

Announcement

- ▶ Homework 1
 - ▶ Available in Blackboard -> Homework
 - ▶ Due: Mar. 19, 11:59pm




Text Clustering

INLP Ch 5

Text Clustering

- ▶ Application
 - ▶ News aggregation website

 Search and browse 25,000

World » [edit](#)

[Heavy Fighting Continues As Pakistan Army Battles Taliban](#)
Voice of America - 10 hours ago


By Barry Newhouse Pakistan's military said its forces have killed 55 to 60 Taliban militants in the last 24 hours in heavy fighting in Taliban-held areas of the northwest.

[Pakistani troops battle Taliban militants for fourth day](#) guardian.co.uk

[Army: 55 militants killed in Pakistan fighting](#) The Associated Press

[Christian Science Monitor](#) - [CNN International](#) - [Bloomberg](#) - [New York Times](#)

[all 3,824 news articles »](#)


[ABC News](#)

[Sri Lanka admits bombing safe haven](#)
guardian.co.uk - 3 hours ago


Sri Lanka has admitted bombing a "safe haven" created for up to 150000 civilians fleeing fighting between Tamil Tiger fighters and the army.

[Chinese billions in Sri Lanka fund battle against Tamil Tigers](#) Times Online

[Huge Humanitarian Operation Under Way in Sri Lanka](#) Voice of America

[BBC News](#) - [Reuters](#) - [AFP](#) - [Xinhua](#)

[all 2,492 news articles »](#)


[WA today](#)

Text Clustering

- ▶ Application
 - ▶ Group customer reviews



Text Clustering: Definition


- ▶ Input:
 - ▶ A set of document $\{d_1, d_2, \dots, d_n\}$
- ▶ Output:
 - ▶ A cluster assignment
 - ▶ $C_1 = \{d_1, d_3, \dots\}$
 - ▶ $C_2 = \{d_2, d_6, \dots\}$
 - ▶ $C_3 = \{d_4, \dots\}$
 - ▶ ...



A Common Method

- ▶ Represent text with feature vectors
- ▶ Apply any clustering algorithm
 - ▶ K-means
 - ▶ Hierarchical agglomerative clustering
 - ▶ Expectation-maximization with mixture of Gaussian
 - ▶ ...

Requires a distance measure
between vectors, e.g., L2

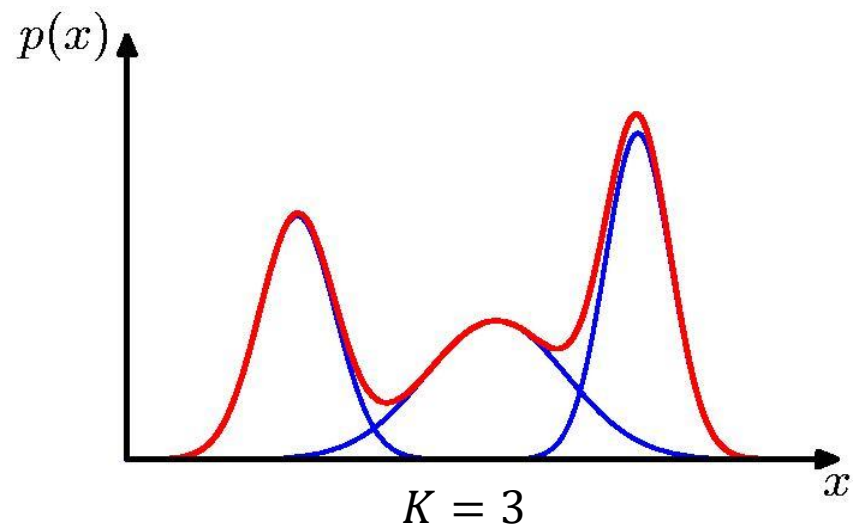


Mixture of Gaussian (MoG)

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \underbrace{\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Component}}$$

Mixing coefficient

$$\forall k : \pi_k \geq 0 \quad \sum_{k=1}^K \pi_k = 1$$

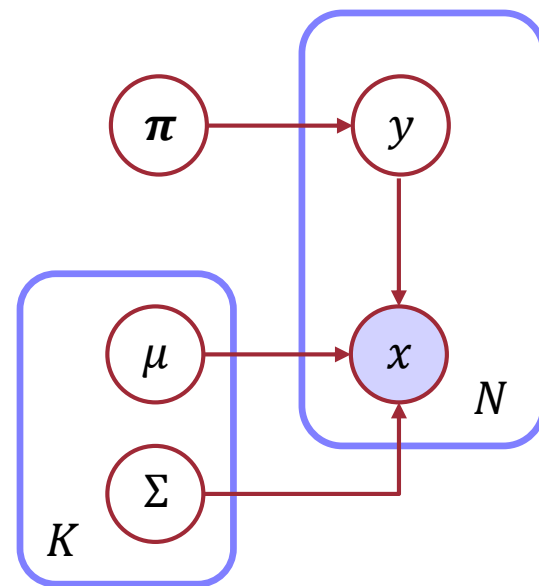


Mixture of Gaussian (MoG)

- ▶ X : data point; Y : cluster label
- ▶ $P(Y)$: Distribution over k components (clusters)
- ▶ $P(X|Y)$: Each component generates data from a **multivariate Gaussian** with mean μ_i and covariance matrix Σ_i

Each data point is sampled from a **generative process**:

1. Choose component $y = i$ with probability π_i
2. Generate data point from $\mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i)$

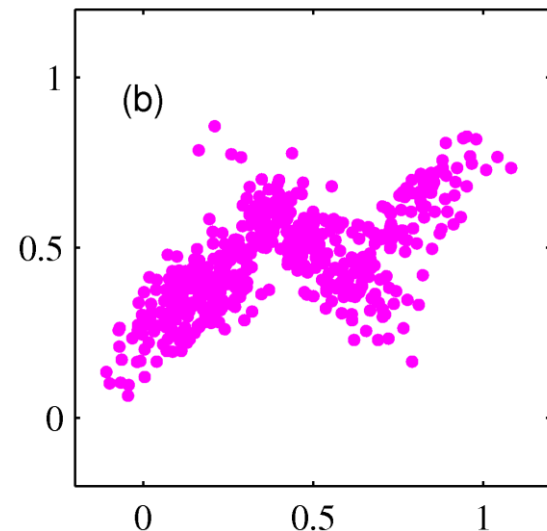


Unsupervised learning for MoG

- ▶ In clustering, we don't know the labels Y !
- ▶ Maximize marginal likelihood:

$$\prod_j P(\mathbf{x}_j) = \prod_j \sum_i P(y_j = i, \mathbf{x}_j) = \prod_j \sum_i \pi_i N(\mathbf{x}_j | \mu_i, \Sigma_i)$$

- ▶ How do we optimize it?
 - ▶ No closed form solution



Expectation-Maximization (EM)

- ▶ Pick K random cluster models (Gaussians)
- ▶ Alternate:
 - ▶ [E step] Assign data instances **proportionately** to different models
 - ▶ [M step] Revise each cluster model **based** on its (**proportionately**) assigned points
- ▶ Stop when no significant change (of marginal likelihood)
- ▶ EM = maximizing marginal likelihood by coordinate ascent



E-step

- ▶ [E step] Assign data instances **proportionately** to different models

- ▶ Compute label distribution of each data point

$$P(y_j = i \mid \mathbf{x}_j, \theta^{(t)}) \propto \pi_i^{(t)} N(\mathbf{x}_j \mid \mu_i^{(t)}, \Sigma_i^{(t)})$$

Just evaluate a
Gaussian at \mathbf{x}_j



M-step

- ▶ **[E step]** Assign data instances **proportionately** to different models

- ▶ Compute label distribution of each data point

$$P(y_j = i | \mathbf{x}_j, \theta^{(t)}) \propto \pi_i^{(t)} N(\mathbf{x}_j | \mu_i^{(t)}, \Sigma_i^{(t)})$$

- ▶ **[M step]** Revise each cluster model **based** on its (**proportionately**) assigned points

- ▶ Compute weighted MLE of parameters given label distributions

$$\mu_i^{(t+1)} = \frac{\sum_j P(y_j = i | \mathbf{x}_j, \theta^{(t)}) \mathbf{x}_j}{\sum_{j'} P(y_{j'} = i | \mathbf{x}_{j'}, \theta^{(t)})}$$

$$\pi_i^{(t+1)} = \frac{\sum_j P(y_j = i | \mathbf{x}_j, \theta^{(t)})}{m}$$

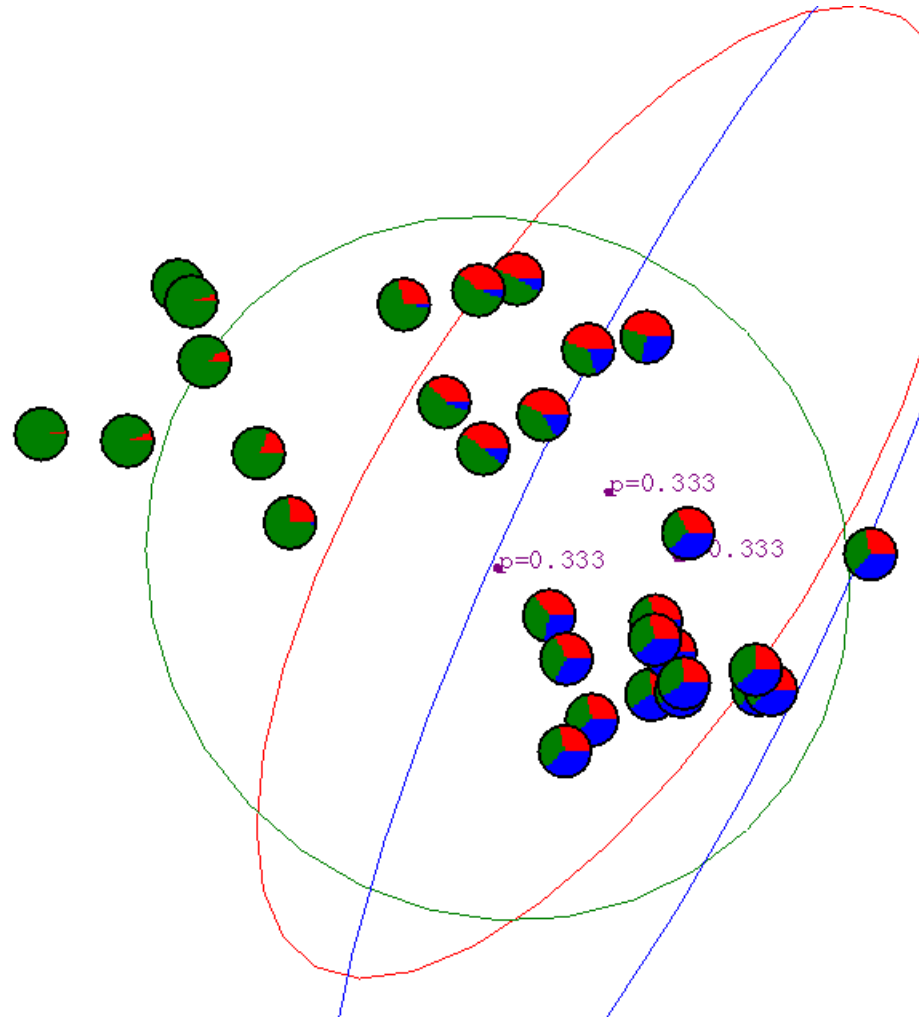
$$\Sigma_i^{(t+1)} = \frac{\sum_j P(y_j = i | \mathbf{x}_j, \theta^{(t)}) [\mathbf{x}_j - \mu_i^{(t+1)}][\mathbf{x}_j - \mu_i^{(t+1)}]^T}{\sum_{j'} P(y_{j'} = i | \mathbf{x}_{j'}, \theta^{(t)})}$$

$m = \#$ training examples



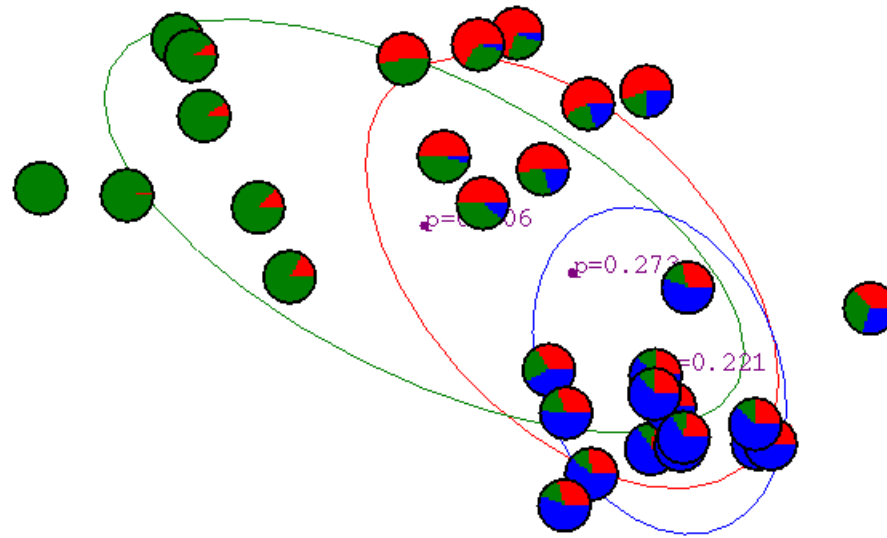
Gaussian Mixture Example

► Start



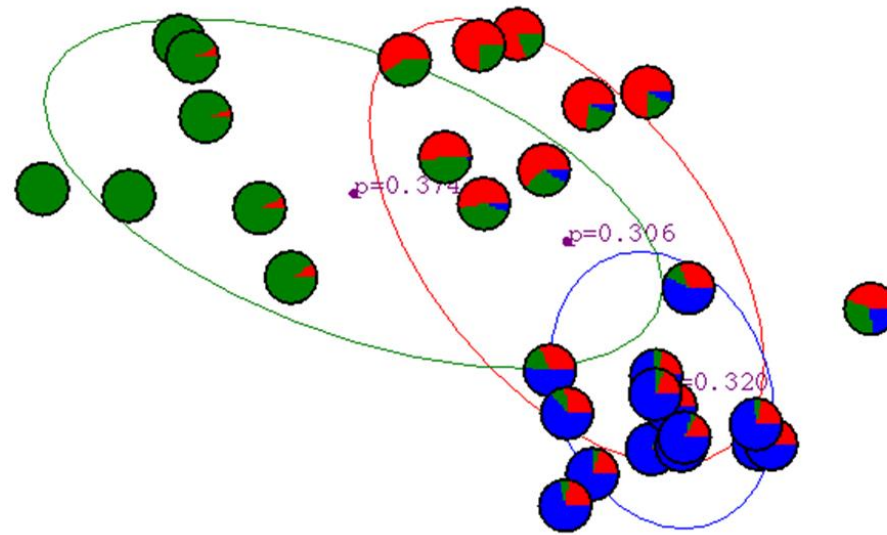
Gaussian Mixture Example

► 1st iteration



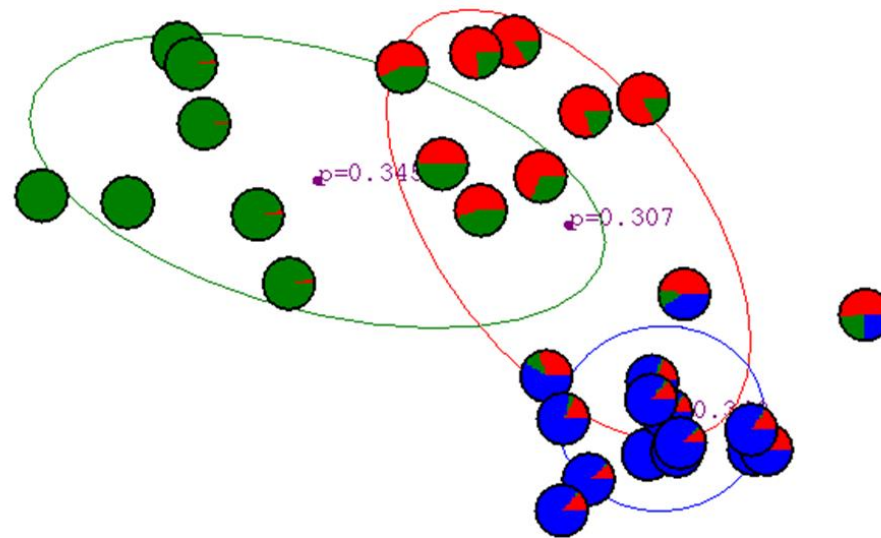
Gaussian Mixture Example

► 2nd iteration



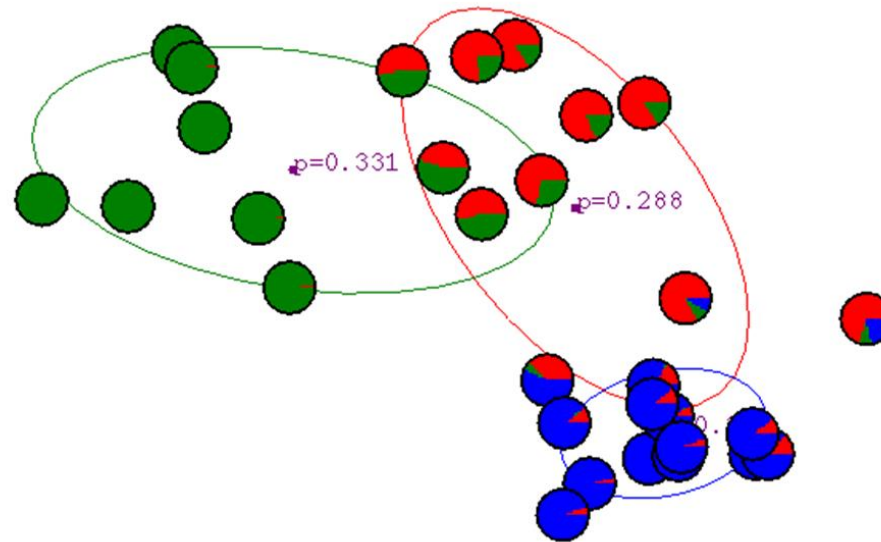
Gaussian Mixture Example

▶ 3rd iteration



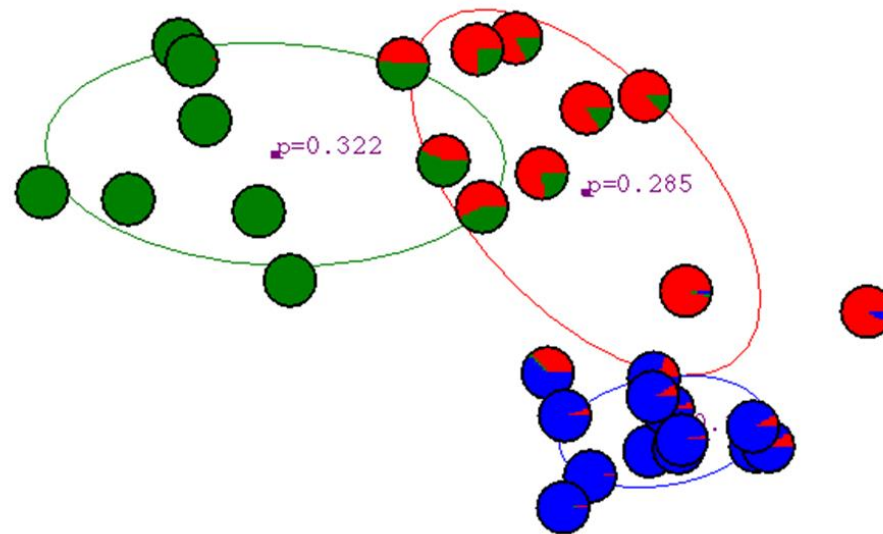
Gaussian Mixture Example

► 4th iteration



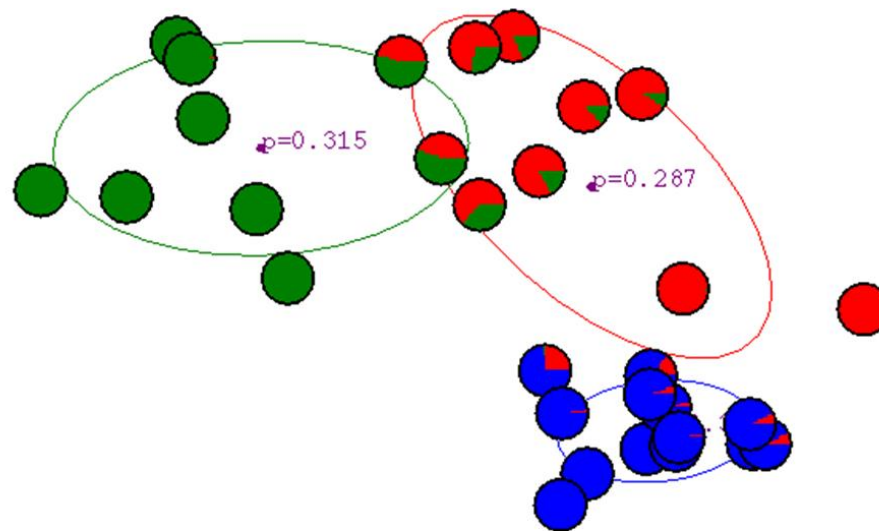
Gaussian Mixture Example

► 5th iteration



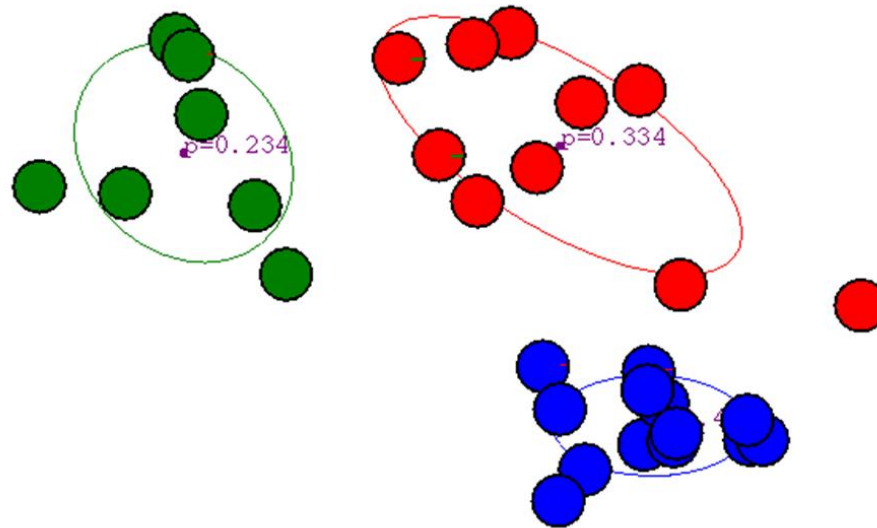
Gaussian Mixture Example

► 6th iteration



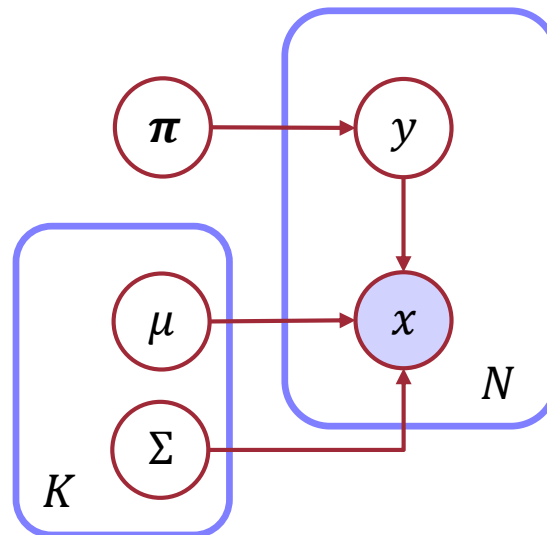
Gaussian Mixture Example

► 20th iteration



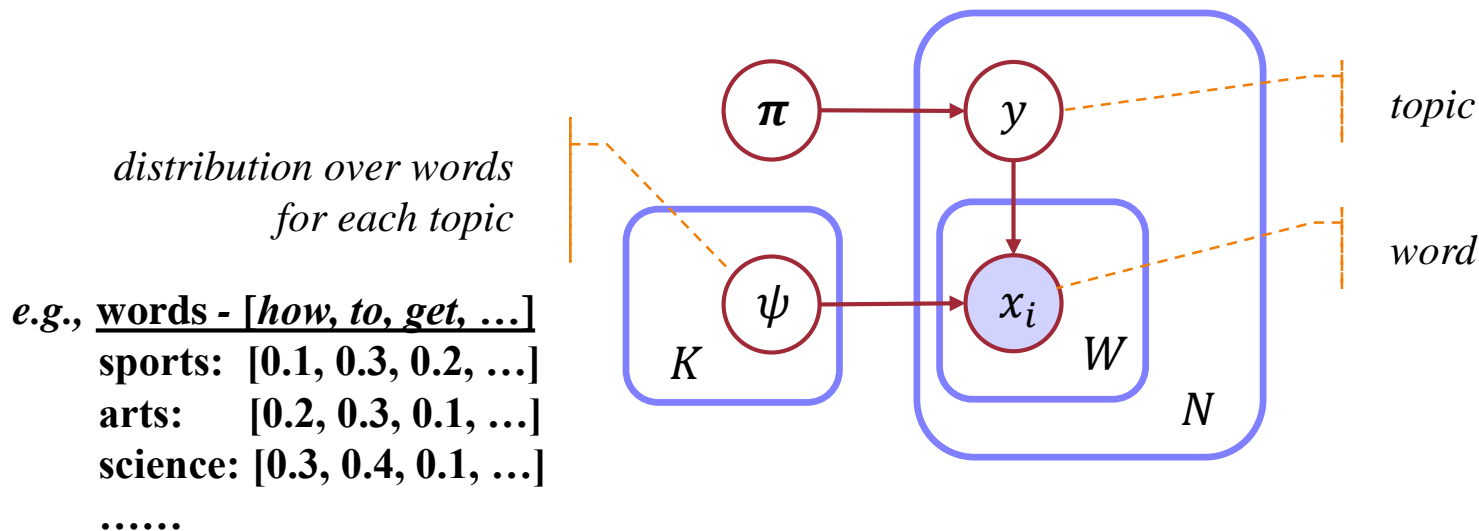
Generative models

- ▶ MoG generates a document feature vector with one of K Gaussian distributions
- ▶ Can we directly generate a document (sequence of words)?
 - ▶ Yes! Generating words with one of K discrete distributions

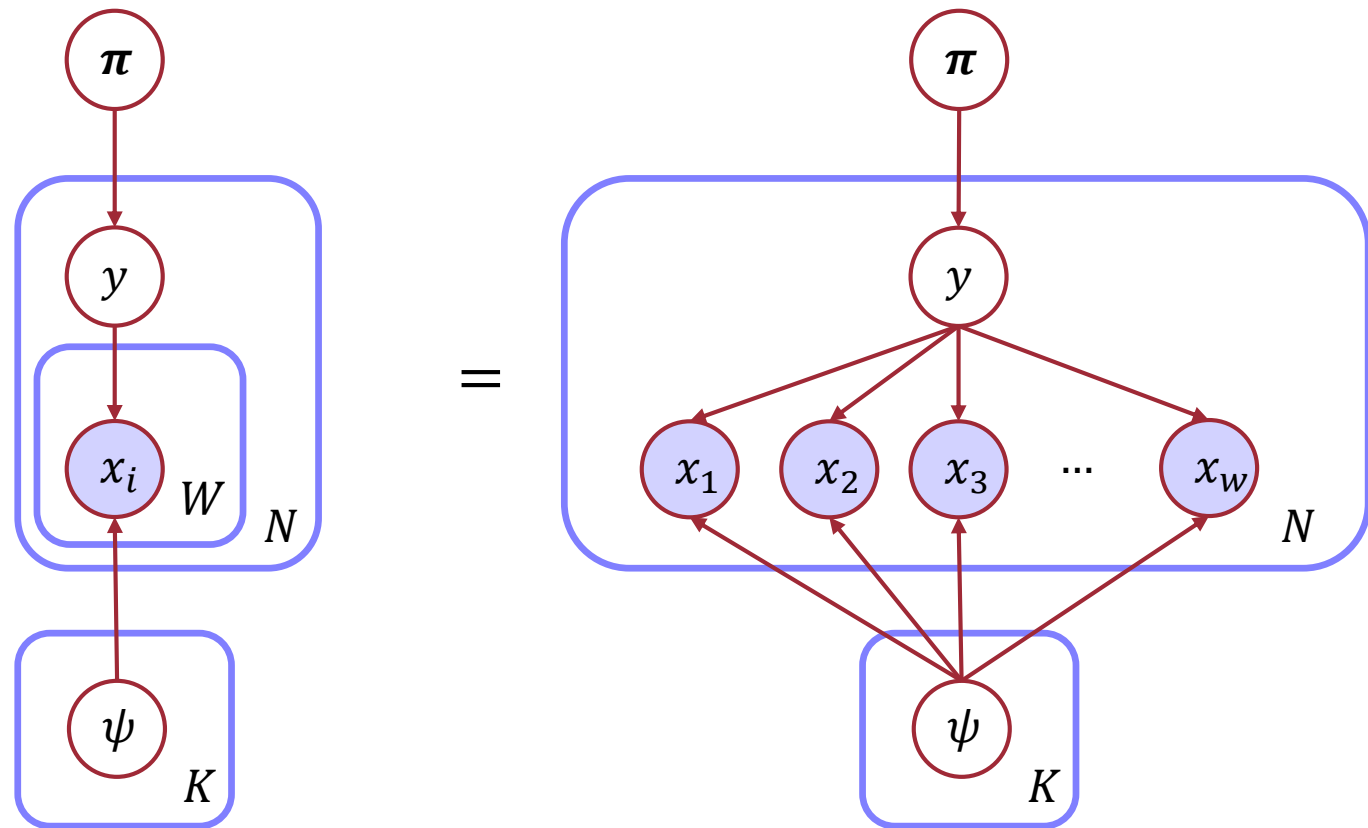


Generative models

- ▶ MoG generates a document feature vector with one of K Gaussian distributions
- ▶ Can we directly generate a document (sequence of words)?
 - ▶ Yes! Generating words with one of K discrete distributions



Generative models



This is exactly a naive Bayes model!



Unsupervised Naïve Bayes

- ▶ We can run EM for unsupervised learning of naive Bayes
 - ▶ i.e., text clustering based on words, not features
- ▶ [E step] Assign **documents** proportionately to different **topics**
 - ▶ Compute topic distribution of each document

$$P(y_j = i \mid x_{j,1:w}, \theta^{(t)}) \propto \pi_i^{(t)} \prod_{k=1}^w P(x_{j,k} \mid \psi_i^{(t)})$$



Unsupervised Naïve Bayes

- ▶ We can run EM for unsupervised learning of naive Bayes
 - ▶ i.e., text clustering based on words, not features
- ▶ [M step] Revise each **topic** based on its (proportionately) assigned **words**
 - ▶ Compute weighted MLE of parameters given topic distributions
 - ▶ Denote $\psi_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,v}\}$

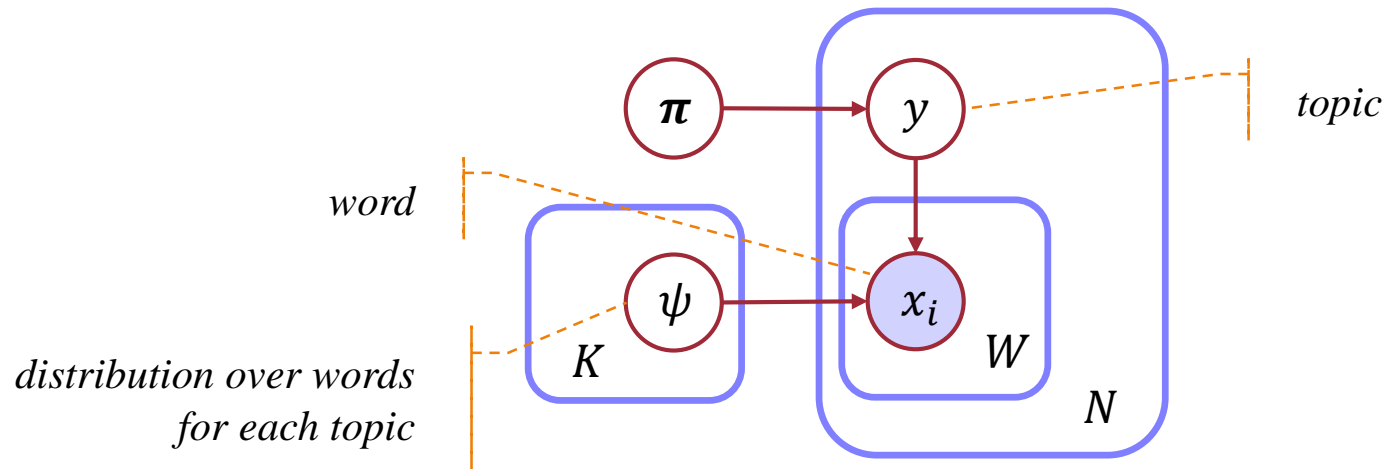
$$p_{i,l}^{(t+1)} = \frac{\sum_j P(y_j=i | x_{j,1:w}, \theta^{(t)}) \sum_k \mathbf{1}(x_{j,k}=l)}{\sum_j P(y_j=i | x_{j,1:w}, \theta^{(t)}) \cdot w_j} \quad \pi_i^{(t+1)} = \frac{\sum_j P(y_j=i | x_{j,1:w}, \theta^{(t)})}{m}$$

where v is the vocabulary size, m is the # of training documents.



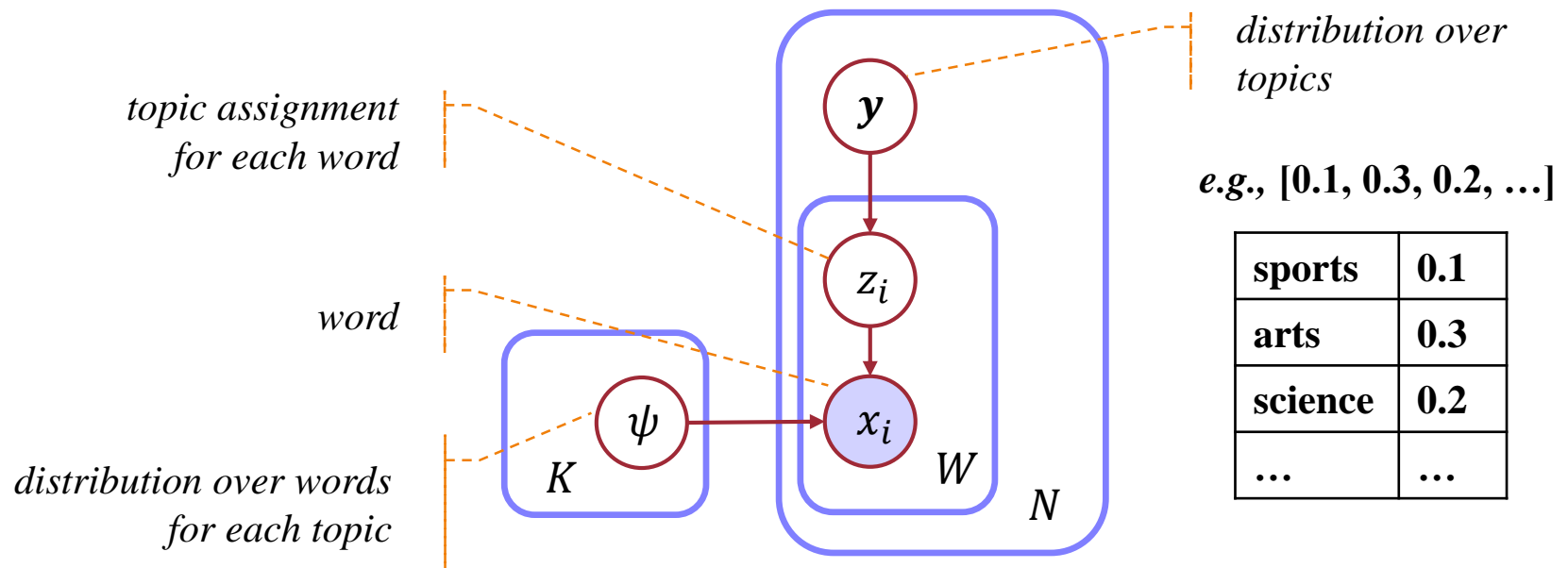
Topic modeling

- ▶ Text clusters may correspond to different topics
- ▶ So far, we assume a single cluster label for each document
- ▶ But, a document may cover multiple topics
 - ▶ Can we learn that?



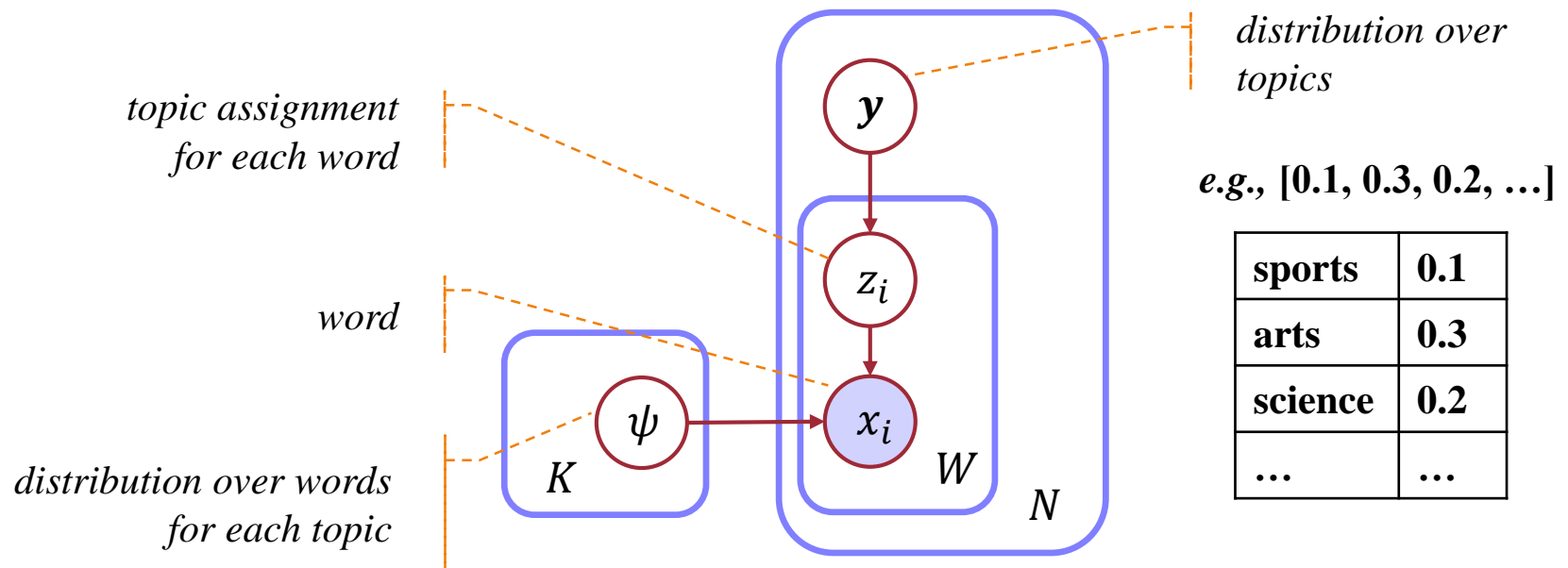
Topic modeling

- ▶ Text clusters may correspond to different topics
- ▶ So far, we assume a single cluster label for each document
- ▶ But, a document may cover multiple topics
 - ▶ Can we learn that?



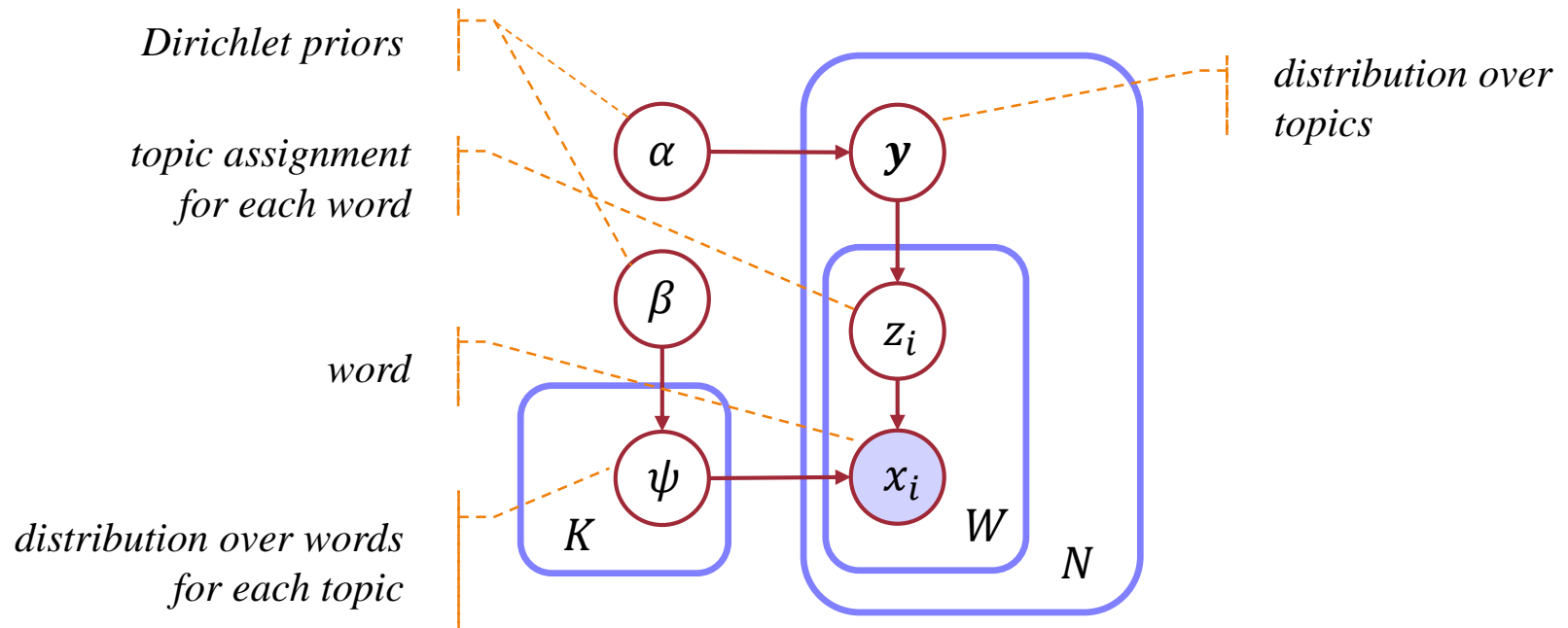
Topic modeling

- ▶ This is called probabilistic latent semantic analysis (pLSA)
 - ▶ Again, we can run EM to learn it
- ▶ We can further add Dirichlet priors over topic & word distributions

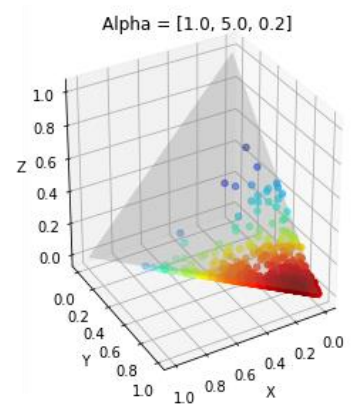
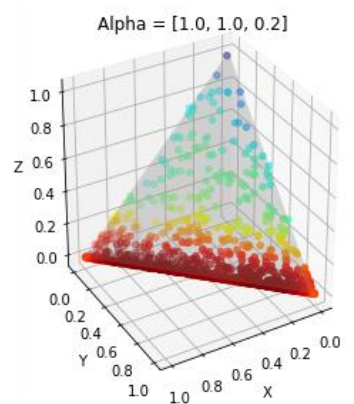
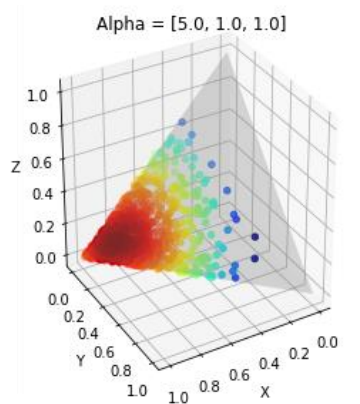
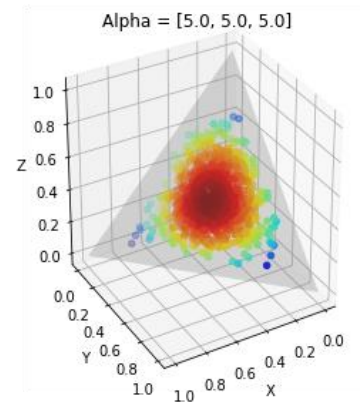
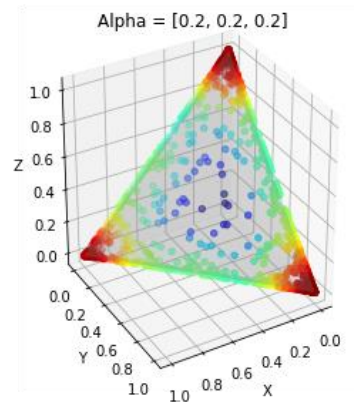
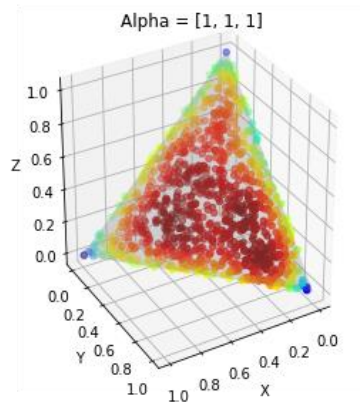


Topic modeling

- ▶ Dirichlet priors: encourage topic & word distributions to be sparse
 - ▶ A document shall cover only a few topics
 - ▶ In a topic, only a subset of words has high frequency

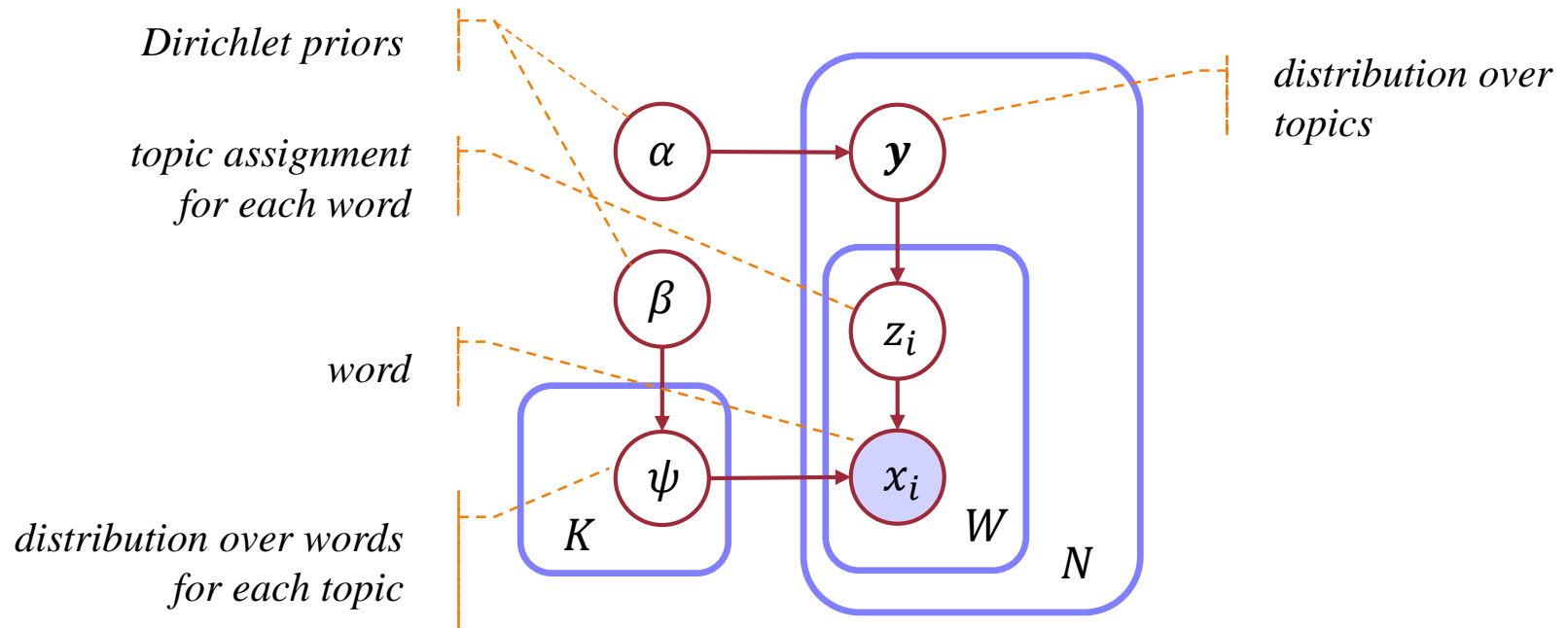


Dirichlet Distribution

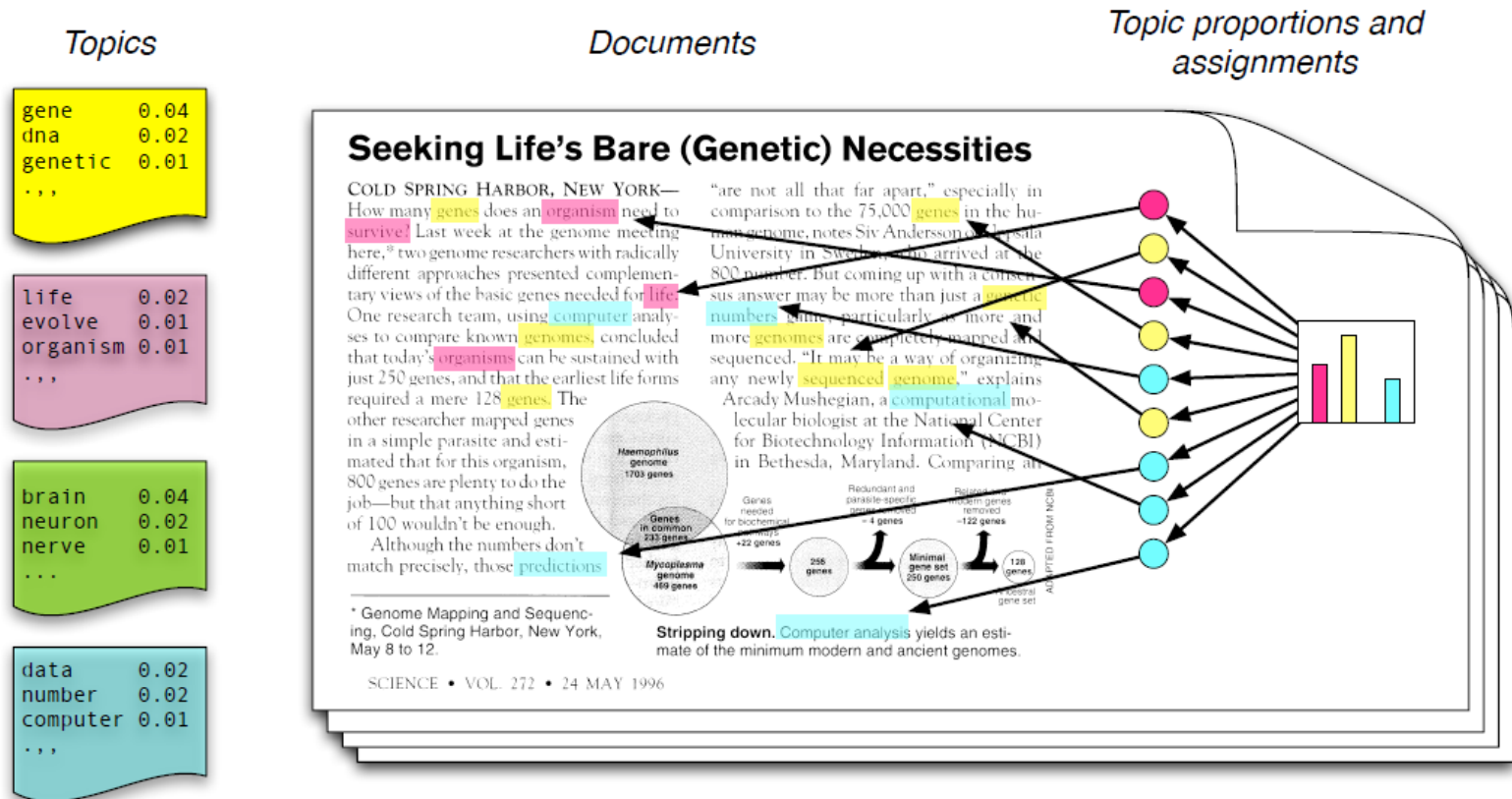


Topic modeling

- ▶ This is called Latent Dirichlet Allocation (LDA)
 - ▶ Learning: variational inference or MCMC

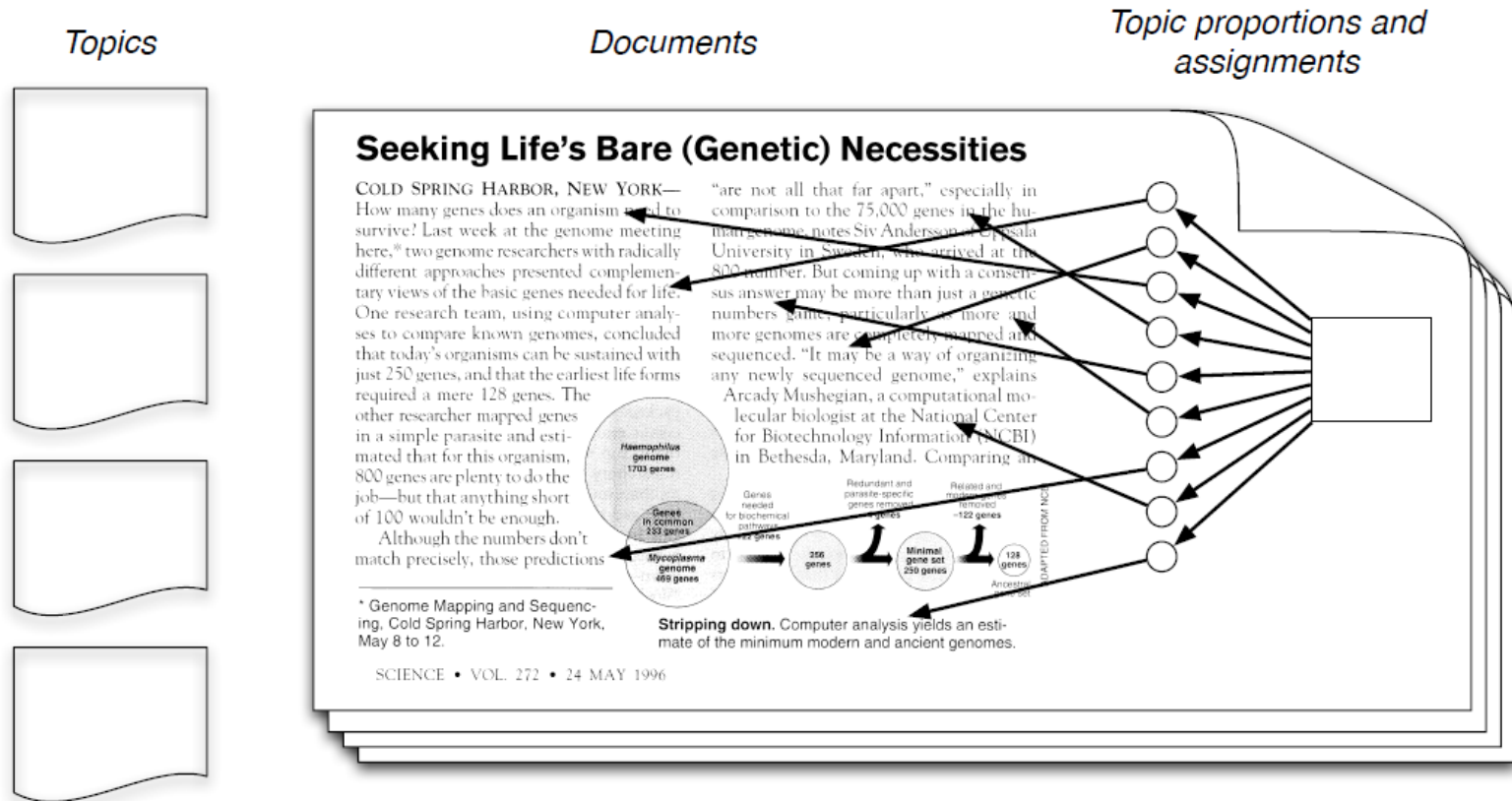


Illustration



- ▶ Each **topic** is a distribution of words; each **document** is a mixture of corpus-wide topics; and each **word** is drawn from one of those topics.

Illustration



- ▶ In reality, we only observe documents. The other structures are hidden variables that must be inferred.

Topics inferred by LDA

“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



Topic assignments in document

► Based on the topics shown in last slide

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.



EM in General

- ▶ Can be used to learn any model with hidden variables (missing data)
- ▶ Alternate:
 - ▶ Compute distributions over hidden variables based on current parameter values
 - ▶ Compute new parameter values to maximize expected log likelihood based on distributions over hidden variables
- ▶ Stop when no changes
- ▶ Can reach a local optimum but not necessarily a global optimum



Math Behind EM

- ▶ EM is coordinate ascent on $F(\theta, Q)$

$$\ell(\theta : \mathcal{D}) \geq F(\theta, Q) = \sum_{j=1}^m \sum_{\mathbf{z}} Q(\mathbf{z} | \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j | \theta)}{Q(\mathbf{z} | \mathbf{x}_j)}$$

↑
Jensen's inequality

- ▶ E-step fixes θ and optimizes Q
- ▶ M-step fixes Q and optimizes θ
- ▶ Convergence of EM
 - ▶ Neither E-step nor M-step decreases $F(\theta, Q)$



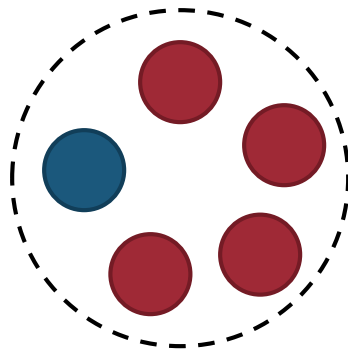


Evaluation

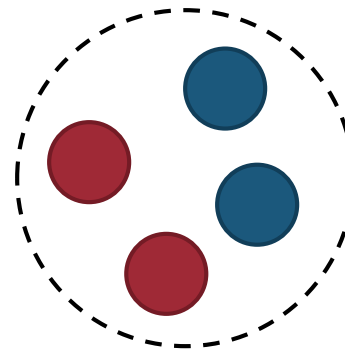


Evaluation of Clustering

- ▶ Many different metrics
 - ▶ Purity / Inverse Purity
 - ▶ Rand index
 - ▶ MUC
 - ▶ B-CUBED
 - ▶ ...



Predicted Cluster 1



Predicted Cluster 2

● Gold Cluster 1
● Gold Cluster 2

Evaluation

► Purity

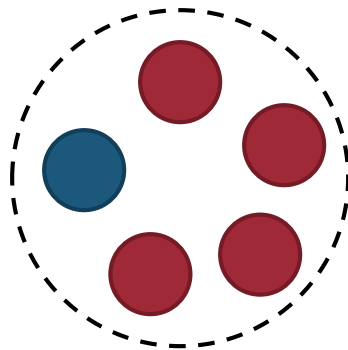
- Extent to which predicted clusters contain a single gold clusters.

$$Purity = \frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d|$$

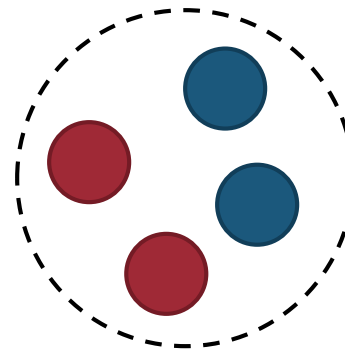
N : # of data points

M : Set of predicted clusters

D : Set of gold clusters



Predicted Cluster 1



Predicted Cluster 2

● Gold Cluster 1
● Gold Cluster 2



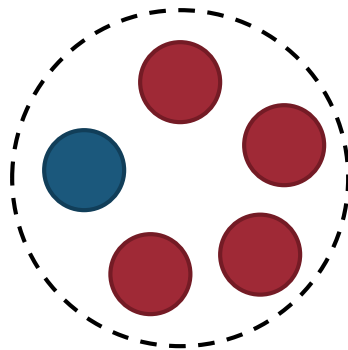
Evaluation

► Purity

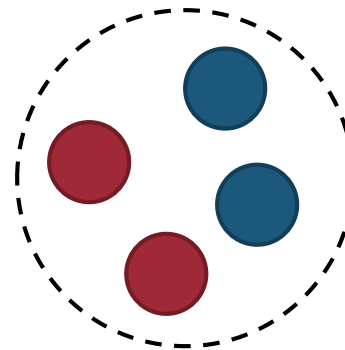
$$\frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d|$$
$$= \frac{1}{9} (\max\{4, 1\} + \max\{2, 2\})$$
$$= \frac{1}{9} (4 + 2) \approx 0.667$$

The Confusion Matrix

	Pred. 1	Pred. 2
Gold. 1	4	2
Gold. 2	1	2



Predicted Cluster 1

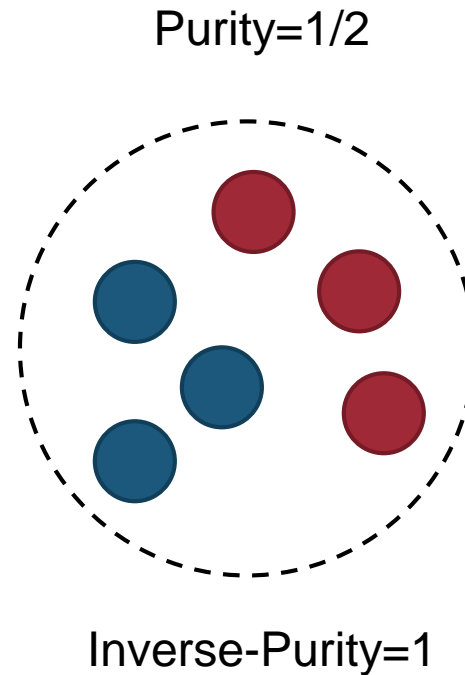
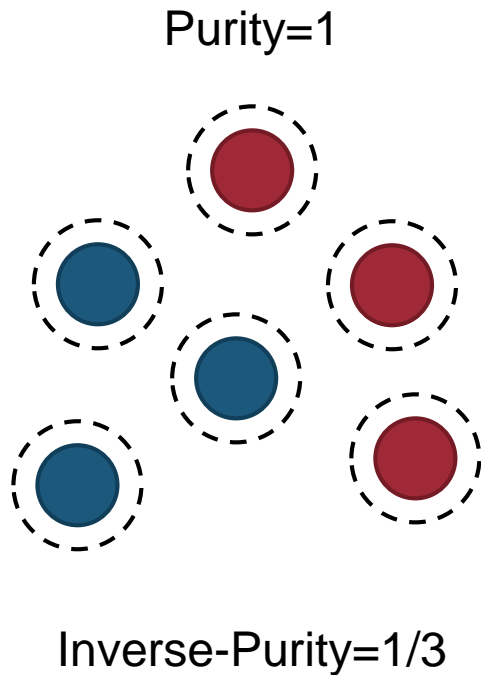


Predicted Cluster 2

● Gold Cluster 1
● Gold Cluster 2

Evaluation

- ▶ Purity: $\frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d|$
- ▶ Inverse Purity
 - ▶ Same formula, with M & D exchanged





Summary



Text Clustering

- ▶ Mixture of Gaussian
- ▶ Unsupervised Naive Bayes
- ▶ Topic models
 - ▶ pLSA, LDA
- ▶ Learning
 - ▶ Expectation-maximization
- ▶ Evaluation

