TTIC 31210:
Advanced Natural Language Processing

Kevin Gimpel

Spring 2019

Lecture 13:
Introduction to Bayesian NLP

Roadmap

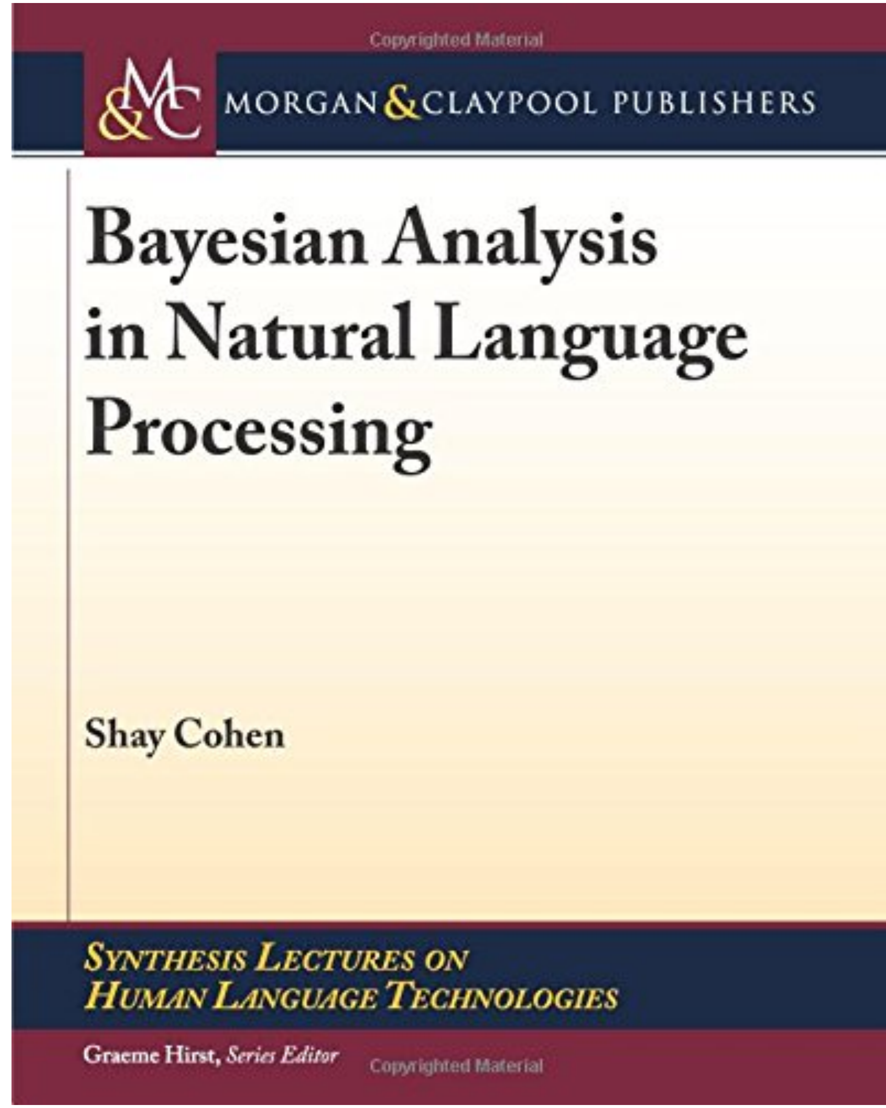
- intro (1 lecture)
- deep learning for NLP (5 lectures)
- structured prediction (4.5 lectures)
- generative models, latent variables, unsupervised learning, variational autoencoders (1.5 lectures)
- **Bayesian methods in NLP (2 lectures)**
- Bayesian nonparametrics in NLP (2 lectures)
- review & other topics (1 lecture)

Assignments

- any questions about Assignment 3?

Additional Reading

- for this segment of the course, the optional text is Cohen (2016, 2019)
- there is a copy of the second edition (2019) in the TTIC library
- readings will be drawn from this book for the next few lectures



Motivation

- in most neural NLP, we assume parameters and architectures are fixed
- consider a one-hidden-layer MLP:

$$p(Y = y \mid \boldsymbol{x}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

Motivation

- in most neural NLP, we assume parameters and architectures are fixed
- consider a one-hidden-layer MLP:

$$p(Y = y \mid \boldsymbol{x}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

- now let's be more explicit about what we're conditioning on:

$$p(Y = y \mid \boldsymbol{x}, \Theta = \{\mathbf{w}, \mathbf{W}\}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

Motivation

- in most neural NLP, we assume parameters and architectures are fixed
- consider a one-hidden-layer MLP:

$$p(Y = y \mid \boldsymbol{x}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

- now let's be more explicit about what we're conditioning on:

$$p(Y = y \mid \boldsymbol{x}, \Theta = \{\mathbf{w}, \mathbf{W}\}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

note: the notation above suggests that we can think of parameters as random variables; this is not uncontroversial.

Motivation

$$p(Y = y \mid \boldsymbol{x}, \Theta = \{\mathbf{w}, \mathbf{W}\}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

- how do we get back to $p(Y = y \mid \boldsymbol{x})$?

Motivation

$$p(Y = y \mid \boldsymbol{x}, \Theta = \{\mathbf{w}, \mathbf{W}\}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

- how do we get back to $p(Y = y \mid \boldsymbol{x})$?
- marginalize over new random variables:

$$p(Y = y \mid \boldsymbol{x}) = \int_{\Theta} p(Y = y, \Theta = \{\mathbf{w}, \mathbf{W}\} \mid \boldsymbol{x}) d\Theta$$

Motivation

$$p(Y = y \mid \boldsymbol{x}, \Theta = \{\mathbf{w}, \mathbf{W}\}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

- how do we get back to $p(Y = y \mid \boldsymbol{x})$?
- marginalize over new random variables:

$$p(Y = y \mid \boldsymbol{x}) = \int_{\Theta} p(Y = y, \Theta = \{\mathbf{w}, \mathbf{W}\} \mid \boldsymbol{x}) d\Theta$$

$$p(Y = y, \Theta = \{\mathbf{w}, \mathbf{W}\} \mid \boldsymbol{x}) = p(Y = y \mid \Theta = \{\mathbf{w}, \mathbf{W}\}, \boldsymbol{x})\, p(\Theta = \{\mathbf{w}, \mathbf{W}\} \mid \boldsymbol{x})$$

Motivation

$$p(Y = y \mid \boldsymbol{x}, \Theta = \{\mathbf{w}, \mathbf{W}\}) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

- how do we get back to $p(Y = y \mid \boldsymbol{x})$?
- marginalize over new random variables:

$$p(Y = y \mid \boldsymbol{x}) = \int_\Theta p(Y = y, \Theta = \{\mathbf{w}, \mathbf{W}\} \mid \boldsymbol{x}) d\Theta$$

- intuitively: don't commit to a single set of parameter values; use them all (with a suitable prior distribution)

Going Further…

- marginalize over architectures & parameters:

$$p(Y = y \mid \boldsymbol{x}, \Lambda = \mathrm{MLP}(\mathbf{w}, \mathbf{W})) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

$$p(Y = y \mid \boldsymbol{x}) = \int_\Lambda p(Y = y, \Lambda = \mathrm{MLP}(\mathbf{w}, \mathbf{W}) \mid \boldsymbol{x}) d\Lambda$$

Going Further…

- marginalize over architectures & parameters:

$$p(Y = y \mid \boldsymbol{x}, \Lambda = \mathrm{MLP}(\mathbf{w}, \mathbf{W})) = \frac{\exp\{\mathbf{w}_y^\top \tanh\left(\mathbf{W} g(\boldsymbol{x})\right)\}}{Z}$$

$$p(Y = y \mid \boldsymbol{x}) = \int_\Lambda p(Y = y, \Lambda = \mathrm{MLP}(\mathbf{w}, \mathbf{W}) \mid \boldsymbol{x}) d\Lambda$$

- the Bayesian framework gives us a vocabulary to discuss this kind of thing and methods for approximating these computations

Why "Bayesian"?

Likelihood
Probability of collecting this data when our hypothesis is true

Bill Howe, UW

Prior
The probability of the hypothesis being true before collecting data

$$P(H|D) = \frac{P(D|H)\,P(H)}{P(D)}$$

Posterior
The probability of our hypothesis being true given the data collected

Marginal
What is the probability of collecting this data under all possible hypotheses?

Bayesian NLP

- typically used with unsupervised learning:
  - we have data
  - we hypothesize some latent variables through which the data are generated
  - we define the “generative story” that describes how latent variables are generated, then how data is generated using latent variables
  - **new**: we parameterize the distributions & add the parameters themselves to the generative story

Generative Story Template

1: Draw a set of parameters θ from $p(\Theta)$

2: Draw a latent structure z from $p(Z \mid \theta)$

3: Draw the observed data x from $p(X \mid z, \theta)$

the above generative story implies the following factorization of the joint distribution:

$$p(x, z, \theta) = p(\theta)p(z \mid \theta)p(x \mid z, \theta)$$

Categorical Distribution

- parameterized by a vector of probabilities, one for drawing each outcome
- i.e., prob. of drawing outcome *i* for variable *X*:

$$p(X = x_i \mid \theta) = \theta_i \qquad\qquad i \in \{1, \ldots, K\}$$

- when we want to draw from this distribution, we will write:

$$x \sim \text{Categorical}(\theta)$$

Categorical vs. Multinomial

- “multinomial” is used frequently to mean categorical in this literature, so we’ll use them interchangeably

- a multinomial is actually more general (permits more than 1 instance of an event)

Vector Form of Categorical Distribution

- form we saw earlier:

$$p(X = x_i \mid \theta) = \theta_i \qquad i \in \{1, \ldots, K\}$$

- we can also write the categorical distribution as a (one-hot) vector random variable *Y*:

$$Y_i = \mathbb{I}[X = i] \qquad\qquad Y \in \{0, 1\}^K$$

$$p(Y = y \mid \theta) = \text{ ?}$$

Vector Form of Categorical Distribution

- form we saw earlier:

$$p(X = x_i \mid \theta) = \theta_i \qquad i \in \{1, \ldots, K\}$$

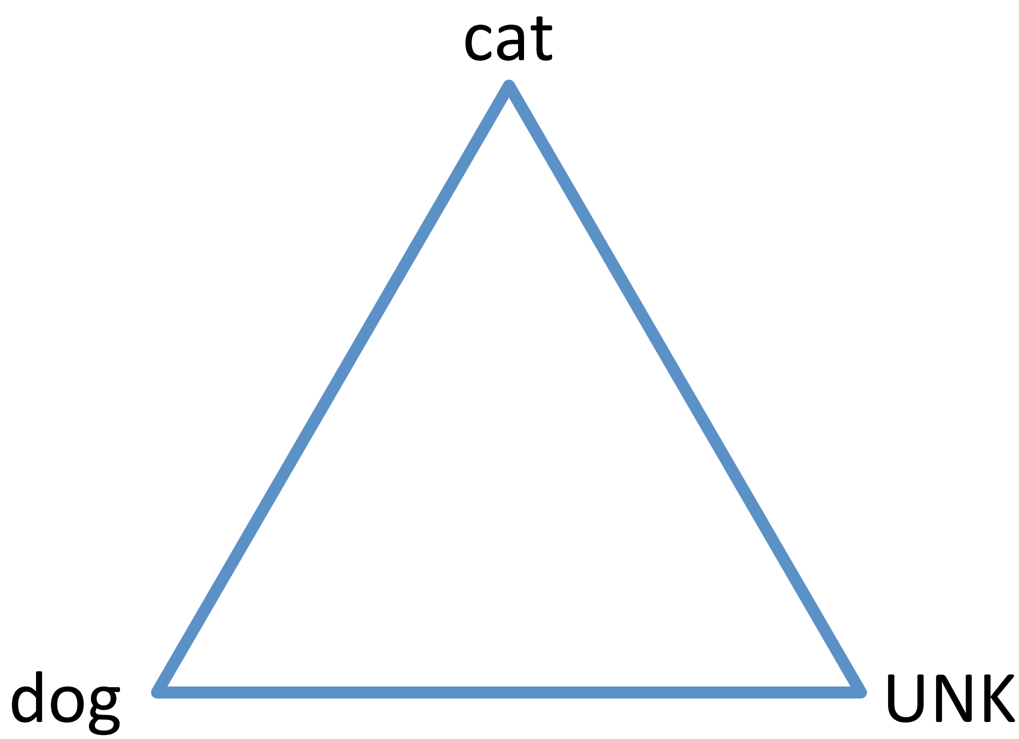
- we can also write the categorical distribution as a (one-hot) vector random variable *Y*:

$$Y_i = \mathbb{I}[X = i] \qquad\qquad Y \in \{0, 1\}^K$$

$$p(Y = y \mid \theta) = \prod_{i=1}^{K} \theta_i^{y_i}$$

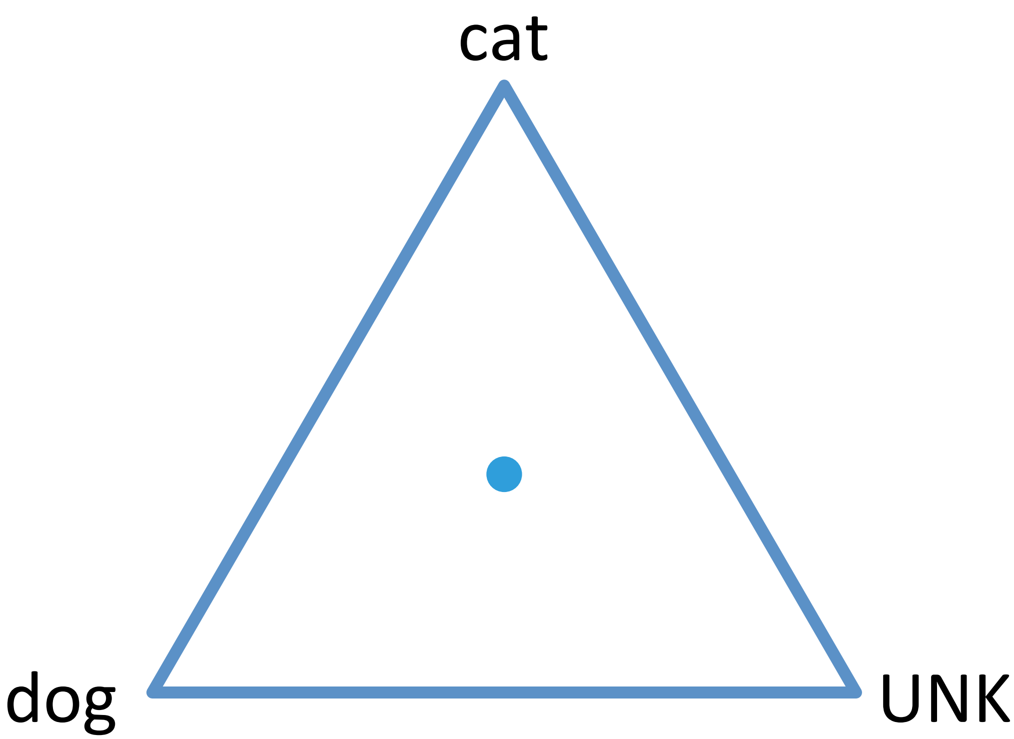
The 2-Simplex

- consider a categorical distribution with 3 outcomes
- e.g., a distribution over words using a vocabulary of size 3:



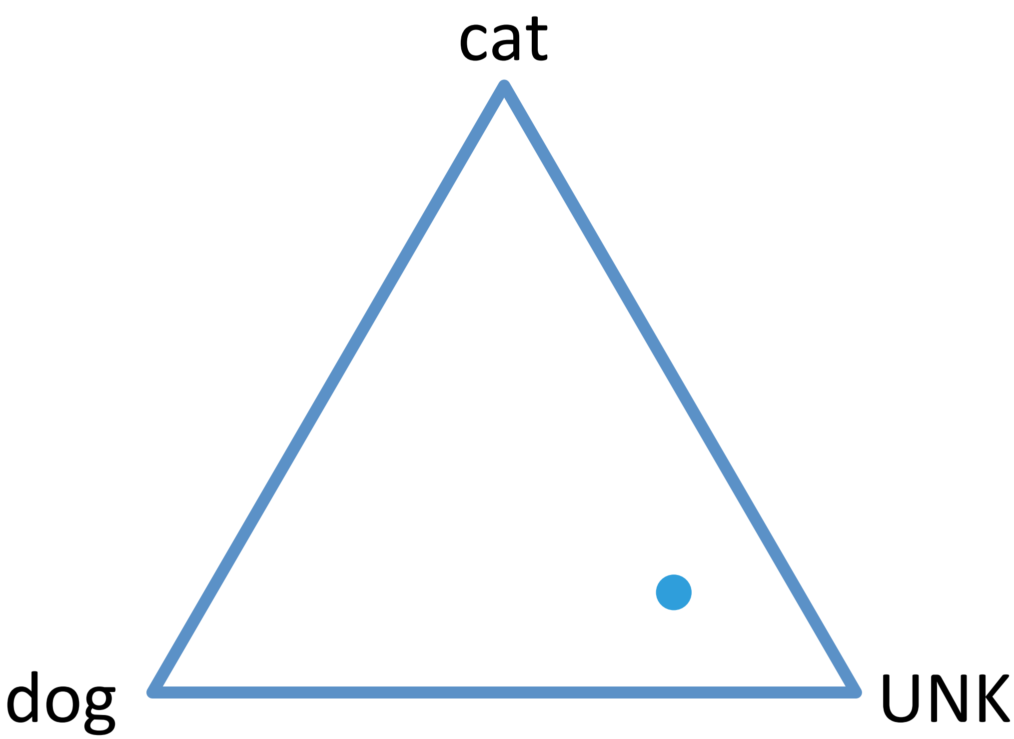
The 2-Simplex

- a point on this simplex represents a categorical distribution over the 3 outcomes
- a uniform distribution:



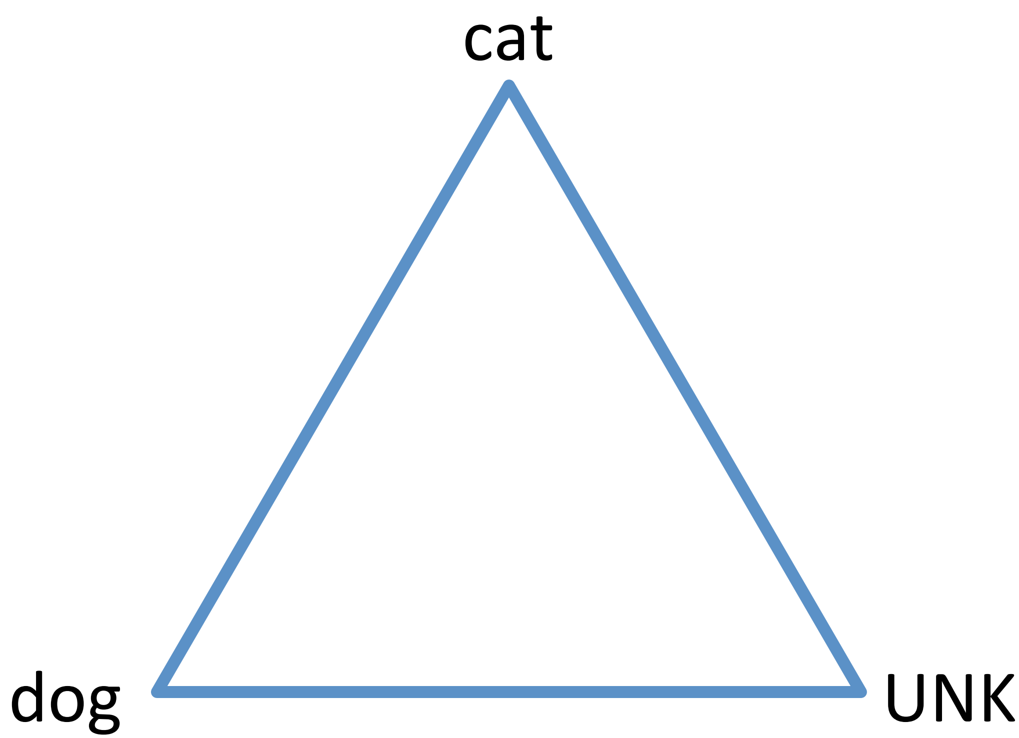
The 2-Simplex

- a distribution that puts most probability on UNK:

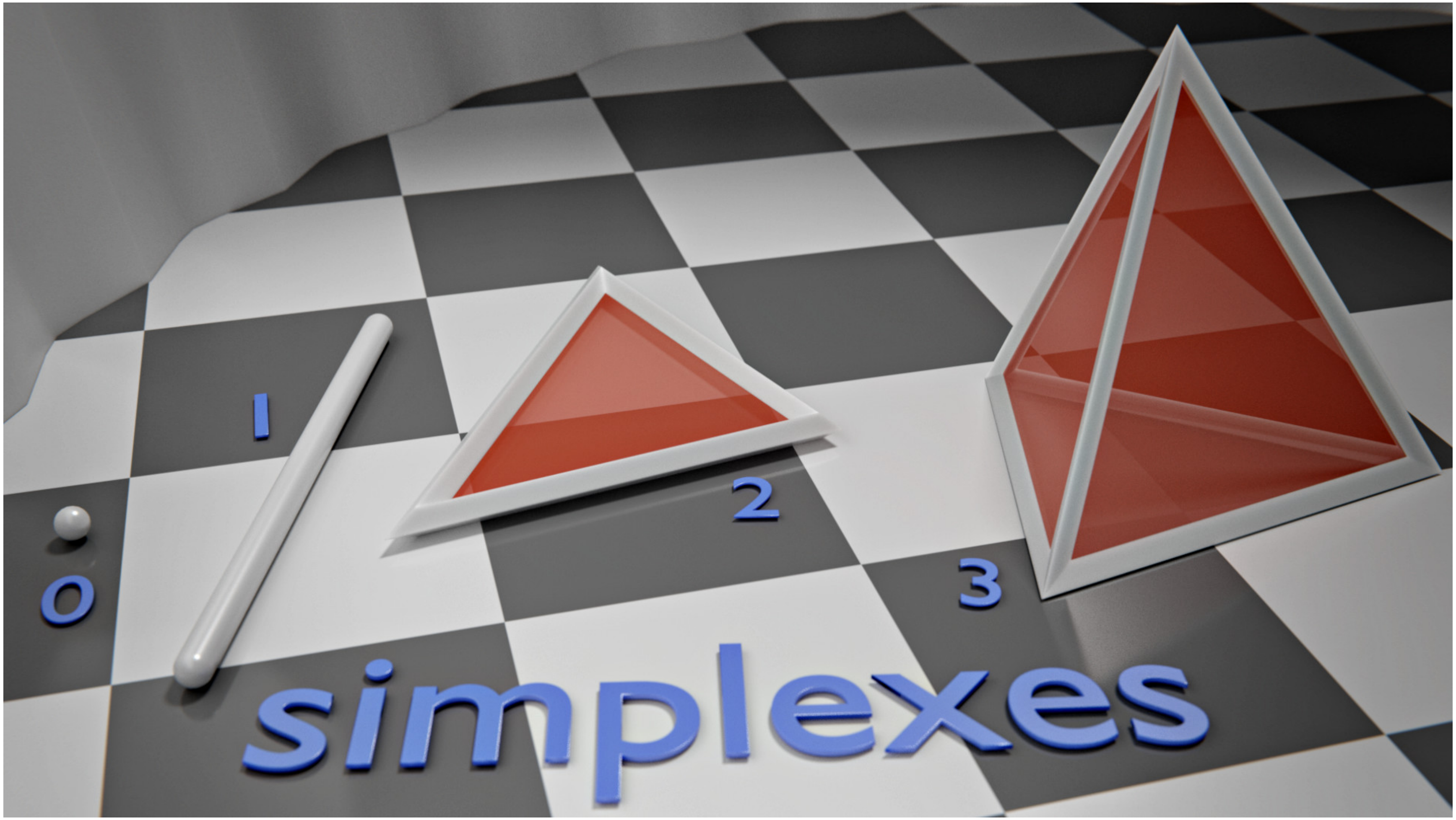


The 2-Simplex

- a categorical distribution with *K* outcomes has *K*-1 parameters



Categorical Parameters lie in (K-1)-Simplex



source: https://en.wikipedia.org/wiki/Simplex#/media/File:Simplexes.jpg

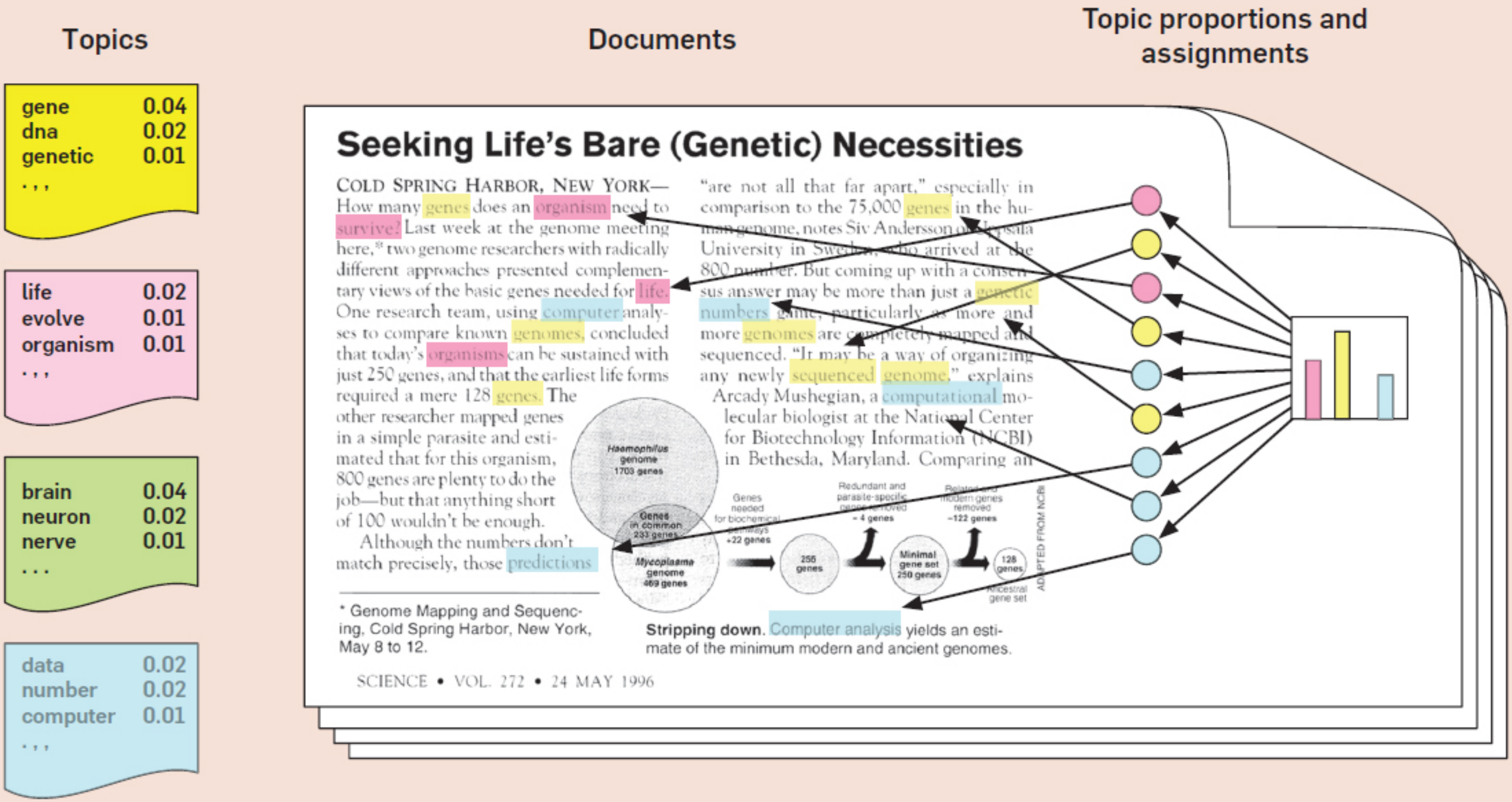
Latent Dirichlet Allocation

David M. Blei BLEI@CS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA 94720, USA

Andrew Y. Ng ANG@CS.STANFORD.EDU
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720, USA

- generative model for document collections using latent variables that can be interpreted as "topics"
- learns a multinomial distribution over words for each topic



Blei (2012): *Probabilistic Topic Models*

Latent Dirichlet Allocation

(Blei et al., 2003)

categorical distributions over words for four 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

Generative Story for Simple LDA

simplified LDA, and only showing generative story for 1 document:

1: Draw a multinomial topic distribution θ from some distribution $p(\Theta)$

2: For each position i in document:

a: Draw a topic $z_i \sim \text{Multinomial}(\theta)$

b: Draw a word $w_i \sim \text{Multinomial}(\beta_{z_i})$

Generative Story for Simple LDA

simplified LDA, and only showing generative story for 1 document:

1: Draw a multinomial topic distribution θ from some distribution $p(\Theta)$

2: For each position i in document:

a: Draw a topic $z_i \sim \text{Multinomial}(\theta)$

b: Draw a word $w_i \sim \text{Multinomial}(\beta_{z_i})$

multinomial distribution over words for topic z_i

Generative Story for Simple LDA

simplified LDA, and only showing generative story for 1 document:

1: Draw a multinomial topic distribution θ from some distribution $p(\Theta)$

2: For each position i in document:

a: Draw a topic $z_i \sim \text{Multinomial}(\theta)$

b: Draw a word $w_i \sim \text{Multinomial}(\beta_{z_i})$

what should we keep in mind when choosing this distribution?

Dirichlet Distribution

- distribution over vectors with entries that are all positive and sum to 1
- so it's kind of like a "distribution over (categorical) distributions"

$$p(\Theta = \theta \mid \alpha) = \frac{1}{B(\alpha)} \prod_i \theta_i^{\alpha_i - 1}$$

normalization term that depends on α

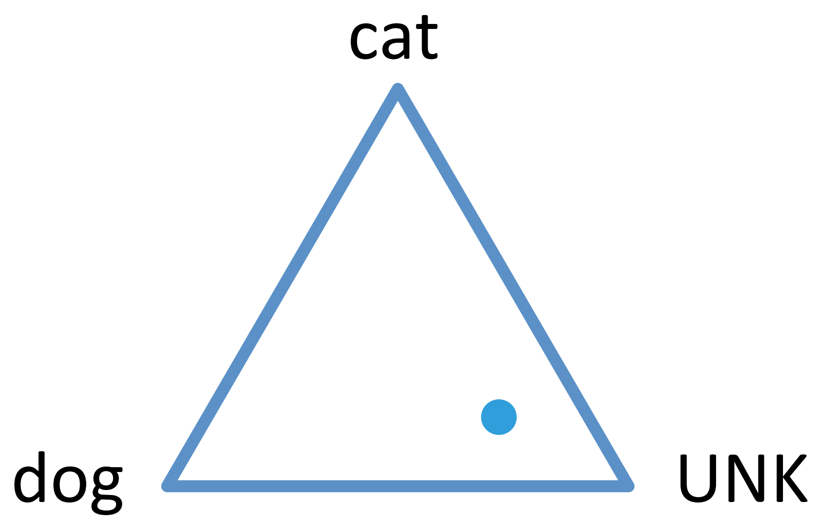
Dirichlet Distribution

- parameterized by a positive vector α

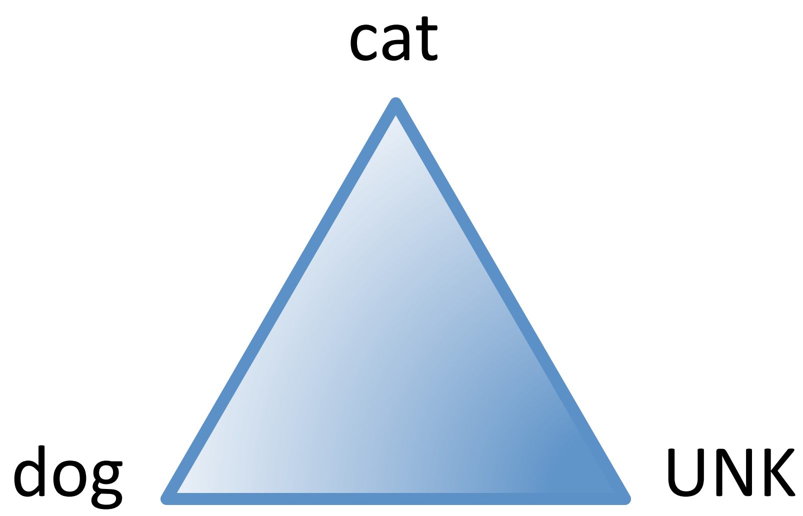
$$p(\Theta = \theta \mid \alpha) = \frac{1}{B(\alpha)} \prod_i \theta_i^{\alpha_i - 1}$$

$$\theta \sim \text{Dirichlet}(\alpha)$$

- categorical = point on the simplex



- Dirichlet = distribution over the simplex



[see Jupyter Notebook]

Generative Story for Simple LDA

simplified LDA, and only showing generative story for 1 document:

1: Draw a multinomial topic distribution θ from some distribution $p(\Theta)$

2: For each position i in document:

a: Draw a topic $z_i \sim \text{Multinomial}(\theta)$

b: Draw a word $w_i \sim \text{Multinomial}(\beta_{z_i})$

Generative Story for Simple LDA

simplified LDA, and only showing generative story for 1 document:

1: Draw a multinomial topic distribution $\theta \sim \text{Dirichlet}(\alpha)$

2: For each position i in document:

a: Draw a topic $z_i \sim \text{Multinomial}(\theta)$

b: Draw a word $w_i \sim \text{Multinomial}(\beta_{z_i})$

Generative Story for LDA

1: For each topic, draw a multinomial word distribution $\beta_i \sim \text{Dirichlet}(\eta)$

2: For each document d:

a: Draw a multinomial topic distribution $\theta \sim \text{Dirichlet}(\alpha)$

b: For each position i in document d:

i: Draw a topic $z_i \sim \text{Multinomial}(\theta)$

ii: Draw a word $w_i \sim \text{Multinomial}(\beta_{z_i})$

- now we show explicitly the generation of the word multinomials (once for the document collection)
- where should the hyperparameters (alpha and psi) come from?

Graphical Model for LDA