Infinite Latent Feature Models and the Indian Buffet Process

Youyuan Zhang
Department of Statistical Science
Duke University
youyuan.zhang@duke.edu

Lin Xiao
Department of Statistical Science
Duke University
lin.xiao@duke.edu

Abstract

By unveiling the latent variables that can generate observed properties of objects is one of the most fundamental issues in unsupervised learning. However, one of the crucial problems of unsupervised learning algorithms is to detect the latent structure. In K-means problem for example, we need to determine the number of clusters. One classic way is by performing model selection, while the other way is to use a Bayesian nonparametric method. One important method of Bayesian nonparametric method is the Indian buffet process (IBP), which is a stochastic process that provides a probability distribution over equivalence classes of binary matrices of bounded rows and potentially infinite columns. In this report, we first implement the Indian buffet process by Gibbs sampling and Metropolis-Hasting algorithm in python. For improvement in efficiency, we perform matrix calculation optimization and utilize JIT, Cython and parallel programming decrease the computation duration. Finally, we use Code Test for checking the validity and effectiveness of our acceleration and optimization.

1 Introduction

1.1 Indian Buffet Process

Indian restaurants in London offer buffets with an apparently infinite number of dishes. Indian buffet process (IBP) is a distribution over infinite binary matrices, specifying how customers (objects) choose dishes (features).

N customers enter a restaurant one after another. Each customer is exposed to a buffet consisting of infinitely many dishes arranged in a line. The first customer starts at the left of the buffet and takes a serving from each dish, stopping after certain number of dishes which follows a Poisson(α) distribution. The ith customer moves along the buffet, sampling dishes based on their "popularity". The ith customer takes dish k with probability $\frac{m_k}{i}$, where m_k is the number of previous customers who have sampled that dish. Having reached the end of all previous sampled dishes, the ith customer then tries a Poisson($\frac{\alpha}{i}$) number of new dishes. In conclusion, except the first customer(object), all the following customers(objects)'s choice can be divided into two parts. The first part depends on all the previous information and can be treated as Bernoulli trials, while the second part goes beyond the number of dishes from before and the number of "new dishes" follows a Poisson distribution with a new rate of $\frac{\alpha}{i}$.

We can indicate which customers chose which dishes using a binary matrix Z with N rows and infinitely many columns, where $z_{ik} = 1$ if the ith customer sampled the kth dish.

1.2 Infinite Latent Feature Model

In our finite model, the D-dimensional vector of properties of an object i, x_i is generated from a Gaussian distribution with mean $z_i A$ and covariance matrix $\Sigma X = \sigma_X^2$ I, where z_i is a K-dimensional binary vector, and A is a $K \times D$ matrix of weights. In matrix notation, E[X] = ZA. If Z is a feature matrix, this is a form of binary factor analysis. The distribution of X given Z, A, and σ X is matrix Gaussian with mean ZA and covariance matrix σ_X^2 I, where I is the identity matrix. The prior on A is also matrix Gaussian, with mean 0 and covariance matrix σ_X^2 I. Integrating out A, we have:

$$P(X|Z,\sigma_X,\sigma_A) = \frac{1}{(2\pi)^{ND/2}(\sigma_X)^{(N-K)D}(\sigma_A)^{KD}(|Z^TZ + \frac{\sigma_X^2}{\sigma_A^2}I|)^{D/2}} exp\{-\frac{1}{2\sigma_X^2}tr(X^T(I - Z(Z^TZ + \frac{\sigma_X^2}{\sigma_A^2}I)^{-1}Z^T)X)\} \quad (1)$$

2 Implementation

2.1 Data structure

- Feature binary matrix Z with entry either 0 or 1
- Number of new features K_{new} having a Poisson distribution
- Parameter α: K_{new}'s distribution parameter.
- σ_X^2: Covariance of Gaussian Distribution on x
- σ_A^2: Covariance of Gaussian Distribution prior on A

2.2 Algorithm for Metropolis Hastings

We will derive the posteriors analytically and update posteriors. And we will run Metropolis Algorithm on σ_X^2, σ_A^2, Z, and Gibbs sampling on K_{new}, α.

2.2.1 Priors

2.2.2 Gibbs samplers

Full conditional posterior for Z is:

$$P(z_{ik}|X, Z_{-(i,k),}, \sigma_X, \sigma_A) \propto P(X|Z,\sigma_X,\sigma_A) * P(z_{ik} = 1|\mathbf{z}_{-i,k})$$