Contents

Indian Institute of Technology Kanpur
CS777 Topics in Learning Theory

Authors: Rushab Munot(14405), Ameya Loya(14080)
Instructor: Purushottam Kar
Date: March 5, 2018

Paper Survey

Inference in Dirichlet Mixture Models and A Probabilistic Approach to Sense Embeddings

1 Inference in Dirichlet Mixture Models

1.1 Introduction

A lot of problems have a hidden/latent structure that generates the observed data. In cases as such we use unsupervised learning to recover the latent structure. However, one of the problems faced is determining the complexity of latent structure. We have no way of knowing the number of clusters or dimensions or variables that fit the data well. One way of going around this problem is to assume the complexity and then choose the model that gives the best performance. an alternative to this approach is non parameterize the latent structure i.e assume unbounded complexity and that the observations are manifestations of only a subset of these classes. We look at the latter way of approaching the problems with latent structure.

1.2 Problem Statement

We have n observations $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$. Each observation has D observable properties i.e. $\boldsymbol{x}^i \in \mathbb{R}^D$ for all $i \in \{1, \ldots, n\}$. Let there be K classes. Each observation belongs to one and only one of these class. We introduce the class assignment indicator variables c_i for each observation $\boldsymbol{x}^i$ such that $c_i \in \{1, \ldots, K\}$. We have(assuming Gaussian mixture model)

$$P(\boldsymbol{x}^i \mid \theta_1, \ldots, \theta_K) = \sum_{j=1}^{k=K} \pi_j \mathcal{N}(\boldsymbol{x}^i \mid \boldsymbol{\mu_j}, S_j) \tag{1}$$

where $\theta_j = (\pi_j, \boldsymbol{\mu_j}, S_j)$. Here π_j, $\boldsymbol{\mu}_j$ and S_j are the mixing proportion, mean of the Gaussian and precision of the jth class respectively.

We define the following priors on the parameters of our model :

$$P(\boldsymbol{x}^i \mid c_i) = \mathcal{N}(\boldsymbol{x}^i \mid \boldsymbol{\mu}_j, S_j) \tag{2}$$

$$P(c_i \mid \boldsymbol{\pi}) = \text{Discrete}(c_i \mid \pi_1, \ldots, \pi_K) \tag{3}$$

$$P(\boldsymbol{\mu}_j, S_j) = P(\boldsymbol{\mu}_j \mid S_j) P(S_j) = \mathcal{N}(\boldsymbol{\mu}_j \mid \boldsymbol{\xi}, (\rho S_j)^{-1}) \mathcal{W}(S_j \mid \beta, (\beta W)^{-1}) \tag{4}$$

$$P(\pi \mid \alpha) = \text{Dir}(\alpha/K, \ldots, \alpha/K) \tag{5}$$

Thus we have,

$$P(\boldsymbol{c} \mid \boldsymbol{\pi}) = \prod_{j=1}^{j=K} {\pi_j}^{n_j} \tag{6}$$

where n_j is the number of observations assigned to cluster j.

1.3 Dirichlet Process Mixtures

We first find out the probability of an assignment given the hyperparameter α. To non parameterize our mixture model we take $K \to \infty$. Also,

$$P(\boldsymbol{c} \mid \alpha) = \int P(\boldsymbol{c} \mid \boldsymbol{\pi}) P(\boldsymbol{\pi} \mid \alpha) d\boldsymbol{\pi} = \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)} \prod_{j=1}^{j=K} \frac{\Gamma(n_j + \alpha/K)}{\Gamma(\alpha/K)} \tag{7}$$

We now turn our attention to individual indicators :

$$P(c_i = j \mid \boldsymbol{c}_{-i}, \alpha) = \frac{n_{-i,j} + \alpha/K}{n + \alpha - 1} \tag{8}$$

where $n_{-i,j}$ is the number of data points associated with jth cluster excluding x_i. Taking the limit,

$$P(c_i = j \mid \boldsymbol{c}_{-i}, \alpha) = \frac{n_{-i,j}}{n + \alpha - 1} \tag{9}$$

Combining for all i, probability that an observation belongs to a new cluster is

$$P(c_i \neq c_j \forall i \neq j \mid \alpha) = \frac{\alpha}{n - 1 + \alpha} \tag{10}$$

2 A Probabilistic Approach to Sense Embeddings

To model any language, we must have a way to represent its words. One way coudl be to represent every word with a one-hot vector corresponding to its dictionary position but that won't contain any useful semantic information, as in distances between word vectors will only denote the difference in their alphabetic ordering. A better approach will be where words with similar meanings represent nearby points in vector space

Vilnis and McCallum proposed one method which represented words by a whole Gaussian distribution instead of a single one point vector and the model learnt its mean and covariance matrix. The idea of point embedding was captured by the mean vector of the Gaussian distribution along with the extra useful information like probability mass and uncertainity across a set of semantics provided by the full distribution.

However, a one vector per word approach is inadequate to model polysemous words. A single word can have multiple meanings when used in different contexts. Consider two senses of the word bank (which has many more senses) - one pertaining to the financial sense, and the other to the bank of a river. These two senses of bank are hardly related to each other in any way, however both of them have the same vector, which is sort of a weighted combination of the two senses.

Since a Gaussian distribution can have only one mode, the learned uncertainty for a polysemous word (words with multiple distinct meanings), can be overly diffuse in order for the model to assign some density to any plausible semantics. Moreover, the mean of the Gaussian can be pulled in many opposing directions, leading to a biased distribution that centers its mass

mostly around one meaning while leaving the others not well represented. Thus, we need a better model that can learn multiple vectors per word, ever vector corresponding to the sense of a word.

2.1 Multimodal Word Distribution by Athiwartkun et. al.

Athiwaratkun and Wilson proposed a probabilistic word embedding that can capture multiple meanings. They modeled each word with a mixture of Gaussians and learnt the parameters using a maximum margin energy-based rankin objective where the energy function describes the affinity between a pair of words.

They represented each word w as a Gaussian mixture with K components such that the density function of w is given by:

$$p_w(\mathbf{x}) = \sum_{i=1}^{K} p_{w,i}\mathcal{N}[\mathbf{x}; \mu_{w,i}, \Sigma_{w,i}]$$

$$= \sum_{i=1}^{K} \frac{p_{w,i}}{\sqrt{2\pi|\Sigma_{w,i}|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_{w,i})^T\Sigma_{w,i}^{-1}(\mathbf{x}-\mu_{w,i})}$$

where $\sum_{i=1}^{K} p_{w,i} = 1$.

So the model has three parameters - $p_{w,i}$ representing the component probability (mixture weight), $\mu_{w,i}$ representing the location of the i_{th} component of word w (similar to point embeddings) and $\Sigma_{w,i}$, covariance matrix of i_{th} Gaussian and contains the uncertainty information. Learning these three parameters draws inspiration from the continuous skip-gram model (**?** Mikolov et al., 2013a), where word embeddings are trained to maximize the probability of observing a word given another nearby word.

2.2 Energy Function

Consider two pairs of words (w, c) and (w, c') where w is sampled from a sentence in a corpus and c is a nearby word within a context window of length (say l) and c is a negative context word. For example, a word w = 'rock' which occurs in the sentence 'I listen to rock music' has context words (I, listen, to, music) and c' is a negative context word obtained from random sampling (e.g. tortoise). The idea behind the energy function is to maximize the energy between words that occur near each other (w and c) and minimize the energy between w and c'.

Athiwaratkun and Wilson used a max-margin ranking objective used for Gaussian embeddings which pushes the similarity of a word and its positive context higher than that of its negative context by a margin m:

$$L_\theta(w, c, c') = max(0, m - logE_\theta(w, c) + logE_\theta(w, c'))$$

where for gaussian mixtures f and g representing words w_f and w_g, log-energy is:

$$logE_\theta(f, g) = \int \left(\sum_{i=1}^{K} p_i\mathcal{N}(x; \mu_{f,i}, \Sigma_{f,i})\right) \times \left(\sum_{i=1}^{K} q_i\mathcal{N}(x; \mu_{g,i}, \Sigma_{g,i})\right) dx$$

$$= log \sum_{j=1}^{K}\sum_{i=1}^{K} p_i q_j e^{\xi_{i,j}}$$

where

$$\xi_{i,j} \equiv log\mathcal{N}(0; \mu_{f,i} - \mu_{g,j}, \Sigma_{f,i} + \Sigma_{g,j})$$

$$= -\frac{1}{2}log(det(\Sigma_{f,i} + \Sigma_{g,j})) - \frac{D}{2}log(2\pi) - \frac{1}{2}(\mu_{f,i} - \mu_{g,j})^T \Sigma_{w,i}^{-1}(\mu_{f,i} - \mu_{g,j})$$

They minimized this objective by mini-batch stochastic gradient descent with respect to mean vectors ($\mu_{\mathbf{w,i}}$), covariance matrices ($\Sigma_{w,i}$) and mixture weights ($p_{w,i}$) as parameters.

3 References

- Mikolov Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space
- Luke Vilnis and Andrew McCallum. "Word representations via gaussian embedding"
- Radford M. Neal, June 2000, Markov Chain Sampling Methods for Dirichlet Process Mixture Models
- Dilan Gorur and Carl Edward Rasmussen Dirichlet Process Gaussian Mixture Models: Choice of the Base Distribution