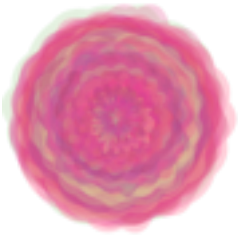

Okada Algorithm
Private Invention
Research Laboratory
Shizuoka City, Japan
Masashi Okada
okadaalgorithm@gmail.com
https://github.com/jirotubuyaki

A Vignette
CRPClustering version 1.3
2018-08-01

# CRPClustering: An R Package for Bayesian Nonparametric Chinese Restaurant Process Clustering with Entropy

## Abstract

Clustering is a scientific method which finds the clusters of data and many related methods are traditionally researched for a long time. Bayesian nonparametrics is statistics which can treat models having infinite parameters. Chinese restaurant process is used in order to compose Dirichlet process. The clustering which uses Chinese restaurant process does not need to decide the number of clusters in advance. This algorithm automatically adjusts it. Then, this package can calculate clusters in addition to entropy as the ambiguity of clusters.

## Introduction

Clustering is a traditional method in order to find the clusters of data and many related methods are studied for several decades. The most popular method is called as K-means[2]. K-means is an algorithmic way in order to search the clusters of data. However its method needs to decide the number of clusters in advance. Therefore if the data is both high dimensions and a complex, deciding the accurate number of clusters is difficult and normal Bayesian methods are too. For that reason, Bayesian nonparametric methods are gradually important as computers are faster than ever. In this package, we implemented Chinese restaurant process clustering (CRP)[4]. CRP can compose infinite dimensional parameters as Dirichlet process[1]. It acts like customers who sit at tables in a restaurant and has a probability to sit at a new table. As a result, Its model always automates clustering. Moreover, we added the method which calculates the entropy[5] of clusters into this package. It can check the ambiguity of the result. Then, we explain the clustering model and how to use it in detail. Finally, an example is plotted on a graph.

## Background

### Chinese Restaurant Process

Chinese restaurant process is a metaphor looks like customers sit at a table in Chinese restaurant. All customers except for $x_i$ have already sat at finite tables. A new customer $x_i$ will sit at either a table which other customers have already sat at or a new table. A new customer tends to sit at a table which has the number of customers more than other tables. A probability equation is given by

$$
\begin{aligned}
&p(z_i = k | x_{1:n}, z_{1:n}^{\backslash i}, \alpha, \mu_0, \rho_0, a_0, b_0) \\
&= \begin{cases}
p(x_i|\mu_k, \tau) \times \frac{n_k^{\backslash i}}{n-1+\alpha} & if \quad k \in K^+(Z_{1:n}^{\backslash i}), \\
p(x_i|\mu_k, \tau) \times \frac{\alpha}{n-1+\alpha} & \mu_k \sim N(\mu_0, (\tau\rho_0)^{-1} I) \qquad if \quad k = |K^+(Z_{1:n}^{\backslash i})| + 1.
\end{cases}
\end{aligned} \tag{1}
$$

where $n_k^{\backslash i}$ denotes the number of the customers at a table $k$ except for $i$ and $\alpha$ is a concentration parameter.

## Markov Chain Monte Carlo Methods for Clustering

Markov chain Monte Carlo (MCMC) methods[3] are algorithmic methods to sample from posterior distributions. If conditional posterior distributions are given by models, it is the best way in order to acquire parameters as posterior distributions. The algorithm for this package is given by

i) Sampling $z_i$ for each $i$ $(i = 1, 2, \cdots, n)$

$$
\begin{aligned}
& p(z_i = k | x_{1:n}, z_{1:n}^{\backslash i}, \alpha, \mu_k, \mu_0, \Sigma_k, \rho_0) \\
& = \begin{cases} p(x_i|\mu_k, \Sigma_k) \times \frac{n_k^{\backslash i}}{n-1+\alpha}, \\ p(x_i|\mu_k, \Sigma_k) \times \frac{\alpha}{n-1+\alpha} \qquad \mu_k \sim N(\mu_0, \Sigma_{table}). \end{cases}
\end{aligned} \tag{2}
$$

$$
z_i \sim Multi(p(z_i = 1), p(z_i = 2), \cdots, p(z_i = \infty)), \tag{3}
$$

where $k$ is a $k$ th cluster and $i$ is a $i$ th data. $mu$ and $sigmatable$ arguments in the "crp_gibbs" function are generating $\mu_k$ parameter for new table.

ii) Sampling $\mu_k$ for each $k$ $(k = 1, 2, \cdots, \infty)$

$$
\mu_k \sim p(\mu_k | x_{1:n}, z_{1:n}, \Sigma_k, \mu_0, \rho_0) = N(\mu_k | \frac{n_k}{n_k + \rho_0} \bar{x_k} + \frac{\rho_0}{n_k + \rho_0} \mu_0, \Sigma_k), \tag{4}
$$

$$
\Sigma_{k_{ij}} = \frac{n_k}{n_k + \rho_0} Cov(x_{ki}, x_{kj}) + \frac{\rho_0}{n_k + \rho_0} (\bar{x_{ki}} - \mu_{0i})(\bar{x_{kj}} - \mu_{0j}), \tag{5}
$$

$$
\bar{x_k} = \frac{1}{n_k} \sum_{i=1}^{n} \delta(z_i = k) x_i. \tag{6}
$$

Iterations i) ii) continue on many times, and $\Sigma_k$ is a variance-covariance matrix of $k$th cluster. $i$ and $j$ are rows and columns' number of $\Sigma_k$. $\rho_0$ is a argument $ro0$ in "crp_gibbs" function. First several durations of iterations which are called as "burn in" are error ranges. For that reason, "burn in" durations are abandoned.

## Clusters Entropy

Entropy denotes the ambiguity of clustering. As a result of a simulation, data $x_i$ joins in a particular table. From the total numbers $n_k$ of the particular table $k$ at the last iteration, a probability $p_k$ at each cluster $k$ is calculated. The entropy equation is given by

$$
Entropy = -\sum_{k=1}^{\infty} \frac{n_k}{n} \log_2 \frac{n_k}{n}. \tag{7}
$$

# Installation

CRPClustering is available through GitHub (https://github.com/jirotubuyaki/CRPClustering) or CRAN (https://CRAN.R-project.org/package=CRPClustering). If download from GitHub, you can use devtools by the commands:

```
> library(devtools)
> install_github("jirotubuyaki/CRPClustering")
```

Once the packages are installed, it needs to be made accessible to the current R session by the commands:

```
> library(CRPClustering)
```

For online help facilities or the details of a particular command (such as the function crp_gibbs) you can type:

```
> help(package="CRPClustering")
```

# Methods

## Implemented by Scala

This package is implemented by Scala. Please install Scala compiler version 2.12.6. Programs are compiled java archive. If you are interested in source codes by Scala. Please check GitHub (https://github.com/jirotubuyaki/CRPClustering).

## Method for Chinese Restaurant Process Clustering

```
> result <- crp_train(as.matrix(data),
                            mu=c(0,0),
                            sigma_table=1,
                            alpha=1,
                            ro_0=1,
                            burn_in=100,
                            iteration=1000
                          )
```

This method calculates CRP clustering. Let arguments be:

- data: a matrix of data for clustering. row is each data $i$ and column is dimensions of each data $i$.
- mu: a vector of center points of data. If data is 3 dimensions, a vector of 3 elements like "c(2,4,7)".
- sigma_table: a numeric of table position variance.
- alpha: a numeric of a CRP concentration rate.
- ro_0: a numeric of a CRP mu change rate.
- burn_in: an iteration integer of burn in.
- iteration: an iteration integer.

Let return be:

- result: an array denotes result clusters. First colum is cluster number. Second is joined data number for each cluster. From next, result clusters' mean and variance matrix.

## Predict Cluster Data Joined

```
> predict <- crp_predict(as.matrix(data), result)
```

Let arguments be:

- data: a matrix of data for clustering. row is each data $i$ and column is dimensions of each data $i$.
- result: return result from method "crp_train".

Let return be:

- predict: an array denotes first colum is joined cluster and nexts are joined probability for each result cluster.

## Visualization Method

```
> crp_plot(as.matrix(data), predict)
```

This method exhibits multi dimentional plot matrix. Let arguments be:

- data: a matrix of data for clustering. Row is each data $i$ and column is dimensions of each data $i$.
- predict: return predict from method "crp_predict".

## Example

Data is generated from ten normal distributions and parameters are set as $mu = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $\alpha = 1$, $sigmatable = 1$, $\rho_0 = 1$, $burnin = 100$, $iteration = 1000$. The result is plotted on a graph and each data joins in any cluster. The graph is given by below:

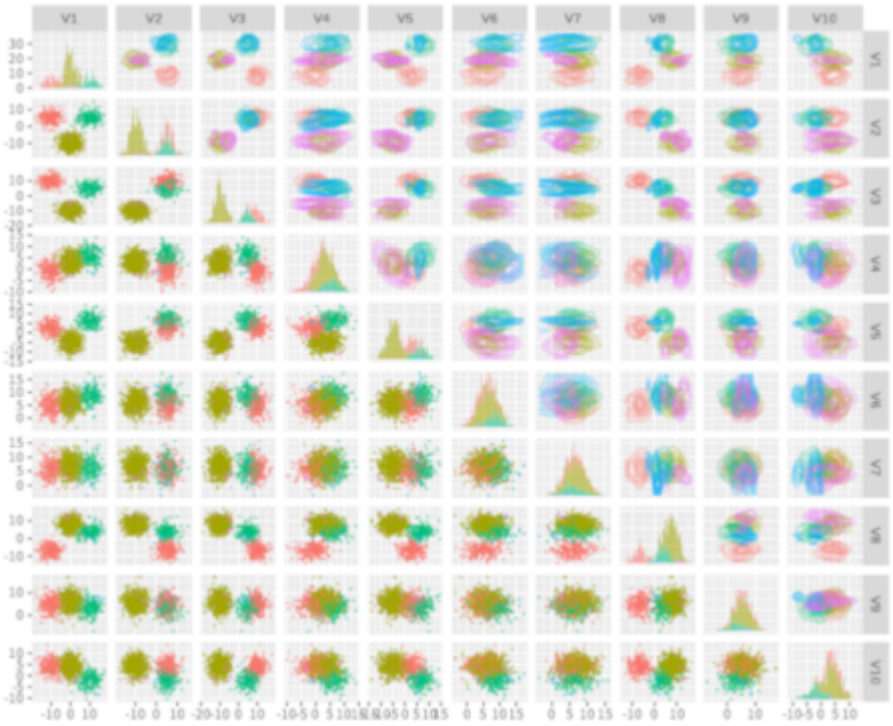


Figure 1: Result of clustering

## Conclusions

Chinese restaurant process clustering was implemented and explained how to use it. Computer resources are limited. Computer processing power is the most important problem. After this, several improvements are planed. Please send suggestions and report bugs to okadaalgorithm@gmail.com.

## Acknowledgments

This activity would not have been possible without the support of my family and friends. To my family, thank you for much encouragement for me and inspiring me to follow my dreams. I am especially grateful to my parents, who supported me all aspects.

# References

[1] Ferguson, Thomas. Bayesian analysis of some nonparametric problems. Annals of Statistics. 1 (2): 209–230., 1973.

[2] Hartigan, J. A.; Wong, M. A. Algorithmas136: A k-means clustering algorithm . Journal of the Royal Statistical Society, Series C. 28 (1): 100–108. JSTOR 2346830, 1979.

[3] Liu, Jun S. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. Journal of the American Statistical Association 89 (427): 958–966., 1994.

[4] Pitman, Jim. Exchangeable and partially exchangeable random partitions. Probability Theory and Related Fields 102 (2): 145–158., 1995.

[5] Elliott H. Lieb; Jakob Yngvason. The physics and mathematics of the second law of thermodynamics. Physics Reports Volume:310 Issue:1 1-96., 1999.