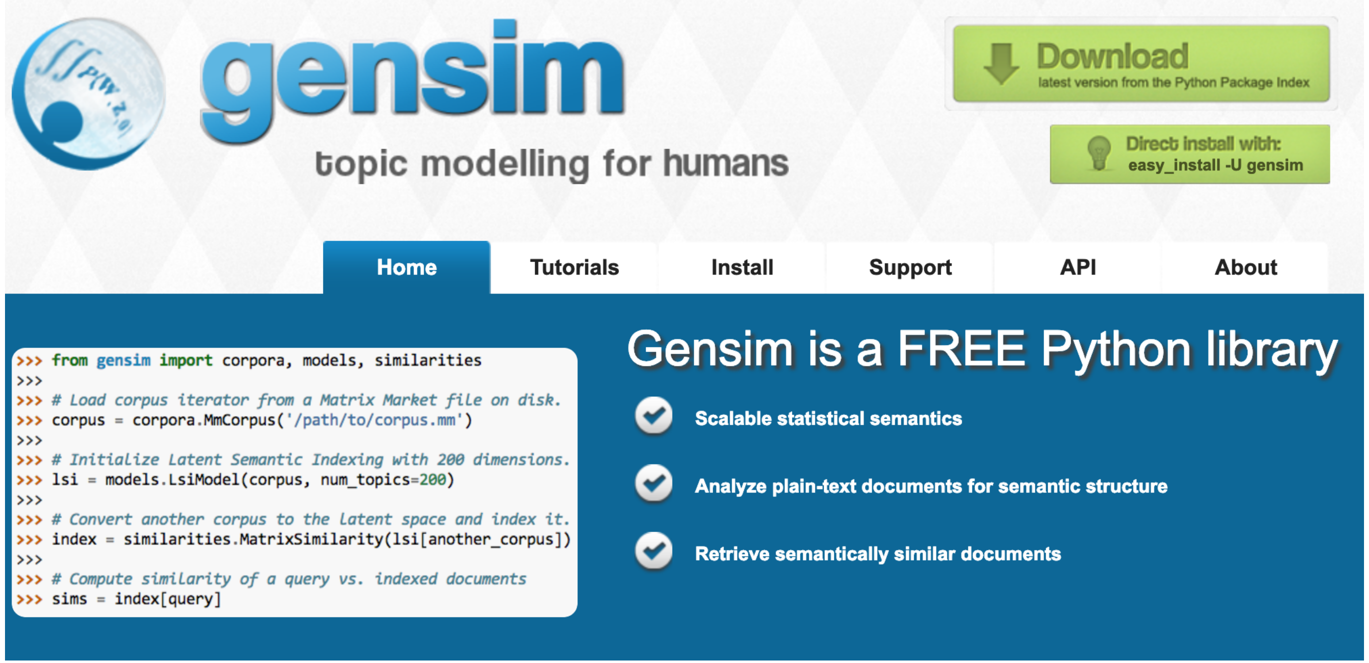
More applications: clinical prognosis



Schulman and Saria (2016). JMLR

Mixture Models

Software



https://radimrehurek.com/gensim/

Mixture Models

Software

Stan® is a state-of-the-art platform for statistical modeling and high-performance statistical computation. Thousands of users rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business.

Users specify log density functions in Stan's probabilistic programming language and get:

- full Bayesian statistical inference with MCMC sampling (NUTS, HMC)
- approximate Bayesian inference with variational inference (ADVI)
- penalized maximum likelihood estimation with optimization (L-BFGS)

https://mc-stan.org/

Mixture Models

We have a set of documents D

Each document modeled as a bag-of-words (bow) over dictionary W.

$x_{w,d}$: the number of times word $w \in W$ appears in document $d \in D$.

Approximate Inference by Sampling

Ultimately, what we are interested in is learning topics

Perhaps instead of finding parameters θ that maximize likelihood

Sample from a distribution $Pr(\theta|D)$ that gives us topic estimates

Approximate Inference by Sampling

Ultimately, what we are interested in is learning topics

Perhaps instead of finding parameters θ that maximize likelihood

Sample from a distribution $Pr(\theta|D)$ that gives us topic estimates

But, we only have talked about $Pr(D|\theta)$ how can we sample parameters?

Approximate Inference by Sampling

Like EM, the trick here is to expand model with *latent* data Z^m

And sample from distribution $Pr(\theta, Z^m|Z)$

Approximate Inference by Sampling

Like EM, the trick here is to expand model with *latent* data Z^m

And sample from distribution $Pr(\theta, Z^m|Z)$

This is challenging, but sampling from $Pr(\theta|Z^m, Z)$ and $Pr(Z^m|\theta, Z)$ is easier

Approximate Inference by Sampling

The *Gibbs Sampler* does exactly that

Property: After some rounds, samples from the conditional distributions

$Pr(\theta|Z^m, Z)$

Correspond to samples from marginal $Pr(\theta|Z) = \sum_{Z^m} Pr(\theta, Z^m|Z)$

Approximate Inference by Sampling

Quick aside, how to simulate data for pLSA?

- Generate parameters $\{p_d\}$ and $\{\theta_t\}$
- Generate $\Delta_{w,d,t}$

Approximate Inference by Sampling

Let's go backwards, let's deal with $\Delta_{w,d,t}$

Approximate Inference by Sampling

Let's go backwards, let's deal with $\Delta_{w,d,t}$

$$\Delta_{w,d,t} \sim \mathrm{Mult}_{\mathrm{x_{w,d}}}(\gamma_{w,d,1}, \ldots, \gamma_{w,d,T})$$

Where $\gamma_{w,d,t}$ was as given by E-step

Approximate Inference by Sampling

Let's go backwards, let's deal with $\Delta_{w,d,t}$

$$\Delta_{w,d,t} \sim \mathrm{Mult}_{\mathrm{x_{w,d}}}(\gamma_{w,d,1}, \ldots, \gamma_{w,d,T})$$

Where $\gamma_{w,d,t}$ was as given by E-step

```
for d in range(num_docs):

  delta[d,w,:] = np.random.multinomial(doc_mat[d,w],

      gamma[d,w,:])
```

Approximate Inference by Sampling

Hmm, that's a problem since we need $x_{w,d}$...

But, we know $Pr(w,d) = \sum_t p_{t,d}\theta_{w,t}$ so, let's use that to generate each $x_{w,d}$ as

$$x_{w,d} \sim \mathrm{Mult}_{n_d}(Pr(1,d), \ldots, Pr(W,d))$$

Approximate Inference by Sampling

Hmm, that's a problem since we need $x_{w,d}$...

But, we know $Pr(w,d) = \sum_t p_{t,d}\theta_{w,t}$ so, let's use that to generate each $x_{w,d}$ as

$$x_{w,d} \sim \mathrm{Mult}_{n_d}(Pr(1,d), \ldots, Pr(W,d))$$

```
for d in range(num_docs):
  doc_mat[d,:] = np.random.multinomial(nw[d], np.sum(p[:,d] * theta), axis=0)
```

Approximate Inference by Sampling

Now, how about p_d? How do we generate the parameters of a Multinomial distribution?

Approximate Inference by Sampling

Now, how about p_d? How do we generate the parameters of a Multinomial distribution?

This is where the Dirichlet distribution comes in...

If $p_d \sim \mathrm{Dir}(\alpha)$, then

$$Pr(p_d) \propto \prod_{t=1}^{T} p_{t,d}^{\alpha_t - 1}$$

Approximate Inference by Sampling

Some interesting properties:

$$E[p_{t,d}] = \frac{\alpha_t}{\sum_{t'} \alpha_{t'}}$$

So, if we set all $\alpha_t = 1$ we will tend to have uniform probability over topics ($1/t$ each on average)

If we increase $\alpha_t = 100$ it will also have uniform probability but will have very little variance (it will almost always be $1/t$)

Approximate Inference by Sampling

So, we can say $p_d \sim \mathrm{Dir}(\alpha)$ and $\theta_t \sim \mathrm{Dir}(\beta)$

Approximate Inference by Sampling

So, we can say $p_d \sim \mathrm{Dir}(\alpha)$ and $\theta_t \sim \mathrm{Dir}(\beta)$

And generate data as (with $\alpha_t = 1$)

```
for d in range(num_docs):
  p[:,d] = np.random.dirichlet(1. * np.ones(num_topics))
```

Approximate Inference by Sampling

So what we have is a *prior* over parameters $\{p_d\}$ and $\{\theta_t\}$: $Pr(p_d|\alpha)$ and $Pr(\theta_t|\beta)$

And we can formulate a distribution for missing data $\Delta_{w,d,t}$:

$$Pr(\Delta_{w,d,t}|p_d, \theta_t, \alpha, \beta) =$$
$$Pr(\Delta_{w,d,t}|p_d, \theta_t)Pr(p_d|\alpha)Pr(\theta_t|\beta)$$

Approximate Inference by Sampling

However, what we care about is the *posterior* distribution $Pr(p_d|\Delta_{w,d,t}, \theta_t, \alpha, \beta)$

What do we do???

Approximate Inference by Sampling

Another neat property of the Dirichlet distribution is that it is *conjugate* to the Multinomial

If $\theta|\alpha \sim \mathrm{Dir}(\alpha)$ and $X|\theta \sim \mathrm{Multinomial}(\theta)$, then

$$\theta|X, \alpha \sim \mathrm{Dir}(X + \alpha)$$

Approximate Inference by Sampling

That means we can sample p_d from

$$p_{t,d} \sim \mathrm{Dir}(\sum_w \Delta_{w,d,t} + \alpha)$$

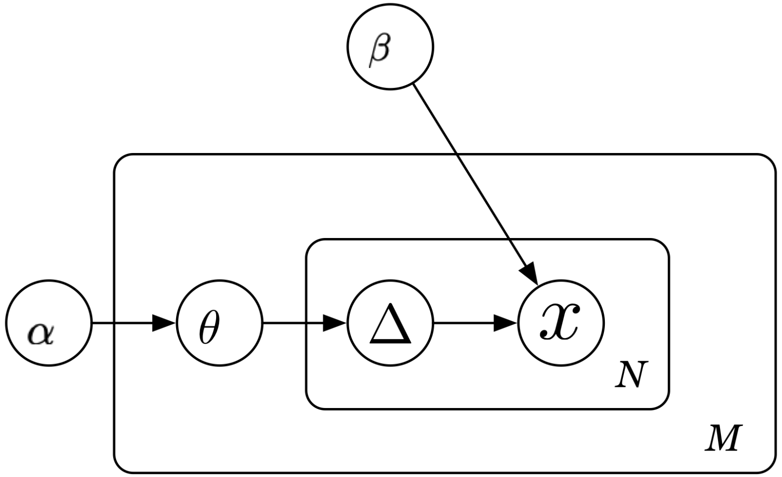
and

$$\theta_{w,t} \sim \mathrm{Dir}(\sum_d \Delta_{w,d,t} + \beta)$$

Approximate Inference by Sampling

Coincidentally, we have just specified the **Latent Dirichlet Allocation** method for topic modeling.

This is the most commonly used method for topic modeling



Blei, Ng, Jordan (2003), JMLR

Approximate Inference by Sampling

We can now specify a full Gibbs Sampler for an LDA mixture model.

Given:

- Word-document counts $x_{w,d}$
- Number of topics K
- Prior parameters α and β

Do: Learn parameters $\{p_d\}$ and $\{\theta_t\}$ for K topics

Approximate Inference by Sampling

Step 0: Initialize parameters $\{p_d\}$ and $\{\theta_t\}$

$$p_d \sim \mathrm{Dir}(\alpha)$$

and

$$\theta_t \sim \mathrm{Dir}(\beta)$$

Approximate Inference by Sampling

Step 1:

Sample $\Delta_{w,d,t}$ based on current parameters $\{p_d\}$ and $\{\theta_t\}$

$$\Delta_{w,d,.} \sim \mathrm{Mult}_{x_{w,d}}(\gamma_{w,d,1}, \ldots, \gamma_{w,d,T})$$

Approximate Inference by Sampling

Step 2:

Sample parameters from

$$p_{t,d} \sim \mathrm{Dir}(\sum_{w} \Delta_{w,d,t} + \alpha)$$

and

$$\theta_{w,t} \sim \mathrm{Dir}(\sum_{d} \Delta_{w,d,t} + \beta)$$

Approximate Inference by Sampling

Step 3:

Get samples for a few iterations (e.g., 200), we want to reach a stationary distribution...

Approximate Inference by Sampling

Step 4:

Estimate $\hat{\Delta}_{w,d,t}$ as the average of the estimates from the last m iterations (e.g., m=500)

Approximate Inference by Sampling

Step 5:

Estimate parameters p_d and θ_t based on estimated $\hat{\Delta}_{w,d,t}$

$$\hat{p}_{t,d} = \frac{\sum_w \hat{\Delta}_{w,d,t} + \alpha}{\sum_t \sum_w \hat{\Delta}_{w,d,t} + \alpha}$$

$$\hat{\theta}_{w,t} = \frac{\sum_d \hat{\Delta}_{w,d,t} + \beta}{\sum_w \sum_d \hat{\Delta}_{w,d,t} + \beta}$$

Mixture models

We have now seen two different mixture models: soft k-means and topic models

Mixture models

We have now seen two different mixture models: soft k-means and topic models

Two inference procedures:

- Exact Inference with Maximum Likelihood using the EM algorithm
- Approximate Inference using Gibbs Sampling

Mixture models

We have now seen two different mixture models: soft k-means and topic models

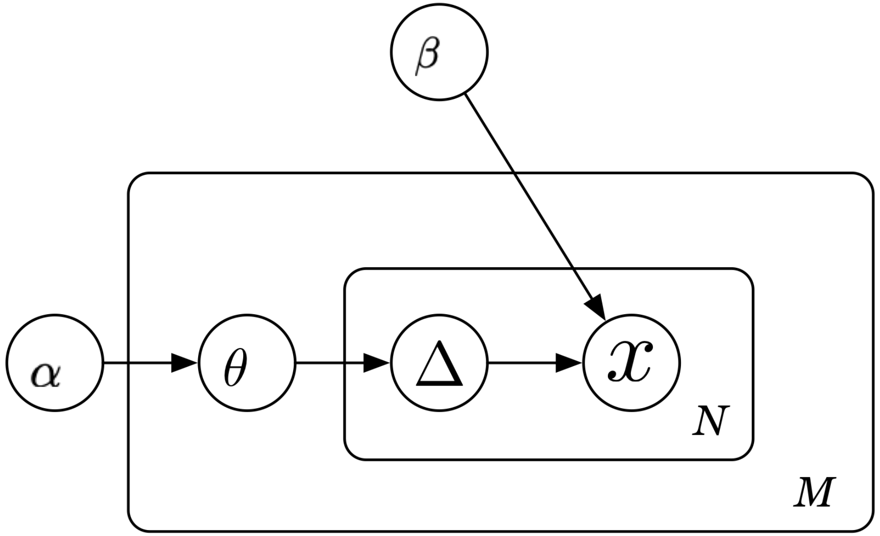
Two inference procedures:

- Exact Inference with Maximum Likelihood using the EM algorithm
- Approximate Inference using Gibbs Sampling

Next, we will go back to Maximum Likelihood but learn about Approximate Inference using Variational Methods

Variational Methods

Consider LDA model again

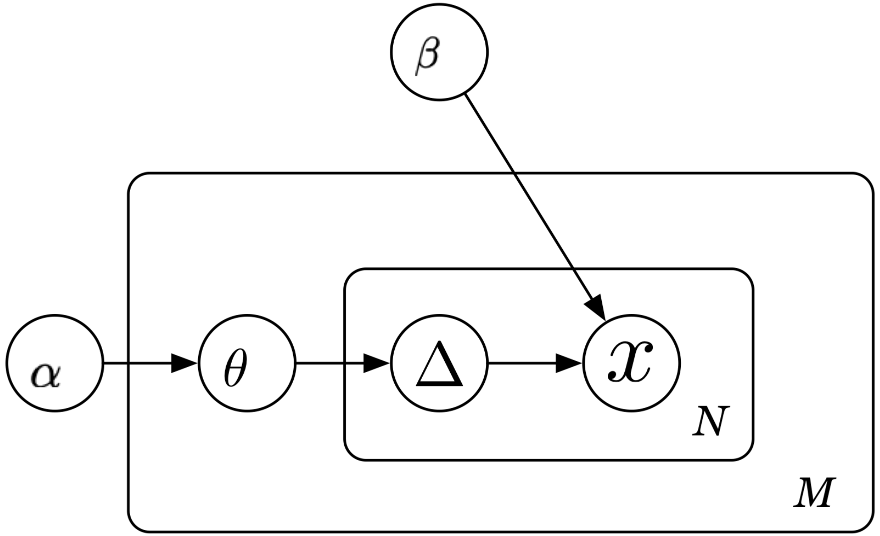


Benefits

- Full document generative model
- Can process new documents (posterior over topics) and words (prior parameters)

Variational Methods

Consider LDA model again



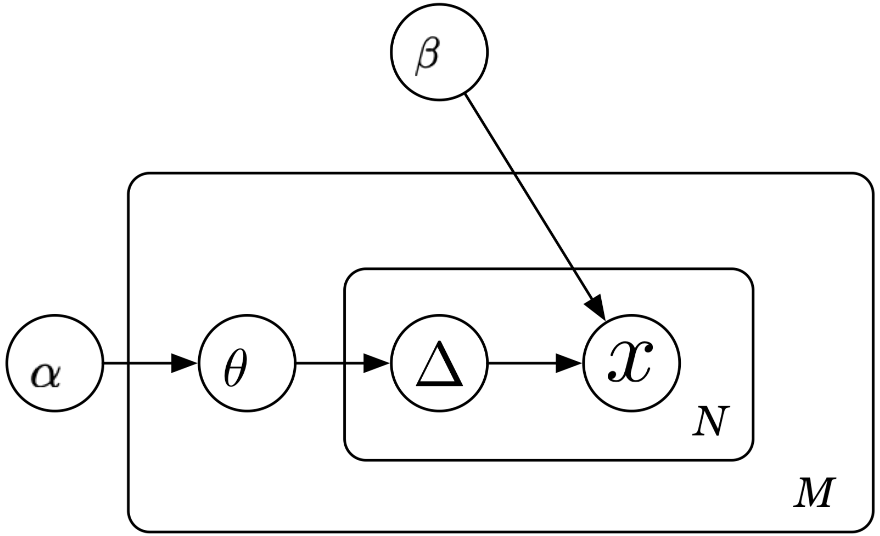
With Gibbs we sampled from

$Pr(\theta, \Delta | x, \alpha, \beta)$

What if we want to estimate parameters again? (maximum *a posteriori* parameters)

Variational Methods

Consider LDA model again

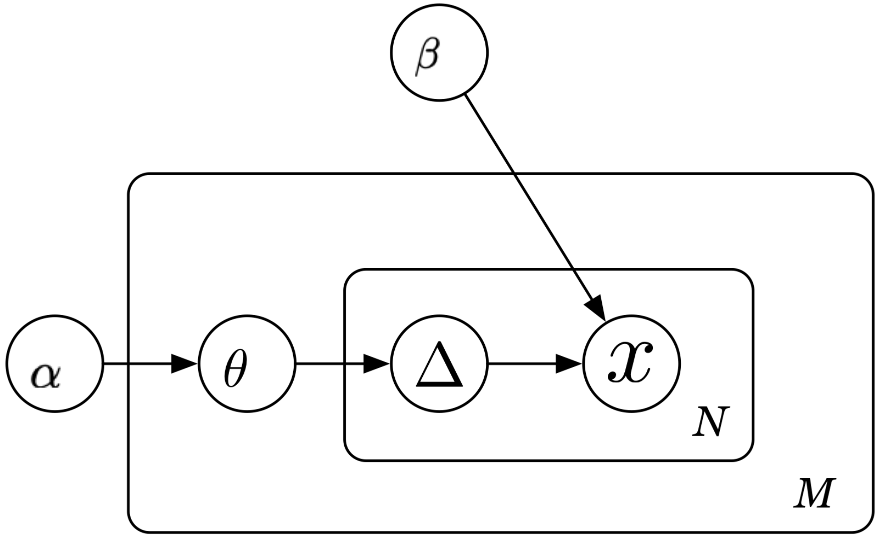


Very difficult to maximize

Harder than pLSA due to Dirichlet priors

Variational Methods

Consider LDA model again

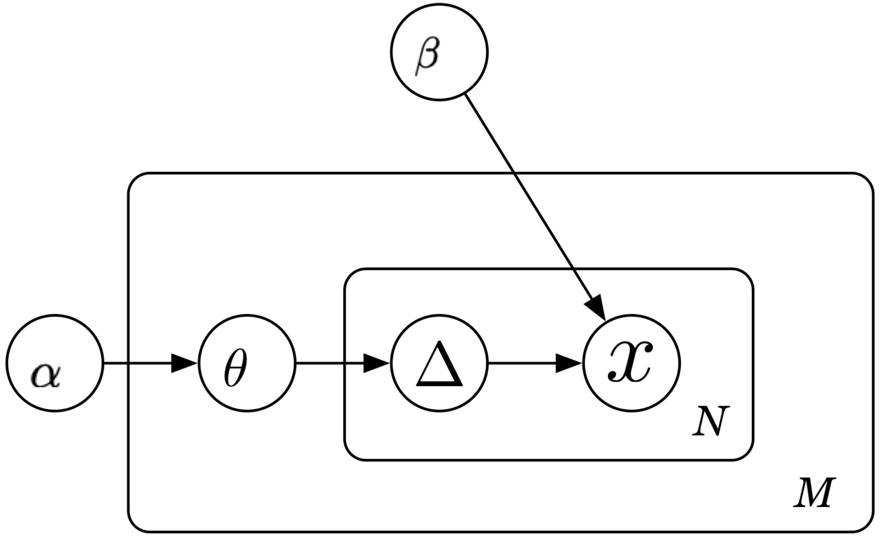


Let's get inspiration from EM: maximize lower bound

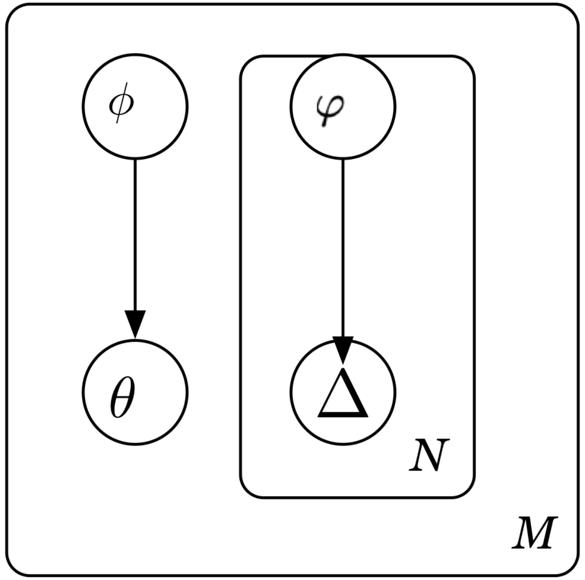
But what should the lower bound be?

Variational Methods

Consider LDA model again

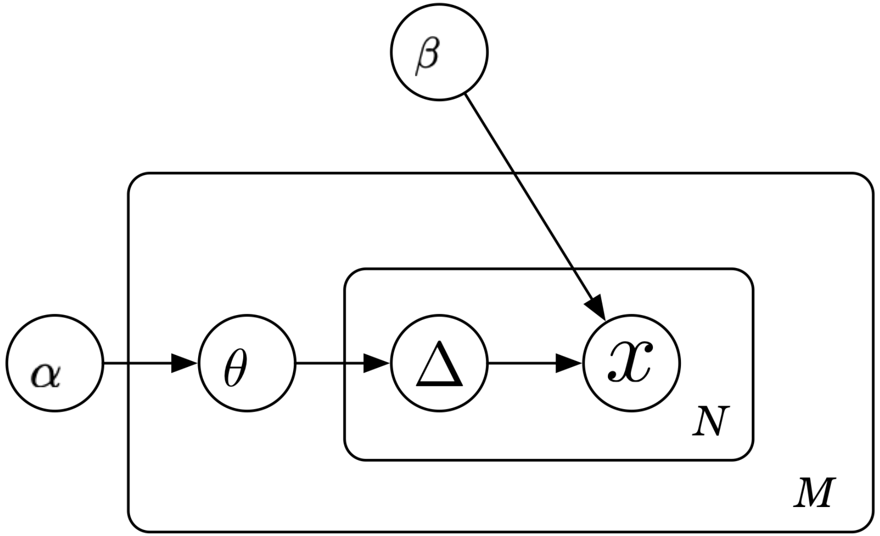


Make missing data and parameters "independent"!

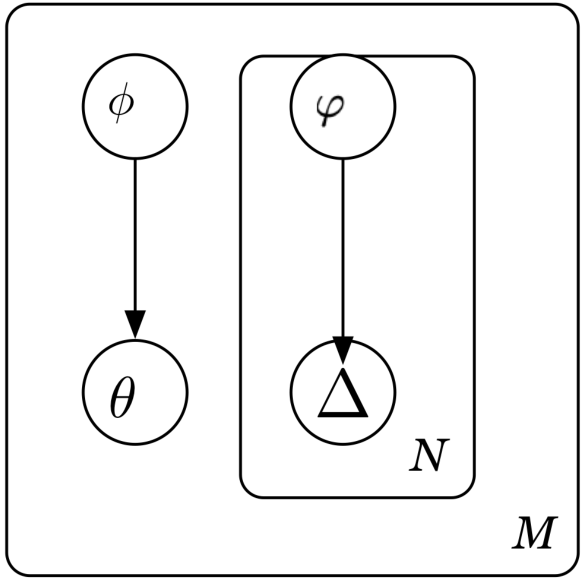


Variational Methods

Consider LDA model again



Find parameters that make simple model *most* similar to original model



Variational Methods

Can then define EM-like algorithm

E-step: define expectation w.r.t. approximate distribution

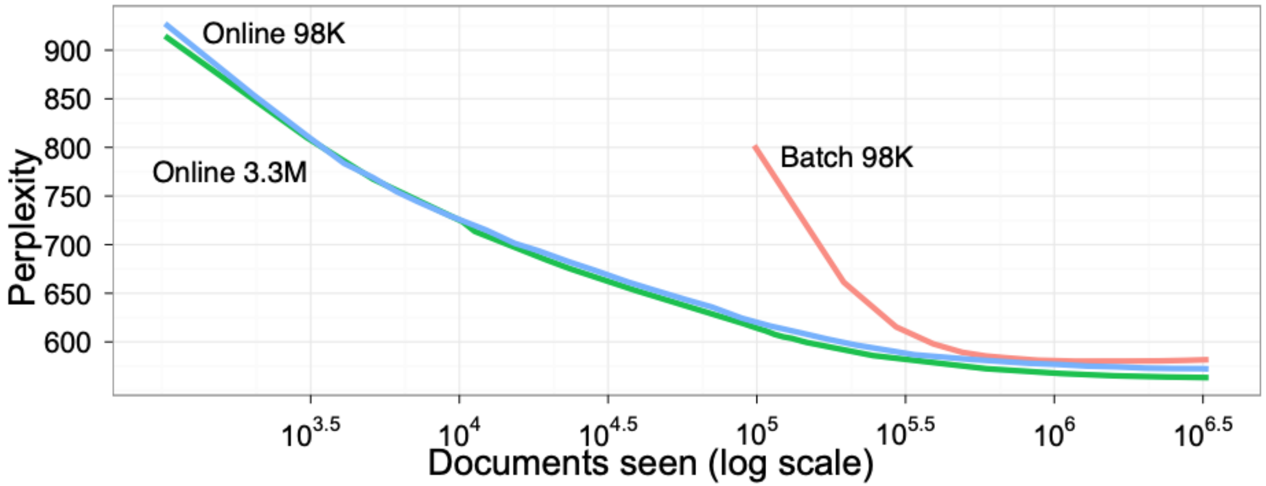
M-step: maximize parameters of approximate distribution

Variational Methods

Net result:

1) Maximum posterior estimates 2) Super simple updates 3) With stochastic approach (update using a few words at a time), extremely scalable

Variational Methods



Documents analyzed	2048	4096	8192	12288	16384	32768	49152	65536
Top eight words	systems	systems	service	service	service	business	business	business
	road	health	systems	systems	companies	service	service	industry
	made	communication	health	companies	systems	companies	companies	service
	service	service	companies	business	business	industry	industry	companies
	announced	billion	market	company	company	company	services	services
	national	language	communication	billion	industry	management	company	company
	west	care	company	health	market	systems	management	management
	language	road	billion	industry	billion	services	public	public

Hoffman, et al. (2010). NIPS

Conclusion

Probabilistic mixture models: powerful model class with many applications

Awesome historical algorithmic development

Outstanding software support