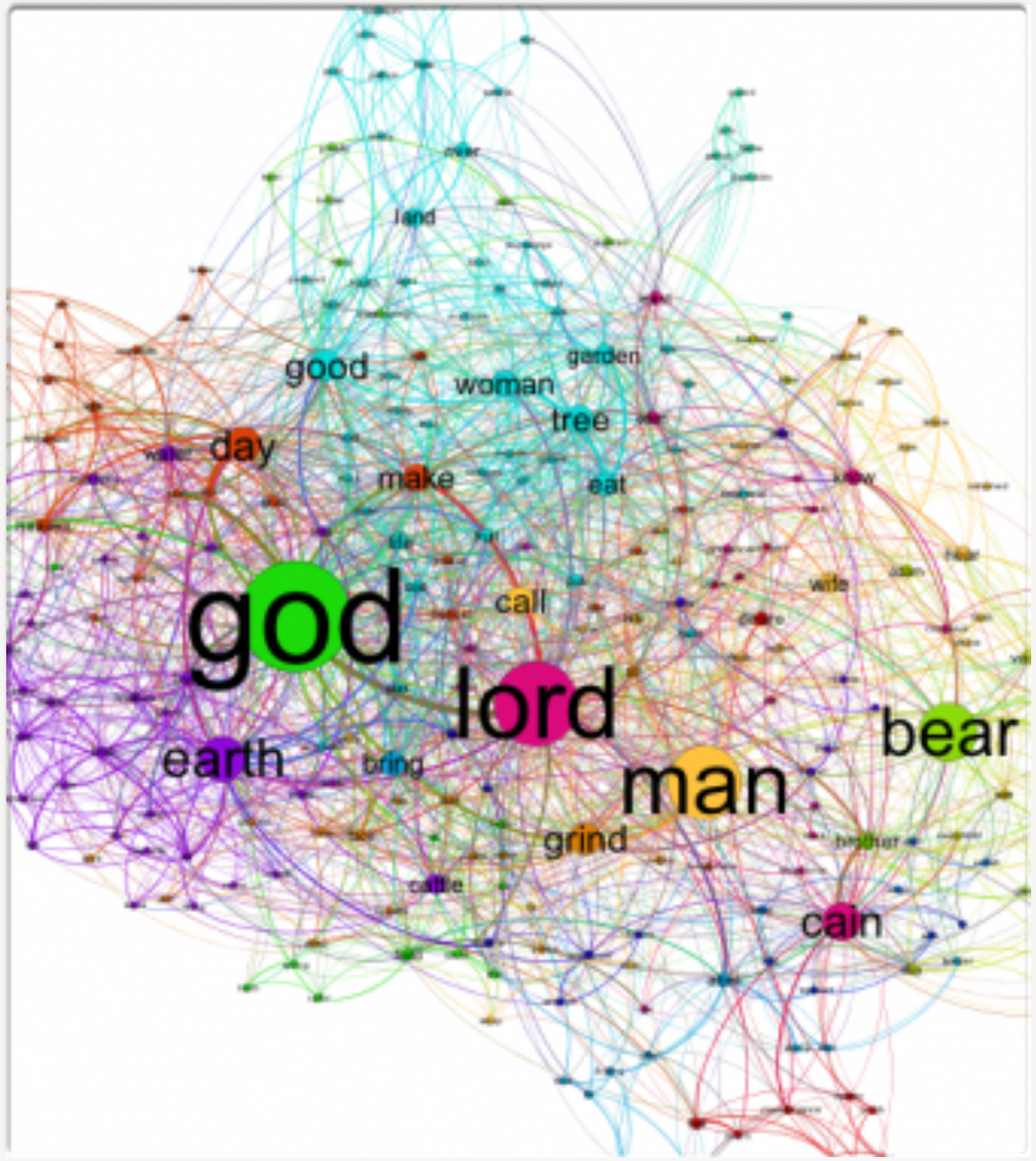
A TUTORIAL ON TEXT MODELING

~~HOW TO WRITE A GOOD ESSAY?~~

BY ED

AGENDA

- Background
- Text Modeling
- Application & Summary



DISTRIBUTIONS - REVIEW

- Bernoulli - toss a coin
- Binomial - toss a coin for multiple times
- Multinomial - toss a dice for multiple times
- ...

DISTRIBUTIONS - BETA

1. Draw 10 random numbers ~ Uniform(0,1) *iid*

2. Sort them in order

3. What's the distribution of the K-th biggest number?

$$P(x \leq X_{(k)} \leq x + \Delta x) = ?$$

DISTRIBUTIONS - BETA

$$E = \{X_1 \in [x, x+\Delta x],$$
$$X_i \in [0, x) \quad (i = 2, \cdots, k),$$
$$X_j \in (x+\Delta x, 1] \quad (j = k+1, \cdots, n)\}$$

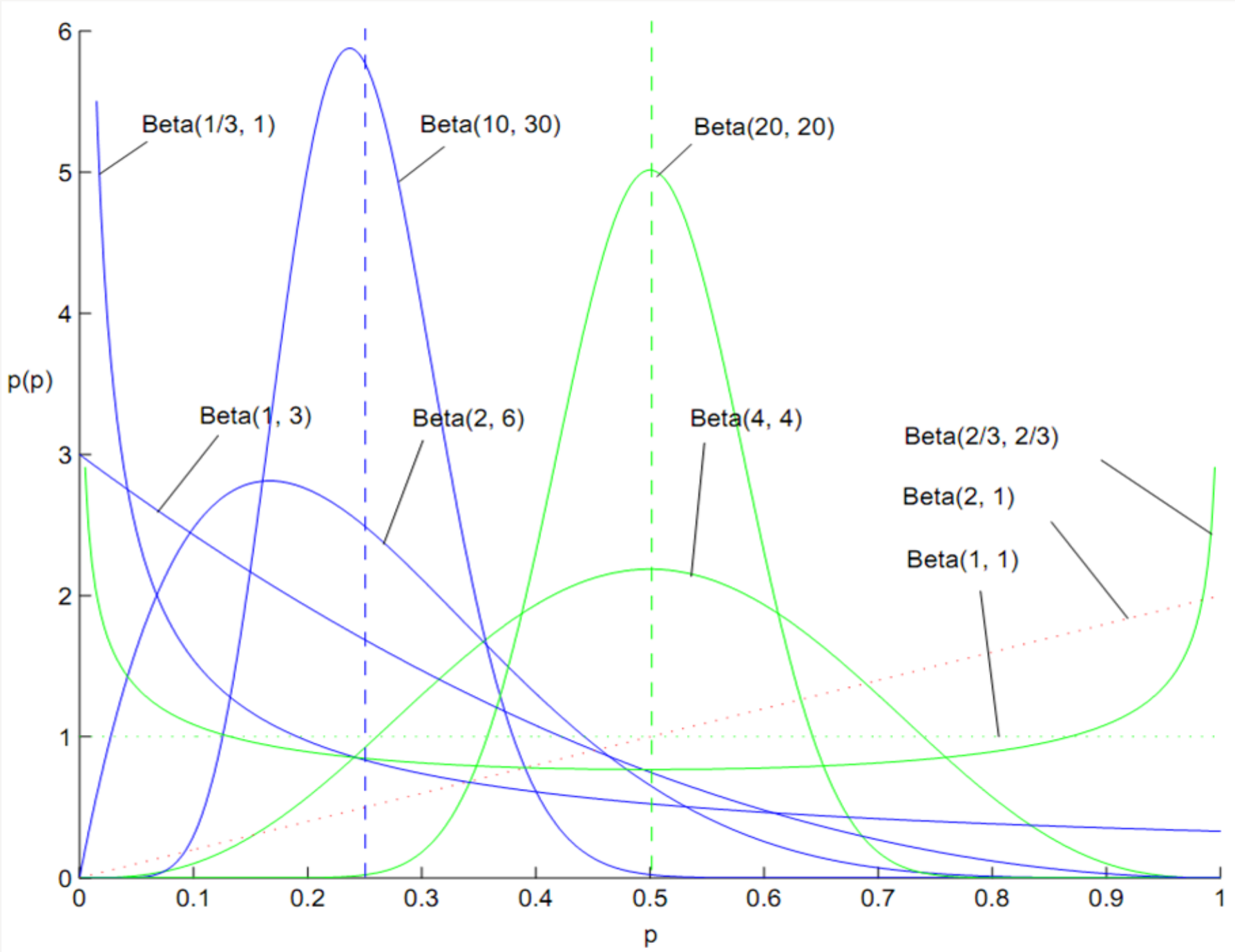
$$P(x \le X_{(k)} \le x+\Delta x)$$
$$= n\binom{n-1}{k-1} P(E) + o(\Delta x)$$
$$= n\binom{n-1}{k-1} x^{k-1}(1-x)^{n-k}\Delta x + o(\Delta x)$$

$$f(x) = \lim_{\Delta x \to 0} \frac{P(x \le X_{(k)} \le x+\Delta x)}{\Delta x}$$
$$= n\binom{n-1}{k-1} x^{k-1}(1-x)^{n-k}$$
$$= \frac{n!}{(k-1)!(n-k)!} x^{k-1}(1-x)^{n-k} \quad x \in [0,1]$$

$$f(x) = \frac{\Gamma(n+1)}{\Gamma(k)\Gamma(n-k+1)} x^{k-1}(1-x)^{n-k}$$

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

DISTRIBUTIONS - BETA



DISTRIBUTIONS - BETA-BINOMIAL CONJUGATE

1. Draw 10 random numbers ~ Uniform(0,1) *iid*

2. Sort them in order

3. Draw another 5 random numbers ~ Uniform (0,1) *iid*, tell you *m* of them are bigger than K-th biggest, *5-m* of them are smaller

4. What's the distribution of the K-th biggest number?

$$P(x \leq X_k \leq x + \Delta x | m_1, m_2) = ?$$

DISTRIBUTIONS - BETA-BINOMIAL CONJUGATE

$$Beta(p|k, n-k+1) + BinomCount(m_1, m_2) = Beta(p|k+m_1, n-k+1+m_2)$$

DISTRIBUTIONS - DIRICHLET-MULTINOMIAL CONJUGATE

- **Dirichlet distribution**

$$f(x_1, x_2, x_3) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1 - 1} x_2^{\alpha_2 - 1} x_3^{\alpha_3 - 1}$$

$$Dir(\vec{p}|\vec{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1}$$

- **Dirichlet-Multinomial conjugate**

$$Dir(\vec{p}|\vec{\alpha}) + MultCount(\vec{m}) = Dir(\vec{p}|\vec{\alpha} + \vec{m})$$

MARKOV CHAIN - DEFINITION

- **Markov property**: given the present state, the future and past states are independent.

$$P(X_{t+1} = x | X_t, X_{t-1}, \cdots) = P(X_{t+1} = x | X_t)$$

- **Steady state**: if aperiodic and every states are connected, then

$$\lim_{n \to \infty} P^n = \begin{bmatrix} \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

MARKOV CHAIN - SAMPLING

- Objective: **Sampling** for a distribution

$$X_0 \sim \pi_0(x)$$

$$X_1 \sim \pi_1(x)$$

$$\dots$$

$$X_n \sim \pi_n(x) = \pi(x)$$

$$X_{n+1} \sim \pi(x)$$

$$X_{n+2} \sim \pi(x)$$

$$\dots$$

MARKOV CHAIN - MCMC

- Detailed balance condition

$$\pi(i)P_{ij} = \pi(j)P_{ji} \qquad \text{for all} \quad i, j$$

- Build a new Transition Matrix

$$p(i)q(i,j) \neq p(j)q(j,i)$$

$$\alpha(i,j) = p(j)q(j,i) \quad \alpha(j,i) = p(i)q(i,j)$$

$$p(i)q(i,j)\alpha(i,j) = p(j)q(j,i)\alpha(j,i)$$

MARKOV CHAIN - MCMC

1. Initialize the algorithm with an arbitrary value x_0 and M.

2. Set $j = 1$.

3. Generate x_j^* from $q\left(x_{j-1}, x_j^*\right)$ and u from $\mathcal{U}[0, 1]$.

4. If $u \leq \alpha\left(x_{j-1}, x_j^*\right)$ then $x_j = x_j^*$, if $u > \alpha\left(x_{j-1}, x_j^*\right)$ then $x_j = x_{j-1}$.

5. If $j \leq M$ then $j \rightsquigarrow j + 1$ and got to 3.

MCMC - GIBBS SAMPLING

- 2-dimension distribution

$$p(x_1, y_1)p(y_2|x_1) = p(x_1)p(y_1|x_1)p(y_2|x_1)$$
$$p(x_1, y_2)p(y_1|x_1) = p(x_1)p(y_2|x_1)p(y_1|x_1)$$

- Transition matrix

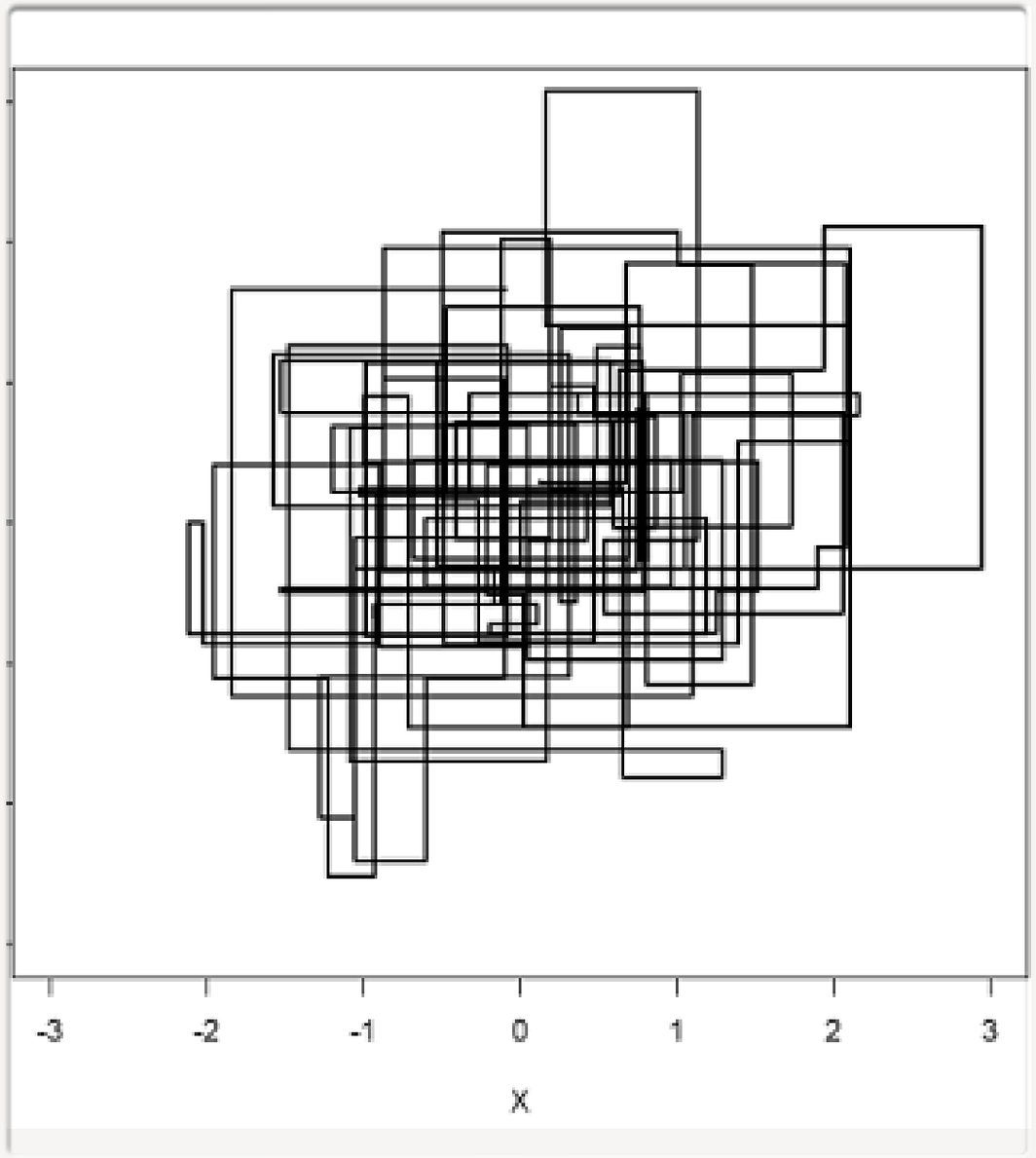
$$q(A \to B) = \begin{cases} p(y_B|x_0) & x_A = x_B = x_0 \\ p(x_B|y_0) & y_A = y_B = y_c \\ 0 & otherwise \end{cases}$$

- Detailed balance condition satisfied!

MCMC - GIBBS SAMPLING

- Initialize X, Y
- Sample

1. $y_{t+1} \sim p(y|x_t)$
2. $x_{t+1} \sim p(x|y_{t+1})$



TEXT MODELING - UNIGRAM

- Probability of the vocabulary (Big Dice)

$$\hat{p}_i = \frac{n_i}{N}.$$

- Big Dice has a prior

$$p(\vec{n}) = Mult(\vec{n}|\vec{p}, N)$$

$$Dir(\vec{p}|\vec{\alpha}) = \frac{1}{\Delta(\vec{\alpha})} \prod_{k=1}^{V} p_k^{\alpha_k - 1} \quad \vec{\alpha} = (\alpha_1, \cdots, \alpha_V)$$

$$p(\vec{p}|\mathcal{W}, \vec{\alpha}) = Dir(\vec{p}|\vec{n} + \vec{\alpha}) = \frac{1}{\Delta(\vec{n} + \vec{\alpha})} \prod_{k=1}^{V} p_k^{n_k + \alpha_k - 1} d\vec{p}$$

$$E(\vec{p}) = \Big(\frac{n_1 + \alpha_1}{\sum_{i=1}^{V}(n_i + \alpha_i)}, \frac{n_2 + \alpha_2}{\sum_{i=1}^{V}(n_i + \alpha_i)}, \cdots, \frac{n_V + \alpha_V}{\sum_{i=1}^{V}(n_i + \alpha_i)} \Big)$$

TEXT MODELING - LSA

$$
\begin{array}{ccccccc}
 & & X & & U & \Sigma & V^T \\
 & & (\mathbf{d}_j) & & & & (\hat{\mathbf{d}}_j) \\
 & & \downarrow & & & & \downarrow \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} & \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot & \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

- Not defined properly normalized probabilities
- No obvious interpretations of LS space directions

TEXT MODELING - PLSA

- Doc-topic distribution

$$\vec{\theta}_1, \cdots, \vec{\theta}_M$$

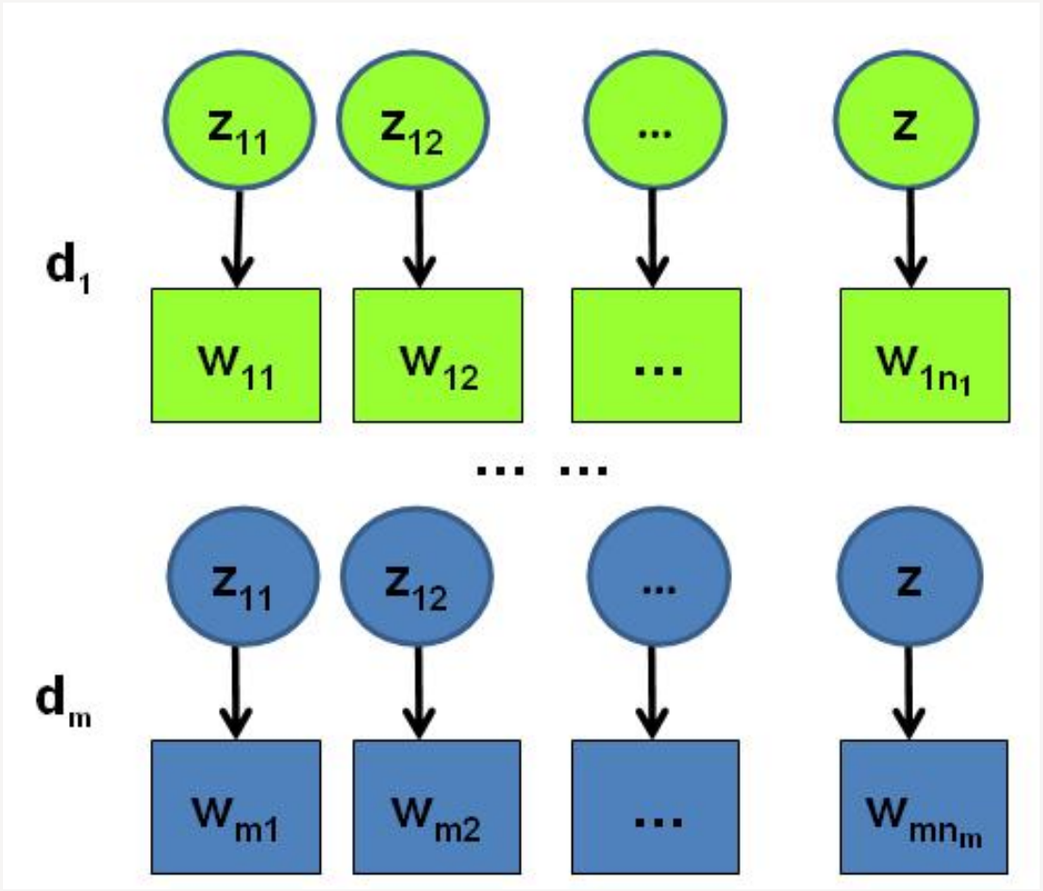
- Topic-word distribution

$$\vec{\varphi}_1, \cdots, \vec{\varphi}_K$$

- Probability of every word in a document

$$p(w|d_m) = \sum_{z=1}^{K} p(w|z)p(z|d_m) = \sum_{z=1}^{K} \varphi_{zw}\theta_{mz}$$

TEXT MODELING - LDA



$$\vec{\alpha} \rightarrow \vec{\theta}_m \rightarrow \vec{z}_m$$

$$\vec{\beta} \rightarrow \vec{\varphi}_k \rightarrow \vec{w}_{(k)}$$

TEXT MODELING - LDA

- Sampling distribution

$$\hat{\theta}_{mk} \cdot \hat{\varphi}_{kt}$$

$$p(z_i = k|\vec{\mathbf{z}}_{\neg i}, \vec{\mathbf{w}}) \propto \frac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{m,\neg i}^{(t)} + \alpha_k)} \cdot \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}(n_{k,\neg i}^{(t)} + \beta_t)}$$

- LDA training
  - Give a random topic for every word
  - Use Gibbs sampling to update every word's topic
  - repeat until convergence (or iteration ends)

TEXT MODELING - LDA

```
while interating:
    for m in LDACorpus:
        for w in m:
            # generate topic distribution for w
            distribution = {}
            for k in Topics:
                # calculate P(z_k | W_^w, z_^w)
                diff = -1 if topic[m][w] == k else 0
                d_factor = doc_topic_distribution[m][k] + diff
                t_factor = topic_word_distribution[k][w] + diff
                distribution[k] = (d_factor + alpha) * (t_factor + beta) / \
                        (sum(topic_word_distribution[k]) + beta * Vocab_size)
            # sample from distribution
            k = sample(distribution)
            topic[m][w] = k
            # update model
            topic_word_distribution[k][w] += 1
            doc_topic_distribution[m][k] += 1
    # check convergence and post process
```

APPLICATION

- Similarity measure
  - Get topic distribution of a new document
  - Jensen-Shannon divergence
- *Caveats*!
  - Computation
  - Improvement?

THANKS!

To Infinity and Beyond !