Package 'nonparametric.bayes'

October 13, 2022

Title Project Code - Nonparametric Bayes

Version 0.0.1

Description
Basic implementation of a Gibbs sampler for a Chinese Restaurant Process along with some visual aids to help understand how the sampling works. This is developed as part of a postgraduate school project for an Advanced Bayesian Nonparametric course. It is inspired by Tamara Broderick's presentation on Nonparametric Bayesian statistics given at the Simons institute.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports mvtnorm, progress

NeedsCompilation no

Author Erik-Cristian Seulean [aut, cre]
(<https://orcid.org/0000-0002-1444-1574>)

Maintainer Erik-Cristian Seulean <erikseulean@gmail.com>

Repository CRAN

Date/Publication 2021-11-29 09:50:05 UTC

R topics documented:

`cluster_datapoints` *Gibbs sampling for the Chinese Restaurant Process Implementation details can be found in the associated paper The algorithm stops at every 1000th iteration and prints the current cluster configuration.*

Description

Gibbs sampling for the Chinese Restaurant Process Implementation details can be found in the associated paper The algorithm stops at every 1000th iteration and prints the current cluster configuration.

Usage

```
cluster_datapoints(
  data,
  sd = 1,
  initialisation = rep(1, nrow(data)),
  sigma0 = matrix(c(1, 0, 0, 1), nrow = 2, byrow = TRUE)
)
```

Arguments

`data`	A matrix of nx2 containing the datapoints
`sd`	Prior standard deviation
`initialisation`	Cluster initialisation for each datapoint. Default initialisation is to set every point in the same cluster.
`sigma0`	Covariance matrix for the points. Default initialisation is set to matrix(c(1, 0, 0, 1), mrow=2, byrow=TRUE)

Value

Returns the cluster assignments after the last iteration. Examples cluster_datapoints(generate_split_data(350, 0.5)$x, sigma0=diag(3^2, 2)) cluster_datapoints(petal, sigma0=petal_sigma0) cluster_datapoints(width, sigma0=width_sigma0) cluster_datapoints(mixed, sigma0=mixed_sigma0)

`generate_dirichlet_clusters` *Draws from a Dirichlet distribution and shows the clusters that were generated by this draw. Varying alpha, will put more or less mass in the first clusters compared to higher clusters (rhos).*

Description

Draws from a Dirichlet distribution and shows the clusters that were generated by this draw. Varying alpha, will put more or less mass in the first clusters compared to higher clusters (rhos).

Usage

```
generate_dirichlet_clusters(a, K)
```

Arguments

a	Parameter that will be passed in to a Gamma distribution in order to draw from the Dirichlet distribution.
K	Number of clusters to draw

Value

No return value

Examples

```
generate_dirichlet_clusters(10, 10)
generate_dirichlet_clusters(0.5, 30)
```

---

generate_dirichlet_clusters_with_sampled_points

Draws from a Dirichlet distribution and shows the clusters that were generated by this draw. Additionally, adds points to these clusters and shows which clusters are occupied

---

Description

Each point is generated one at a time, need to hit enter to generate a new point. Typing "x" will stop the clustering and the function will return.

Usage

```
generate_dirichlet_clusters_with_sampled_points(n, a, K)
```

Arguments

n	Number of points to be drawn in the clusters
a	Parameter that will be passed in to a Gamma distribution in order to draw from the Dirichlet distribution.
K	Number of clusters to draw

Value

No return value

Examples

```
generate_dirichlet_clusters_with_sampled_points(15, 0.5, 20)
```

`generate_split_data` *Generates a dataset used to exemplify clustering The cluster centers are set relatively far away to see how well the algorithm performs in simple scenarios*

Description

Generates a dataset used to exemplify clustering The cluster centers are set relatively far away to see how well the algorithm performs in simple scenarios

Usage

```
generate_split_data(n, sd)
```

Arguments

`n`	Number of datapoints to generate
`sd`	Standard deviation from the cluster center

Value

Returns the datapoints and the cluster assignments. The cluster assignments can be used to calculate the performance of the clustering.

`rdirichlet` *Generate a sample from a Dirichlet distirbution Using: https://en.wikipedia.org/wiki/Dirichlet_distribution#Random_number_generation*

Description

Generate a sample from a Dirichlet distirbution Using: https://en.wikipedia.org/wiki/Dirichlet_distribution#Random_number

Usage

```
rdirichlet(n, alpha)
```

Arguments

`n`	Number of observations.
`alpha`	A vector containing the parameters for the Dirichlet distribution.

Value

A sample of n observations from the Dirichlet distribution.

Examples

```
rdirichlet(n=1, alpha=c(2, 2, 2))
```

rDPM *Sequentially generate draws from a Dirichlet process mixture model, by showing step by step the iterations taken. The plot is centered at 0, with x and y from -5 to 5. The mixture draws the centres for clusters from a Normal distribution with mean mu and standard deviation sigma_0 Additional to plotting the points, it also returns the points sampled.*

Description

Hit enter to keep drawing until max n or type "x" to exit.

Usage

```
rDPM(n, alpha, mu, sigma_0, sigma)
```

Arguments

n	Number of observations.
alpha	Alpha corresponding to GEM(alpha) used to draw the rho vector.
mu	Mean of the Normal distribution used to draw the clusters.
sigma_0	Standard deviation of the Normal distribution used to draw the points around the cluster centre.
sigma	Standard deviation for cluster centers

Value

Returns the n observations sampled from the DPMM distribution.

Examples

```
rDPM(n=30, alpha=3, mu=0, sigma_0=1.5, sigma=0.7)
```

rDPM_visual *Sequentially generate draws from a Dirichlet process mixture model, by showing step by step the iterations taken. The plot is centered at 0, with x and y from -5 to 5. The mixture draws the centres for clusters from a Normal distribution with mean mu and standard deviation sigma_0*

Description

Hit enter to keep drawing until max n, type x to exit.

Usage

```
rDPM_visual(n, alpha, mu, sigma_0, sigma)
```

Arguments

n	Number of observations.
alpha	Alpha corresponding to GEM(alpha) used to draw the rho vector.
mu	Mean of the Normal distribution used to draw the clusters.
sigma_0	Standard deviation of the Normal distribution used to draw the points around the cluster centre.
sigma	Standard deviation for the cluster centre.

Value

Returns the n observations sampled from the DPMM distribution.

Examples

```
rDPM_visual(n=30, alpha=3, mu=0, sigma_0=1.5, sigma=0.7)
```

Index