Latent variable models for discrete data

Jianfei Chen

Department of Computer Science and Technology
Tsinghua University, Beijing 100084

chris.jianfei.chen@gmail.com

Janurary 13, 2014

Murphy, Kevin P. Machine learning: a probabilistic perspective. The MIT Press, 2012. Chapter 27.

Introduction

We want to model three types of discrete data

- Sequence of tokens: $p(\mathbf{y}_{i,1:L_i})$
- Bag of words: $p(\mathbf{n}_i)$
- Discrete features: $p(\mathbf{y}_{i,1:R})$

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: $p(\mathbf{y}_{i,1:L_i}|q_i = k) = \prod_{l=1}^{L_i} \mathrm{Cat}(y_{il}|\mathbf{b}_k)$
- Discrete features: $p(\mathbf{y}_{i,1:R}|q_i = k) = \prod_{r=1}^{R} \mathrm{Cat}(y_{ir}|\mathbf{b}_k^{(r)})$
- Bag of words (known L_i): $p(\mathbf{n}_i|L_i, q_i = k) = \mathrm{Mu}(\mathbf{n}_i|L_i, \mathbf{b}_k)$
- Bag of words (unknown L_i): $p(\mathbf{n}_i|q_i = k) = \prod_{v=1}^{V} \mathrm{Poi}(n_{iv}|\lambda v k)$

Mixture models

Theorem

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

$$p(X_1, \cdots, X_k | n) = \mathrm{Mu}(\mathbf{X} | n, \pi)$$

where $\pi_i = \frac{\lambda_i}{\sum_k \lambda_k}$.

Exponential Family PCA

latent semantic analysis (LSA) / latent semantic indexing (LSI)

- Sequence of tokens: $p(\mathbf{y}_{i,1:L_i}|\mathbf{z}_i) = \prod_{l=1}^{L_i} \mathrm{Cat}(y_{il}|S(\mathbf{W}\mathbf{z}_i))$
- Discrete features: $p(\mathbf{y}_{i,1:R}|\mathbf{z}_i) = \prod_{r=1}^{R} \mathrm{Cat}(y_{ir}|S(\mathbf{W}_r\mathbf{z}_i))$
- Bag of words (known L_i): $p(\mathbf{n}_i|L_i, \mathbf{z}_i) = \mathrm{Mu}(\mathbf{n}_i|L_i, S(\mathbf{W}\mathbf{z}_i))$
- Bag of words (unknown L_i): $p(\mathbf{n}_i|\mathbf{z}_i) = \prod_{v=1}^{V} \mathrm{Poi}(n_{iv}|exp(\mathbf{w}_{v,:}\mathbf{z}_i))$

where $S(\cdot)$ is the softmax transformation, $\mathbf{z}_i \in \mathbb{R}^K$, $\mathbf{W}, \mathbf{W}_r \in \mathbb{R}^{V\times K}$.
Inference

- coordinate ascent / degenerated EM (problem: overfitting?)
- variational EM / MCMC

LSA / PLSI / LDA

- Unigram: $p(\mathbf{y}_{i,1:L_i}|q_i = k) = \prod_{l=1}^{L_i} \mathrm{Cat}(y_{il}|\mathbf{b}_k)$
- LSI: $p(\mathbf{y}_{i,1:L_i}|\mathbf{z}_i) = \prod_{l=1}^{L_i} \mathrm{Cat}(y_{il}|S(\mathbf{W}\mathbf{z}_i))$
- PLSI: $p(\mathbf{y}_{i,1:L_i}|\pi_i) = \prod_{l=1}^{L_i} \mathrm{Cat}(y_{il}|\mathbf{B}\pi_i)$
- LDA: $p(\mathbf{y}_{i,1:L_i}|\pi_i) = \prod_{l=1}^{L_i} \mathrm{Cat}(y_{il}|\mathbf{B}\pi_i), \pi_i \sim Dir(\pi_i|\alpha)$

LDA for other data types

- Bag of words:$p(\mathbf{n}_i|L_i, \pi_i) = \mathrm{Mu}(\mathbf{n}_i|L_i, \mathbf{B}\pi_i)$
- Discrete features:$p(\mathbf{y}_{i,1:R}|\pi_i) = \prod_{r=1}^{R} \mathrm{Cat}(y_{ir}|\mathbf{B}^{(r)}\pi_i)$

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

Marlin, Benjamin M. "Modeling user rating profiles for collaborative filtering." Advances in neural information processing systems. 2003.

Gamma-Poisson Model

LDA

- models $p(\mathbf{n}_i|L_i, \pi_i) = \mathrm{Mu}(\mathbf{n}_i|L_i, \mathbf{B}\pi_i)$
- Prior $\pi_i \sim \mathrm{Dir}(\alpha)$
- Constraint $0 \leq \pi_{ik}, \sum_j \pi_{ik} = 1, 0 \leq B_{vk}, \sum_v B_{vk} = 1$

GaP

- models $p(\mathbf{n}_i|\mathbf{z}_i^+) = \prod_{v=1}^V \mathrm{Poi}(n_{iv}|\mathbf{b}_{v,:}^\top \mathbf{z}_i^+)$
- Prior $p(\mathbf{z}_i^+) = \prod_k \mathrm{Ga}(z_{ik}^+|\alpha_k, \beta_k)$
- 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 view as GaP when prior $\alpha_k = \beta_k = 0$.

Seung, D., and L. Lee. "Algorithms for non-negative matrix factorization." Advances in neural information processing systems.

Latent Dirichlet Allocation (LDA)

Notation

$$\pi_z|\alpha \sim \mathrm{Dir}(\alpha) \tag{1}$$
$$q_{il}|\pi_i \sim \mathrm{Cat}(\pi_i) \tag{2}$$
$$\mathbf{b}_k|\gamma \sim \mathrm{Dir}(\gamma) \tag{3}$$
$$y_{il}|q_{il} = k, \mathbf{B} \sim \mathrm{Cat}(\mathbf{b}_k) \tag{4}$$

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

D. Blei et al. "Latent dirichlet allocation." JMLR
G. Heinrich. "Parameter estimation for text analysis."
D. Ramage, et al. "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora." EMNLP

http://www.cs.princeton.edu/courses/archive/fall11/cos597C/lectures/variational-inference-i.pdf

Evaluation: Perplexity

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

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

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

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

Approximations

- N is finite
- $p(\mathbf{y}_{1:N}) = \delta_{\mathbf{y}^*_{1:N}}(\mathbf{y}_{1:N})$

Evaluation: Perplexity

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

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

H. Wallach, et al. "Evaluation methods for topic models." ICML 2009

Evaluation: Coherence

TODO

D. Newman et al. "Automatic evaluation of topic coherence." NAACL HLT 2010.

Inference

Exponential number of *inference algorithms*

- Variational inference vs sampling vs both
- Collapsed vs non-collpased
- Online vs stocastic vs offline
- Empirical Bayes vs fully Bayes
- Other algorithms: expectation propagation, etc.

Inference: towards large scale

- algorithms
  - Online / stocastic
  - 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

Teh et al. "Hierarchical dirichlet processes." Journal of the american statistical association (2006).

Extensions of LDA

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

Restricted Boltzmann machines

Restricted Boltzmann machines

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

where $\mathbf{h}, \mathbf{v}$ are binary vectors.
factorized posterior

$$p(\mathbf{h}|\mathbf{v}, \theta) = \prod_{k} p(h_k|\mathbf{v}, \theta)$$

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 visiable (other models exist, see Table 27.2)

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

$$E(\mathbf{v}, \mathbf{h}; \theta) = \mathbf{v}^\top \mathbf{W} \mathbf{h} \tag{6}$$

$$p(\mathbf{h}|\mathbf{v}, \theta) = \prod_k \mathrm{Ber}(h_k | sigm(\mathbf{w}_{:,k}^\top, \mathbf{v})) \tag{7}$$

$$p(\mathbf{v}|\mathbf{h}, \theta) = \prod_r \mathrm{Ber}(v_r | sigm(\mathbf{w}_{r,:}^\top, \mathbf{h})) \tag{8}$$

Restricted Boltzmann machines

Goal: maximize $p(\mathbf{v}|\theta)$

$$\nabla_{\mathbf{w}} l = E_{p_e mp(\cdot|\theta)}[\mathbf{v}\mathbf{h}^\top] - E_{p(\cdot|\theta)}[\mathbf{v}\mathbf{h}^\top]$$

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 retriver, data miners find correct & fast solutions for them...