Templates

for scalable data analysis

3 Distributed Latent Variable Models

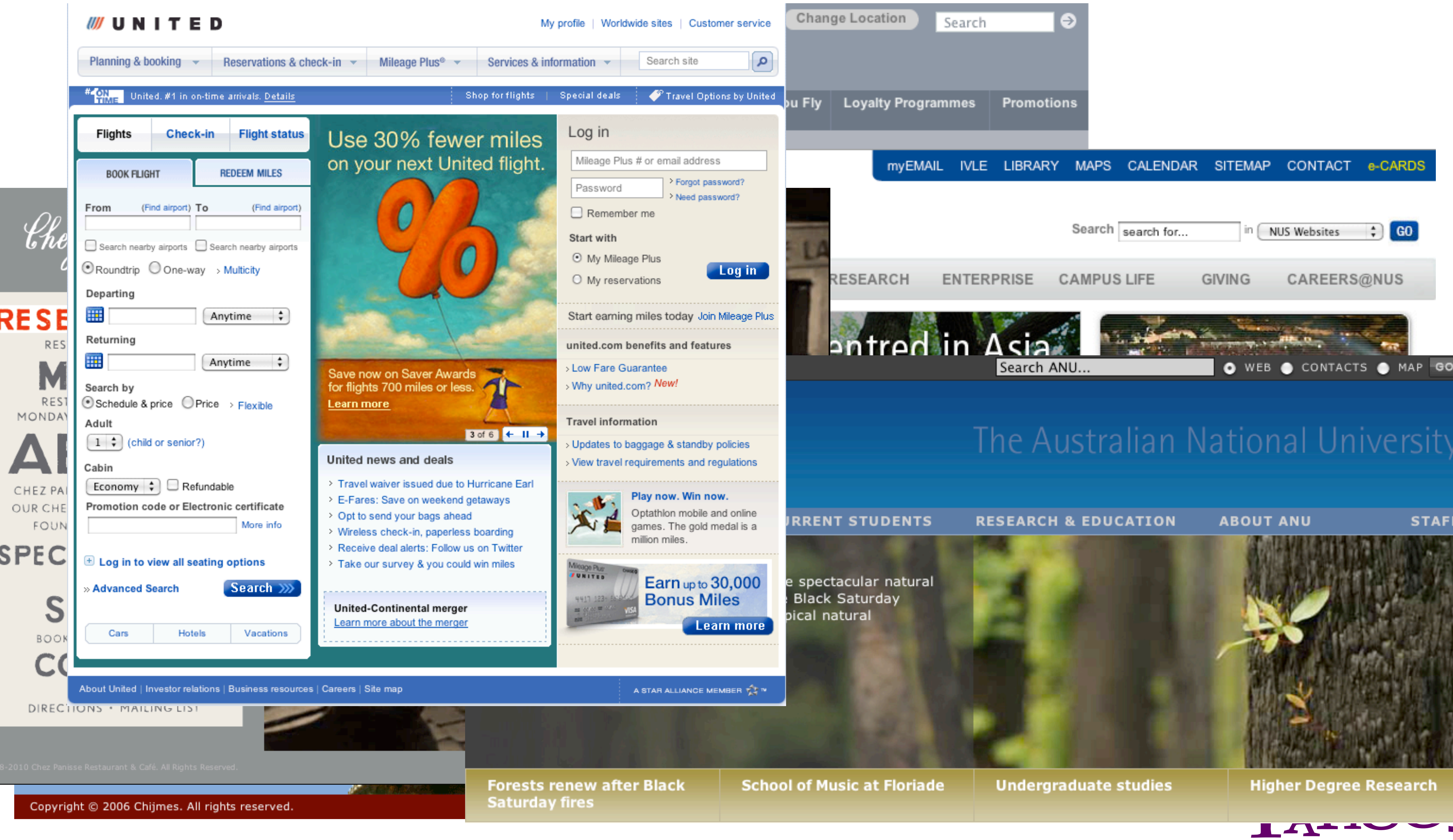
Amr Ahmed, Alexander J Smola, Markus Weimer
Yahoo! Research & UC Berkeley & ANU

- Variations on a theme
  inference for mixtures
- Parallel inference
  parallelization templates
- Samplers
  scaling up LDA

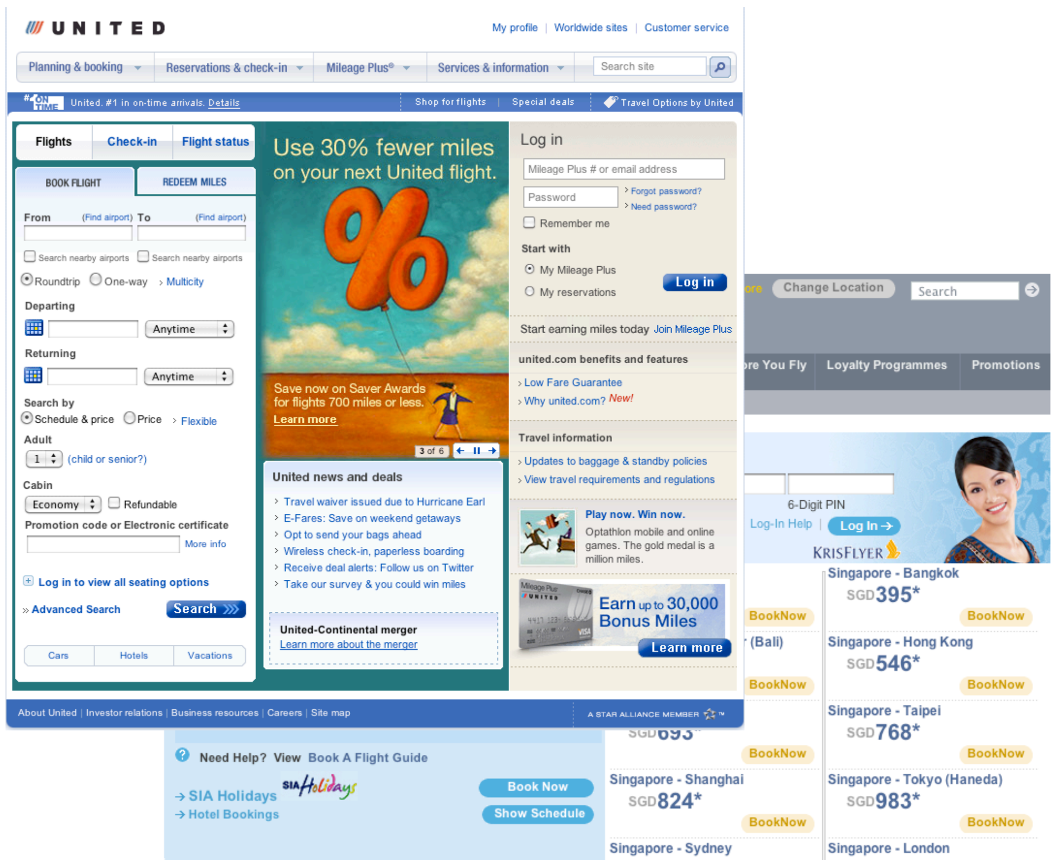

MAGIC Etch A Sketch SCREEN
Inference for
Mixture Models
OHIO ART

Clustering

Clustering

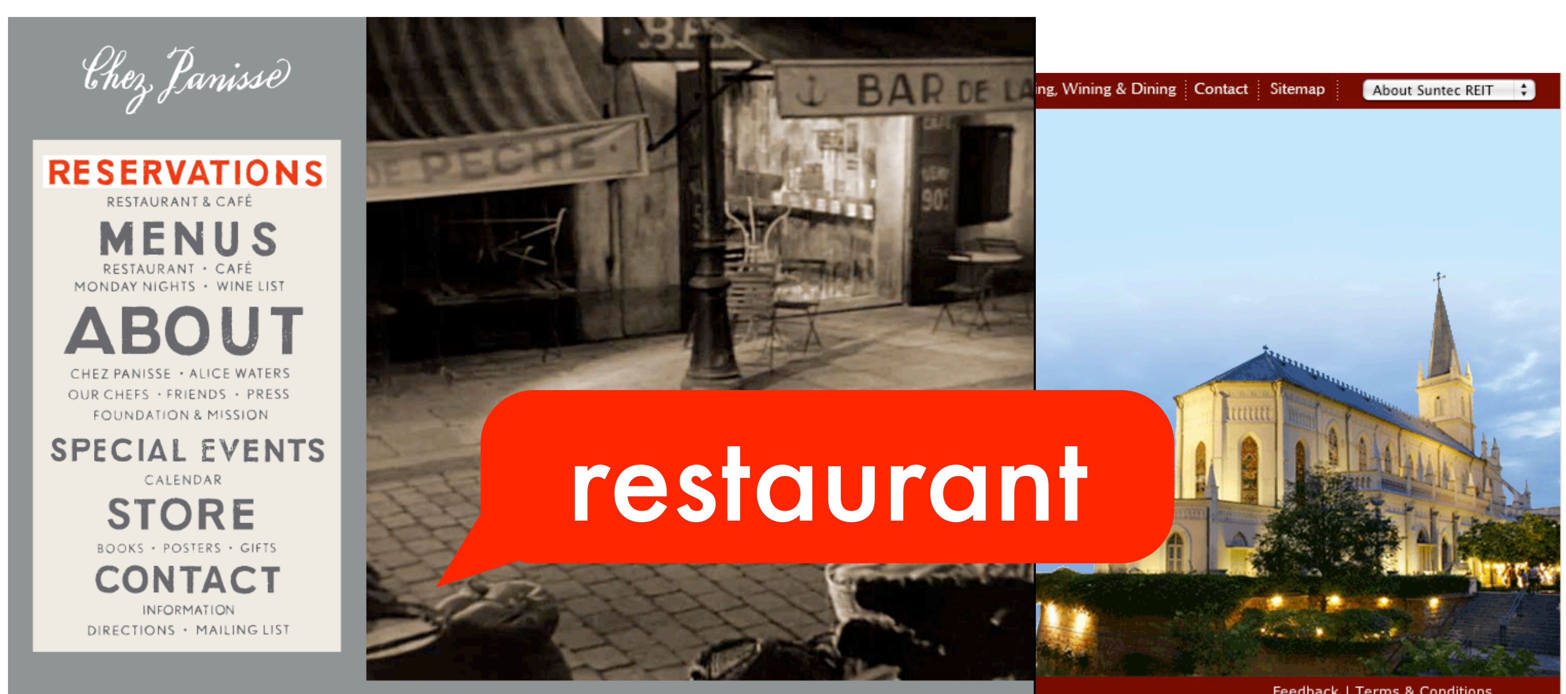

UNITED
My profile | Worldwide sites | Customer service
Planning & booking
Reservations & check-in
Mileage Plus®
Services & information
Search site
United. #1 in on-time arrivals. Details
Shop for flights
Special deals
Travel Options by United
Flights
Check-in
Flight status
BOOK FLIGHT
REDEEM MILES
From (Find airport) To (Find airport)
Search nearby airports
Search nearby airports
Roundtrip
One-way
Multicity
Departing
Anytime
Returning
Anytime
Search by
Schedule & price
Price
Flexible
Adult
(child or senior?)
Cabin
Economy
Refundable
Promotion code or Electronic certificate
More info
Log in to view all seating options
Advanced Search
Search
Cars
Hotels
Vacations
Use 30% fewer miles on your next United flight.
Save now on Saver Awards for flights 700 miles or less.
Learn more
3 of 6
United news and deals
Travel waiver issued due to Hurricane Earl
E-Fares: Save on weekend getaways
Opt to send your bags ahead
Wireless check-in, paperless boarding
Receive deal alerts: Follow us on Twitter
Take our survey & you could win miles
United-Continental merger
Learn more about the merger
Log in
Mileage Plus # or email address
Password
Forgot password?
Need password?
Remember me
Start with
My Mileage Plus
My reservations
Log in
Start earning miles today Join Mileage Plus
united.com benefits and features
Low Fare Guarantee
Why united.com? New!
Travel information
Updates to baggage & standby policies
View travel requirements and regulations
Play now. Win now.
Optathlon mobile and online games. The gold medal is a million miles.
Earn up to 30,000 Bonus Miles
Learn more
About United | Investor relations | Business resources | Careers | Site map
A STAR ALLIANCE MEMBER
Change Location
Search
Loyalty Programmes
Promotions
myEMAIL
IVLE
LIBRARY
MAPS
CALENDAR
SITEMAP
CONTACT
e-CARDS
Search
search for...
in
NUS Websites
GO
RESEARCH
ENTERPRISE
CAMPUS LIFE
GIVING
CAREERS@NUS
Search ANU...
WEB
CONTACTS
MAP
GO
The Australian National University
RESEARCH & EDUCATION
ABOUT ANU
STAFF
Forests renew after Black Saturday fires
School of Music at Floriade
Undergraduate studies
Higher Degree Research
DIRECTIONS • MAILING LIST
Copyright © 2006 Chijmes. All rights reserved.

Clustering

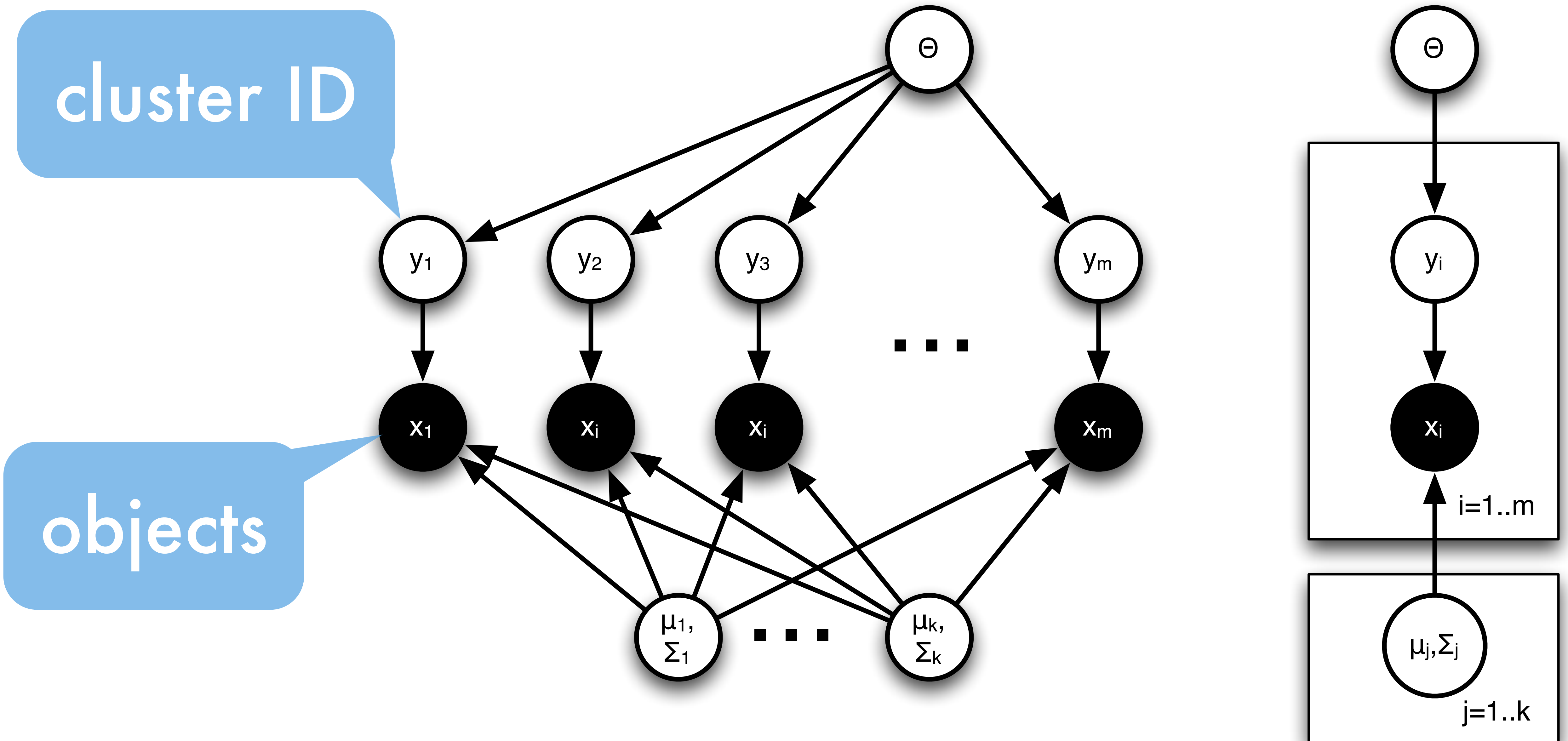


YAHOO!

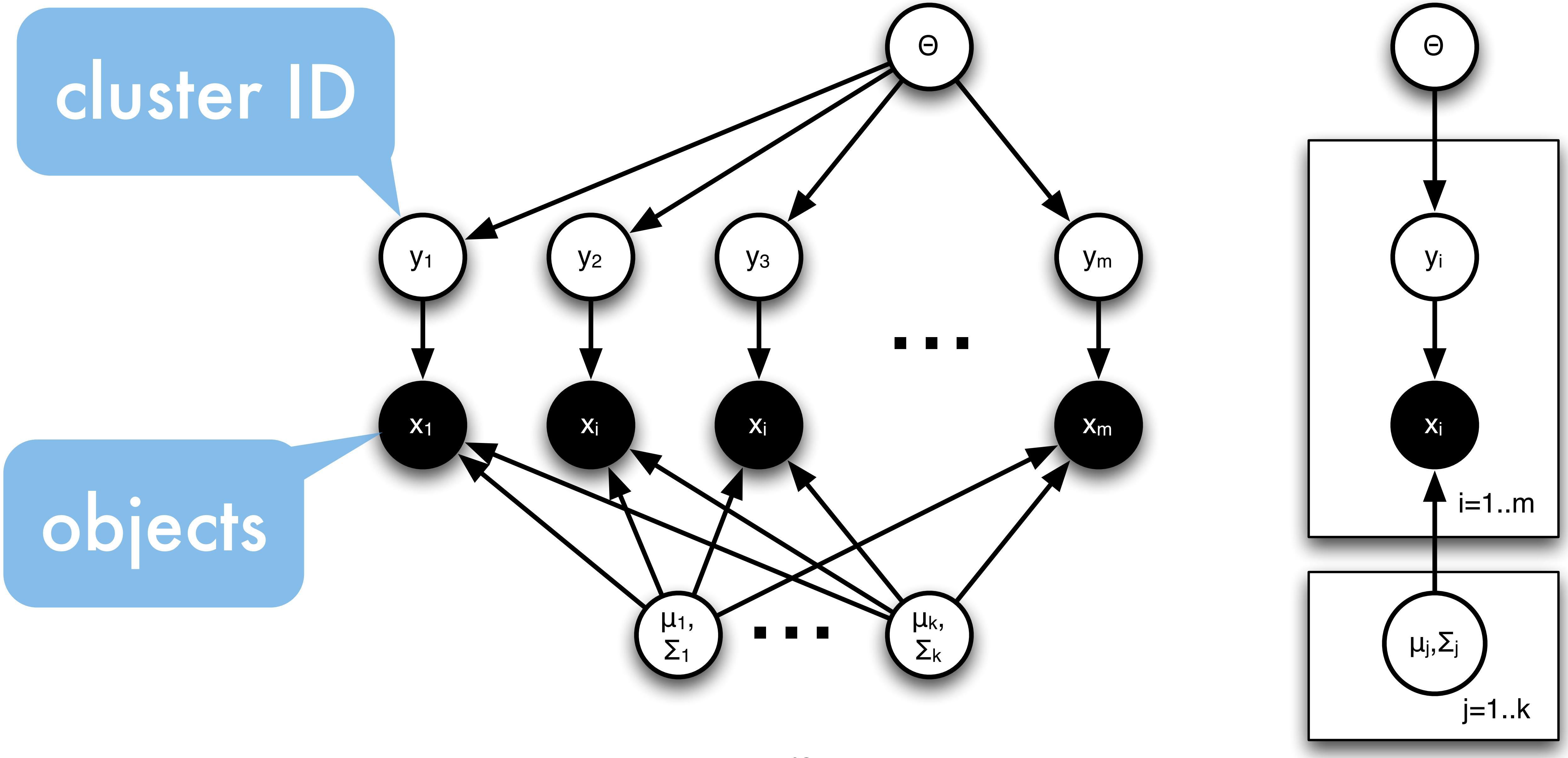
Clustering



Generative Model



Generative Model

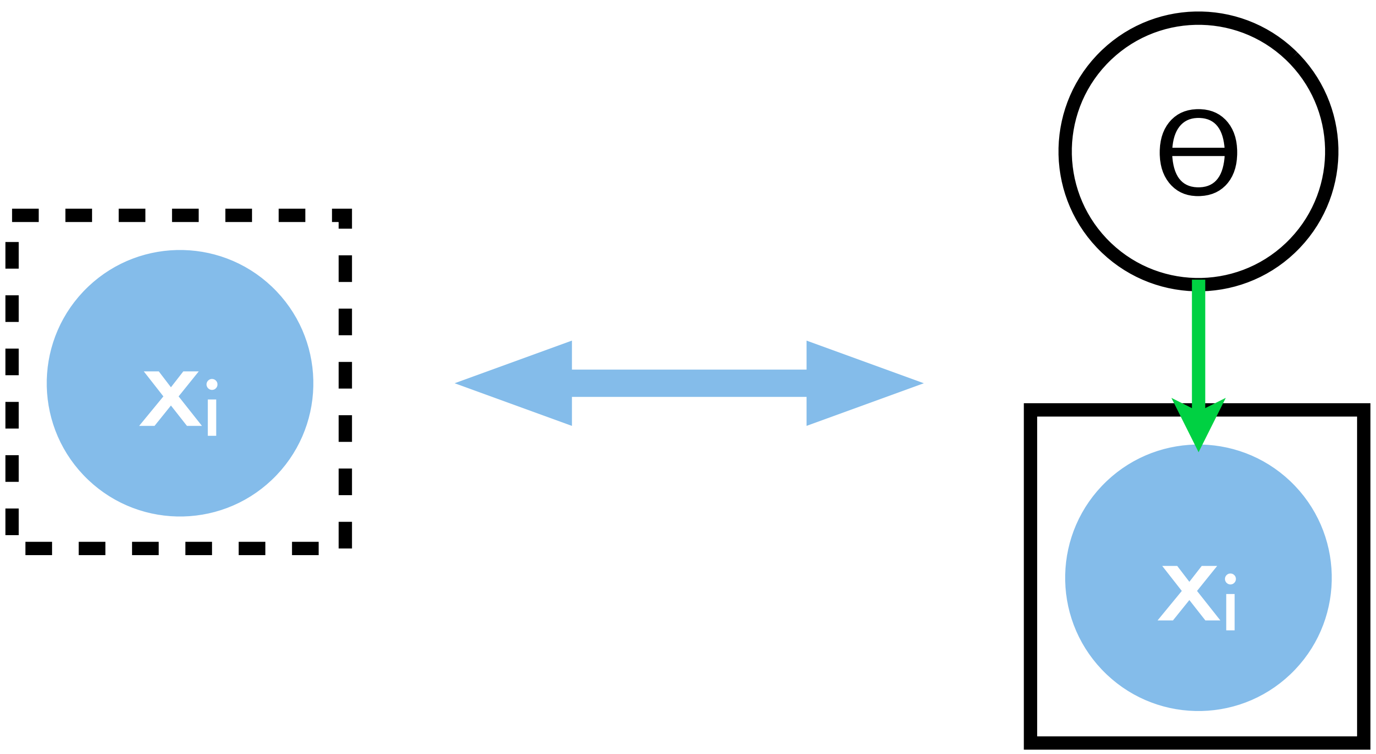


$$p(X, Y|\theta, \sigma, \mu) = \prod_{i=1}^{n} p(x_i|y_i, \sigma, \mu)p(y_i|\theta)$$

deFinetti

Any distribution over exchangeable random variables can be written as conditionally independent.

$$p(x_1, \ldots, x_n) = \int dp(\theta) \prod_{i=1}^{n} p(x_i|\theta)$$



Inference should be easy - **$\Theta|x_i$ and $x_i|\Theta$**

Conjugates and Collapsing

- **Exponential Family**

$$p(x|\theta) = \exp\left(\langle \phi(x), \theta \rangle - g(\theta)\right)$$

- **Conjugate Prior**

$$p(\theta|\mu_0, m_0) = \exp\left(m_0 \langle \mu_0, \theta \rangle - m_0 g(\theta) - h(m_0\mu_0, m_0)\right)$$

- **Posterior**

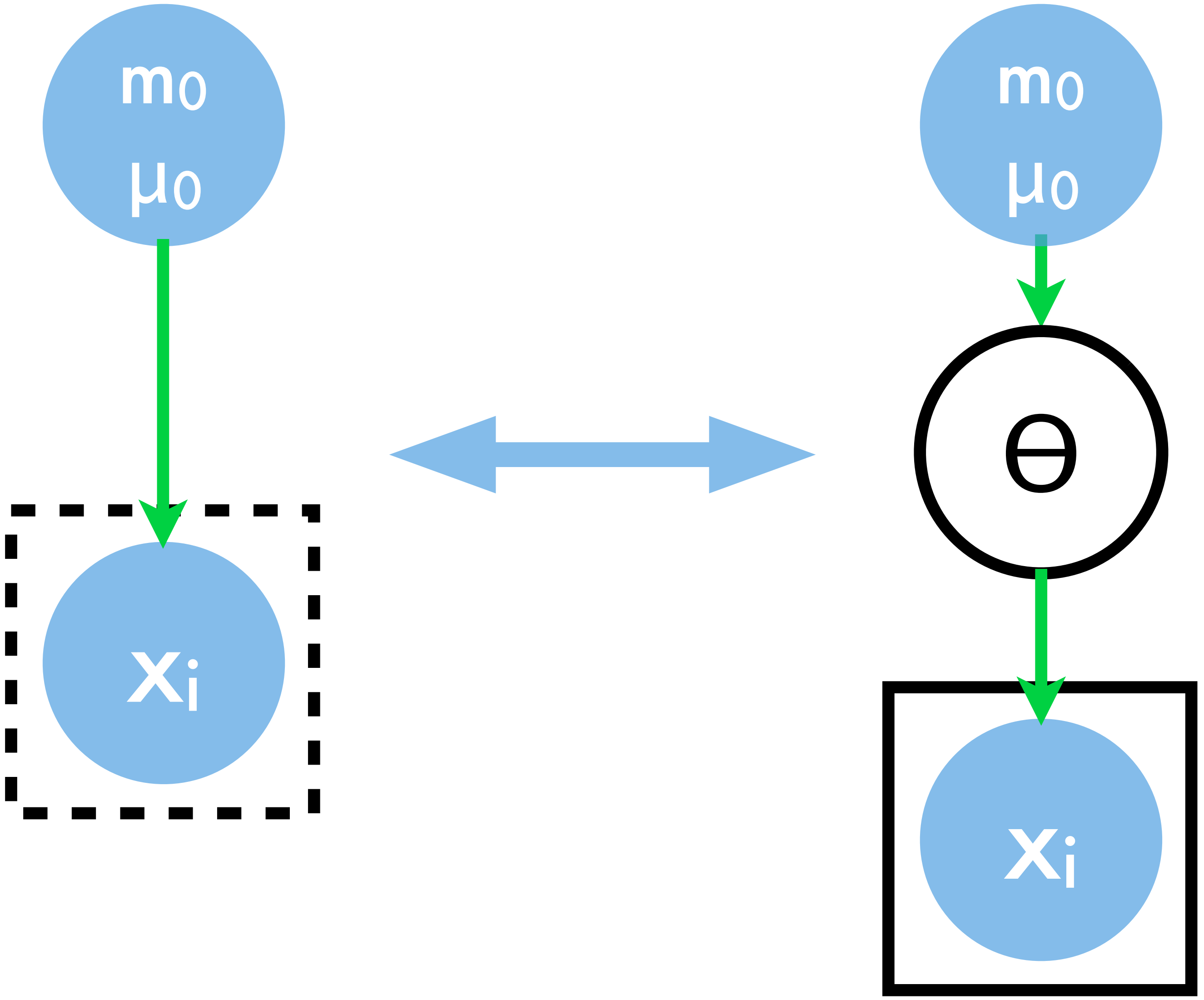
$$p(\theta|X, \mu_0, m_0) \propto \exp\left(\langle m_0\mu_0 + m\mu[X], \theta \rangle - (m_0 + m)g(\theta) - h(m_0\mu_0, m_0)\right)$$

- **Collapsing the natural parameter**

$$p(X|\mu_0, m_0) = \exp\left(h(m_0\mu_0 + m\mu[X], m_0 + m) - h(m_0\mu_0, m_0)\right)$$

data

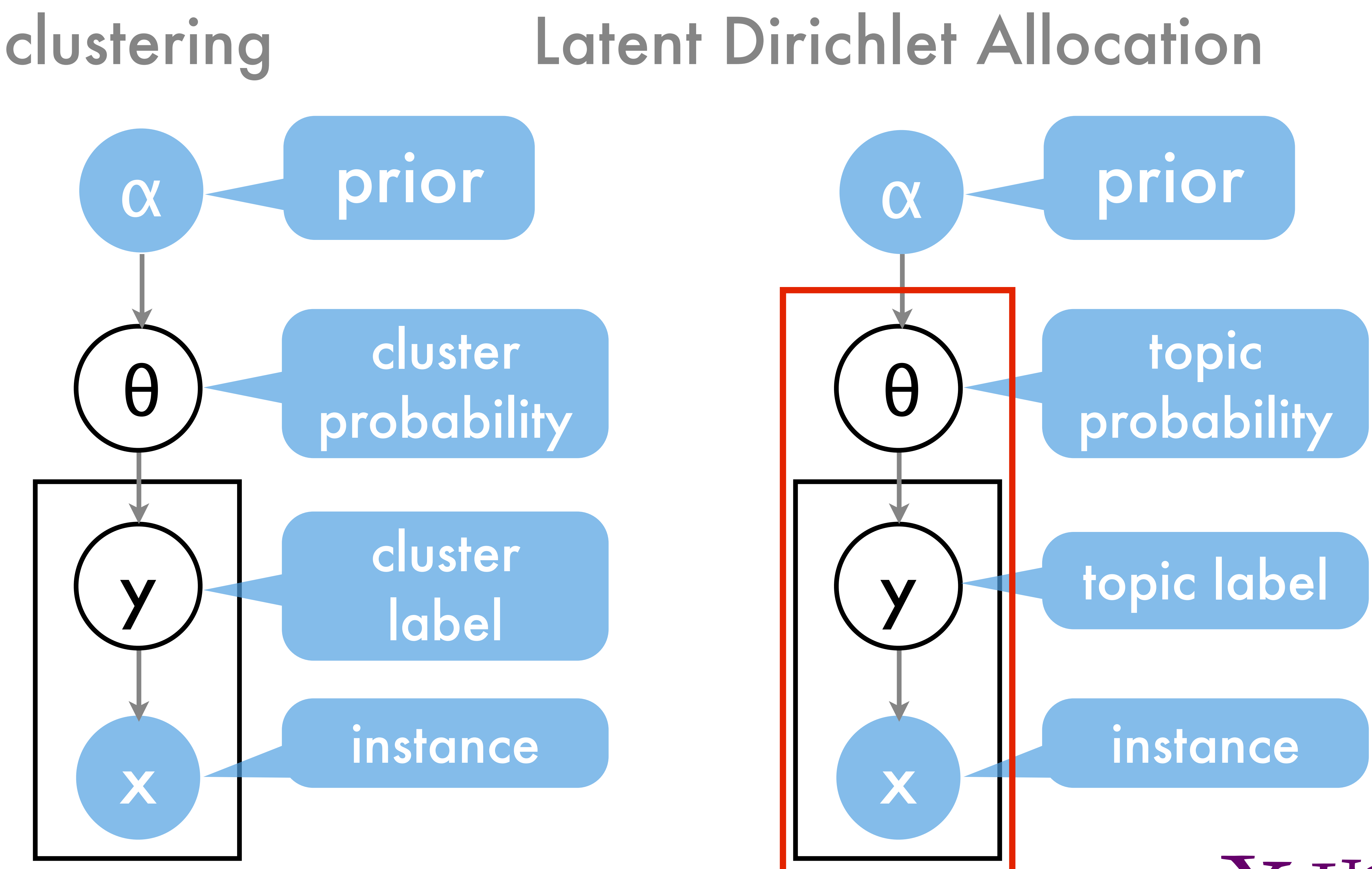
Conjugates and Collapsing



collapsed representation

deFinetti

Clustering & Topic Models



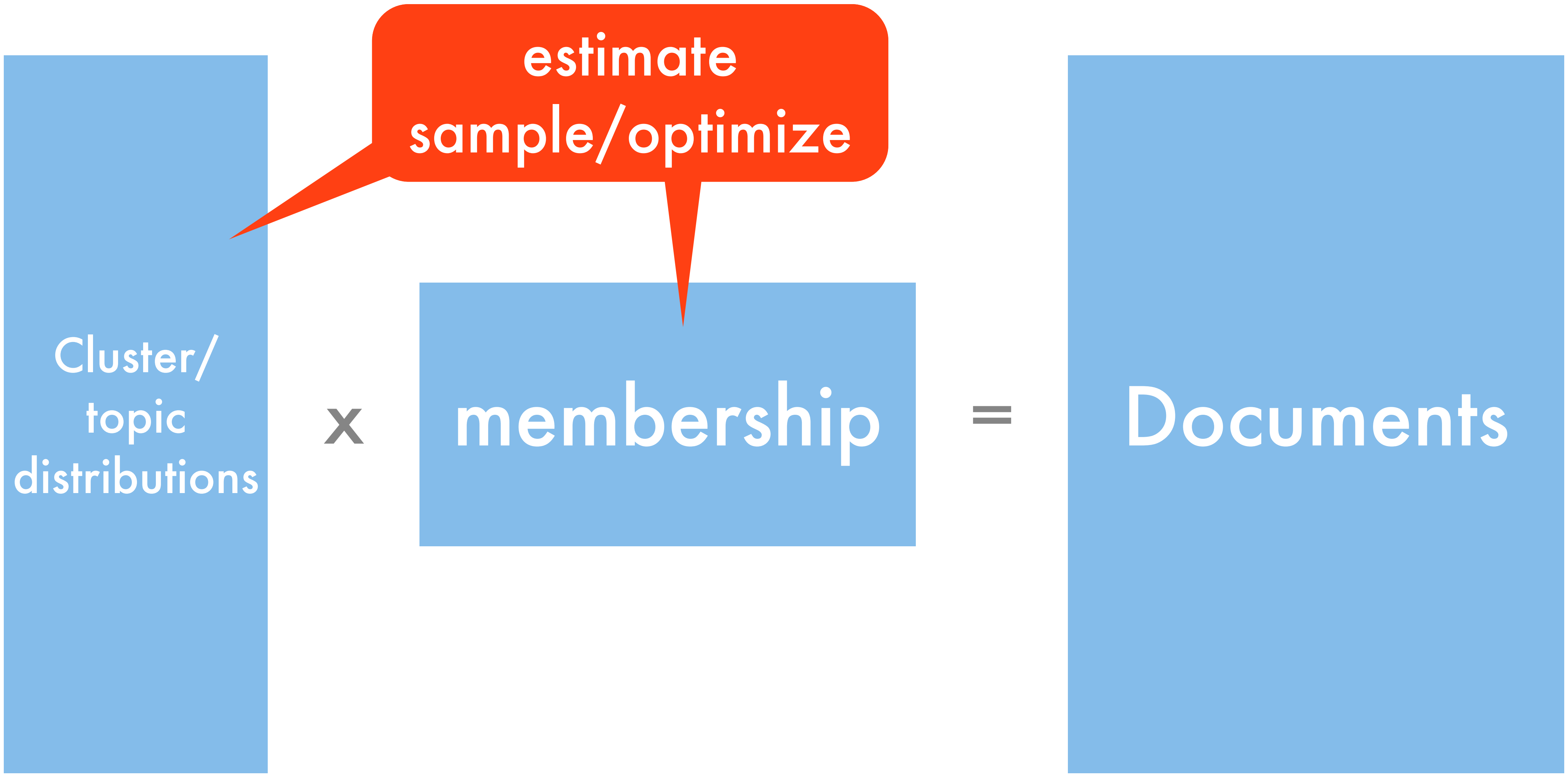
YAHOO!

Clustering & Topic Models

Cluster/ topic distributions x membership = Documents

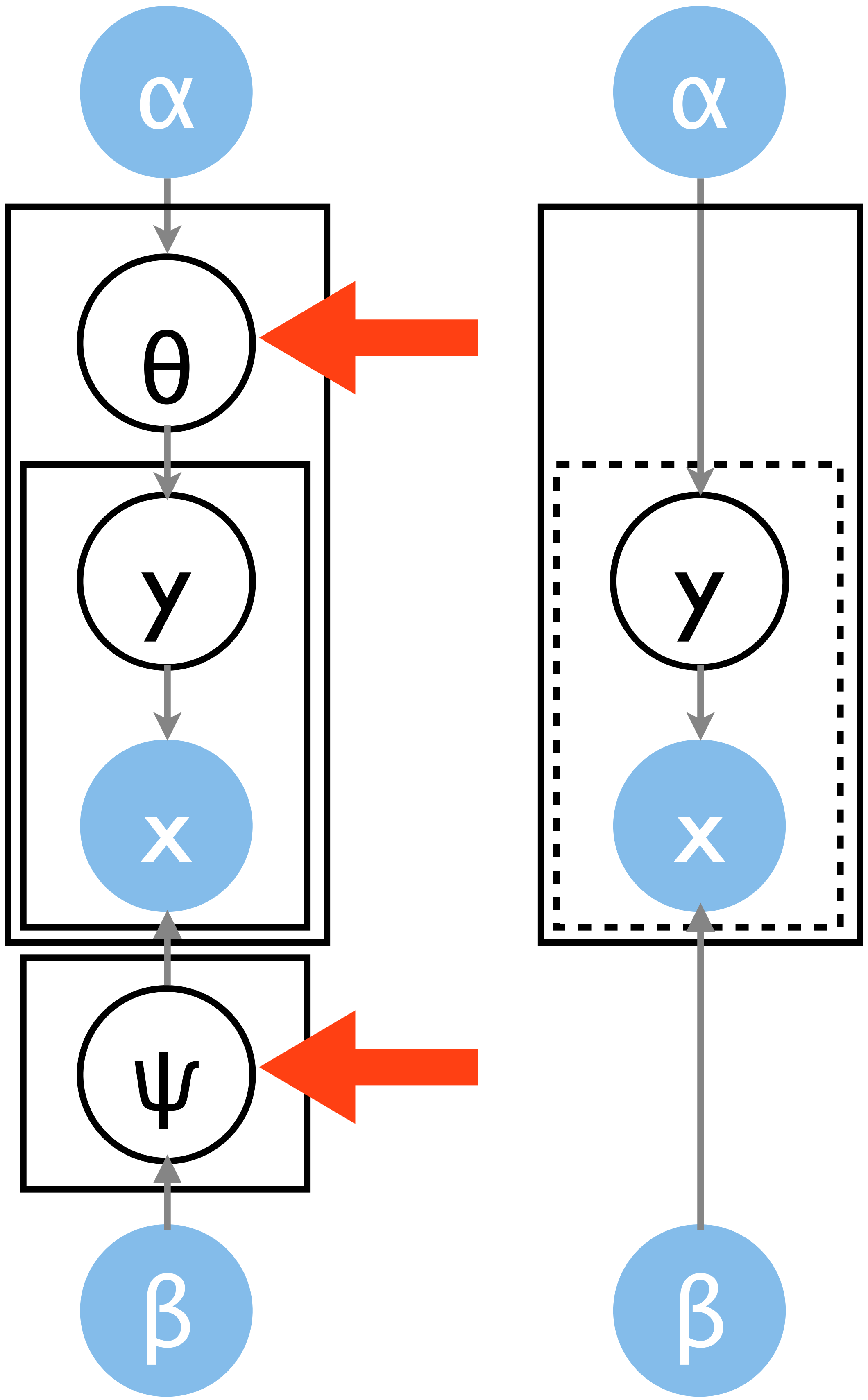
clustering: (0, 1) matrix
topic model: stochastic matrix
LSI: arbitrary matrices

Clustering & Topic Models



clustering: (0, 1) matrix
topic model: stochastic matrix
LSI: arbitrary matrices

V1 - Brute force maximization



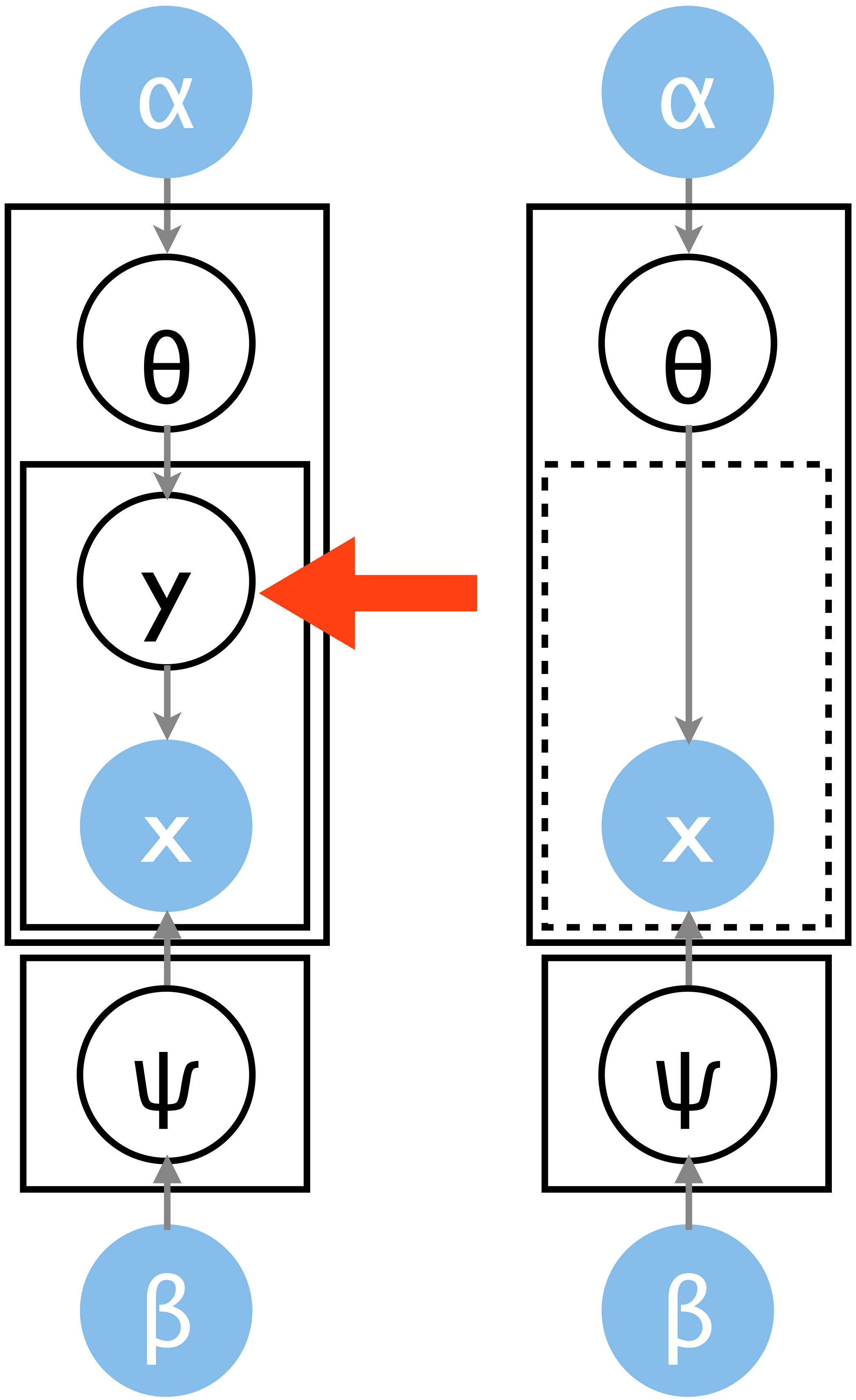
Hal Daume; Joey Gonzalez

- Integrate out latent parameters θ and ψ

$$p(X, Y | \alpha, \beta)$$

- Discrete maximization problem in Y

- Hard to implement
- Overfits a lot (mode is not a typical sample)
- Parallelization infeasible



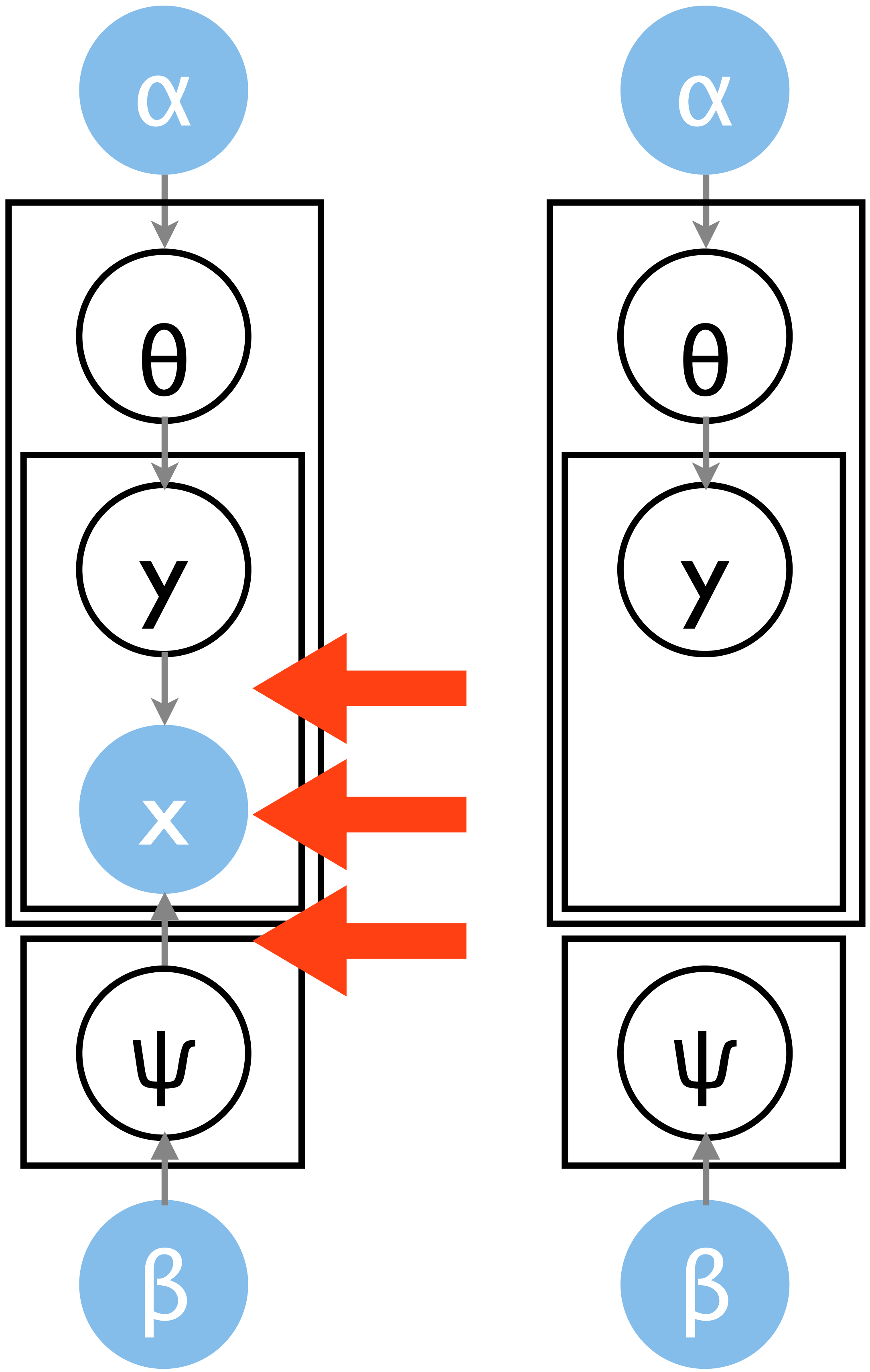
Hoffmann, Blei, Bach (in VW)

V2 - Brute force maximization

- Integrate out latent parameters y

$$p(X, \psi, \theta | \alpha, \beta)$$

- Continuous nonconvex optimization problem in θ and ψ
- Solve by stochastic gradient descent over documents
- Easy to implement
- Does not overfit much
- Great for small datasets
- Parallelization difficult/impossible
- Memory storage/access is O(T W) (this breaks for large models)
  - 1M words, 1000 topics = 4GB
  - Per document 1MFlops/iteration



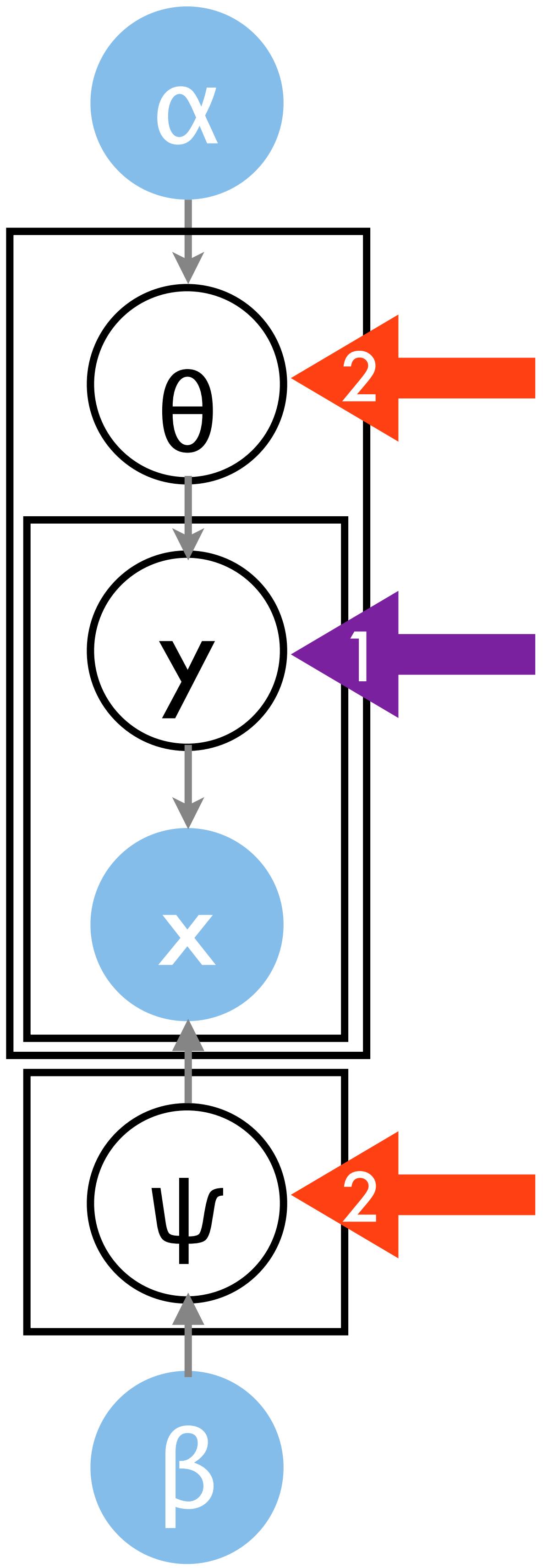
Blei, Ng, Jordan

V3 - Variational approximation

- Approximate intractable joint distribution by tractable factors

$$\begin{aligned}\log p(x) &\geq \log p(x) - D(q(y) \| p(y|x)) \\ &= \int dq(y) \left[\log p(x) + \log p(y|x) - q(y)\right] \\ &= \int dq(y) \log p(x, y) + H[q]\end{aligned}$$

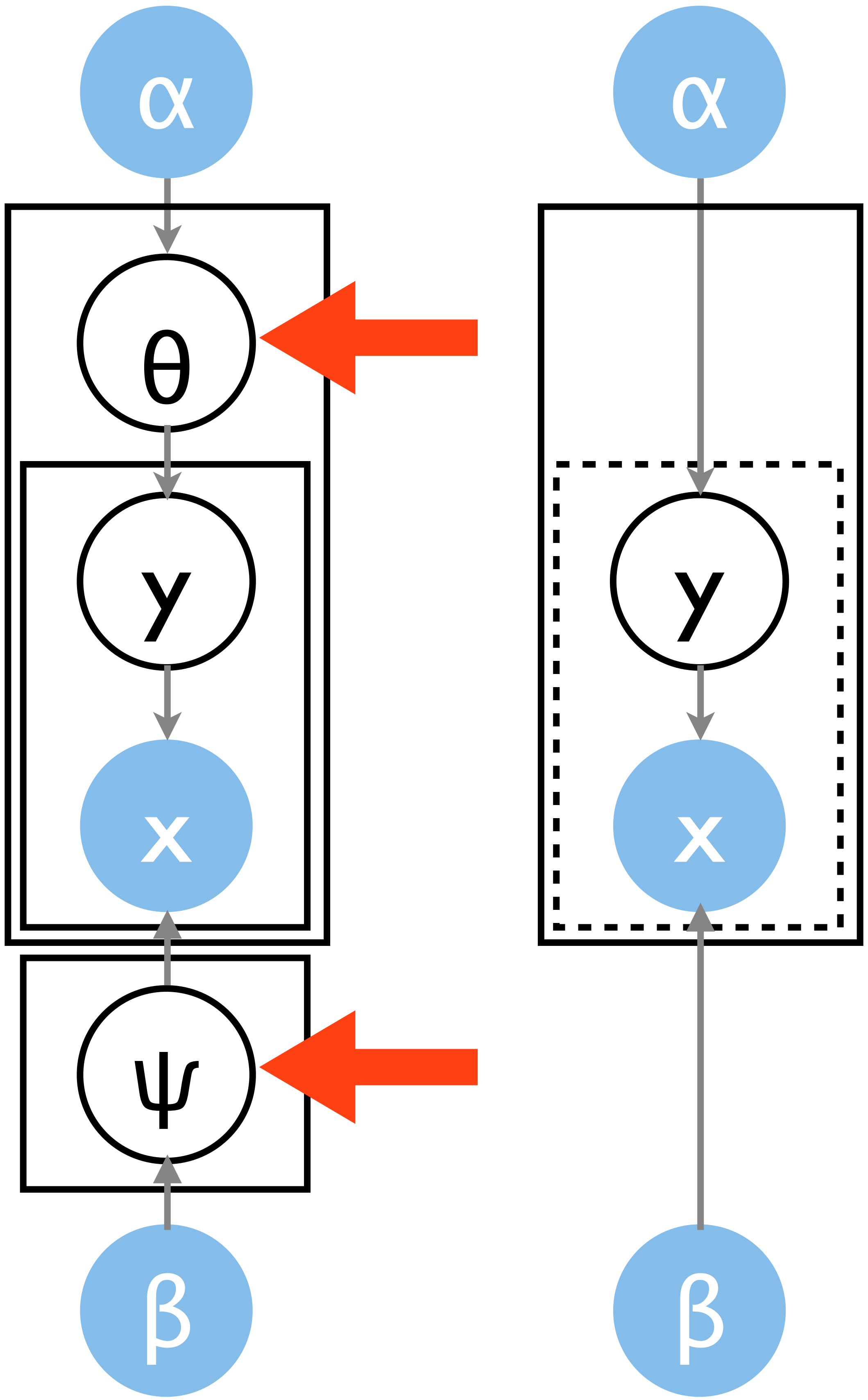
- Alternating convex optimization problem
- Dominant cost is matrix matrix multiply
- Easy to implement
- Great for small topics/vocabulary
- Parallelization easy (aggregate statistics)
- Memory storage is O(T W) (this breaks for large models)
- Model not quite as good as sampling



V4 - Uncollapsed Sampling

- Sample y_{ij}|rest
  Can be done in parallel
- Sample θ|rest and ψ|rest
  Can be done in parallel
- Compatible with MapReduce (only aggregate statistics)
- Easy to implement
- Children can be conditionally independent*
- Memory storage is O(T W) (this breaks for large models)
- Mixes slowly

*for the right model



Griffiths & Steyvers 2005

V5 - Collapsed Sampling

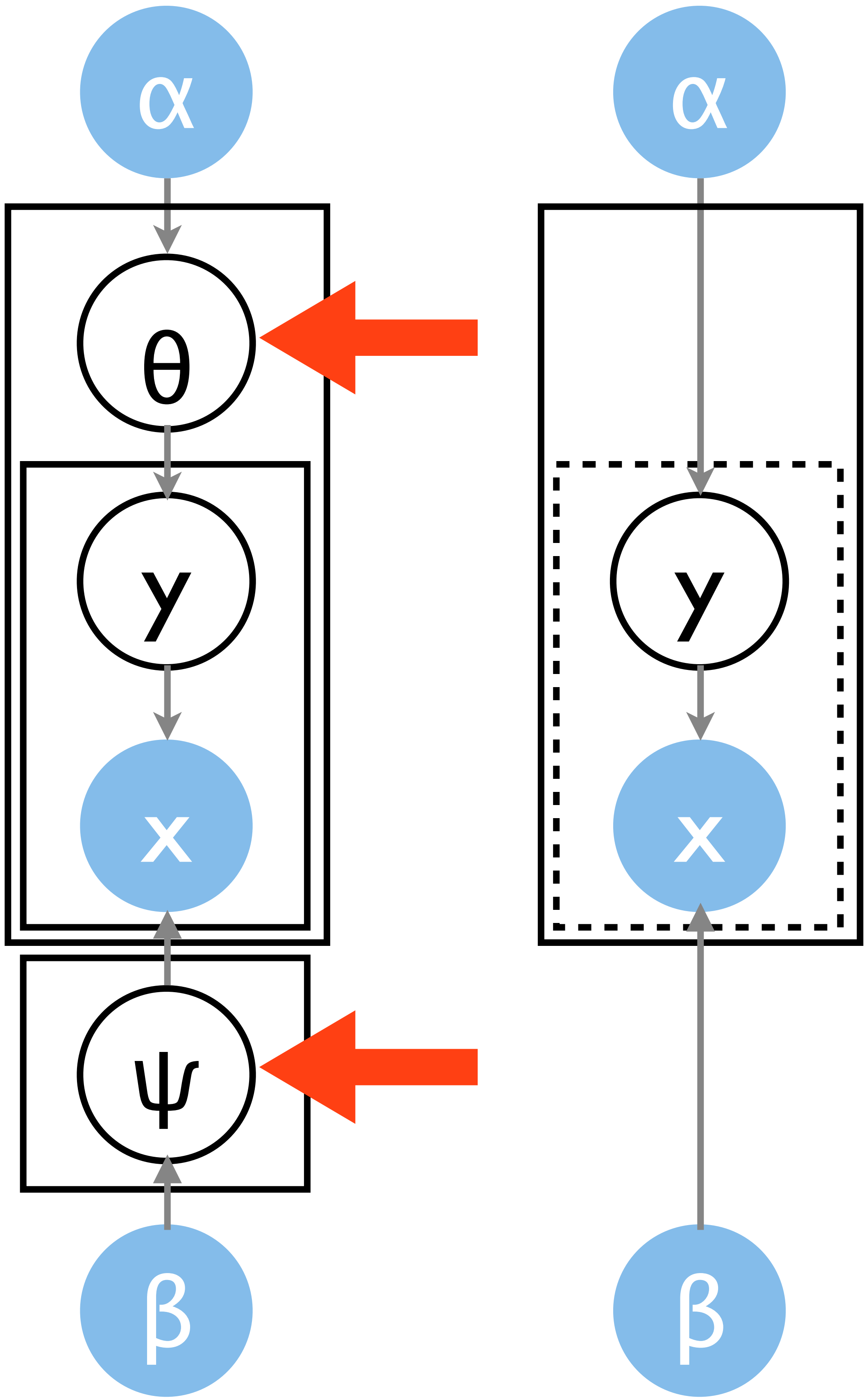
- Integrate out latent parameters θ and ψ

$$p(X, Y|\alpha, \beta)$$

- Sample one topic assignment $y_{ij}|X,Y^{-ij}$ at a time from

$$\frac{n^{-ij}(t,d) + \alpha_t}{n^{-i}(d) + \sum_t \alpha_t} \qquad \frac{n^{-ij}(t,w) + \beta_t}{n^{-i}(t) + \sum_t \beta_t}$$

- Fast mixing
- Easy to implement
- Memory efficient
- Parallelization infeasible (variables lock each other)



Griffiths & Steyvers 2005

V5 - Collapsed Sampling

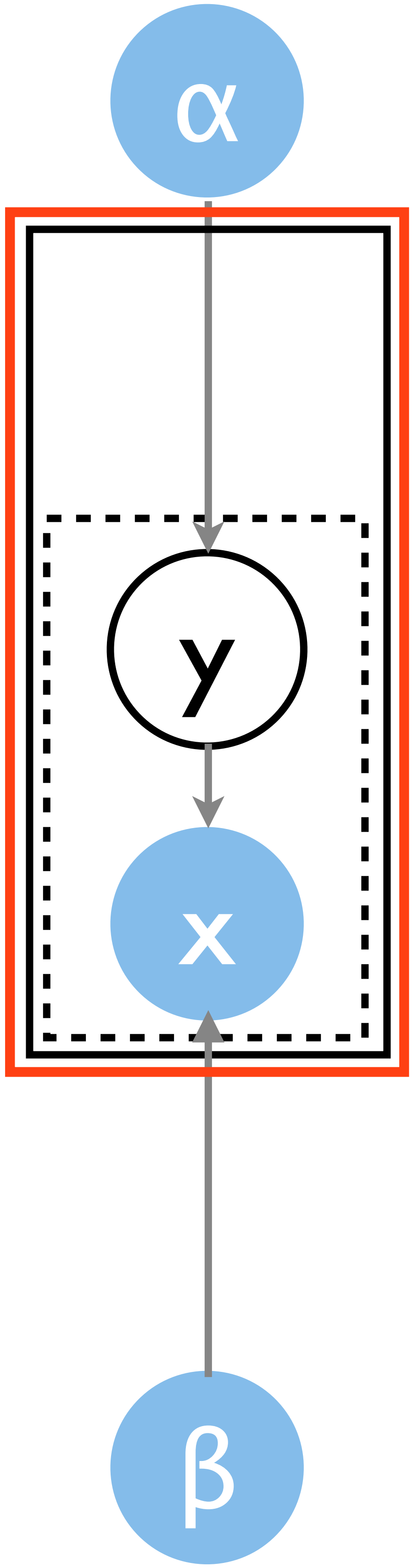
- Integrate out latent parameters θ and ψ

$$p(X, Y|\alpha, \beta)$$

- Sample one topic assignment $y_{ij}|X, Y^{-ij}$ at a time from

$$\frac{n^{-ij}(t,d) + \alpha_t}{n^{-i}(d) + \sum_t \alpha_t} \quad \frac{n^{-ij}(t,w) + \beta_t}{n^{-i}(t) + \sum_t \beta_t}$$

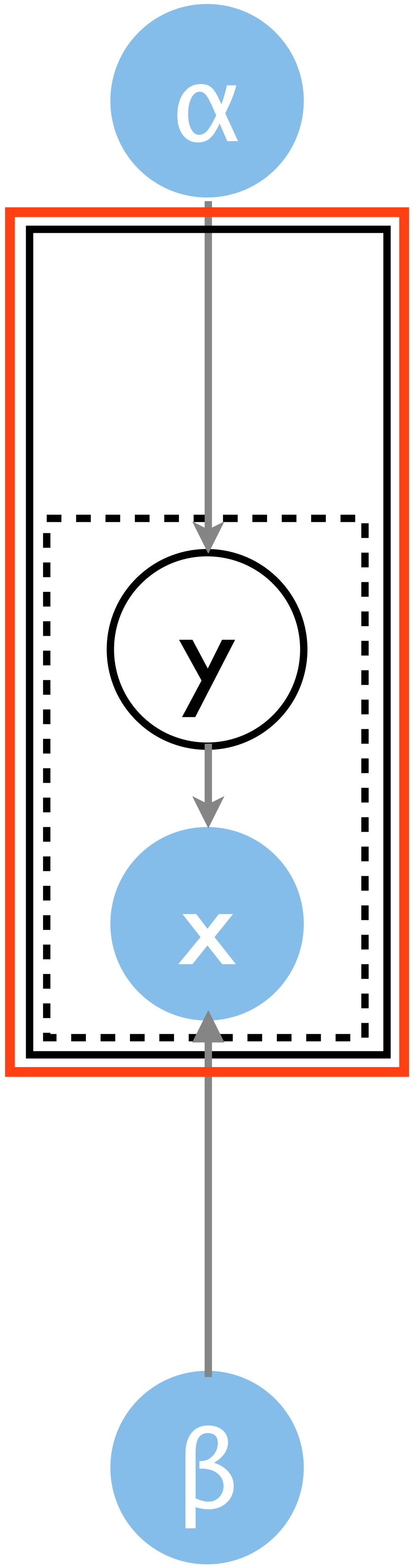
- Fast mixing
- Easy to implement
- Memory efficient
- Parallelization infeasible (variables lock each other)



Asuncion, Smyth, Welling, ... UCI
Mimno, McCallum, ... UMass

V6 - Approximating the Distribution

- Collapsed sampler per machine
$$\frac{n^{-ij}(t,d)+\alpha_t}{n^{-i}(d)+\sum_t \alpha_t} \quad \frac{n^{-ij}(t,w)+\beta_t}{n^{-i}(t)+\sum_t \beta_t}$$
- Defer synchronization between machines
  - no problem for n(t)
  - big problem for n(t,w)
- Easy to implement
- Can be memory efficient
- Easy parallelization
- Mixes slowly/worse likelihood



S. and Narayanamurthy, 2009
Ahmed, Gonzalez, et al., 2012

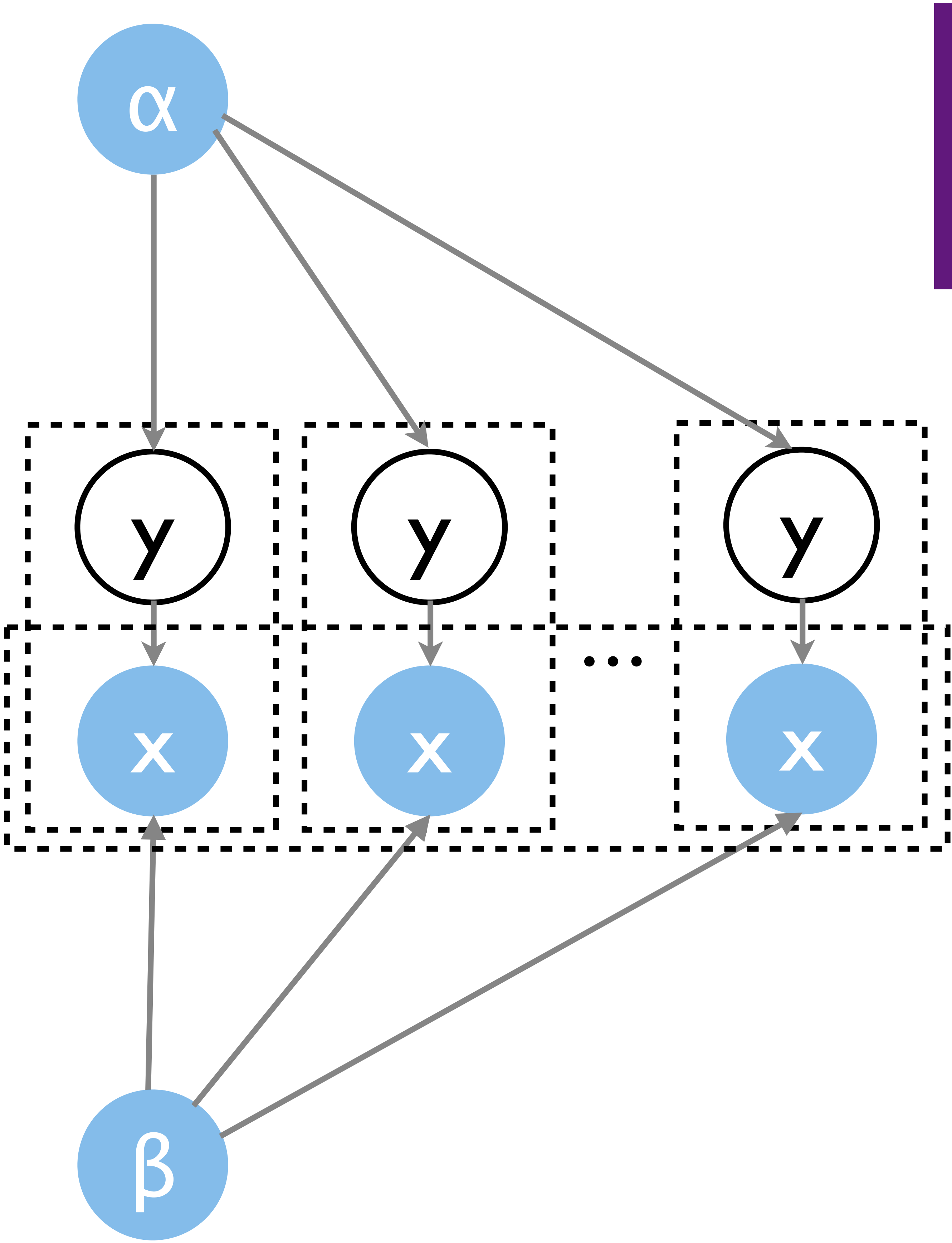
V7 - Better Approximations of the Distribution

- Collapsed sampler

$$\frac{n^{-ij}(t,d)+\alpha_t}{n^{-i}(d)+\sum_t \alpha_t} \quad \frac{n^{-ij}(t,w)+\beta_t}{n^{-i}(t)+\sum_t \beta_t}$$

- Make local copies of state
  - Implicit for multicore (delayed updates from samplers)
  - Explicit copies for multi-machine
- Not a hierarchical model (Welling, Asuncion, et al. 2008)
- Memory efficient (only need to view its own sufficient statistics)
- Multicore / Multi-machine
- Convergence speed depends on synchronizer quality

V8 - Sequential Monte Carlo



Canini, Shi, Griffiths, 2009
Ahmed et al., 2011

- Integrate out latent θ and ψ

$$p(X, Y|\alpha, \beta)$$

- Chain conditional probabilities

$$p(X, Y|\alpha, \beta) = \prod_{i=1}^{m} p(x_i, y_i|x_1, y_1, \ldots x_{i-1}, y_{i-1}, \alpha, \beta)$$

- For each particle sample

$$y_i \sim p(y_i|x_i, x_1, y_1, \ldots x_{i-1}, y_{i-1}, \alpha, \beta)$$

- Reweight particle by next step data likelihood

$$p(x_{i+1}|x_1, y_1, \ldots x_i, y_i, \alpha, \beta)$$

- Resample particles if weight distribution is too uneven

V8 - Sequential Monte Carlo

- One pass through data
- Data sequential parallelization is open problem
- Nontrivial to implement
  - Sampler is easy
  - Inheritance tree through particles is messy
- Need to estimate data likelihood (integration over y), e.g. as part of sampler

- This is multiplicative update algorithm with log loss ...

Canini, Shi, Griffiths, 2009
Ahmed et al., 2011

- Integrate out latent θ and ψ

$$p(X, Y|\alpha, \beta)$$

- Chain conditional probabilities

$$p(X, Y|\alpha, \beta) = \prod_{i=1}^{m} p(x_i, y_i|x_1, y_1, \ldots x_{i-1}, y_{i-1}, \alpha, \beta)$$

- For each particle sample

$$y_i \sim p(y_i|x_i, x_1, y_1, \ldots x_{i-1}, y_{i-1}, \alpha, \beta)$$

- Reweight particle by next step data likelihood

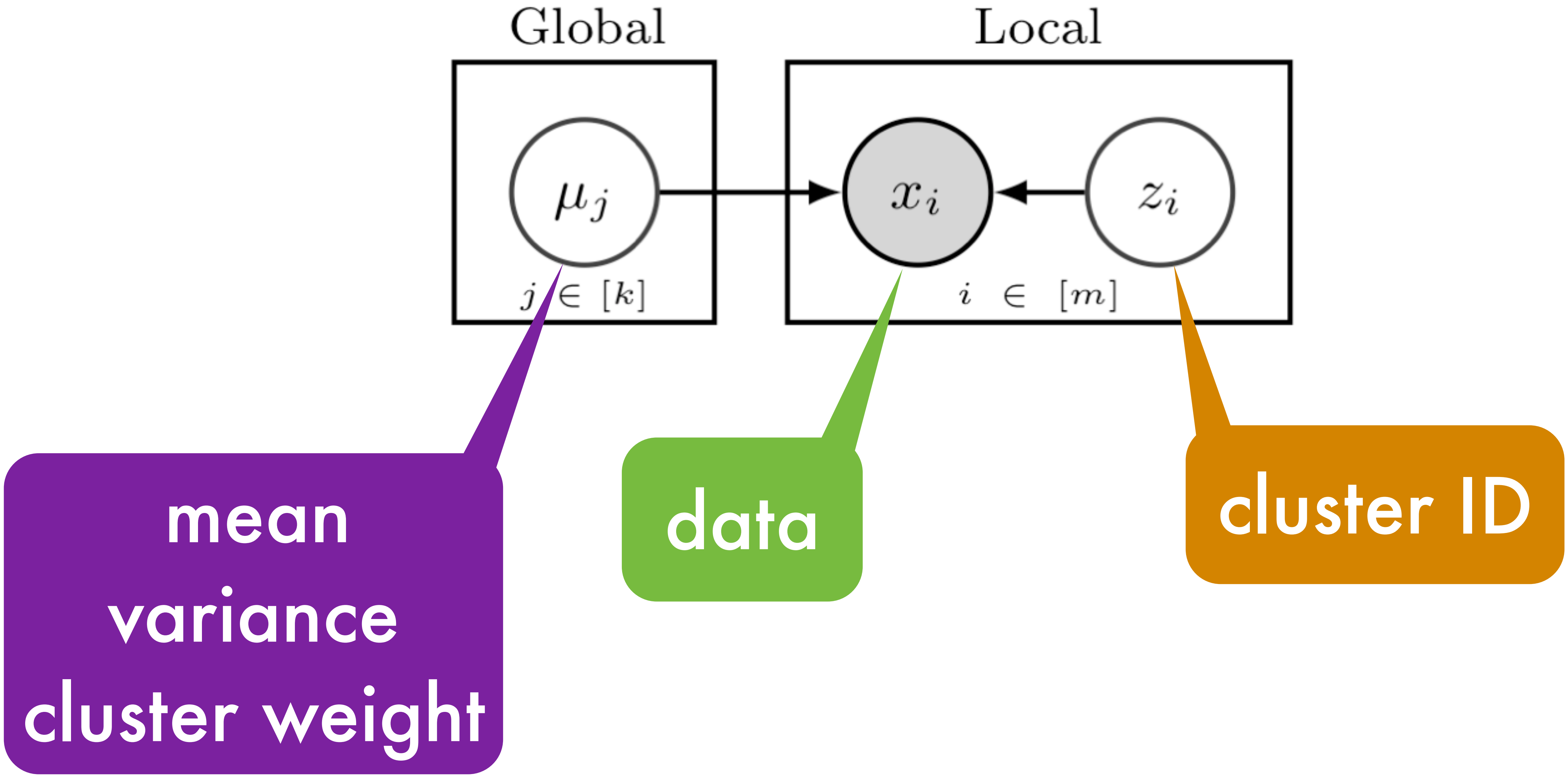
$$p(x_{i+1}|x_1, y_1, \ldots x_i, y_i, \alpha, \beta)$$

- Resample particles if weight distribution is too uneven

	Uncollapsed	Variational approximation	Collapsed natural parameters	Collapsed topic assignments
Optimization	overfits too costly	easy parallelization big memory footprint	overfits too costly	easy to optimize big memory footprint difficult parallelization
Sampling	slow mixing conditionally independent	n.a.	fast mixing difficult parallelization approximate inference by delayed updates particle filtering sequential	sampling difficult

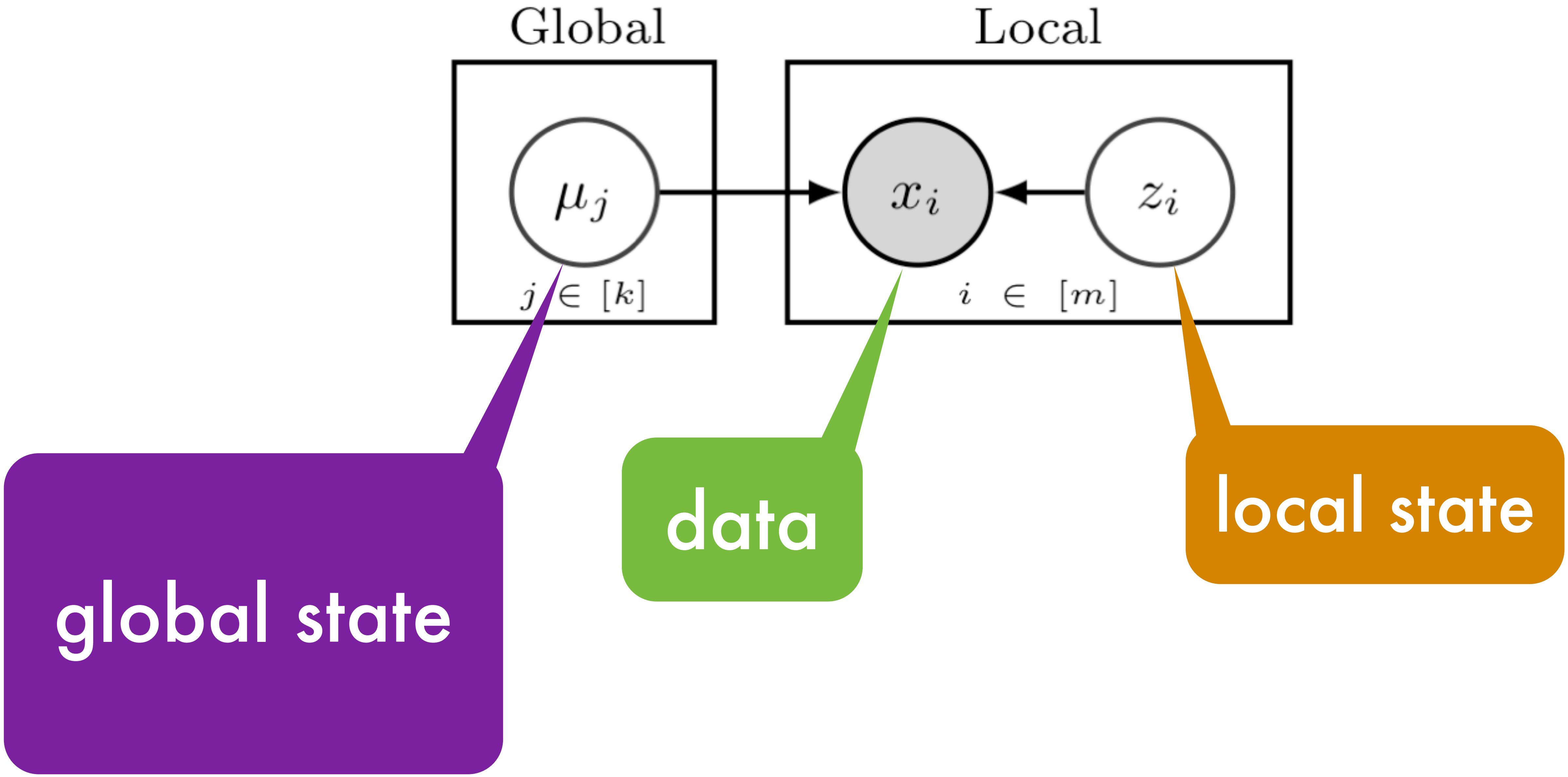

OHIO ART
MAGIC Etch A Sketch® SCREEN
Parallel
Inference

3 Problems

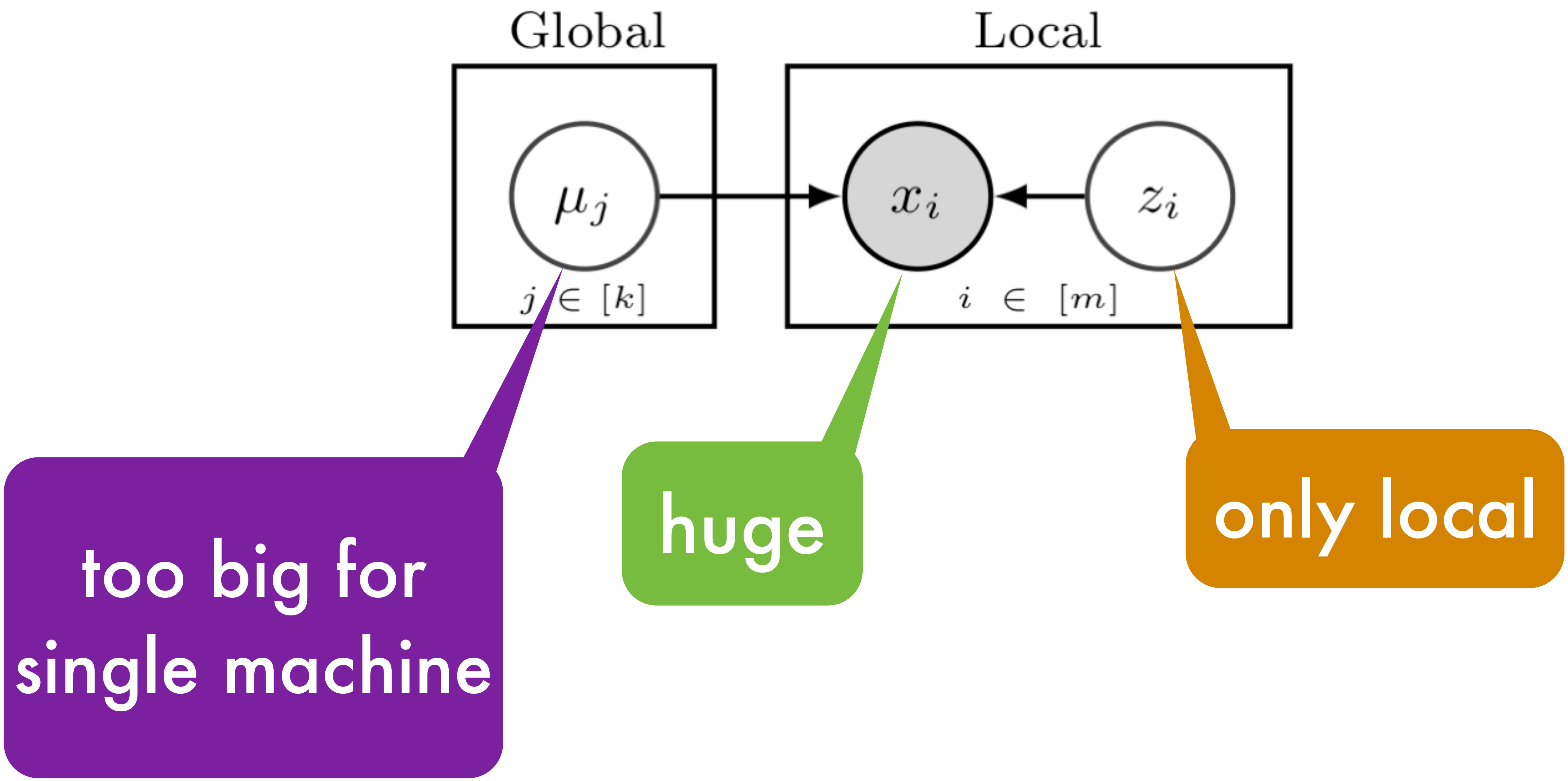


Global
Local
μ_j
x_i
z_i
$j \in [k]$
$i \in [m]$
mean
variance
cluster weight
data
cluster ID

3 Problems

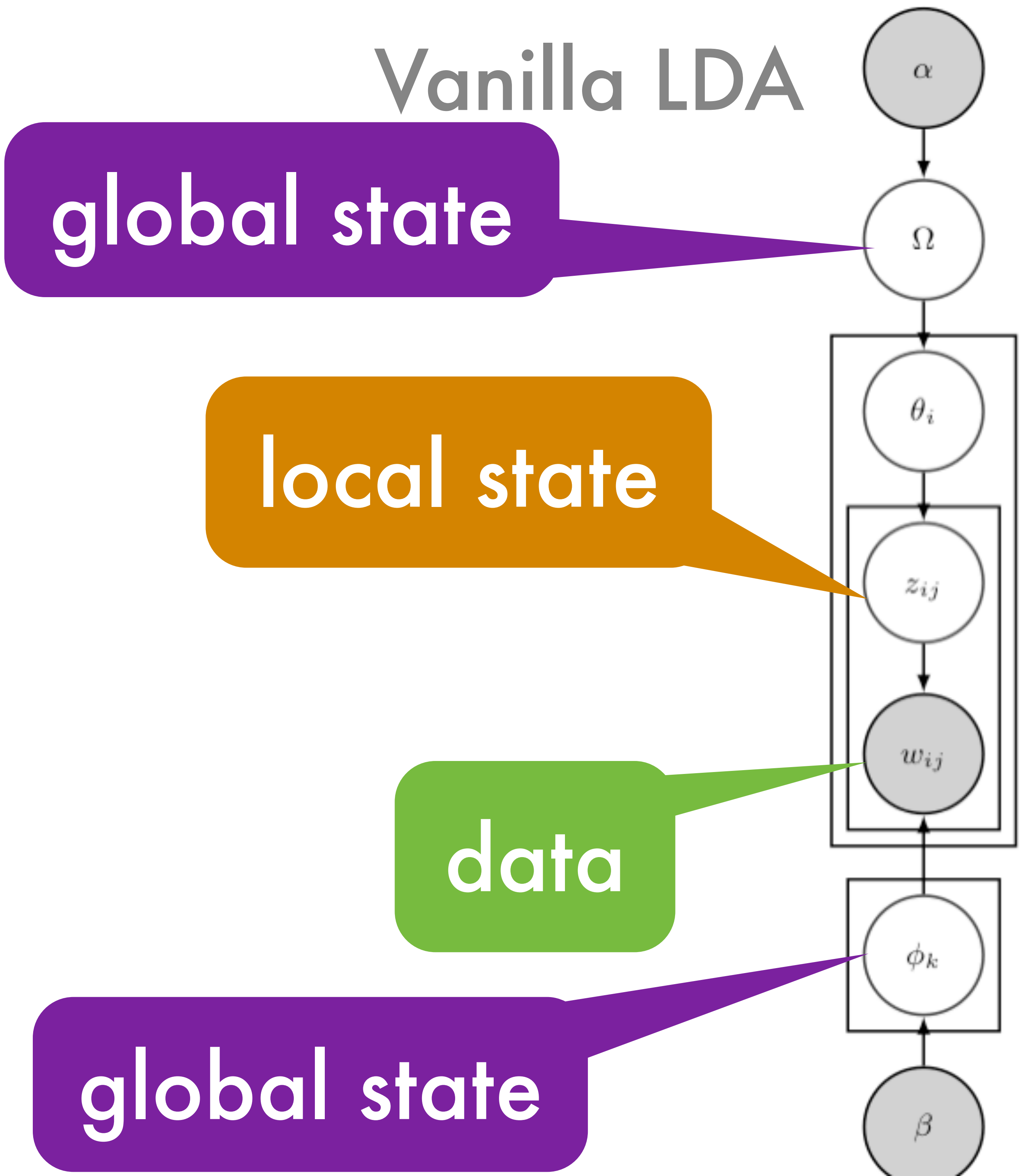

Global
Local
μ_j
x_i
z_i
$j \in [k]$
$i \in [m]$
global state
data
local state

3 Problems

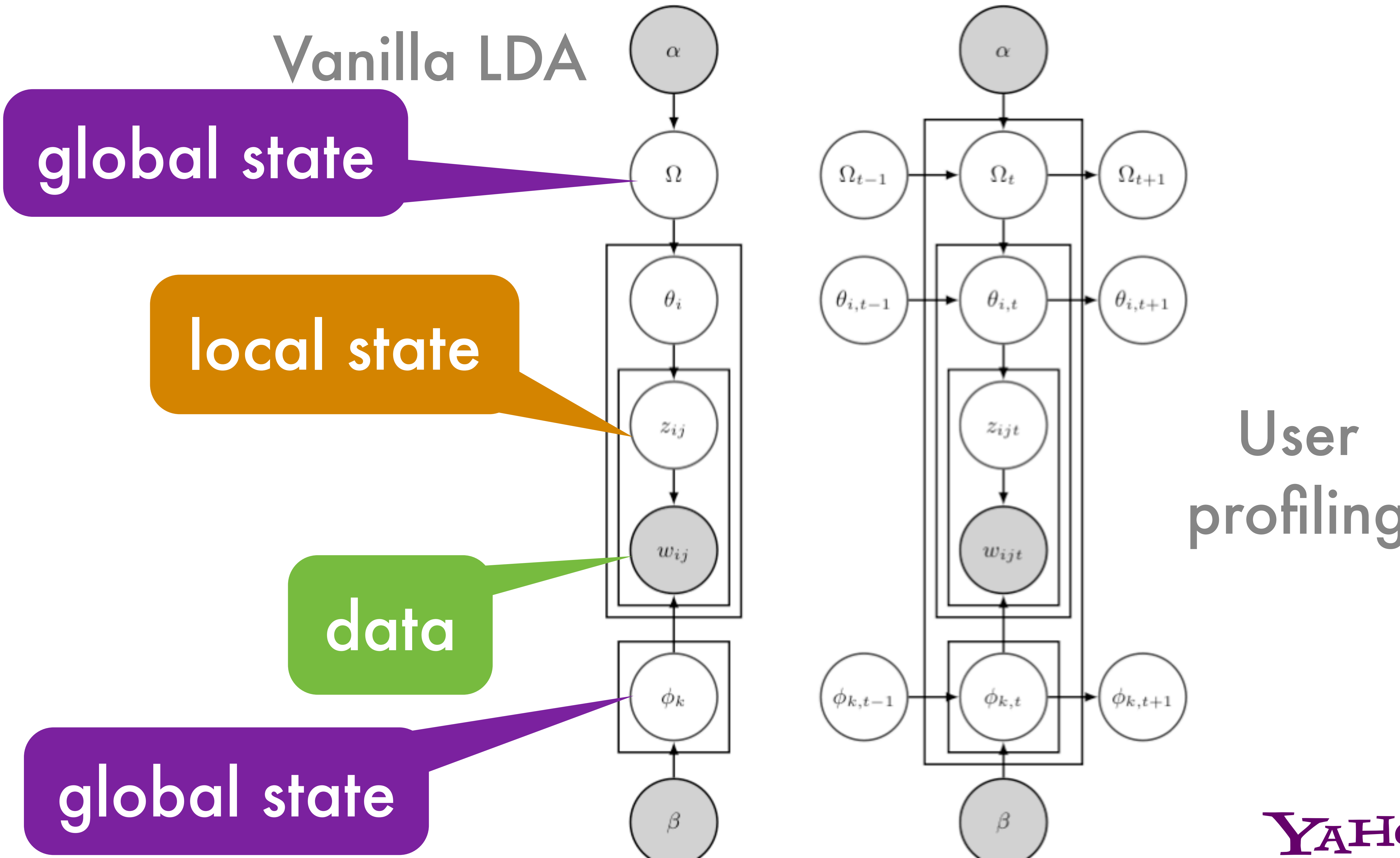


Global
Local
μ_j
x_i
z_i
$j \in [k]$
$i \in [m]$
too big for single machine
huge
only local

3 Problems


Vanilla LDA
global state
local state
data
global state
α
Ω
θ_i
z_{ij}
w_{ij}
ϕ_k
β

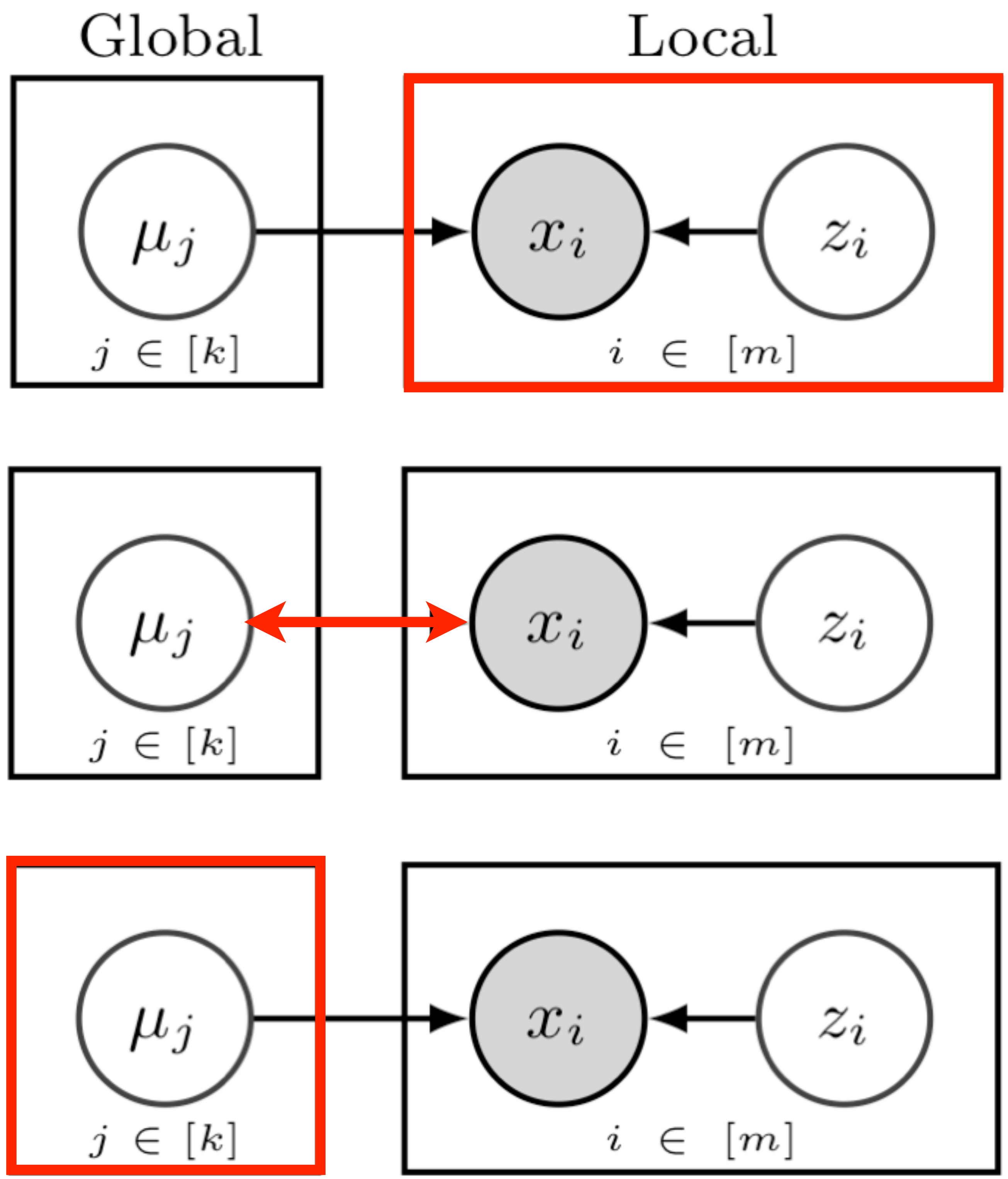
3 Problems


Vanilla LDA
global state
local state
data
global state
User profiling

3 Problems

local state
is too large

does not fit
into memory



network load
& barriers

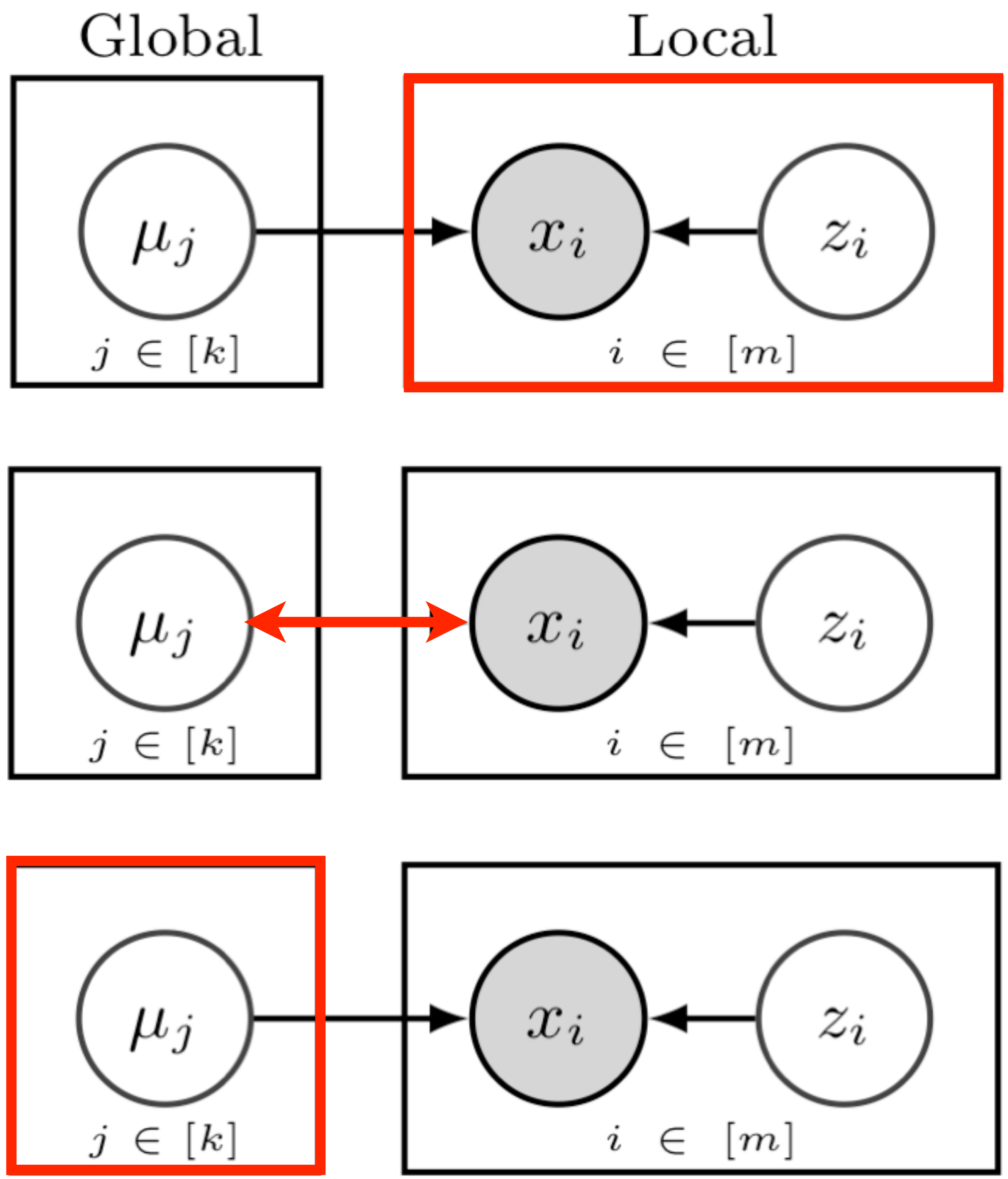
global state
is too large

does not fit
into memory

3 Problems

local state is too large

global state is too large



stream local data from disk

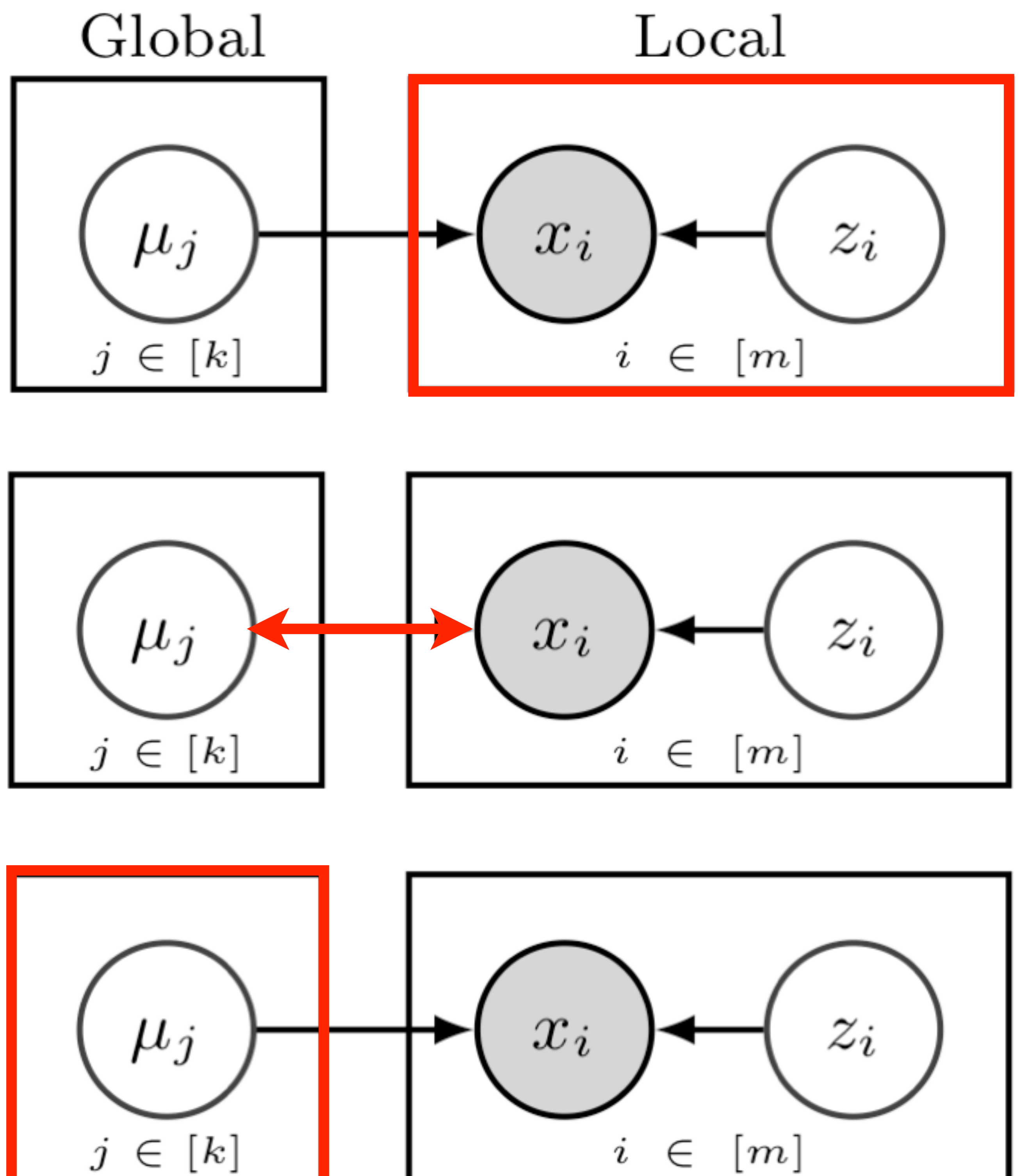
network load & barriers

does not fit into memory

3 Problems

local state is too large

global state is too large



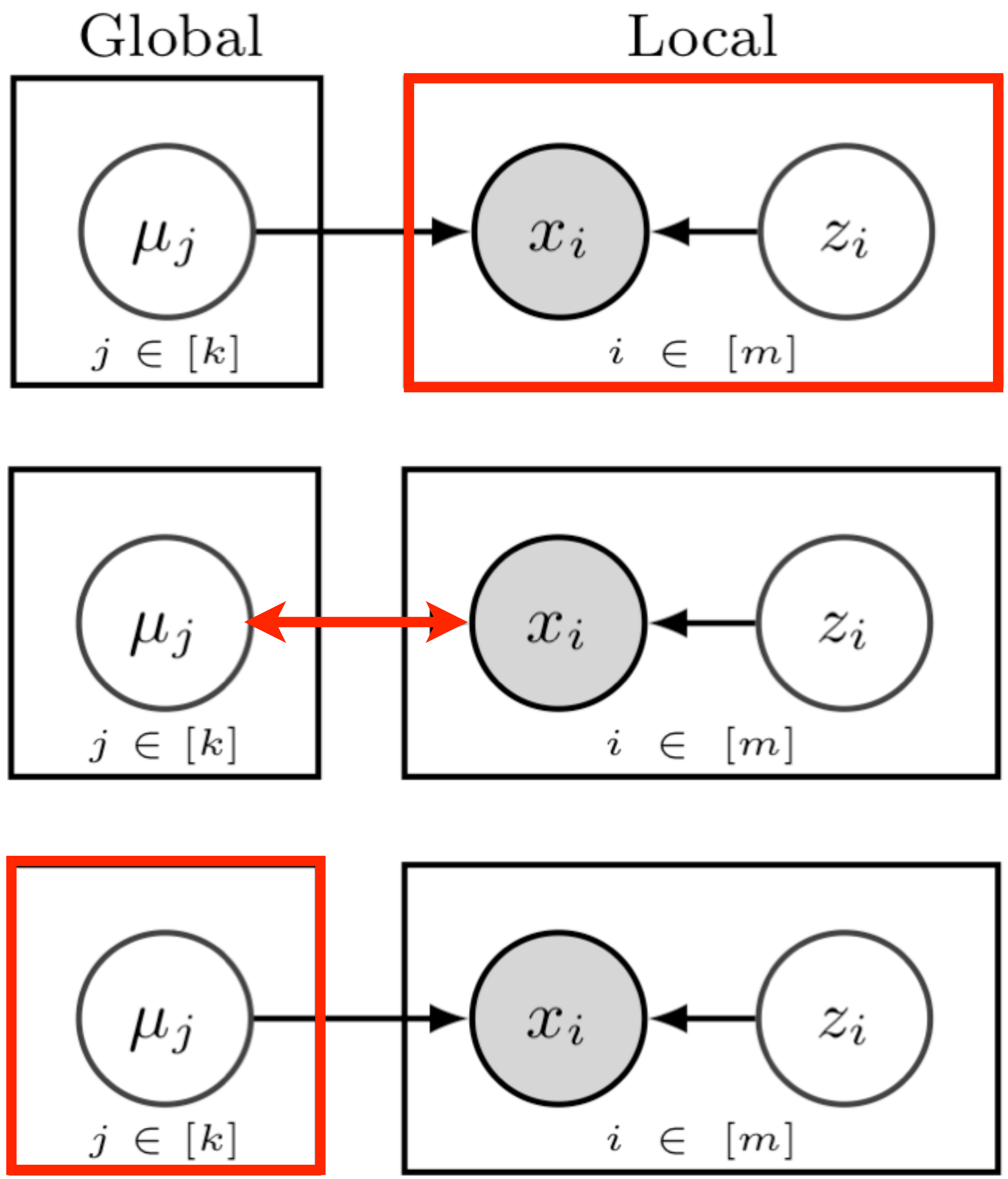
stream local data from disk

asynchronous synchronization

does not fit into memory

YAHOO!

3 Problems



local state is too large

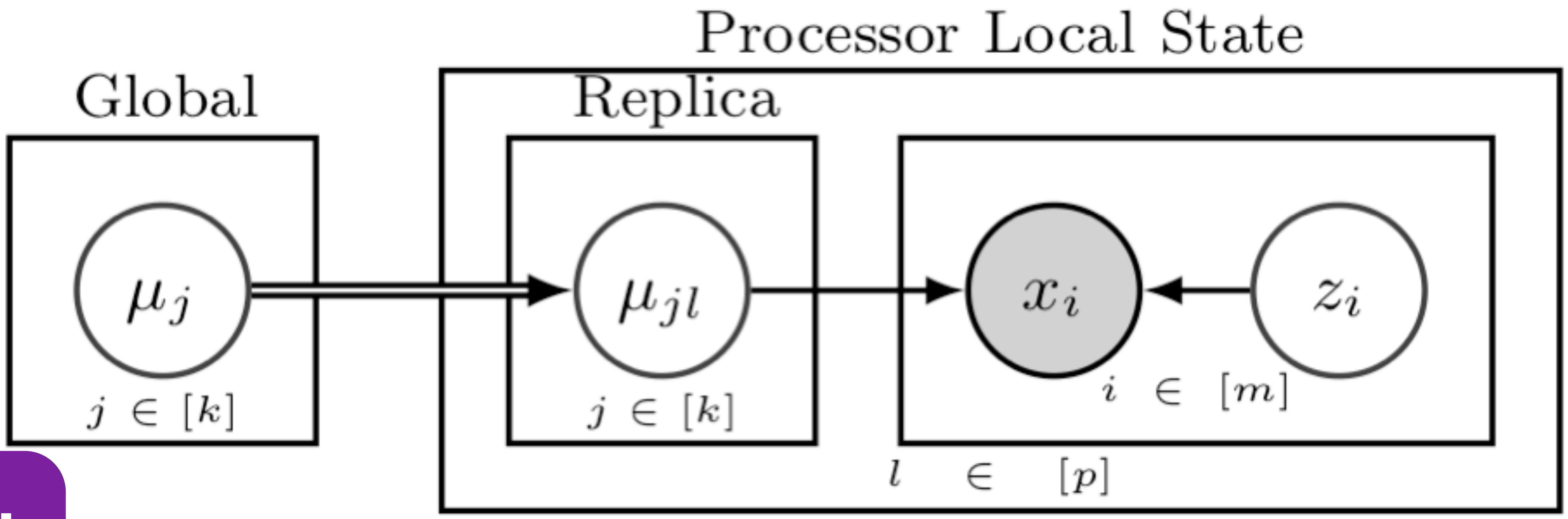
stream local data from disk

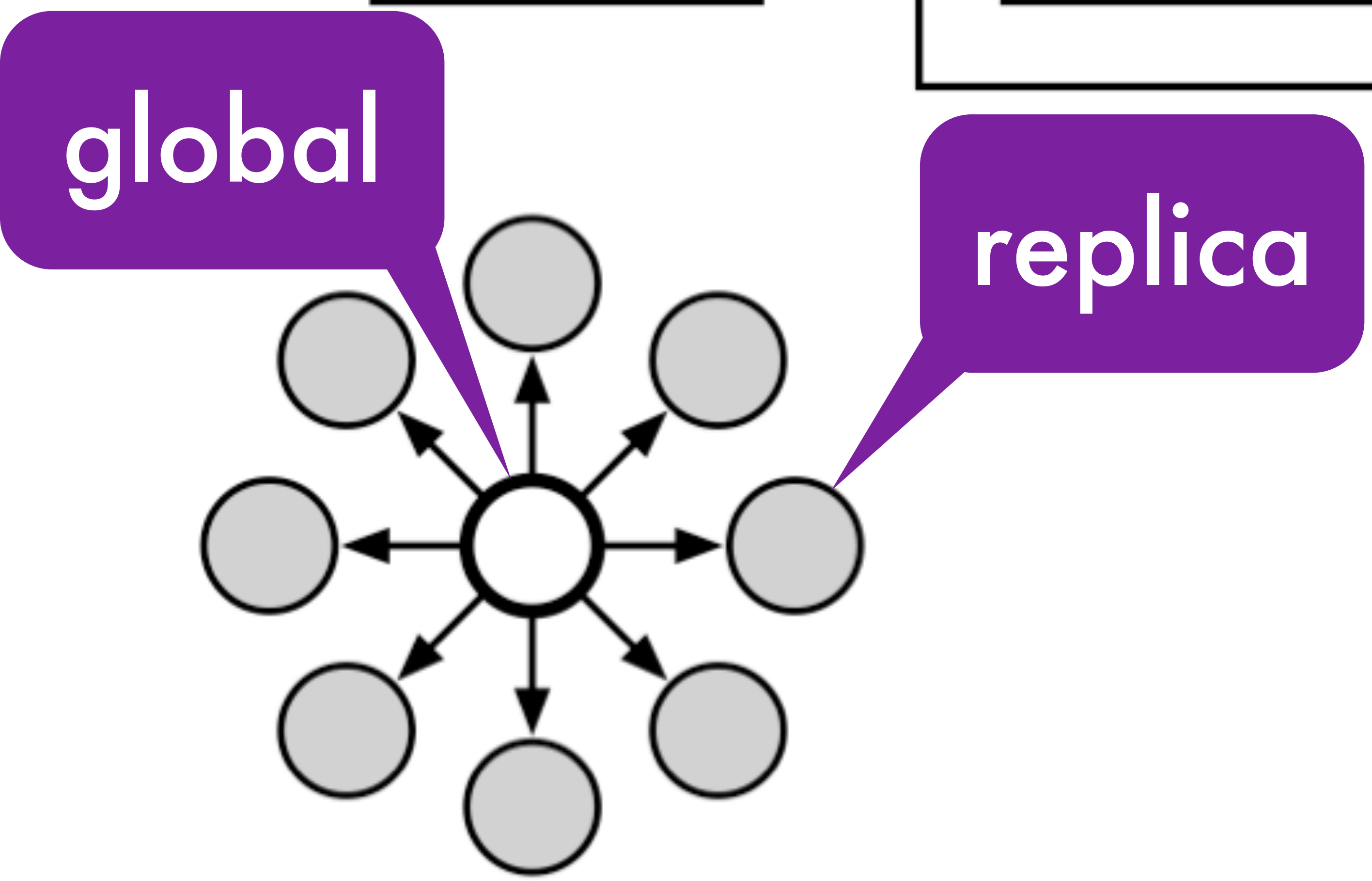
asynchronous synchronization

global state is too large

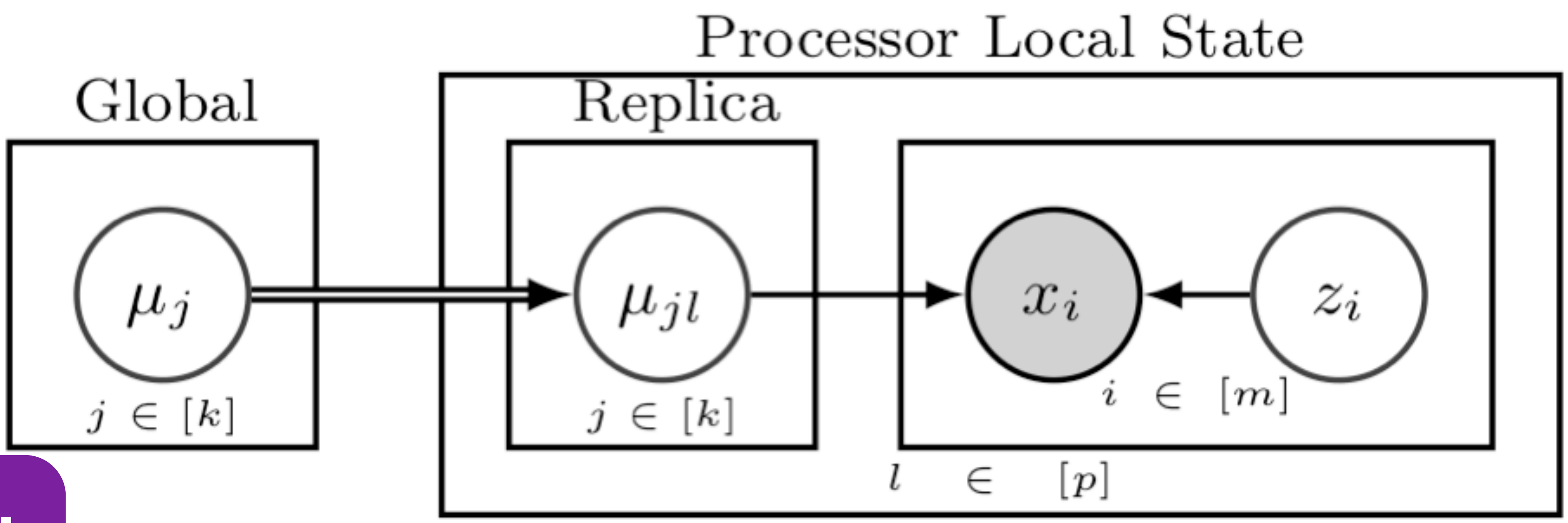
partial view

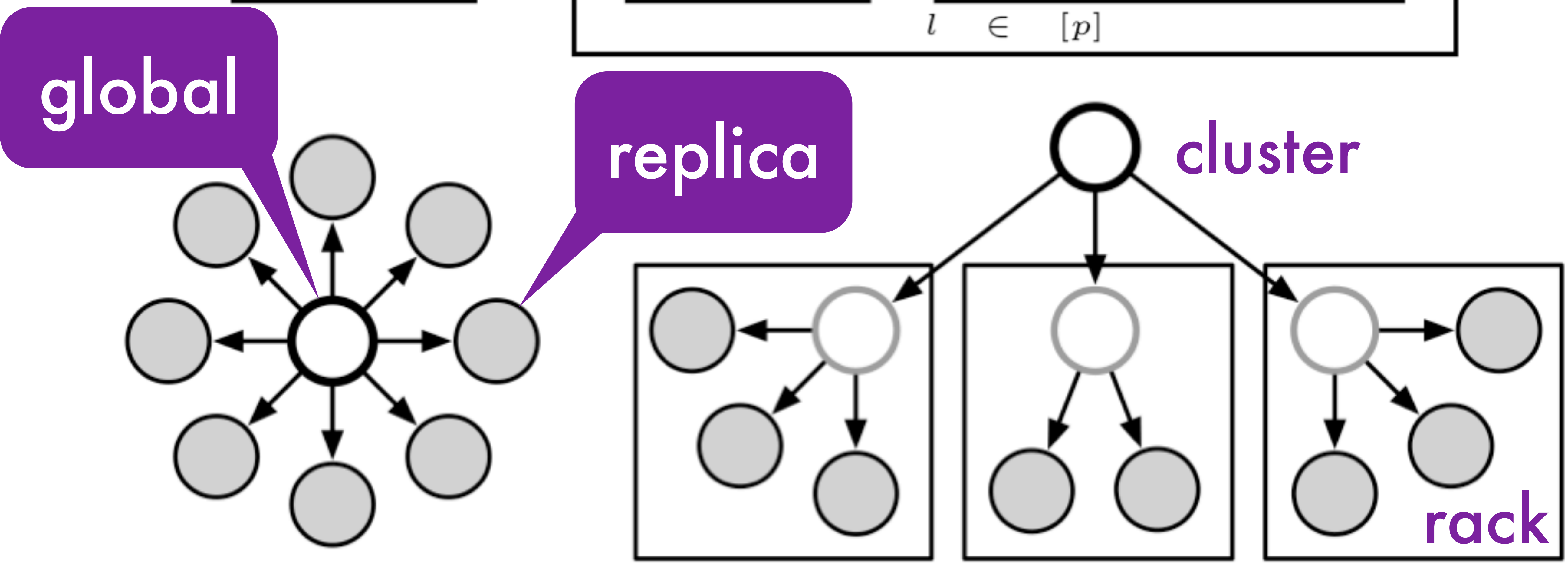
Distribution





Distribution



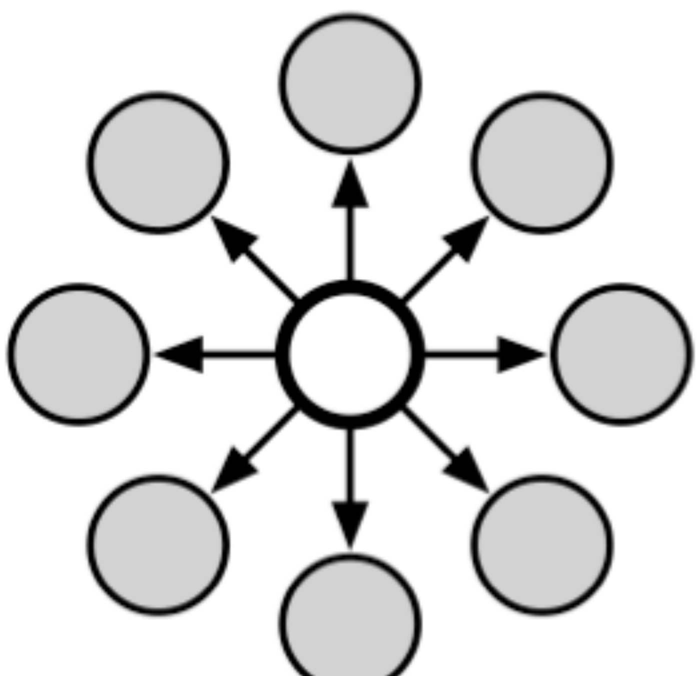


Synchronization

- Child updates local state
  - Start with common state
  - Child stores old and new state
  - Parent keeps global state
- Transmit differences asynchronously
  - Inverse element for difference
  - Abelian group for commutativity (sum, log-sum, cyclic group, exponential families)

local to global

$$\delta \leftarrow x - x^{\text{old}}$$
$$x^{\text{old}} \leftarrow x$$
$$x^{\text{global}} \leftarrow x^{\text{global}} + \delta$$



global to local

$$x \leftarrow x + (x^{\text{global}} - x^{\text{old}})$$
$$x^{\text{old}} \leftarrow x^{\text{global}}$$

Synchronization

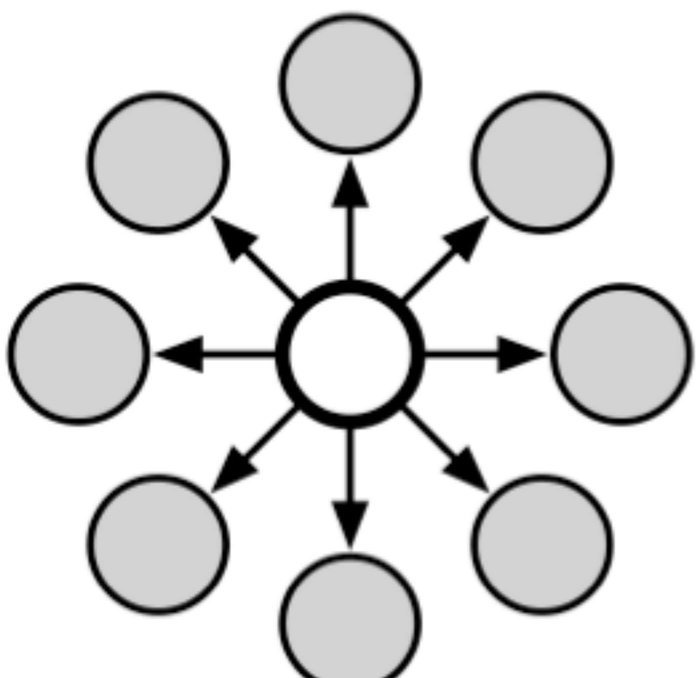
- Naive approach (dumb master)
  - Global is only (key,value) storage
  - Local node needs to lock/read/write/unlock master
  - Needs a 4 TCP/IP roundtrips - latency bound
- Better solution (smart master)
  - Client sends message to master / in queue / master incorporates it
  - Master sends message to client / in queue / client incorporates it
  - Bandwidth bound (>10x speedup in practice)

local to global

$$\delta \leftarrow x - x^{\text{old}}$$

$$x^{\text{old}} \leftarrow x$$

$$x^{\text{global}} \leftarrow x^{\text{global}} + \delta$$



global to local

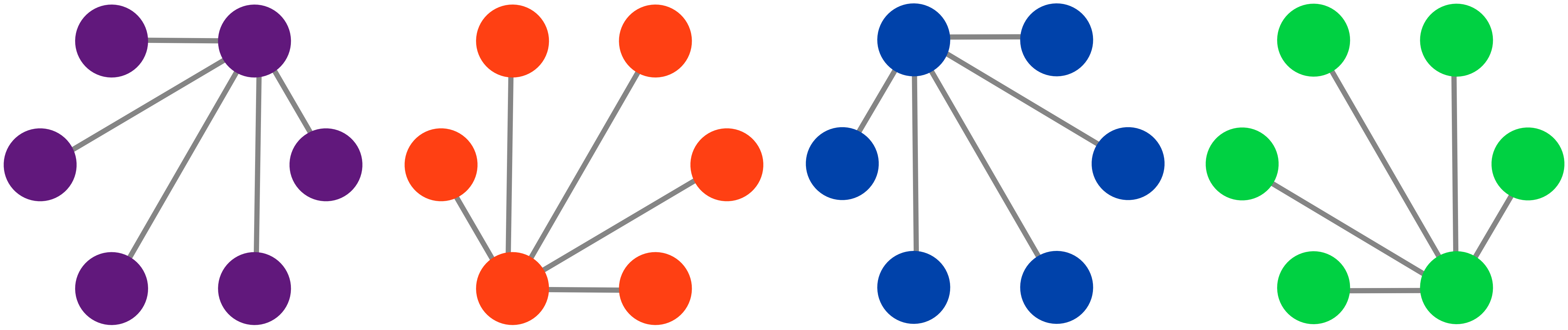
$$x \leftarrow x + (x^{\text{global}} - x^{\text{old}})$$

$$x^{\text{old}} \leftarrow x^{\text{global}}$$

Distribution

- Dedicated server for variables
  - Insufficient bandwidth (hotspots)
  - Insufficient memory
- Select server e.g. via consistent hashing

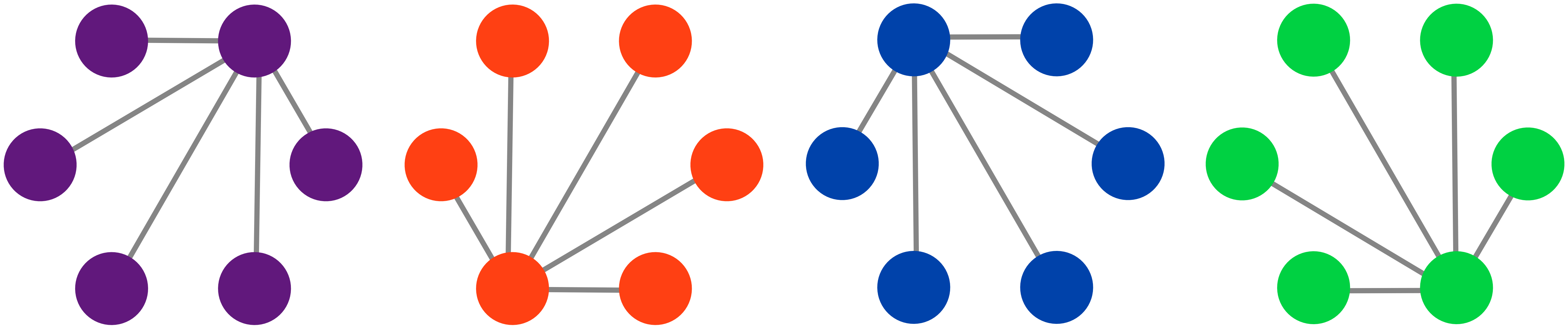
$$m(x) = \operatorname*{argmin}_{m \in M} h(x, m)$$



Distribution & fault tolerance

- Storage is O(1/k) per machine
- Communication is O(1) per machine
- Fast snapshots O(1/k) per machine (stop sync and dump state per vertex)

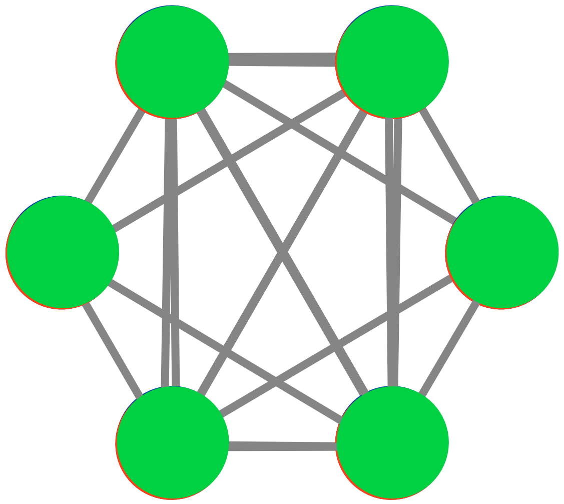
$$m(x) = \underset{m \in M}{\operatorname{argmin}}\, h(x, m)$$



Distribution & fault tolerance

- Storage is O(1/k) per machine
- Communication is O(1) per machine
- Fast snapshots O(1/k) per machine (stop sync and dump state per vertex)

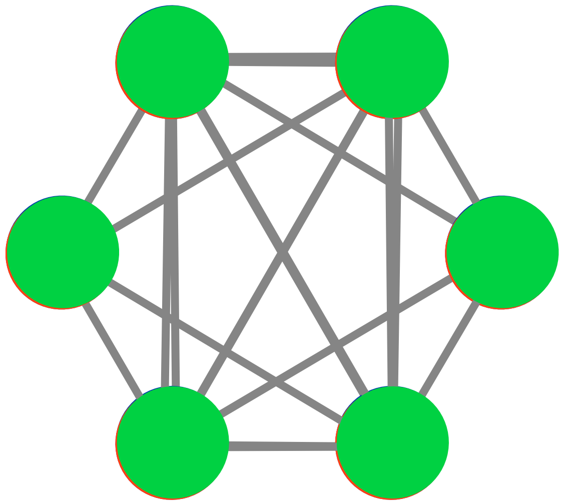
$$m(x) = \operatorname*{argmin}_{m \in M} h(x, m)$$



Distribution & fault tolerance

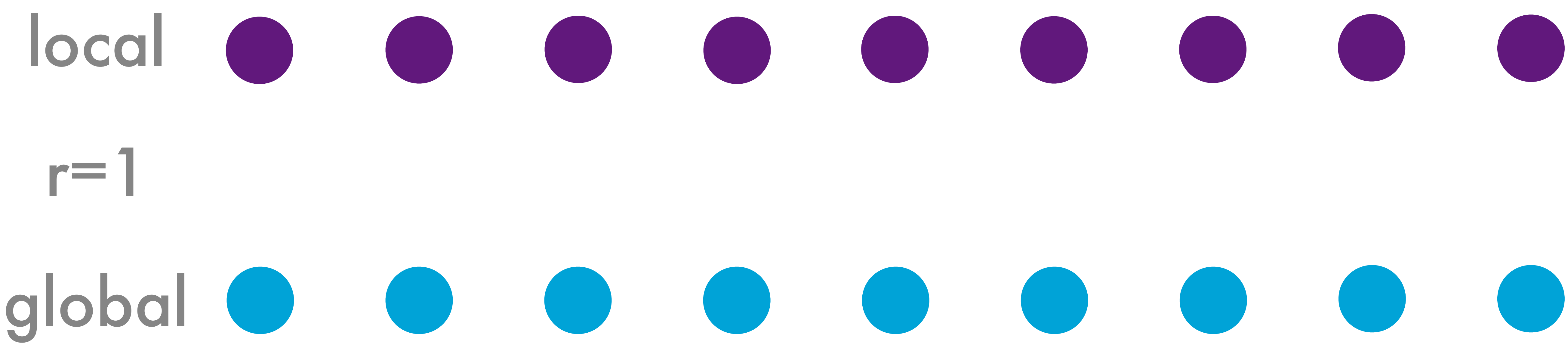
- Storage is O(1/k) per machine
- Communication is O(1) per machine
- Fast snapshots O(1/k) per machine (stop sync and dump state per vertex)
- O(k) open connections per machine
- O(1/k) throughput per machine

$$m(x) = \operatorname*{argmin}_{m \in M} h(x, m)$$



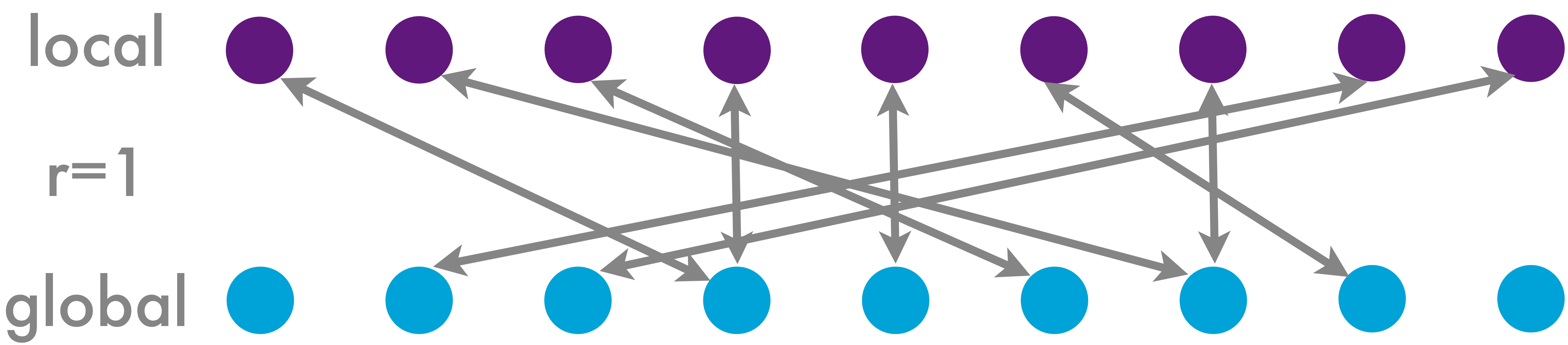
Synchronization

- Data rate between machines is O(1/k)
- Machines operate asynchronously (barrier free)
- Solution
  - Schedule message pairs
  - Communicate with r random machines simultaneously



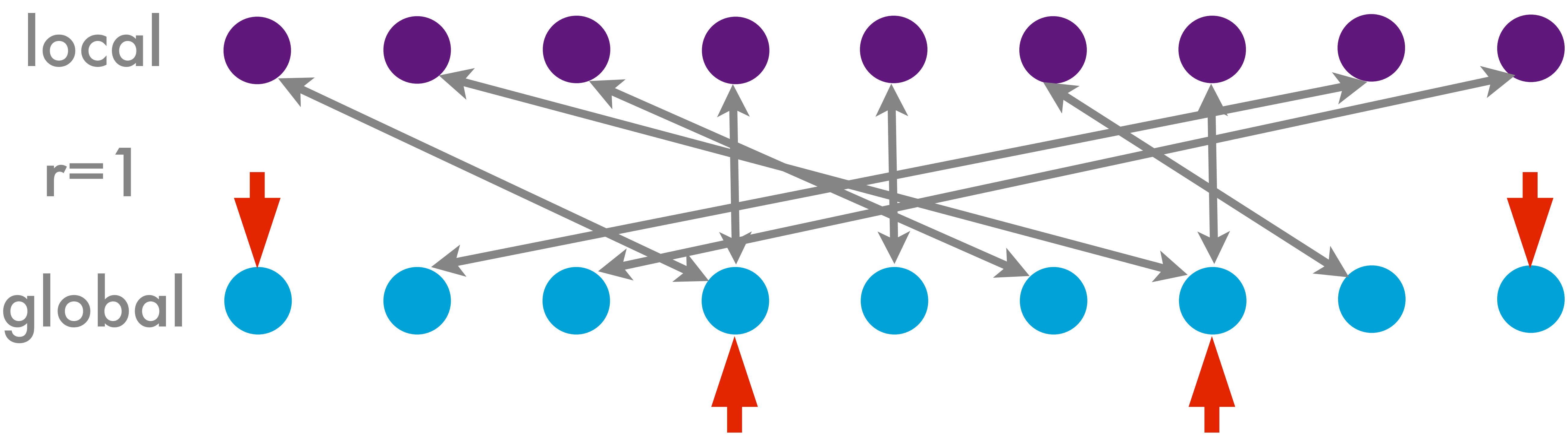
Synchronization

- Data rate between machines is O(1/k)
- Machines operate asynchronously (barrier free)
- Solution
  - Schedule message pairs
  - Communicate with r random machines simultaneously



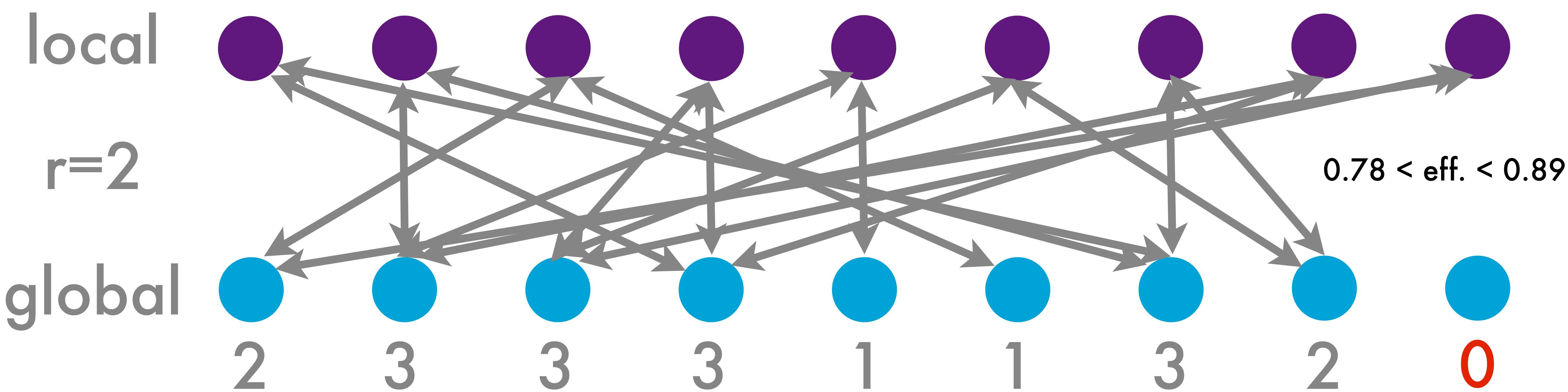
Synchronization

- Data rate between machines is O(1/k)
- Machines operate asynchronously (barrier free)
- Solution
  - Schedule message pairs
  - Communicate with r random machines simultaneously



Synchronization

- Data rate between machines is O(1/k)
- Machines operate asynchronously (barrier free)
- Solution
  - Schedule message pairs
  - Communicate with r random machines simultaneously



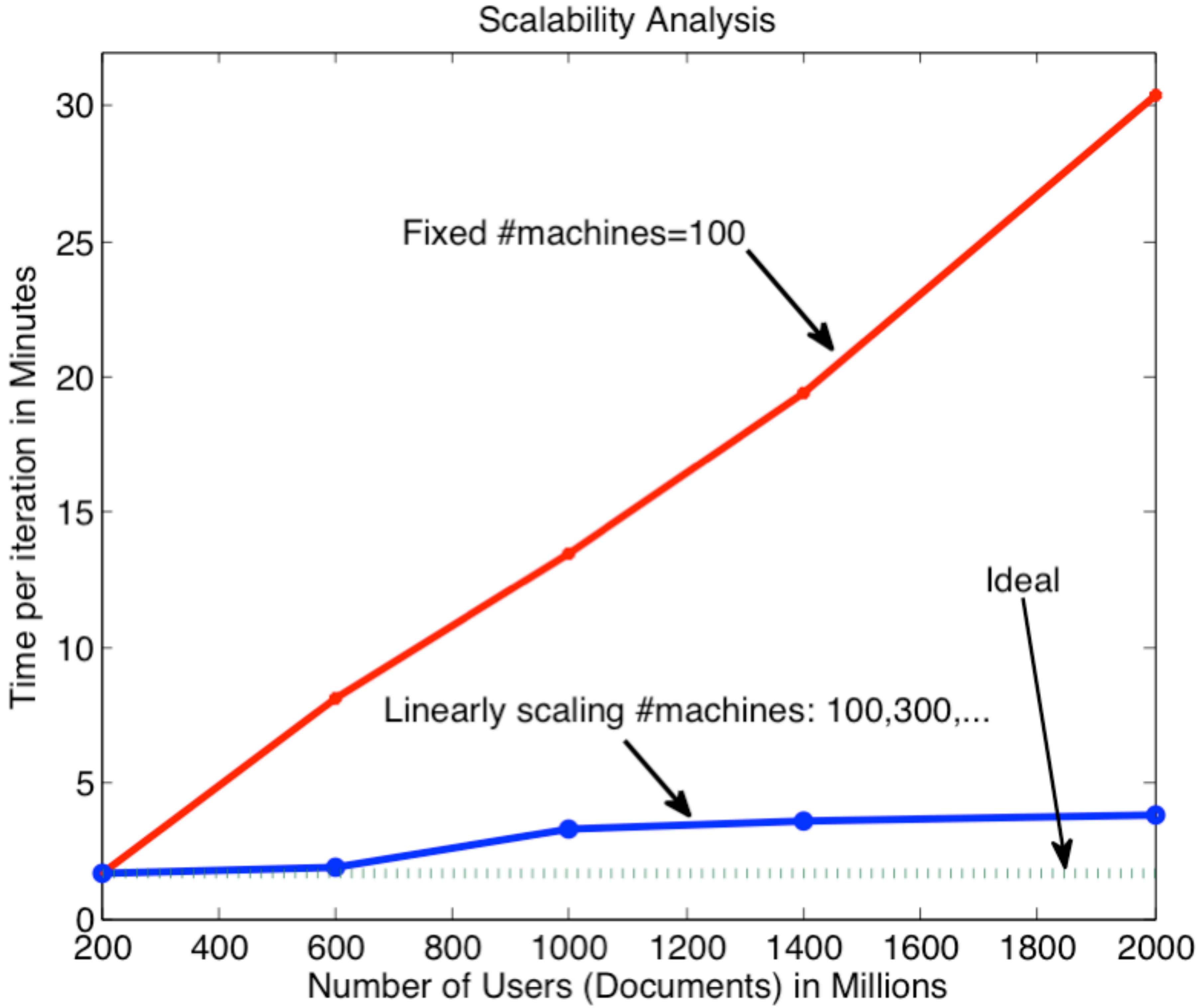
Synchronization

- Data rate between machines is O(1/k)
- Machines operate asynchronously (barrier free)
- Solution
  - Schedule message pairs
  - Communicate with r random machines simultaneously
  - Use Luby-Rackoff PRPG for load balancing
- Efficiency guarantee

$$1 - e^{-r} \sum_{i=0}^{r} \left[1 - \frac{i}{r}\right] \frac{r^i}{i!} \leq \text{Eff} \leq 1 - e^{-r}$$

4 simultaneous connections are sufficient

Scalability




OHIO ART
MAGIC Etch A Sketch SCREEN
Samplers

Sampling

- Brute force sampling over large number of items is expensive
  - Ideally want work to scale with entropy of distribution over labels.
  - Sparsity of distribution typically only known after seeing the instances
- Decompose (dense) probability into dense invariant and sparse variable terms
- Use fast proposal distribution & rejection sampling

Exploiting Sparsity

- Decomposition (Mimno & McCallum, 2009)
  Only need to update **sparse** terms per word

$$p(t|w_{ij}) \propto \beta_w \frac{\alpha_t}{n(t) + \bar{\beta}} + \beta_w \frac{n(t, d = i)}{n(t) + \bar{\beta}} + \frac{n(t, w = w_{ij}) \left[n(t, d = i) + \alpha_t\right]}{n(t) + \bar{\beta}}$$

dense but 'constant'

sparse

- Does not work for clustering (too many factors)

Exploiting Sparsity

- Context LDA (Petterson et al., 2009)
  The smoothers are word and topic dependent

$$p(t|w_{ij}) \propto \beta(w,t)\frac{\alpha_t}{n(t)+\bar\beta(t)} + \bar\beta(w,t)\frac{n(t,d=i)}{n(t)+\bar\beta(t)} + \frac{n(t,w=w_{ij})\left[n(t,d=i)+\alpha_t\right]}{n(t)+\bar\beta(t)}$$

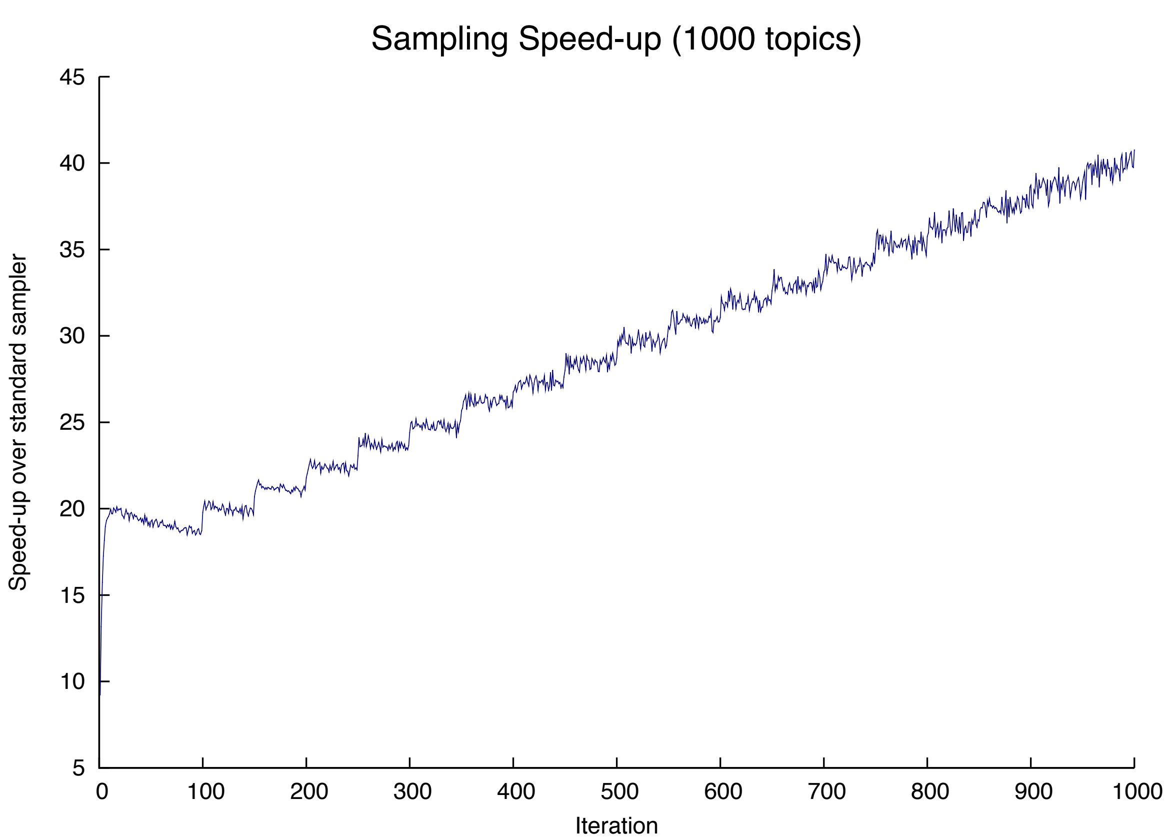
topic dependent, dense

- Simple sparse factorization doesn't work
- Use Cauchy Schwartz to upper-bound first term

$$\sum_t \beta(w,t)\frac{\alpha_t}{n(t)+\bar\beta(t)} \leq \left\|\beta(w,\cdot)\right\| \left\|\frac{\alpha.}{n(\cdot)+\bar\beta(\cdot)}\right\|$$

Collapsed vs Variational

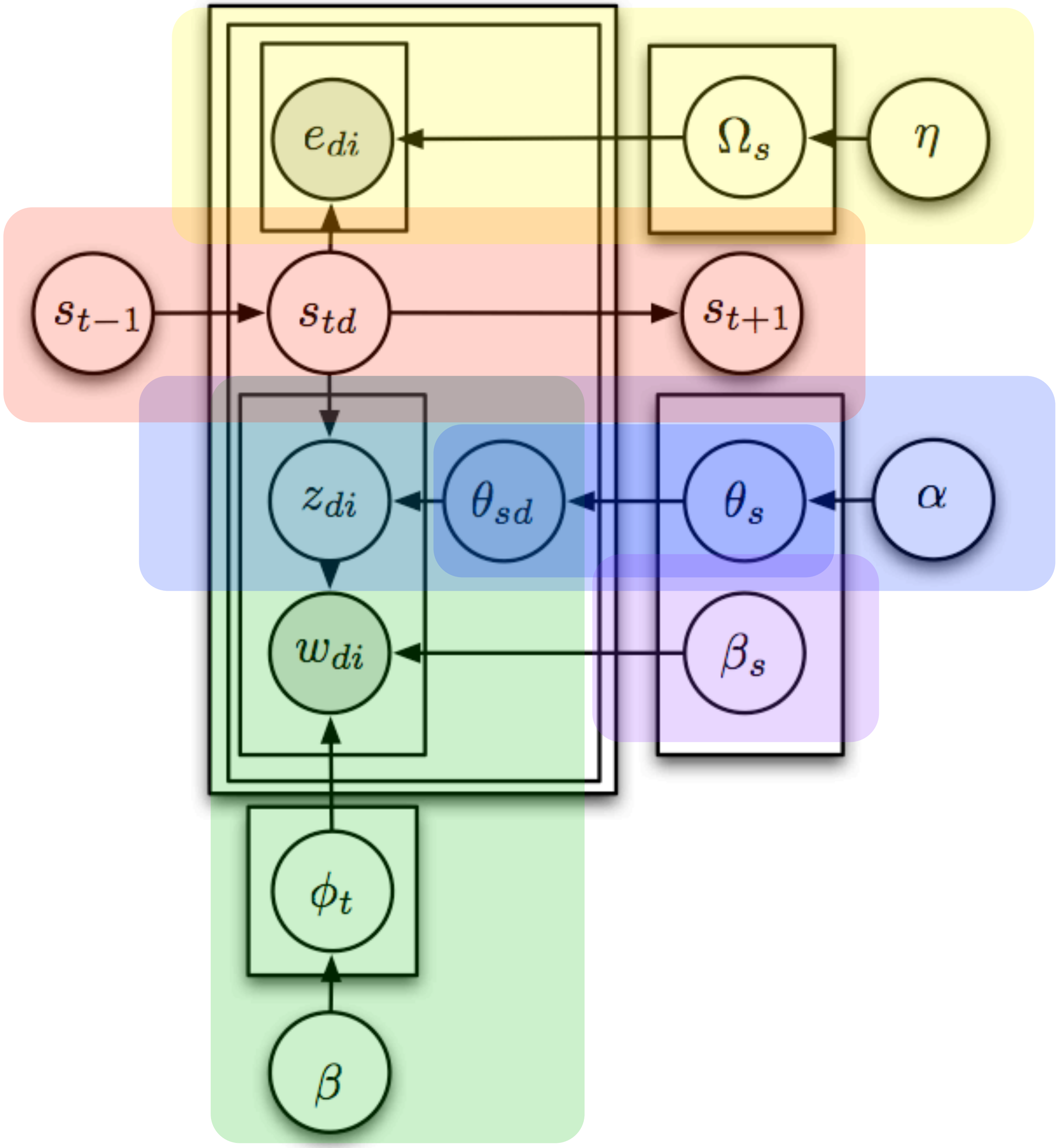
- Memory requirements (1k topics, 2M words)
  - Variational inference: 8GB RAM (no sparsity)
  - Collapsed sampler: 1.5GB RAM (rare words)
- Burn-in & sparsity exploit saves a lot



unif	doc	doc,word

- Cauchy Schwartz bound
- multilingual LDA
- word context
- smoothing over time

Fast Proposal



- In reality sparsity often not true for real proposal
- Guess sparse proxy
- In the storylines model this are the entities

- Variations on a theme
  inference for mixtures
- Parallel inference
  parallelization templates
- Samplers
  scaling up LDA