Review

* Matrix Factorization

How to approximate $X \approx VY$?

VQ: minimize $\|X - VY\|$ subject to $\begin{cases} Y_{\alpha n} \in \{0,1\} \\ \sum_\alpha Y_{\alpha n} = 1 \end{cases}$

PCA: minimize $\|X - VY\|$ subject to $V^T V = I$

NMF: minimize $\|X - VY\|$ subject to $V_{i\alpha}, Y_{\alpha n} \geq 0$

* Exponential family PCA

Assume X_{ij} has exponential family distribution with natural parameter Θ_{ij}

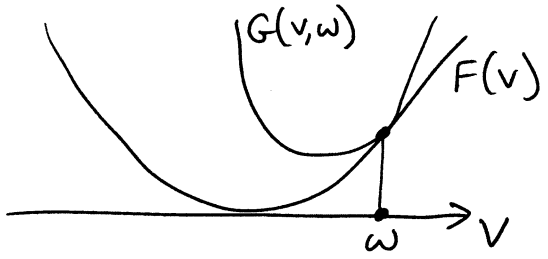
How to model X with $\Theta = VY$?

Maximize log-likelihood:

$$\mathcal{L} = \sum_{ij} \log P(X_{ij})$$
$$= \sum_{ij} \left[\log P_0(X_{ij}) + X_{ij}(VY)_{ij} - G((VY)_{ij}) \right]$$

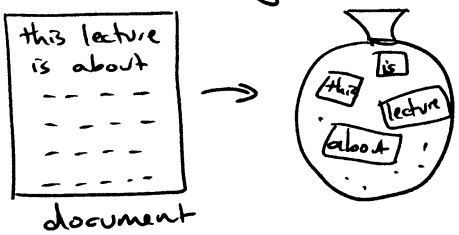
* Auxiliary functions

- Properties: $G(v,v) = F(v)$ and
$G(v,w) \geq F(v)$ for all v, w
- Update: $V_{t+1} \leftarrow \operatorname{argmin}_V G(v, V_t)$

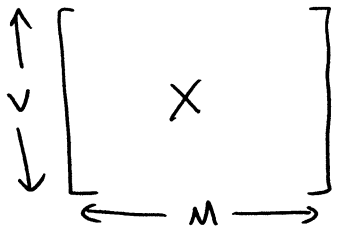


Document Modeling

* Bag-of-words model
  - Represent document by vector of (raw or normalized) word counts.
  - Ignore word ordering: words are "exchangeable"



  - Corpus as term-by-document matrix



V = # words/tokens in vocabulary

M = # documents in corpus

X_{ij} = raw or normalized count of word i in doc. j

$N_j = \sum_i X_{ij}$ = # words in j^{th} document

  - How to model X?

    * Matrix factorization: $X = VY$
      - PCA known as "latent semantic indexing" in this context
      - NMF can be applied to raw word counts
      - But neither PCA/NMF provide probabilistic models of how documents are generated

Modeling Assumptions

* Notation / terminology:

  Document = sequence of N words $\vec{w} = \{w_1, w_2, \cdots, w_N\}$

  Corpus = collection of M documents $D = \{\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_M\}$

* Bag-of-words / exchangeability assumption!

  - summarize corpus by word-document counts X_{ij}
  - data enters likelihood only via matrix X_{ij}

* What interesting models are there for $P(\vec{w})$ and $P(D)$?

  (not n-grams, which model word ordering)

Early Work

* Unigram model
  - draw words iid from single multinomial distribution
  - document probability

$$P(\vec{w}) = P(w_1, w_2, \ldots, w_N) = \prod_{n=1}^{N} P(w_n) = \prod_{i=1}^{V} P(w=i)^{\text{count}(w=i)}$$

  - corpus probability

$$P(D) = P(\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_M) = \prod_{j=1}^{M} P(\vec{w}_j) = \prod_{j=1}^{M} \prod_{i=1}^{V} P(w=i)^{x_{ij}}$$

(x_{ij} ← word-document count)

  - Limitation.
    No notion of "topic" to correlate words

* Mixture of Unigrams
  - For each document:
    - Draw topic z from single multinomial distribution $P(z)$
    - Draw words iid from conditional multinomial distribution $P(w|z)$
  - Document Probability (K = # topics (fixed))

$$P(w_1, w_2, \ldots, w_N) = \sum_{z=1}^{K} P(z) \prod_{n=1}^{N} P(w_n|z)$$

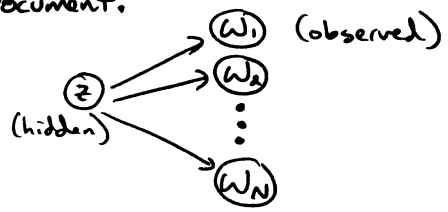
  - Corpus Probability

$$P(D) = \prod_{j=1}^{M} P(\vec{w}_j) = \prod_{j=1}^{M} \left\{ \sum_{z} P(z) \prod_{i=1}^{V} P(w=i|z)^{x_{ij}} \right\}$$

  - Strength: words correlated by topics
  - Weakness: one topic per document

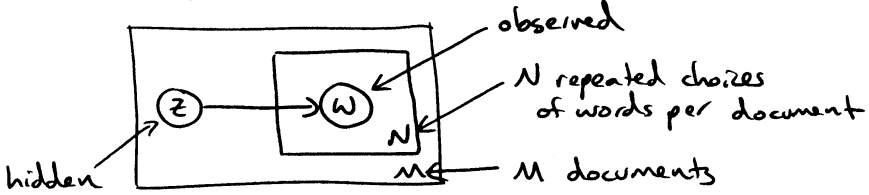
- Graphical model

Document:



Corpus: above is replicated M times

Plate notation:



* Probabilistic latent semantic indexing (PLSI)

- also known as an "aspect" model
- for each document d, and for each word in d:
  - draw a topic z from multinomial distribution $P(z|d)$
  - draw a word from $P(w|z)$
- document probability

$$P(\vec{w}|d) = \prod_{n=1}^{N}\left[\sum_{z} P(z|d)\,P(w_n|z)\right]$$

- corpus probability

$$P(D) = \prod_{j=1}^{M} P(\vec{w}_j|d)$$

$$= \prod_{j=1}^{M}\prod_{i=1}^{V}\left[\sum_{z} P(w=i|z)P(z|d_j)\right]^{x_{ij}}$$

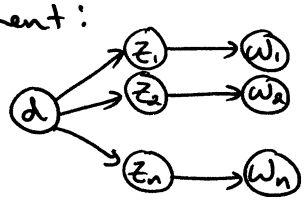
log-likelihood of corpus:

$$\mathcal{L} = \log P(D)$$

$$= \sum_{ij} x_{ij} \log\left[\sum_{z} P(w=i|z)P(z|d_j)\right]$$

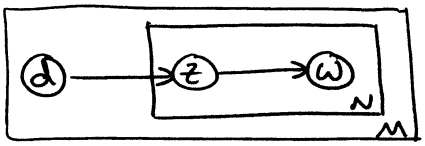
- Graphical model

document:



Corpus: replicate above M times

Plate notation:



M documents
N repeated choices of topic and word

— Strengths
- multiple topics per document
- EM algorithm can estimate $P(w|z)$ and $P(z|d_j)$ can discover "hidden" topics
- Inference is tractable

— Weaknesses
- Only assigns probabilities to documents in corpus. (doesn't generalize to new documents)
- # parameters grows linearly in M = # docs
  ⇒ overfitting

Latent Dirichlet Allocation (LDA)

How to model distributions over topics?

* Dirichlet distribution
  - a "distribution over distributions"
  - let θ take values in simplex

    $\theta = (\theta_1, \theta_2, \ldots, \theta_K)$ with $\theta_K \geq 0$ and $\sum_{k=1}^{K} \theta_k = 1$
  - Exponential family distribution

    $$P(\theta|\alpha) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \theta_1^{\alpha_1 - 1} \theta_2^{\alpha_2 - 1} \cdots \theta_K^{\alpha_K - 1}$$

    parameters of Dirichlet distribution $\alpha_K > 0$

    normalization constant
  - Parameters

    $$E[\theta_k] = \frac{\alpha_k}{\alpha_0} \text{ where } \alpha_0 = \sum_{k=1}^{K} \alpha_k$$
  - Gamma function

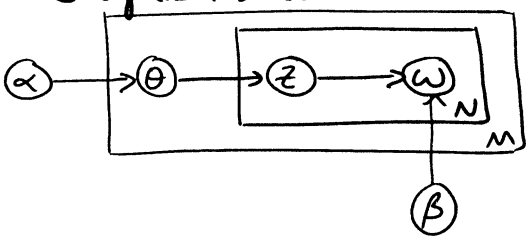
    Definition: $\Gamma(x) = \int_0^\infty dt\ e^{-t} t^{x-1}$

    Properties: $\Gamma(x+1) = x\Gamma(x)$ prove using integration by parts

    $\Gamma(x+1) = x!$ for integers $x > 0$

* Generative Model for LDA
  - for each document $\vec{w}$ in some corpus D:
    1) Choose N = # words in document by some distribution (not modeled)
    2) Choose θ from $P(\theta|\alpha)$
    3) Do N times:
       (a) choose $z_n \in \{1, 2, \ldots, K\}$ from multinomial with weights θ
       (b) choose w_n from multinomial $P(w_n|z_n, \beta)$ specifically: $P(\vec{w}=j \mid z=K) = \beta_{kj}$ matrix of parameters

- Graphical Model



- Modeling Parameters

α_k for $k = 1 \ldots K$ corpus/document-topic model

β_{kj} for $j = 1 \ldots V$ topic-word model

parameters = $K + KV$ no explicit linear dependence on M