Latent variable models for discrete data

Jianfei Chen

Department of Computer Science and Technology
Tsinghua University, Beijing 100084
chris.jianfei.chen@gmail.com

January 13, 2014

We want to model three types of discrete data

- Sequence of tokens: $p(y_{i,1:L_i})$
- Bag of words: $p(n_i)$
- Discrete features: $p(y_{i,1:R})$
Outline

- **Mixture Models**
- **LSA / PLSI / LDA / GaP / NMF**
- **LDA**
  - Evaluation
  - Inference
  - Variants: CTM, DTM, LDA-HMM, SLDA, MedLDA, etc.
- **RBM**
Mixture models

\[
p(y) = \sum_k p(y|q_i = k) p(q_i = k)
\]

- Sequence of tokens: \( p(y_{i,1:L_i}|q_i = k) = \prod_{l=1}^{L_i} \text{Cat}(y_{il}|b_k) \)
- Discrete features: \( p(y_{i,1:R}|q_i = k) = \prod_{r=1}^{R} \text{Cat}(y_{ir}|b_k^{(r)}) \)
- Bag of words (known \( L_i \)): \( p(n_i|L_i, q_i = k) = \text{Mu}(n_i|L_i, b_k) \)
- Bag of words (unknown \( L_i \)): \( p(n_i|q_i = k) = \prod_{v=1}^{V} \text{Poi}(n_{iv}|\lambda_{vk}) \)
Theorem

If \( \forall i, X_i \sim \text{Poi}(\lambda_i) \), let \( n = \sum_i X_i \)

\[
p(X_1, \cdots, X_k | n) = \text{Mu}(\mathbf{X} | n, \pi)
\]

where \( \pi_i = \frac{\lambda_i}{\sum_k \lambda_k} \).
latent semantic analysis (LSA) / latent semantic indexing (LSI)

- Sequence of tokens: \( p(y_{i,1:L_i} | z_i) = \prod_{l=1}^{L_i} \text{Cat}(y_{il} | S(Wz_i)) \)
- Discrete features: \( p(y_{i,1:R} | z_i) = \prod_{r=1}^{R} \text{Cat}(y_{ir} | S(W_r z_i)) \)
- Bag of words (known \( L_i \)): \( p(n_i | L_i, z_i) = \text{Mu}(n_i | L_i, S(W z_i)) \)
- Bag of words (unknown \( L_i \)): \( p(n_i | z_i) = \prod_{v=1}^{V} \text{Poi}(n_{iv} | \exp(w_v : z_i)) \)

where \( S(\cdot) \) is the softmax transformation, \( z_i \in \mathbb{R}^K \), \( W, W_r \in \mathbb{R}^{V \times K} \).

Inference

- coordinate ascent / degenerated EM (problem: overfitting?)
- variational EM / MCMC
Unigram: \[ p(y_{i,1:L_i} | q_i = k) = \prod_{l=1}^{L_i} \text{Cat}(y_{il} | b_k) \]

LSI: \[ p(y_{i,1:L_i} | z_i) = \prod_{l=1}^{L_i} \text{Cat}(y_{il} | \mathbf{S}(\mathbf{W}z_i)) \]

PLSI: \[ p(y_{i,1:L_i} | \pi_i) = \prod_{l=1}^{L_i} \text{Cat}(y_{il} | \mathbf{B}\pi_i) \]

LDA: \[ p(y_{i,1:L_i} | \pi_i) = \prod_{l=1}^{L_i} \text{Cat}(y_{il} | \mathbf{B}\pi_i), \pi_i \sim \text{Dir}(\pi_i | \alpha) \]

LDA for other data types

- Bag of words: \[ p(n_i | L_i, \pi_i) = \text{Mu}(n_i | L_i, \mathbf{B}\pi_i) \]
- Discrete features: \[ p(y_{i,1:R} | \pi_i) = \prod_{r=1}^{R} \text{Cat}(y_{ir} | \mathbf{B}^{(r)}\pi_i) \]

Question: What is dual parameter? Why is it convenient?

Gamma-Poisson Model

LDA
- models \( p(n_i|L_i, \pi_i) = \text{Mu}(n_i|L_i, B\pi_i) \)
- Prior \( \pi_i \sim \text{Dir}(\alpha) \)
- Constraint \( 0 \leq \pi_{ik}, \sum_j \pi_{ik} = 1, 0 \leq B_{vk}, \sum_v B_{vk} = 1 \)

GaP
- models \( p(n_i|z_i^+) = \prod_{v=1}^V \text{Poi}(n_{iv}|b_v^\top z_i^+) \)
- Prior \( p(z_i^+) = \prod_k \text{Ga}(z_{ik}^+|\alpha_k, \beta_k) \)
- Constraint \( 0 \leq z_{ik}, 0 \leq B_{vk} \)

Can use sparse-inducing prior (27.17)
GaP only have non-negative constraints
Non-negative matrix factorization

Given non-negative matrix $V$, find non-negative matrix factors $W, H$ such that

$$V \approx WH$$

$$V_i \approx \sum_k W_{ik}H_k$$

Can be viewed as GaP when prior $\alpha_k = \beta_k = 0$.

Latent Dirichlet Allocation (LDA)

Notation

\[ \pi_z | \alpha \sim \text{Dir}(\alpha) \] (1)

\[ q_{il} | \pi_i \sim \text{Cat}(\pi_i) \] (2)

\[ b_k | \gamma \sim \text{Dir}(\gamma) \] (3)

\[ y_{il} | q_{il} = k, B \sim \text{Cat}(b_k) \] (4)

- Geometric interpretation
- Simplex: handle ambiguity (?)
- Unidentifiable: Labeled LDA

D. Blei et al. "Latent dirichlet allocation." JMLR
G. Heinrich. "Parameter estimation for text analysis."

Perplexity of language model $q$ given language $p$ is defined as (both $p$, $q$ are stochastic process)

$$\text{perplexity}(p, q) = 2^{H(p, q)}$$

where $H(p, q)$ is cross-entropy

$$H(p, q) = \lim_{N \to \infty} -\frac{1}{N} \sum_{y_1:N} p(y_{1:N}) \log q(y_{1:N})$$

Approximations
- $N$ is finite
- $p(y_{1:N}) = \delta_{y^*:N}(y_{1:N})$
Evaluation: Perplexity

\[ H(p, q) = -\frac{1}{N} \log q(y^*_{1:N}) \]

Intuition: weighted average branching factor

For unigram model

\[ H = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{L_i} \sum_{l=1}^{L_i} \log q(y^*_{il}) \]

For LDA

\[ H = -\frac{1}{N} \sum_{i=1}^{N} p(y^*_{i,1:L_i}) \]

- Use variational evidence lower bound (ELBO)
- Use annealed importance sampling
- Use validation set and plug in approximation

TODO

Exponential number of *inference algorithms*
- Variational inference vs sampling vs both
- Collapsed vs non-collapsed
- Online vs stochastic vs offline
- Empirical Bayes vs fully Bayes
- Other algorithms: expectation propagation, etc.
Inference: towards large scale

- algorithms
  - Online / stochastic
  - Sparsity
  - Spectral methods
- system
  - Distributed: Yahoo-LDA, Petuum, Parameter-Server, etc.
  - GPU: BIDMach, etc.
Model Selection

- Compute evidence with AIS / ELBO
- Cross validation
- Bayesian non-parametrics

Extensions of LDA

- Correlation: Correlated topic model
- Time series: Dynamic topic model
- Syntax: LDA-HMM
- Supervision: many
  - 1D categorial label: SLDA (generative), DLDA (discriminate), MedLDA (regularized)
  - nD label: MR-LDA, random effects mixture of experts, conditional topic random field, Dirichlet multinomial regression LDA
  - $K$ labels per document: labeled LDA
  - labels per word: TagLDA
- Structural: RTM
Restricted Boltzmann machines

\[
p(h, v | \theta) = \frac{1}{Z(\theta)} \prod_{r=1}^{R} \prod_{k=1}^{K} \psi_{rk}(v_r, h_k)
\]

where \( h, v \) are binary vectors.

factorized posterior

\[
p(h | v, \theta) = \prod_{k} p(h_k | v, \theta)
\]

advantage: symmetric, both posterior inference (backward) and generating (forward) are easy.

- Exponential family harmonium (harmonium is 2-layer UGM)
Restricted Boltzmann machines

Binary latent and binary visible (other models exist, see Table 27.2)

\[
p(v, h|\theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta)) \tag{5}
\]

\[
E(v, h; \theta) = v^\top Wh \tag{6}
\]

\[
p(h|v, \theta) = \prod_k \text{Ber}(h_k|\text{sigm}(w_{:,k}^\top v)) \tag{7}
\]

\[
p(v|h, \theta) = \prod_r \text{Ber}(v_r|\text{sigm}(w_{r,:}^\top h)) \tag{8}
\]
Goal: maximize $p(v | \theta)$

$$\nabla_w l = E_{p_{emp}(\cdot | \theta)}[vh^\top] - E_p(\cdot | \theta)[vh^\top]$$
Conclusions

Why there are many things to do

- Exponential number of inference algorithms
- Exponential number of models
- Exponential $\times$ exponential number of solutions
- Application, evaluation, theory (e.g. spectral), etc.

Need a way for information retriever, data miners find correct & fast solutions for them...