Package 'GSCAD'

October 6, 2017

Type Package

Title Implementing GSCAD Method for Image Denoising and Inpainting

Version 0.1.0

Maintainer Simeng Qu <simengqu@gmail.com>

Description Method proposed in ``Simultaneous Sparse Dictionary Learning and Pruning" (Qu and Wang (2016) <arXiv:1605.07870>) is implemented. The idea is to conduct a linear decomposition of a signal using a few atoms of a learned and usually over-completed dictionary instead of a pre-defined basis. A proper size of the to-be-learned dictionary is determining at the same time during the procedure. Application includes image denoising and image inpainting.

Depends R (>= 3.2.3)

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, Matrix, fields, methods

RoxygenNote 6.0.1

NeedsCompilation yes

Author Simeng Qu [aut, cre],
Xiao Wang [aut]

Repository CRAN

Date/Publication 2017-10-06 11:51:17 UTC

R topics documented:

AddHoles 2
AddNoise 3
denoise 3
denoiseImage 4
gscad.denoise 5
gscad.DL 6

ImageReCon 8
ImageSplit 9
inpaint 9
inpaintImage 10
lena 11
lena_crop 11
NormMax 12
ODCT 12
PlotDic 13
UpDic 14

Index **15**

`AddHoles` *Corrupt a clean image.*

Description

Corrupt a clean image with randomly selected locations. The corrupted pixcels are set to 0.

Usage

```
AddHoles(I, pc)
```

Arguments

`I`	Clean image.
`pc`	Percetage of pixels to corrupt

Value

A corrupted image `corruptedImage` and `maskImage` to record the corrupted locations.

Examples

```
I=lena_crop
I_noise=AddNoise(I,30)
```

AddNoise *Add noise to a clean image.*

Description

Add white noise from N(0,sigma^2) to each pixel of clean image I

Usage

```
AddNoise(I, sigma)
```

Arguments

I	Clean image.
sigma	Noise level.

Value

A noisy image of the same size of I.

Examples

```
sigma=25; I=lena_crop
I_noise=AddNoise(I,sigma)
```

denoise *Use a given dictionary D to denoise image.*

Description

Denoise Y give D in Y=DA.

Usage

```
denoise(Y, D, sigma)
```

Arguments

Y	Each column of Y is a vectorized image patch to be denoised.
D	D is the dictionary used in Y=DA to denoise.
sigma	Noise level.

Details

See https://arxiv.org/abs/1605.07870

Value

The denoised matrix Y.

Examples

```
I = lena_crop #use a smaller image as an example
## add noise
sigma=20;
I_noise=AddNoise(I,sigma)
## spliting image into patches
m=64;
Y_nc = ImageSplit(I_noise,sqrt(m),sqrt(m));
mu=colMeans(Y_nc)
Y=Y_nc-rep(mu,each=nrow(Y_nc))
## use ODCT dictionary
D0=ODCT(64,100)
## denoise
## Not run:
Y_denoise=denoise(Y,D0,sigma)

## End(Not run)
```

---

denoiseImage *Use a given dictionary D to denoise image.*

Description

The noisy image is split into sqrt(m) by sqrt(m) patches. Each patch is vectorized into a column of matrix Y. Using the given D, the sparse coding A_hat in Y=DA is obtained. Then Y_denoise=DA_hat. The final denoised image is reconstruncted on the denoised patches.

Usage

```
denoiseImage(I_noise, D, sigma, stepsize = 1)
```

Arguments

I_noise	The image to be denoised. In form of matrix.
D	D is the dictionary used in Y=DA to denoise.
sigma	Noise level.
stepsize	(optional) The stepsize when splicting the image. Default is 1

Details

See https://arxiv.org/abs/1605.07870

Value

The denoised image in for of a matrix.

Examples

```
I = lena_crop #use a smaller image as an example
## add noise
sigma=20;
I_noise=AddNoise(I,sigma)
## use ODCT dictionary
D0=ODCT(64,100)
## denoise
## Not run:
I_clean=denoiseImage(I_noise,D0,sigma)

## End(Not run)
```

---

gscad.denoise *Use GSCAD to denoise image*

Description

Use the GSCAD method under dictionary learning framework to learn a proper sized dictionary and denoise image. The noisy image is split into m by m patches and a dictionary with each atom of size m is learned. The final denoised image is reconstructed on the denoised patches.

Usage

```
gscad.denoise(I_noise, sigma, D0 = NULL, m, p0, c = 3.7, lambda = 0.05,
  maxrun = 20, maxrun_ADMM = 20, err_bnd = 1e-04, err_bnd2 = 1e-04,
  rho = 16, cor_bnd = 1, L = NULL)
```

Arguments

I_noise	The image to be denoised. In form of matrix.
sigma	Noise level.
D0	Initial dictionary. If D0 specified, m and p0 are not needed, otherwise D0 is evaluated as overcompleted DCT basis using function ODCT(m,p0). Either D0 or (m,p0) needs to be specified.
m	The size of the small patches to be split.
p0	Initial size of the dictionary.
c, lambda	Parameters for GSCAD.
maxrun	(optional) Maximun number of outer iterations to run. Default is 20.
maxrun_ADMM	(optional) Maximun number of iterations to run for updating dictionary using ADMM. Default is 20.

err_bnd	(optional) Stopping criterion for iterations. Default is 1e-4.
err_bnd2	(optional) Stopping criterion for updating dictionary
rho	(optional) Parameter for ADMM. Default is 16.
cor_bnd	(optional) When normalize dictionary, checking if the correlation of any two atoms are above the cor_bnd, one of the atom is removed. Default is 1.
L	(optional) This parameter controls the maximum number of non-zero elements in each column of sparsecolding A.Default is m.

Details

See https://arxiv.org/abs/1605.07870

Value

The learned dictionary `dictionary`, its size p and the denoised image `fitted image`.

Examples

```
I = lena_crop #use a smaller image as an example
## add noise
sigma=20;
I_noise=AddNoise(I,sigma)
## denoising using GSCAD
## Not run:
out=gscad.denoise(I_noise,sigma,m=64,p0=100)

## End(Not run)
```

`gscad.DL` *Learn dictionary under GSCAD regularization*

Description

This function learns the dictionary D under the framework of matrix factorization $Y = DA$. Y is a given m by n matrix of n samples. D is an m by p matrix, where p is unkonw as well, and A ia a p by n matrix. Both D and A are unkonwn. GSCAD regularization is applied to D and lasso regularization is applied to A.

Usage

```
gscad.DL(Y, D0 = NULL, p0, sigma = 0, c = 3.7, lambda = 0.05,
  maxrun = 20, maxrun_ADMM = 20, err_bnd = 1e-06, err_bnd2 = 1e-04,
  rho = 16, cor_bnd = 1, L = 30, LassoMode = 1, LassoLambda = NULL)

gscad.DLmask(Y, Mask, D0 = NULL, p0, sigma = 0, c = 3.7, lambda = 0.05,
  maxrun = 20, maxrun_ADMM = 20, err_bnd = 1e-06, err_bnd2 = 1e-06,
  rho = 16, LassoMode = 1, LassoLambda = NULL)
```

Arguments

Y	An m by n matrix in Y=DA. Each column of Y is a sample of size m (usually a vectorized sqrt(m) by sqrt(m) patches).
D0	Initial dictionary. If D0 specified, p0 is not needed, otherwise D0 is evaluated as overcompleted DCT basis using function ODCT(m,p0). Either D0 or p0 needs to be specified.
p0	Initial size of the dictionary.
sigma	Noise level.
c, lambda	Parameters for GSCAD.
maxrun	(optional) Maximun number of outer iterations to run. Default is 20.
maxrun_ADMM	(optional) Maximun number of iterations to run for updating dictionary using ADMM. Default is 20.
err_bnd	(optional) Stopping criterion for iterations. Default is 1e-6.
err_bnd2	(optional) Stopping criterion for updating dictionary UpDic. Default is 1e-4.
rho	(optional) Parameter for ADMM. Default is 16.
cor_bnd	(optional) When normalize dictionary, checking if the correlation of any two atoms are above the cor_bnd, one of the atom is removed. Default is 1.
L	(optional) This parameter controls the maximum number of non-zero elements in each column of sparsecolding A.
LassoMode	(optional) At the sparse coding stage, the optimization can be done in three modes L1COEFFS (0), L2ERROR (1), PENALTY(2). Default is 1.
LassoLambda	(optional) Tuning parameter for Lasso
Mask	0,1 matrix of the same size as Y to indicate the location of corrupted pixels.

Details

See https://arxiv.org/abs/1605.07870

Value

The learned dictionary dictionary

Functions

- gscad.DLmask: Adding Mask 0,1 matrix of the same size as Y to indicate the location of corrupted pixel

Examples

```
I = lena_crop #use a smaller image as an example
## add noise
sigma=20;m=64
I_noise=AddNoise(I,sigma)
## spliting image into patches
Y_nc = ImageSplit(I_noise,sqrt(m),sqrt(m));
```

```
mu=colMeans(Y_nc)
Y=Y_nc-rep(mu,each=nrow(Y_nc))
## learning dictionary
## Not run:
dictionary=gscad.DL(Y,p0=256,sigma=sigma)

## End(Not run)
I=lena_crop
## corrupt 30% of the image
out_corrupt=AddHoles(I,0.3)
I_corrupt=out_corrupt$corruptedImage
I_mask=out_corrupt$maskImage
## split image
m=64
Y_nc = ImageSplit(I_corrupt,sqrt(m),sqrt(m));
M = ImageSplit(I_mask,sqrt(m),sqrt(m));
mu=colSums(Y_nc*M)/colSums(M)
Y=Y_nc-M*rep(mu,each=nrow(Y_nc))
mask = matrix(as.logical(M),ncol=ncol(M))
## learn dictionary for inpainting, this function is slow
## Not run:
dic=gscad.DLmask(Y, mask,  p0=100, sigma=1)

## End(Not run)
```

`ImageReCon` *Combine patches of image into full image*

Description

Suppose an image matrix A of size M by N is splited into (M-m+1)(N-n+1) overlapped small patches of size m by n. Given a matrix B of size (mn) by (M-m+1)(N-n+1) , where each column of B is a vectorized small patch, this function try to reconstruct the full image A.

Usage

```
ImageReCon(B, mm, nn, m, n, stepsize = 1)
```

Arguments

`B`	An (mn) by (M-m+1)(N-n+1) matrix.
`mm, nn`	Full image of size (mm by nn).
`m, n`	Size of the small patches (m by n).
`stepsize`	(optional) The stepsize when splicting the image. Default is 1

Value

A mm by nn matrix

ImageSplit *Splict image into small patches*

Description

Splict an image matrix A of size M by N into overlapped small patches of size m by n. Total of (M-m+1)(N-n+1) patches.

Usage

```
ImageSplit(A, m, n, stepsize = 1)
```

Arguments

A	An M by N matrix.
m, n	Size of the small patches (m by n).
stepsize	(optional) The stepsize when splicting the image. Default is 1

Value

A (mn) by (M-m+1)(N-n+1) matrix, each column of which is a vectorized small patch.

inpaint *Use a given dictionary D to inpaint image*

Description

Given D, obtain the sparse coding A_hat in Y=DA. Then Y_inpaint=DA_hat. See https://arxiv.org/abs/1605.07870

Usage

```
inpaint(Y, Mask, D, L = 30, eps = NULL, sigma = 0)
```

Arguments

Y	Each column of Y is a vectorized image patch to be denoised.
Mask	0,1 matrix of the same size as Y to indicate the location of corrupted pixels.
D	D is the dictionary used in Y=DA to inpaint.
L	(optional) This parameter controls the maximum number of non-zero elements in each column of sparsecolding A.
eps	(optional) A lasso tuning paramter
sigma	Noise level.

Value

The inpainted matrix Y.

Examples

```
I=lena_crop
## corrupt 30% of the image
out_corrupt=AddHoles(I,0.3)
I_corrupt=out_corrupt$corruptedImage
I_mask=out_corrupt$maskImage
## split image
m=64
Y_nc = ImageSplit(I_corrupt,sqrt(m),sqrt(m));
M = ImageSplit(I_mask,sqrt(m),sqrt(m));
mu=colSums(Y_nc*M)/colSums(M)
Y=Y_nc-M*rep(mu,each=nrow(Y_nc))
mask = matrix(as.logical(M),ncol=ncol(M))
## use ODCT dictionary
D0=ODCT(64,100)
## inpaint
Y_inpaint=inpaint(Y,mask,D0)
```

---

inpaintImage *Use a given dictionary D to inpaint image*

Description

The corrupted image is split into sqrt(m) by sqrt(m) patches. Each patch is vectorized into a column of matrix Y. Mask matrix M is constructed to indicate the location of missing pixels. Using the given D, the sparse coding A_hat in Y=DA is obtained. Then Y_inpaint=DA_hat. Then the final inpainted image is reconstruncted. See https://arxiv.org/abs/1605.07870

Usage

```
inpaintImage(I_corrupt, I_mask, D, L = 30, eps = NULL, sigma = 0,
  stepsize = 1)
```

Arguments

I_corrupt	The image to be inpainted. In form of matrix.
I_mask	0,1 matrix of the same size as I_corrupt to indicate the location of corrupted pixels.
D	D is the dictionary used in Y=DA to inpaint.
L	(optional) This parameter controls the maximum number of non-zero elements in each column of sparsecolding A.
eps	(optional) A lasso tuning paramter
sigma	Noise level.
stepsize	(optional) The stepsize when splicting the image. Default is 1

Value

The denoised image in for of a matrix.

Examples

```
I=lena_crop
## corrupt 30% of the image
out_corrupt=AddHoles(I,0.3)
I_corrupt=out_corrupt$corruptedImage
I_mask=out_corrupt$maskImage
## use ODCT dictionary
D0=ODCT(64,100)
## inpaint
out=inpaintImage(I_corrupt,I_mask,D0)
```

---

lena *Image data Lena*

Description

Lena is a benchmark testing image for image processing. The 512 by 512 matrix is created by reading in the png formed image.

Usage

```
lena
```

Format

A 512 by 512 matrix

---

lena_crop *Image data Lena_crop*

Description

Cropped from Image Lena.

Usage

```
lena_crop
```

Format

A 100 by 100 matrix

NormMax *Normalize dictionary atoms using l_∞ norm*

Description

Remove zero columns of D. Then let $D = (D_1, ..., D_p)$. The normalization is done by

$$D_ik = D_ik/max_kD_ik, fori = 1, ..., p.$$

For any D_i, and D_j, if Cor(D_i,D_j)> cor_bnd, one of D_i and D_j is eliminated.

Usage

```
NormMax(D, cor_bnd = 0.95)
```

Arguments

D	The dictionary to be normalized.
cor_bnd	An upper bound of the correlation between atoms.

Value

The nomalized dictionary.

Examples

```
D=matrix(rnorm(50),nrow=5)
D=NormMax(D)
```

ODCT *Overcomplete Discrete Cosine Transform (ODCT) Generating Function.*

Description

ODCT returns a matrix which is used in dictionary learning as an initial dictionary matrix.

Usage

```
ODCT(m, p)
```

Arguments

m	Number of grid in each base or the size of each atom in the dictionary.
p	Number of bases or the size of the dictionary.

Value

A m by p matrix. Each column is a DCT base.

Examples

```
dic=ODCT(64,100)
# generating an initial dictionary of size 100 with each atom/image patch of size 64.
```

PlotDic *Plot atoms of a given dictionary*

Description

Given a learned dictionary D (m by p) matrix, this function plots all p atoms of D corresponding to p columns. Each column is mapped to a m1 by m2 small pactch.

Usage

```
PlotDic(D, n1 = NULL, n2 = NULL, m1 = NULL, m2 = NULL, title = NULL,
  color = F, background = "black")
```

Arguments

D	Dictionary of size m by p
n1, n2	(Optional) Number of patches to be displayed in each column and each row. If specified, both n1 and n2 have to be specified.
m1, m2	(Optional) Size of the small patch to be mapped. Default is m1=m2=sqrt(m). If specified, both n1 and n2 have to be specified.
title	Title of the plot.
color	(Optional) If TRUE, the plot is generated using tim.colors() option. If FALSE, the plot is generated using gray scale. Default is FALSE.
background	(Optional) Backgroup color. Default is black.

Value

Plot will be generated.

Examples

```
D=matrix(runif(64*100),nrow=64)
# PlotDic(D)
```

UpDic *Update dictionary in GSCAD*

Description

When the sparce coding A={alpha_1,...,alpha_n} is given, update dictionary by solving problem (6) in https://arxiv.org/abs/1605.07870 using ADMM. #'

Usage

```
UpDic(Y, alpha, rho = 1, lambda = 0.01, c = 3.7, maxrun_ADMM = 100,
  err_bnd = 1e-04)

UpDicMask(Y, alpha, mask, rho = 1, lambda = 0.01, c = 3.7,
  maxrun_ADMM = 100, err_bnd = 1e-04)
```

Arguments

Y	The image to be denoised. Inform of matrix.
alpha	Sparse coding.
rho	Parameter for the augmented Lagrangian function.
c, lambda	parameters for GSCAD
maxrun_ADMM	Maximun number of iterations run for ADMM
err_bnd	Stopping criterion for iterations.
mask	0,1 matrix of the same size as Y to indicate the location of corrupted pixels.

Value

The updated sparse dictionary .

Functions

- UpDicMask: Adding mask 0,1 matrix of the same size as Y to indicate the location of corrupted pixel

Index

∗Topic **datasets**
lena, 11
lena_crop, 11

AddHoles, 2
AddNoise, 3

denoise, 3
denoiseImage, 4

gscad.denoise, 5
gscad.DL, 6
gscad.DLmask *(*gscad.DL*)*, 6

ImageReCon, 8
ImageSplit, 9
inpaint, 9
inpaintImage, 10

lena, 11
lena_crop, 11

NormMax, 12

ODCT, 12

PlotDic, 13

UpDic, 14
UpDicMask *(*UpDic*)*, 14