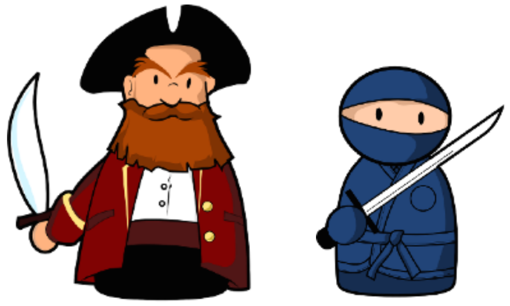
Latent Dirichlet Allocation

Computational Linguistics

Alexander Koller

17 January 2020

with help from Christoph Teichmann
and illustrations by Martín Villalba



Today

- Today's lecture is about a method called *Latent Dirichlet Allocation (LDA).*

- We care about it for two reasons:
  - It's an unsupervised method for identifying *topics* and words that are representative of them.
  - It's a showcase for a family of statistical models called *Bayesian models* which have many uses in CL.

Let's start simple

- You and I are playing a coin-tossing game.
  I see you throw 63x H, 37x T.
  Should I believe that the coin is fair?

- Our model of the coin has one parameter, $p = P(H)$.

- Maximum-likelihood estimate: $p = 0.63$, i.e. not fair.

- But what about
  - my uncertainty about p?
  - my prior beliefs about the fairness of the coin?

Bayesian Models

- ML estimation and similar methods deliver *point estimates:* a single value for each parameter that optimizes some criterion.
    - Likelihood: P(observations | parameters)

- Bayesian models: estimate a *probability distribution* P(parameters | observations) over parameters.
    - uncertainty about parameter values is reflected at all times in the pd

$$\underbrace{P(\text{parameters} \mid \text{observations})}_{\text{posterior}} \propto \underbrace{P(\text{observations} \mid \text{parameters})}_{\text{likelihood}} * \underbrace{P(\text{parameters})}_{\text{prior}}$$

The Dirichlet distribution

- Take the parameter p itself as the value of a random variable.
  - need a probability distribution over real numbers; more specifically, over tuples of numbers that sum to one
- We use the *Dirichlet distribution.*

  $p_1, \ldots, p_K \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_K)$ means:

$$\mathrm{Dir}_{\boldsymbol{\alpha}}(p_1, \ldots, p_K) = \frac{1}{B(\boldsymbol{\alpha})} \cdot p_1^{\alpha_1 - 1} \cdot \ldots \cdot p_K^{\alpha_K - 1}$$

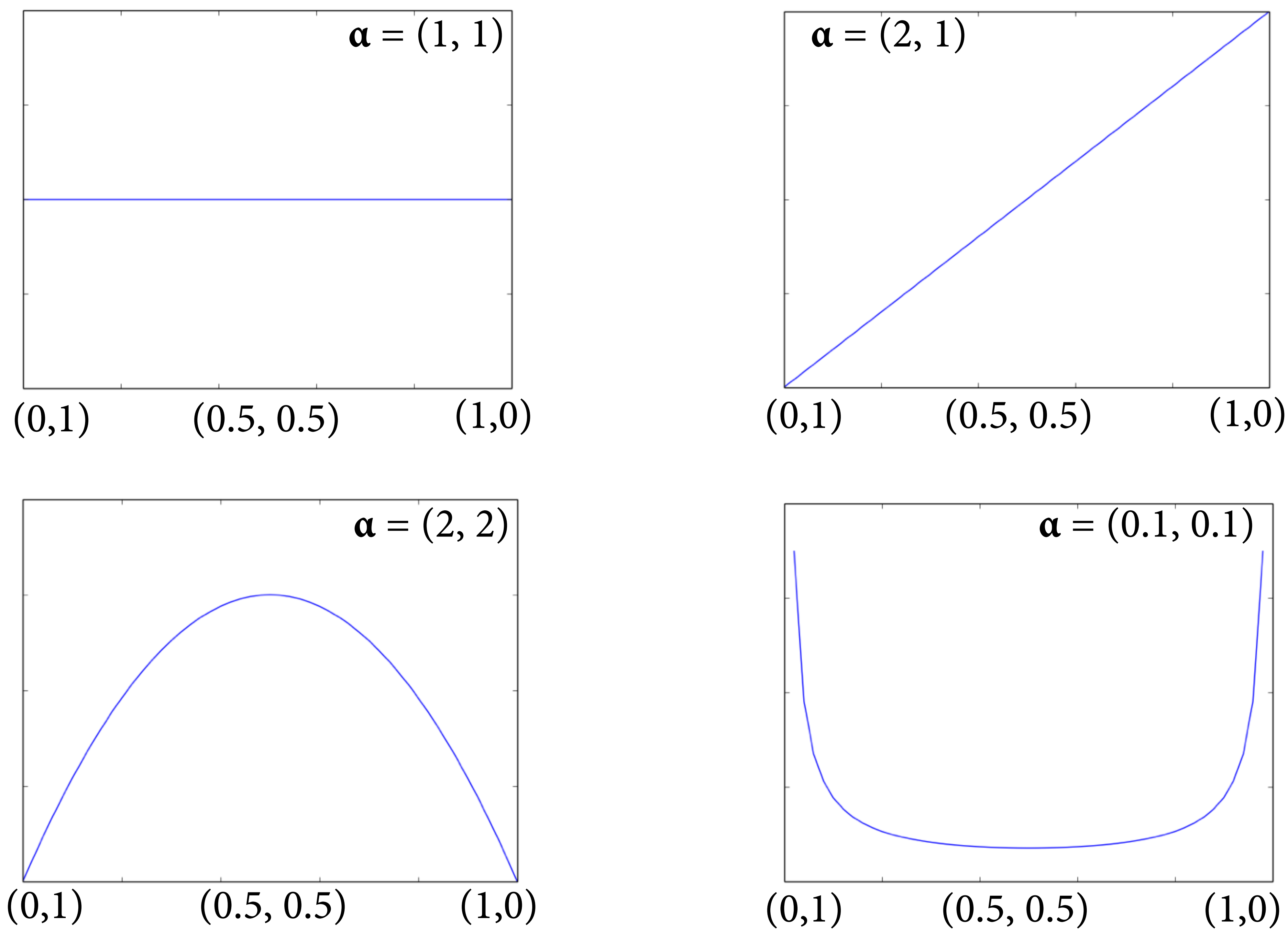
Dir only defined if the p_i sum to 1

this is the *beta function* (needed to normalize to 1)

$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ are called *hyperparameters*

Dirichlet distributions, K = 2

$$\mathrm{Dir}_{\boldsymbol{\alpha}}(p_1, \ldots, p_K) = \frac{1}{B(\boldsymbol{\alpha})} \cdot p_1^{\alpha_1 - 1} \cdot \ldots \cdot p_K^{\alpha_K - 1}$$



Bayesian parameter estimation

- We are interested in pd P(M) over our model M = (p). This model is very simple; will make more complex later.

- Before we make any observations, we have a *prior distribution:* P(M) = $\mathrm{Dir}_{\alpha,\alpha}$(p, 1-p)

- We can then *update* this to a *posterior distribution* based on observed data:

$$P(M \mid D) = \frac{P(D \mid M) \cdot P(M)}{P(D)} \propto P(D \mid M) \cdot P(M)$$

posterior likelihood prior

Calculating posteriors

prior: $P(p) = \mathrm{Dir}_{\alpha,\alpha}(p, 1-p) \propto p^{\alpha-1} \cdot (1-p)^{\alpha-1}$

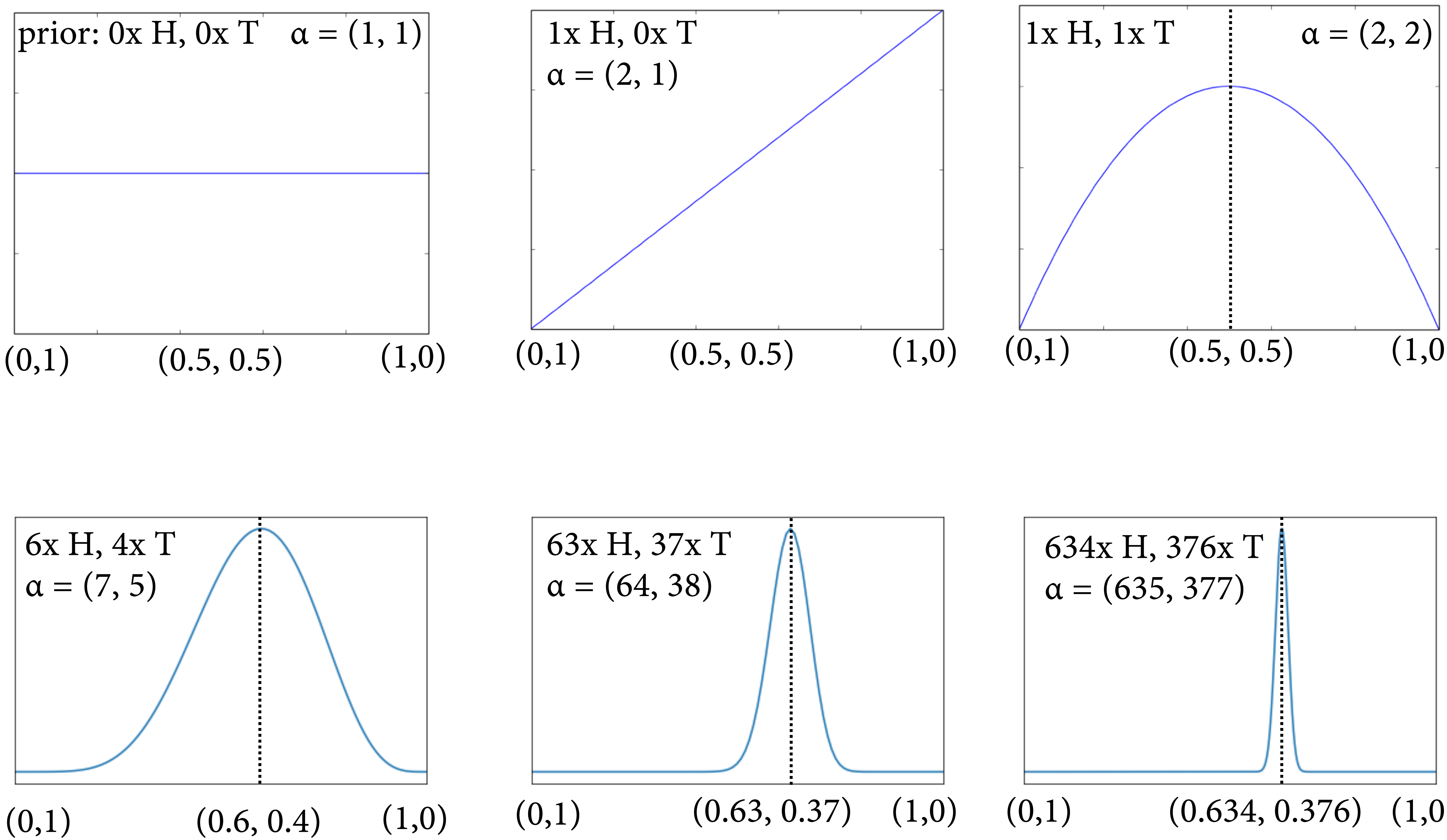
likelihood: $P(i \times \mathrm{H}, k \times \mathrm{T} \mid p) = p^i \cdot (1-p)^k$

posterior:
$$\begin{aligned} P(p \mid i \times \mathrm{H}, k \times \mathrm{T}) &\propto P(i \times \mathrm{H}, k \times \mathrm{T} \mid p) \cdot P(p) \\ &\propto p^i \cdot (1-p)^k \cdot p^{\alpha-1} \cdot (1-p)^{\alpha-1} \\ &= p^{i+\alpha-1} \cdot (1-p)^{k+\alpha-1} \end{aligned}$$

More precisely, we have:

$$P(p \mid i \times \mathrm{H}, k \times \mathrm{T}) = \mathrm{Dir}_{\alpha+i,\alpha+k}(p, 1-p)$$

Informed posteriors



Conjugate distributions

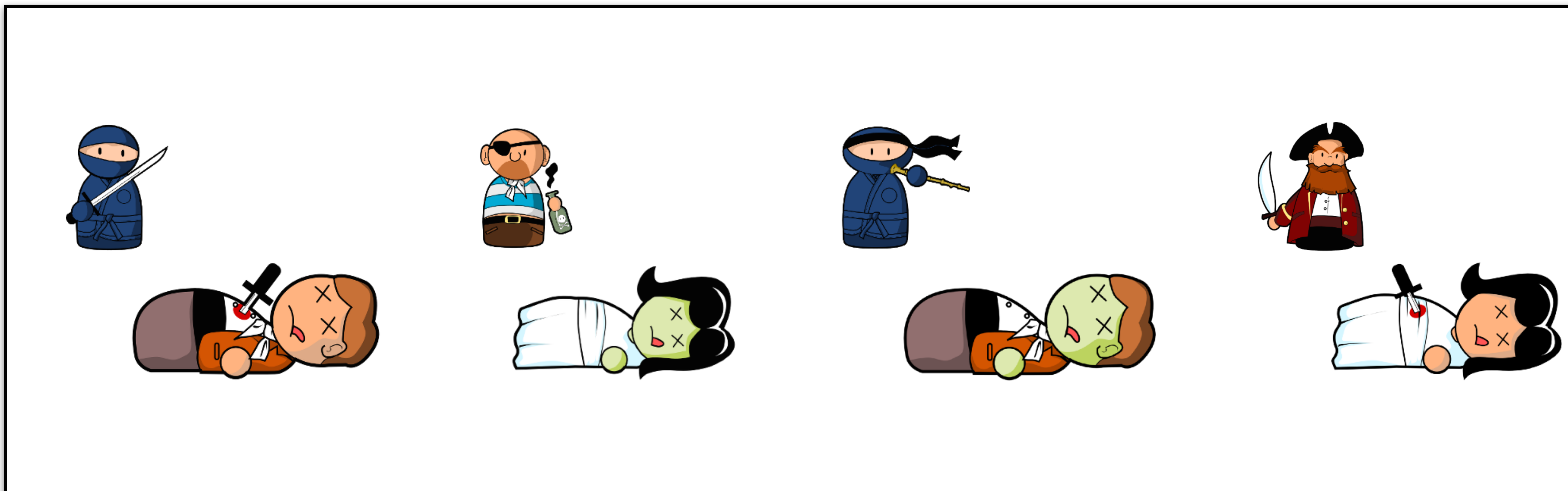
- Crucially, P(M) and P(M | D) have the same shape (product of Dirichlets). This is because Dirichlet and Categorical are *conjugate distributions.*
  - because K = 2 for the coin, we really only used the Beta (not Dirichlet) and Bernoulli (not Categorical) distributions

- This is makes the math very convenient.

- The hyperparameters of the Dirichlets are updated by adding the observed counts to the hp. of the priors.
  - priors thus perform smoothing in a very principled way

The next step

Say you come across some people who have been stabbed or poisoned.
You know that each of them was killed by a pirate or a ninja.
You can tell how each person died, but not by whom they were killed.



Our task

- We observe N people with their causes of death.

- Questions we are interested in:
  - Who killed each villager?
    $z_1, \ldots, z_N \in \{pi, ni\}$
  - How many were killed by pirates, how many by ninjas?
    $P(pi) = \theta_{pi}$, $P(ni) = \theta_{ni}$; thus, $\theta_{pi} + \theta_{ni} = 1$
  - How likely is it that a pirate chooses to stab someone?
    $P(st \mid pi) = \phi_{st|pi}$; thus, $P(po \mid pi) = \phi_{po|pi} = 1 - \phi_{st|pi}$
  - How likely is it that a ninja chooses to stab someone?
    $P(st \mid ni) = \phi_{st|ni}$; thus, $P(po \mid ni) = \phi_{po|ni} = 1 - \phi_{st|ni}$

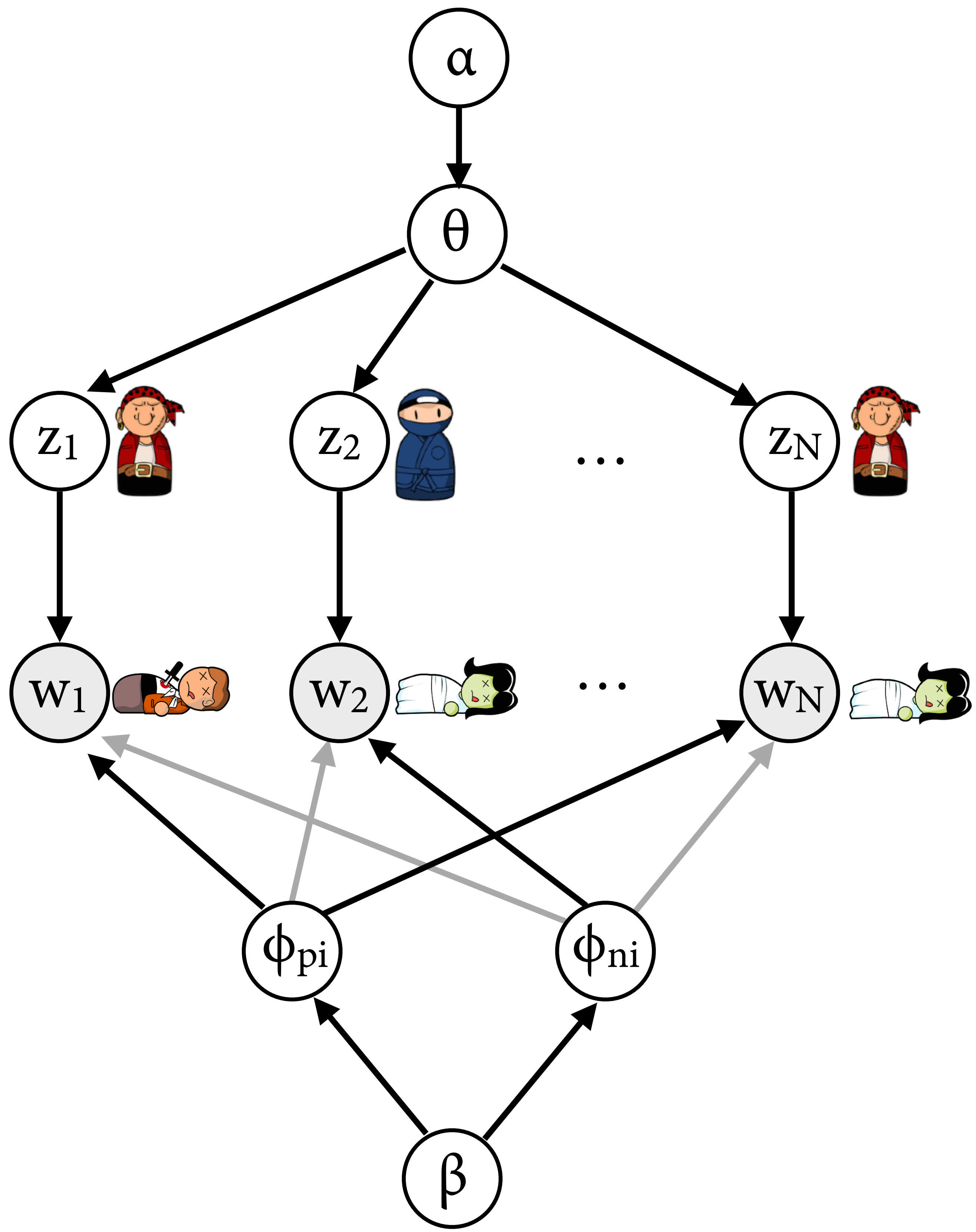
Fundamental approach

- Goal: Bayesian model with parameters θ, ϕ_{pi}, ϕ_{ni}.
  - maximum likelihood: try to estimate concrete values for each parameter
  - Bayesian: estimate *probability distribution* $P(\theta, \phi_{pi}, \phi_{ni})$

- In practice, the model will have *latent variables* z, which cannot be observed directly (e.g. pirate/ninja).

- Will marginalize over model parameters and work with P(z | observations) directly.

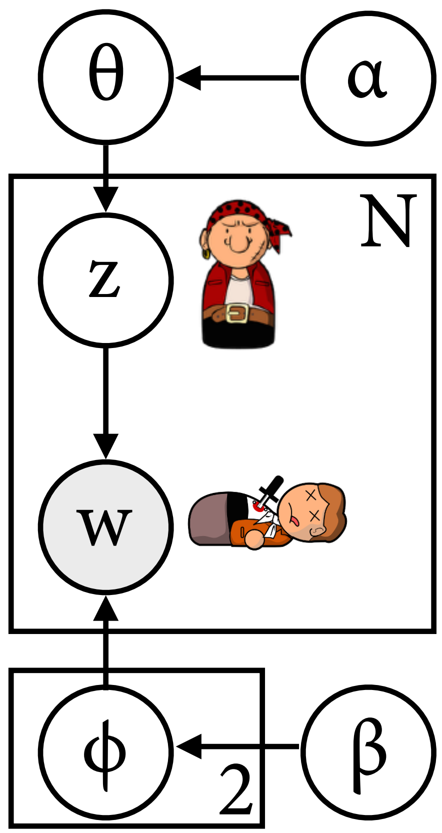
Generative story: Idea



Generative story

- We assume deaths are generated as follows:

$(\theta_{pi}, \theta_{ni}) \sim \mathrm{Dir}(\alpha, \alpha)$
$(\phi_{st|pi}, \phi_{po|pi}), (\phi_{st|ni}, \phi_{po|ni}) \sim \mathrm{Dir}(\beta, \beta)$
$z_1, \ldots, z_K \sim \mathrm{Categorical}(\theta)$
$w_i \sim \mathrm{Categorical}(\phi_{z_i})$

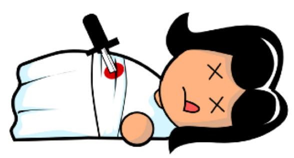


- That is:
  - $P(z_i = pi) = \theta_{pi}$, $P(z_i = ni) = \theta_{ni}$
  - if z_i came out as "pi", then $P(w_i = st) = \phi_{st|pi}$

I abbreviate $\theta = (\theta_{pi}, \theta_{ni})$, $\phi_{pi} = (\phi_{st|pi}, \phi_{po|pi})$, $\phi_{ni} = (\phi_{st|ni}, \phi_{po|ni})$.
α, β are assumed given and are called *hyperparameters.*

Supervised learning

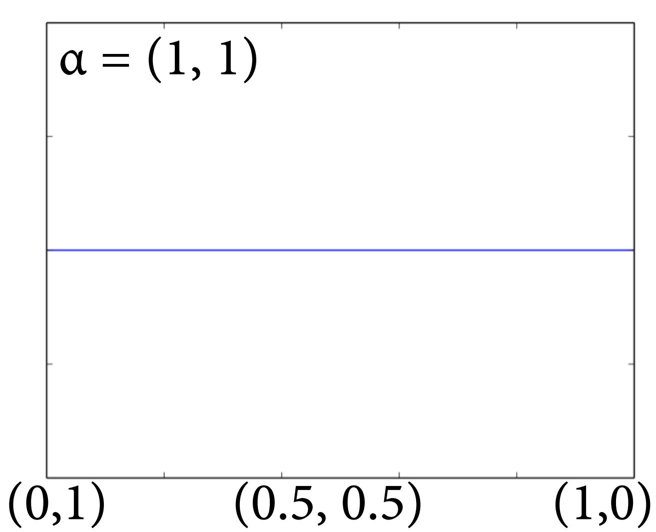
If all killers are known, P(M | D) is easy to compute.

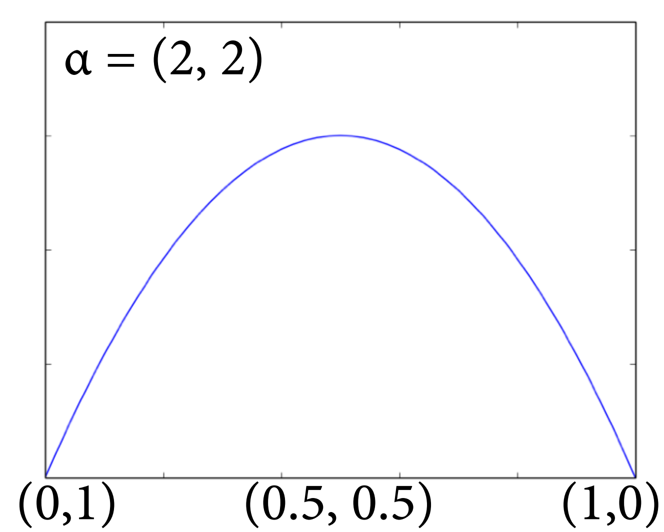
i	z_i	w_i
1		
2		

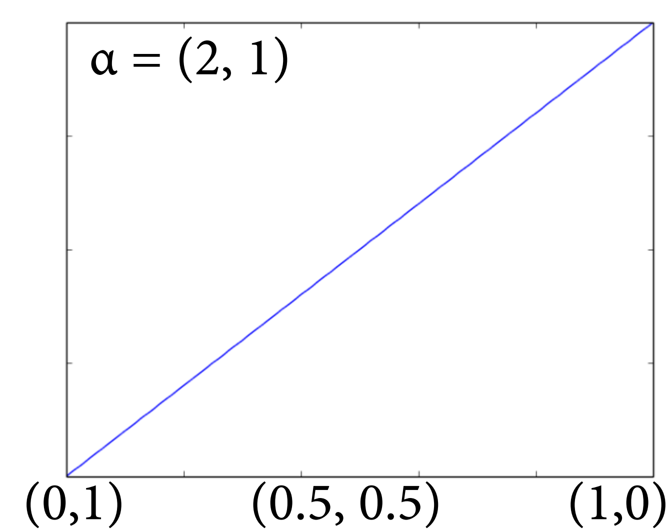
$$P(M) = \mathrm{Dir}_{\alpha,\alpha}(\theta)\cdot \mathrm{Dir}_{\beta,\beta}(\phi_{\mathrm{pi}})\cdot \mathrm{Dir}_{\beta,\beta}(\phi_{\mathrm{ni}})$$
$$\propto \theta_{\mathrm{pi}}^{\alpha-1}\cdot\theta_{\mathrm{ni}}^{\alpha-1}\cdot\phi_{\mathrm{st|pi}}^{\beta-1}\cdot\phi_{\mathrm{po|pi}}^{\beta-1}\cdot\phi_{\mathrm{st|ni}}^{\beta-1}\cdot\phi_{\mathrm{po|ni}}^{\beta-1}$$

$$P(D\mid M) = P(z_1=\mathrm{pi}, w_1=\mathrm{st}, z_2=\mathrm{ni}, w_2=\mathrm{po})$$
$$=\theta_{\mathrm{pi}}\cdot\phi_{\mathrm{st|pi}}\cdot\theta_{\mathrm{ni}}\cdot\phi_{\mathrm{po|ni}}$$

$$P(M\mid D)\propto P(D\mid M)\cdot P(M)$$
$$\propto \theta_{\mathrm{pi}}^{\alpha}\cdot\theta_{\mathrm{ni}}^{\alpha}\cdot\phi_{\mathrm{st|pi}}^{\beta}\cdot\phi_{\mathrm{po|pi}}^{\beta-1}\cdot\phi_{\mathrm{st|ni}}^{\beta-1}\cdot\phi_{\mathrm{po|ni}}^{\beta}$$
$$\propto \mathrm{Dir}_{\alpha+1,\alpha+1}(\theta)\cdot\mathrm{Dir}_{\beta+1,\beta}(\phi_{\mathrm{pi}})\cdot\mathrm{Dir}_{\beta,\beta+1}(\phi_{\mathrm{ni}})$$

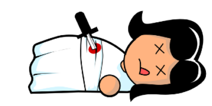






Unsupervised learning

- In the original scenario, we can only observe deaths, not killers. Then P(D | M) is less convenient:

i	z_i	w_i
1	??	
2	??	

$$P(D \mid M) = P(w_1 = \text{st}, w_2 = \text{po} \mid M)$$
$$= \sum_{k_1, k_2 \in \{\text{pi}, \text{ni}\}} P(z_1 = k_1, w_1 = \text{st}, z_2 = k_2, w_2 = \text{po} \mid M)$$

- This sums over a number of terms that is exponential in N, and thus infeasible to compute.

- $M = (\theta, \phi_{pi}, \phi_{ni})$

Latent variables

- Many interesting quantities can be expressed in terms of expected values over the latent variables.

$$P(z \mid w) = \int P(z, M \mid w)\, \mathrm{d}M = \int P(z \mid M, w) \cdot P(M \mid w)\, \mathrm{d}M$$

- Some examples:

ninja/pirate mixing proportion

$$\frac{1}{N} \cdot E_{P(z|w)}[C(z_i = \text{ninja})]$$

pirate habits

$$E_{P(z|w)}[C(z_i = \text{pirate}, w_i = \text{stab})] \;/\; E_{P(z|w)}[C(z_i = \text{pirate})]$$

probability that first villager was killed by a pirate

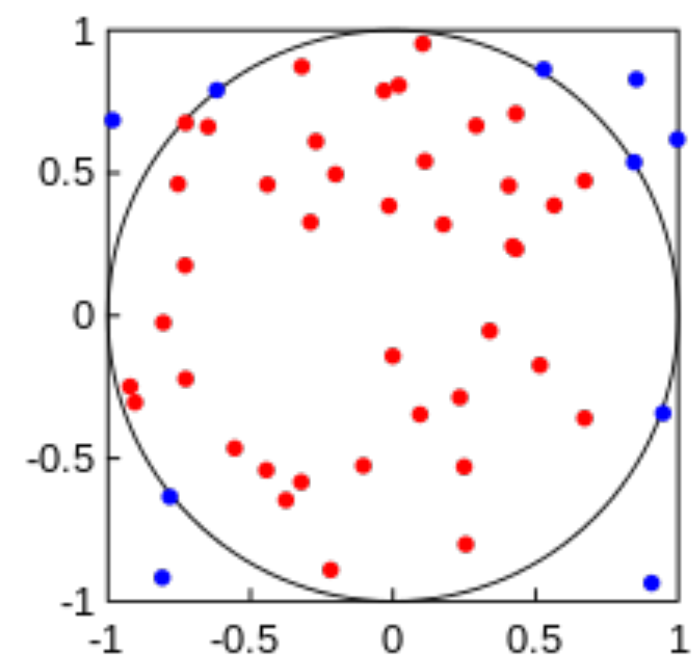
$$E_{P(z|w)}[\ \|z_1 = \text{pirate}\|\ ]$$

Estimating expected values

- Expected values can be approximated by *sampling*. To compute $E_{P(X)}[f(X)]$:
  - draw S samples $x^{(1)}, \ldots, x^{(S)}$ from P(X)
  - estimate $E[f(X)] \approx \frac{1}{S} \cdot \sum_{i=1}^{S} f(x^{(i)})$

- Example: To estimate π, sample points from square and count how many fall into the circle.

$$\pi/4 \approx E_{P(x,y)}[\ \|x^2 + y^2 \leq 1\|\ ]$$

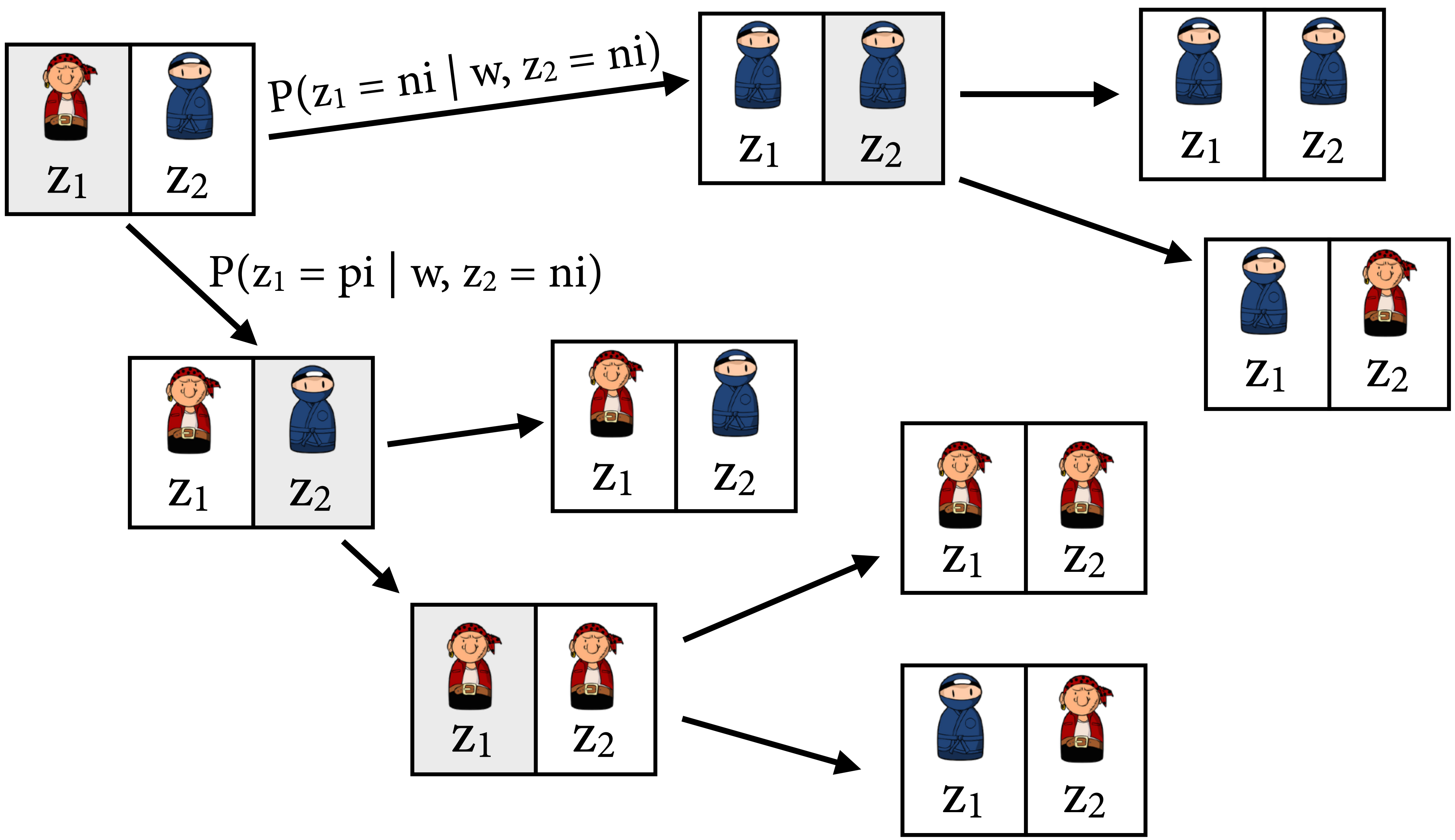


EVs under latent variables

- We could estimate expected values under $P(z \mid w)$ using sampling. However, $P(z \mid w)$ is usually of a form that makes direct sampling difficult.

- Instead, we can use *Gibbs sampling*:
  - Start from an initial guess $z_1, \ldots, z_N$ for the latent variables.
  - Repeatedly resample guess for some z_i conditioned on all other z's, i.e. from $P(z_i \mid w, z_{-i})$. This is much easier than sampling from $P(z \mid w)$ itself.
  - Can prove that probability of observing a sample for z as a whole converges to $P(z \mid w)$.

Gibbs Sampling



Transition probabilities

- It remains to determine the transition probabilities $P(z_i \mid w, z_{-i})$.

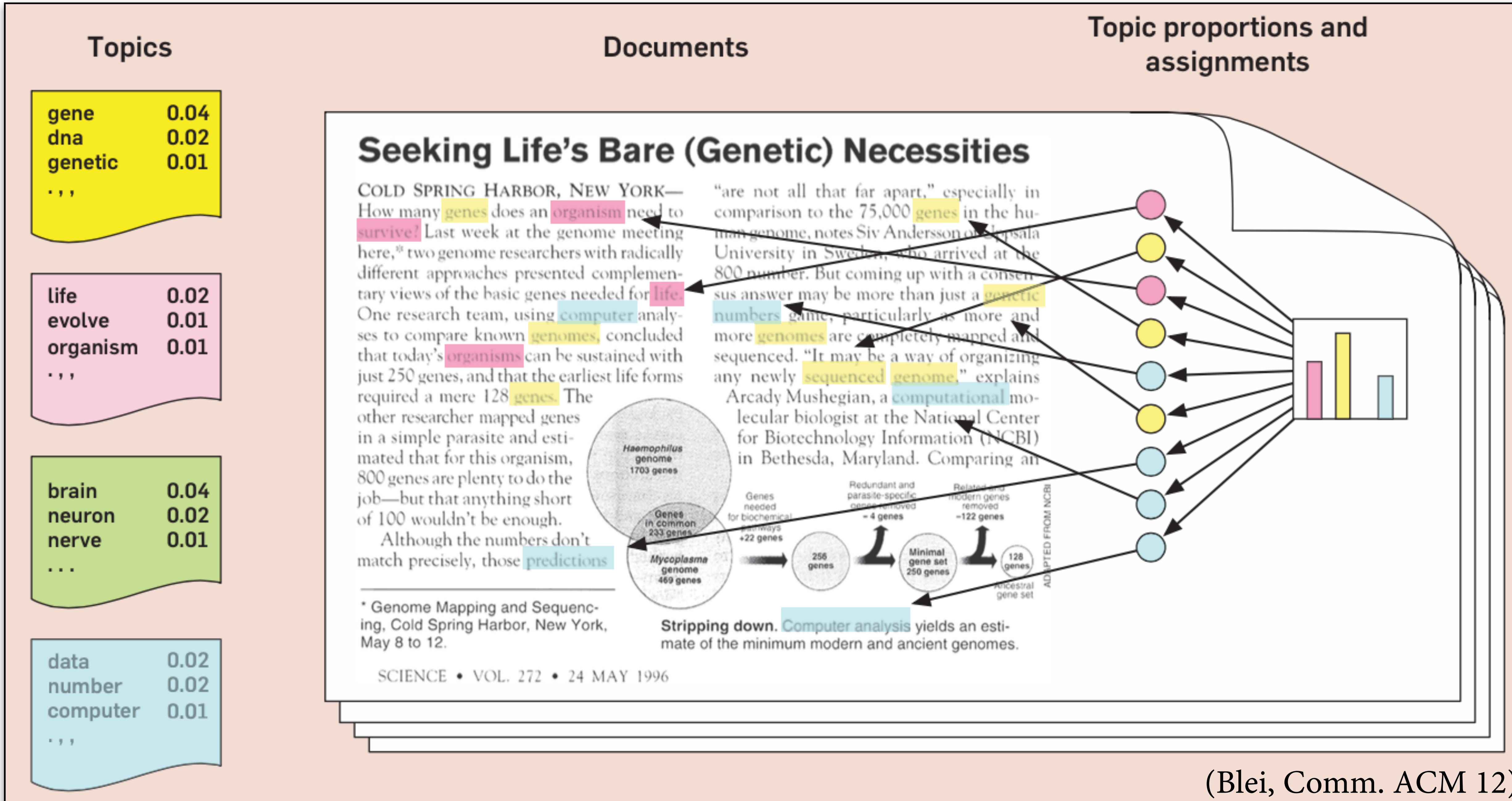
- Formula turns out to be remarkably simple:

$$
\begin{aligned}
P(z_i = \mathrm{pi} \mid w, z_{-i}) &\propto P(w, z_{-i}, z_i = \mathrm{pi}) \\
&= \int\int P(w, z_{-i}, z_i = \mathrm{pi}, \theta, \phi)\, \mathrm{d}\theta\, \mathrm{d}\phi \\
&= \dots \\
&\propto (n_{\mathrm{pi}}^{(-i)} + \alpha_{\mathrm{pi}}) \frac{n_{\mathrm{pi},w_i}^{(-i)} + \beta_{w_i|\mathrm{pi}}}{\sum_{w'} n_{\mathrm{pi},w'}^{(-i)} + \beta_{w'|\mathrm{pi}}}
\end{aligned}
$$

\# people other than i that were killed by pirates in current sample

\# people other than i that were killed by pirates using method w'

Topic models



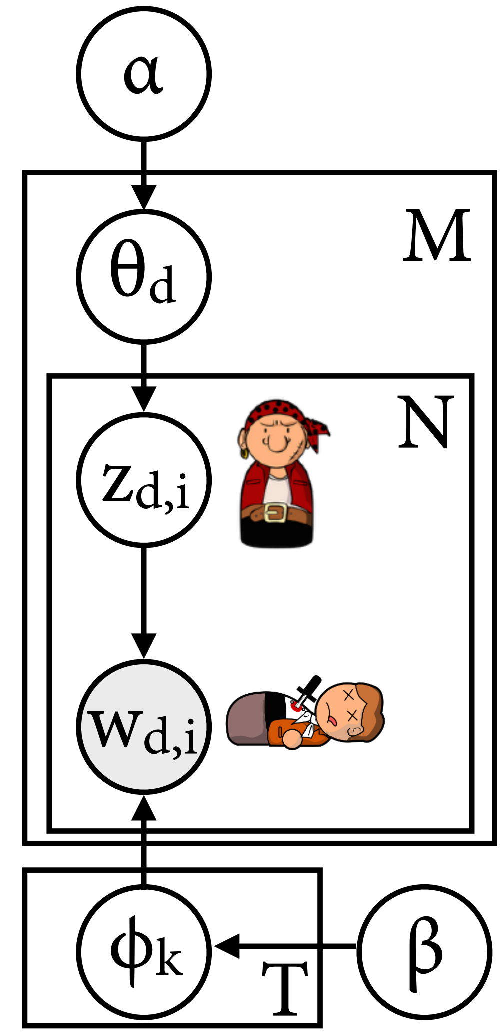
(Blei, Comm. ACM 12)

learn: word probs. for (abstract) *topics* ⟵ given: raw documents ⟶ learn: topic mixture in each document

Latent Dirichlet Allocation

- Topic modeling is almost the same problem as the pirate/ninja problem:
  - abstract topics = {pirate, ninja}
  - words in document = {stabbed, poisoned}

- Full LDA makes two changes:
  - can have T topics instead of just two, and also more than two different words
  - there are M > 1 *documents,* and each document can have its own mixture θ_d of topics



Gibbs sampler for LDA

prob of reassigning token #i as topic t

t occurs with word w_i except at position i

t occurs in document that contains position i, except at position i

$$P(z_i = t \mid z_{-i}, w) \propto \frac{n^{(w_i)}_{-i,t} + \beta}{n^{(\cdot)}_{-i,t} + W \cdot \beta} \cdot \frac{n^{(d_i)}_{-i,t} + \alpha}{n^{(d_i)}_{-i,\cdot} + T \cdot \alpha}$$

t occurs anywhere in corpus, except at position i

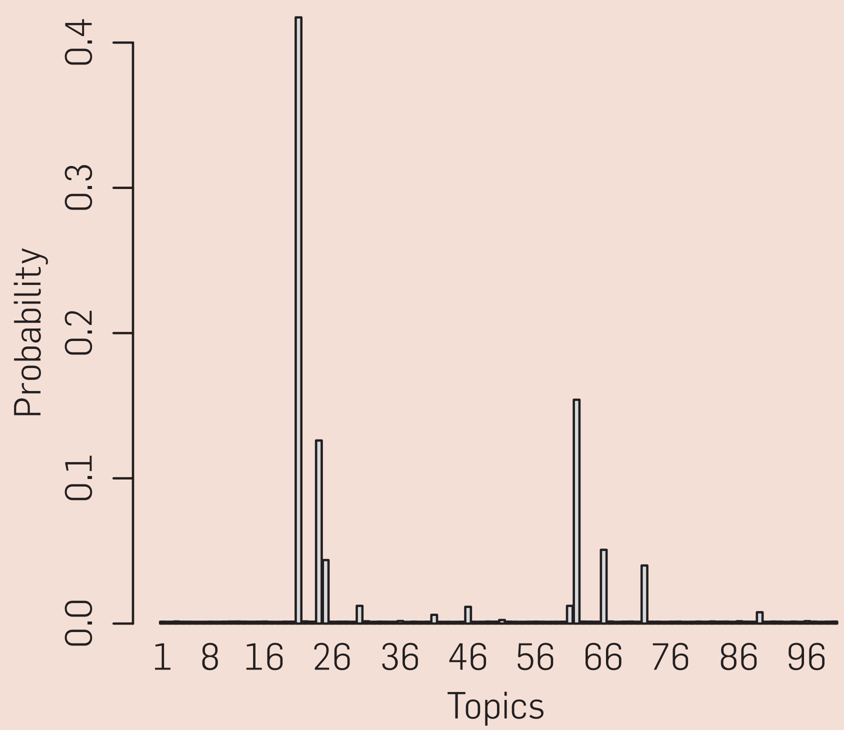
tokens in that document, minus one (for position i)

W = vocabulary size / T = number of topics

(Griffiths & Steyvers 2004)

Examples

(Blei 2012)



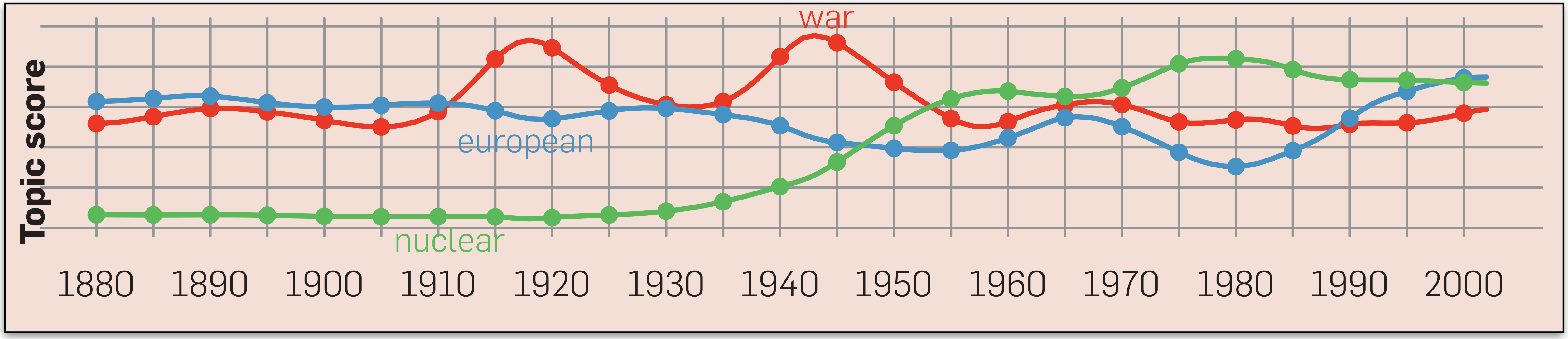
“Genetics”	**“Evolution”**	**“Disease”**	**“Computers”**
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

topic mixture for
one article in *Science*

15 words with highest $\phi_{k,w}$
for each topic over whole corpus
(with made-up topic label)

Examples

development of topics from *Science* over time (1880-2002)



(Blei 2012)

Conclusion

- LDA and extensions for topic modeling.
  - Topics interesting in their own right, also useful in various applications.
  - Simplest useful Bayesian model in NLP.
- We used (collapsed) Gibbs sampling to approximate expected values.
  - Alternative is *Variational Bayes:* approximate P(M|D) on paper, then solve integral exactly.
- Limitation: Number T of topics must be given. We will fix this next time.