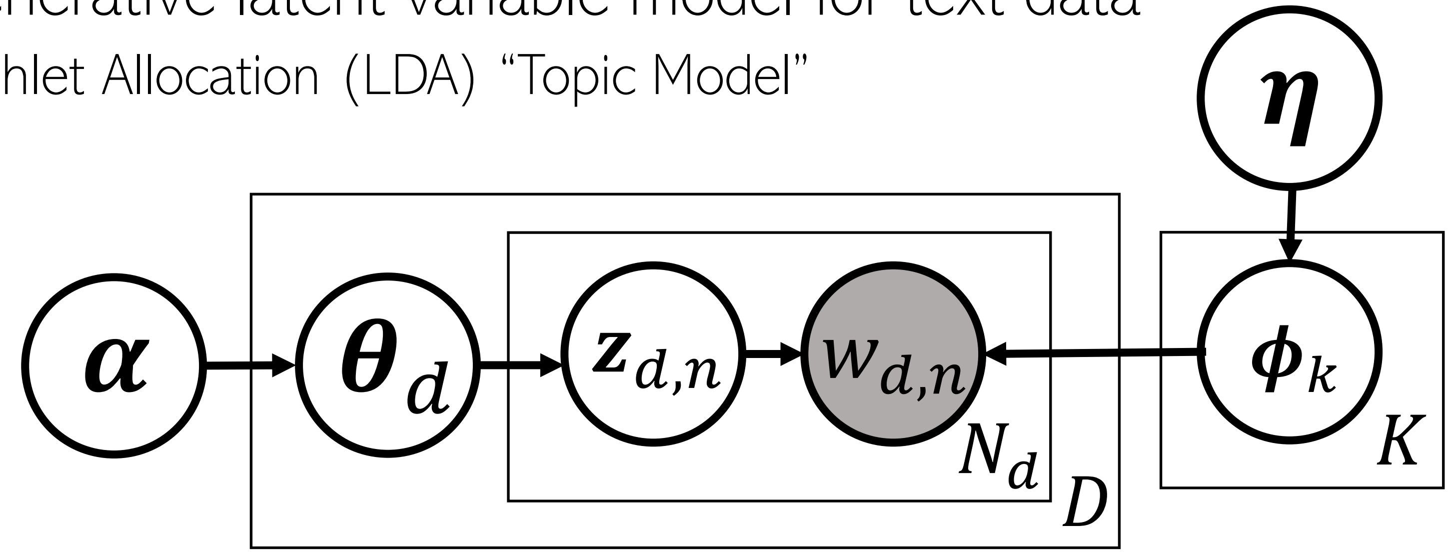
Assorted Topics (1)

CS772A: Probabilistic Machine Learning

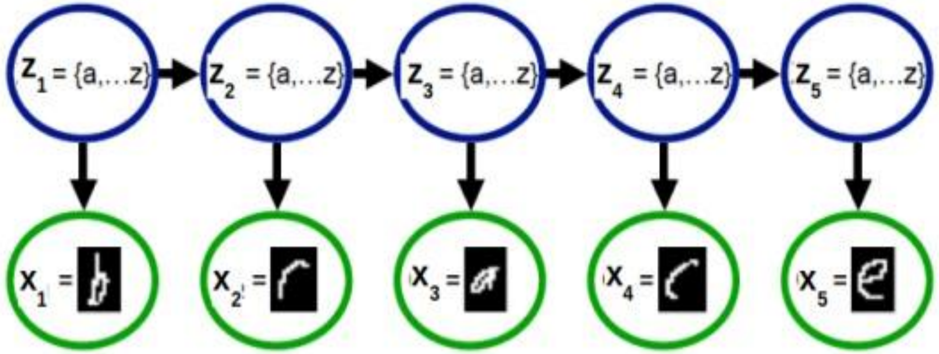
Piyush Rai

Plan today..

- A classical generative latent variable model for text data
  - Latent Dirichlet Allocation (LDA) "Topic Model"



- Probabilistic models for sequential data
  - HMM, state-space models



Latent Dirichlet Allocation (LDA)
a.k.a. “Topic Model”

Motivation: Multinomial Mixture Model for Text

- Assume D documents, and document d has N_d words in it
- We can represent doc d by a word count vector $\boldsymbol{w}_d$
- Assuming a vocab of V unique words, $\boldsymbol{w}_d$ is a $V \times 1$ vector of counts
  - w_{dv} = no of times word v appears in doc d
- Let's model the docs by a mixture of K multinomial distributions, each V-dim
  - The k^{th} multinomial modeled by a V-dim prob vector $\boldsymbol{\phi}_k$ (sums to 1)
  - $\boldsymbol{\phi}_k$ can be thought of as a "topic vector" (or just "topic"), ϕ_{kv}: prob of word v in topic k
- Generative model and plate diagram below

Each topic is a prob. distribution over word tokens

Each representing a "topic" (K topics)

Limitation: Each doc d belongs to a single cluster $\boldsymbol{z}_d$ and all words in a document assumed to be from the same topic. This is unrealistic/restrictive

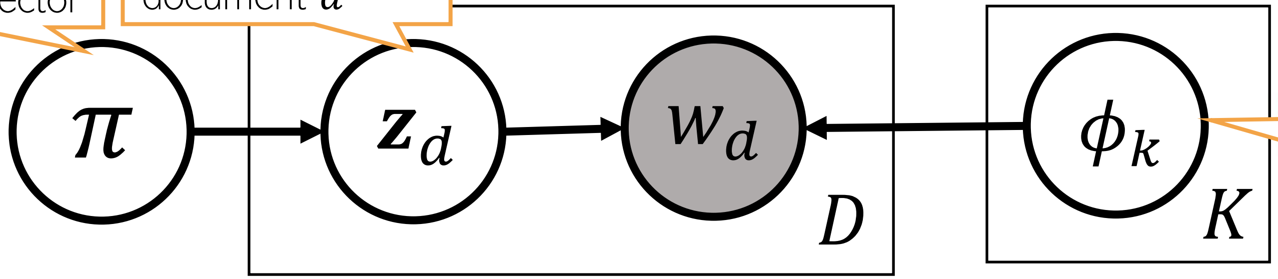
$$\boldsymbol{z}_d \sim \text{multinoulli}(\pi)$$

$$\boldsymbol{w}_d \sim \text{multinomial}(\phi_{z_d}, N_d)$$

Counts will sum to N_d

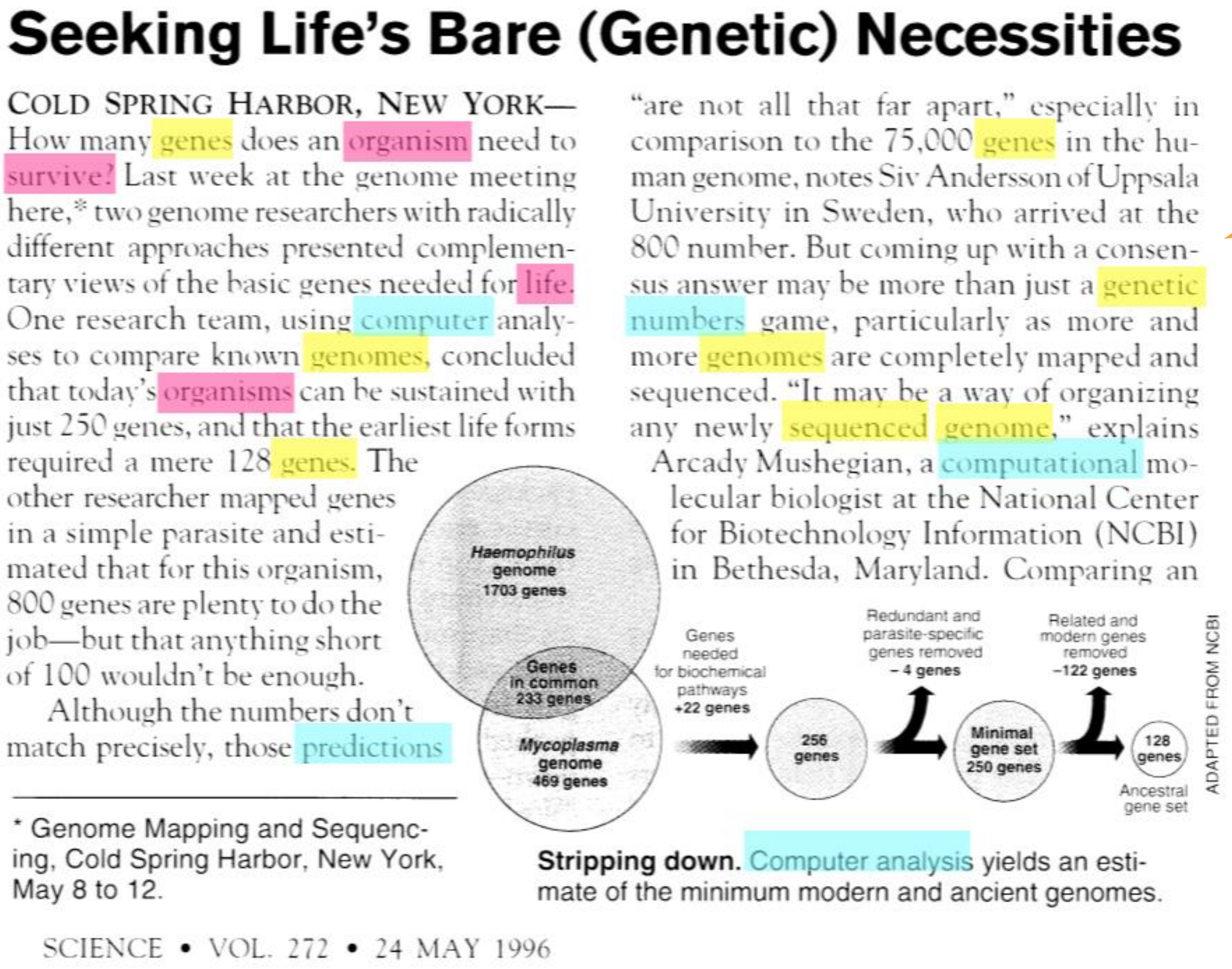
Topic Mixing proportion vector

Cluster/topic of document d



K topic distributions

Documents can be about multiple topics



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

How do we find the word-topic associations in each document?

How do we use them to learn topics in the given text collection?

How do we learn low-dim document representations in terms of the topics they represent?

A More Fine-Grained Mixture Model for Text

- Assume a corpus-level topic mixing proportions $\boldsymbol{\alpha}$ ($K \times 1$ prob vector)
- Also assume doc-level topic mixing props $\boldsymbol{\theta}_d$ ($K \times 1$ prob vector)
- Instead of assuming a single cluster $\boldsymbol{z}_d$ for doc $\boldsymbol{d}$, cluster each word in it
  - $\boldsymbol{z}_{d,n} \in \{1,2,\dots,K\}$ denotes the cluster/topic of word $\boldsymbol{w}_{d,n} \in \{1,2,\dots,V\}$

Each assumed a one-hot $K \times 1$ vector

- Can obtain the "average" clustering for doc $\boldsymbol{d}$ using $\boldsymbol{\theta}_d$ or $\bar{\boldsymbol{z}}_d = \frac{1}{N_d}\sum_{n=1}^{N_d} \boldsymbol{z}_{d,n}$
- The generative model is as follows

Locally-conjugate. Easy Gibbs sampling, VI, etc

Latent Dirichlet Allocation* (LDA) Topic Model

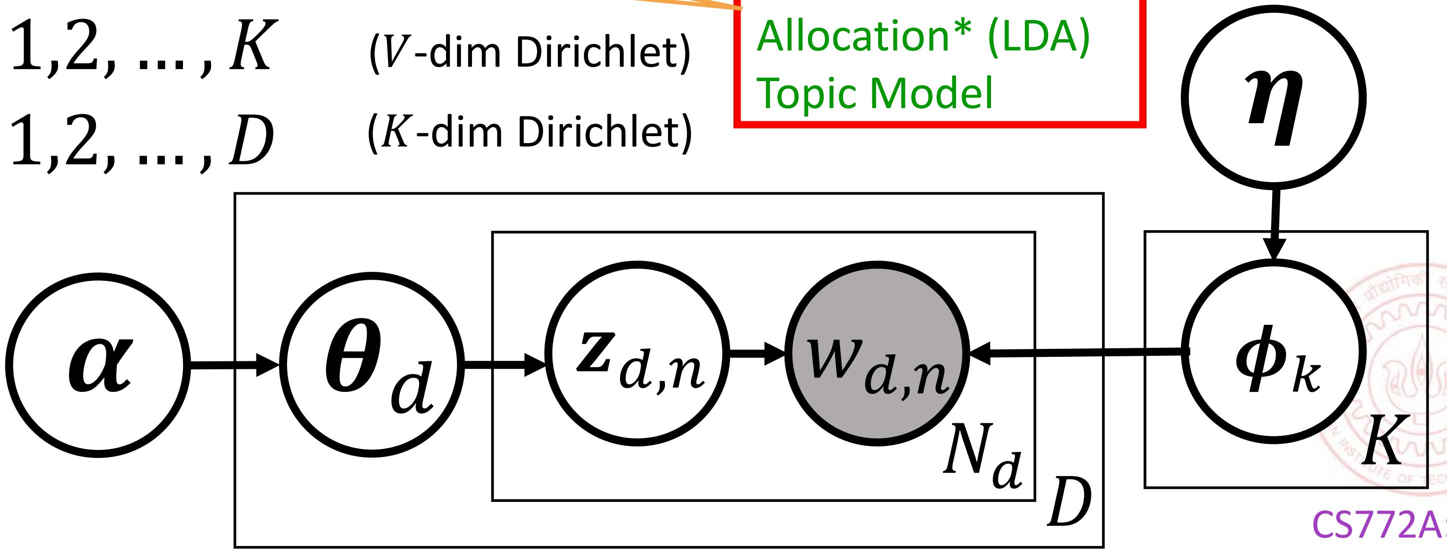
Somewhat similar to Dir-Mult PCA model

$$\phi_k \sim \text{Dirichlet}(\boldsymbol{\eta}) \quad k = 1,2,\dots,K \qquad (V\text{-dim Dirichlet})$$

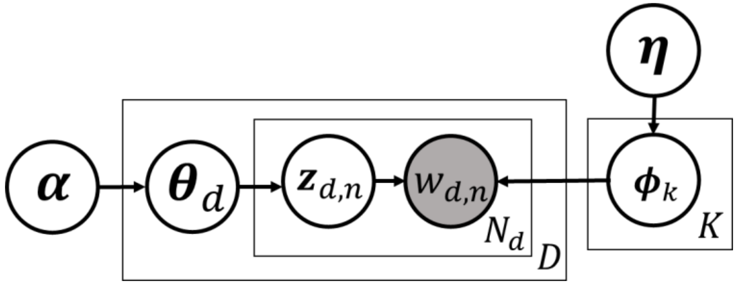
$$\theta_d \sim \text{Dirichlet}(\boldsymbol{\alpha}) \quad d = 1,2,\dots,D \qquad (K\text{-dim Dirichlet})$$

$$\boldsymbol{z}_{d,n} \sim \text{multinoulli}(\theta_d)$$

$$\boldsymbol{w}_{d,n} \sim \text{multinoulli}(\phi_{z_{d,n}})$$



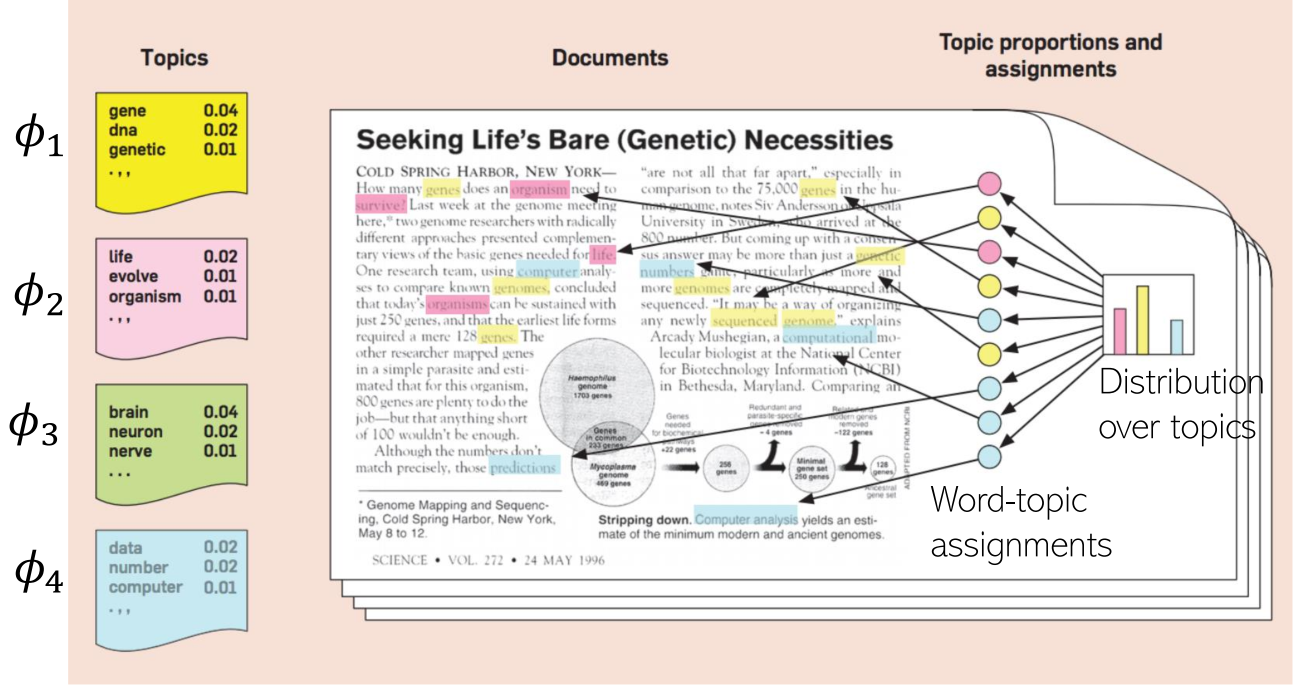
Latent Dirichlet Allocation (LDA)

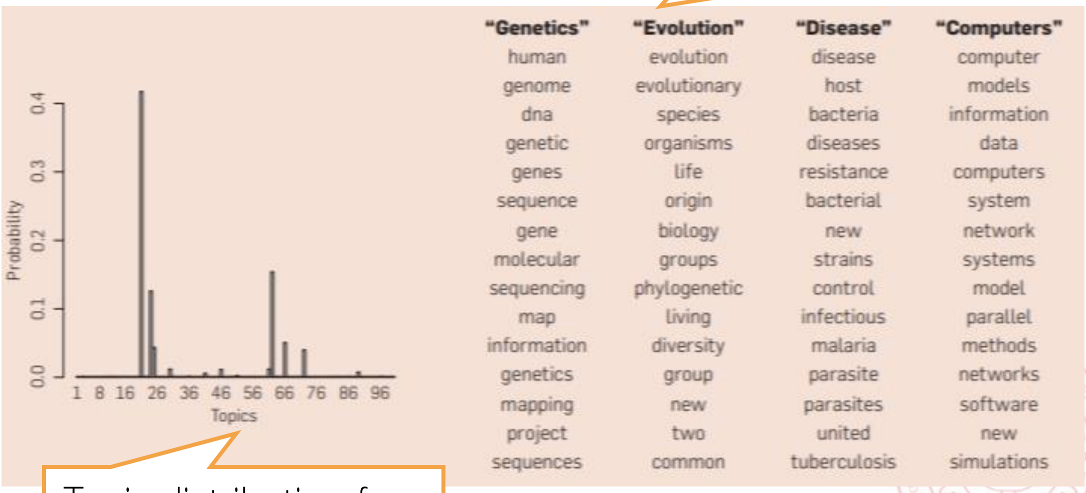


- A very widely used probabilistic model for text data
- Nice and easy insights into the text collection
  - Each $\boldsymbol{\phi}_k = [\phi_{k1}, \dots, \phi_{kV}]$ can be interpreted as topic ($\boldsymbol{\phi}_{kv}$ = prob. of word $\boldsymbol{v}$ in topic $\boldsymbol{k}$)
  - $\boldsymbol{\theta}_d = [\theta_{d1}, \dots, \theta_{dK}]$: how much each topic is present in document $\boldsymbol{d}$ (topic distribution)
  - $\bar{\boldsymbol{z}}_d = \frac{1}{N_d}\sum_{n=1}^{N_d} \boldsymbol{z}_{d,n}$ also has a similar interpretation as $\boldsymbol{\theta}_d$

A topic is a set of words that tend to co-occur together

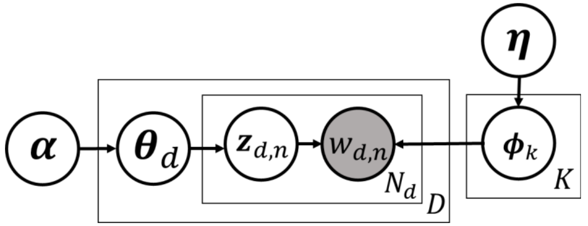
15 most frequent (most probable) words from four most prominent topics in this doc





Topic distribution for the document on left

LDA: Inference and Evaluation



- LDA is locally conjugate. Many inference methods (VI, variational EM, Gibbs samp, etc)

$$p(\mathbf{Z}, \Theta, \Phi | \mathbf{W}, \alpha, \eta) = \frac{p(\mathbf{W}|\Phi, \mathbf{Z}) p(\mathbf{Z}|\Theta) p(\Phi|\eta) p(\Theta|\alpha)}{p(\mathbf{W}|\alpha, \eta)}$$ (assuming hyperparams α, η are fixed)

  - Can even collapse some variables and do collapsed Gibbs or collapsed VB
    - E.g., collapse $\boldsymbol{\theta}_d$ and $\boldsymbol{\phi}_k$ (if needed, these can be approximated using $\mathbf{Z}$)
- Many ways to evaluate how well LDA performs on some data
  - Extrinsic measures: Perform LDA and use its output for another task (e.g., classification)
  - Perplexity is another intrinsic measure to evaluate LDA-style models

Lower is better

Test set with M docs

Marginal likelihood of all words in the d^{th} test doc

$$perplexity(D_{\text{test}}) = \exp\left\{ - \frac{\sum_{d=1}^{M} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d} \right\}$$

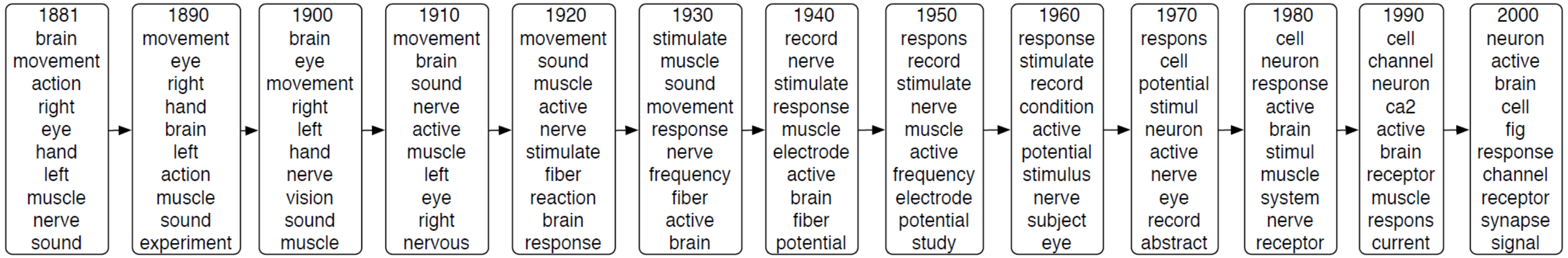
LDA: Limitations and Extensions

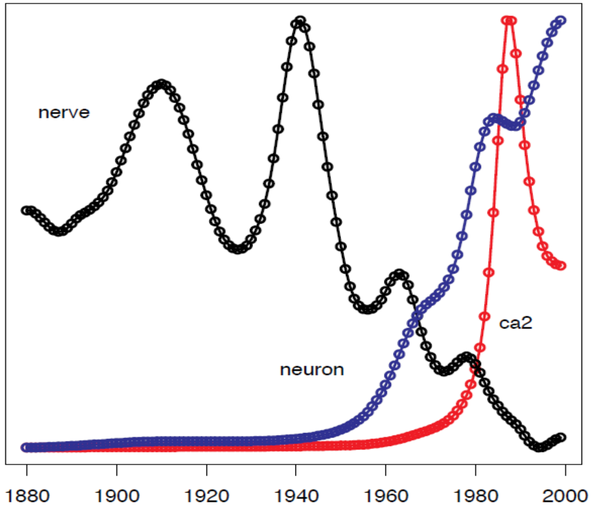
- LDA assumes topics remain static over time (improvement: Dynamic Topic Model)

Assume a first-order Markov evolution for each topic w.r.t. time

$$w_k^t \sim \mathcal{N}(w_k^{t-1}, \sigma^2 I) \qquad \phi_k^t = \mathcal{S}(w_k^t)$$

Simplex transformation (convert w_k^t into a probability vector)





Evolution of topic "Neuroscience" (learned from the journal Science)

- LDA assumes topics are uncorrelated (improvement: Corr-LDA)
  - Use a logistic normal distribution on θ_d (cov matrix of log-normal makes component correlated)
- LDA ignores the sequential structure in the text (improvement: HMM-LDA)

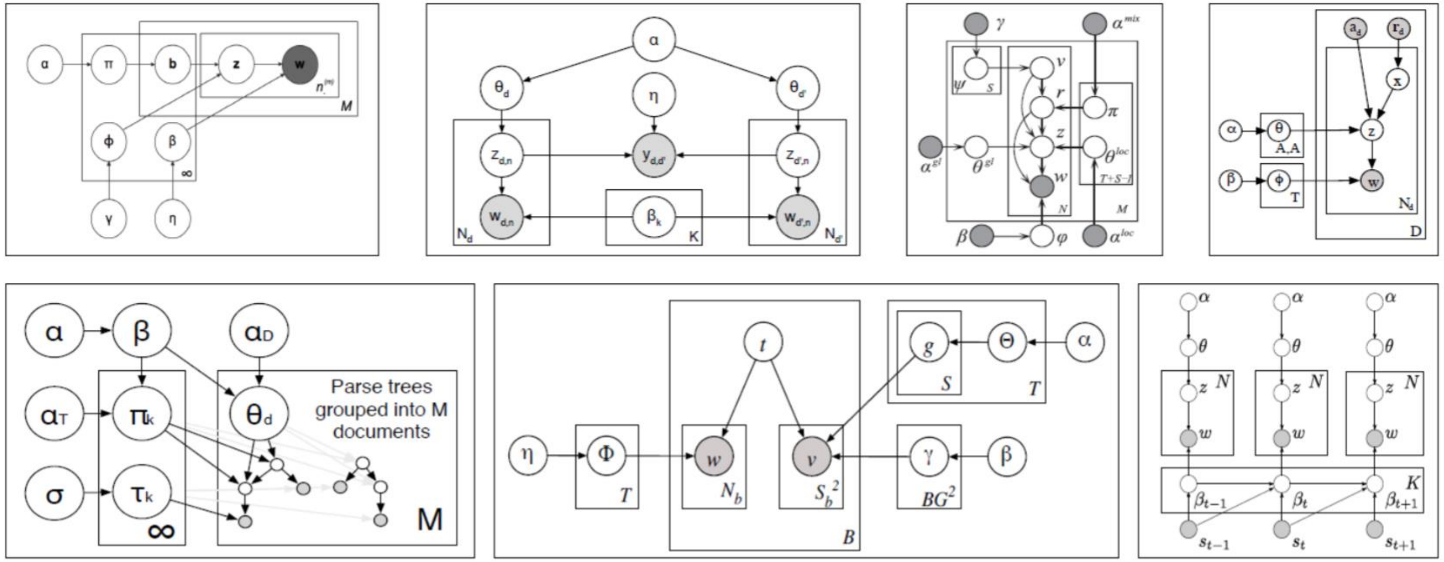
Fig courtesy: Dynamic Topic Models (Blei and Lafferty, 2006)

LDA Extensions (Contd)

Also: "Neural" Topic Models are popular nowadays (z to x mapping and vice-versa modeled via deep nets). Also, some topic models use pre-computed word-embeddings rather than one-hot Representation of each word

- LDA for non-text data, e.g., images
  - Each image can be represented as a bag of "visual words" and LDA can be applied
- Supervised/Labeled LDA (when we have have a label for each document)
- LDA for paired/multimodality data (e.g., images and text caption)
- LDA for graph-structured data instead of documents

Plate diagrams for some LDA extensions



LDA is also equivalent to doing a non-negative matrix fact. of the $V \times D$ word-document matrix $\mathbf{X}$ using a Poisson likelihood model*

$$\mathbf{X} \sim \text{Poisson}(\mathbf{\Phi\Theta})$$

$\mathbf{\Phi}$ $(V \times K)$ and $\mathbf{\Theta}$ $(K \times D)$ can be given any non-negative priors (Dirichlet/gamma)

This can be extended to "deep" matrix factorization** (modeling $\mathbf{\Theta}$ using many layers)

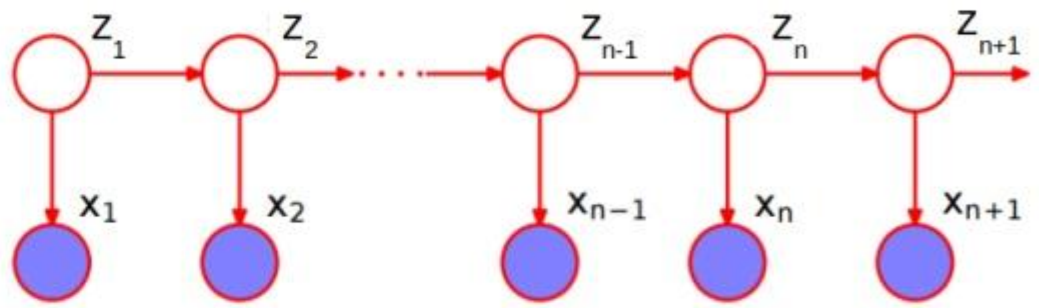
*Sec 4 and 5 of "Beta-Negative Binomial Process and Poisson Factor Analysis" (Zhou et al, 2012)

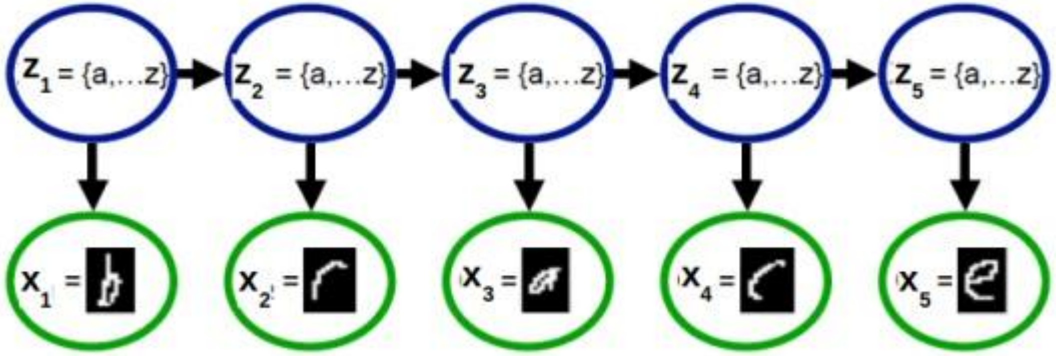
** Poisson-gamma belief networks" (Zhou et al, 2015)

Probabilistic Models for Sequential Data

Latent Variable Models for Sequential Data

- Task: Given a sequence of observations, infer the latent state of each observation





Observation model

$$\boldsymbol{x}_n|\boldsymbol{z}_n \sim p(\boldsymbol{x}_n|\boldsymbol{z}_n) \quad \text{(i.i.d. draws of } \boldsymbol{x}_n \text{ given } \boldsymbol{z}_n)$$

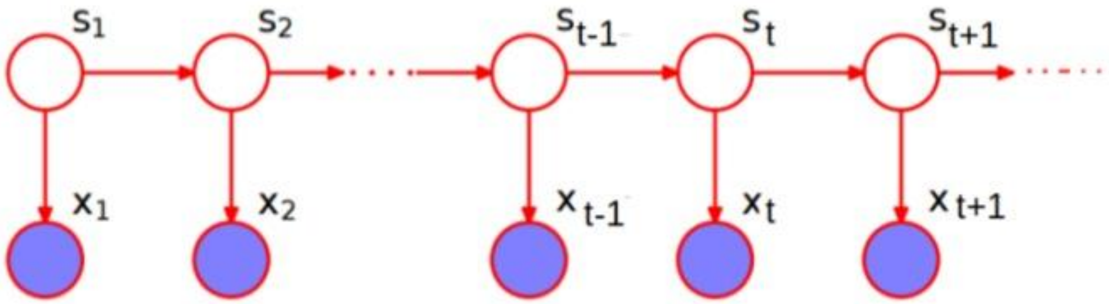
State-transition model

$$\boldsymbol{z}_n|\boldsymbol{z}_{n-1} \sim p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) \quad \text{(first-order dependence b/w } \boldsymbol{z}_n\text{'s)}$$

- If z_n's are discrete, we have a hidden Markov model (HMM) $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}=\ell) = \text{multinoulli}(\boldsymbol{\pi}_\ell)$
- If z_n's are real-valued, we have a state-space model (SSM) $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) = \mathcal{N}(\mathbf{A}\boldsymbol{z}_{n-1}, \mathbf{I}_K)$

State-Space Models

- In the most general form, the state-transition and observation models of an SSM



Using '**s**' instead of '**z**' to refer to states

Using 't' to denote the 'time-step'

HMM is similar to SSM except the state-transition model is a discrete distribution

g_t, h_t can be linear or nonlinear functions

$$\boldsymbol{s}_t|\boldsymbol{s}_{t-1} = g_t(\boldsymbol{s}_{t-1}) + \epsilon_t \quad \text{(must be a cont. dist. over } \boldsymbol{s}_t\text{)}$$

$$\boldsymbol{x}_t|\boldsymbol{s}_t = h_t(\boldsymbol{s}_t) + \delta_t \quad \text{(can be any dist. over } \boldsymbol{x}_t\text{)}$$

- Assuming Gaussian noise in the state-transition and observation models

This is a Gaussian SSM

$$\boldsymbol{s}_t|\boldsymbol{s}_{t-1} \sim \mathcal{N}(\boldsymbol{s}_t|g_t(\boldsymbol{s}_{t-1}), \mathbf{Q}_t)$$

$$\boldsymbol{x}_t|\boldsymbol{s}_t \sim \mathcal{N}(\boldsymbol{x}_t|h_t(\boldsymbol{s}_t), \mathbf{R}_t)$$

If g_t, h_t, Q_t, R_t are independent of t then it is called a stationary model

g_t, h_t, Q_t, R_t may be known or can be learned

State-Space Models: A Simple Example

- Consider the linear Gaussian SSM

$$\boldsymbol{s}_t | \boldsymbol{s}_{t-1} = \mathbf{A}_t \boldsymbol{s}_{t-1} + \epsilon_t$$

$$\mathbf{x}_t | \boldsymbol{s}_t = \mathbf{B}_t \boldsymbol{s}_t + \delta_t$$

- Suppose $\boldsymbol{x}_t \in \mathbb{R}^2$ denotes the (noisy) observed 2D location of an object
- Suppose $\boldsymbol{s}_t \in \mathbb{R}^6$ denotes the "state" vector

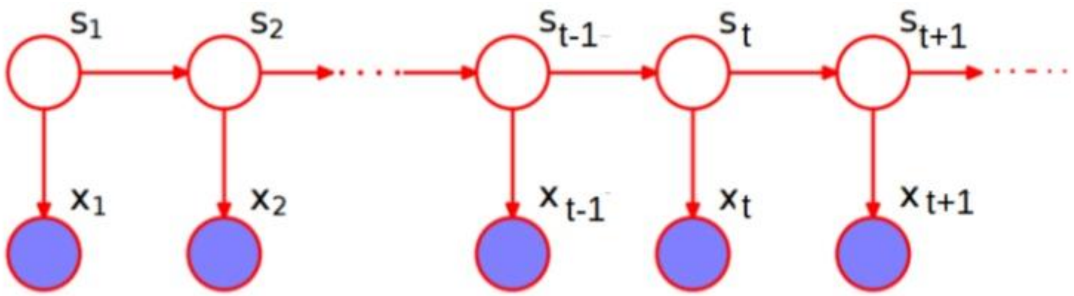
$$\boldsymbol{s}_t = [\text{pos1, vel1, accel1, pos2, vel2, accel2}]$$

- Here is an example SSM for this problem with pre-defined $\mathbf{A_t}$ and $\mathbf{B_t}$ matrices

$$\mathbf{s}_t = \overbrace{\boxed{\begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & e^{-\alpha \Delta t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & e^{-\alpha \Delta t} \end{bmatrix}}}^{\mathbf{A}_t} \mathbf{s}_{t-1} + \epsilon_t \qquad \mathbf{x}_t = \overbrace{\boxed{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}}}^{\mathbf{B}_t} \mathbf{s}_t + \delta_t$$

Typical Inference Task for Gaussian SSM

- One of the key tasks: Given sequence $x_1, x_2, \dots, x_T$, infer latent $s_1, s_2, \dots, s_T$



- Usually two ways of inferring the latent states
  - Infer $p(s_t|x_1, x_2, \dots, x_t)$: Called the "filtering" problem

$$p(\mathbf{s}_t|\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t) \propto \underbrace{p(\mathbf{x}_t|\mathbf{s}_t)}_{\mathcal{N}(\mathbf{x}_t|\mathbf{B}\mathbf{s}_t, \mathbf{R})} \int \underbrace{p(\mathbf{s}_t|\mathbf{s}_{t-1})}_{\mathcal{N}(\mathbf{s}_t|\mathbf{A}\mathbf{s}_{t-1}, \mathbf{Q})} p(\mathbf{s}_{t-1}|\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}) d\mathbf{s}_{t-1}$$

Turns out to be another Gaussian

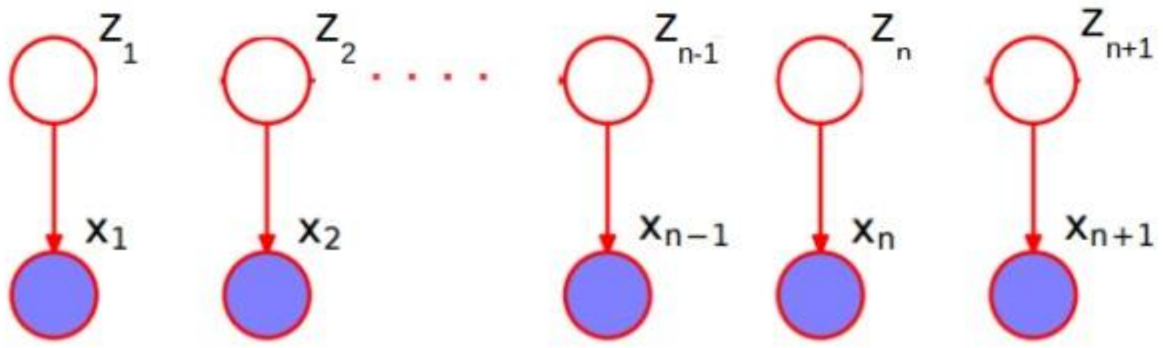
A Gaussian

Kalman Filtering is a popular algorithm for a linear Gaussian SSM

  - Infer $p(s_t|x_1, x_2, \dots, x_t, \dots, x_T)$: Called the "smoothing" problem
- Some other tasks one can solve for using an SSM
  - Predicting future states $p(s_{t+h}|x_1, x_2, \dots, x_t)$ for $h \geq 1$, given observations thus far
  - Predicting future observations $p(x_{t+h}|x_1, x_2, \dots, x_t)$ for $h \geq 1$, given observations thus far

A Special Case

- What if we have i.i.d. latent states, i.e.,. $p(z_n|z_{n-1}) = p(z_n)$?



- Discrete case (HMM) becomes a simple mixture model $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1} = \ell) = p(\boldsymbol{z}_n) = \text{multinoulli}(\boldsymbol{\pi})$
- Real-valued case (SSM) becomes a PPCA model $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) = p(\boldsymbol{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I_K})$ or $\mathcal{N}(\boldsymbol{\mu}, \Psi)$
- Inference algos for HMM/SSM are thus very similar to that of mixture models/PPCA
  - Only main difference is how the latent variables z_n's are inferred since they aren't i.i.d.
  - E.g., if using EM, only E step needs to change (Bishop Chap 13 has EM for HMM and SSM)