Big Data and Large Scale Inference

Amr Ahmed & Alex Smola

Research at Google

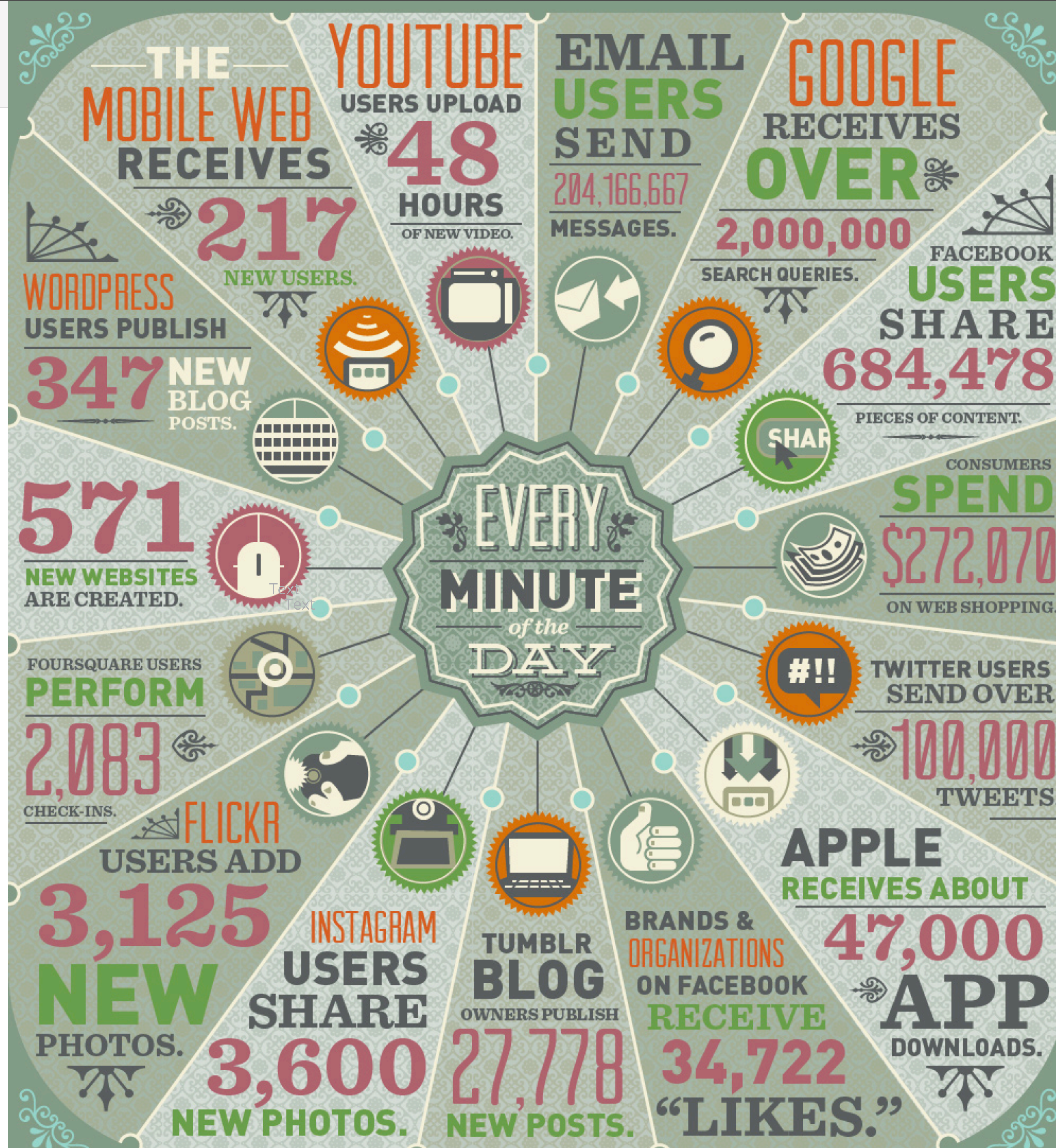
Data on the Internet

Size calibration

- **Tiny** (2 cores)
  (512MB, 50MFlops, 1000 examples)
- **Small** (4 cores)
  (4GB, 10GFlops, 100k examples)
- **Medium** (16 cores)
  (32GB, 100GFlops, 1M examples)
- **Large** (256 cores)
  (512GB, 1TFlops, 100M examples)
- **Massive**
  ... need to work hard to make it work

Google

This is not a toy dataset



http://wadejohnston1962.files.wordpress.com/2012/09/datainoneminute.jpg

Google User generated content

- Webpages (content, graph)
- Clicks (ad, page, social)
- Users (OpenID, FB Connect)
- e-mails (Hotmail, Y!Mail, Gmail)
- Photos, Movies (Flickr, YouTube, Vimeo ...)
- Cookies / tracking info (see Ghostery)
- Installed apps (Android market etc.)
- Location (Latitude, Loopt, Foursquared, Google Now)
- User generated content (Wikipedia & co)
- Ads (display, text, DoubleClick, Yahoo)
- Comments (Disqus, Facebook)
- Reviews (Yelp, Y!Local)
- Third party features (e.g. Experian)
- Social connections (LinkedIn, Facebook)
- Purchase decisions (Netflix, Amazon)
- Instant Messages (YIM, Skype, Gtalk)
- Search terms (Google, Bing)
- Timestamp (everything)
- News articles (BBC, NYTimes, Y!News)
- Blog posts (Tumblr, Wordpress)
- Microblogs (Twitter, Jaiku, Meme)
- Link sharing (Facebook, Delicious, Buzz)
- Network traffic

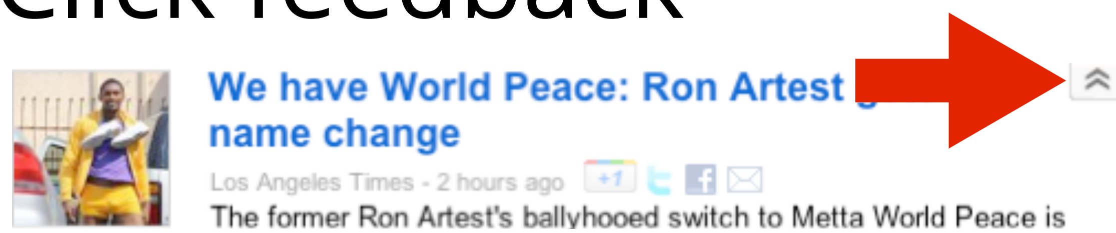
Source: place source info here

>1B images, 40h video/minute

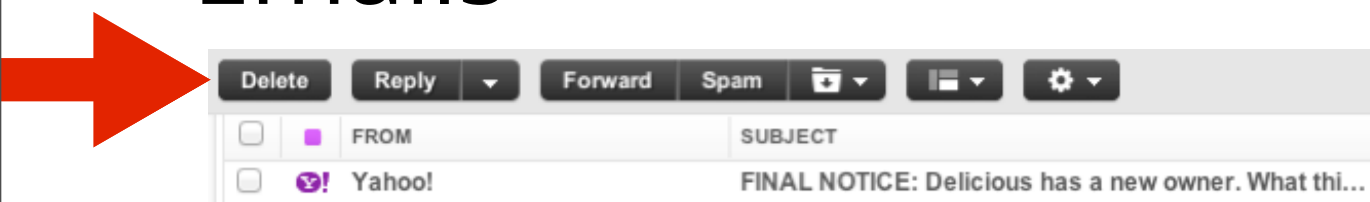
Google Some machine learning problems

- Ads

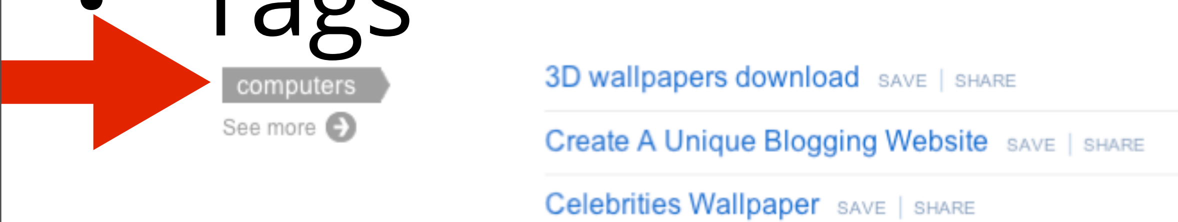
- Click feedback



- Emails

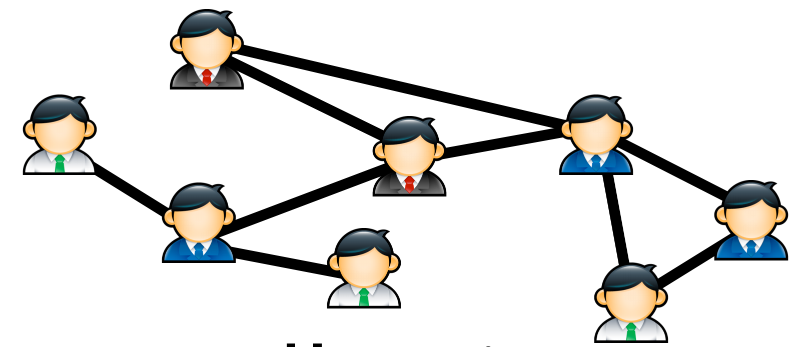


- Tags



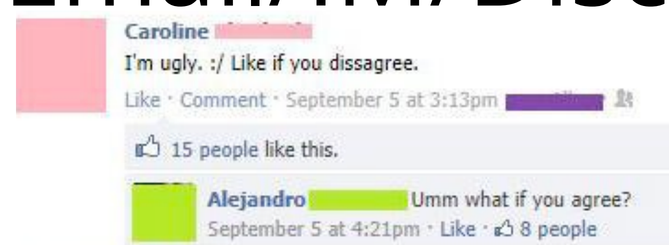
- Location

- Graphs

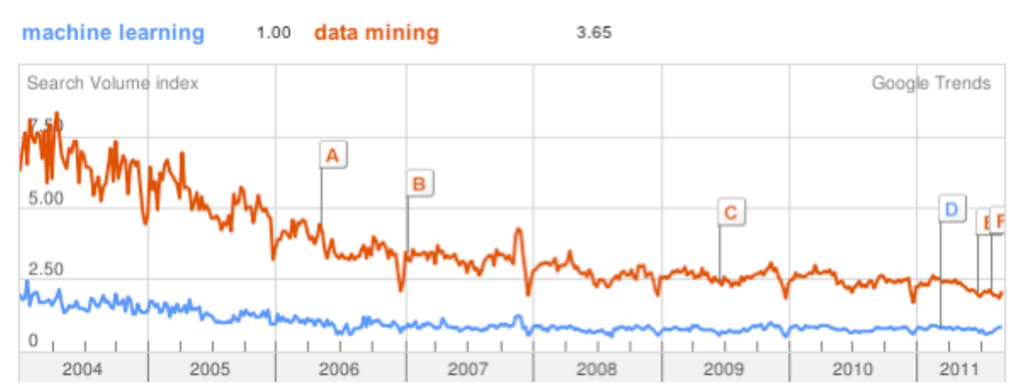


- Document collections

- Email/IM/Discussions



- Query stream



labeled label free

Summary

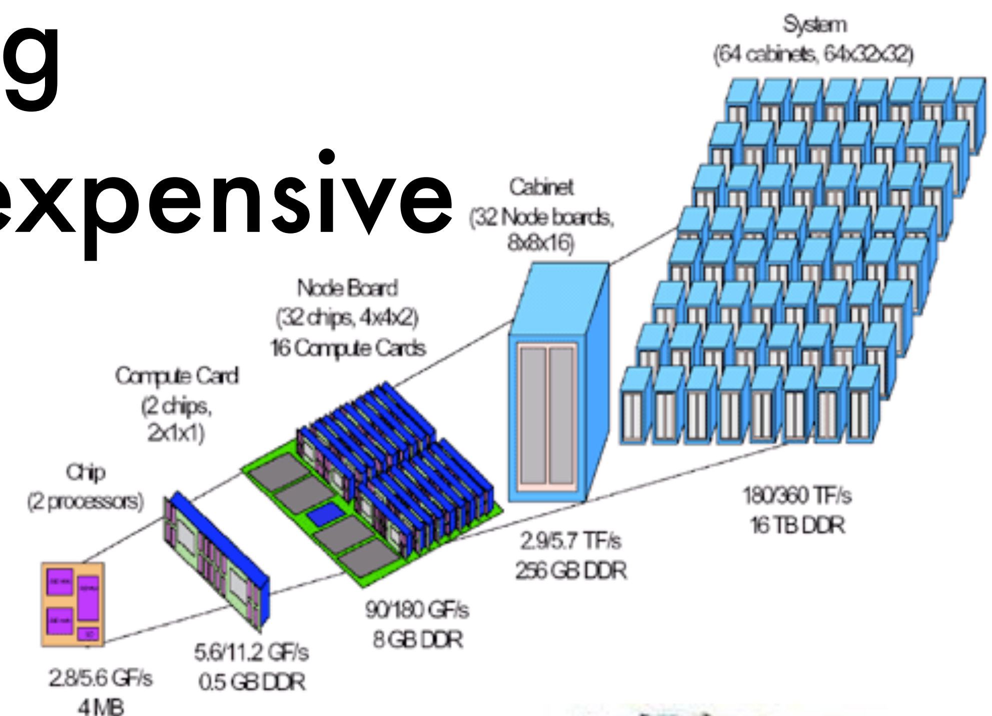
- Essentially infinite amount of data
- Labeling is prohibitively expensive
- Not scalable for i18n
- Even for *supervised* problems unlabeled data abounds. Use it.
- User-understandable structure for representation purposes
- Solutions are often customized to problem

We can only cover building blocks in tutorial.

Hardware

Commodity Hardware

- High Performance Computing
  Very reliable, custom built, expensive



- Consumer hardware
  Cheap, efficient, easy to replicate,
  not very reliable, deal with it!

The Joys of Real Hardware

Typical first year for a new cluster:

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)

~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)

~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)

~1 network rewiring (rolling ~5% of machines down over 2-day span)

~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)

~5 racks go wonky (40-80 machines see 50% packetloss)

~8 network maintenances (4 might cause ~30-minute random connectivity losses)

~12 router reloads (takes out DNS and external vips for a couple minutes)

~3 router failures (have to immediately pull traffic for an hour)

~dozens of minor 30-second blips for dns

~1000 individual machine failures

~thousands of hard drive failures

slow disks, bad memory, misconfigured machines, flaky machines, etc.

Slide courtesy of Jeff Dean

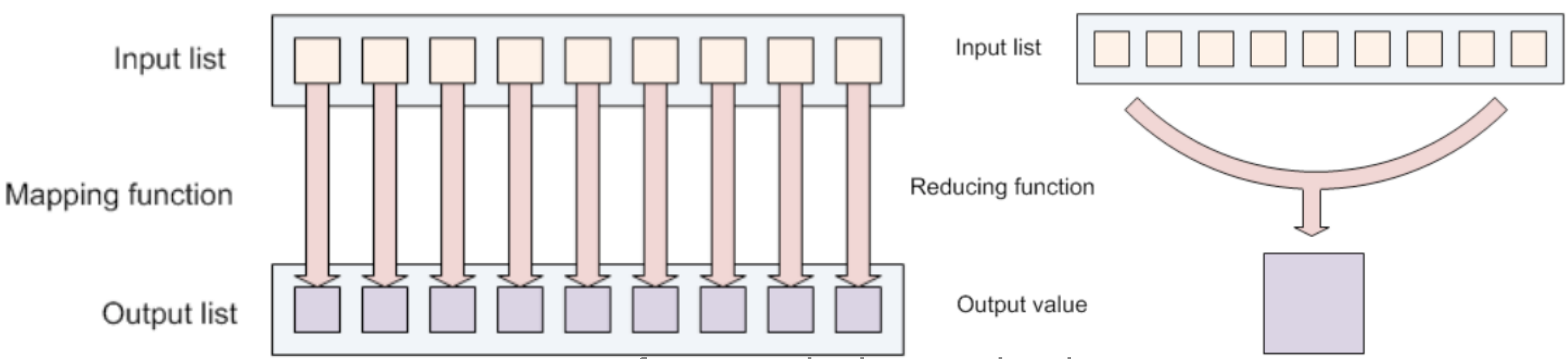
Google

Scaling problems

- Data (lower bounds)
  - >10 Billion documents (webpages, e-mails, ads, tweets)
  - >100 Million users on Google, Facebook, Twitter, Yahoo, Hotmail
  - >1 Million days of video on YouTube
  - >10 Billion images on Facebook
- Processing capability for single machine 1TB/hour
  But we have much more data
- Parameter space for models is too big for a single machine
  Personalize content for many millions of users
- Process on **many cores** and **many machines simultaneously**

Map Reduce

- 1000s of (faulty) machines
- Lots of jobs are mostly embarrassingly parallel (except for a sorting/transpose phase)
- Functional programming origins
  - Map(key,value) processes each (key,value) pair and outputs a new (key,value) pair
  - Reduce(key,value) reduces all instances with same key to aggregate



from Ramakrishnan, Sakrejda, Canon, DoE 2011

Map Reduce

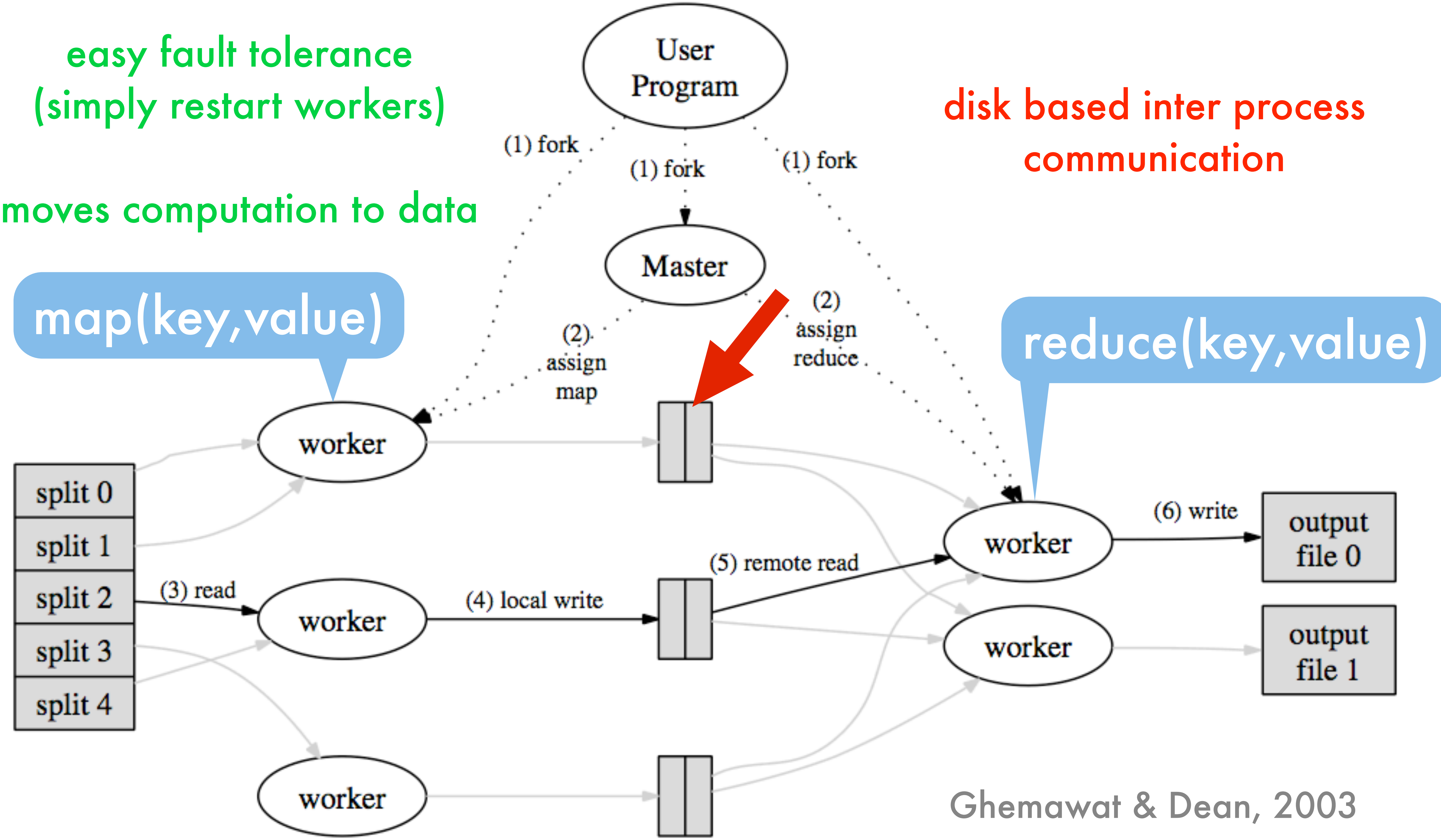
- 1000s of (faulty) machines
- Lots of jobs are mostly embarrassingly parallel
  (except for a sorting/transpose phase)
- Functional programming origins
  - Map(key,value)
    processes each (key,value) pair and outputs a new (key,value) pair
  - Reduce(key,value)
    reduces all instances with same key to aggregate
- Example - extremely naive wordcount
  - Map(docID, document)
    for each document emit many (wordID, count) pairs
  - Reduce(wordID, count)
    sum over all counts for given wordID and emit (wordID, aggregate)

Map Reduce

easy fault tolerance
(simply restart workers)

moves computation to data

disk based inter process
communication



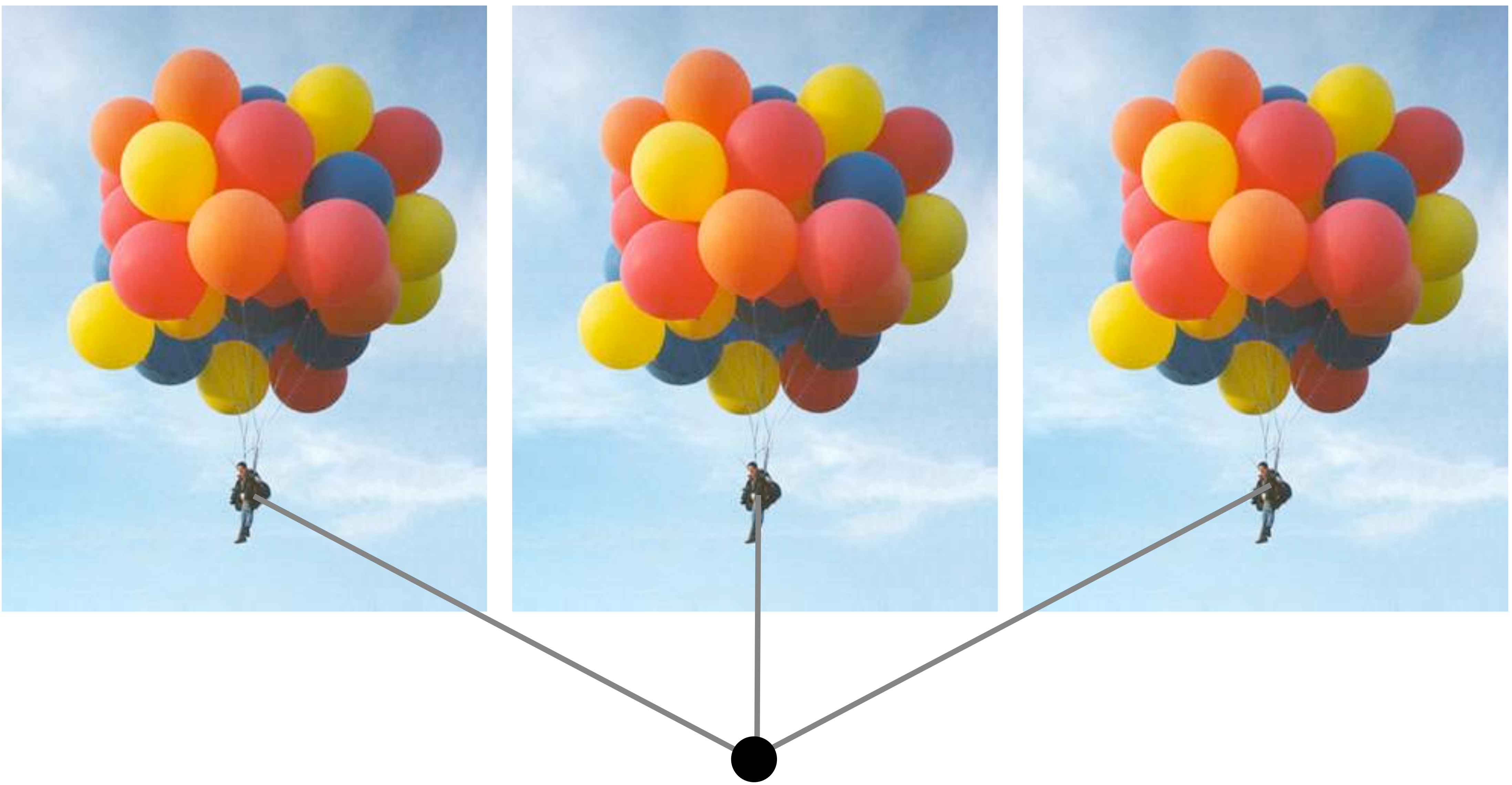
Ghemawat & Dean, 2003

Map Combine Reduce

- Combine aggregates keys before sending to the reducer (saves bandwidth)
- Map must be stateless in blocks
- Reduce must be commutative in data
- Fault tolerance
  - Start jobs where the data is
    (move code note data - nodes run the file system, too)
  - Restart machines if maps fail (have replicas)
  - Restart reducers based on intermediate data
- Good fit for many algorithms
- Good if only a small number of MapReduce iterations needed
- Need to request machines at each iteration (time consuming)
- State lost in between maps
- Communication only via file I/O

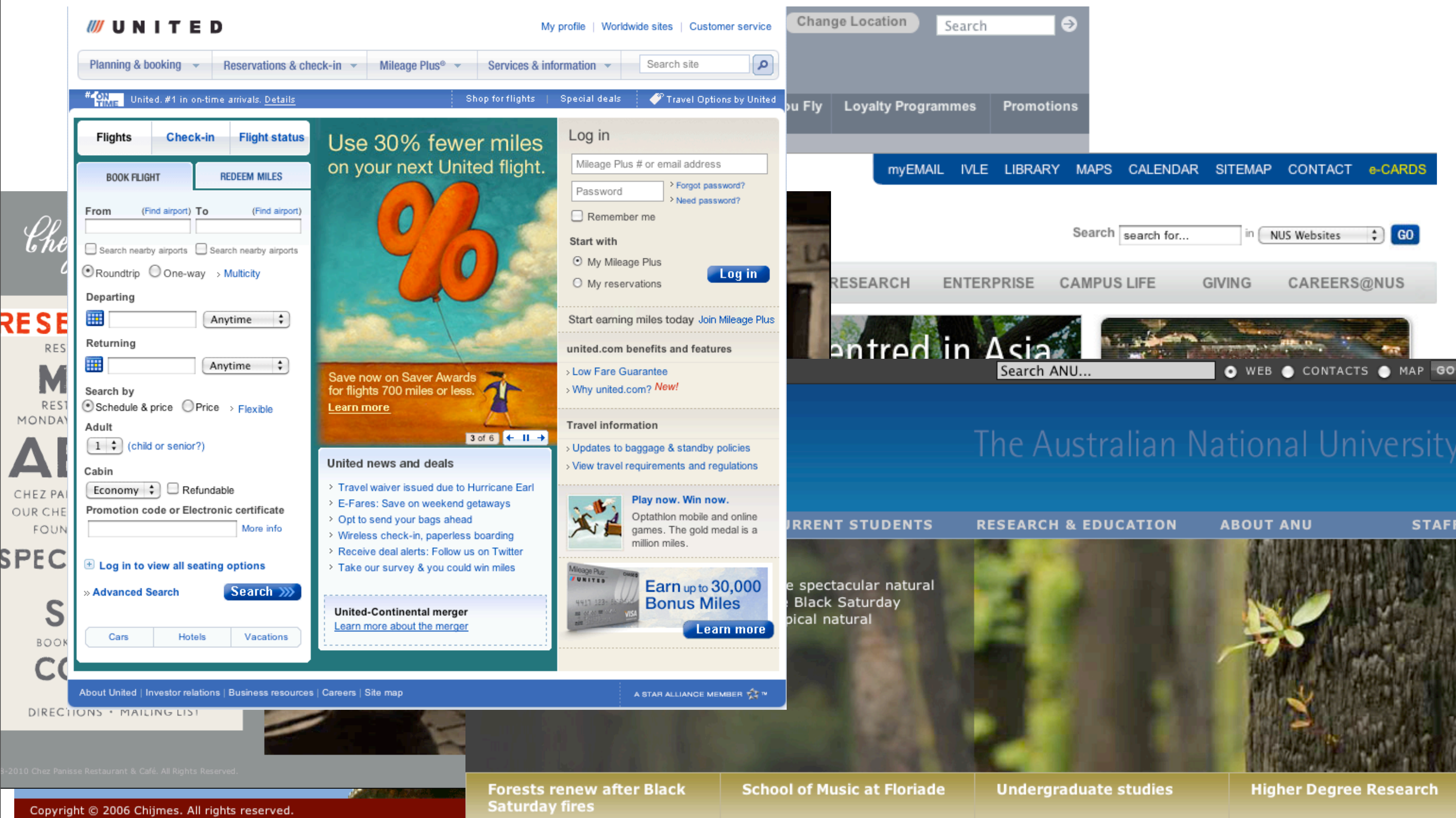
Motivation - Topic models

Clustering

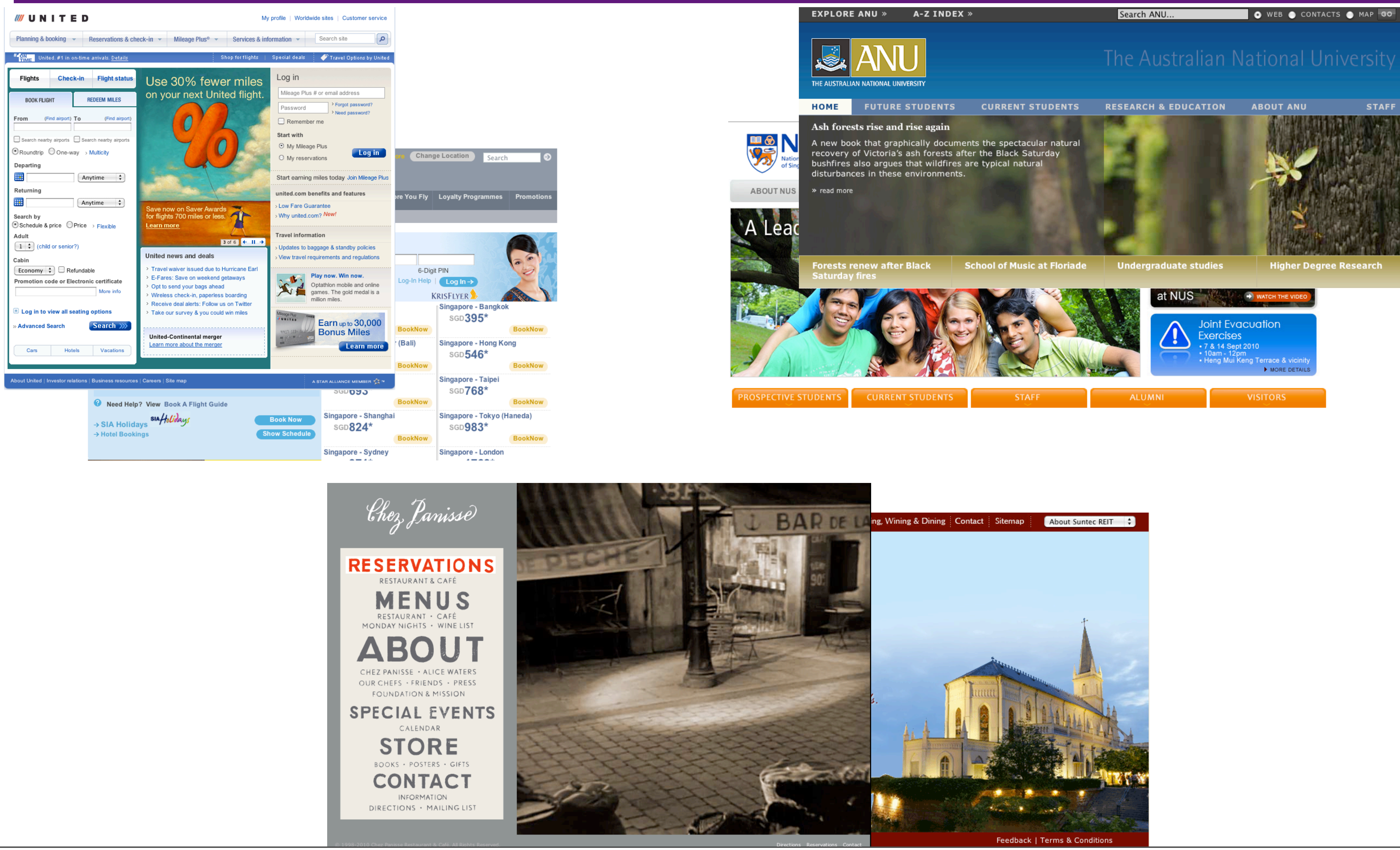


Clustering

Clustering

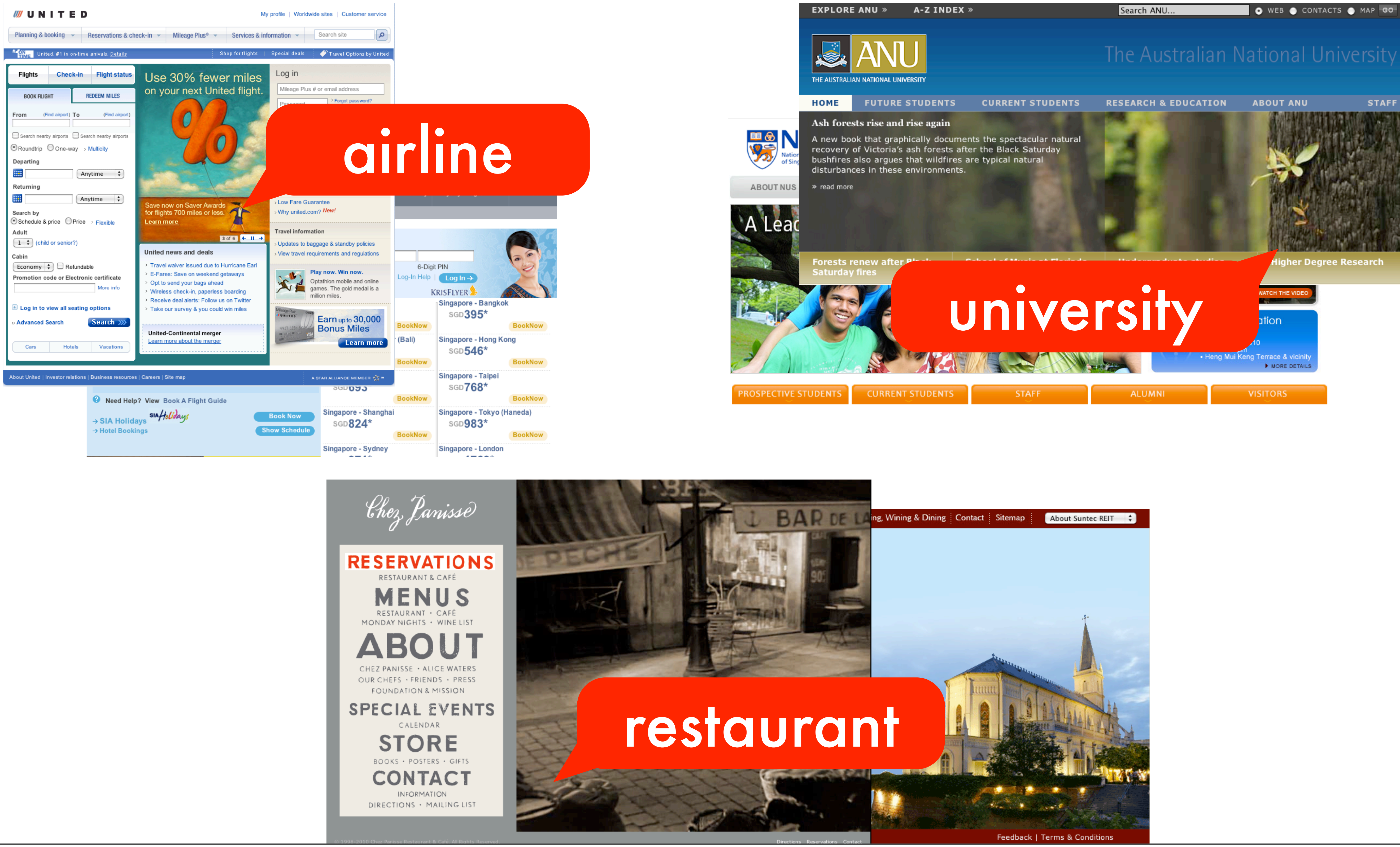

UNITED
My profile | Worldwide sites | Customer service
Planning & booking
Reservations & check-in
Mileage Plus®
Services & information
Search site
United. #1 in on-time arrivals. Details
Shop for flights | Special deals | Travel Options by United
Flights
Check-in
Flight status
BOOK FLIGHT
REDEEM MILES
From (Find airport) To (Find airport)
Search nearby airports
Roundtrip One-way Multicity
Departing
Anytime
Returning
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

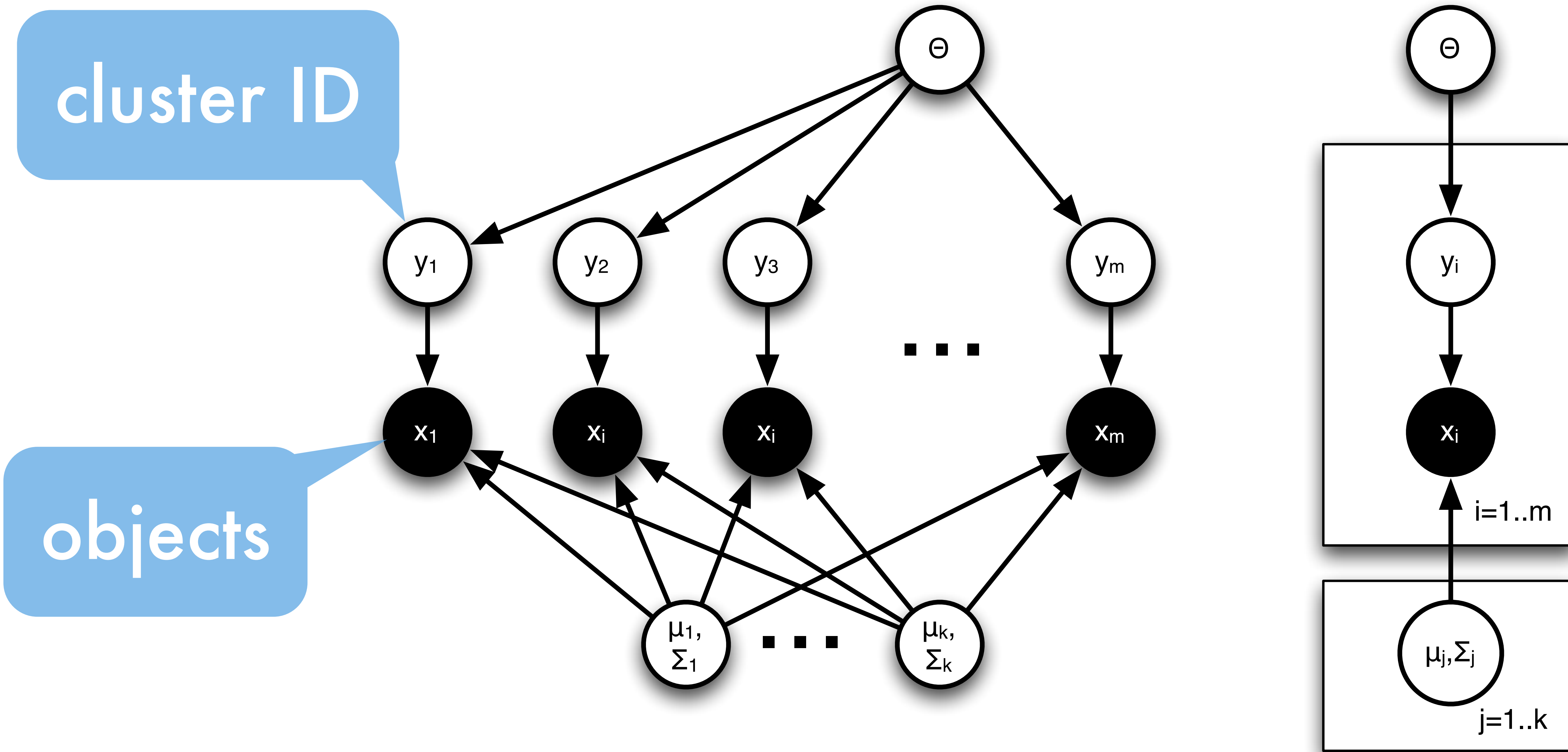


UNITED
Use 30% fewer miles on your next United flight.
Log in
United news and deals
EXPLORE ANU »
A-Z INDEX »
ANU
THE AUSTRALIAN NATIONAL UNIVERSITY
The Australian National University
HOME
FUTURE STUDENTS
CURRENT STUDENTS
RESEARCH & EDUCATION
ABOUT ANU
STAFF
Ash forests rise and rise again
Forests renew after Black Saturday fires
School of Music at Floriade
Undergraduate studies
Higher Degree Research
Joint Evacuation Exercises
PROSPECTIVE STUDENTS
CURRENT STUDENTS
STAFF
ALUMNI
VISITORS
Singapore - Bangkok
SGD395*
Singapore - Hong Kong
SGD546*
Singapore - Taipei
SGD768*
Singapore - Shanghai
SGD824*
Singapore - Tokyo (Haneda)
SGD983*
Singapore - Sydney
Singapore - London
Earn up to 30,000 Bonus Miles
Chez Panisse
RESERVATIONS
RESTAURANT & CAFÉ
MENUS
RESTAURANT · CAFÉ
MONDAY NIGHTS · WINE LIST
ABOUT
CHEZ PANISSE · ALICE WATERS
OUR CHEFS · FRIENDS · PRESS
FOUNDATION & MISSION
SPECIAL EVENTS
CALENDAR
STORE
BOOKS · POSTERS · GIFTS
CONTACT
INFORMATION
DIRECTIONS · MAILING LIST
Contact
Sitemap
About Suntec REIT
Feedback | Terms & Conditions

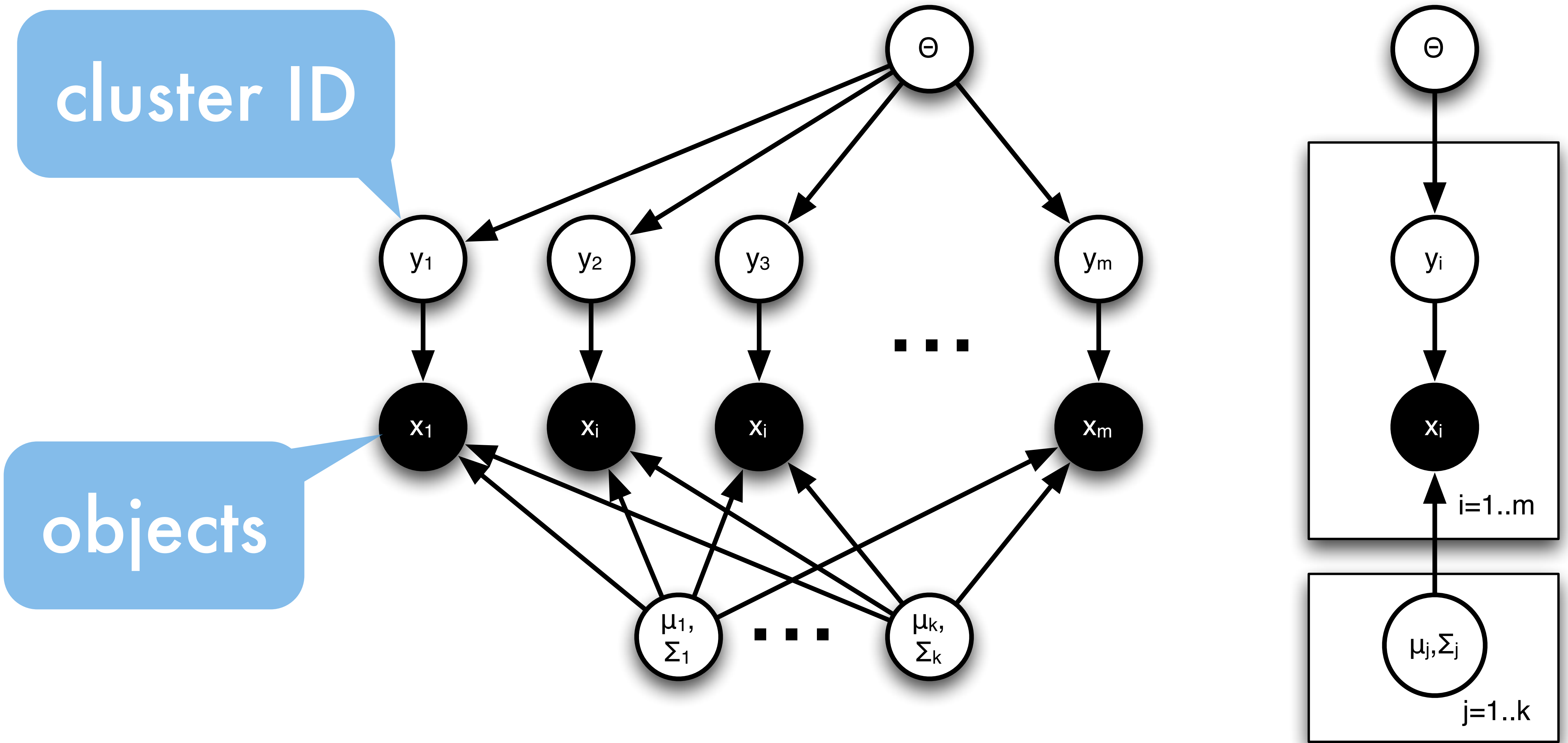
Clustering


airline
university
restaurant
UNITED
Use 30% fewer miles on your next United flight.
ANU
The Australian National University
HOME
FUTURE STUDENTS
CURRENT STUDENTS
RESEARCH & EDUCATION
ABOUT ANU
STAFF
Ash forests rise and rise again
PROSPECTIVE STUDENTS
CURRENT STUDENTS
STAFF
ALUMNI
VISITORS
Chez Panisse
RESERVATIONS
MENUS
ABOUT
SPECIAL EVENTS
STORE
CONTACT

Generative Model



Generative Model



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

What can we cluster?

What can we cluster?

products

urls

mails

text

news

users

queries

locations

spammers

ads

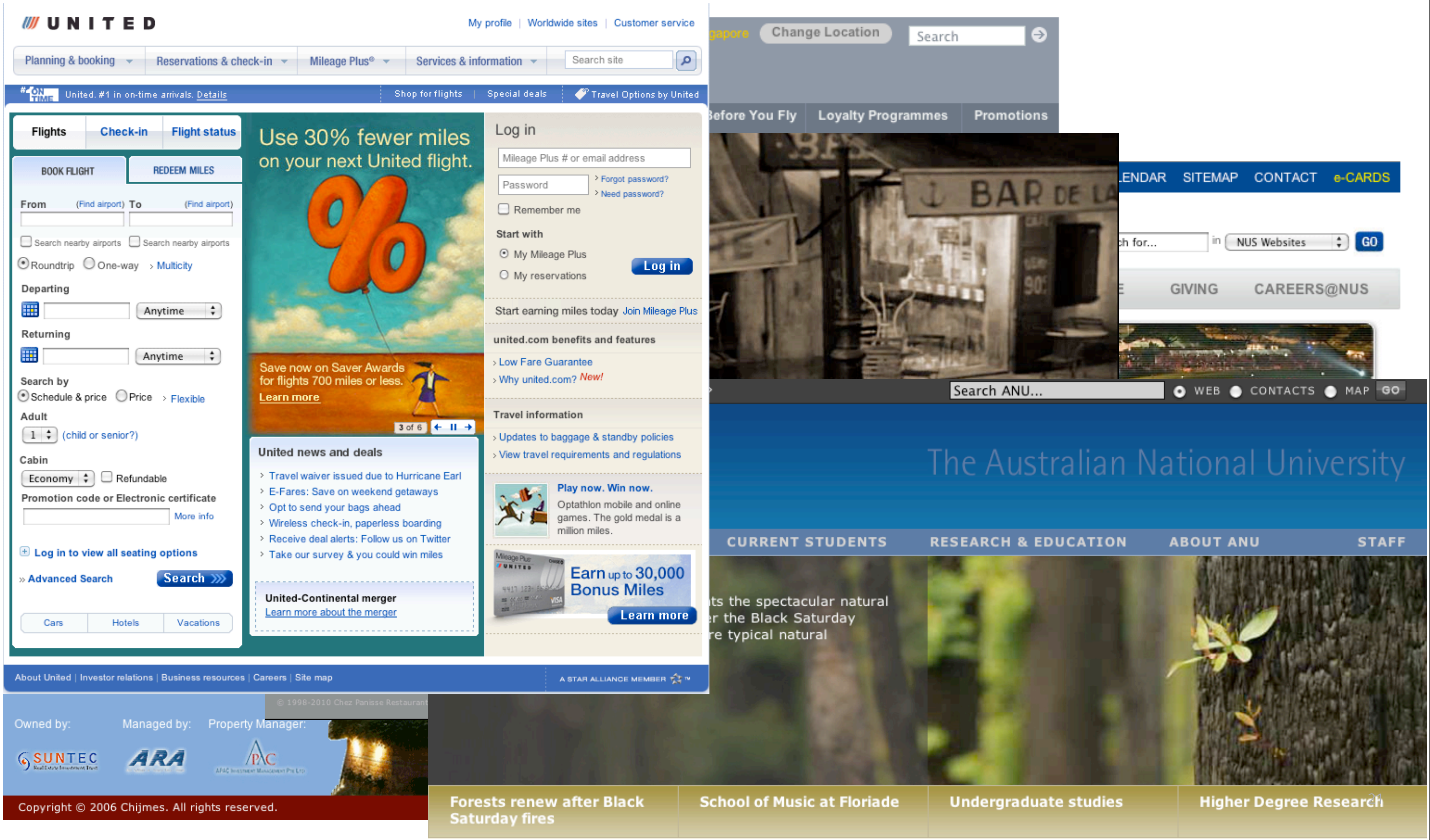
events

abuse

Grouping objects

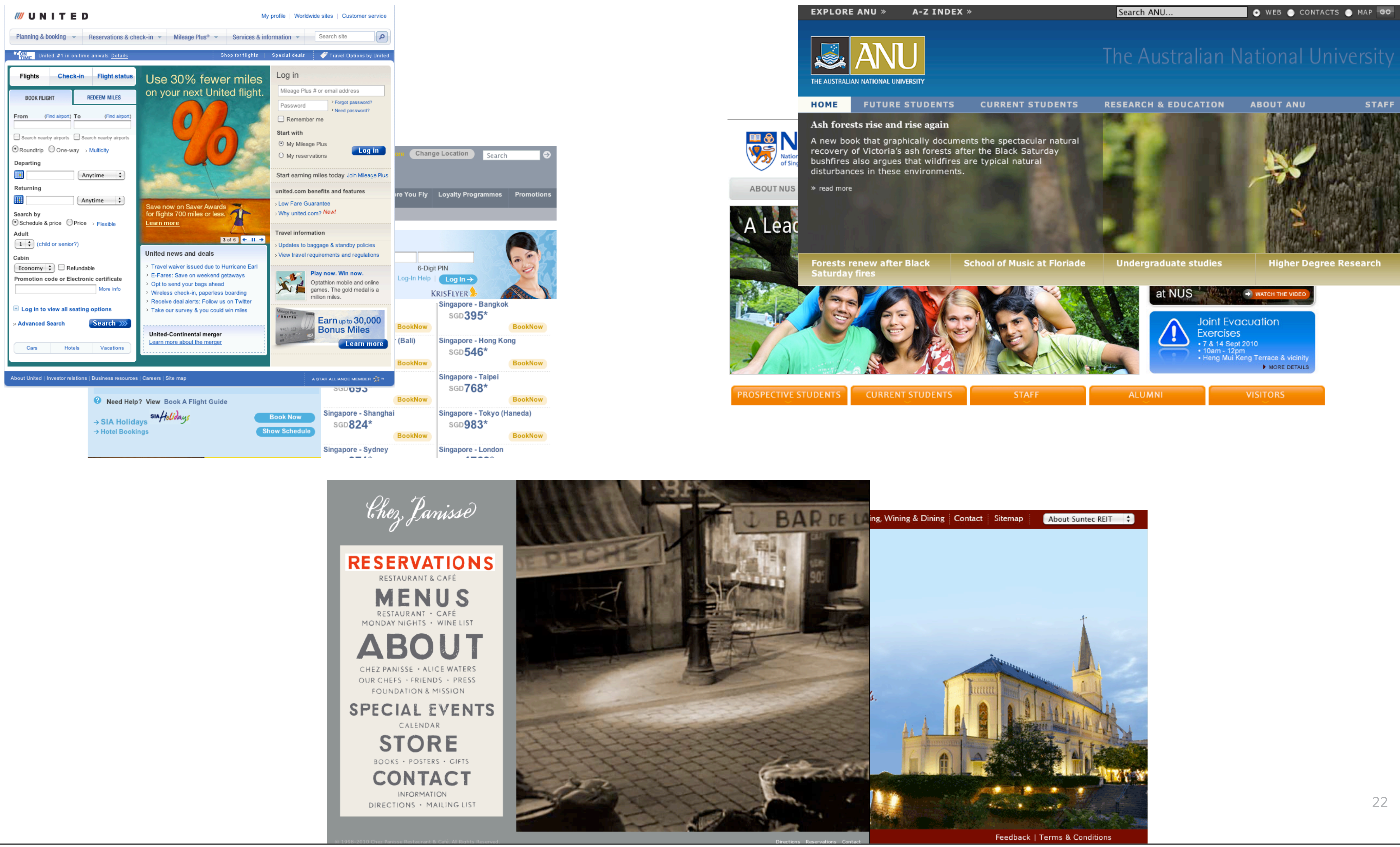
Grouping objects

Grouping objects

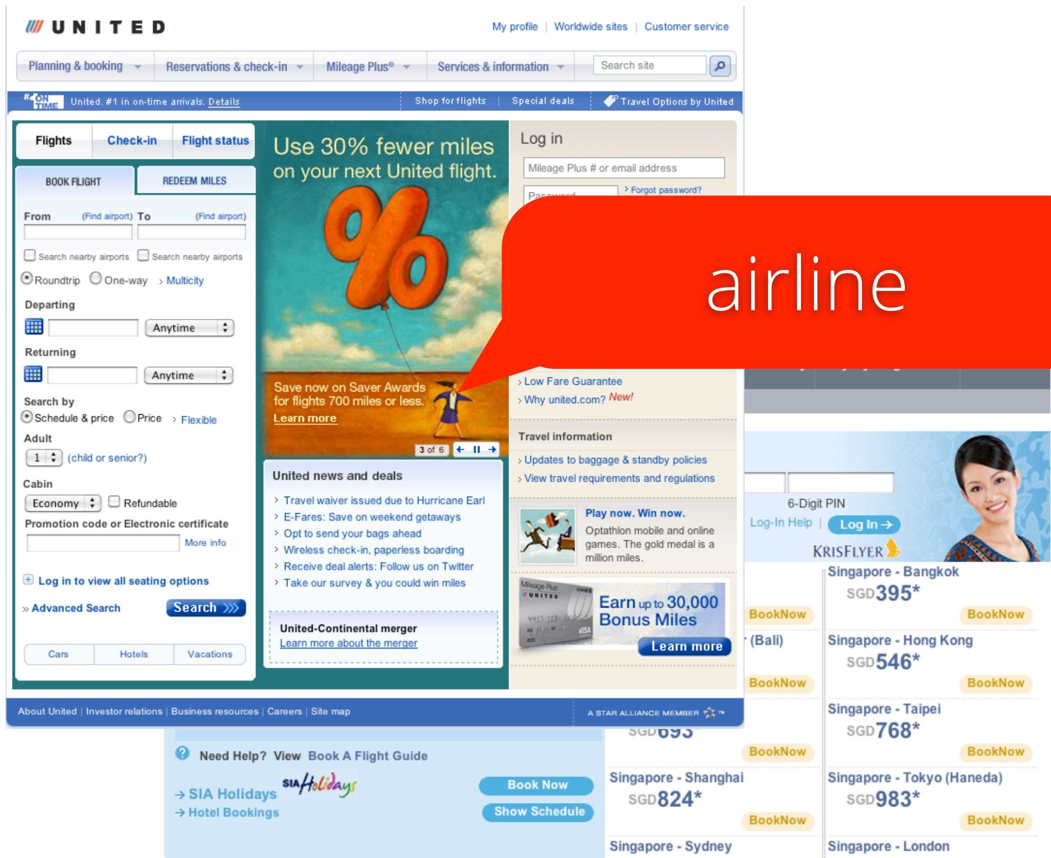


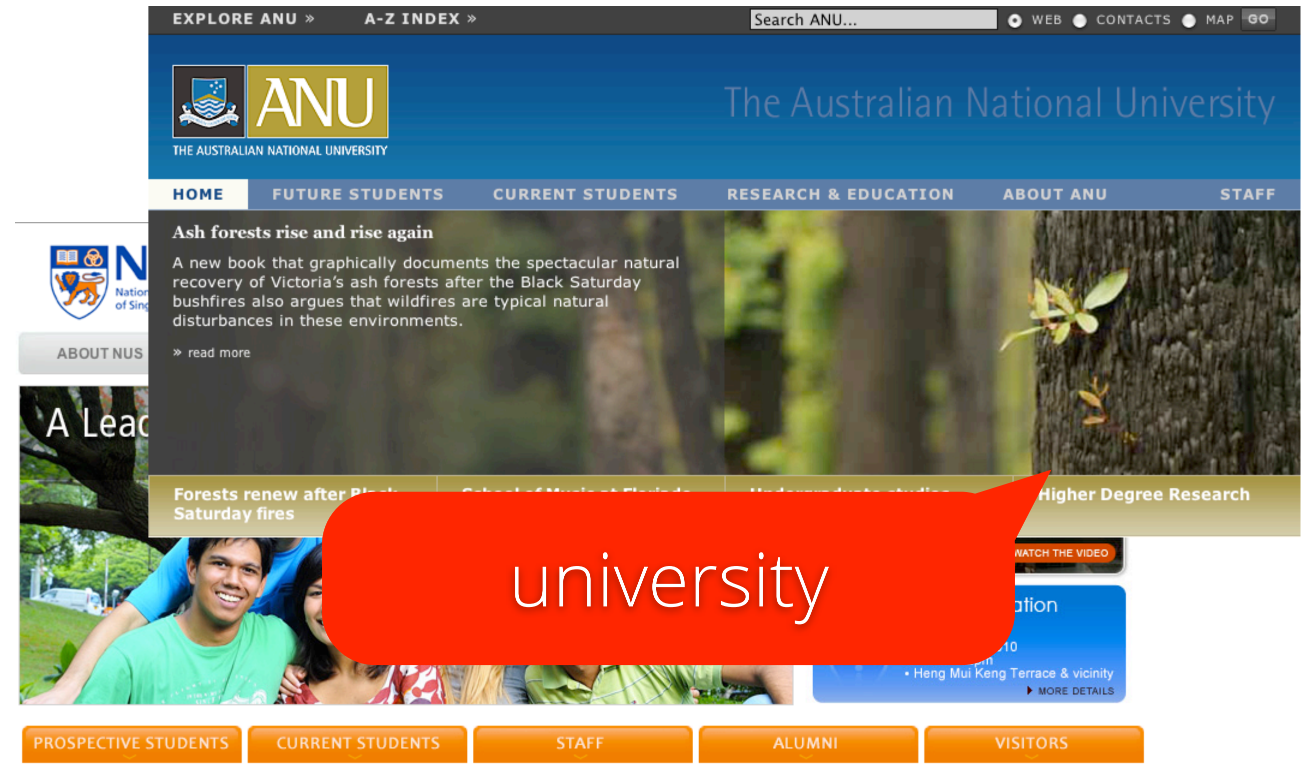
Google™

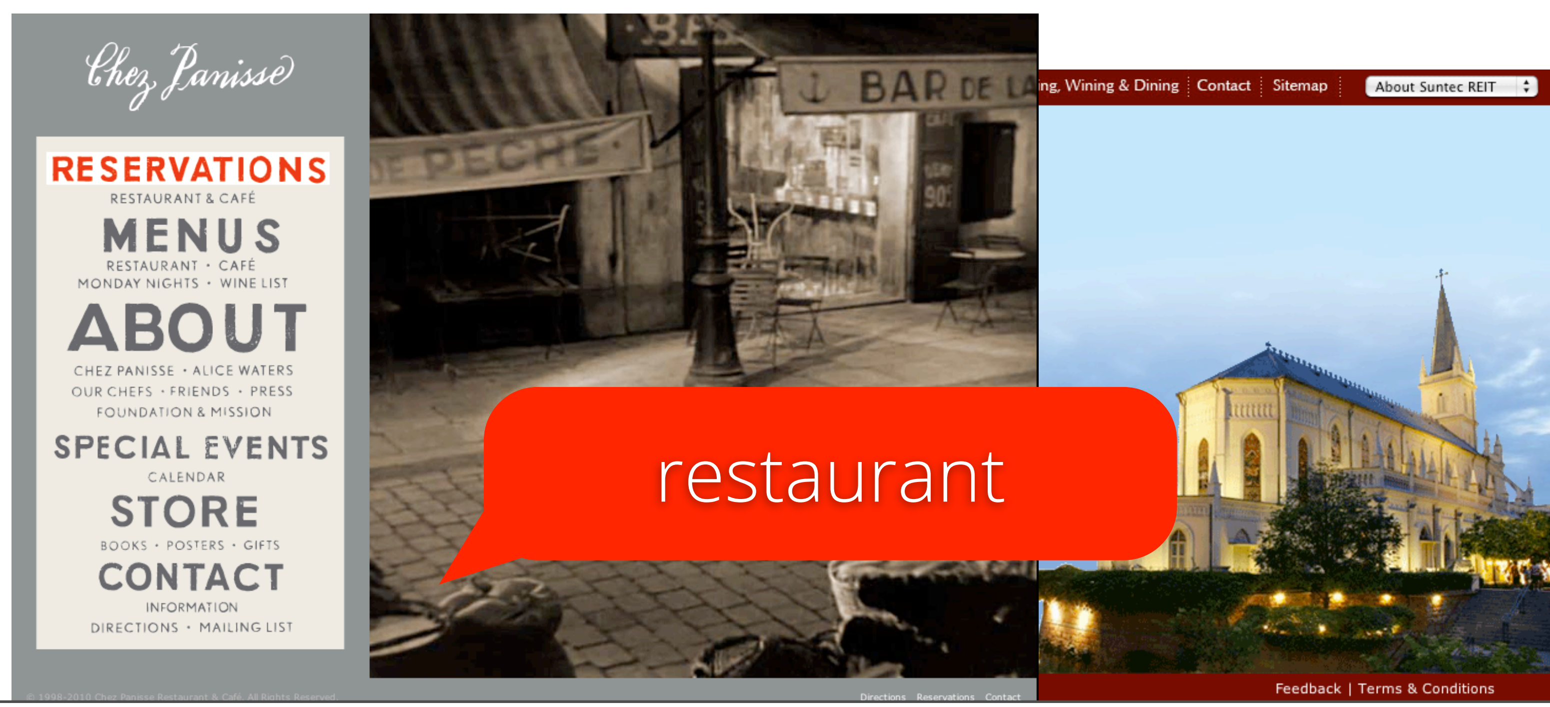
Grouping objects



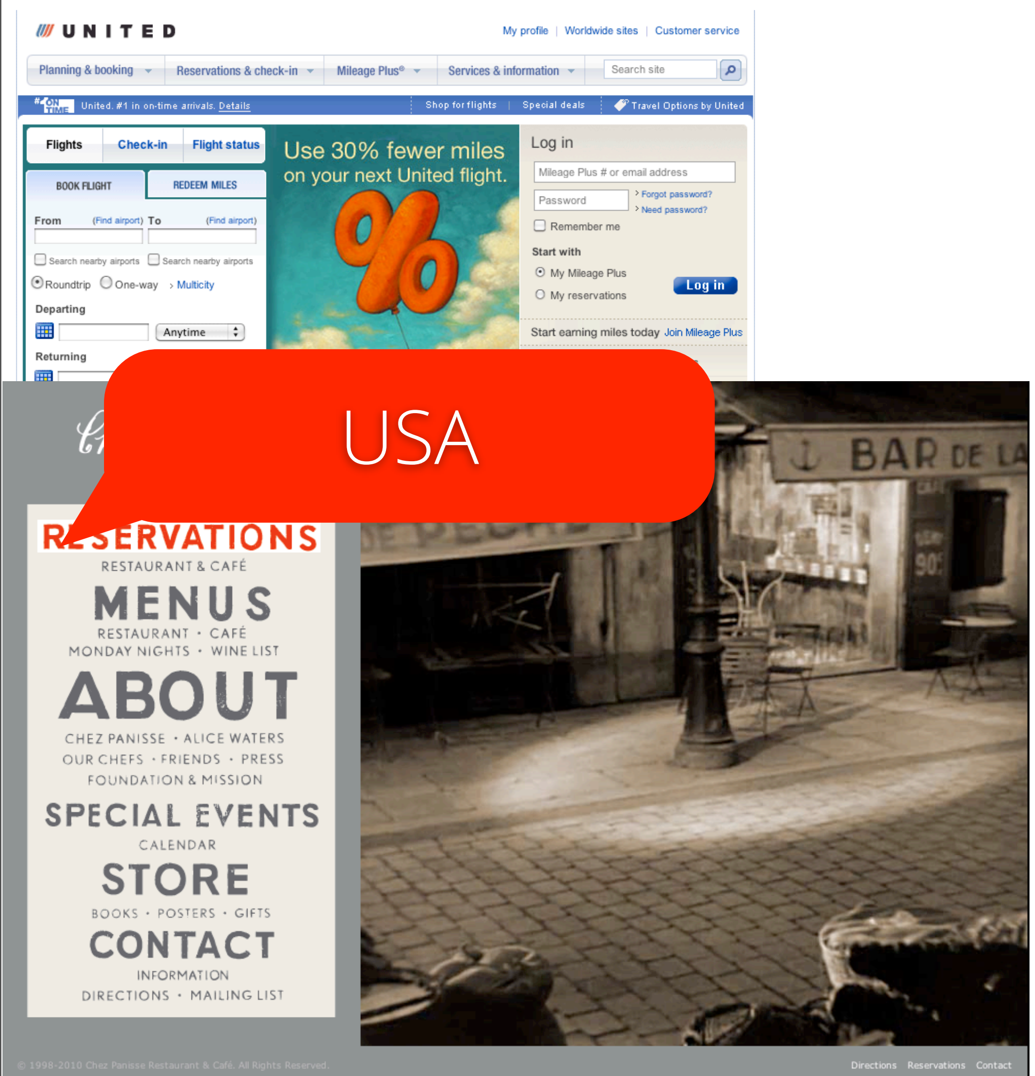
Grouping objects

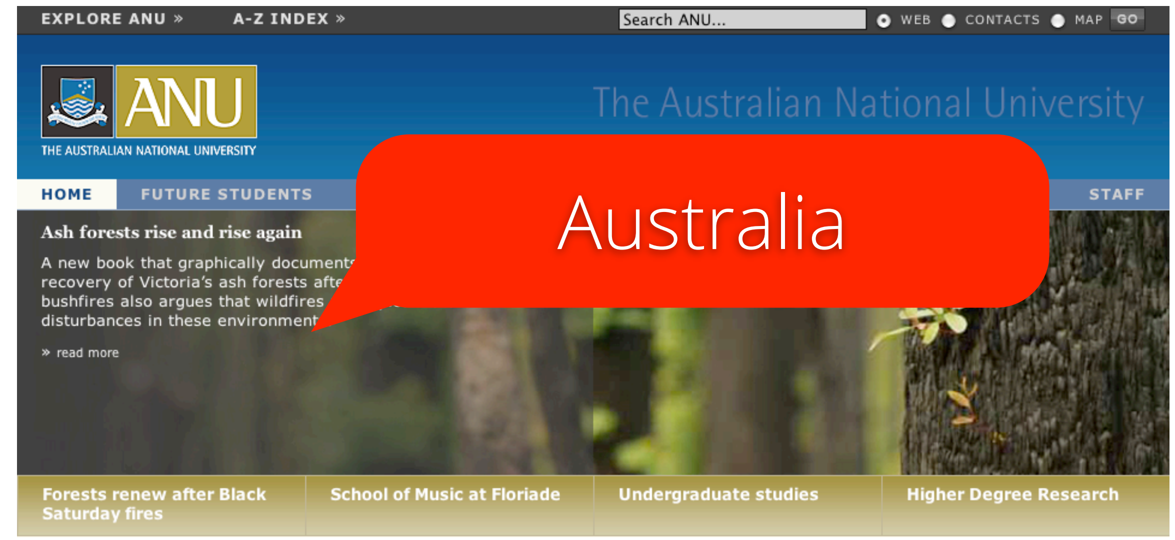






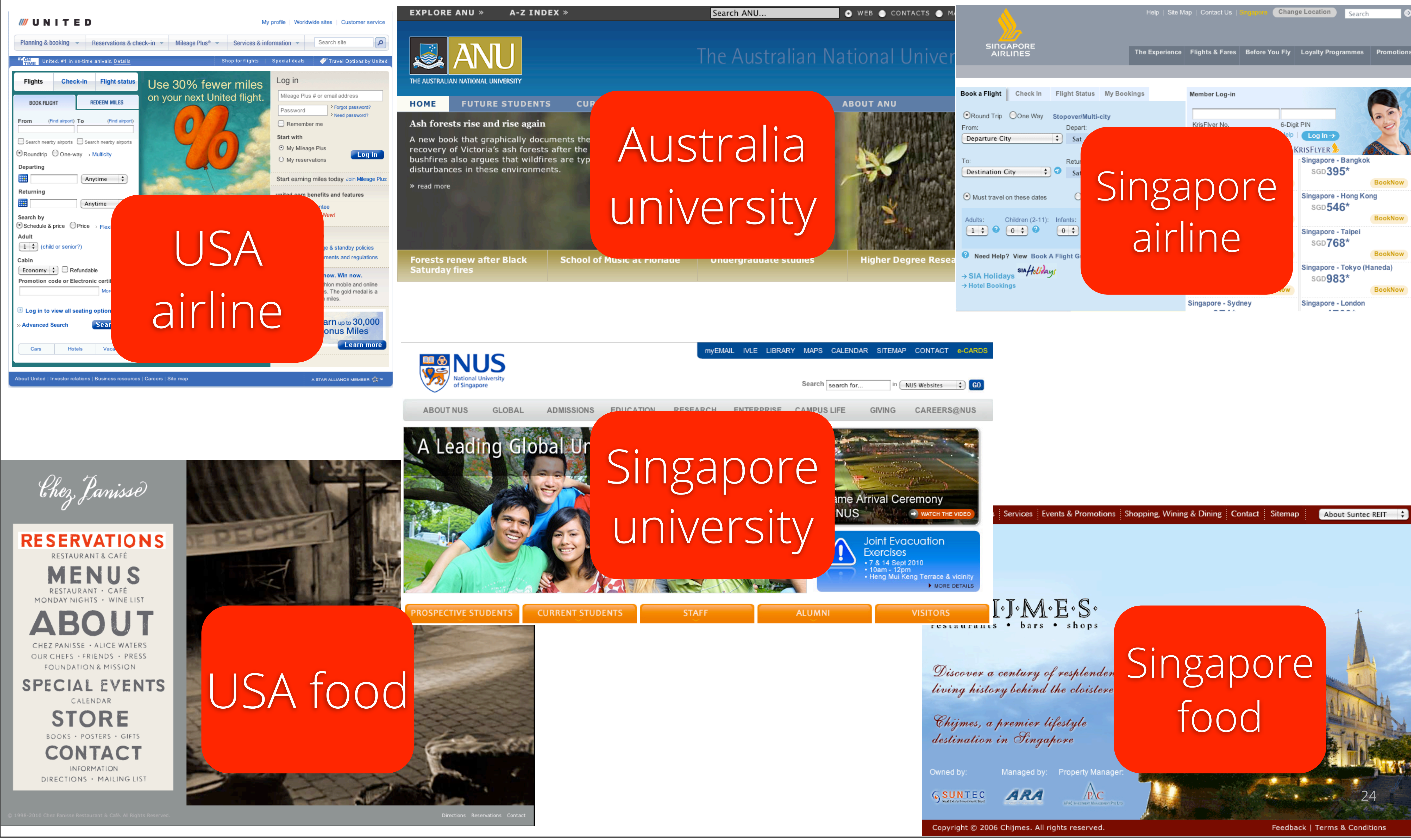
Grouping objects





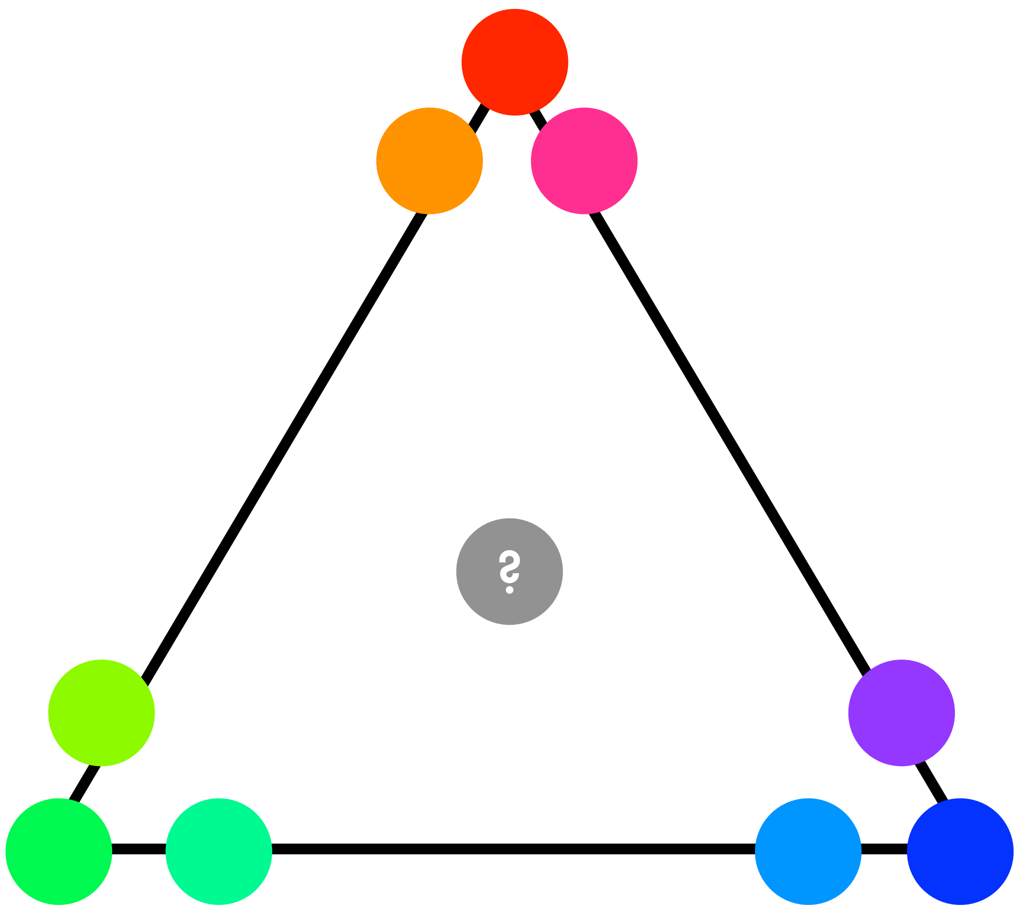
Google

Topic Models



Clustering & Topic Models

Clustering

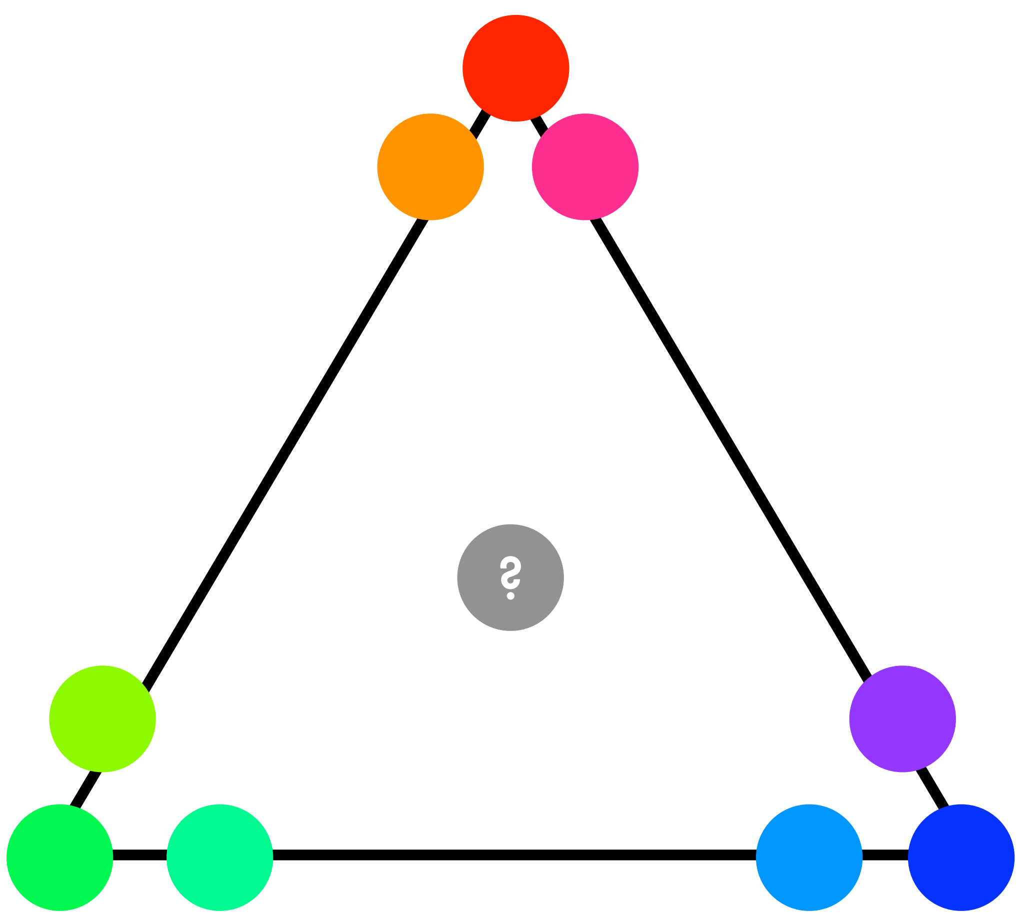


group objects
by prototypes

Google

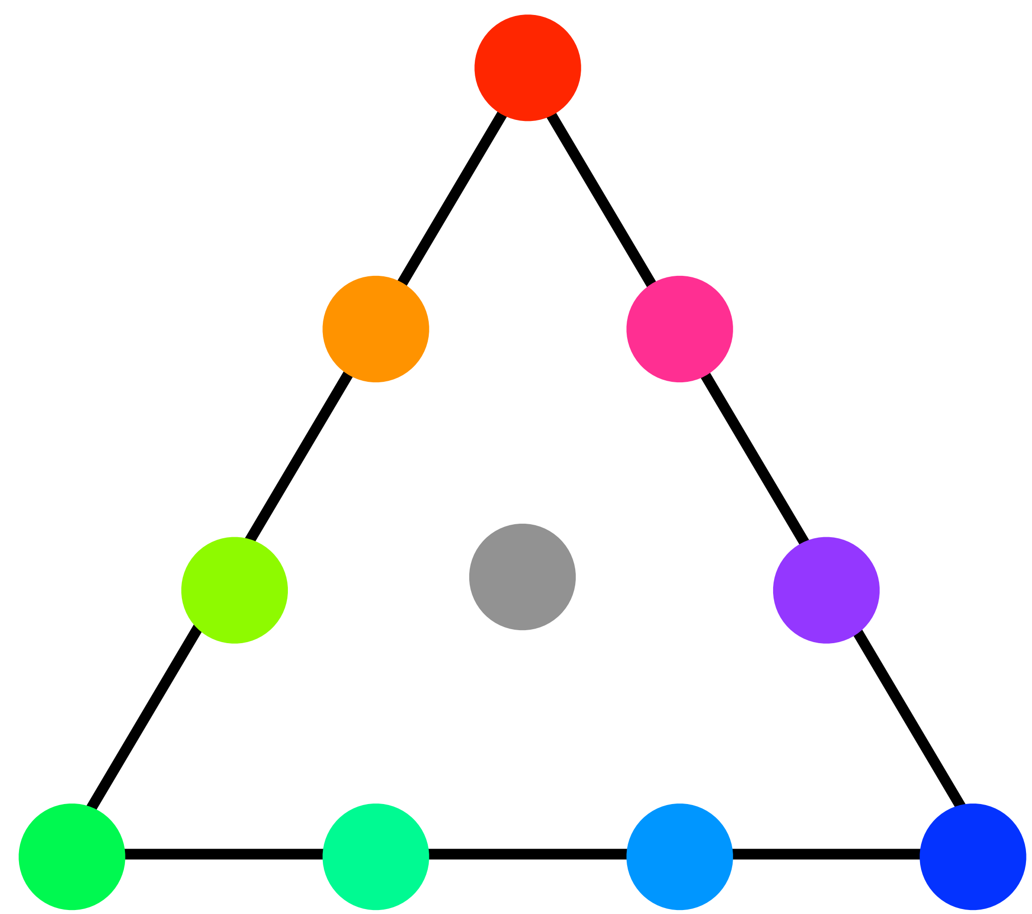
Clustering & Topic Models

Clustering



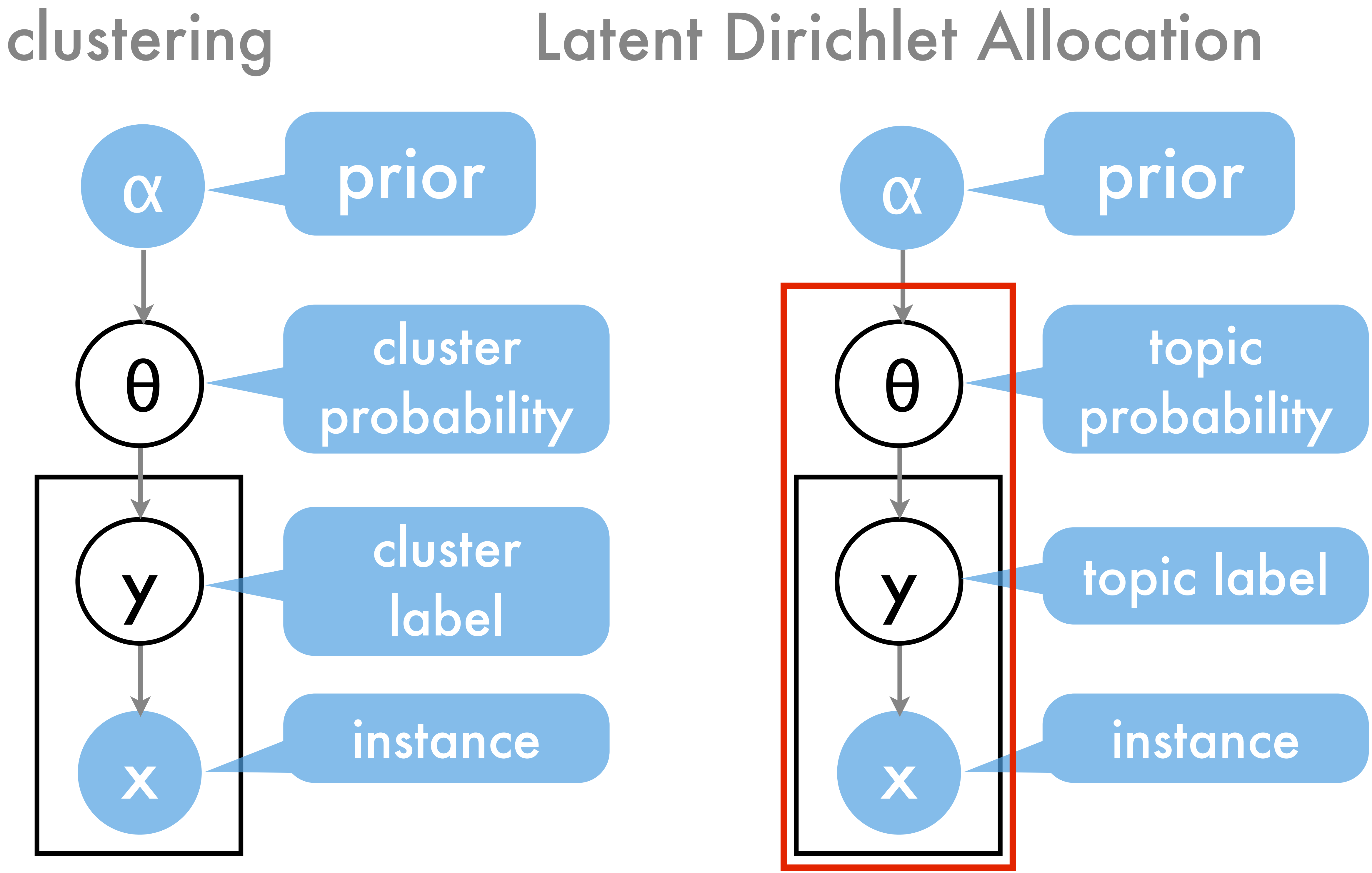
group objects
by prototypes

Topics

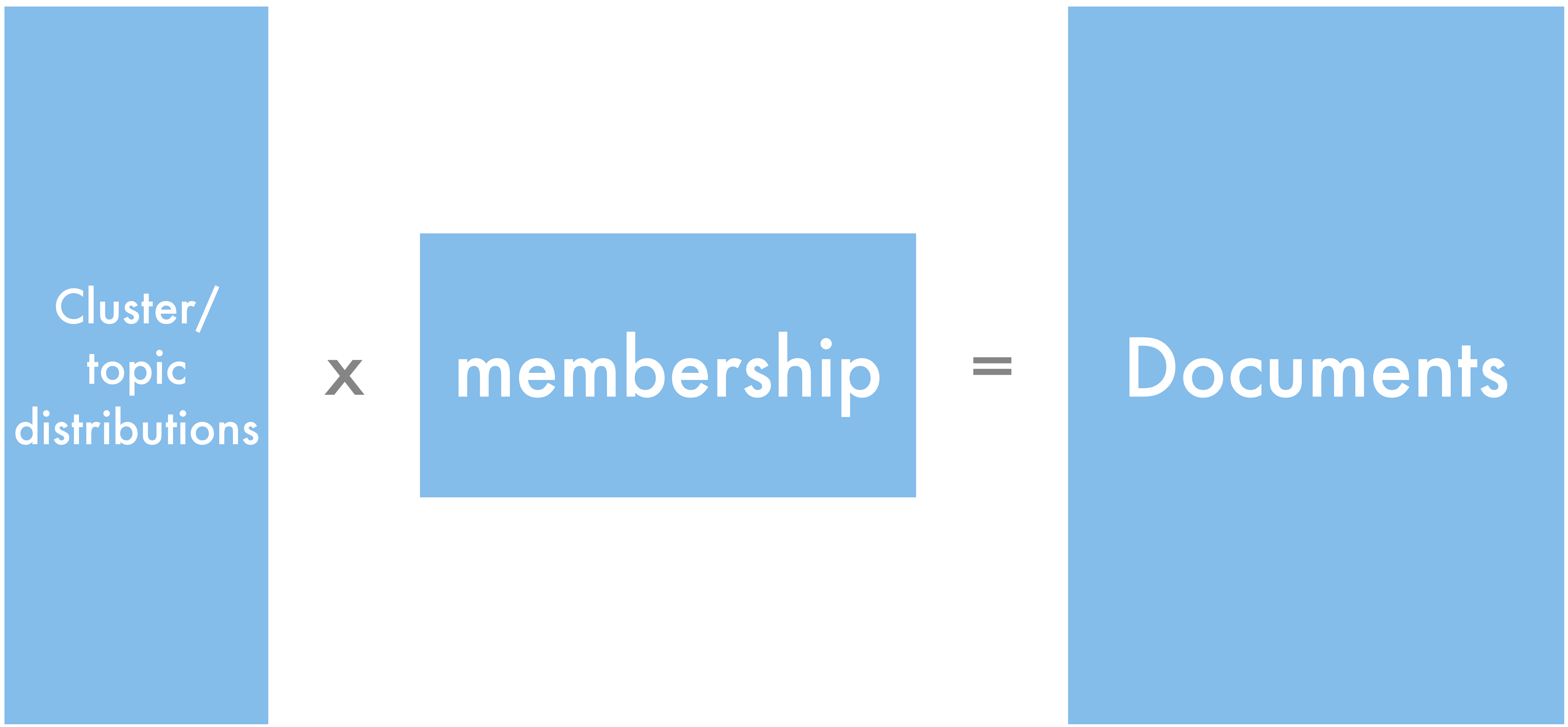


decompose objects
into prototypes

Clustering & Topic Models



Clustering & Topic Models



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

Topics in text

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.

Latent Dirichlet Allocation; Blei, Ng, Jordan, JMLR 2003

Example Topics

"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

Latent Dirichlet Allocation; Blei, Ng, Jordan, JMLR 2003

Dirichlet Distribution

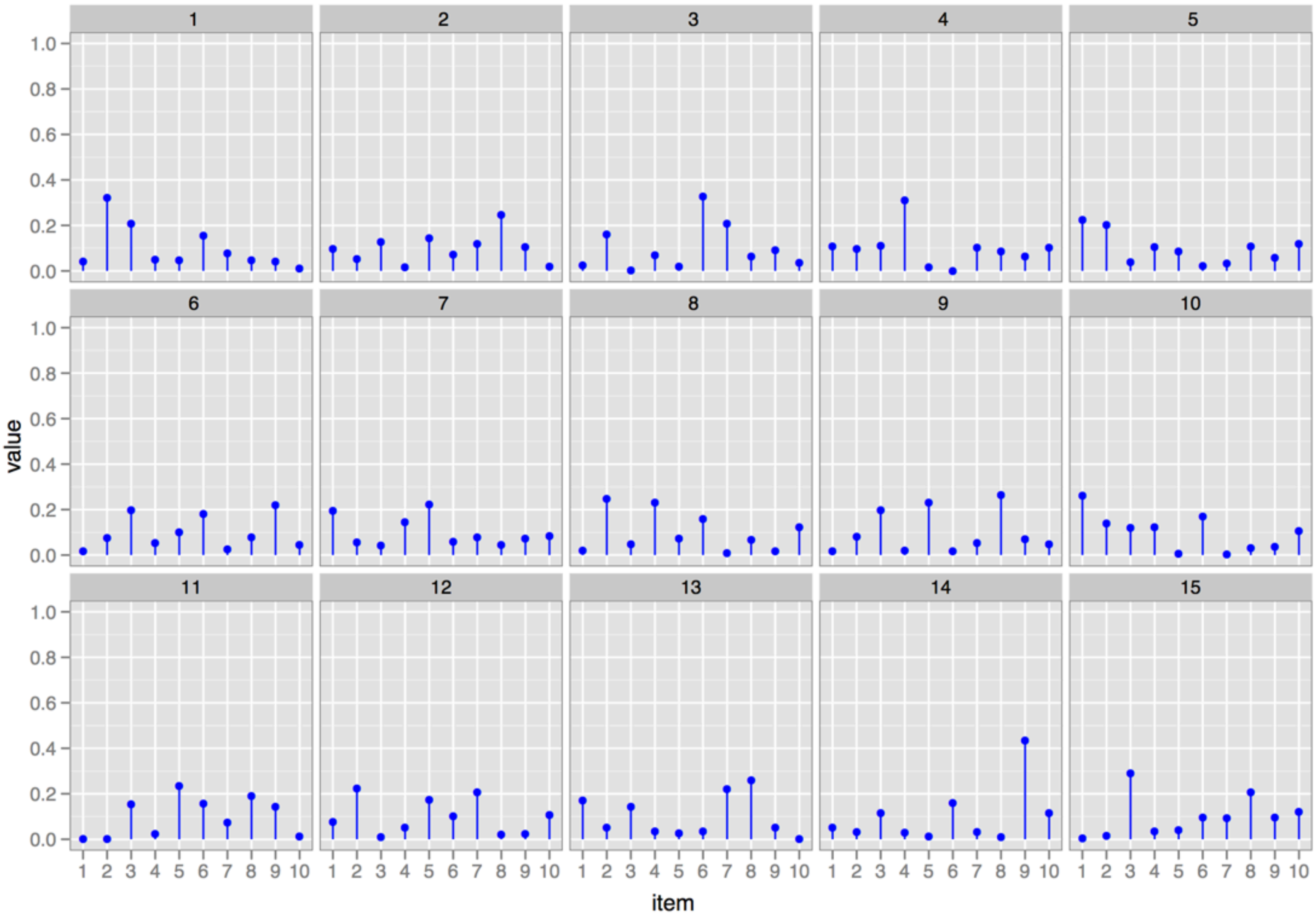
- Is a distribution over the simplex, i.e. positive vectors that sum to 1:

$$P(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}$$

- α controls the shape of the distribution
- Expectations: $$E[\theta_i|\alpha] = \frac{\alpha_i}{\sum_i \alpha_i}$$

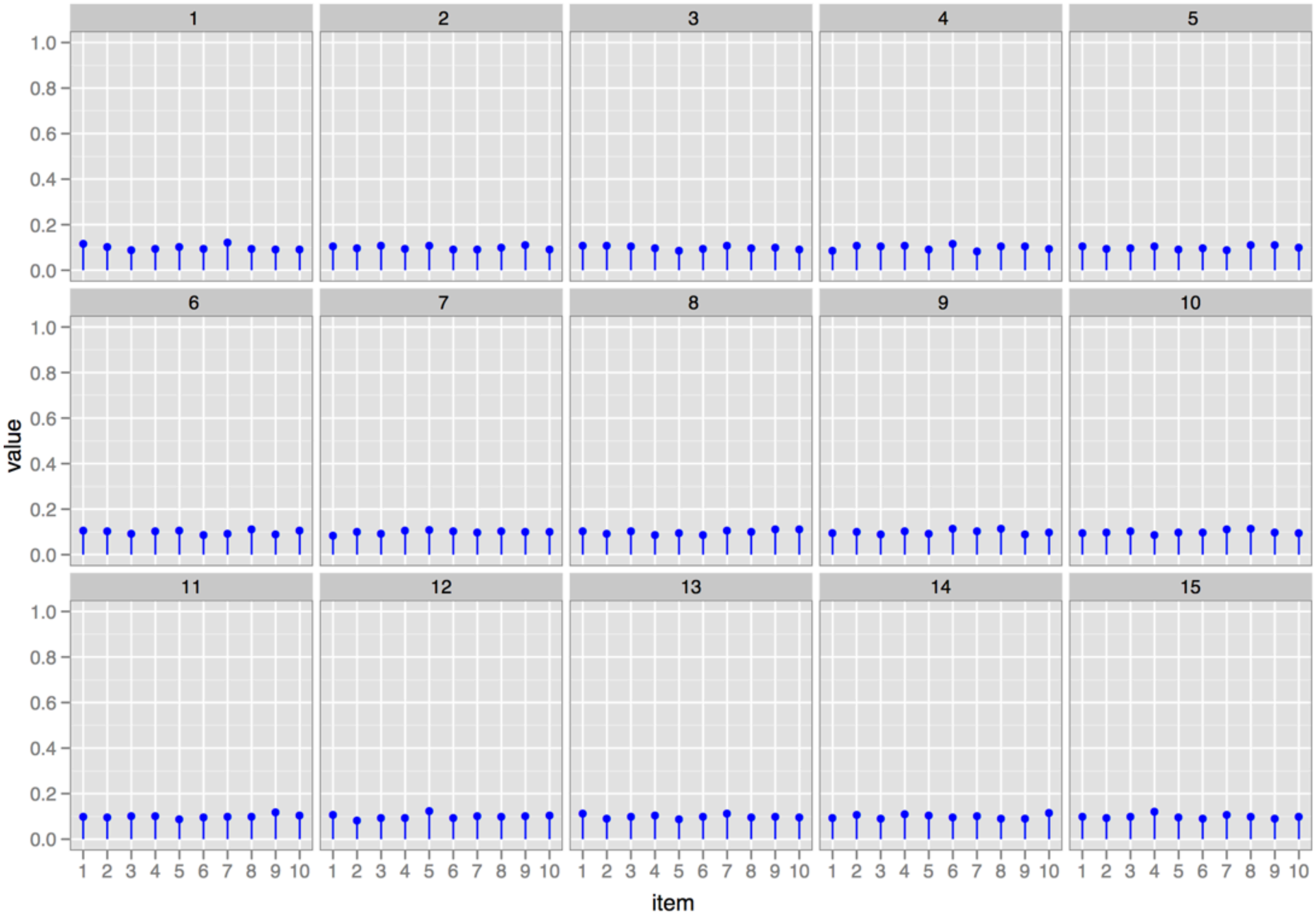
- Conjugate to the multinomial distribution

α = 1



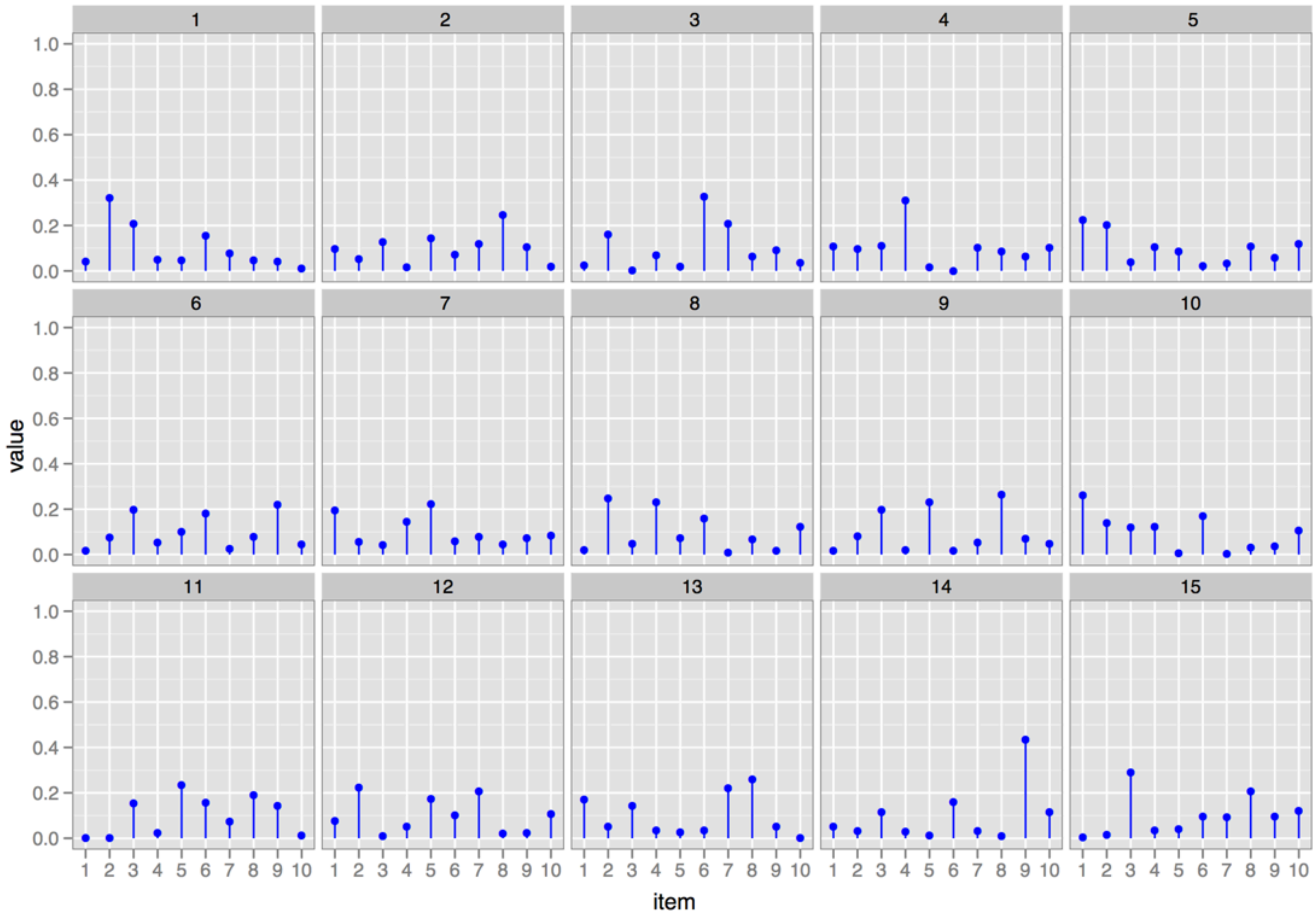
[Blei, LDA tutorial]

$\alpha = 100$



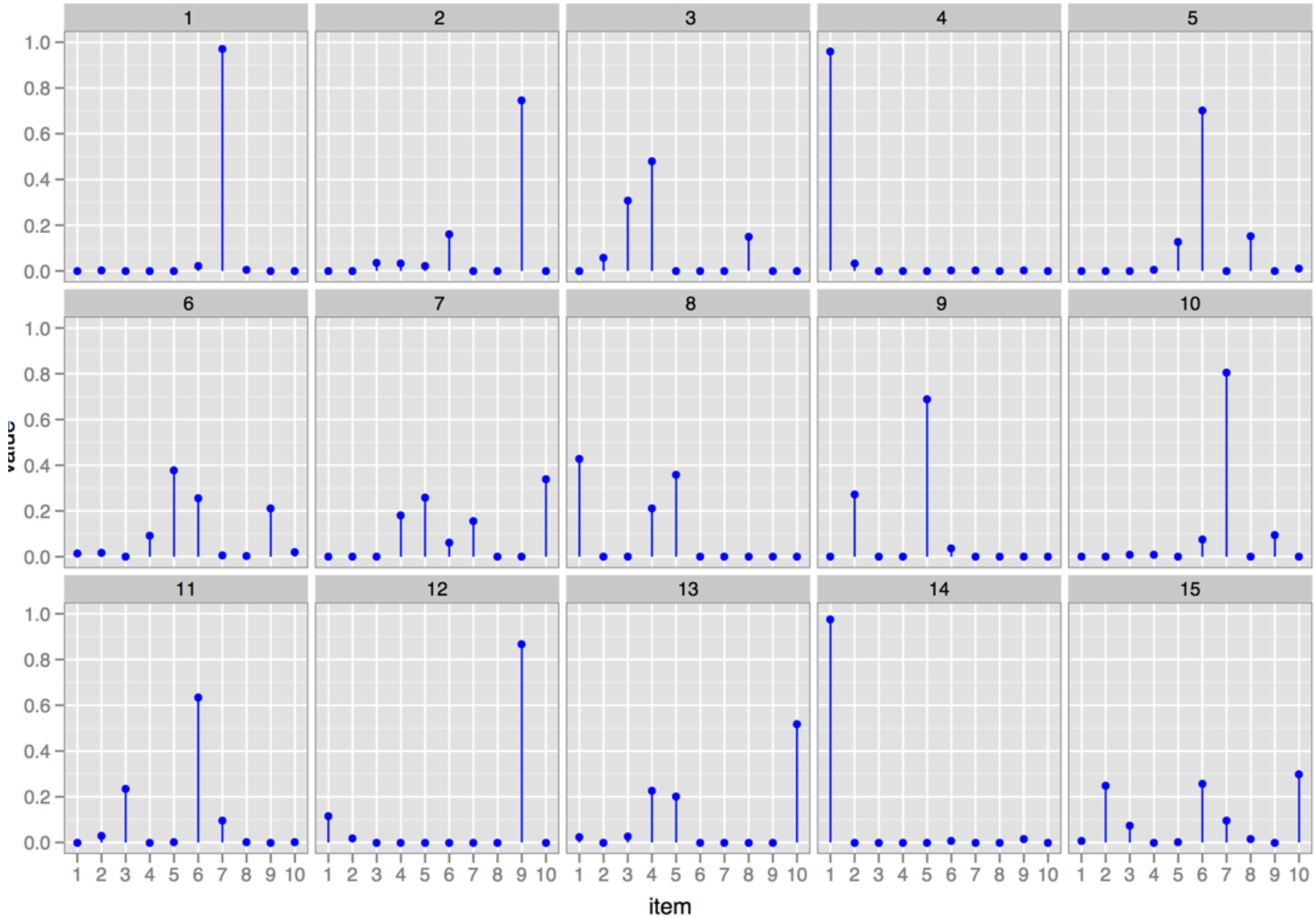
[Blei, LDA tutorial]

$\alpha = 1$



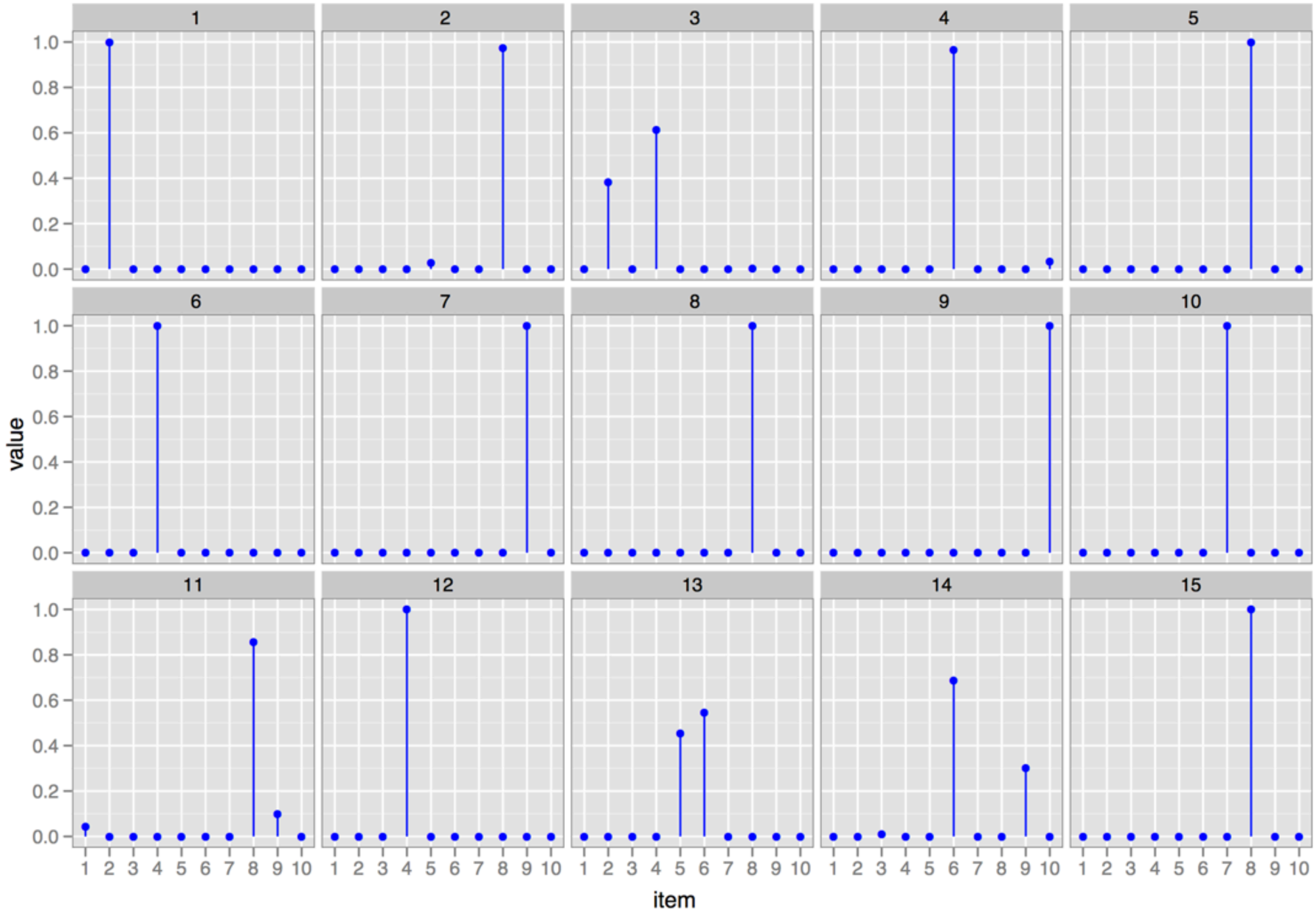
[Blei, LDA tutorial]

α = .1



[Blei, LDA tutorial]

α = .01



[Blei, LDA tutorial]

Dirichlet Distribution

- Is a distribution over the simplex, i.e. positive vectors that sum to 1:

$$P(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}$$

- Conjugate to the multinomial distribution

$$P(\theta|\alpha, x) = \frac{\Gamma(\sum_i x_i + \alpha_i)}{\prod_i \Gamma(x_i + \alpha_i)} \prod_i \theta_i^{x_i + \alpha_i - 1}$$

Dirichlet Distribution

- Prior

$$P(\theta|\alpha) \sim Dir(\alpha_1, ..., \alpha_k)$$

$$E[\theta_i|\alpha] = \frac{\alpha_i}{\sum_i \alpha_i}$$

- Posterior

$$P(\theta|x, \alpha) \sim Dir(x_1 + \alpha_1, ..., x_k + \alpha_k)$$

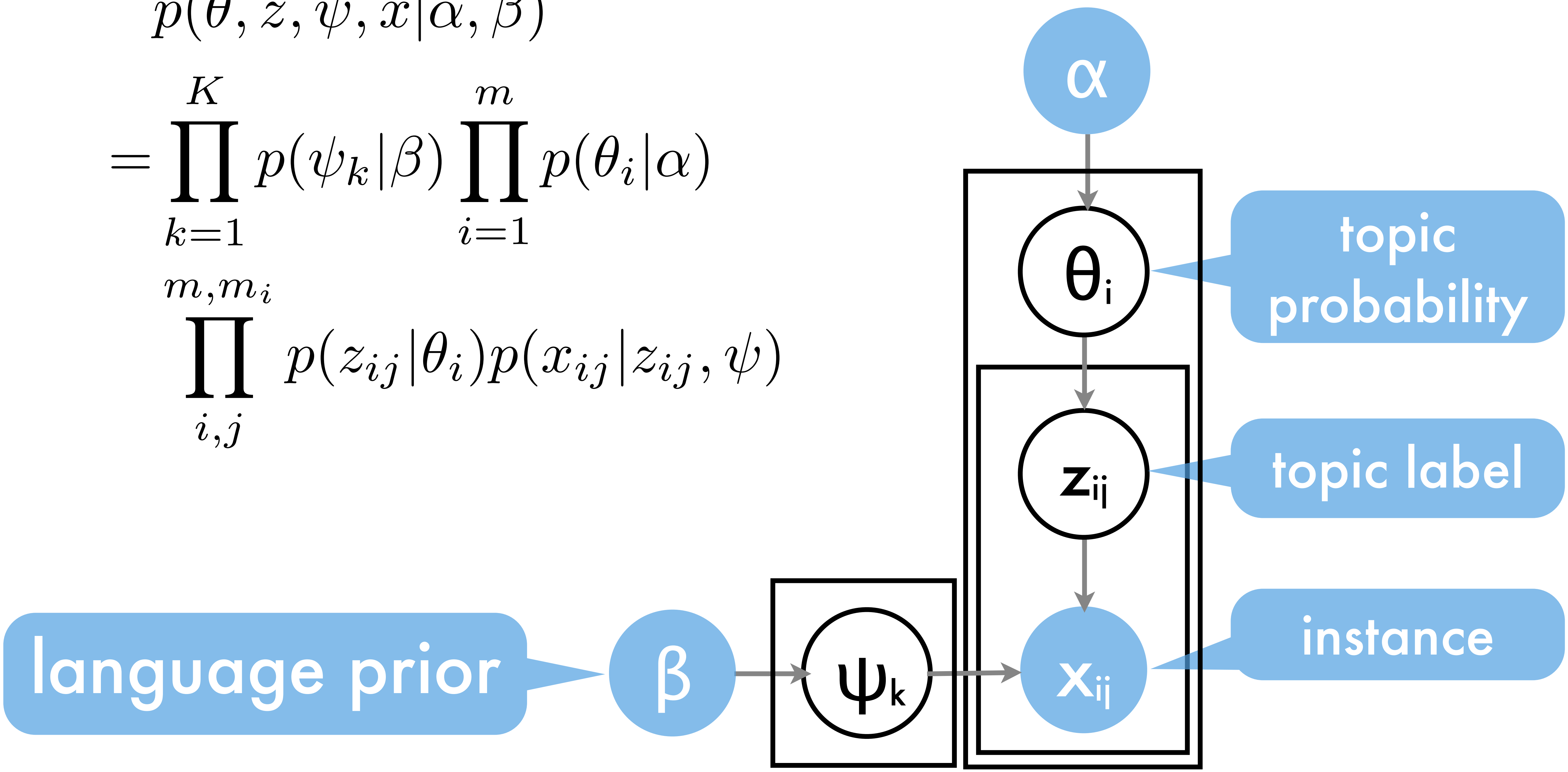
$$E[\theta_i|x, \alpha] = \frac{x_i + \alpha_i}{\sum_i x_i + \alpha_i}$$

Joint Probability Distribution

$$p(\theta, z, \psi, x | \alpha, \beta)$$

$$= \prod_{k=1}^{K} p(\psi_k|\beta) \prod_{i=1}^{m} p(\theta_i|\alpha)$$

$$\prod_{i,j}^{m,m_i} p(z_{ij}|\theta_i) p(x_{ij}|z_{ij}, \psi)$$



Joint Probability Distribution

sample Ψ independently

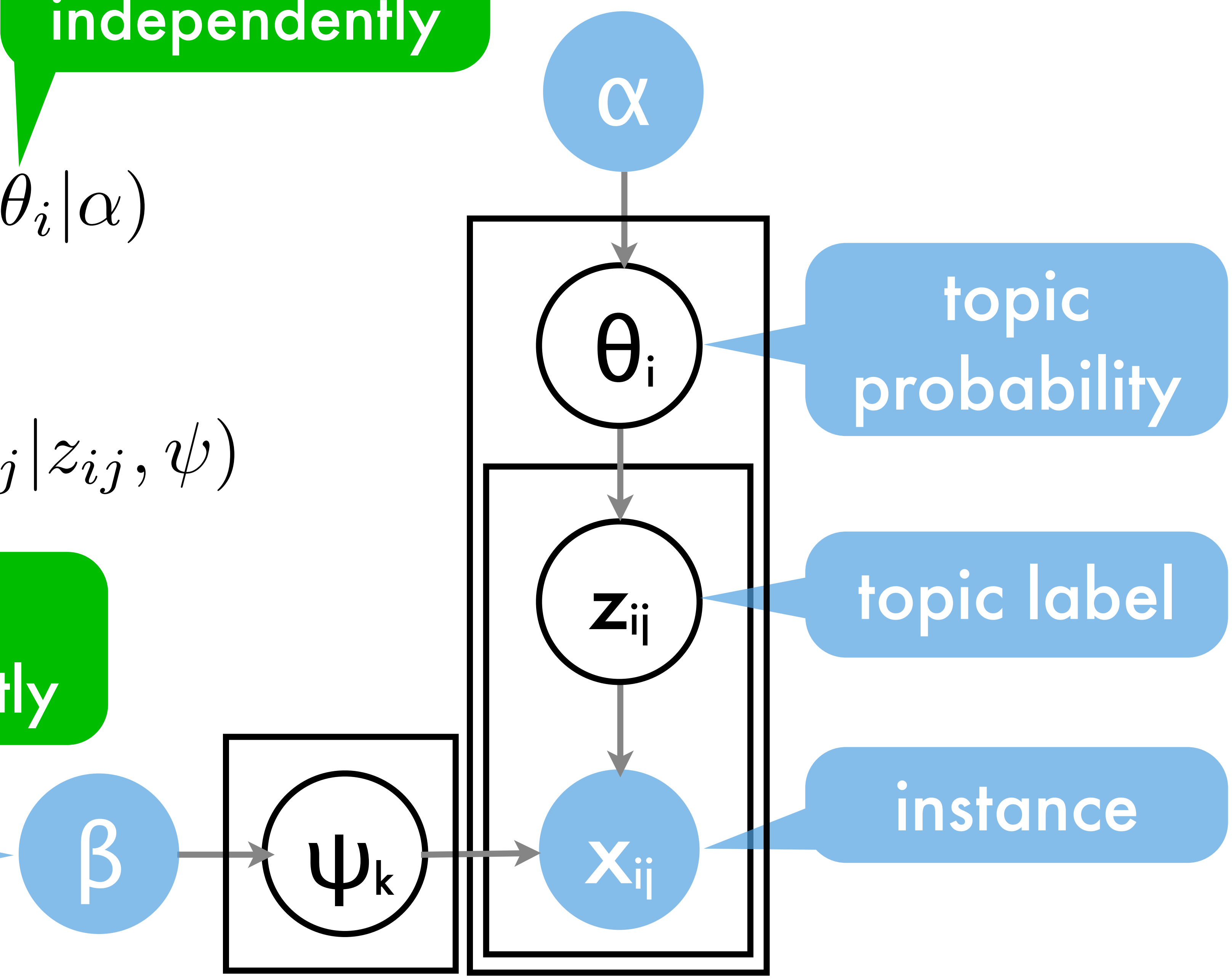
sample θ independently

$$p(\theta, z, \psi, x|\alpha, \beta)$$

$$= \prod_{k=1}^{K} p(\psi_k|\beta) \prod_{i=1}^{m} p(\theta_i|\alpha)$$

$$\prod_{i,j}^{m,m_i} p(z_{ij}|\theta_i) p(x_{ij}|z_{ij}, \psi)$$

sample z independently



Joint Probability Distribution

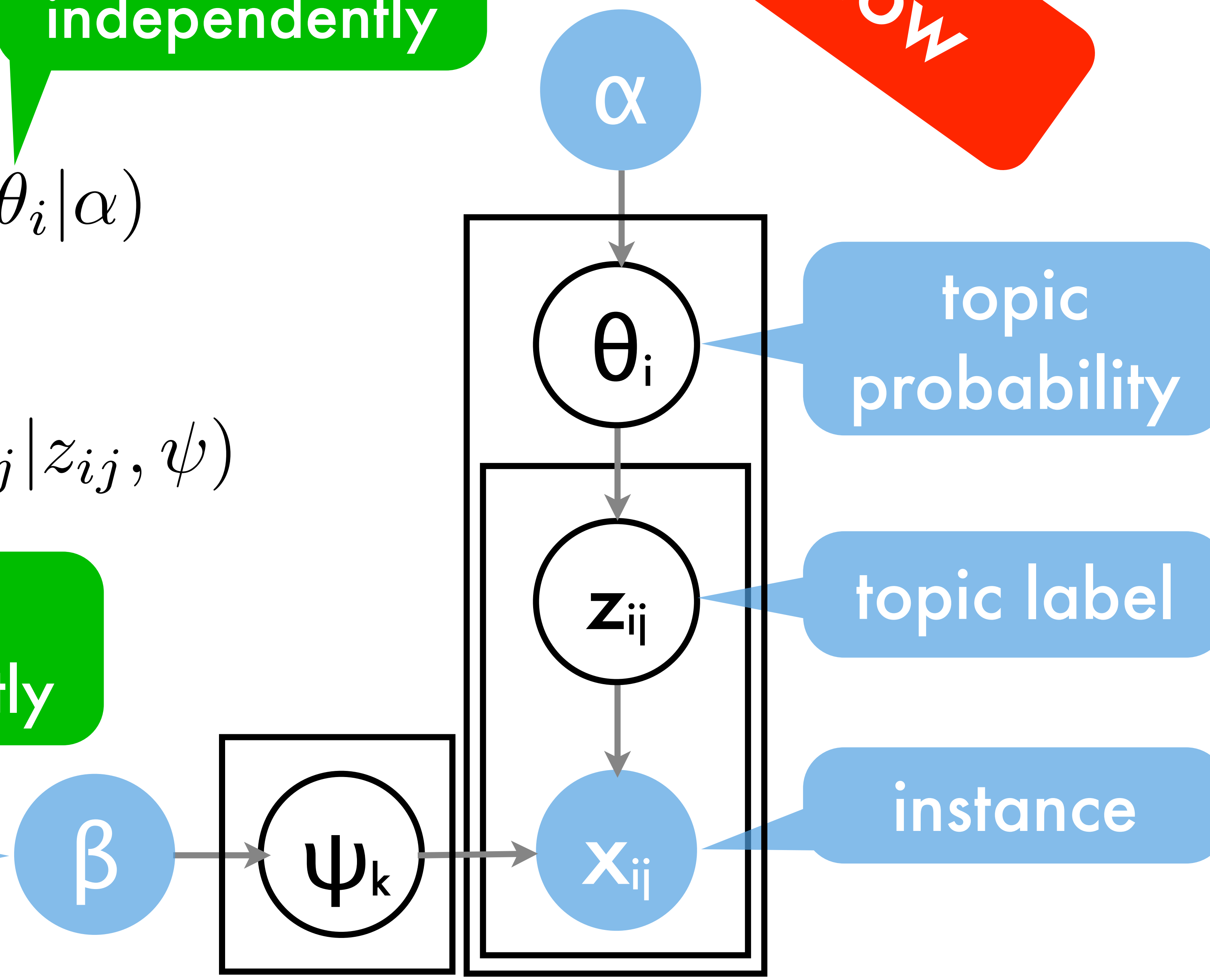
sample Ψ independently

sample θ independently

slow

$$
\begin{aligned}
&p(\theta, z, \psi, x \mid \alpha, \beta) \\
&= \prod_{k=1}^{K} p(\psi_k \mid \beta) \prod_{i=1}^{m} p(\theta_i \mid \alpha) \\
&\prod_{i,j}^{m, m_i} p(z_{ij} \mid \theta_i) p(x_{ij} \mid z_{ij}, \psi)
\end{aligned}
$$

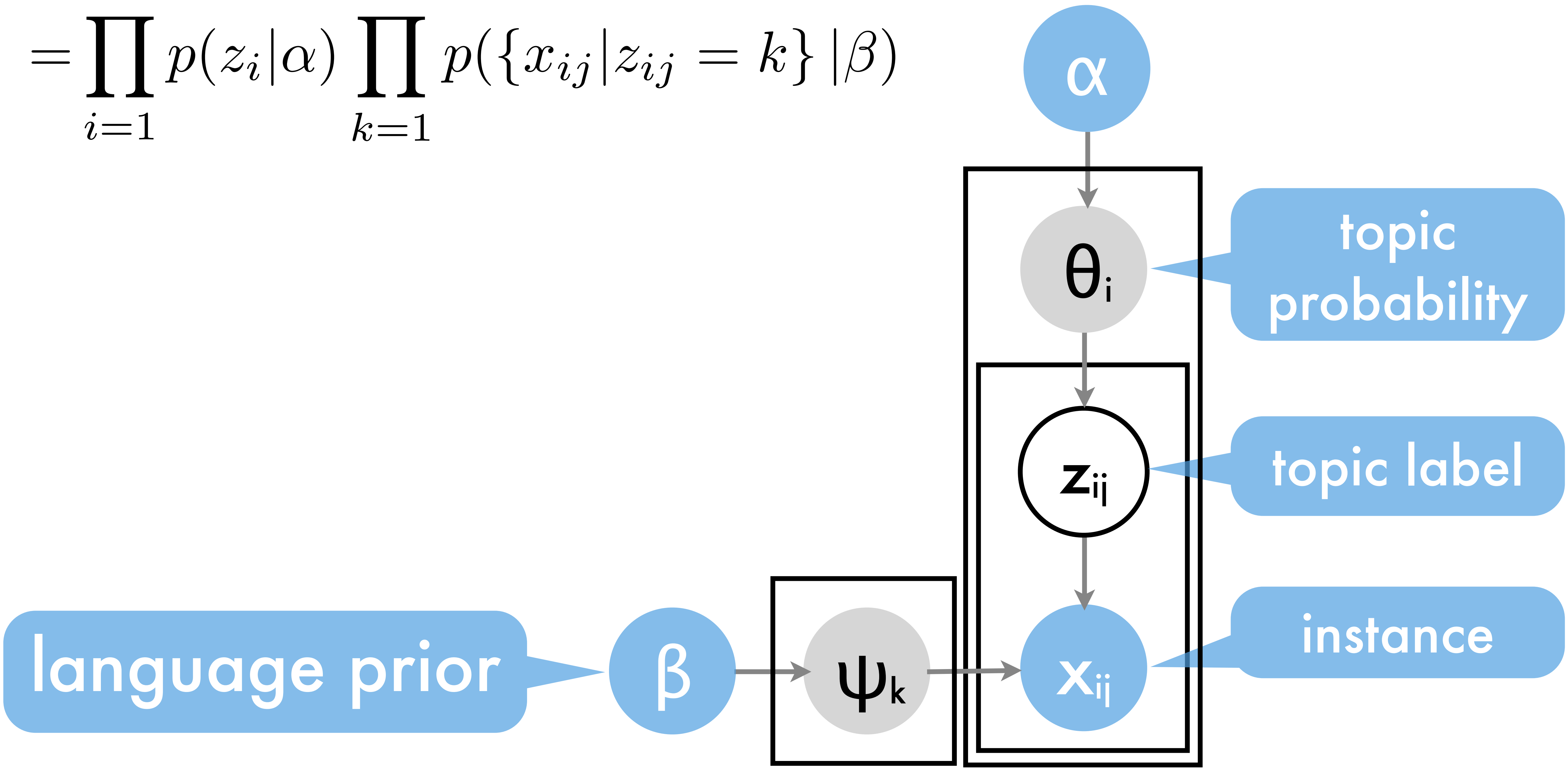
sample z independently



Collapsed Sampler

$$p(z, x|\alpha, \beta)$$

$$= \prod_{i=1}^{m} p(z_i|\alpha) \prod_{k=1}^{k} p(\{x_{ij}|z_{ij} = k\} |\beta)$$

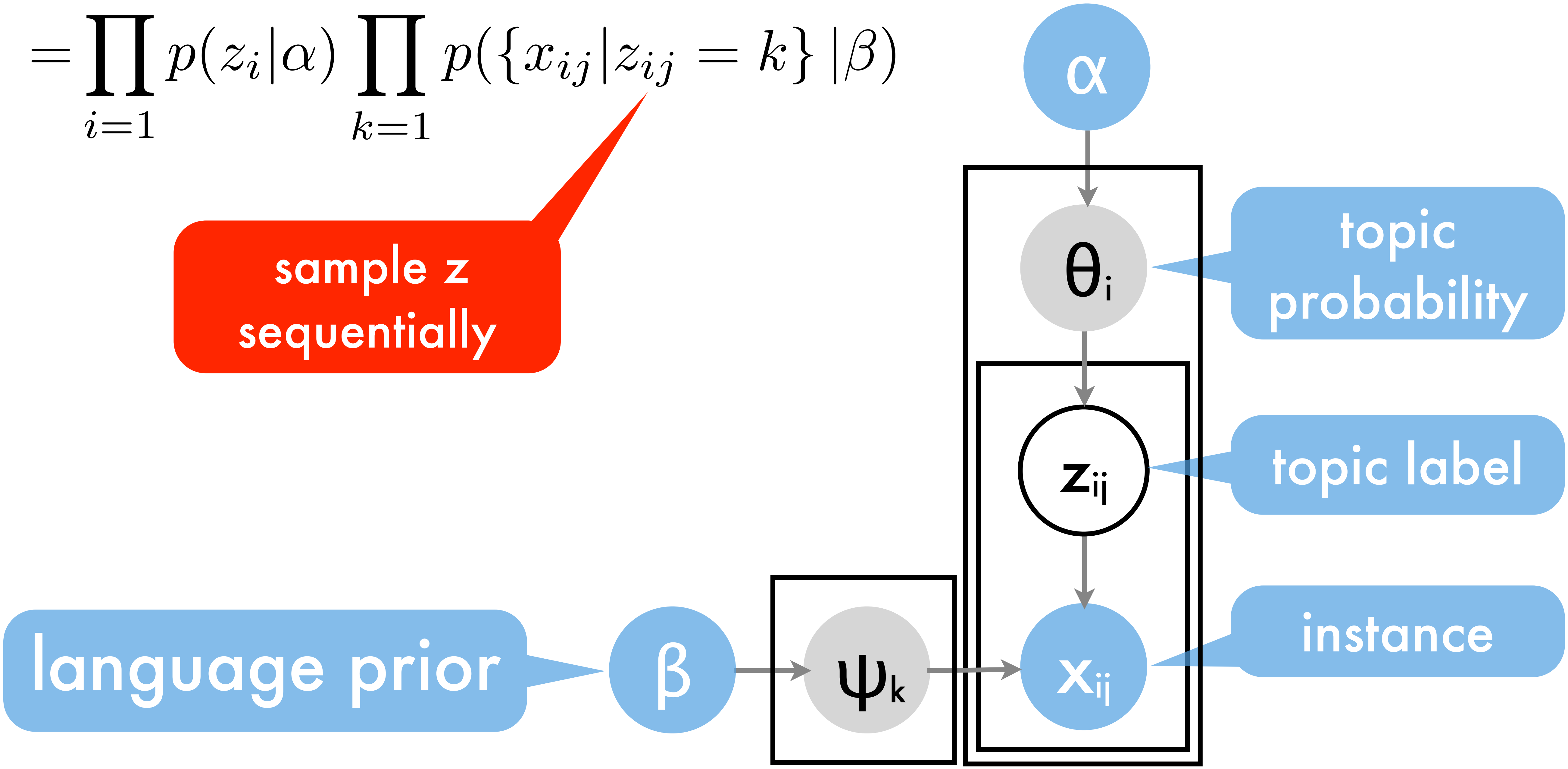


Collapsed Sampler

$$p(z, x|\alpha, \beta)$$

$$= \prod_{i=1}^{m} p(z_i|\alpha) \prod_{k=1}^{k} p(\{x_{ij}|z_{ij} = k\} |\beta)$$

sample z sequentially



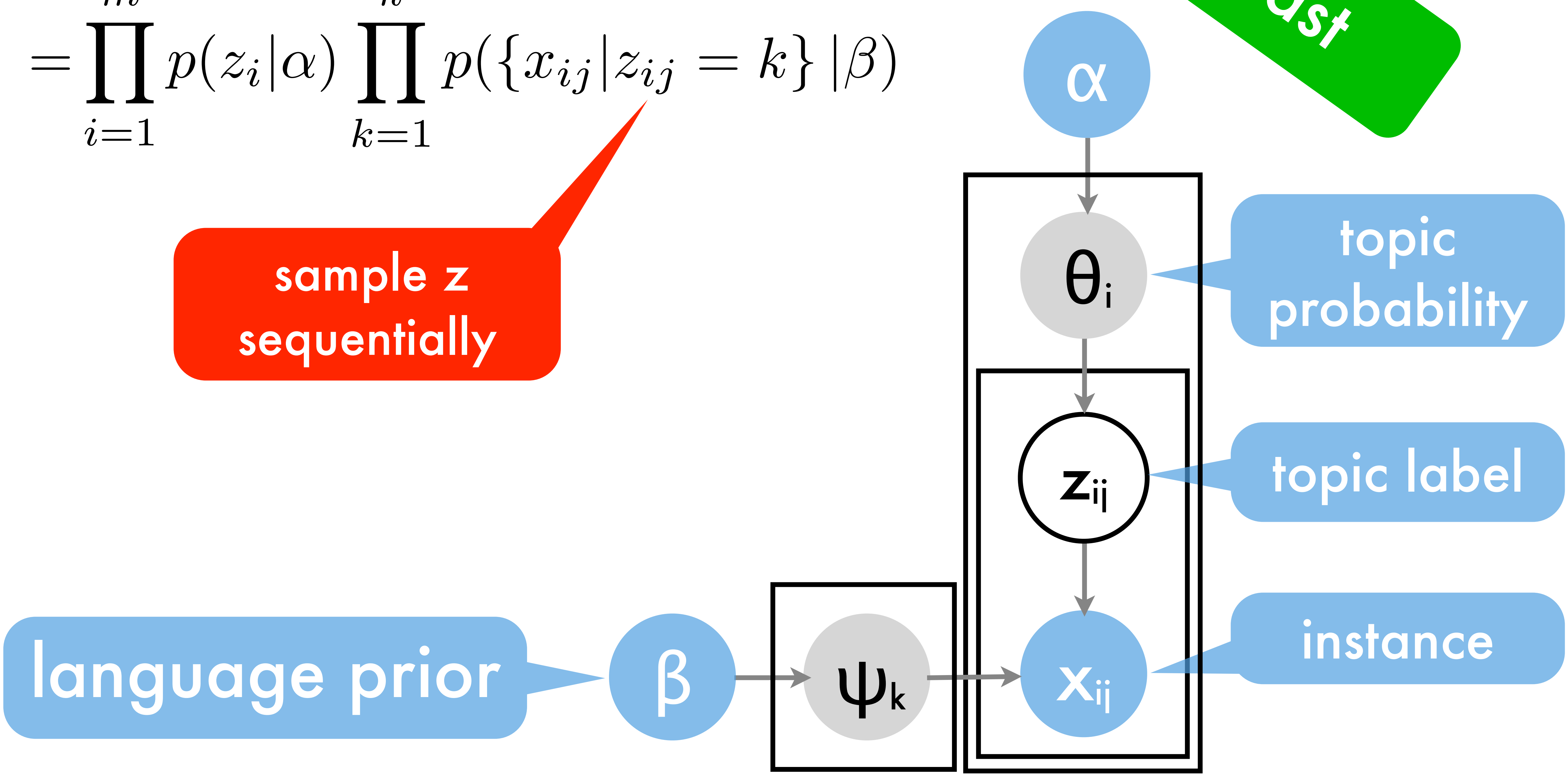
Collapsed Sampler

$$p(z, x|\alpha, \beta)$$

$$= \prod_{i=1}^{m} p(z_i|\alpha) \prod_{k=1}^{k} p(\{x_{ij}|z_{ij} = k\}|\beta)$$

sample z sequentially

fast



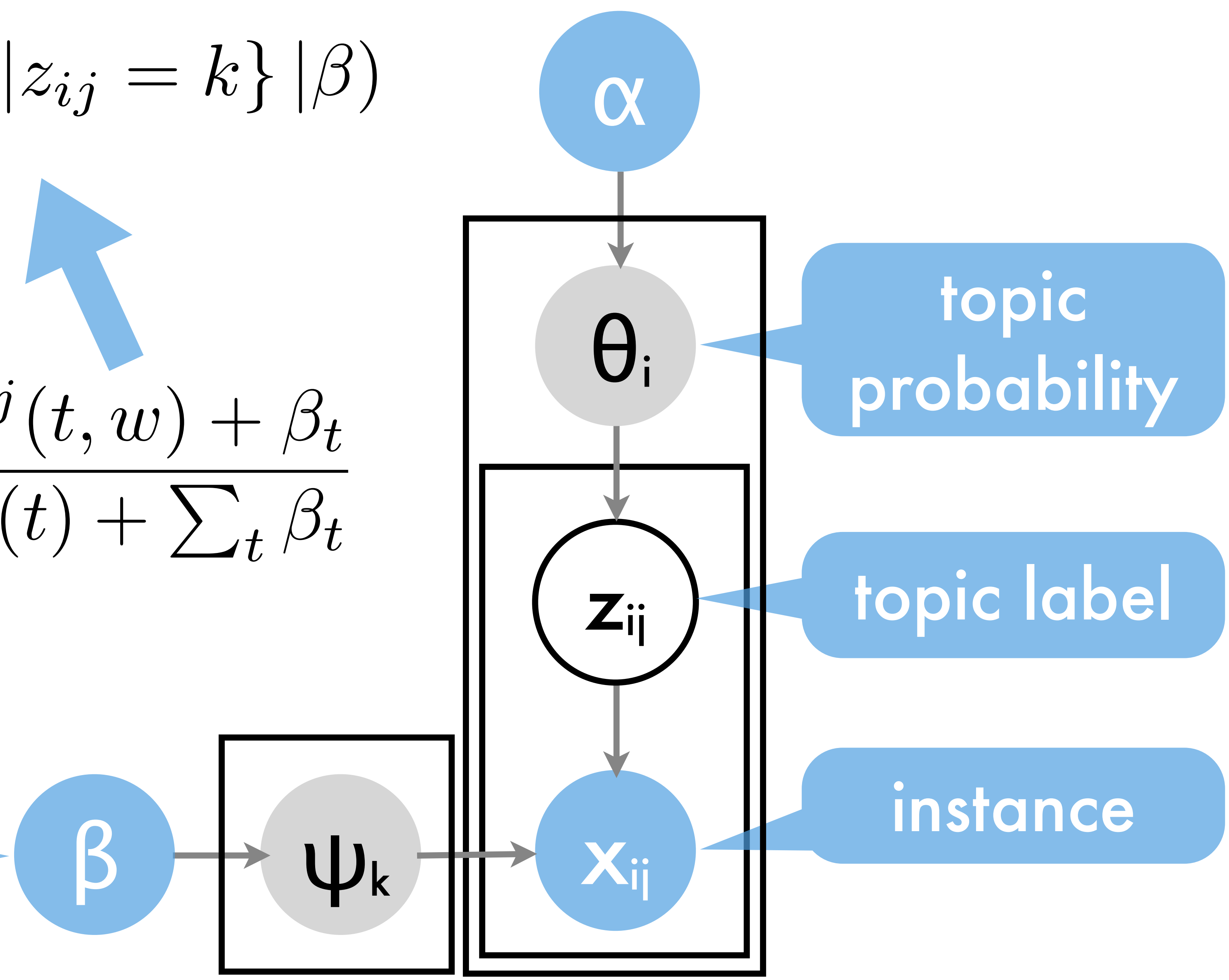
Collapsed Sampler

Griffiths & Steyvers, 2005

$$p(z, x|\alpha, \beta)$$

$$= \prod_{i=1}^{m} p(z_i|\alpha) \prod_{k=1}^{k} p(\{x_{ij}|z_{ij} = k\}\,|\beta)$$

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



Collapsed Sampler

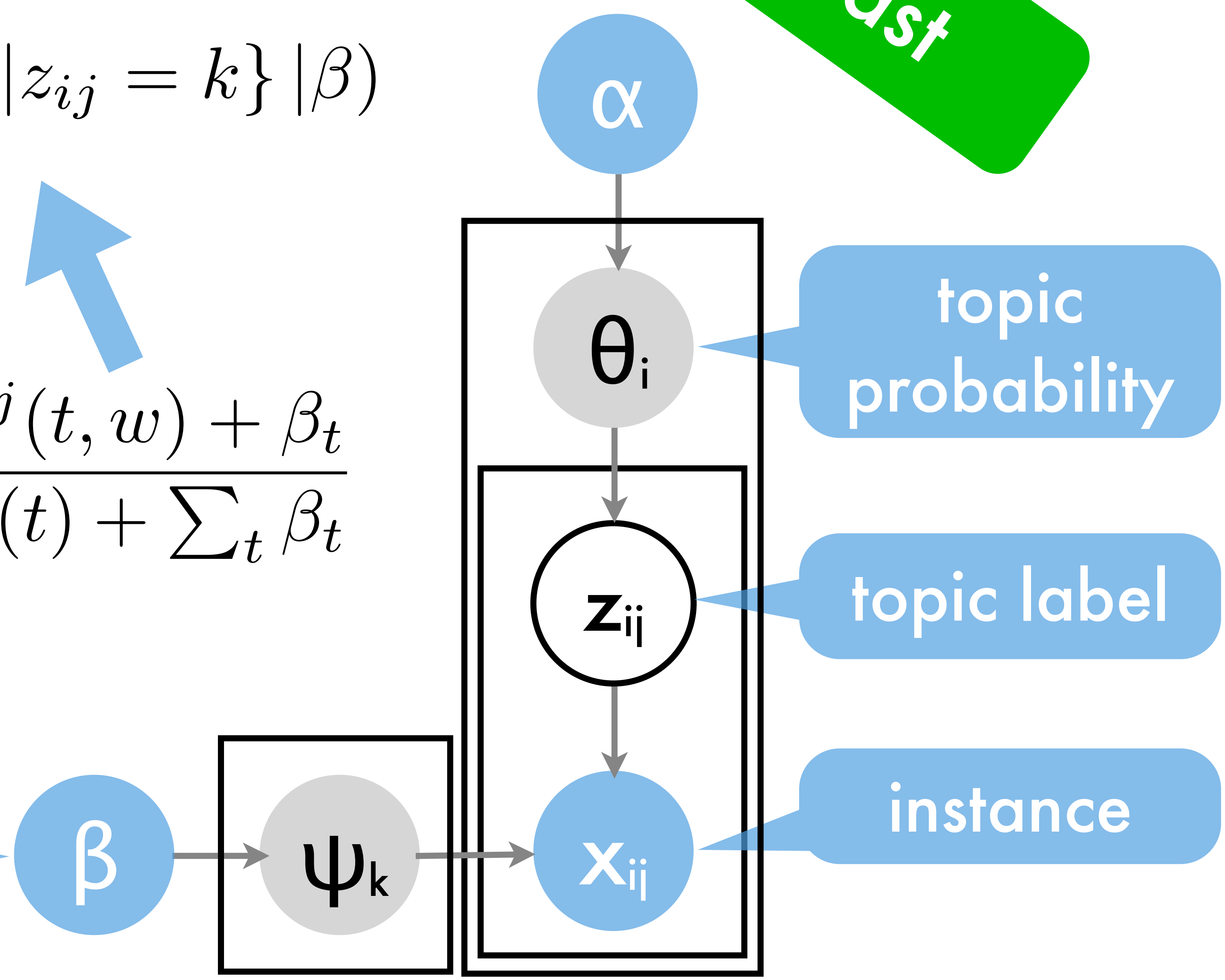
Griffiths & Steyvers, 2005

fast

$$p(z, x|\alpha, \beta)$$

$$= \prod_{i=1}^{m} p(z_i|\alpha) \prod_{k=1}^{k} p(\{x_{ij}|z_{ij} = k\}|\beta)$$

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



Derivations (was on the board)

total rule of probability

Chain rule

conditional independence

$$
\begin{aligned}
p(z_{ij} = t | z^{-ij}, \alpha) &= \int_\theta p(\theta, z_{ij} = t | z^{-ij}, \alpha) d\theta \\
&= \int_\theta p(\theta | z^{-ij}, \alpha) p(z_{ij} = t | \theta, z^{-ij}, \alpha) d\theta \\
&= \int_\theta p(\theta | z^{-ij}, \alpha) p(z_{ij} = t | \theta) d\theta \\
&= \int_\theta p(\theta | z^{-ij}, \alpha) \theta_t d\theta \\
&= E_{p(\theta | z^{-ij}, \alpha)}[\theta_t] \\
&= \text{mean of the posterior of } \theta \text{ given other topic assignments}
\end{aligned}
$$

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

Derivation for the second factor follows similarly

Sequential Algorithm (Gibbs sampler)

- For 1000 iterations do
  - For each document do
    - For each word in the document do
      - Resample topic for the word
      - Update local (document, topic) table
      - Update CPU local (word, topic) table
      - Update global (word, topic) table

Sequential Algorithm (Gibbs sampler)

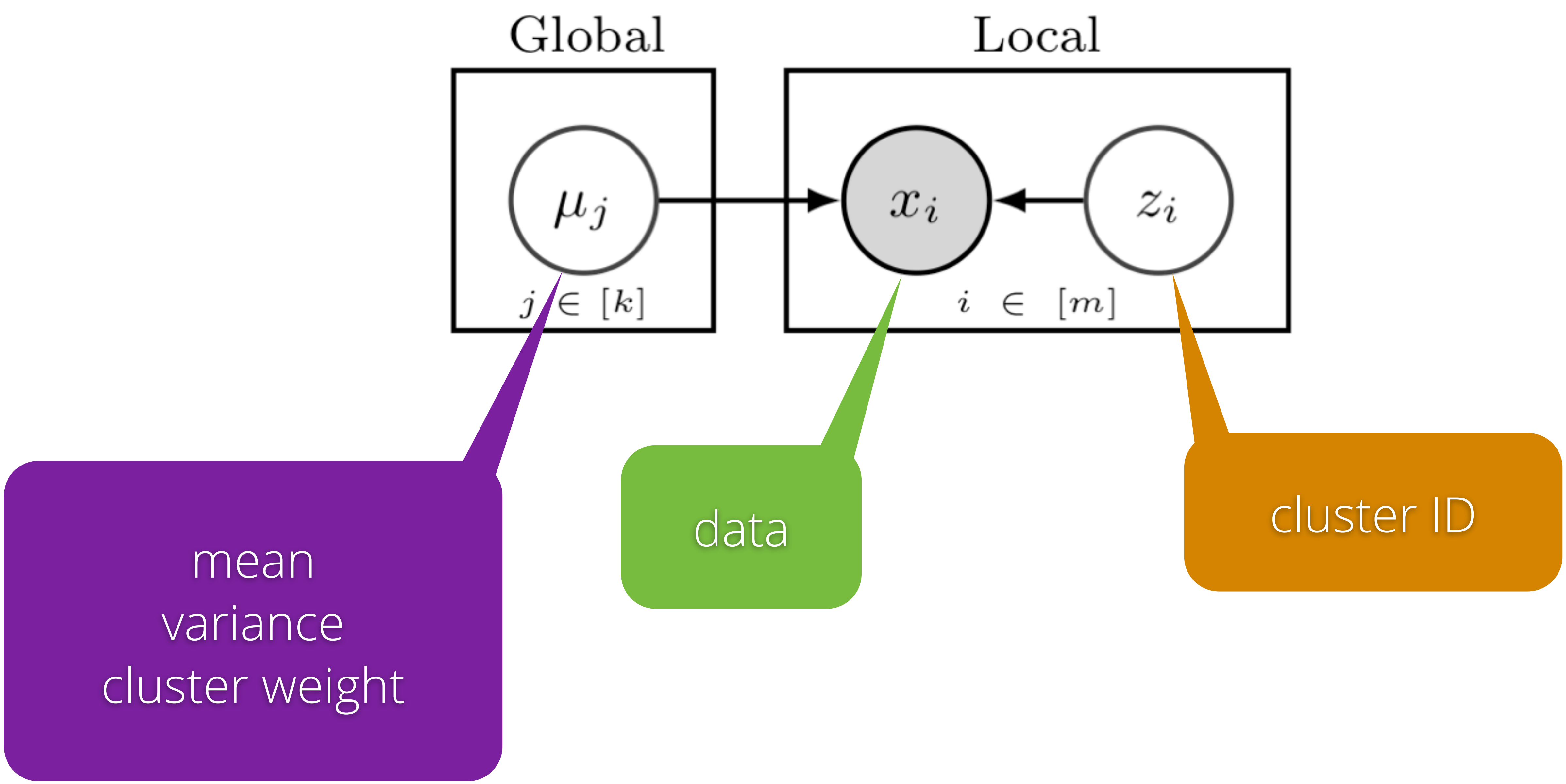
- For 1000 iterations do
  - For each document do
    - For each word in the document do
      - Resample topic for the word
      - Update local (document, topic) table
      - Update CPU local (word, topic) table
      - Update global (word, topic) table

this kills parallelism

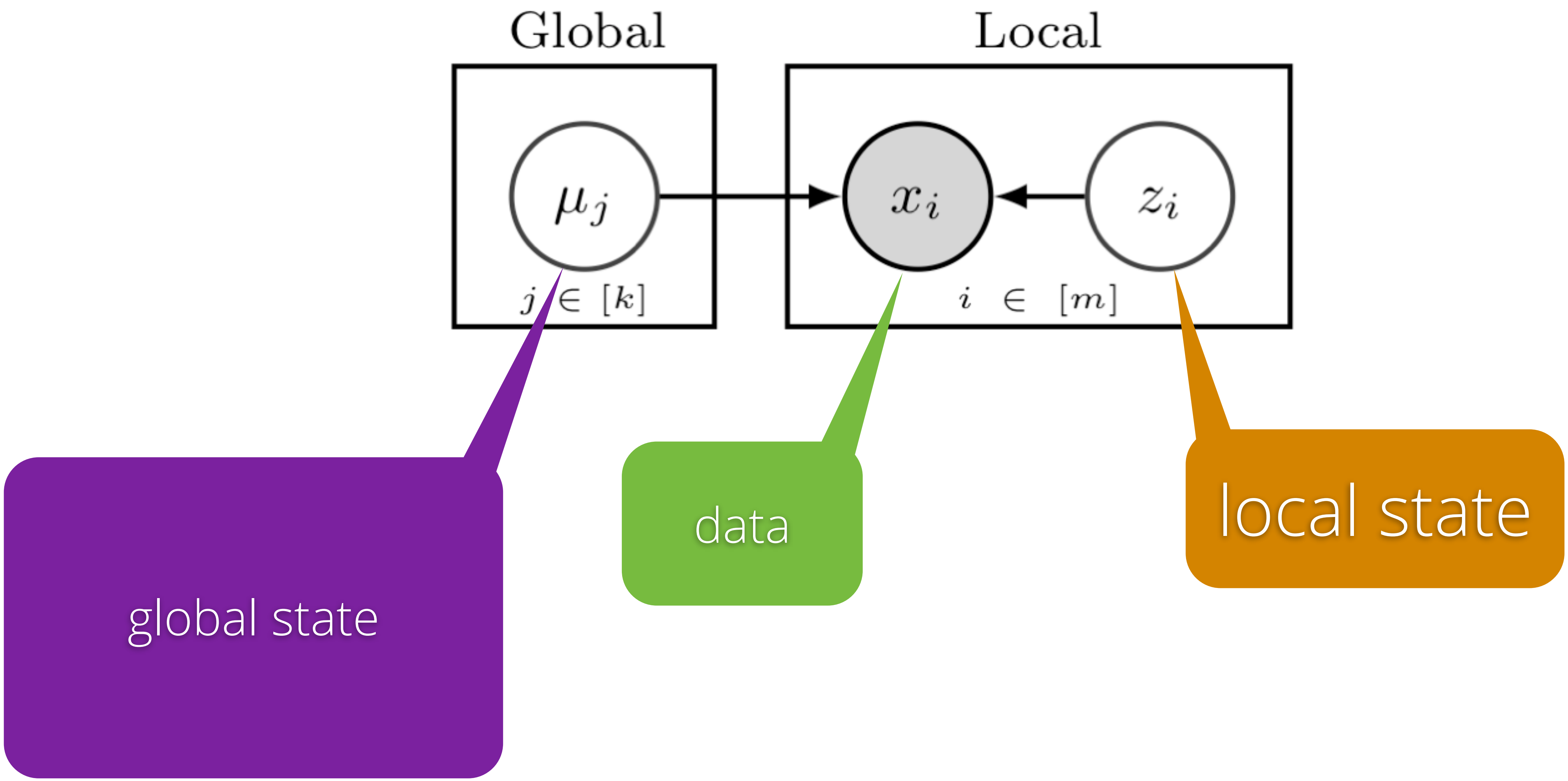
Design Principles

Scaling Problems

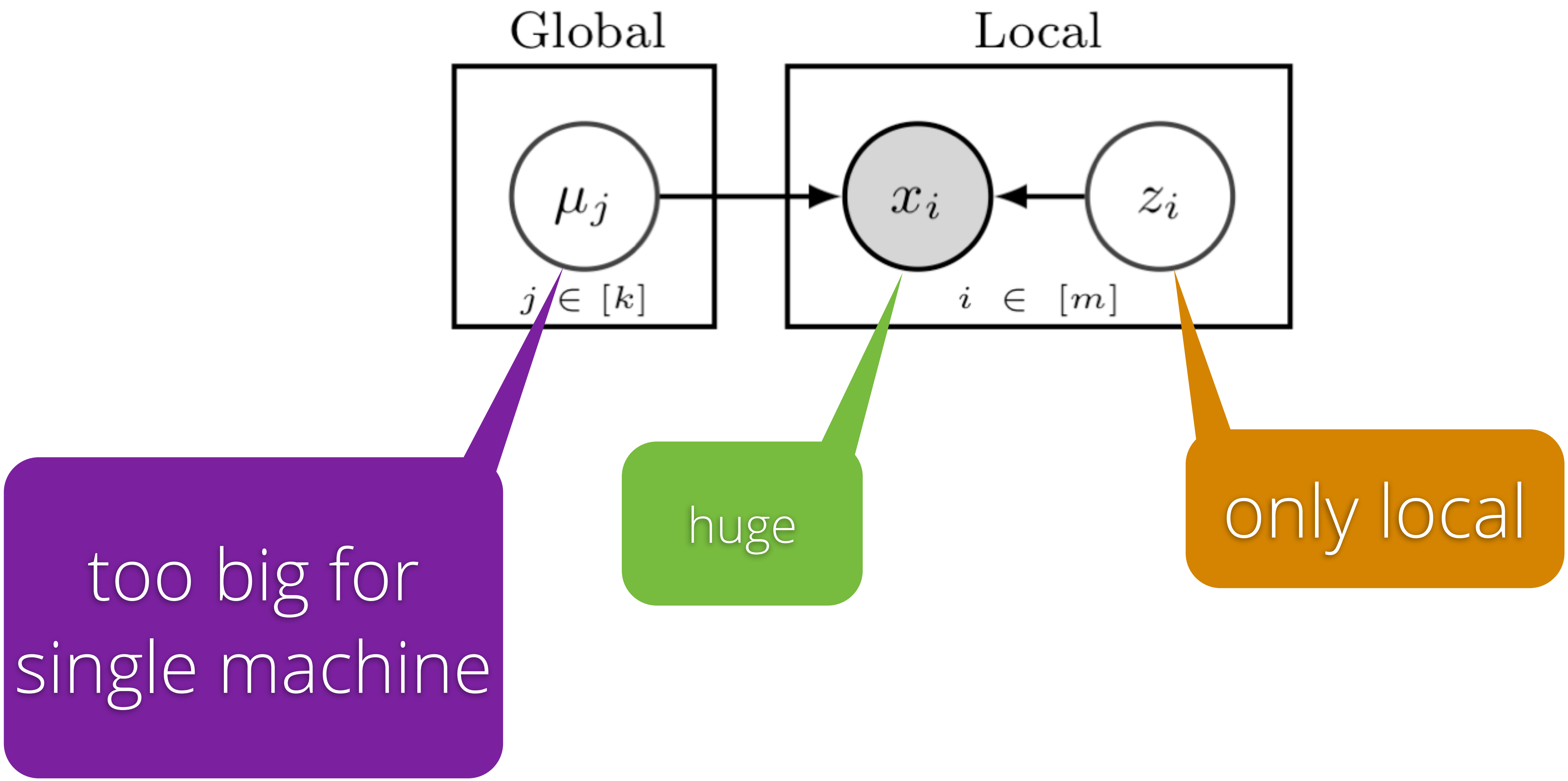
3 Problems


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

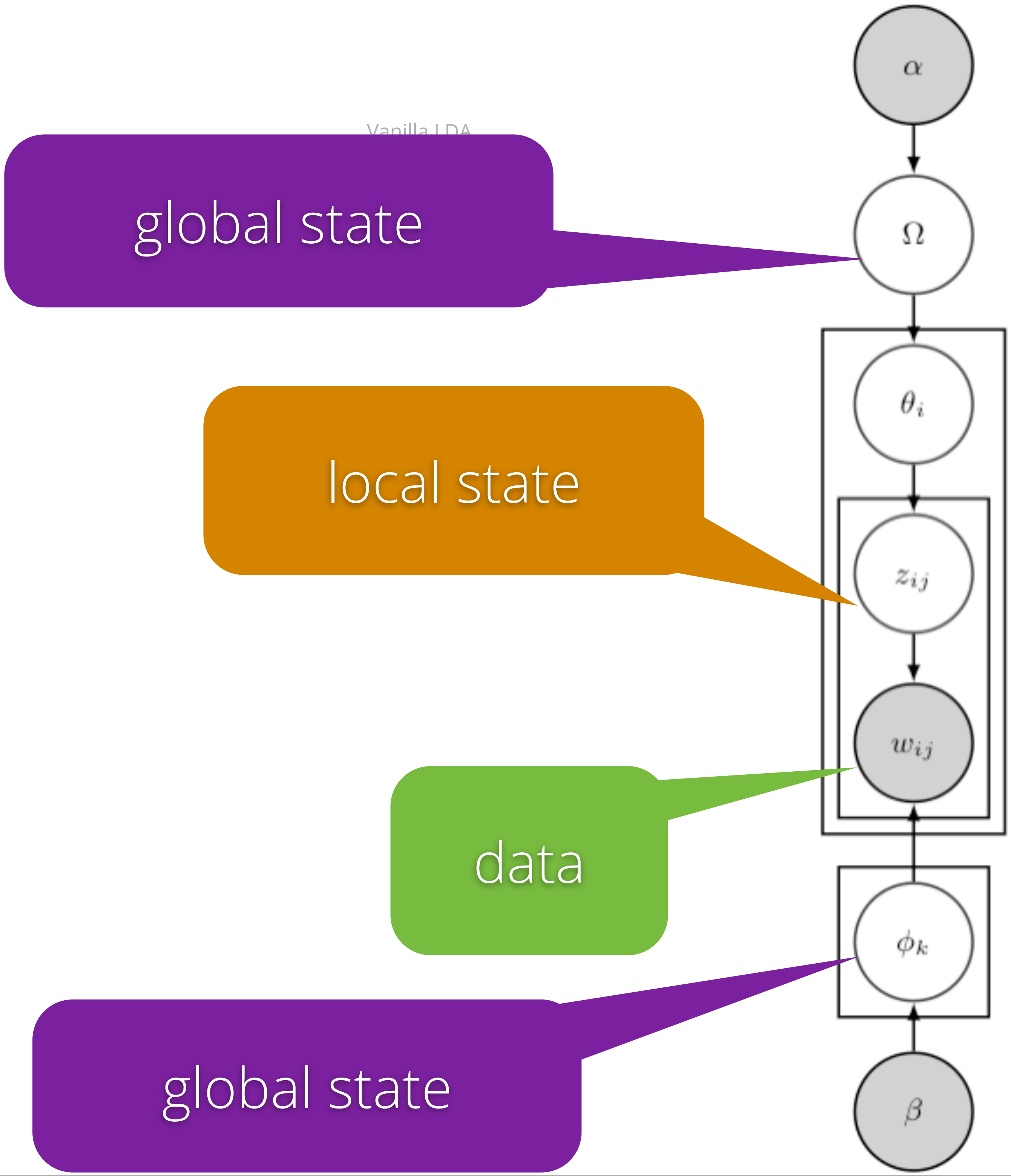
3 Problems


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

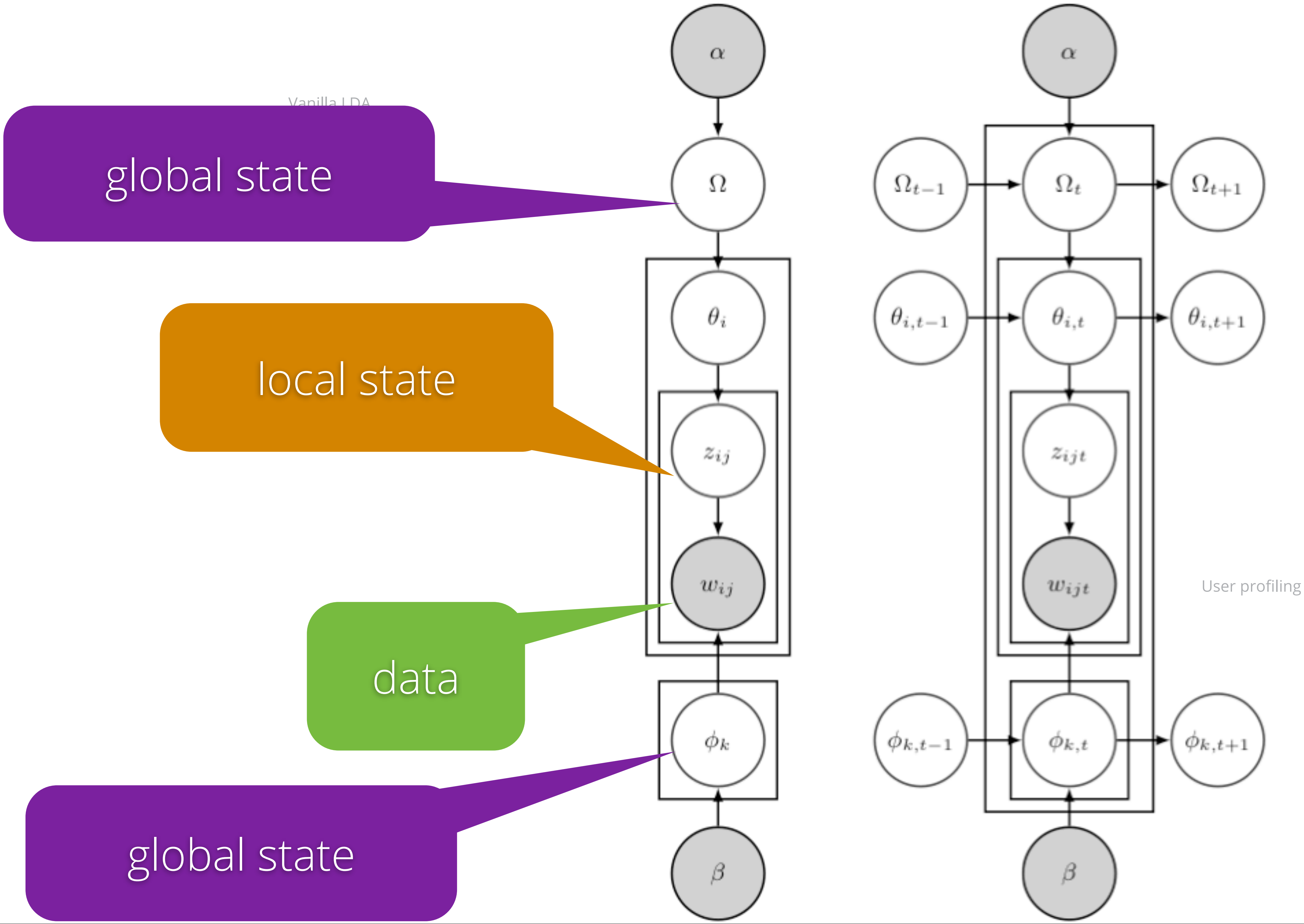
3 Problems


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

3 Problems



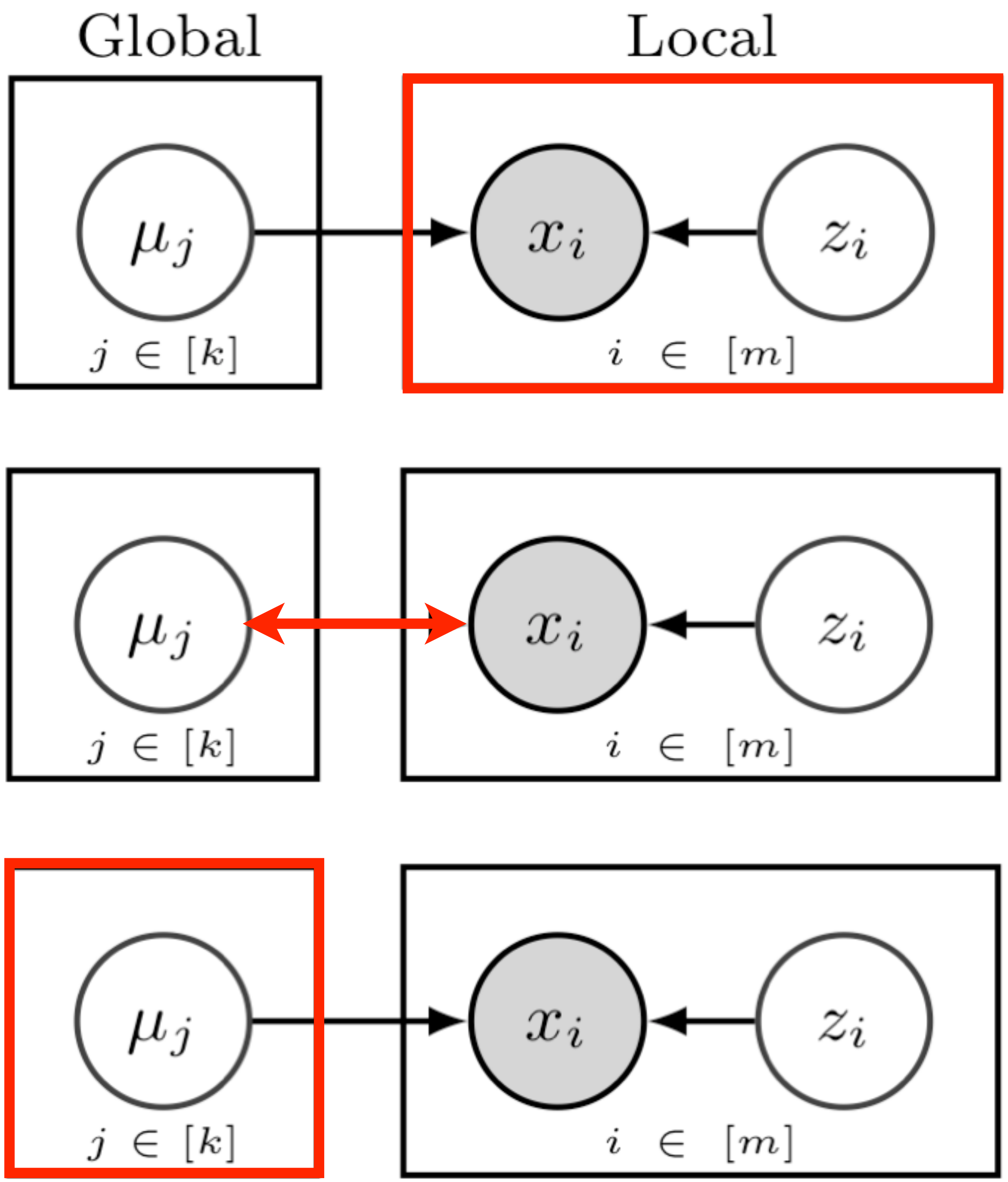
3 Problems


global state
local state
data
global state
α
Ω
θ_i
z_{ij}
w_{ij}
ϕ_k
β
Ω_{t-1}
Ω_t
Ω_{t+1}
$\theta_{i,t-1}$
$\theta_{i,t}$
$\theta_{i,t+1}$
z_{ijt}
w_{ijt}
$\phi_{k,t-1}$
$\phi_{k,t}$
$\phi_{k,t+1}$
User profiling

3 Problems

local state
is too large

global state
is too large

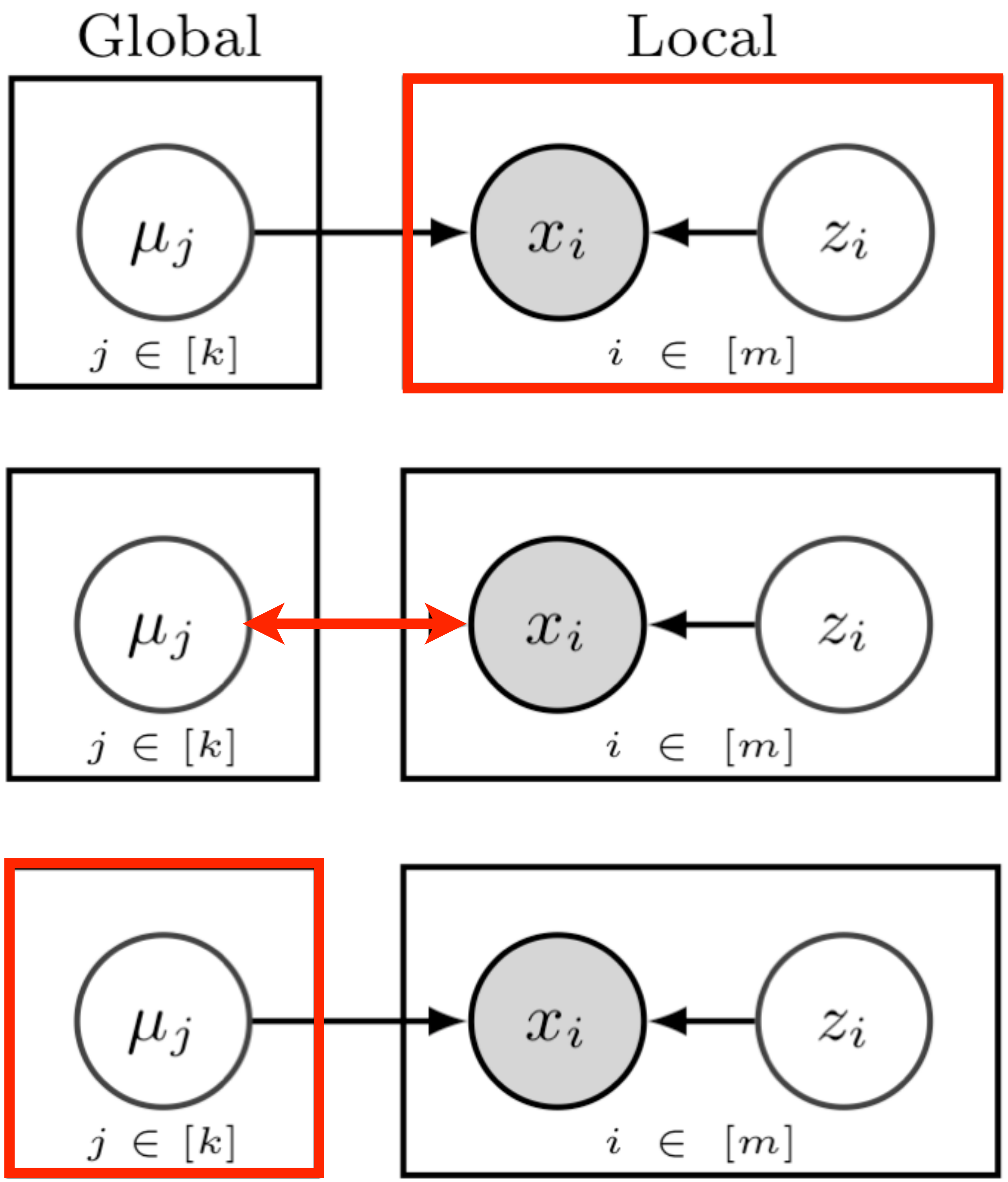


does not fit
into memory

network load
& barriers

does not fit
into memory

3 Problems



local state
is too large

stream local
data from disk

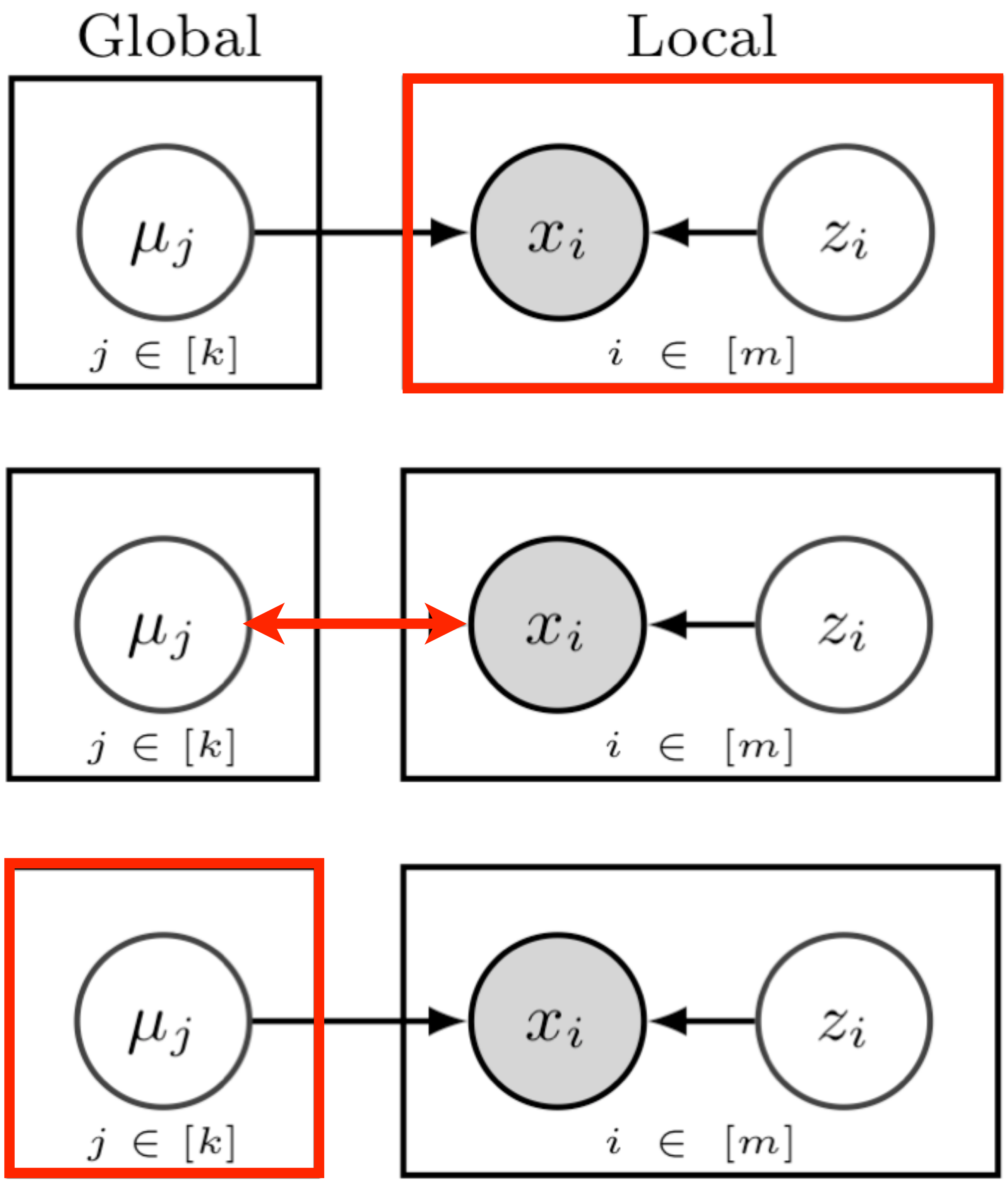
network load
& barriers

global state
is too large

does not fit
into memory

3 Problems

local state
is too large



stream local
data from disk

asynchronous
synchronization

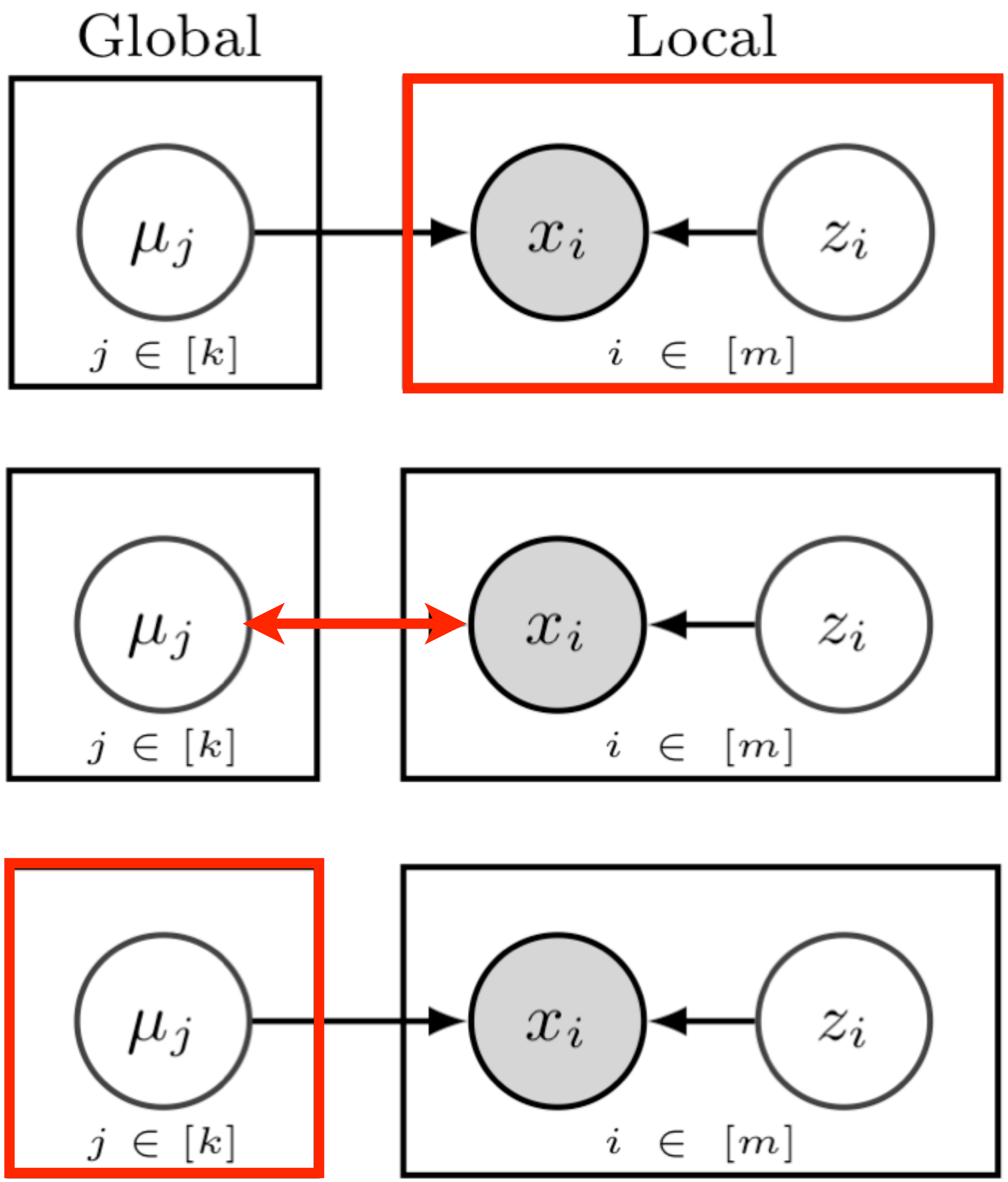
global state
is too large

does not fit
into memory

3 Problems

local state
is too large

global state
is too large



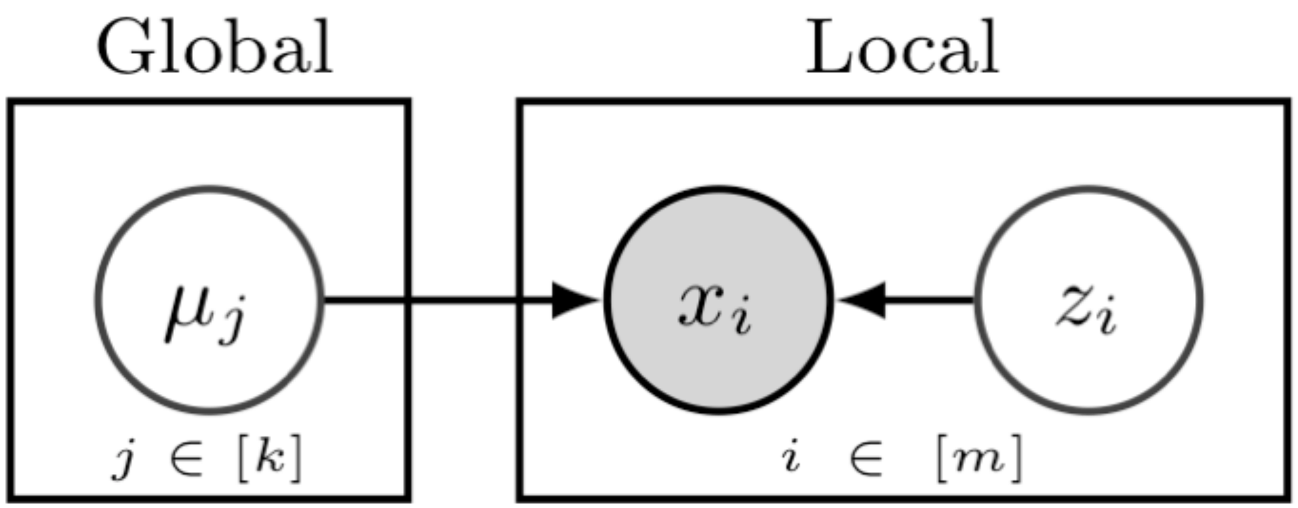
stream local data from disk

asynchronous synchronization

partial view

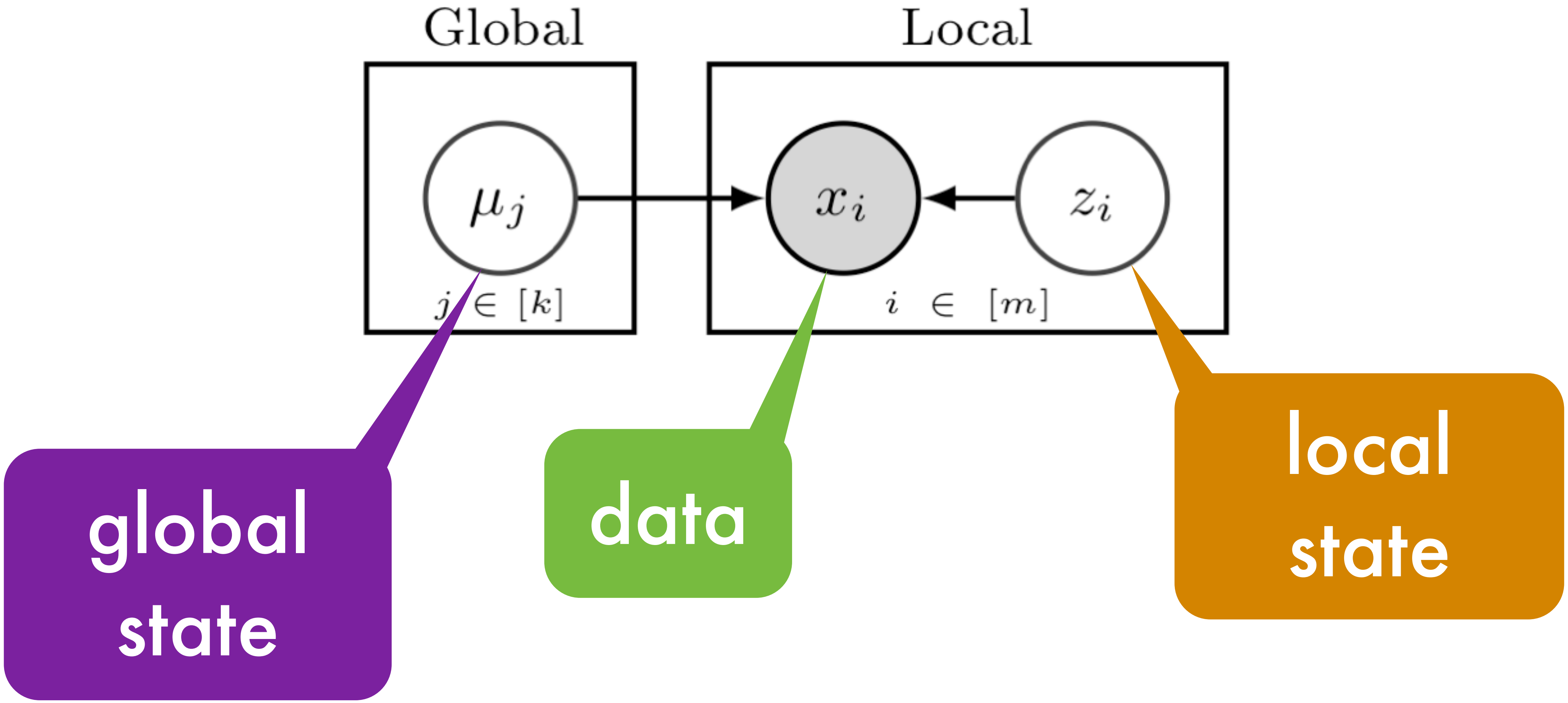
Global state synchronization

Challenges

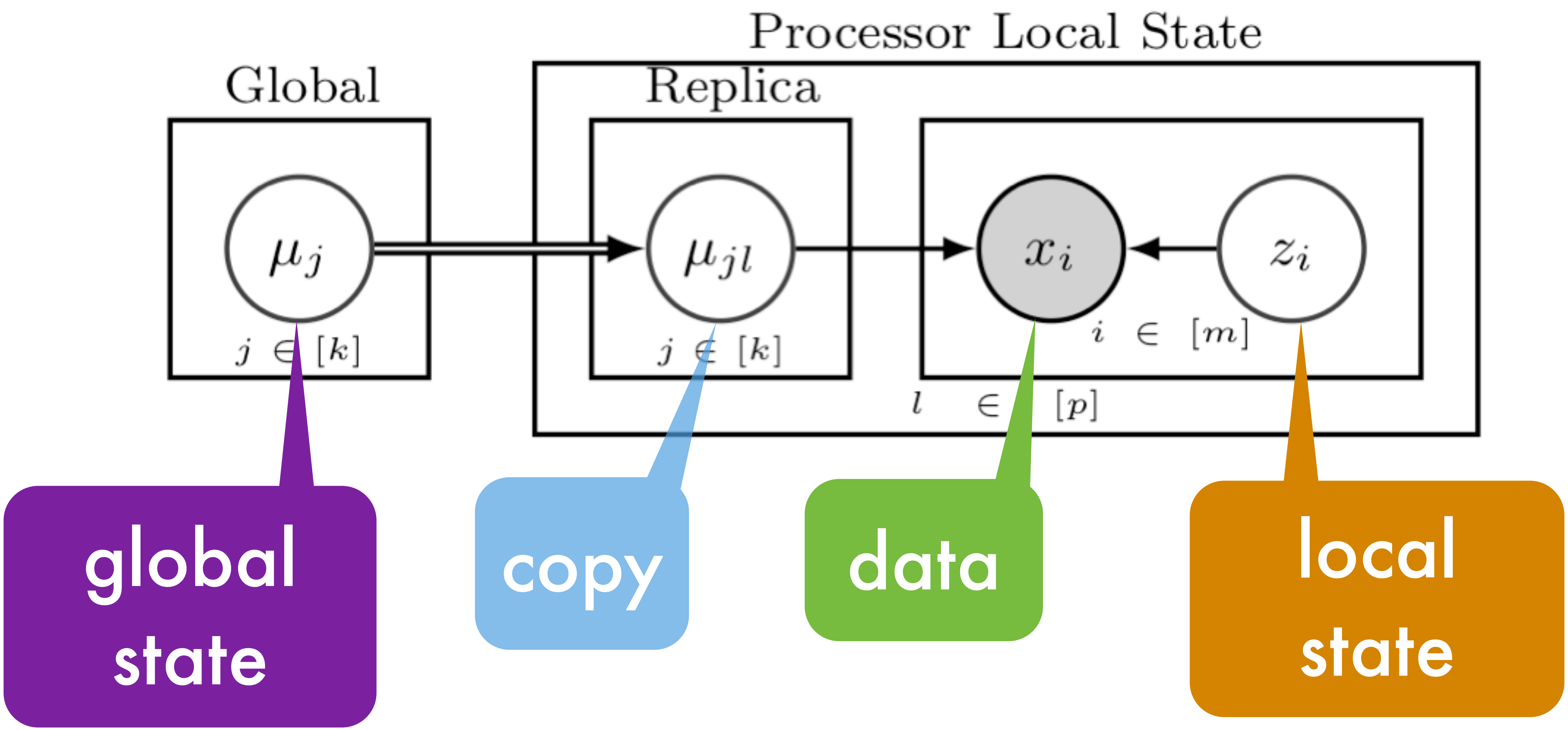


- Distribution (global)
- Synchronization (global)
- Fault tolerance
- Storage (local)

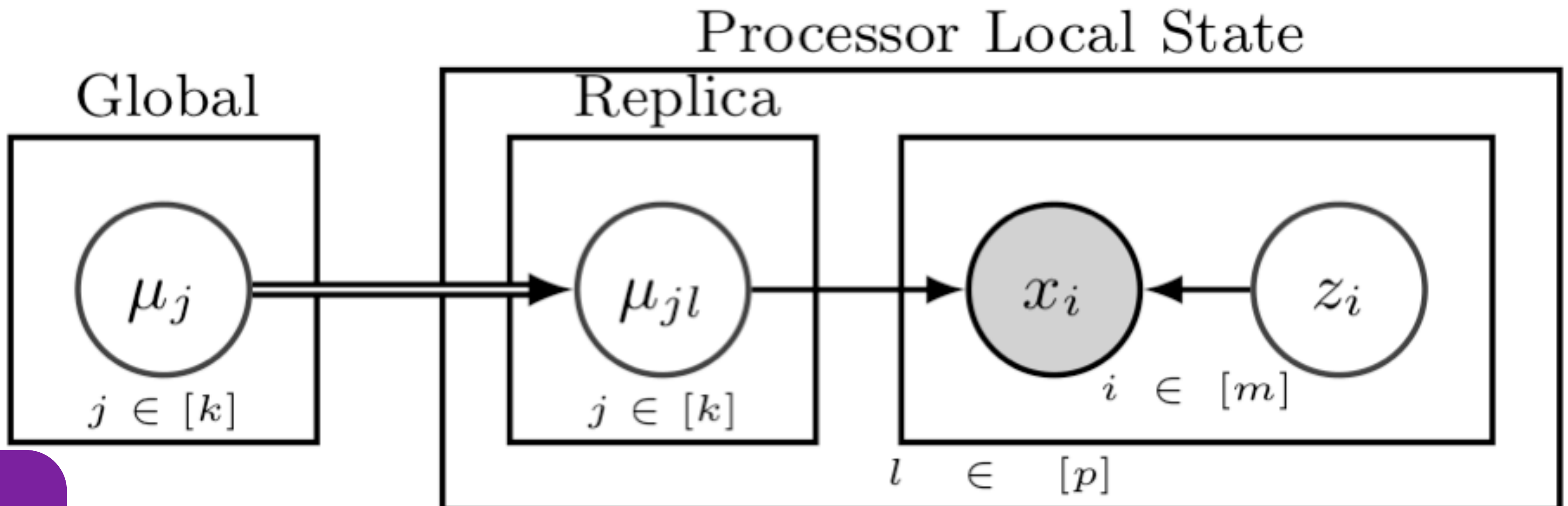
Distribution

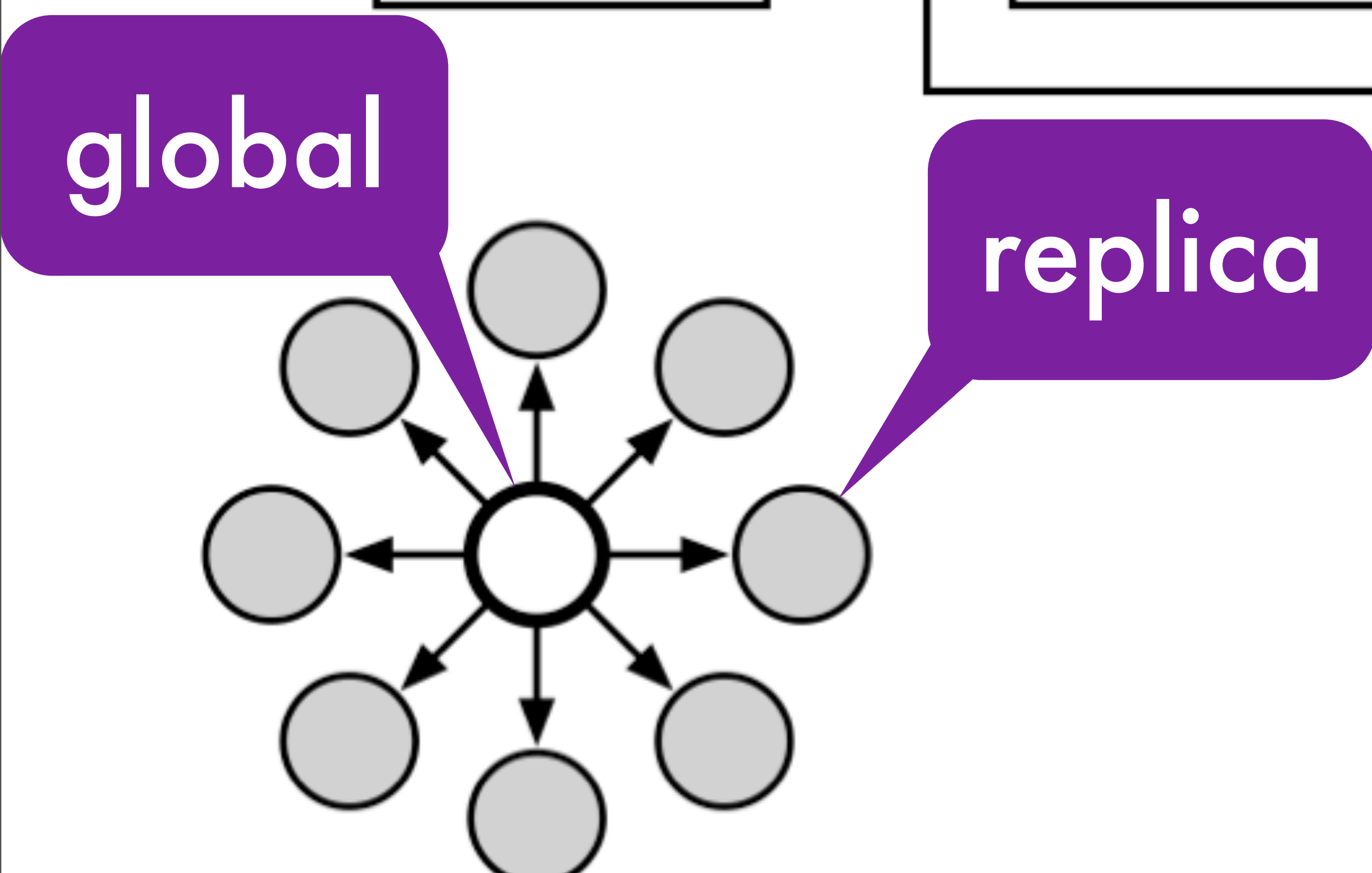


Distribution

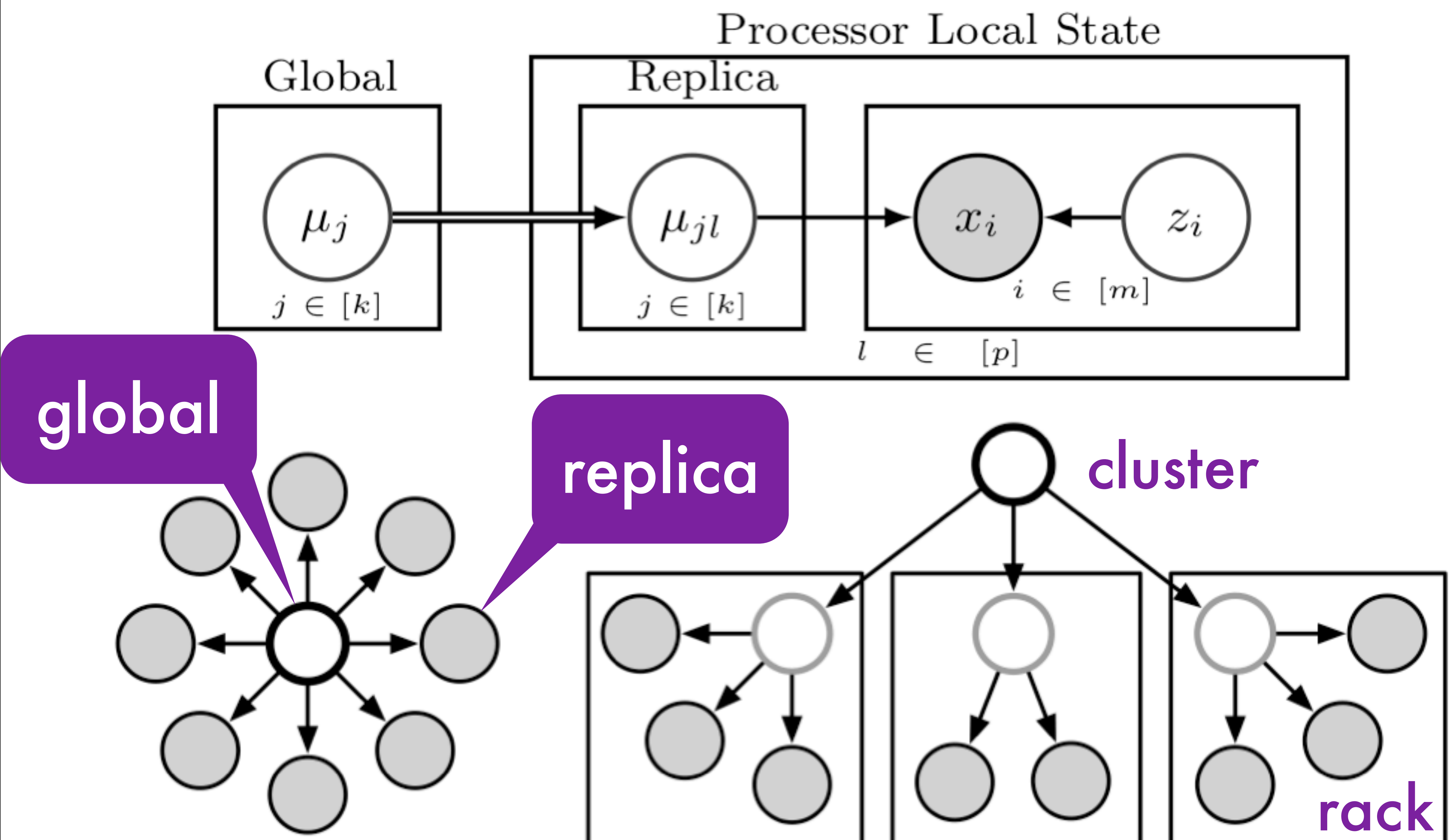


Distribution





Distribution



Synchronization

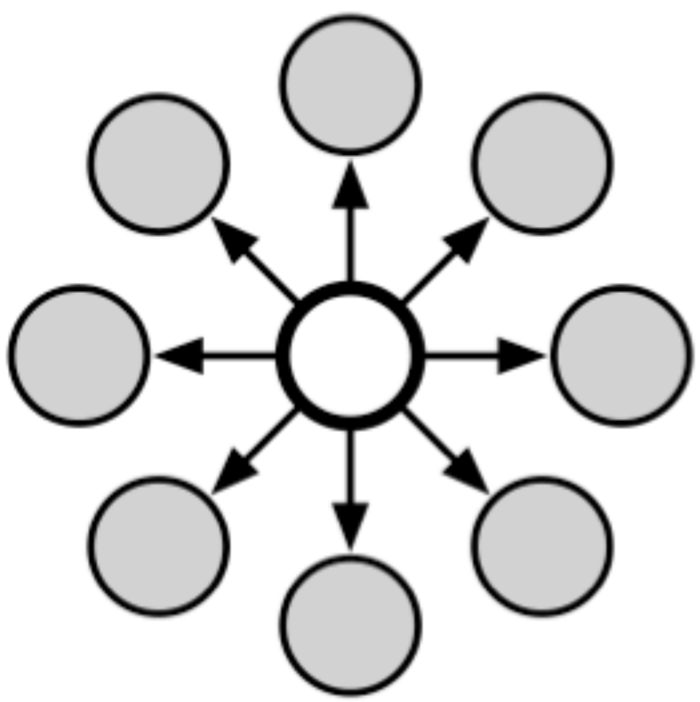
- Child updates local state
  - Start with common state
  - Child stores old and new state
  - Parent keeps global state
- Transmit differences asynchronously
  - Inverse element for difference
  - Abelian group for commutativity (sum, log-sum, cyclic group, exponential families)

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

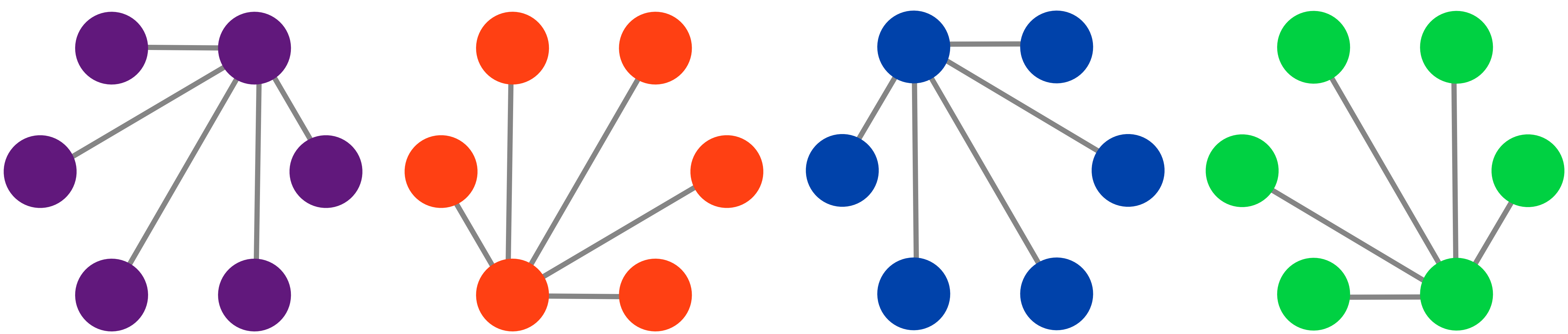
$$x \leftarrow x + \left(x^{\text{global}} - x^{\text{old}}\right)$$

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

Distribution

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

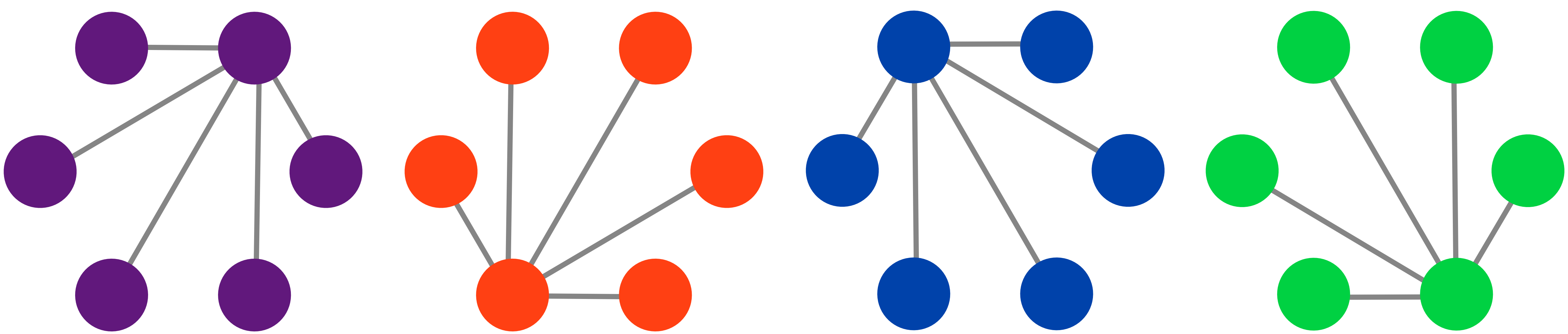
$$m(x) = \operatorname*{argmin}_{m \in M} h(x, m)$$



Distribution & fault tolerance

- Storage is O(1/k) 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

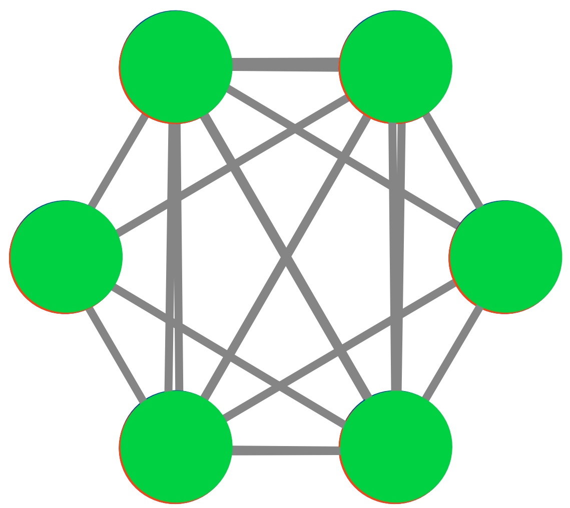
$$m(x) = \underset{m \in M}{\operatorname{argmin}}\, h(x, m)$$



Distribution & fault tolerance

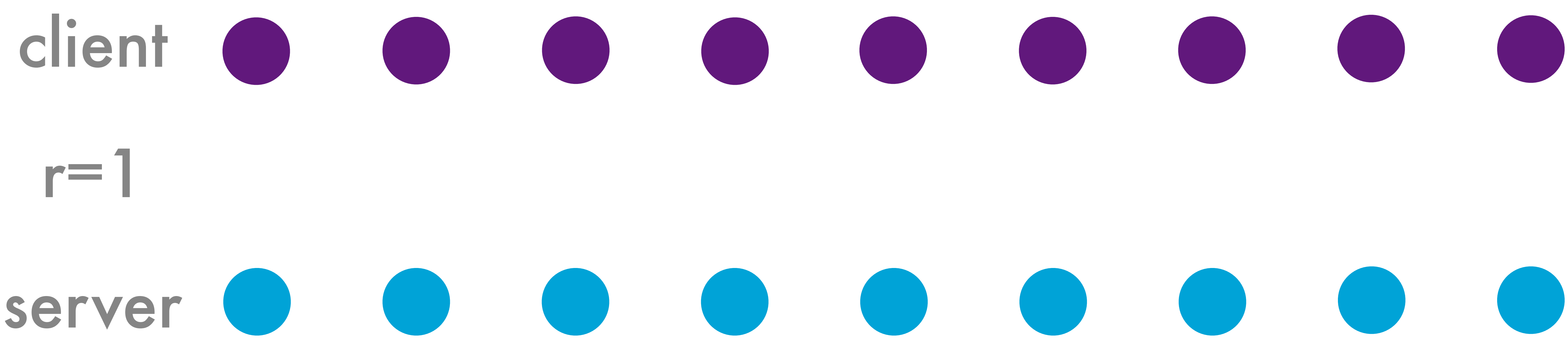
- Storage is O(1/k) per machine
- Fast snapshots O(1/k) per machine (stop sync and dump state per vertex)
- O(k) open connections per machine
- O(1/k) throughput per machine

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



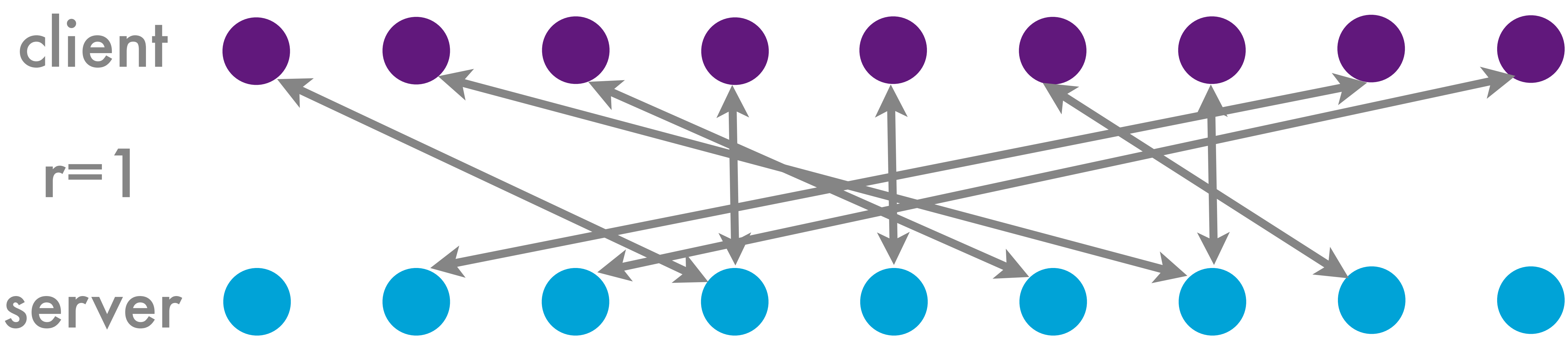
Synchronization

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



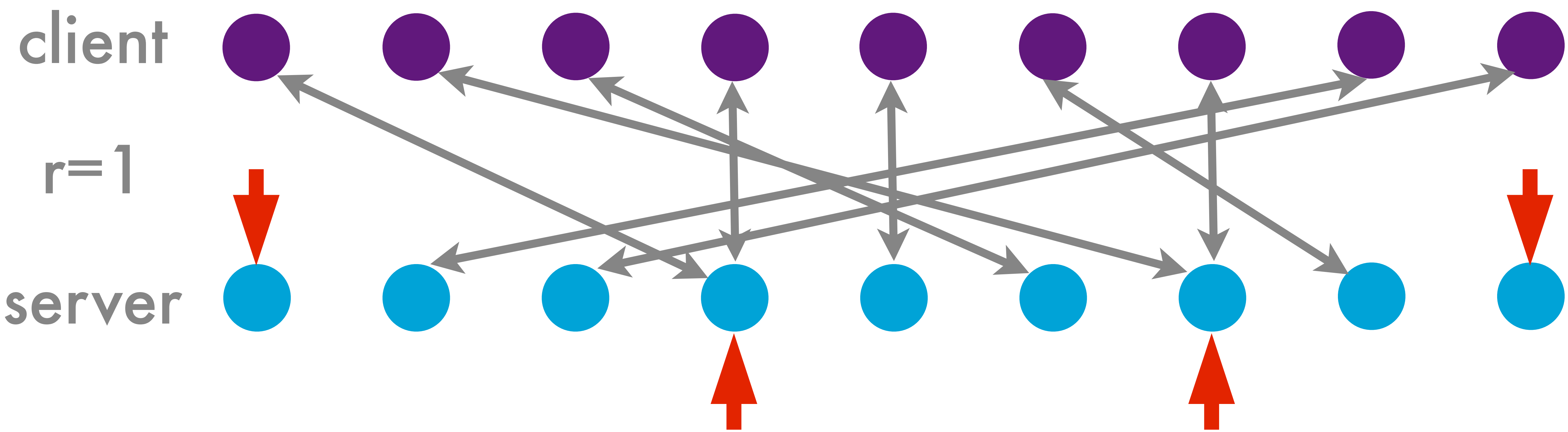
Synchronization

- 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
- Efficiency guarantee [Ahmed et. al WSDM 2012]

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

4 simultaneous connections are sufficient

Architecture

Sequential Algorithm (Gibbs sampler)

- For 1000 iterations do
  - For each document do
    - For each word in the document do
      - Resample topic for the word
      - Update local (document, topic) table
      - Update CPU local (word, topic) table
      - Update global (word, topic) table

Sequential Algorithm (Gibbs sampler)

- For 1000 iterations do
  - For each document do
    - For each word in the document do
      - Resample topic for the word
      - Update local (document, topic) table
      - Update CPU local (word, topic) table
      - Update global (word, topic) table

this kills parallelism

Distributed asynchronous sampler

- For 1000 iterations do (independently per computer)
  - For each thread/core do
    - For each document do
      - For each word in the document do
        - Resample topic for the word
        - Update local (document, topic) table
        - Generate computer local (word, topic) message
    - In parallel update local (word, topic) table
  - In parallel update global (word, topic) table

Distributed asynchronous sampler

- For 1000 iterations do (independently per computer)
  - For each thread/core do
    - For each document do
      - For each word in the document do
        - » Resample topic for the word
        - » Update local (document, topic) table
        - » Generate computer local (word, topic) message
    - In parallel update local (word, topic) table
  - In parallel update global (word, topic) table

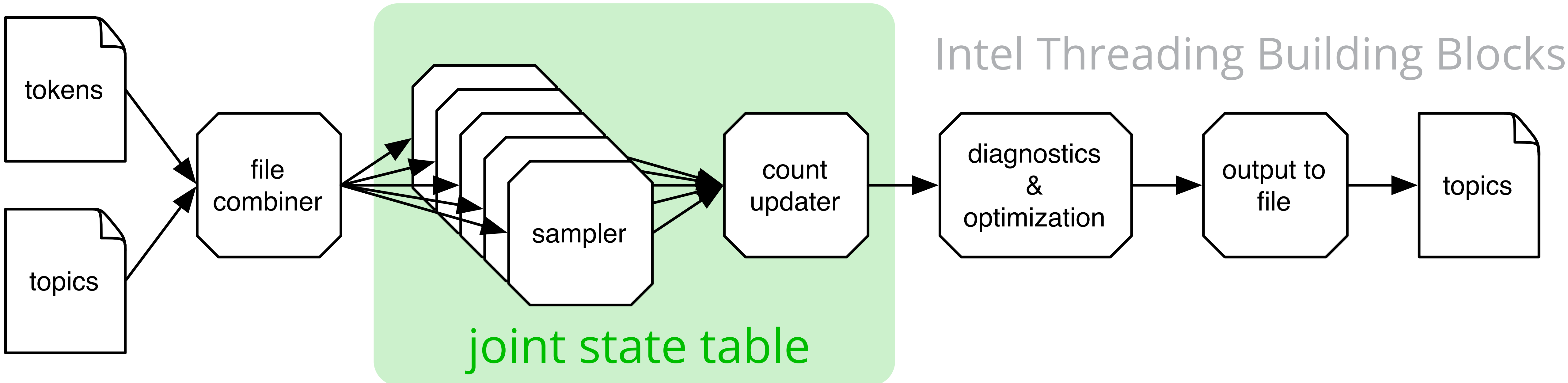
concurrent cpu hdd net

minimal view

continuous sync

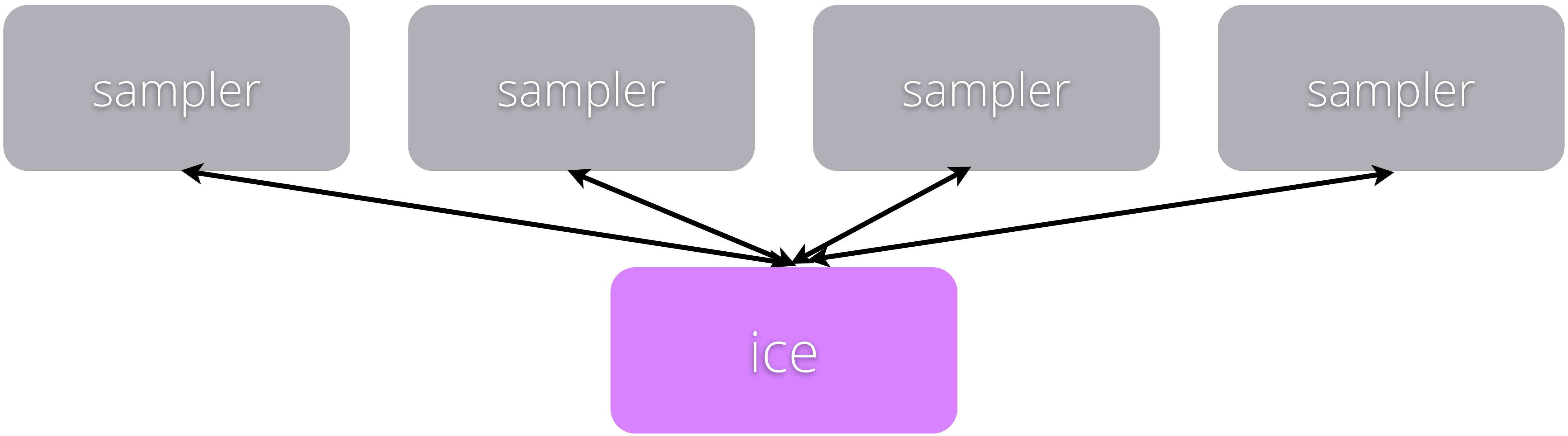
barrier free

Multicore Architecture



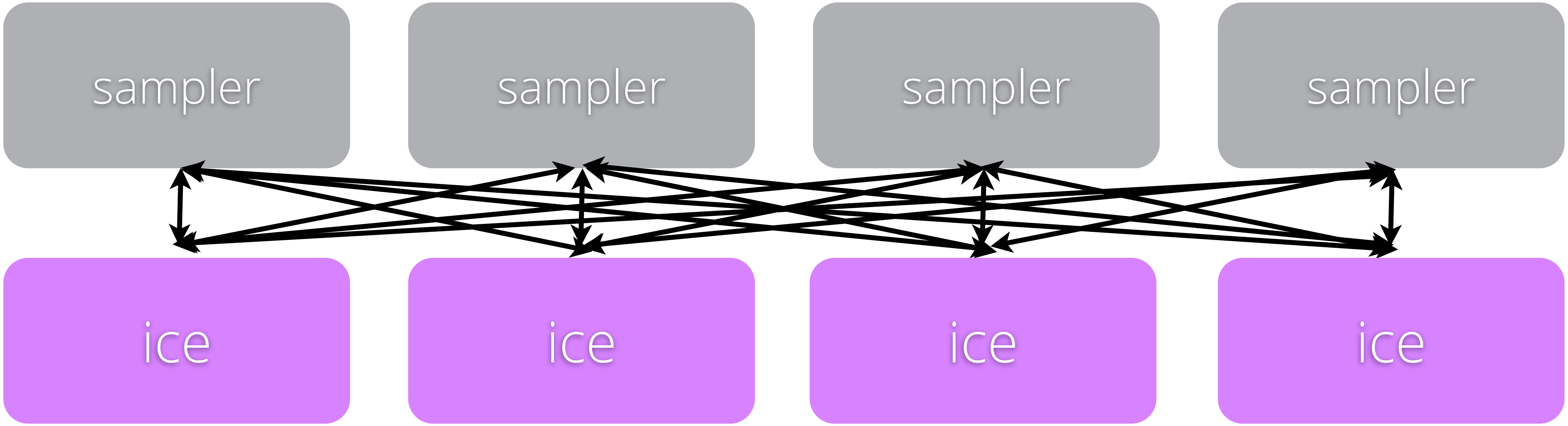
- Decouple multithreaded sampling and updating (almost) avoids stalling for locks in the sampler
- Joint state table
  - much less memory required
  - samplers syncronized (10 docs vs. millions delay)
- Hyperparameter update via stochastic gradient descent
- No need to keep documents in memory (streaming)

Cluster Architecture



- Distributed (key,value) storage via ICE
- Background asynchronous synchronization
  - single word at a time to avoid deadlocks
  - no need to have joint dictionary
  - uses disk, network, cpu simultaneously

Cluster Architecture



- Distributed (key,value) storage via ICE
- Background asynchronous synchronization
  - single word at a time to avoid deadlocks
  - no need to have joint dictionary
  - uses disk, network, cpu simultaneously

Making it work

- Startup
  - –Naive: randomly initialize topics on each node (read from disk if already assigned - hotstart)
  - –Forward sampling for startup much faster
  - –Aggregate changes on the fly
- Failover
  - –State constantly being written to disk (worst case we lose 1 iteration out of 1000)
  - –Restart via standard startup routine
- Achilles heel: need to restart from checkpoint if even a single machine dies.

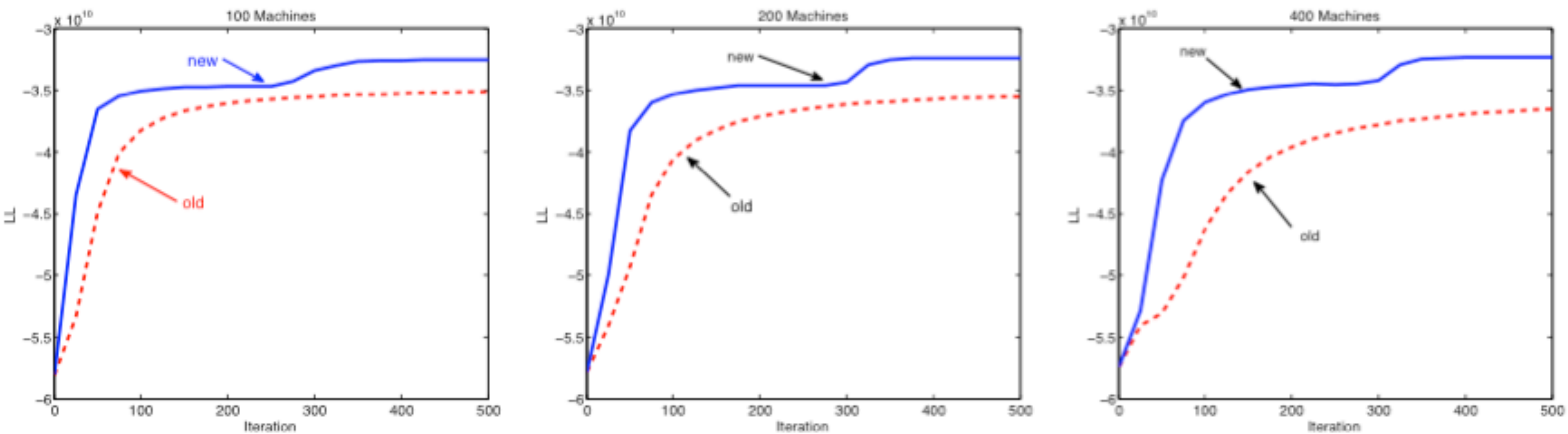
Easily extensible

- Better language model (topical n-grams)
  can process millions of users (vs 1000s)
- Conditioning on side information (upstream)
  estimate topic based on authorship, source, joint user model ...
- Conditioning on dictionaries (downstream)
  integrate topics between different languages
- Time dependent sampler for user model
  approximate inference per episode

Speed (2010 numbers)

- **1M documents per day** on 1 computer
  (1000 topics per doc, 1000 words per doc)
- **350k documents per day** per node
  (context switches & memcached & stray reducers)
- 8 Million docs (Pubmed)
  (sampler does not burn in well - too short doc)
  - Irvine: 128 machines, 10 hours
  - Yahoo: 1 machine, 11 days

Fast sampler



- 8 Million documents, 1000 topics, {100,200,400} machines, LDA
- Red (symmetric latency bound message passing)
- Blue (asynchronous bandwidth bound message passing & message scheduling)
  - 10x faster synchronization time
  - 10x faster snapshots
  - Scheduling improves 10% already on 150 machines

Google

Summary

- Data
- Hardware
- Distributed latent variable inference
- Many models
  - User profiling
  - Multi-domain analysis
  - Social network analysis