

# Package: gourd (via r-universe)

May 21, 2026

**Title** Shrinkage Partition Distribution

**Version** 0.2.20

**Description** Implementation of the distribution introduced in our paper  
``Dependent Random Partitions by Shrinking Toward an Anchor''.

**License** MIT + file LICENSE | Apache License 2.0

**Depends** R (>= 4.3)

**Suggests** fields

**SystemRequirements** Cargo (Rust's package manager), rustc

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Config/Roxido/TemplateVersion** 26.03.14.08.27

**Config/Roxido/MinCLIVersion** 0.8.0

**Config/pak/sysreqs** libclang-dev

**Repository** <https://dbdahl.r-universe.dev>

**Date/Publication** 2026-03-14 15:06:13 UTC

**RemoteUrl** <https://github.com/dbdahl/dbdahl.r-universe.dev>

**RemoteRef** pkg/gourd

**RemoteSha** 09e19c4c8b72c509171244773409301c7e1881e1

## Contents

CenteredPartition . . . . .	2
CRPPartition . . . . .	3
FixedPartition . . . . .	3
JensenLiuPartition . . . . .	4
LocationScalePartition . . . . .	4
mkDistrPtr . . . . .	5
prPartition . . . . .	6
samplePartition . . . . .	7



---

CRPPartition	<i>Probabilities for the Chinese Restaurant Process (CRP) Partition Distribution</i>
--------------	--

---

**Description**

This function specifies the Chinese restaurant process (CRP) partition distribution for given concentration and discount parameters.

**Usage**

```
CRPPartition(nItems, concentration, discount = 0)
```

**Arguments**

nItems	The number of items in each partition, as an integer greater than or equal to 1.
concentration	The concentration parameter, as a numeric value greater than $-1 * discount$ .
discount	The discount parameter, as a numeric value in $[0, 1)$ .

**Value**

An object of class PartitionDistribution representing this partition distribution.

**Examples**

```
CRPPartition(nItems=5, concentration=1.5, discount=0.1)
```

---

FixedPartition	<i>Probabilities for the Fixed Partition Distribution</i>
----------------	---

---

**Description**

This function specifies the fixed partition distribution for a given anchor partition. This is a point-mass distribution at the anchor partition.

**Usage**

```
FixedPartition(anchor)
```

**Arguments**

anchor	An integer vector giving the anchor partition (a.k.a., center partition, location partition).
--------	---

**Value**

An object of class PartitionDistribution representing this partition distribution.

**Examples**

```
FixedPartition(anchor=c(1,1,1,2,2))
```

---

JensenLiuPartition      *Probabilities for the Jensen Liu Partition Distribution*

---

**Description**

This function specifies the Jensen Liu (2008) partition distribution for given concentration and permutation parameters.

**Usage**

```
JensenLiuPartition(concentration, permutation)
```

**Arguments**

concentration      The concentration parameter, as a numeric value greater than  $-1 * discount$ .  
 permutation      A vector containing the integers  $1, 2, \dots, n$  giving the order in which items are allocated to the partition.

**Value**

An object of class PartitionDistribution representing this partition distribution.

**Examples**

```
JensenLiuPartition(concentration=0.5, permutation=c(1,5,4,2,3))
```

---

LocationScalePartition      *Probabilities for the Location Scale Partition Distribution*

---

**Description**

This function specifies the location scale partition distribution for given anchor partition, shrinkage, and permutation parameters. The shrinkage parameter is the reciprocal of the scale parameter.

**Usage**

```
LocationScalePartition(anchor, shrinkage, concentration, permutation)
```

**Arguments**

anchor	An integer vector giving the anchor partition (a.k.a., center partition, location partition).
shrinkage	A scalar giving the value of the shrinkage parameter, i.e., the reciprocal of the scale parameter.
concentration	The concentration parameter, as a numeric value greater than $-1 * discount$ .
permutation	A vector containing the integers $1, 2, \dots, n$ giving the order in which items are allocated to the partition.

**Value**

An object of class PartitionDistribution representing this partition distribution.

**Examples**

```
scale <- 0.1
LocationScalePartition(anchor=c(1,1,1,2,2), shrinkage=1/scale,
                       concentration=0.6, permutation=c(1,5,4,2,3))
```

---

mkDistrPtr

---

*Make a Pointer to Partitional Distribution Parameters*


---

**Description**

Users should not call this function. This is an internal function exported for the sake of developers of packages depending on this package.

**Usage**

```
mkDistrPtr(distr, excluded = NULL, included = NULL)
```

**Arguments**

distr	An object of class 'PartitionDistribution'.
excluded	A character vector of explicitly excluded partition distributions.
included	A character vector of explicitly included partition distributions.

**Value**

A pointer

---

prPartition                      *Compute the (Log of) the Probability of a Partition*

---

### Description

This function computes (the log of) the probability of the supplied partitions for the given partition distribution.

### Usage

```
prPartition(distr, partition, log = TRUE)
```

### Arguments

distr	A specification of the partition distribution, i.e., an object of class <code>PartitionDistribution</code> as returned by, for example, a function such as <a href="#">CRPPartition</a> .
partition	A matrix of integers giving cluster labels on the rows. For partition $k$ , items $i$ and $j$ are in the same cluster if and only if <code>partition[k, i] == partition[k, j]</code> .
log	A logical indicating whether the probability (FALSE) or its natural logarithm (TRUE) is desired.

### Value

A numeric vector giving either probabilities or log probabilities for the supplied partitions.

### See Also

[CRPPartition](#), [ShrinkagePartition](#), [LocationScalePartition](#), [CenteredPartition](#), [samplePartition](#)

### Examples

```
concentration <- 1.0
discount <- 0.1
nSamples <- 3

distr <- CRPPartition(nItems=5, concentration=concentration, discount=discount)
x <- samplePartition(distr, nSamples, nCores=1)
prPartition(distr, x)

anchor <- c(1,1,1,2,2)
permutation <- c(1,5,4,2,3)
n_items <- length(permutation)

distr <- ShrinkagePartition(anchor=anchor, shrinkage=c(0,0,0,0.3,0.3),
                           permutation=permutation, grit=0.2,
                           CRPPartition(nItems=n_items, concentration=concentration, discount=discount))
x <- samplePartition(distr, nSamples, nCores=1)
prPartition(distr, x)
```

---

samplePartition      *Sample From a Partition Distribution*

---

## Description

This function samples from a partition distribution.

## Usage

```
samplePartition(
  distr,
  nSamples,
  randomizePermutation = FALSE,
  randomizeShrinkage = c("fixed", "common", "cluster", "idiosyncratic")[1],
  randomizeGrit = FALSE,
  shrinkage_shape = 5,
  shrinkage_rate = 1,
  grit_shape1 = 1,
  grit_shape2 = 1,
  nCores = 0
)
```

## Arguments

distr	A specification of the partition distribution, i.e., an object of class <code>PartitionDistribution</code> as returned by, for example, a function such as <a href="#">CRPPartition</a> .
nSamples	An integer giving the number of partitions to sample.
randomizePermutation	Should the permutation be uniformly randomly sampled for each partition?
randomizeShrinkage	Should the shrinkage be random for each sample? Specifically, the shrinkage is the same for every observations and sampled from a gamma distribution with parameters <code>shrinkage_shape</code> and <code>shrinkage_rate</code> .
randomizeGrit	Should the grit of the <a href="#">ShrinkagePartition</a> be sampled from a beta distribution for each partition? This is ignored for other distributions.
shrinkage_shape	The shape parameter of the gamma distribution for randomizing the shrinkage.
shrinkage_rate	The rate parameter of the gamma distribution for randomizing the shrinkage.
grit_shape1	The first parameter of the beta distribution for randomizing the grit.
grit_shape2	The first parameter of the beta distribution for randomizing the grit.
nCores	The number of CPU cores to use. A value of zero indicates to use all cores on the system.

## Details

Note that the centered partition distribution [CenteredPartition](#) uses MCMC to sample.

**Value**

An integer matrix containing a partition in each row using cluster label notation.

**See Also**

[CRPPartition](#), [ShrinkagePartition](#), [LocationScalePartition](#), [CenteredPartition](#), [prPartition](#)

**Examples**

```
concentration <- 1.0
discount <- 0.1
nSamples <- 3

distr <- CRPPartition(nItems=5, concentration=concentration, discount=discount)
x <- samplePartition(distr, nSamples, nCores=1)
prPartition(distr, x)

anchor <- c(1,1,1,2,2)
permutation <- c(1,5,4,2,3)
n_items <- length(permutation)

distr <- ShrinkagePartition(anchor=anchor, shrinkage=c(0,0,0,0.3,0.3),
                           permutation=permutation, grit=0.2,
                           CRPPartition(nItems=n_items, concentration=concentration, discount=discount))
x <- samplePartition(distr, nSamples, nCores=1)
prPartition(distr, x)
```

---

ShrinkagePartition      *Probabilities for the Shrinkage Partition Distribution*

---

**Description**

This function specifies the shrinkage partition distribution given an anchor partition, shrinkage, permutation and a baseline distribution.

**Usage**

```
ShrinkagePartition(
  anchor,
  shrinkage,
  permutation,
  grit,
  baseline,
  optimized = TRUE
)
```

**Arguments**

anchor	An integer vector giving the anchor partition (a.k.a., center partition, location partition).
shrinkage	A numeric vector of length equal to the length of anchor (i.e., the number of items) giving the shrinkage probability for each item. This can also be a scalar, in which case that value is used for each item.
permutation	A vector containing the integers $1, 2, \dots, n$ giving the order in which items are allocated to the partition.
grit	A numeric value controlling the amount of clustering, with small values encouraging few clusters and large values encouraging more clusters. Values between 0 and 1 (exclusive) ensure that, as shrinkage goes to infinity, the partition distribution concentrates on the anchor partition.
baseline	An object of class <code>PartitionDistribution</code> representing a partition distribution. Currently, only <code>UniformPartition</code> , <code>JensenLiuPartition</code> and <code>CRPPartition</code> are supported.
optimized	When the baseline distribution is a CRP with zero discount, should optimized calculations be used?

**Value**

An object of class `PartitionDistribution` representing this partition distribution.

**Examples**

```
ShrinkagePartition(anchor=c(1,1,1,2,2), shrinkage=c(10,10,10,3,3),
  permutation=c(1,5,4,2,3), grit=0.1,
  baseline=CRPPartition(nItems=5, concentration=1.5, discount=0.1))

ShrinkagePartition(anchor=c(1,1,1,2,2), shrinkage=c(1,1,1,3,3),
  permutation=c(1,5,4,2,3), grit=0.2,
  baseline=UniformPartition(nItems=5))

ShrinkagePartition(anchor=c(1,1,1,2,2), shrinkage=c(0,0,0,3,3),
  permutation=c(1,5,4,2,3), grit=0.2,
  baseline=JensenLiuPartition(concentration=0.5, permutation=c(1,5,4,2,3)))
```

---

summarize\_prior\_on\_shrinkage\_and\_grit

*Summarize Implications of the Prior Distribution of the SP Distribution*

---

**Description**

To aid in the prior elicitation process for the shrinkage and grit parameters in the Shrinkage Partition (SP) distribution, this function computes the log of the prior joint density of the shrinkage and grit parameters and the expected Rand index and the expected entropy using samples from the SP distribution for combinations of shrinkage and grit parameters. These computed quantities can then be displayed as shown in the examples below.

**Usage**

```
summarize_prior_on_shrinkage_and_grit(
  anchor,
  shrinkage_shape = 4,
  shrinkage_rate = 1,
  grit_shape1 = 2,
  grit_shape2 = 2,
  use_crp = TRUE,
  concentration = 1,
  shrinkage_n = 25,
  grit_n = 25,
  n_mc_samples = 100,
  domain_specification = list(n_mc_samples = 1000, percentile = 0.95),
  a = 1,
  n_cores = 0
)
```

**Arguments**

<code>anchor</code>	Anchor partition in the Shrinkage Partition (SP) distribution as a numeric vector of cluster labels.
<code>shrinkage_shape</code>	Shape parameter of the gamma prior distribution for the shrinkage parameter of the SP distribution.
<code>shrinkage_rate</code>	Rate parameter of the gamma prior distribution for the shrinkage parameter of the SP distribution.
<code>grit_shape1</code>	First shape parameter of the beta prior distribution for the grit parameter of the SP distribution.
<code>grit_shape2</code>	Second shape parameter of the beta prior distribution for the grit parameter of the SP distribution.
<code>use_crp</code>	Use the Chinese restaurant process (CRP) which serves as the baseline distribution? If FALSE, the Jensen Liu partition (JLP) is used instead.
<code>concentration</code>	Concentration parameter of the baseline distribution for the SP distribution.
<code>shrinkage_n</code>	Length of the evenly-spaced grid for the shrinkage parameter of the SP distribution.
<code>grit_n</code>	Length of the evenly-spaced grid for the grit parameter of the SP distribution.
<code>n_mc_samples</code>	Number of Monte Carlo samples used to compute the expectation of the Rand index and entropy.

domain_specification	List to control the domain for the computations, containing either: 1. Elements named <code>shrinkage_lim</code> and <code>grit_lim</code> (each vectors of length two indicating the lower and upper bound), or 2. Elements named <code>n_mc_samples</code> (giving the number of samples to draw to learn the domain) and <code>percentile</code> (indicating which percentile should be used to determine the upper bound).
a	The argument <code>a</code> is a nonnegative scalar in $[0, 2]$ giving (for Binder loss) the cost of placing two items in separate clusters when in truth they belong to the same cluster. Without loss of generality, the cost (under Binder loss) of placing two items in the same cluster when in truth they belong to separate clusters is fixed $2-a$ . For VI, <code>a</code> has a similar interpretation, although is not a unit cost. See Dahl, Johnson, Müller (2021).
n_cores	Number of CPU cores to use, where 0 (default) indicates all cores.

### Value

A list containing five elements. Vectors `shrinkage` and `grit` give the locations of grid values. Matrix `log_density` gives the log of the prior density at the combinations of `shrinkage` and `grit` values. Likewise, matrices `expected_vi`, `expected_binder`, `expected_rand_index`, and `expected_entropy` give expectations at the combinations of `shrinkage` and `grit` values. See the examples below to see how one might use this output.

### References

D. B. Dahl, D. J. Johnson, and P. Müller (2022), Search Algorithms and Loss Functions for Bayesian Clustering, *Journal of Computational and Graphical Statistics*, 31(4), 1189-1201, doi:10.1080/10618600.2022.2069779.

### Examples

```
anchor <- rep(1:4, each = 13)
out <- summarize_prior_on_shrinkage_and_grit(anchor, n_mc_samples = 100, n_cores = 1)
if (requireNamespace("fields") ) {
  fields::image.plot(out$shrinkage, out$grit, out$expected_rand_index,
                    xlab = "Shrinkage", ylab = "Grit")
  contour(out$shrinkage, out$grit, exp(out$log_density), add = TRUE, labcex = 1.0)
}
image(out$shrinkage, out$grit, out$expected_entropy, xlab = "Shrinkage", ylab = "Grit")
contour(out$shrinkage, out$grit, exp(out$log_density), add = TRUE, labcex = 1.0)
```

### Description

This function specifies the uniform partition distribution.

**Usage**

```
UniformPartition(nItems)
```

**Arguments**

nItems            An integer giving the number of items in each partition.

**Value**

An object of class `PartitionDistribution` representing this partition distribution.

**Examples**

```
UniformPartition(nItems=5)
```

# Index

CenteredPartition, [2](#), [6–8](#)  
CRPPartition, [2](#), [3](#), [6–9](#)  
  
FixedPartition, [3](#)  
  
JensenLiuPartition, [2](#), [4](#), [9](#)  
  
LocationScalePartition, [4](#), [6](#), [8](#)  
  
mkDistrPtr, [5](#)  
  
prPartition, [6](#), [8](#)  
  
samplePartition, [6](#), [7](#)  
ShrinkagePartition, [6](#), [7](#), [8](#), [8](#)  
summarize\_prior\_on\_shrinkage\_and\_grit,  
[9](#)  
  
UniformPartition, [2](#), [9](#), [11](#)