

# Package ‘Barycenter’

October 12, 2022

**Encoding** latin1

**Type** Package

**Title** Regularized Wasserstein Distances and Barycenters

**Version** 1.3.1

**Date** 2018-04-05

**Author** Marcel Klatt

**Maintainer** Marcel Klatt <marcel.klatt@stud.uni-goettingen.de>

**Description** Computations of entropy regularized Wasserstein Distances, a.k.a. dual-Sinkhorn divergences, and entropy regularized Wasserstein Barycenters. Relevant papers are Marco Cuturi (2013) <[arXiv:1306.0895](#)>, Marco Cuturi (2014) <[arXiv:1310.4375](#)> and Jason Altschuler et al. <[arXiv:1705.09634](#)>.

**License** GPL-2

**LazyData** TRUE

**LinkingTo** Rcpp, RcppArmadillo

**Depends** R (>= 2.10.0), Rcpp (>= 0.11.1)

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-05-04 09:05:27 UTC

## R topics documented:

eight . . . . .	2
Greenkhorn . . . . .	2
Sinkhorn . . . . .	3
three . . . . .	4
WaBarycenter . . . . .	5
<b>Index</b>	<b>7</b>

---

eight *MNIST dataset of the digit eight (small)*

---

### Description

A list of five images of the digit eight. Each image is a 28 x 28 matrix, representing the weights of the corresponding pixels scaled s.t. they sum up to one. The images are included in a list. This dataset is based on the MNIST dataset.

### Usage

```
eight
```

### Format

List

### Examples

```
#Take a look into the dataset, e.g. the first image of the digit eight.
eight[[1]]
image(eight[[1]])
```

---

Greenkhorn *Greenkhorn Distances (approximation to EMD)*

---

### Description

The Greenkhorn algorithm to approximate the earth movers distance (EMD), a.k.a. Wasserstein distance, between two probability vectors  $r$  and  $c$  with specified cost-matrix  $costm$ .

### Usage

```
Greenkhorn(r, c, costm, lambda = 1, maxIter = 10000, tolerance=10-8)
```

### Arguments

$r$	( $n \times 1$ ) row vector in the probability simplex (nonnegative summing to one).
$c$	( $1 \times m$ ) row vector in the probability simplex (nonnegative summing to one).
$costm$	( $n \times m$ ) matrix of pairwise distances/costs between bins with mass described by $r$ and bins with mass described by $c$ .
$lambda$	Non-negative regularization parameter (for small $lambda$ the Sinkhorn Distance is close to the EMD).
$maxIter$	Maximum number of iterations.
$tolerance$	A threshold for the integrated stopping criterion based on marginal differences.

**Value**

Returns a list containing the regularized transport plan represented as a  $n \times m$  matrix as well as the Sinkhorn distance between the given marginals  $r$  and  $c$ .

**Author(s)**

Marcel Klatt

**References**

Altschuler, J., Weed, J. and Rigollet, P.: Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration, Advances in Neural Information Processing Systems 30 (NIPS 2017)

**Examples**

```
#Sinkhorn Distances between the first image to the second image in the dataset eight.
#We creat costm simply using a distance matrix on the grid [0,1]x[0,1].
n <- seq(0,1,length.out = dim(eight[[1]])[2])
costm <- as.matrix(dist(expand.grid(n,rev(n)), diag=TRUE, upper=TRUE))
r <- matrix(eight[[1]],28*28,1)
c <- matrix(eight[[2]],1,28*28)
Greenkhorn(r, c, costm)$Distance
```

---

Sinkhorn

*Sinkhorn Distances (upper bound to EMD)*


---

**Description**

The Sinkhorn algorithm to compute  $N$  dual-Sinkhorn divergences, i.e. upper bounds on the earth movers distance (EMD), a.k.a. Wasserstein distance.

**Usage**

```
Sinkhorn(a, b, costm, lambda = 1, maxIter = 10000, tolerance=10^(-8))
```

**Arguments**

$a$	Either a $d_1 \times 1$ column vector in the probability simplex (nonnegative summing to one) or a $d_1 \times N$ matrix, where each column is in the probability simplex.
$b$	A $d_1 \times N$ matrix of $N$ vectors in the probability simplex.
$costm$	A matrix of pairwise distances/costs between bins described in $a$ and bins in the $b_1, \dots, b_N$ histograms.
$lambda$	Non-negative regularization parameter (for small $lambda$ the Sinkhorn Distance is close to the EMD).
$maxIter$	Maximum number of iterations.
$tolerance$	A threshold for the integrated stopping criterion based on marginal differences.

**Value**

Returns the Sinkhorn Distances between the given bins. If  $a$  is given by a  $d_1 \times 1$  column vector the function returns the distances

$$[D(a, b_1), \dots, D(a, b_N)].$$

If  $a$  is given by a  $d_1 \times N$  matrix the function returns the distances

$$[D(a_1, b_1), \dots, D(a_N, b_N)].$$

If  $a$  and  $b$  are simply given by two  $d_1 \times 1$  and  $d_2 \times 1$  vectors each in the probability simplex, respectively, then the functions returns a list containing the Sinkhorn Distance as well as the corresponding regularized transport plan.

**Author(s)**

Marcel Klatt

**References**

Cuturi, M.: Sinkhorn Distances: Lightspeed Computation of Optimal Transport, Advances in Neural Information Processing Systems 26, 2013

**Examples**

```
#Sinkhorn Distances between the first image to the remaining four images in the dataset eight.
#We creat costm simply using a distance matrix on the grid [0,1]x[0,1].
n <- seq(0,1,length.out = dim(eight[[1]])[2])
costm <- as.matrix(dist(expand.grid(n,rev(n)), diag=TRUE, upper=TRUE))
a <- matrix(eight[[1]],28*28,1)
b <- matrix(c(eight[[2]],eight[[3]],eight[[4]],eight[[5]]),28*28,4)
Sinkhorn(a, b, costm)
```

---

three

*MNIST dataset of the digit three*

---

**Description**

A list of 4351 images of the digit three. Each image is a 28 x 28 matrix, representing the weights of the corresponding pixels scaled s.t. they sum up to one. The images are included in a large list. This dataset is based on the MNIST dataset.

**Usage**

three

**Format**

List

## Examples

```
#Take a look into the dataset, for example the 25th image of the digit three.
three[[25]]
image(three[[25]])
```

---

WaBarycenter

*Regularized Wasserstein Barycenters*

---

## Description

WaBarycenter takes in a list of matrices representing joint measures on the row and column space and outputs the corresponding Barycenter. The list has to consist of matrices having all the same dimensions, for instance, each matrix represents the normalized weights of the corresponding pixels of images.

## Usage

```
WaBarycenter(images, maxIter = 10, lambda = FALSE, costm = FALSE)
```

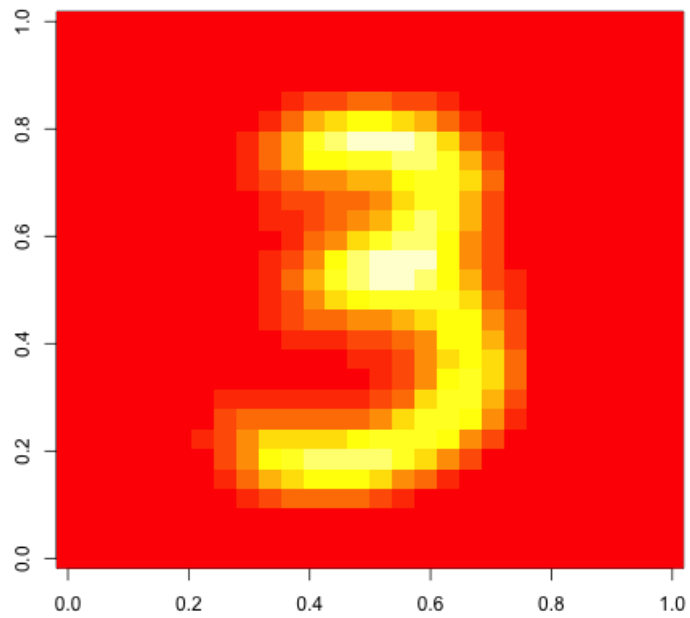
## Arguments

images	A list of matrices satisfying the prerequisites described above.
maxIter	Maximum number of iterations.
lambda	Non-negative regularization parameter (for large lambda the regularized Barycenter is close to its true counterpart). If FALSE the algorithm uses a lambda depending on costm.
costm	A matrix of pairwise distances between the locations. If FALSE the algorithm uses the usual euclidean distance matrix on a [0,1]x[0,1] equidistant pixel grid.

## Value

The Barycenter of the matrices, represented by a  $n \times m$  matrix.

Given the MNIST dataset, a Barycenter of the digit three is shown below. The Barycenter is based on 4351 images each represented by a 28 x 28 pixel grid, respectively. The values for lambda and maxIter were set by default. The dataset is also available in this package (c.f. [three](#)).

**Author(s)**

Marcel Klatt

**References**

Cuturi, M.: Fast Computation of Wasserstein Barycenters, Proceedings of the International Conference on Machine Learning, Beijing, China, 2014

**Examples**

```
#Computation of a Barycenter based on five images representing the digit eight, respectively.  
WaBarycenter(eight,lambda=10)  
#For a more reasonable but longer computation!  
## Not run: WaBarycenter(eight)
```

# Index

- \* **datasets**
  - three, [4](#)
- \* **dataset**
  - eight, [2](#)
- eight, [2](#)
- Greenhorn, [2](#)
- Sinkhorn, [3](#)
- three, [4](#), [5](#)
- WaBarycenter, [5](#)