Mr. LDA: A Flexible Large Scale Topic Modeling Package using Variational Inference in MapReduce

Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja

COLLEGE OF
INFORMATION
STUDIES

Mr. LDA: A Flexible Large Scale Topic Modeling Package using Variational Inference in MapReduce

Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja

COLLEGE OF
INFORMATION
STUDIES

Introductions

MR LDA

- MR = MapReduce
- LDA = latent Dirichlet allocation
- MR LDA = Ke

Introductions

MR LDA

- MR = MapReduce
- LDA = latent Dirichlet allocation
- MR LDA = Ke

- First author
- Immigration issues prevented presentation

Roadmap

- Review of topic models
- The need for scalability
- Variational inference vs. Gibbs sampling
- Mr LDA
  - A scalable topic modeling package
  - Using variational inference
- Extensions
  - Extending psychologically-inspired word lists
  - Discovering topics consistent across languages

Outline

Why topic models?

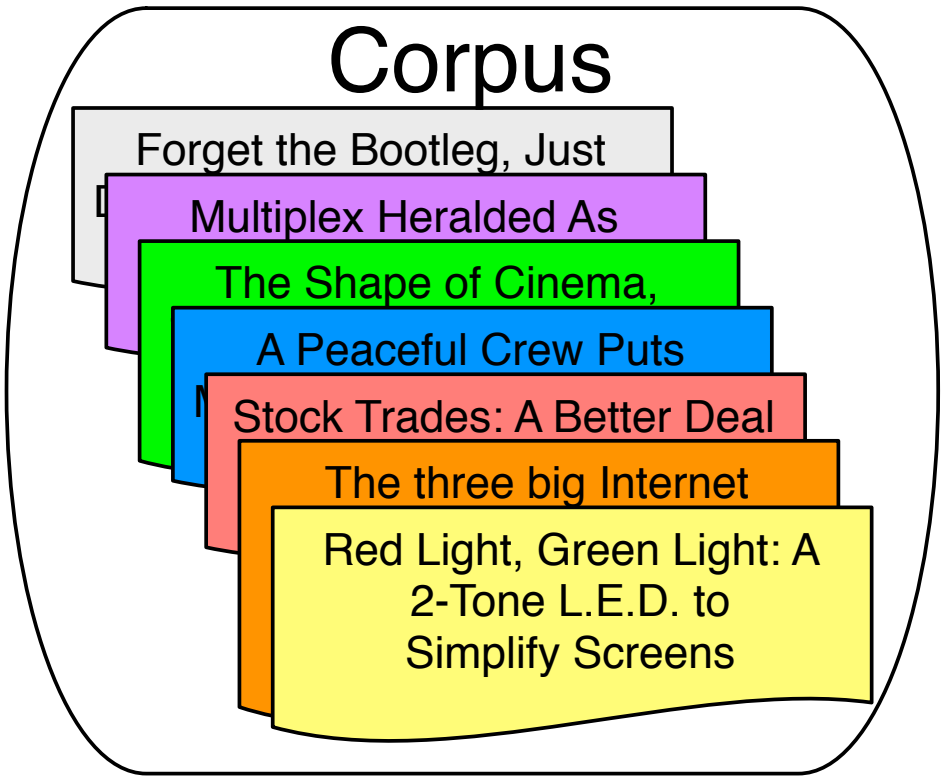
- Suppose you have a huge number of documents
- Want to know what's going on
- Can't read them all (e.g. every New York Times article from the 90's)
- Topic models offer a way to get a corpus-level view of major themes

Why topic models?

- Suppose you have a huge number of documents
- Want to know what's going on
- Can't read them all (e.g. every New York Times article from the 90's)
- Topic models offer a way to get a corpus-level view of major themes
- Unsupervised

Conceptual Approach

From an **input corpus** and number of topics $K \rightarrow$ words to topics



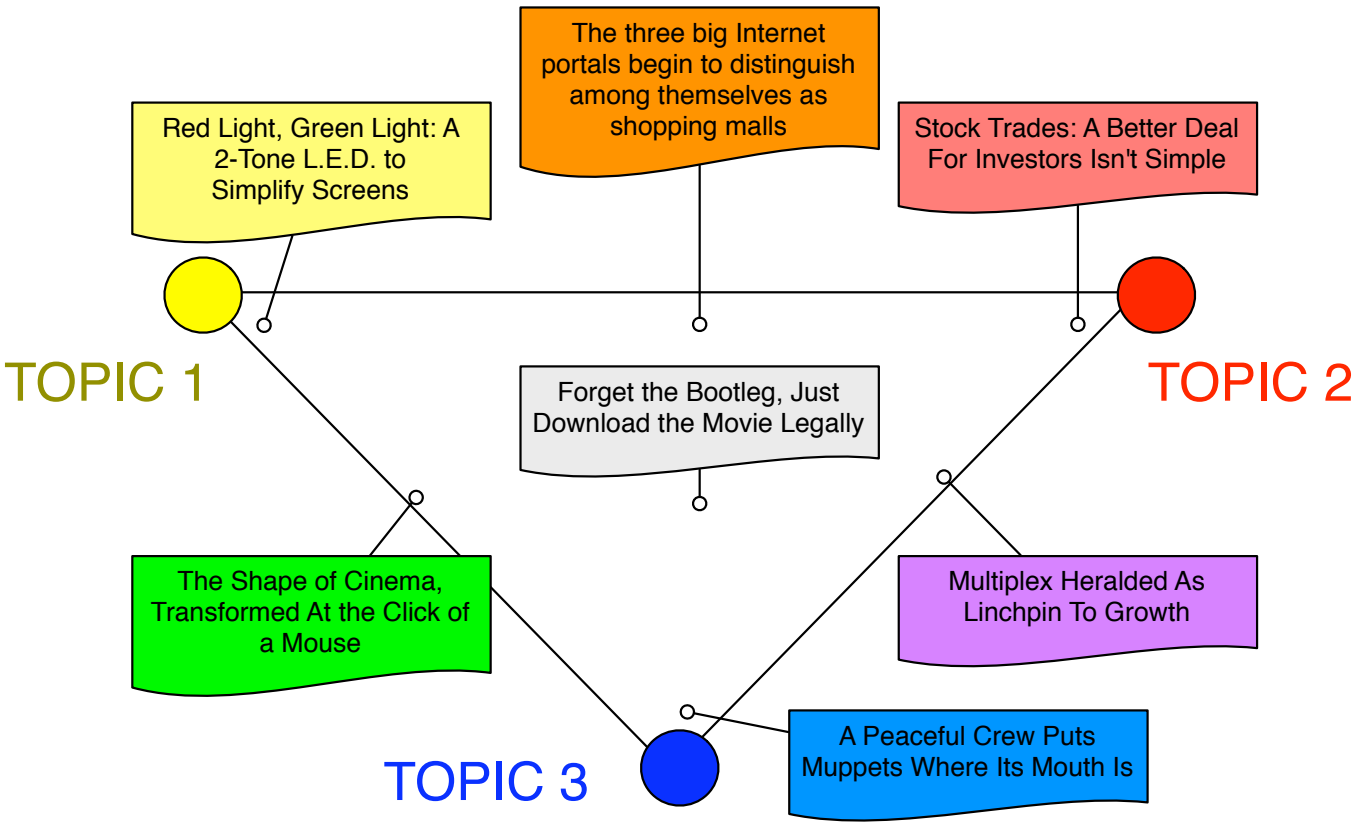
Conceptual Approach

From an input corpus and number of topics $K \rightarrow$ **words to topics**

TOPIC 1	TOPIC 2	TOPIC 3
computer, technology, system, service, site, phone, internet, machine	sell, sale, store, product, business, advertising, market, consumer	play, film, movie, theater, production, star, director, stage

Conceptual Approach

- For each document, what topics are expressed by that document?

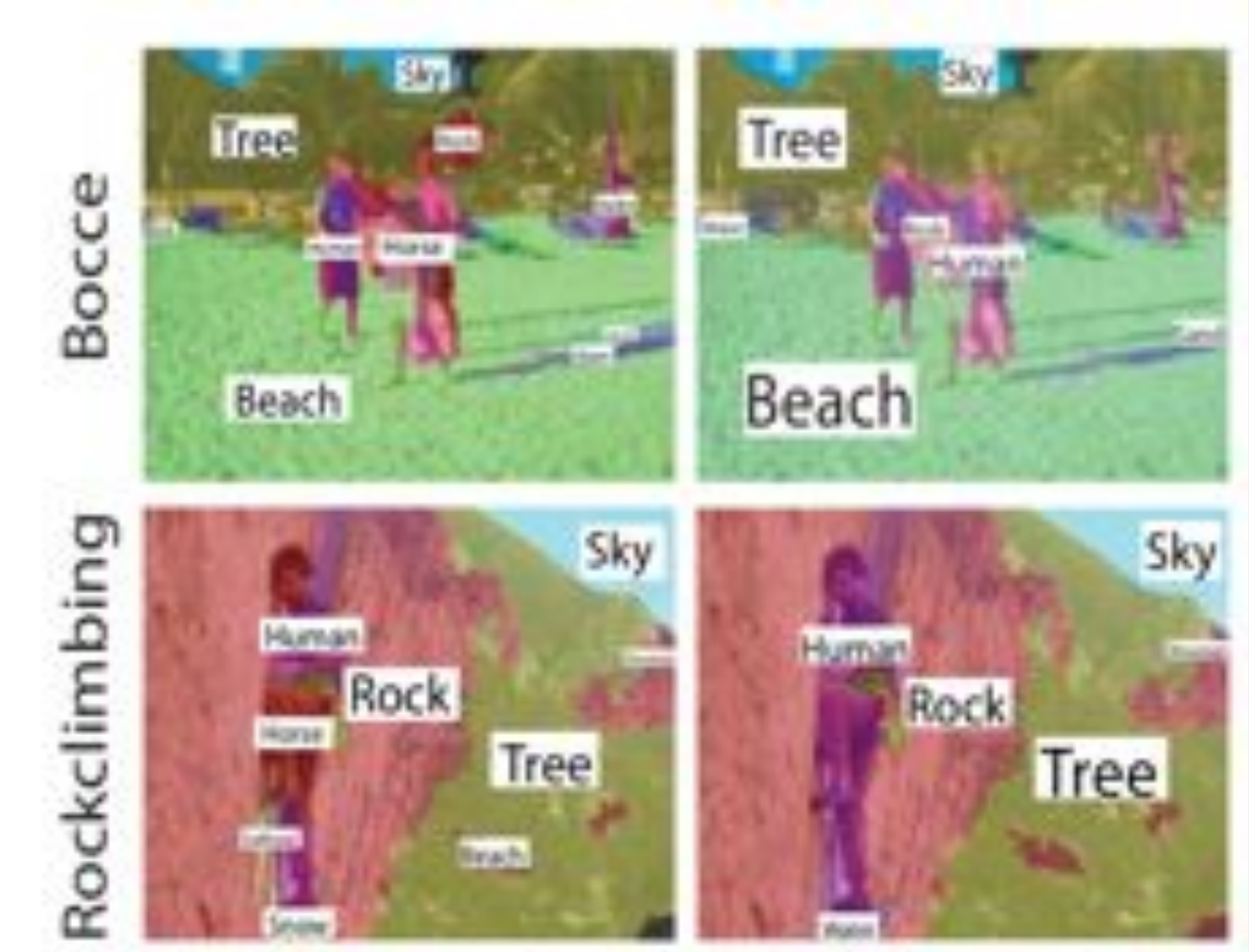


Topic Models: What's Important

- Topic models
  - Topics to words - multinomial distribution
  - Documents to topics - multinomial distribution
- Statistical structure inferred from data
- Have semantic coherence because of language use
- We use latent Dirichlet allocation (LDA) [*Blei et al.* 2003], a fully Bayesian version of pLSI [*Hofmann* 1999], probabilistic version of LSA [*Landauer and Dumais* 1997]

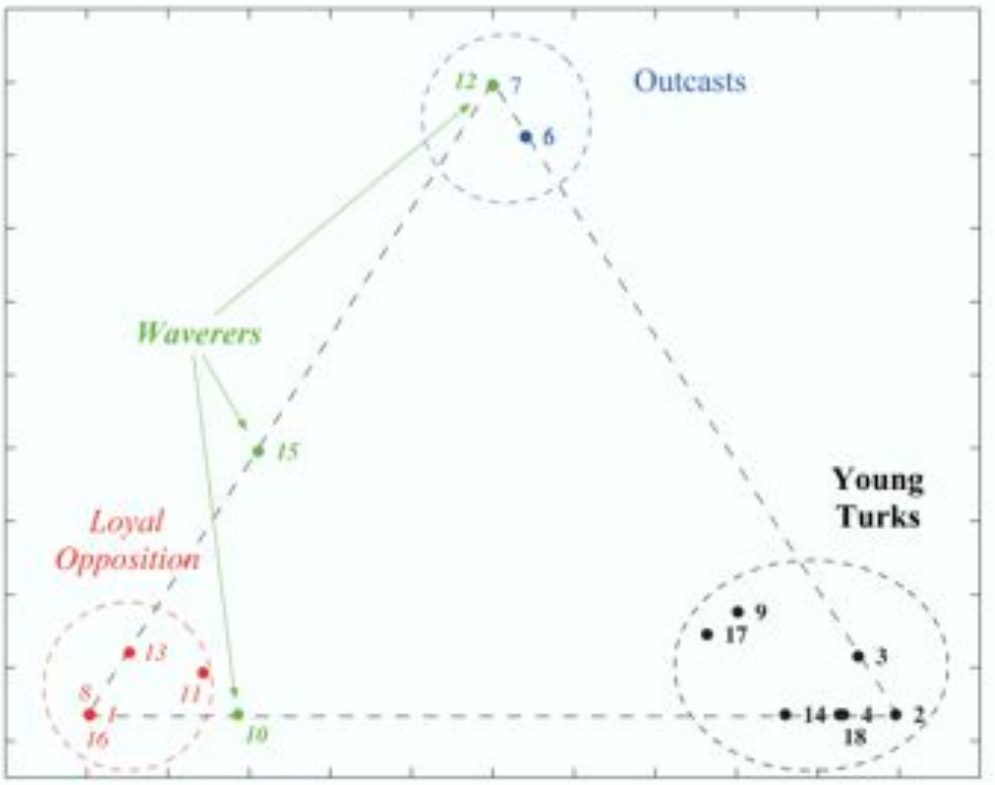
Applications

Computer Vision [*Li Fei-Fei and Perona* 2005]



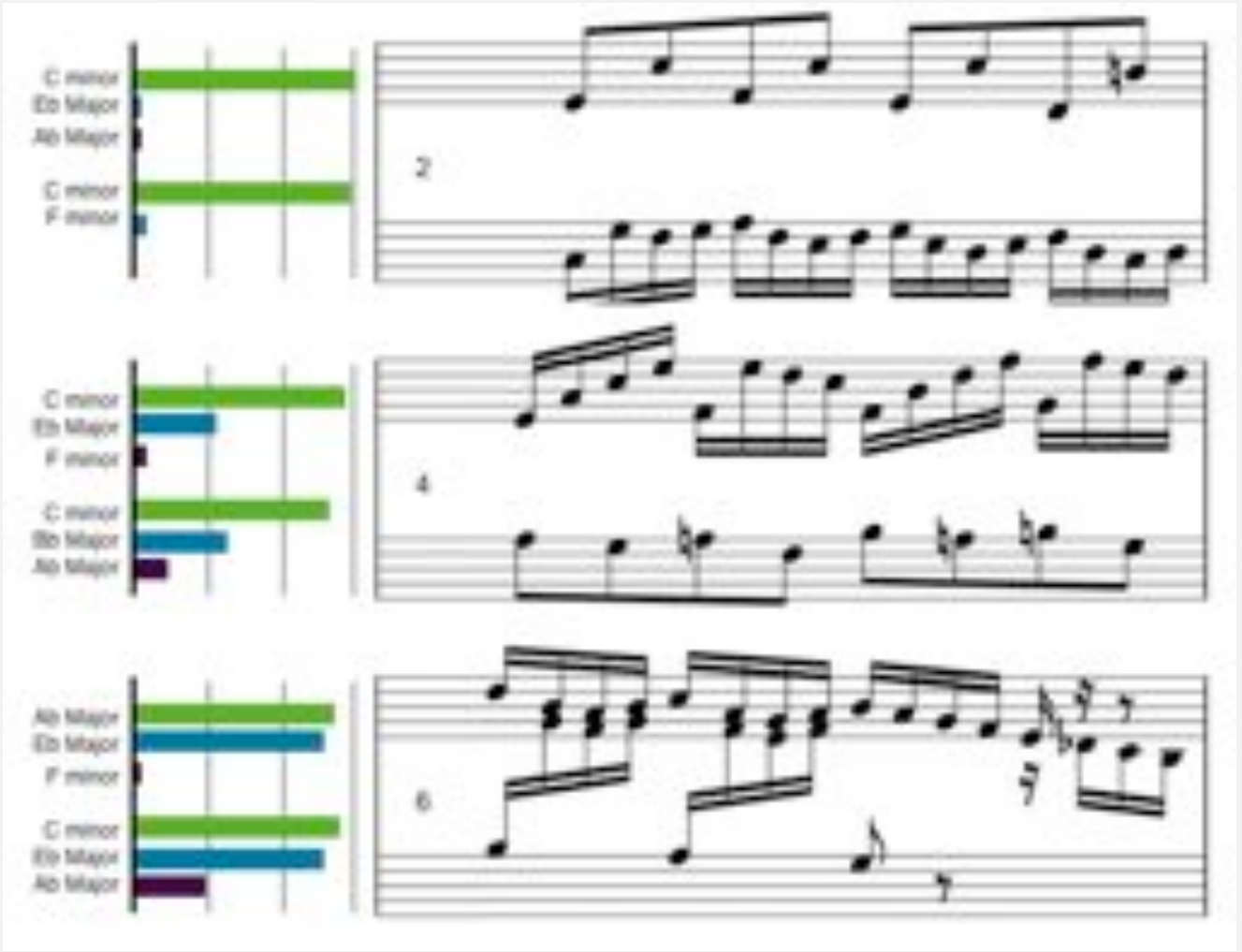
Applications

Social Networks [*Airoldi et al.* 2008]



Applications

Music [*Hu and Saul* 2009]



Why large-scale?

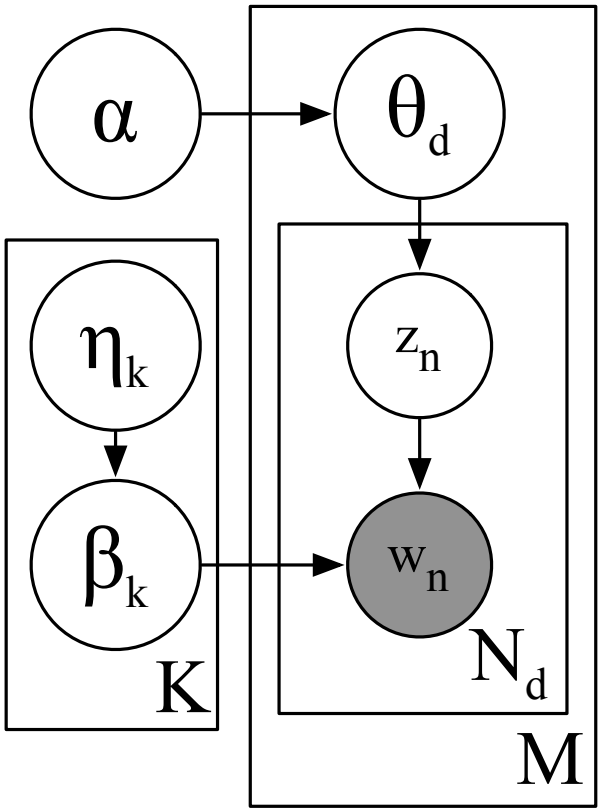
- The most interesting datasets are the **big** ones
- These datasets don't fit on a single machine
- Thus we can't depend on analysis that sits on a single machine

MapReduce

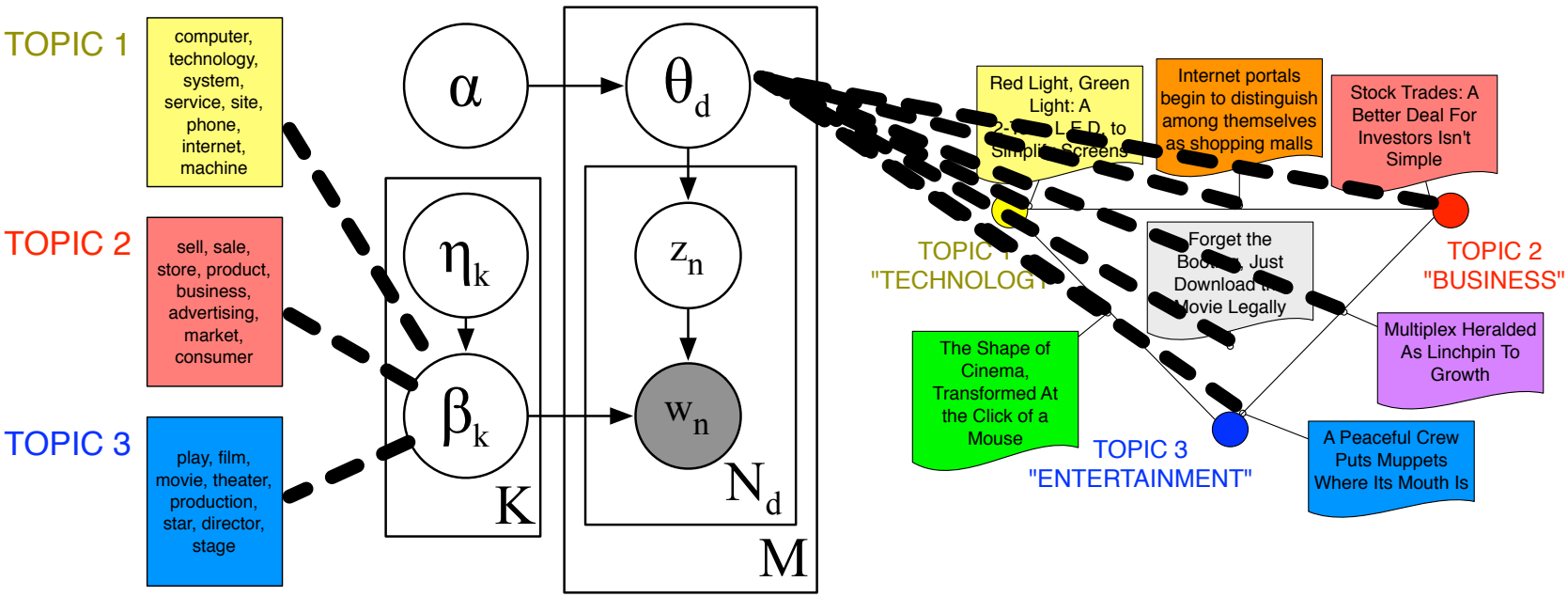
- Framework proposed by Google [*Dean and Ghemawat* 2004]
- Hadoop, OSS implementation by Yahoo [*White* 2010]
- Central concept
  - **Mappers** process small units of data
  - **Reducers** aggregate / combine results of mappers into final result
  - **Drivers** Run a series of jobs to get the work done
  - Overall framework distributes intermediate results where they need to go

Outline

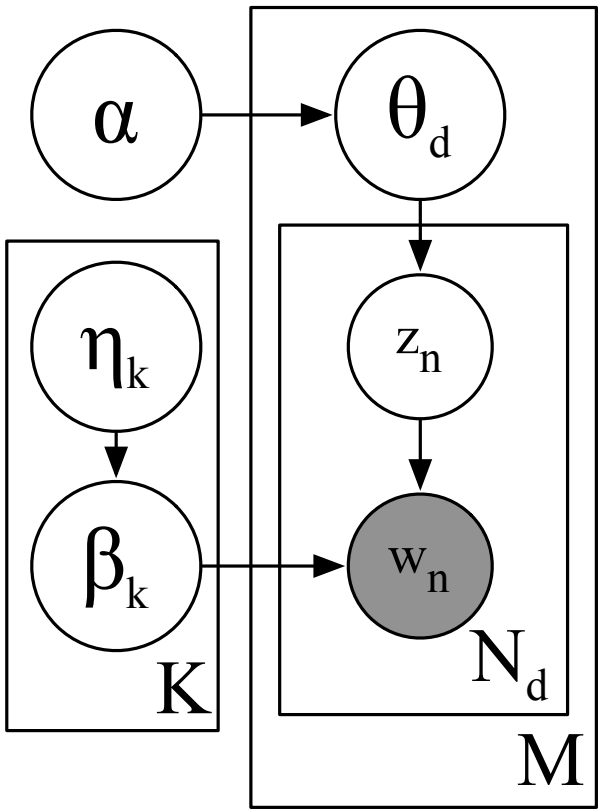
Inference



Inference

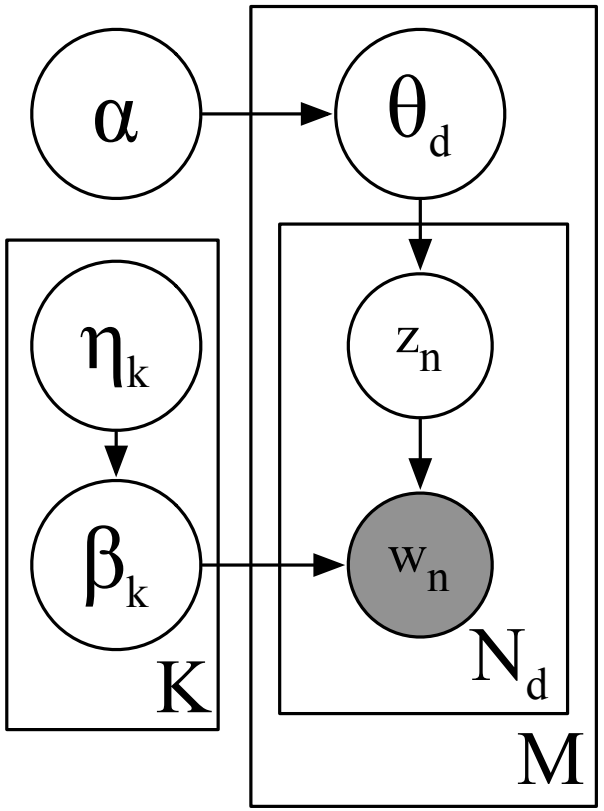

TOPIC 1
computer, technology, system, service, site, phone, internet, machine
TOPIC 2
sell, sale, store, product, business, advertising, market, consumer
TOPIC 3
play, film, movie, theater, production, star, director, stage
α
θd
ηk
zn
βk
wn
K
Nd
M
Red Light, Green Light: A
Internet portals begin to distinguish among themselves as shopping malls
Stock Trades: A Better Deal For Investors Isn't Simple
Forget the Box, Just Download the Movie Legally
Multiplex Heralded As Linchpin To Growth
The Shape of Cinema, Transformed At the Click of a Mouse
A Peaceful Crew Puts Muppets Where Its Mouth Is
TOPIC 2
"BUSINESS"
TOPIC 3
"ENTERTAINMENT"

Inference



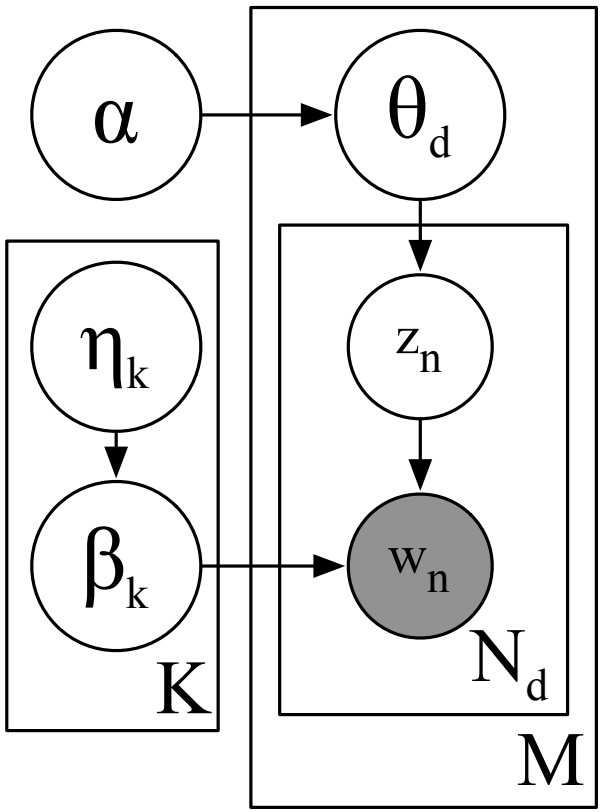
- Generative models tell a story of how your data came to be
- There are missing pieces to that story (e.g. the topics)
- Statistical inference fills in the missing pieces

Inference



- Generative models tell a story of how your data came to be
- There are missing pieces to that story (e.g. the topics)
- Statistical inference fills in the missing pieces
- Hard problem - requires looking at the entire dataset
- Why we need large scale solutions

Inference



- Generative models tell a story of how your data came to be
- There are missing pieces to that story (e.g. the topics)
- Statistical inference fills in the missing pieces
- Hard problem - requires looking at the entire dataset
- Why we need large scale solutions
- Use MapReduce!

Inference

Variational

- Few, expensive iterations
- Deterministic
- Conjugate easier, tractable without
- Easy convergence diagnosis

MCMC / Gibbs

- Many, cheap iterations
- Random
- Effective for **conjugate** distributions
- Tricky convergence diagnosis

Inference

Variational

- First LDA implementation [*Blei et al.* 2003]
- Master-Slave LDA [*Nallapati et al.* 2007]
- Apache Mahout

MCMC / Gibbs

- Popular [*Griffiths and Steyvers* 2004]
- Sparsity helps [*Yao et al.* 2009]
- Assume shared memory? [*Asuncion et al.* 2008]
- YahooLDA [*Smola and Narayanamurthy* 2010]

Expectation Maximization Algorithm

- Input: z (hidden variables), ξ (parameters), D (data)
- Start with initial guess of z, parameters ξ
- Repeat
  - Compute the expected value of latent variables z
  - Compute the parameters ξ that maximize likelihood L (use calculus)
- With each iteration, objective function L goes up

Expectation Maximization Algorithm

- Input: z (hidden variables), ξ (parameters), D (data)
- Start with initial guess of z, parameters ξ
- Repeat
  - **E-Step** Compute the expected value of latent variables z
  - Compute the parameters ξ that maximize likelihood L (use calculus)
- With each iteration, objective function L goes up

Expectation Maximization Algorithm

- Input: z (hidden variables), ξ (parameters), D (data)
- Start with initial guess of z, parameters ξ
- Repeat
  - **E-Step** Compute the expected value of latent variables z
  - **M-Step** Compute the parameters ξ that maximize likelihood L (use calculus)
- With each iteration, objective function L goes up

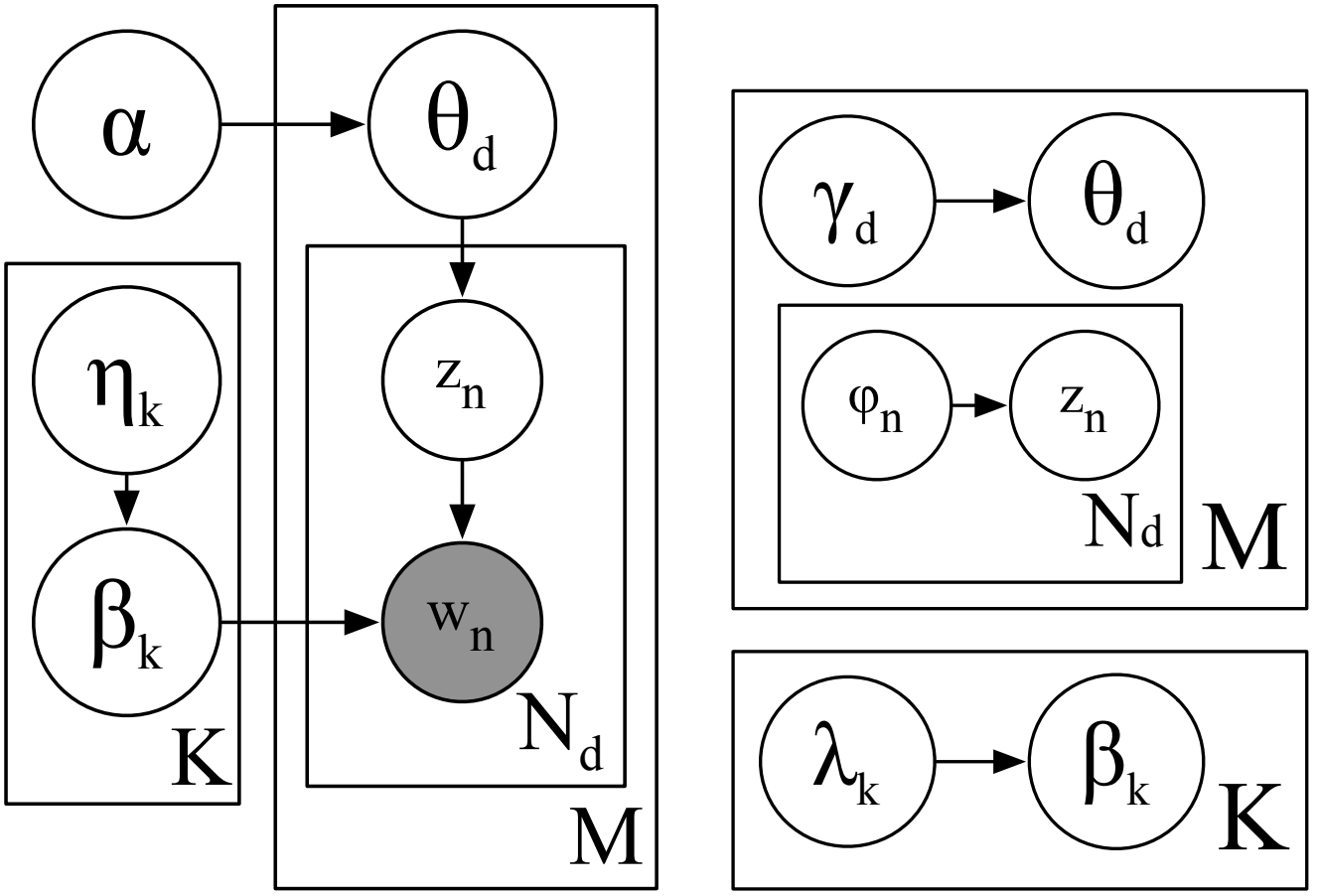
Theory

- Sometimes you can't actually optimize L
- So we instead optimize a lower bound based on a "variational" distribution q

$$\mathcal{L} = \mathbb{E}_q \left[\log \left(p(\mathrm{D}|Z)p(Z|\xi)\right)\right] - \mathbb{E}_q \left[\log q(Z)\right] \tag{1}$$

- $L - \mathcal{L} = KL(q||p)$
- This is called variational EM (normal EM is when $p = q$)
- Makes the math possible to optimize $\mathcal{L}$

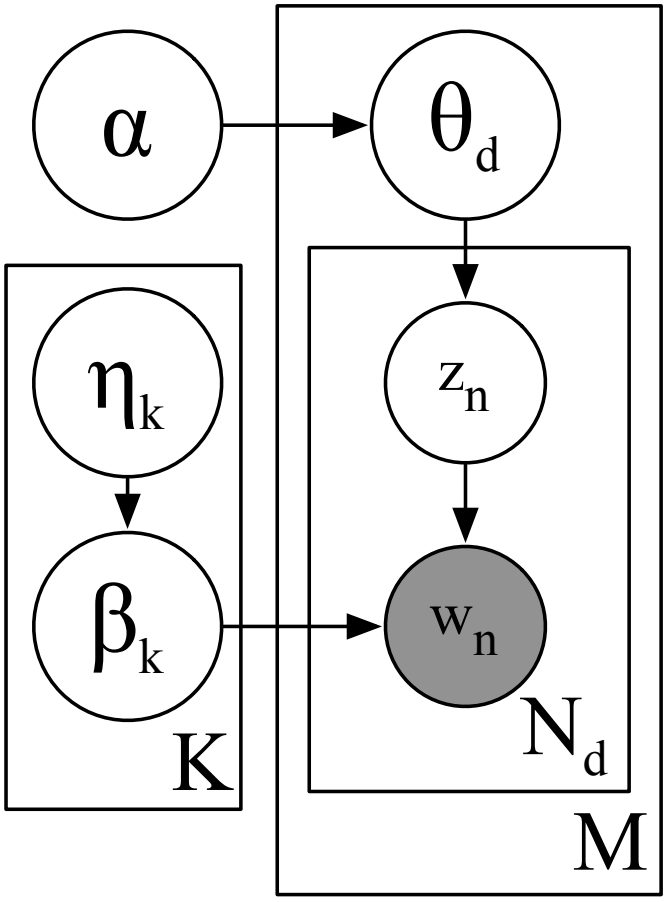
Variational distribution



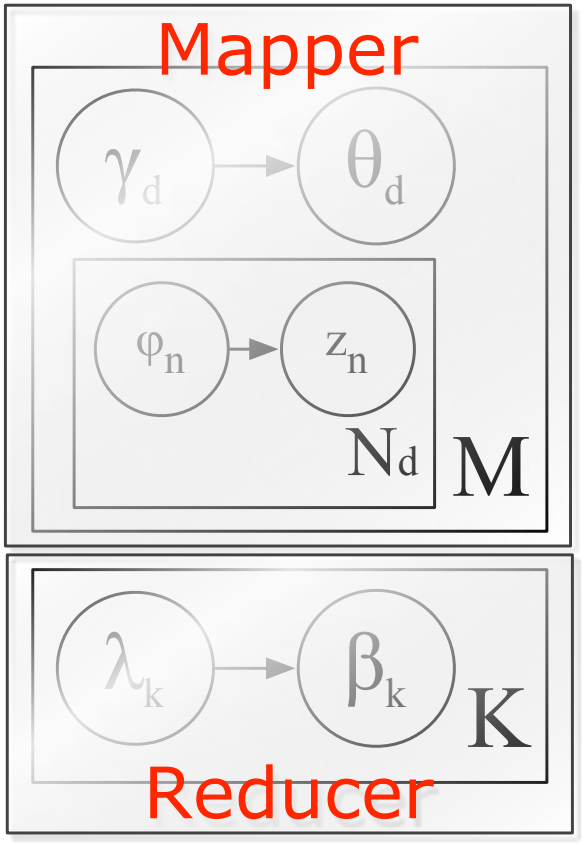
(a) LDA

(b) Variational

Variational distribution



(c) LDA



(d) Variational

Updates - Important Part

- ϕ How much the n^{th} word in a document expressed topic k
- $\gamma_{d,k}$ How much the k^{th} topic is expressed in a document d
- $\beta_{v,k}$ How much word v is associated with topic k

$$\phi_{d,n,k} \propto \beta_{w_{d,n},k} \cdot e^{\Psi(\gamma_k)}$$

$$\gamma_{d,k} = \alpha_k + \sum_{n=1}^{N_d} \phi_{d,n,k},$$

$$\beta_{v,k} \propto \eta + \sum_{d=1}^{C} (w_v^{(d)} \phi_{d,v,k})$$

This is the algorithm!

Updates - Important Part

- ϕ How much the n^{th} word in a document expressed topic k (Mapper)
- $\gamma_{d,k}$ How much the k^{th} topic is expressed in a document d (Mapper)
- $\beta_{v,k}$ How much word v is associated with topic k

$$\phi_{d,n,k} \propto \beta_{w_{d,n},k} \cdot e^{\Psi(\gamma_k)}$$

$$\gamma_{d,k} = \alpha_k + \sum_{n=1}^{N_d} \phi_{d,n,k},$$

$$\beta_{v,k} \propto \eta + \sum_{d=1}^{C} (w_v^{(d)} \phi_{d,v,k})$$

This is the algorithm!

Updates - Important Part

- ϕ How much the n^{th} word in a document expressed topic k (Mapper)
- $\gamma_{d,k}$ How much the k^{th} topic is expressed in a document d (Mapper)
- $\beta_{v,k}$ How much word v is associated with topic k (Reducer)

$$\phi_{d,n,k} \propto \beta_{w_{d,n},k} \cdot e^{\Psi(\gamma_k)}$$

$$\gamma_{d,k} = \alpha_k + \sum_{n=1}^{N_d} \phi_{d,n,k},$$

$$\beta_{v,k} \propto \eta + \sum_{d=1}^{C} (w_v^{(d)} \phi_{d,v,k})$$

This is the algorithm!

Updates - Important Part

- ϕ How much the n^{th} word in a document expressed topic k (Mapper)
- $\gamma_{d,k}$ How much the k^{th} topic is expressed in a document d (Mapper)
- $\beta_{v,k}$ How much word v is associated with topic k (Reducer)

$$\phi_{d,n,k} \propto \beta_{w_{d,n},k} \cdot e^{\Psi(\gamma_k)}$$

$$\gamma_{d,k} = \alpha_k + \sum_{n=1}^{N_d} \phi_{d,n,k},$$

$$\beta_{v,k} \propto \eta + \sum_{d=1}^{C} (w_v^{(d)} \phi_{d,v,k})$$

This is the algorithm!

Other considerations

- Thus far, no difference from Mahout or [*Nallapati et al.* 2007]
- Computing objective function $\mathcal{L}$ to assess convergence
- Updating hyperparameters
  - Many implementations don't do this
  - Critical for topic quality and good likelihood

Objective Function

Expanding Equation 1 gives us $\mathcal{L}(\gamma, \phi; \alpha, \beta)$ for one document:

$$\begin{aligned}\mathcal{L}(\gamma, \phi; \alpha, \beta) &= \sum_{d=1}^{C} \mathcal{L}_d(\gamma, \phi; \alpha, \beta) \\ &= \underbrace{\sum_{d=1}^{C} \mathcal{L}_d(\alpha)}_{\text{Driver}} + \underbrace{\sum_{d=1}^{C} (\underbrace{\mathcal{L}_d(\gamma, \phi) + \mathcal{L}_d(\phi) + \mathcal{L}_d(\gamma)}_{\text{computed in mapper}}),}_{\text{computed in Reducer}}\end{aligned}$$

Updating hyperparameters

We use a Newton-Raphson method which requires the Hessian matrix and the gradient,

$$\alpha_{\text{new}} = \alpha_{\text{old}} - \mathcal{H}^{-1}(\alpha_{\text{old}}) \cdot g(\alpha_{\text{old}}),$$

Updating hyperparameters

We use a Newton-Raphson method which requires the Hessian matrix and the gradient,

$$\alpha_{\text{new}} = \alpha_{\text{old}} - \mathcal{H}^{-1}(\alpha_{\text{old}}) \cdot g(\alpha_{\text{old}}),$$

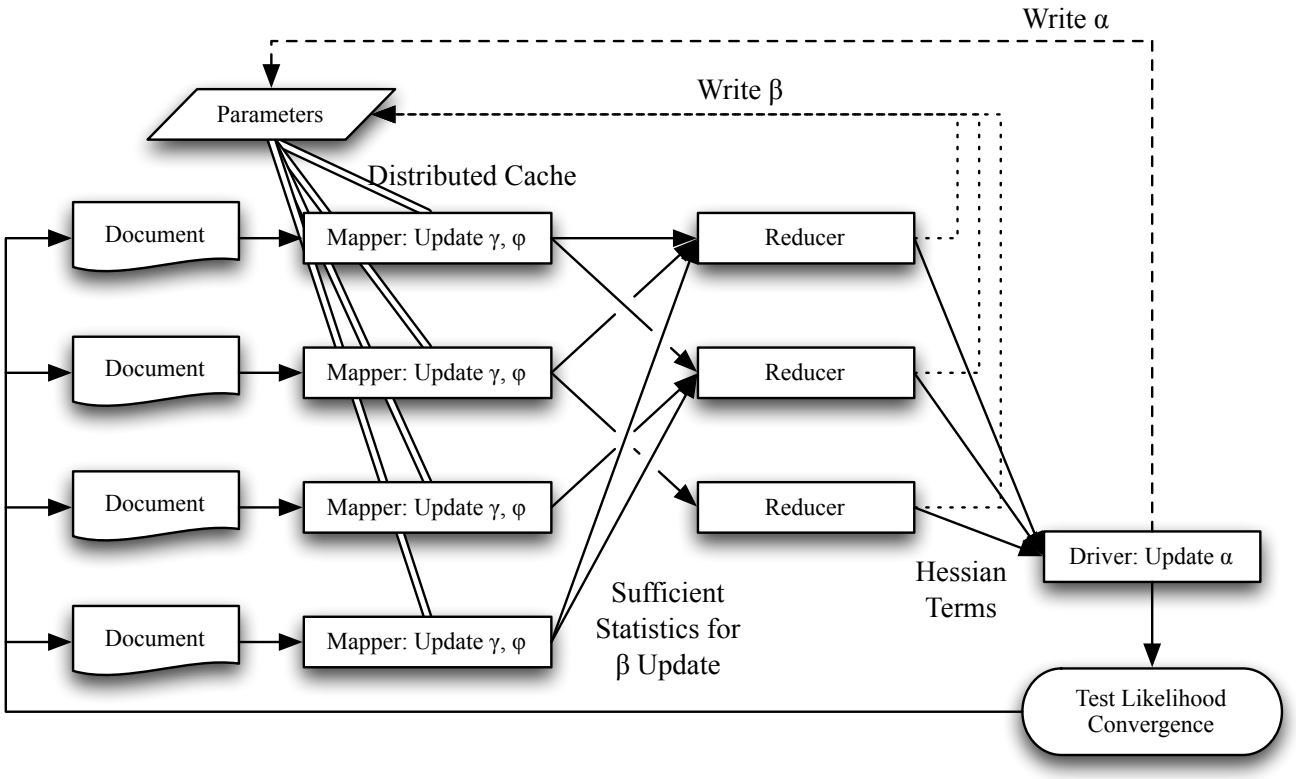
where the Hessian matrix $\mathcal{H}$ and gradient $g(\alpha)$ are

$$\mathcal{H}(k,l) = \delta(k,l) C\Psi'(\alpha_k) - C\Psi'\left(\sum_{l=1}^{K} \alpha_l\right),$$

$$g(k) = \underbrace{C\left(\Psi\left(\sum_{l=1}^{K} \alpha_l\right) - \Psi(\alpha_k)\right)}_{\text{computed in driver}} + \underbrace{\sum_{d=1}^{C} \underbrace{\Psi(\gamma_{d,k}) - \Psi\left(\sum_{l=1}^{K} \gamma_{d,l}\right)}_{\text{computed in mapper}}}_{\text{computed in reducer}}.$$

Complexity

Removing document-dependence: update $O(K^2)$ in the driver


Write α
Write β
Parameters
Distributed Cache
Document
Document
Document
Document
Mapper: Update γ, φ
Mapper: Update γ, φ
Mapper: Update γ, φ
Mapper: Update γ, φ
Reducer
Reducer
Reducer
Sufficient
Statistics for
β Update
Hessian
Terms
Driver: Update α
Test Likelihood
Convergence

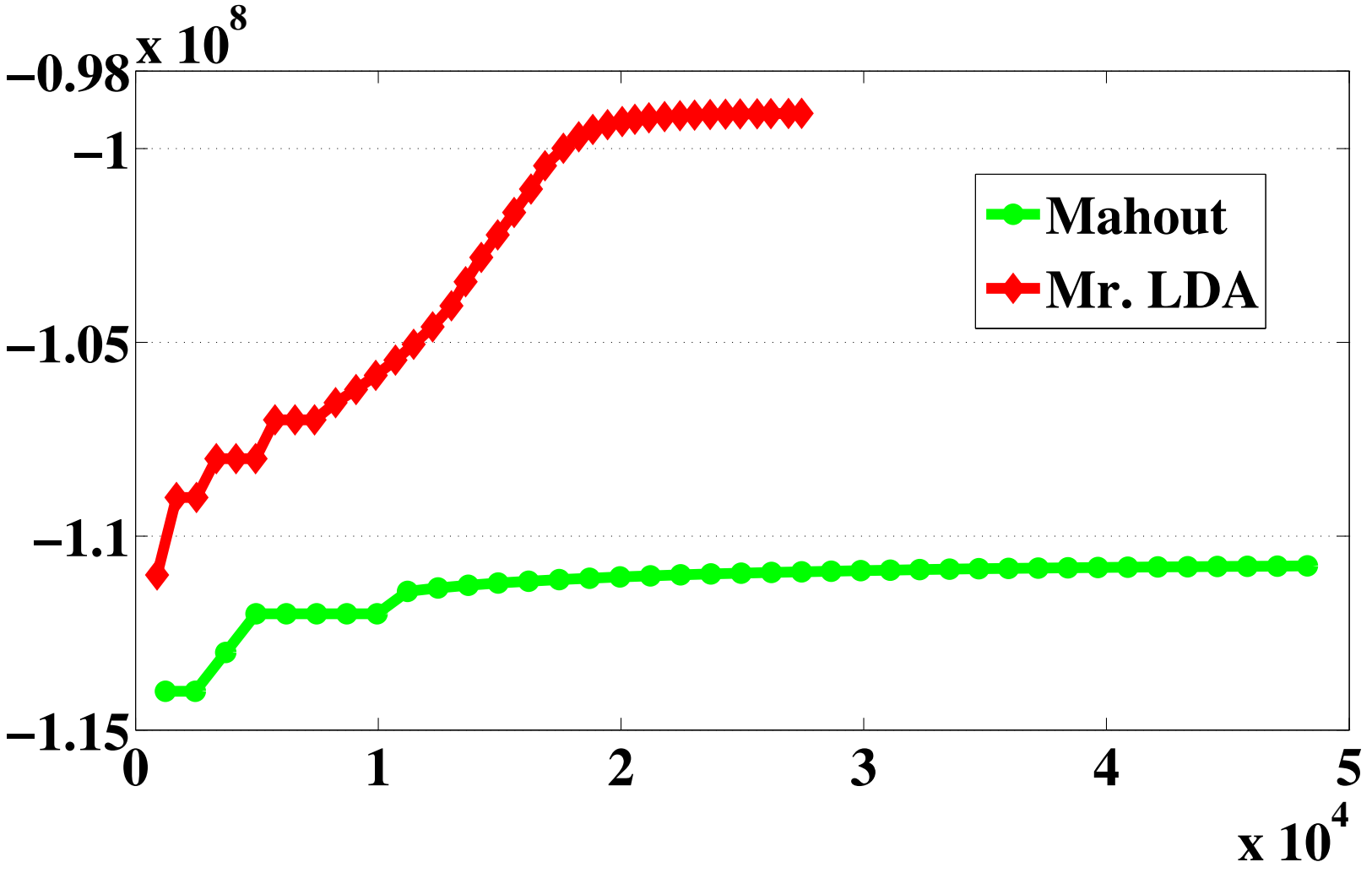
Other implementation details

- Computing Ψ function is expensive, so we cache / approximate values
- The number of intermediate values swamp the system, so we employ in-mapper combiners [*Lin and Dyer* 2010]
- Initialization

Other implementation details

- Computing Ψ function is expensive, so we cache / approximate values
  - Always helps
- The number of intermediate values swamp the system, so we employ in-mapper combiners [*Lin and Dyer* 2010]
  - Only helps with many topics
- Initialization
  - Helps in first iterations

Comparison with Mahout



Held-out likelihood vs. time (sec)
TREC (100 topics, 500k documents)

Outline

How are psychological factors expressed in blogs?

- Linguistic Inquiry in Word Count [*Pennebaker and Francis* 1999]
- Example psychological processes:
  - Anger: hate, kill, annoyed
  - Negative Emotions: hurt, ugly, nasty
- What words cooccur with these words in a particular *corpus*?

How are psychological factors expressed in blogs?

- Linguistic Inquiry in Word Count [*Pennebaker and Francis* 1999]
- Example psychological processes:
  - Anger: hate, kill, annoyed
  - Negative Emotions: hurt, ugly, nasty
- What words cooccur with these words in a particular *corpus*?
- Use LIWC categories as an informed prior to "seed" topics

$$\beta_{v,k} \propto \eta_{v,k} + \sum_{d=1}^{C} (w_v^{(d)} \phi_{d,v,k})$$

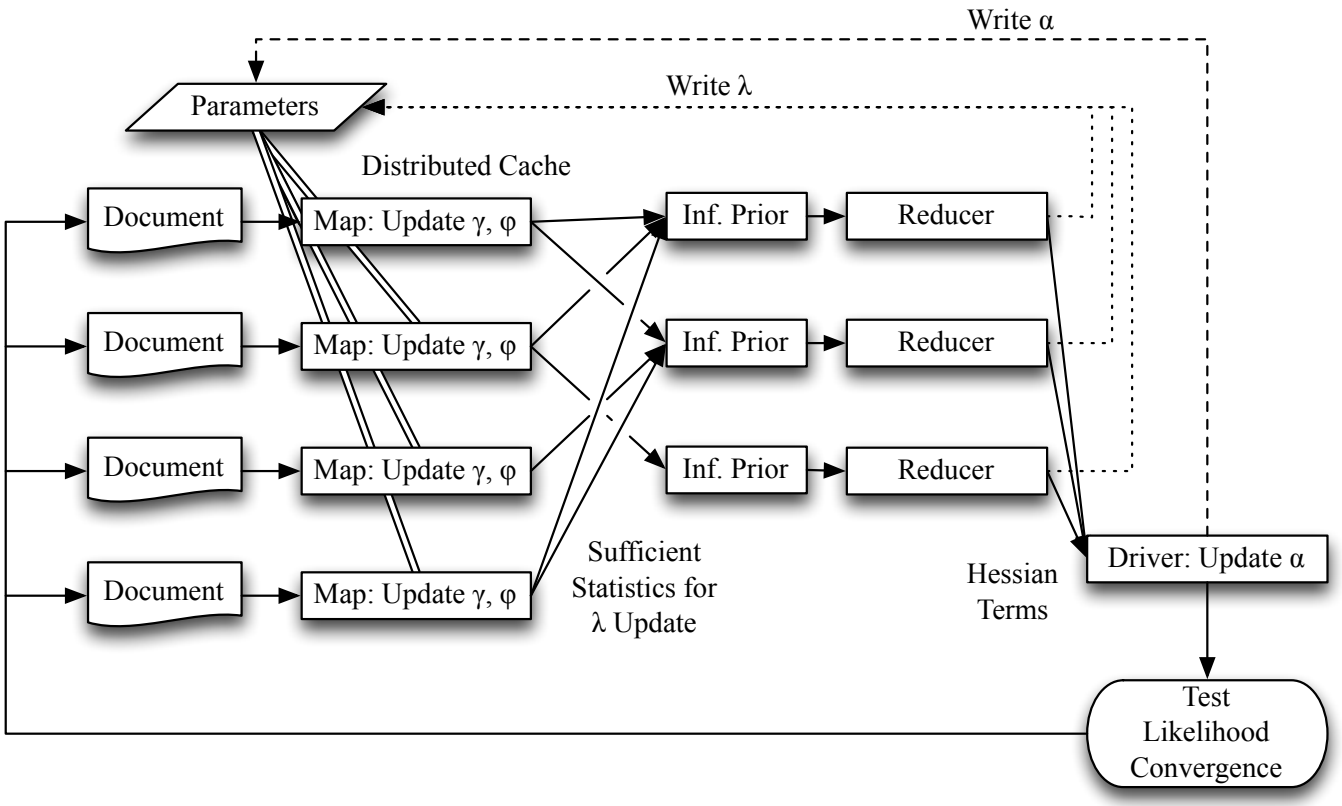
How are psychological factors expressed in blogs?

- Linguistic Inquiry in Word Count [*Pennebaker and Francis* 1999]
- Example psychological processes:
  - Anger: hate, kill, annoyed
  - Negative Emotions: hurt, ugly, nasty
- What words cooccur with these words in a particular *corpus*?
- Use LIWC categories as an informed prior to "seed" topics

$$\beta_{v,k} \propto \eta_{v,k} + \sum_{d=1}^{C} (w_v^{(d)} \phi_{d,v,k})$$

- Not possible in SparseLDA-based models

Workflow for Informed Prior



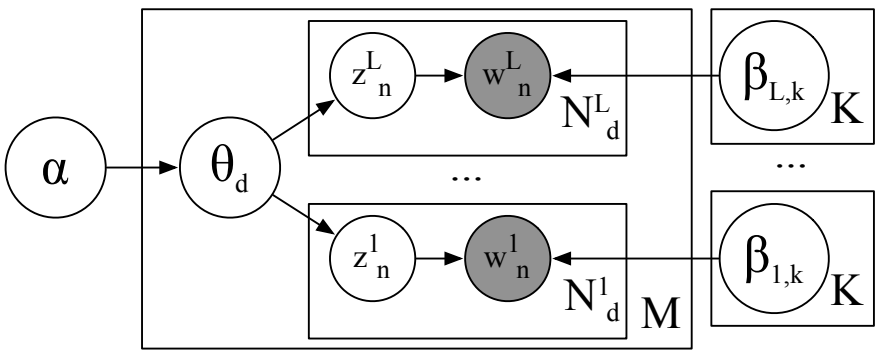
Psychologically-Informed Topics from Blogs

Affective Processes	Negative Emotions	Positive Emotions	Anxiety	Anger	Sadness
easili	sorri	lord	bird	iraq	level
dare	crappi	prayer	diseas	american	grief
truli	bullshit	pray	shi	countri	disord
lol	goddamn	merci	infect	militari	moder
needi	messi	etern	blood	nation	miseri
jealousi	shitti	truli	snake	unit	lbs
friendship	bitchi	humbl	anxieti	america	loneli
betray	angri	god	creatur	force	pain

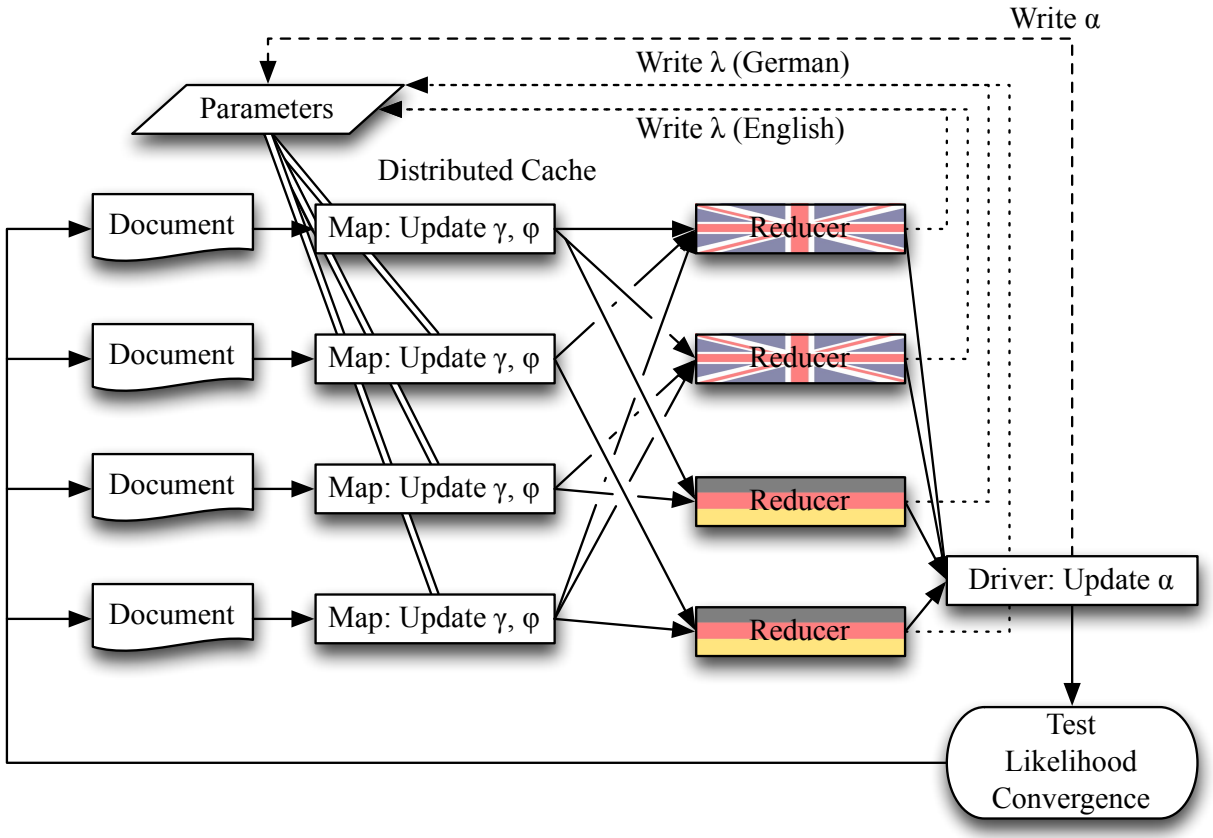
Using 50 topics on Blog Authorship corpus [*Koppel et al.* 2006]

Polylingual LDA

- Assumes documents have multiple "faces" [*Mimno et al.* 2009]
- Topics also assumed to have per-language distribution
- As long as documents talk about the same thing, learns consistent topics across languages
- First variational inference algorithm



Workflow for Polylingual LDA



Aligned topics from all of Wikipedia

English	game	opera	greek	league	said	italian	soviet
	games	musical	turkish	cup	family	church	political
	player	composer	region	club	could	pope	military
	players	orchestra	hugarian	played	childern	italy	union
	released	piano	wine	football	death	catholic	russian
	comics	works	hungary	games	father	bishop	power
	characters	symphony	greece	career	wrote	roman	israel
	character	instruments	turkey	game	mother	rome	empire
	version	composers	ottoman	championship	never	st	republic
German	spiel	musik	ungarn	saison	frau	papst	regierung
	spieler	komponist	turkei	gewann	the	rom	republik
	serie	oper	turkischen	spielte	familie	ii	sowjetunion
	the	komponisten	griechenland	karriere	mutter	kirche	kam
	erschien	werke	rumanien	fc	vater	di	krieg
	gibt	orchester	ungarischen	spielen	leben	bishof	land
	commics	wiener	griechischen	wechselte	starb	italien	bevolkerung
	veroffentlic	komposition	istanbul	mannschaft	tod	italienisch	ende
	2	klavier	serbien	olympischen	kinder	konig	reich

Which large-scale implementation is right for me?

- Yahoo LDA [*Smola and Narayanamurthy* 2010]
  - Fastest
  - Sparse Gibbs sampling
  - Great when you can use `memcached`
- Mahout
  - Variational
  - Simplest
- **Mr LDA**
  - Designed for extensibility
  - Multilingual
  - Hyperparameter updating [*Wallach et al.* 2009]
  - Likelihood monitoring

Conclusion

- Mr LDA: A scalable implementation for topic modeling
- Extensible variational inference
- Next steps
  - Supporting more modeling assumptions (including non-conjugacy)
  - Nonparametrics (over topics and vocabulary)
  - Multiple starts

Download the Code

`http://mrlda.cc`

Ke Zhai

- First author
- Immigration issues prevented presentation

MR LDA

- MR = MapReduce
- LDA = latent Dirichlet allocation

Ke Zhai

- First author
- Immigration issues prevented presentation

MR LDA

- MR = MapReduce
- LDA = latent Dirichlet allocation
- MR LDA = Ke

Merci!

- Jimmy Lin
- NSF #1018625


Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing.
2008.
Mixed membership stochastic blockmodels.
Journal of Machine Learning Research, 9:1981–2014.


Arthur Asuncion, Padhraic Smyth, and Max Welling.
2008.
Asynchronous distributed learning of topic models.
In *Proceedings of Advances in Neural Information Processing Systems*.


David M. Blei, Andrew Ng, and Michael Jordan.
2003.
Latent Dirichlet allocation.
Journal of Machine Learning Research, 3:993–1022.


Jeffrey Dean and Sanjay Ghemawat.
2004.
MapReduce: Simplified data processing on large clusters.
pages 137–150, San Francisco, California.


Thomas L. Griffiths and Mark Steyvers.
2004.
Finding scientific topics.
Proceedings of the National Academy of Sciences, 101(Suppl 1):5228–5235.


Thomas Hofmann.
1999.
Probabilistic latent semantic analysis.
In *Proceedings of Uncertainty in Artificial Intelligence*.


Diane Hu and Lawrence K. Saul.

Map$(d, \vec{w})$

repeat
  for all $v \in [1, V]$ **do**
    for all $k \in [1, K]$ **do**
      Update $\phi_{v,k} = \beta_{v,k} \times \exp(\Psi(\gamma_{d,k}))$.
    end for
    Normalize row $\phi_{v,*}$, such that $\sum_{k=1}^{K} \phi_{v,k} = 1$.
    Update $\sigma = \sigma + \vec{w}_v \phi_v$, where ϕ_v is a K-dimensional vector, and $\vec{w}_v$ is the count of v in this document.
  end for
  Update row vector $\gamma_{d,*} = \alpha + \sigma$.
until convergence
for all $k \in [1, K]$ **do**
  for all $v \in [1, V]$ **do**
    Emit key-value pair $\langle k, \triangle \rangle : \vec{w}_v \phi_v$.
    Emit key-value pair $\langle k, v \rangle : \vec{w}_v \phi_v$. {order inversion}
  end for
  Emit key-value pair $\langle \triangle, k \rangle : (\Psi(\gamma_{d,k}) - \Psi(\sum_{l=1}^{K} \gamma_{d,l}))$.
  {emit the γ-tokens for α update}
  Output key-value pair $\langle k, d \rangle - \gamma_{d,k}$ to file.
end for
Emit key-value pair $\langle \triangle, \triangle \rangle - \mathcal{L}$, where $\mathcal{L}$ is log-likelihood of this document.

Map$(d, \vec{w})$

repeat
  for all $v \in [1, V]$ **do**
    for all $k \in [1, K]$ **do**
      Update $\phi_{v,k} = \beta_{v,k} \times \exp(\Psi(\gamma_{d,k}))$.
    end for
    Normalize row $\phi_{v,*}$, such that $\sum_{k=1}^{K} \phi_{v,k} = 1$.
    Update $\sigma = \sigma + \vec{w}_v \phi_v$, where ϕ_v is a K-dimensional vector, and $\vec{w}_v$ is the count of v in this document.
  end for
  Update row vector $\gamma_{d,*} = \alpha + \sigma$.
until convergence
for all $k \in [1, K]$ **do**
  for all $v \in [1, V]$ **do**
    Emit key-value pair $\langle k, \triangle \rangle : \vec{w}_v \phi_v$.
    Emit key-value pair $\langle k, v \rangle : \vec{w}_v \phi_v$. {order inversion}
  end for
  Emit key-value pair $\langle \triangle, k \rangle : (\Psi(\gamma_{d,k}) - \Psi(\sum_{l=1}^{K} \gamma_{d,l}))$.
  {emit the γ-tokens for α update}
  Output key-value pair $\langle k, d \rangle - \gamma_{d,k}$ to file.
end for
Emit key-value pair $\langle \triangle, \triangle \rangle - \mathcal{L}$, where $\mathcal{L}$ is log-likelihood of this document.

Map$(d, \vec{w})$

repeat
  for all $v \in [1, V]$ **do**
    for all $k \in [1, K]$ **do**
      Update $\phi_{v,k} = \beta_{v,k} \times \exp(\Psi(\gamma_{d,k}))$.
    end for
    Normalize row $\phi_{v,*}$, such that $\sum_{k=1}^{K} \phi_{v,k} = 1$.
    Update $\sigma = \sigma + \vec{w}_v \phi_v$, where ϕ_v is a K-dimensional vector, and $\vec{w}_v$ is the count of v in this document.
  end for
  Update row vector $\gamma_{d,*} = \alpha + \sigma$.
until convergence
for all $k \in [1, K]$ **do**
  for all $v \in [1, V]$ **do**
    Emit key-value pair $\langle k, \triangle \rangle : \vec{w}_v \phi_v$.
    Emit key-value pair $\langle k, v \rangle : \vec{w}_v \phi_v$. {order inversion}
  end for
  Emit key-value pair $\langle \triangle, k \rangle : (\Psi(\gamma_{d,k}) - \Psi(\sum_{l=1}^{K} \gamma_{d,l}))$.
  {emit the γ-tokens for α update}
  Output key-value pair $\langle k, d \rangle - \gamma_{d,k}$ to file.
end for
Emit key-value pair $\langle \triangle, \triangle \rangle - \mathcal{L}$, where $\mathcal{L}$ is log-likelihood of this document.

Input:
KEY - key pair $\langle p_{\text{left}}, p_{\text{right}} \rangle$.
VALUE - an iterator $\mathcal{I}$ over sequence of values.

Reduce

Compute the sum σ over all values in the sequence $\mathcal{I}$.
if $p_{\text{left}} = \triangle$ **then**
  if $p_{\text{right}} = \triangle$ **then**
    Output key-value pair $\langle \triangle, \triangle \rangle - \sigma$ to file.
    {output the model likelihood $\mathcal{L}$ for convergence checking}
  else
    Output key-value pair $\langle \triangle, p_{\text{right}} \rangle - \sigma$ to file.
    {output the γ-tokens to update α-vectors, Section **??**}
  end if
else
  if $p_{\text{right}} = \triangle$ **then**
    Update the normalization factor $n = \sigma$. {order inversion}
  else
    Output key-value pair $\langle k, v \rangle : \frac{\sigma}{n}$. {output normalized β value}
  end if
end if

Input:
KEY - key pair $\langle p_{\text{left}}, p_{\text{right}} \rangle$.
VALUE - an iterator $\mathcal{I}$ over sequence of values.

Reduce

Compute the sum σ over all values in the sequence $\mathcal{I}$.
if $p_{\text{left}} = \triangle$ **then**
    if $p_{\text{right}} = \triangle$ **then**
        Output key-value pair $\langle \triangle, \triangle \rangle - \sigma$ to file.
        {output the model likelihood $\mathcal{L}$ for convergence checking}
    else
        Output key-value pair $\langle \triangle, p_{\text{right}} \rangle - \sigma$ to file.
        {output the γ-tokens to update α-vectors, Section **??**}
    end if
else
    if $p_{\text{right}} = \triangle$ **then**
        Update the normalization factor $n = \sigma$. {order inversion}
    else
        Output key-value pair $\langle k, v \rangle : \frac{\sigma}{n}$. {output normalized β value}
    end if
end if

Input:
KEY - key pair $\langle p_{\text{left}}, p_{\text{right}} \rangle$.
VALUE - an iterator $\mathcal{I}$ over sequence of values.

Reduce

Compute the sum σ over all values in the sequence $\mathcal{I}$.
if $p_{\text{left}} = \triangle$ **then**
    if $p_{\text{right}} = \triangle$ **then**
        Output key-value pair $\langle \triangle, \triangle \rangle - \sigma$ to file.
        {output the model likelihood $\mathcal{L}$ for convergence checking}
    else
        Output key-value pair $\langle \triangle, p_{\text{right}} \rangle - \sigma$ to file.
        {output the γ-tokens to update α-vectors, Section **??**}
    end if
else
    if $p_{\text{right}} = \triangle$ **then**
        Update the normalization factor $n = \sigma$. {order inversion}
    else
        Output key-value pair $\langle k, v \rangle : \frac{\sigma}{n}$. {output normalized β value}
    end if
end if