Recognition Toolbox

Created by Eva Janousova

Recognition Toolbox provides algorithms for reduction and classification of two-dimensional (2-D) or three-dimensional (3-D) medical image data acquired with diverse medical imaging techniques. The algorithms were implemented as functions in MATLAB® environment. The list of the functions including purpose of the function, a syntax example and superordinate functions is given in Tab. 1. The input and output data and variables of the functions are listed in Tab. 2. A hierarchy of the functions included in the Recognition Toolbox is depicted in Fig. 1.

Steps of data image analysis using the Recognition Toolbox are visualized in Fig. 2. The first step comprises reading the 2-D or 3-D image data of two groups of subjects (e.g. patients and healthy controls) in NIFTI[1] (Neuroimaging Informatics Technology Initiative) format and their preprocessing. It is possible to extract background voxels and non-brain tissue voxels from the images using a mask defined by user during preprocessing. Afterwards, the 2-D or 3-D images are transformed into 1-D vectors and stored as rows in a matrix Z. The second step is reduction of the data matrix Z using principal component analysis based on the covariance matrix of persons (pPCA). The third step includes classification using selected classifiers. Three classifiers, namely the average linkage method, the centroid method and the modified maximum uncertainty linear discriminant analysis (mMLDA), are implemented in the Recognition Toolbox. Results of the classifiers are votes (group identifiers) which can be placed into one matrix. The matrix is an input into voting algorithms. Images are classified into the group of patients or healthy controls according to the majority vote of the classifiers. The Recognition Toolbox also enables performing leave-one-out cross-validation (LOOCV). It is possible to choose if the LOOCV is executed only during the classification step or even during the reduction step.

[1] http://nifti.nimh.nih.gov/nifti-1

Tab. 1: A list of functions included in the Recognition Toolbox.

Function name	**Purpose**	**Syntax example**	**Superordinate functions**
averagelink.m	Classify image data into two groups using the average linkage.	[vote] = averagelink(Z,gi,n,loo)	recog
centroid.m	Classify image data into two groups using the centroid method.	[vote] = centroid(Z,gi,n,loo)	recog
mMLDA.m	Classify image data into two groups using the mMLDA.	[vote] = mMLDA(Z,gi,n,loo)	recog
ppca_eig.m	Compute projection matrix of pPCA.	[V,pvar] = ppca_eig(Z,n)	ppca_reduc recog
ppca_reduc.m	Reduce image data using pPCA.	[Zred,pvar,Zmean,V]= ppca_reduc(Z,n)	recog
recog.m	Do recognition of image data. The main function in the Recognition Toolbox. It interconnects functions about data preprocessing, reduction and classification.	[votes,gis]=recog (dpath,namsel,gi,mask, n,loo, clas)	
recog_effic.m	Compute efficiency of classifiers.	[effic] = recog_effic(votes,gis)	
recog_preproc.m	Read and preprocess image data.	[Z]=recog_preproc (dpath,namsel,mask)	recog
recog_voting.m	Compute efficiency of recognition using voting of various combinations of classifiers.	[effvot]=recog_voting (votes,no,eftype,gis)	
thomaz_meang.m	Compute centroids of two groups during computation of mMLDA.	[Mg]=thomaz_meang (X,ns,nt)	thomaz_MLDA mMLDA recog
thomaz_mecs.m	Maximize entropy of each group during computation of mMLDA.	[Se,tt] = thomaz_mecs(Sp,Sg)	thomaz_MLDA mMLDA recog
thomaz_MLDA.m	Compute classification scores using projection matrix of MLDA.	[Xred,L] = thomaz_MLDA (X,ns,nt,nr,i_tren)	mMLDA recog
thomaz_repeatc.m	Build a matrix by repeating each column of original matrix nt-times. It is used during computation of mMLDA.	[H] = thomaz_repeatc(X,nt)	thomaz_MLDA mMLDA recog
vote_effic.m	Compute all possible combinations of selected number of classifiers and calculate classification efficiency of voting of the combinations of classifiers.	[eff,call]=vote_effic (votes,vot_gt,Nc,i_pat)	recog_voting

Tab. 2: A list of input and output data and variables of the functions in the Recognition Toolbox.

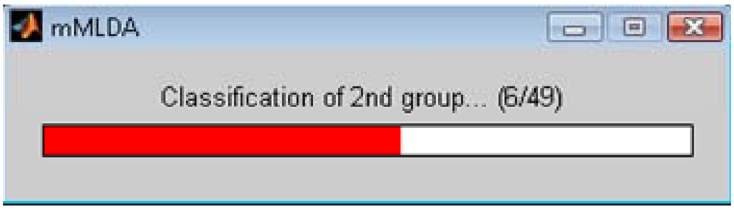
Variable	**Description**	**Used as input variable by**	**Used as output variable by**
call	all possible combinations of classifiers		vote_effic
clas	vector of classifiers which are used (1..mMLDA, 2..centroid method, 3..average linkage; for example: [1,2,3] - all classifiers are used; [1,3] - mMLDA and average linkage are used for classification)	recog	
dpath	directory with the image data in NIFTI format	recog, recog_preproc	
eff	efficiency of voting combinations		vote_effic
effic	efficiency of classifiers		recog_effic
effvot	efficiency of voting combinations and combinations of classifiers		recog_voting
eftype	type of efficiency which is used for sorting of results of voting (1..accuracy, 2..sensitivity, 3..specificity, 4..multiple of accuracy and sensitivity, 5..multiple of sensitivity and specificity)	recog_voting	
gi	group identifiers of images	averagelink, centroid, mMLDA, recog	
gis	sorted group identifiers of selected images	recog_effic, recog_voting	recog
H	matrix built by repeating each column i of matrix X nt(i) times		thomaz_repeatc
i_pat	indexes of patient images	vote_effic	
i_tren	indexes of all training images	thomaz_MLDA	
L	eigenvectors (sorted) of matrix X; each column represents an eigenvector		thomaz_MLDA
loo	binary variable: 0..leave-one-out cross-validation (LOOCV) is used during data reduction and classification; 1..LOOCV is used only during classification	averagelink, centroid, mMLDA, recog	
mask	mask (a binary image matrix with the same size as MRI brain data, for example see gmbrainmask.mat in example_data; if you do not want to mask your images, set mask=[])	recog, recog_preproc	
Mg	matrix of group sample means		thomaz_meang
namsel	names of selected images which are input into recognition	recog, recog_preproc	
Nc	number of classifiers which will vote	vote_effic	

Variable	Description	Used as input variable by	Used as output variable by
n	number of eigenvectors which you want to discard (a positive number means discarding N eigenvectors with smallest eigenvalues, a negative number means discarding N eigenvectors with largest eigenvalues and zero means discarding no eigenvectors); absolute value of n must be less than a number of images minus one	averagelink, centroid, mMLDA, ppca_eig, ppca_reduc, recog	
no	number of combinations with the highest efficiency which you want to display	recog_voting	
nr	reduction of dimension (nr <= ns-1)	thomaz_MLDA	
ns	number of groups	thomaz_MLDA, thomaz_meang	
nt	number of training images in each group	thomaz_MLDA, thomaz_meang, thomaz_repeatc	
pvar	explained variability by pPCA		ppca_eig, ppca_reduc
Se	The Smix matrix based on the maximum entropy		thomaz_mecs
Sg	sample group covariance matrix	thomaz_mecs	
Sp	spooled (common) covariance matrix	thomaz_mecs	
tt	CPU total time in seconds of MECS calculation		thomaz_mecs
V	pPCA projection matrix which consists of eigenvectors		ppca_eig, ppca_reduc
vot_gt	ground truth of votes	vote_effic	
vote	vector of votes (group identifiers) which are results of the centroid method for each person		averagelink, centroid, mMLDA
votes	matrix of votes (group identifiers) which are results of classifiers for each person (rows are classifiers, columns are persons)	recog_effic, recog_voting, vote_effic	recog
X	matrix containing the training. set, whose each line points to a sample data	thomaz_MLDA, thomaz_meang, thomaz_repeatc	
Xred	reduced matrix X		thomaz_MLDA
Z	2-D image data matrix where rows are persons and columns are voxels	averagelink, centroid, mMLDA, ppca_eig, ppca_reduc	recog_preproc
Zmean	mean person image (mean over all voxels)		ppca_reduc
Zred	reduced 2-D image data matrix by pPCA		ppca_reduc

```
↳ recog
    ↳ recog_preproc
    ↳ ppca_reduc
         ↳ ppca_eig
    ↳ mMLDA
         ↳ thomaz_MLDA
               ↳ thomaz_repeatc
               ↳ thomaz_meang
               ↳ thomaz_mecs
    ↳ centroid
    ↳ averagelink
↳ recog_effic
↳ recog_voting
    ↳ vote_effic
```

Fig. 1: Hierarchy of the functions in the Recognition Toolbox.

The functions involved in the Recognition Toolbox are described in full details in following chapters. The functions are in alphabetical order. The *averagelink*, *centroid* and *mMLDA* functions display a waitbar (Fig. 3) for illustration how many images have been classified up to now.



Classification of 2nd group... (6/49)

Fig. 3: Waitbar displayed by classification functions. It shows a number of successfully classified images.

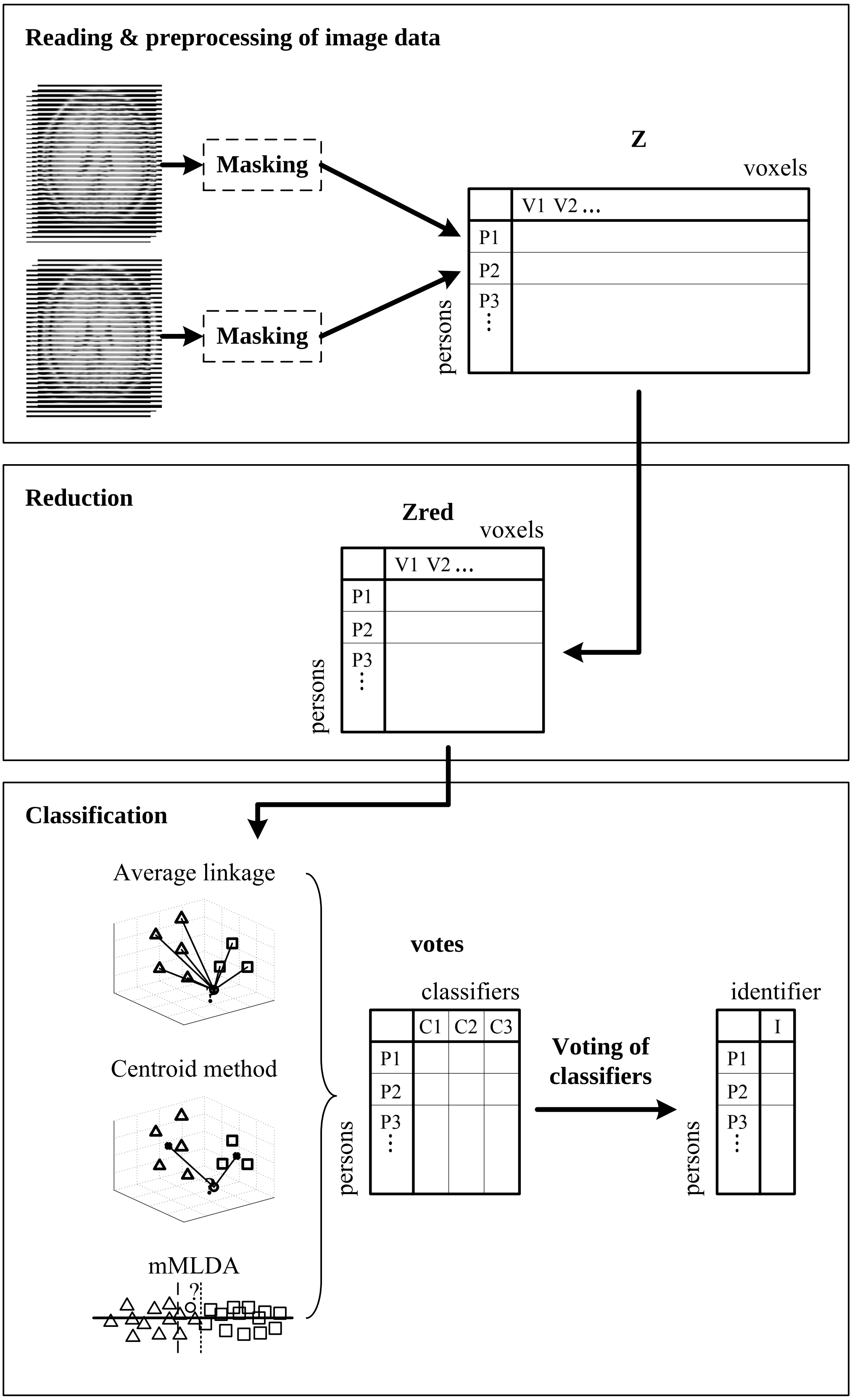


Fig. 2: Block scheme of recognition of 3-D brain images using three classifiers. 3-D image data which can be masked are converted into 1-D row vectors of an image data matrix Z. The image data matrix is reduced using pPCA and classified into a group of patients or into a group of healthy controls by mMLDA, the centroid method and/or the average linkage. The votes (group identifiers) are an input into the voting algorithms with the result of a majority group identifier for each input image.

1. AVERAGELINK

Purpose

Classify image data into two groups using the average linkage.

Syntax

[vote] = averagelink(Z,gi,n,loo)

Input data and variables:

Z	2-D image data matrix where rows are persons and columns are voxels
gi	group identifiers of images
n	number of eigenvectors which you want to discard (a possitive number means discarding N eigenvectors with smallest eigenvalues, a negative number means discarding N eigenvectors with largest eigenvalues and zero means discarding no eigenvectors); absolute value of n must be less than a number of images minus one
loo	binary variable: 0..leave-one-out cross-validation (LOOCV) is used during data reduction and classification; 1..LOOCV is used only during classification

Output data and variables:

vote	vector of votes (group identifiers) which are results of the centroid method for each person

Description

The *averagelink* function consists of the average linkage clustering method (Fig. 4) and validation using LOOCV. There is an opportunity to choose if the LOOCV is used during data reduction and classification or only during classification. LOOCV means that every image (a row vector of the matrix *Z*) is chosen as a testing image stepwise and the *averagelink* function assigns a group identifier to the image. The ground truth group identifiers are saved in *gi*. The *averagelink* function shows a waitbar (see an example waitbar in Fig. 3).

Algorithm

If *loo=0* (it means LOOCV is used during data reduction and classification):

- Repeat for each image:
  - Set the image as the testing image; all remaining images are the training images.
  - Reduce the training images using the *ppca_reduc* function; the parameter *n* enables to choose how many eigenvectors are discarded during creation of a projection matrix of pPCA.
  - Use the results of *ppca_reduc* for subtraction of a mean training image from the testing image and for reduction of the testing image.
  - Compute Euclidean distances of the reduced testing image from all reduced training images.

    - Calculate a mean Euclidean distance of the testing image from training images of the first group and a mean Euclidean distance of the testing image from training images of the second group.
    - Assign the testing image into the group with the shortest mean Euclidean distance.

If *loo=1* (it means LOOCV is used only during classification):

- Reduce all images using the *ppca_reduc* function; the parameter *n* enables to choose how many eigenvectors are discarded during creation of a projection matrix of pPCA.
- Repeat for each reduced image:
    - Set the image as the testing image; all remaining images are the training images.
    - Compute Euclidean distances of the testing image from all training images.
    - Calculate a mean Euclidean distance of the testing image from training images of the first group and a mean Euclidean distance of the testing image from training images of the second group.
    - Assign the testing image into the group with the shortest mean Euclidean distance.

Notes

The mathematical background of the average linkage method is described in Appendix B.

References

For more information about the average linkage method, see (Legendre & Legendre, 1998).

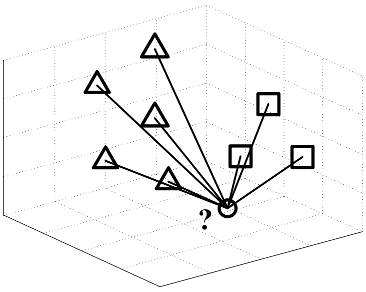


Fig. 4: Illustration of the average linkage classification method. Squares stand for images of patients and triangles stand for images of normal controls. A testing image (a circle) is assigned into the group with the shortest mean Euclidean distance. Axes are image voxels (for easier imagination only three axes are displayed here).

2. CENTROID

Purpose

Classify image data into two groups using the centroid method.

Syntax

[vote] = centroid(Z,gi,n,loo)

Input data and variables:

Z	2-D image data matrix where rows are persons and columns are voxels
gi	group identifiers of images
n	number of eigenvectors which you want to discard (a possitive number means discarding N eigenvectors with smallest eigenvalues, a negative number means discarding N eigenvectors with largest eigenvalues and zero means discarding no eigenvectors); absolute value of n must be less than a number of images minus one
loo	binary variable: 0..leave-one-out cross-validation (LOOCV) is used during data reduction and classification; 1..LOOCV is used only during classification

Output data and variables:

vote	vector of votes (group identifiers) which are results of the centroid method for each person

Description

The *centroid* function consists of the centroid clustering method (Fig. 5) and validation using LOOCV. As in the *averagelink* function, there is also an opportunity to choose if the LOOCV is used during data reduction and classification or only during classification. LOOCV means that every image (a row vector of the matrix *Z*) is chosen as a testing image stepwise and the *centroid* function assigns a group identifier to the image. The ground truth group identifiers are saved in *gi*. The *centroid* function displays a waitbar (see an example waitbar in Fig. 3).

Algorithm

If *loo=0* (it means LOOCV is used during data reduction and classification):

- Repeat for each image:
  - ○ Set the image as the testing image; all remaining images are the training images.
  - ○ Reduce the training images using the *ppca_reduc* function; the parameter *n* enables to choose how many eigenvectors are discarded during creation of a projection matrix of pPCA.
  - ○ Use the results of *ppca_reduc* for subtraction of a mean training image from the testing image and for reduction of the testing image.
  - ○ Compute a centroid (a mean image) of the first group and a centroid of the second group.

        - Calculate Euclidean distances of the testing image from the both centroids.
        - Assign the testing image into the group represented by the closer centroid.

If *loo=1* (it means LOOCV is used only during classification):

- Reduce all images using the *ppca_reduc* function; the parameter *n* enables to choose how many eigenvectors are discarded during creation of a projection matrix of pPCA.
- Repeat for each reduced image:
    - Set the image as the testing image; all remaining images are the training images.
    - Compute a centroid (a mean image) of the first group and a centroid of the second group.
    - Calculate Euclidean distances of the testing image from the both centroids.
    - Assign the testing image into the group represented by the closer centroid.

Notes

The mathematical background of the centroid method is described in Appendix C.

References

For more information about the centroid method, see (Legendre & Legendre, 1998).

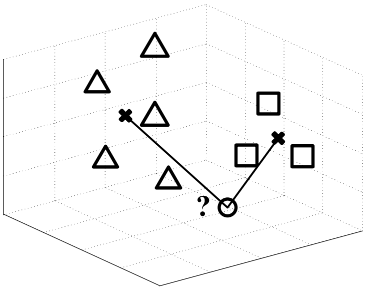


Fig. 5: Illustration of the centroid method. Squares stand for images of patients and triangles stand for images of normal controls. Crosses represent centroids of both groups of persons. A testing image (a circle) is assigned into the group represented by the closer centroid. Axes are image voxels (for easier imagination only three axes are displayed here).

3. MMLDA

Purpose

Classify image data into two groups using the mMLDA.

Syntax

[vote] = mMLDA(Z,gi,n,loo)

Input data and variables:

Z	2-D image data matrix where rows are persons and columns are voxels
gi	group identifiers of images
n	number of eigenvectors which you want to discard (a possitive number means discarding N eigenvectors with smallest eigenvalues, a negative number means discarding N eigenvectors with largest eigenvalues and zero means discarding no eigenvectors); absolute value of n must be less than a number of images minus one
loo	binary variable: 0..leave-one-out cross-validation (LOOCV) is used during data reduction and classification; 1..LOOCV is used only during classification

Output data and variables:

vote	vector of votes (group identifiers) which are results of the centroid method for each person

Description

The *mMLDA* function consists of the modified maximum uncertainty linear discriminant analysis method (Fig. 6) and validation using LOOCV. As in the *averagelink* function and the *centroid* function, there is also an opportunity to choose if the LOOCV is used during data reduction and classification or only during classification. LOOCV means that every image (a row vector of the matrix *Z*) is chosen as a testing image stepwise and the *mMLDA* function assigns a group identifier to the image. The ground truth group identifiers are saved in *gi*. The *mMLDA* function shows a waitbar (see an example waitbar in Fig. 3).

Algorithm

If *loo=0* (it means LOOCV is used during data reduction and classification):

- Repeat for each image:
    - Set the image as the testing image; all remaining images are the training images.
    - Reduce the training images using the *ppca_reduc* function; the parameter *n* enables to choose how many eigenvectors are discarded during creation of a projection matrix of pPCA.
    - Use the results of *ppca_reduc* for subtraction of a mean training image from the testing image and for reduction of the testing image.
    - Compute projection matrix of MLDA and classification scores of training images using *thomaz_MLDA*.

    - Calculate mean classification scores of both groups and a classification boundary using the weighted mean.
    - Compute classification score of the testing image with the use of the projection matrix of MLDA.
    - Classify the testing image into one of the groups depending on whether its classification score falls above or below the boundary.

If *loo=1* (it means LOOCV is used only during classification):

- Reduce all images using the *ppca_reduc* function; the parameter *n* enables to choose how many eigenvectors are discarded during creation of a projection matrix of pPCA.
- Repeat for each reduced image:
    - Set the image as the testing image; all remaining images are the training images.
    - Compute projection matrix of MLDA and classification scores of training images using *thomaz_MLDA*.
    - Calculate mean classification scores of both groups and a classification boundary using the weighted mean.
    - Compute classification score of the testing image with the use of the projection matrix of MLDA.
    - Classify the testing image into one of the groups depending on whether its classification score falls above or below the boundary.

Notes

The mathematical background of mMLDA is described in Appendix D.

References

To learn more about the MLDA, see (Thomaz et al., 2007a; Thomaz et al., 2007b; Fujita et al., 2008). For more information about the classification boundary based on the weighted mean, see (Culhane et al., 2002).

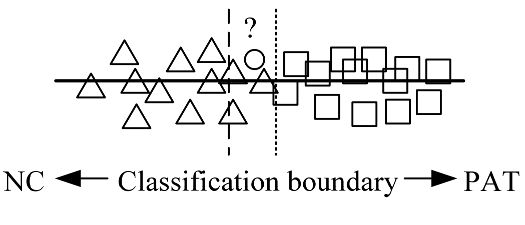


Fig. 6: Illustration of MLDA and mMLDA classification. Squares stand for classification scores of patient (PAT) images and triangles stand for classification scores of normal control (NC) images. The dotted line represents the classification boundary of MLDA and the dashed line represents the classification boundary of mMLDA. A new image (the circle) is classified as PAT by mMLDA and as NC by MLDA.

4. PPCA_EIG

Purpose

Compute projection matrix of pPCA.

Syntax

[V,pvar] = ppca_eig(Z,n)

Input data and variables:

Z	2-D image data matrix where rows are persons and columns are voxels
n	number of eigenvectors which you want to discard (a possitive number means discarding N eigenvectors with smallest eigenvalues, a negative number means discarding N eigenvectors with largest eigenvalues and zero means discarding no eigenvectors); absolute value of n must be less than a number of images minus one

Output data and variables:

V	pPCA projection matrix which consists of eigenvectors
pvar	explained variability by pPCA

Description

The *ppca_eig* function is a subordinate function of the *ppca_reduc* function. It is the core function of person PCA. A user can choose a number of eigenvectors *n* which are discarded during creation of a projection matrix *V* of pPCA. The parameter *n* influences the explained variability *pvar*.

Algorithm

- Compute the covariance matrix of persons and its eigenvectors and eigenvalues.
- Sort eigenvalues from largest to smallest, choose non-zero eigenvalues and sort and choose the corresponding eigenvectors.
- Discard *n* eigenvectors.
- Transform eigenvectors of the covariance matrix of persons into eigenvectors of the covariance matrix of voxels and save them as columns into the projection matrix *V*.

Notes

The mathematical background of pPCA is described in Appendix A.

References

To learn more about the computation of eigenvectors of the covariance matrix of voxels from eigenvectors of the covariance matrix of persons, see (Demirci et al., 2008; Fukunaga, 1990).

5. PPCA_REDUC

Purpose

Reduce image data using pPCA.

Syntax

[Zred,pvar,Zmean,V] = ppca_reduc(Z,n)

Input data and variables:

Z	2-D image data matrix where rows are persons and columns are voxels
n	number of eigenvectors which you want to discard (a possitive number means discarding N eigenvectors with smallest eigenvalues, a negative number means discarding N eigenvectors with largest eigenvalues and zero means discarding no eigenvectors); absolute value of n must be less than a number of images minus one

Output data and variables:

Zred	reduced 2-D image data matrix by pPCA
pvar	explained variability by pPCA
Zmean	mean person image (mean over all voxels)
V	pPCA projection matrix which consists of eigenvectors

Description

The *ppca_reduc* function serves for reduction of image data matrix *Z* using the projection matrix *V* of pPCA. The *ppca_reduc* function calls the *ppca_eig* function which allows for computation of the pPCA projection matrix. A number of eigenvectors *n* to be discarded is passed to the *ppca_eig* function. The results of the *ppca_reduc* function are the reduced image data matrix *Zred*, explained varialibity *pvar* and two variables important for reduction of a testing image. They are a mean person image *Zmean* which is subtracted from the testing image and the pPCA projection matrix *V* which is used for reduction of the testing image.

Algorithm

- Compute a mean person image *Zmean* and subtract it from all images of the data matrix *Z* (it means the row vector *Zmean* is subtracted from each row of the image data matrix *Z*).
- Call the function ppca_eig and reduce the centered matrix *Z* using the projection matrix *V*.

Notes

The mathematical background of pPCA is described in Appendix A.

References

To learn more about pPCA, see (Demirci et al., 2008; Fukunaga, 1990).

6. RECOG

Purpose

Do recognition of image data. It is the main function in the Recognition Toolbox. It interconnects functions about data preprocessing, reduction and classification.

Syntax

```
[votes,gis] = recog(dpath,namsel,gi,mask,n,loo,clas)
```

Input data and variables:

dpath directory with the image data in NIFTI format

namsel names of selected images which are input into recognition

gi group identifiers of images

mask mask (a binary image matrix with the same size as MRI brain data, for example see gmbrainmask.mat in example_data; if you do not want to mask your images, set mask=[])

n number of eigenvectors which you want to discard (a possitive number means discarding N eigenvectors with smallest eigenvalues, a negative number means discarding N eigenvectors with largest eigenvalues and zero means discarding no eigenvectors); absolute value of n must be less than a number of images minus one

loo binary variable: 0..leave-one-out cross-validation (LOOCV) is used during data reduction and classification; 1..LOOCV is used only during classification

clas vector of classifiers which are used (1..mMLDA, 2..centroid method, 3..average linkage; for example: [1,2,3] - all classifiers are used)

Output data and variables:

votes matrix of votes (group identifiers) which are results of classifiers for each person (rows are classifiers, columns are persons)

gis sorted group identifiers of selected images

Example

```
[vote]=recog('C:\Users\Janousova\MATLAB\pPCA_toolbox\example_data',
namsel,gi,[],-3,0,[1,3])
```

- original 3-D images are in the directory 'C:\Users\Janousova\MATLAB\pPCA_toolbox\example_data'
- names of selected images are given in *namsel*
- group identifiers of selected images are given in *gi*
- images will not be masked
- 3 eigenvectors which corresponds to the 3 largest eigenvalues will be discarded from the pPCA projection matrix
- LOOCV is used during data reduction and classification
- mMLDA and average linkage are used for classification

Description

The *recog* function is the main function in the Recognition Toolbox. It integrates functions for reading, preprocessing, reduction and classification of the image data. It calls the function *recog_preproc* for reading and preprocessing of selected 2-D or 3-D images in NIFTI format in directory *dpath*. The names of the selected images are given in *namsel* and group identifiers of these images are stored in *gi*. It is possible to extract background or non-brain voxels using *mask* defined by user during preprocessing. The *recog* function also calls the classification functions (*averagelink*, *centroid* and *mMLDA*) according to the variable *clas*. The result of classification of image data reduced by *ppca_reduc* is a matrix *votes*. During reduction, *n* eigenvectors can be discarded from computation of the pPCA projection matrix. It is enabled to reduce all images simultaneously and use LOOCV afterwards if *loo=1*; or perform LOOCV during classification and reduction if *loo=0*.

Algorithm

- Sort group identifiers of images and corresponding names of images in ascending order.
- Call the *recog_preproc* function for image data reading and preprocessing.
- Call the classification functions according to the vector *clas*.

Notes

The directory *dpath* must contain 2-D or 3-D images in NIFTI format. In the vector *gi*, a group identifier of patients must be smaller (for example is equal to 1) than a group identifier of healthy controls (for example is equal to 2). The reason is correct computation of sensitivity and specificity.

7. RECOG_EFFIC

Purpose

Compute efficiency of classifiers.

Syntax

[effic] = recog_effic(votes,gis)

Input data and variables:

votes matrix of votes (group identifiers) which are results of classifiers for each person (rows are classifiers, columns are persons)

gis sorted group identifiers of selected images

Output data and variables:

effic efficiency of classifiers

Description

The *recog_effic* function computes efficiency of recognition of 2-D or 3-D image data. Input variables are results of the *recog* function. An output variable effic is a matrix where rows are classifiers and columns are true positive results of recognition, false negative results, true negative results, false positive results, accuracy, sensitivity, specificity and precision.

Algorithm

- Create a vector with ground truth of votes using a vector *gis*.
- Compute efficiency of each classifier.

References

To learn more about the evaluation of efficiency of classifiers, see (Altman, 1999).

8. RECOG_PREPROC

Purpose

Read and preprocess image data.

Syntax

[Z] = recog_preproc(dpath,namsel,mask)

Input data and variables:

dpath	directory with the image data in NIFTI format
namsel	names of selected images which are input into recognition
mask	mask (a binary image matrix with the same size as MRI brain data, for example see gmbrainmask.mat in example_data; if you do not want to mask your images, set mask=[])

Output data and variables:

Z	2-D image data matrix where rows are persons and columns are voxels

Description

The *recog_preproc* function serves for reading and preprocessing of 2-D or 3-D image data in NIFTI format (Fig. 7). It is possible to extract background and non-brain voxels using a mask defined by user. The mask must have the same size as input images. The result of the *recog_preproc* function is the 2-D image data matrix *Z* which contains images transformed into 1-D row vectors. The *recog_preproc* function shows a waitbar (Fig. 8) for illustration how many images have been read up to now.

Algorithm

- Compute a number of columns of the matrix *Z* (it means a number of voxels in masked image if the variable *mask* is not empty, or a number of voxels in original image).
- Repeat for each image:
    - Read a 2-D or 3-D image in NIFTI format.
    - Transform the image into the 1-D vector.
    - If the variable *mask* is not empty, pick out voxels where *mask=1*.
    - Save the row image vector into the matrix *Z*.

Notes

The directory *dpath* must contain 2-D or 3-D images in NIFTI format.

References

For more information about the NIFTI format, see (http://nifti.nimh.nih.gov/nifti-1).

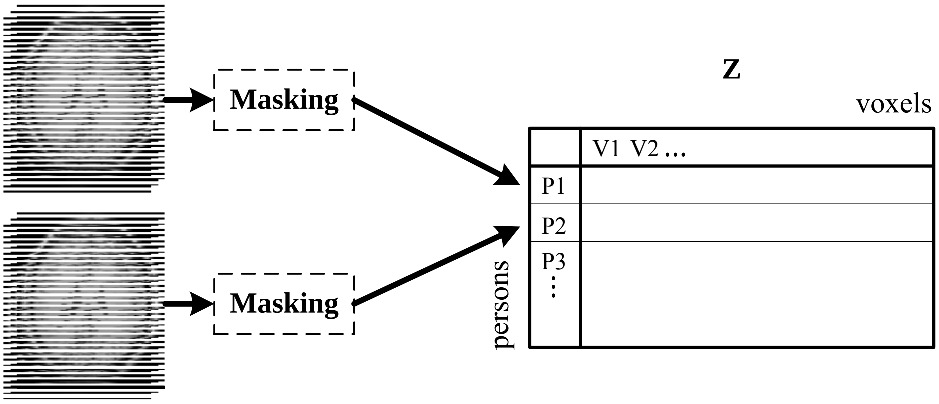
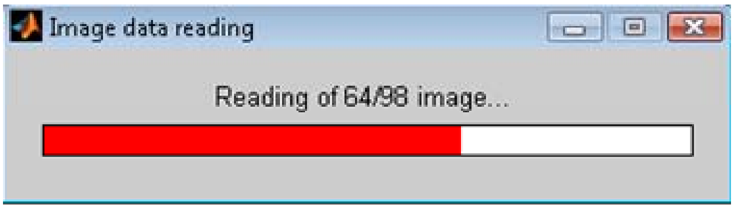


Fig. 7: Example reading and preprocessing of 3-D image data. During preprocessing, the read images are transformed into 1-D row vectors and are saved into the matrix **Z**. The masking step is optional.



Reading of 64/98 image...

Fig. 8: Waitbar of the *recog_preproc* function. It shows a number of successfully read images.

9. RECOG_VOTING

Purpose

Compute efficiency of recognition using voting of combinations of classifiers.

Syntax

[effvot] = recog_voting (votes,no,eftype,gis)

Input data and variables:

votes	matrix of votes (group identifiers) which are results of classifiers for each person (rows are classifiers, columns are persons)
no	number of combinations with the highest efficiency to be displayed
eftype	type of efficiency which is used for sorting of results of voting (1..accuracy, 2..sensitivity, 3..specificity, 4..multiple of accuracy and sensitivity, 5..multiple of sensitivity and specificity)
gis	sorted group identifiers of selected images

Output data and variables:

effvot	efficiency of voting combinations and combinations of classifiers

Description

The *recog_voting* function computes efficiency of recognition of 2-D or 3-D image datasets using voting of classifiers. Input variables *votes* and *gis* are results of the *recog* function. An output variable effvot is a matrix where rows are classifiers and columns are true positive, false negative, true negative and false positive results of recognition, accuracy, sensitivity, specificity, precision and identifiers of classifiers. The identifier of a classifier is a number of the row of the matrix *votes*. The matrix *effvot* can be sorted in descending order according to accuracy, sensitivity, specificity, multiple of accuracy and sensitivity and multiple of sensitivity and specificity. The way of sorting is selected according to the variable *efftype*.

Algorithm

- Create a vector with ground truth of votes using a vector *gis*.
- Compute a vector with odd numbers of classifiers which will vote.
- Repeat for each odd number of classifiers: Call *vote_effic* function to classify images into the group of patients or healthy controls according to the majority vote of the classifiers and to compute classification efficiency of the combination of classifiers.
- Sort the vector with efficiencies of voting according to the *efftype*.
- Choose first *no* results of voting.

References

To learn more about the evaluation of classifier efficiency, see (Altman, 1999).

10. THOMAZ_MEANG

Purpose

Compute centroids of two groups during computation of mMLDA.

Syntax

[Mg] = thomaz_meang(X,ns,nt)

Input data and variables:

X	matrix containing the training set, whose each line points to a sample data
ns	number of groups
nt	number of training images in each group

Output data and variables:

Mg	matrix of group sample means

Description

The *thomaz_meang* function is an auxiliary function used by the *thomaz_MLDA* function. It enables computation of centroids of two groups of persons.

Algorithm

- Repeat for each group of persons: Sum all images of persons from the group and compute the mean image.

Notes

The function was proposed by Carlos Thomaz and it was modified to be incorporated into the Recognition Toolbox.

11. THOMAZ_MECS

Purpose

Maximize entropy of each group during computation of mMLDA.

Syntax

[Se,tt] = thomaz_mecs(Sp,Sg)

Input data and variables:

Sp	spooled (common) covariance matrix
Sg	sample group covariance matrix

Output data and variables:

Se	The Smix matrix based on the maximum entropy
tt	CPU total time in seconds of MECS calculation

Description

The *thomaz_mecs* function is an auxiliary function used by the *thomaz_MLDA* function. It allows calculation of a modified within-class scatter matrix based on the largest dispersion criterion.

Algorithm

- Construct a new matrix of eigenvalues based on the largest dispersion criterion.
- Compute a modified within-class scatter matrix.

Notes

The function was proposed by Carlos Thomaz and it was modified to be incorporated into the Recognition Toolbox.

The formulas of using largest dispersion criterion for computation of a modified within-class scatter matrix are given in Appendix D.

References

To learn more about the largest dispersion criterion, see (Thomaz et al., 2007b; Fujita et al., 2008).

12. THOMAZ_MLDA

Purpose

Compute classification scores using projection matrix of MLDA.

Syntax

[X_red,L] = thomaz_MLDA(X,ns,nt,nr,i_tren)

Input data and variables:

X	matrix containing the training set, whose each line points to a sample data
ns	number of groups
nt	number of training images in each group
nr	reduction of dimension (nr <= ns-1)
i_tren	indexes of all training images

Output data and variables:

Xred	reduced matrix X
L	eigenvectors (sorted) of matrix X; each column represents an eigenvector

Description

The *thomaz_MLDA* function computes classification scores of training image data using multiplication of projection matrix of maximum uncertainty linear discriminant analysis with the matrix *X*. Before input into the *thomaz_MLDA* function, the training image data *X* must be reduced using pPCA to avoid instable classification results caused by the small sample size problem. The *thomaz_MLDA* function is a subordinary function of the mMLDA function. During calculation of the MLDA projection matrix the *thomaz_MLDA* function calls the *thomaz_meang*, *thomaz_mecs* and *thomaz_repeatc* functions.

Algorithm

- Compute a within-class scatter matrix and a between-class scatter matrix. The *thomaz_repeatc* and *thomaz_meang* functions are used during computing.
- Calculate a matrix S_p which is derived from the within-class scatter matrix.
- Call the *thomaz_mecs* function. Find eigenvalues and eigenvectors of the matrix S_P and construct a modified within-class scatter matrix based on the largest dispersion criterion.
- Compute eigenvectors and eigenvalues of the matrix which is derived by multiplication of inverse modified within-class scatter matrix and the between-class scatter matrix.
- Sort the eigenvalues and corresponding eigenvectors.
- Build the projection matrix of MLDA from the eigenvectors which are column vectors.
- Compute classification scores *X_red* of training images using the MLDA projection matrix.

Notes

The function was proposed by Carlos Thomaz and it was modified to be incorporated into the Recognition Toolbox. The mathematical background of MLDA is described in Appendix D.

References

To learn more about the MLDA, see (Thomaz et al., 2007a; Thomaz et al., 2007b; Fujita et al., 2008).

13. THOMAZ_REPEATC

Purpose

Build a matrix by repeating each column of original matrix nt-times. It is used during computation of mMLDA.

Syntax

[H] = thomaz_repeatc(X,nt)

Input data and variables:

X	matrix containing the training set, whose each line points to a sample data
nt	number of training images in each group

Output data and variables:

H	matrix built by repeating each column i of matrix X nt(i) times

Description

The *thomaz_repeatc* function is an auxiliary function used by the *thomaz_MLDA* function. It enables building a matrix *H* by repeating each column *i* of matrix *X nt(i)*-times.

Algorithm

- Repeat each column *i* of matrix *X nt(i)*-times to create a matrix *H.*

Notes

The function was proposed by Carlos Thomaz and it was modified to be incorporated into the Recognition Toolbox.

14. VOTE_EFFIC

Purpose

Compute all possible combinations of a selected number of classifiers and calculate classification efficiency of voting of the combinations of classifiers.

Syntax

[effic,call] = vote_effic(votes,vot_gt,Nc,i_pat)

Input data and variables:

votes	matrix of votes (group identifiers) which are results of classifiers for each person (rows are classifiers, columns are persons)
vot_gt	ground truth of votes
Nc	number of classifiers which will vote
i_pat	indexes of patient images

Output data and variables:

eff	efficiency of voting combinations
call	all possible combinations of classifiers

Description

The *recog_effic* function computes efficiency of recognition of 2-D or 3-D image data for all possible combinations of *Nc* classifiers. Input variables are results of the *recog* function. An output variable effic is a matrix where rows are classifiers and columns are true positive results of recognition, false negative results, true negative results, false positive results, accuracy, sensitivity, specificity and precision. The *recog_effic* function is a subordinate function of the *recog_voting* function which combines results of voting of all odd *Nc* numbers of classifiers.

Algorithm

- Generate all possible combinations of *Nc* classifiers.
- Compute efficiency for each combination of classifiers.

References

To learn more about the evaluation of efficiency of classifiers, see (Altman, 1999).

APPENDIX

APPENDIX A Mathematical background of PCA based on the covariance matrix of persons (pPCA)

Let N x n data matrix **X** be an input into the pPCA. The **X** is composed of N input images with n voxels. According to linear algebra rules, nonzero eigenvalues of a covariance matrix of voxels $\mathbf{X}^T\mathbf{X}$ and a covariance matrix of persons $\mathbf{X}\mathbf{X}^T$ are the same and eigenvectors corresponding to the higher dimensional covariance matrix can be derived from the eigenvectors of the smaller one by:

$$\mathrm{v}_j = \frac{\mathrm{X}^T \mathrm{w}_j}{\sqrt{\lambda_j (N-1)}}, \tag{1}$$

where $\mathbf{v}_j$ is the j^{th} eigenvector of the covariance matrix of voxels, $\mathbf{X}^T$ is the transposed image data matrix, $\mathbf{w}_j$ is the j^{th} eigenvector of the covariance matrix of persons, λ_j is the j^{th} eigenvalue of the covariance matrix of subjects and N is the number of input images. The proof of the transformation is given in (Demirci et al., 2008; Fukunaga, 1990).

The pPCA algorithm can be described in this way:

1. Calculate N x N covariance matrix of persons $\mathbf{C}_s$ of the data matrix **X** by $\mathbf{C}_s = \frac{1}{N-1}\left(\mathbf{X}-\overline{\mathbf{X}}\right)\left(\mathbf{X}-\overline{\mathbf{X}}\right)^T$, where N is the number of input images and $\overline{\mathbf{X}}$ is a matrix with all rows equal to a mean image $\overline{\mathbf{x}}$ which is defined by $\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$, where $\mathbf{x}_i$, $i = 1,...,N$, are rows of the matrix **X**.
2. Find λ_j eigenvalues and $\mathbf{w}_j$ eigenvectors of the covariance matrix of persons $\mathbf{C}_\mathrm{s}$, $j = 1,...,N$.
3. Select all $m = N - 1$ eigenvectors that correspond to all nonzero eigenvalues.
4. Compute eigenvectors $\mathbf{v}_j$ of the covariance matrix of voxels $\mathbf{C}_\mathrm{v}$ by $\mathrm{v}_j = \frac{\mathrm{X}^T \mathrm{w}_j}{\sqrt{\lambda_j (N-1)}}$.
5. Construct n x m projection matrix $\boldsymbol{V}_{pPCA}$ with column-wise computed eigenvectors $\mathbf{v}_j$.
6. Compute a reduced data matrix **Y** with the size of N x m by $\mathbf{Y} = \left(\mathbf{X}-\overline{\mathbf{X}}\right)\cdot \mathbf{V}_{pPCA}$.

APPENDIX B Mathematical background of the average linkage method

The average linkage is one of the clustering methods where images are classified into a group according to distances in a feature space. The average linkage enables classification of images reduced by pPCA as well as original 3-D images. However, 3-D images must be transformed into 1-D vectors before classification. The average linkage method can be described using following two steps:

1. Compute a mean Euclidean distance between a testing image $\mathbf{y}$ and all patient images by $\overline{d}_1(\mathbf{y}, \mathbf{y}_1) = \frac{1}{n_1}\sum_{i=1}^{n_1} d_i(\mathbf{y}, \mathbf{y}_{1i})$, where n_1 is the number of patients and $\mathbf{y}_{1i}$ is the i^{th} patient's image; and calculate a mean Euclidean distance between a testing image $\mathbf{y}$ and all healthy control ones by $\overline{d}_2(\mathbf{y}, \mathbf{y}_2) = \frac{1}{n_2}\sum_{i=1}^{n_2} d_i(\mathbf{y}, \mathbf{y}_{2i})$, where n_2 is the number of healthy controls and $\mathbf{y}_{2i}$ is the i^{th} control's image.
2. Assign the testing image $\mathbf{y}$ into the group with the shortest mean Euclidean distance; it means the goal is to find $\min\left(\overline{d}(\mathbf{y}, \mathbf{y}_1), \overline{d}(\mathbf{y}, \mathbf{y}_2)\right)$.

APPENDIX C Mathematical background of the centroid method

The centroid method falls into the clustering methods. The centroid method enables classification of images reduced by pPCA and classification of original 3-D images which are transformed into 1-D vectors before classification. The algorithm of the centroid method consists of two main steps:

1. Compute a centroid of the group of patients $\overline{\mathbf{y}}_1$ by $\overline{\mathbf{y}_1} = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{y}_{1i}$, where n_1 is the number of patients and $\mathbf{y}_{1i}$ is the i^{th} patient's image; calculate a centroid of the group of healthy controls $\overline{\mathbf{y}}_2$ by $\overline{\mathbf{y}}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2} \mathbf{y}_{2i}$, where n_2 is the number of healthy controls and $\mathbf{y}_{2i}$ is the i^{th} control's image.
2. Assign a testing image $\mathbf{y}$ to the group represented by the closer centroid; it means the goal is to find the minimum Euclidean distance $\min\left(d(\mathbf{y}, \overline{\mathbf{y}}_1), d(\mathbf{y}, \overline{\mathbf{y}}_2)\right)$.

APPENDIX D Mathematical background of modified MLDA

MLDA belongs to linear discriminant analysis methods. It was proposed by Thomaz et al. (2007a) to classify brain images of preterm infants. The Recognition Toolbox contains modification of Thomaz's MLDA. The modification was designed to improve classification results.

A reduced data matrix **Y** which was computed using pPCA (see Appendix A) is the input into the classification algorithm. Original 3-D images without any reduction are not allowed to be an input into the MLDA unlike the centroid method or the average linkage. The size of the reduced data matrix **Y** is N x m, where N is the number of input images and $m = N - 1$ is the number of features. Let π_1 be a group of patients and π_2 be a group of healthy control persons. Steps of data reduction by MLDA are described in detail in (Thomaz et al., 2007b; Fujita et al., 2008) and can be shortly summarized in this way:

1. Let a within-class scatter matrix $\mathbf{S}_w$ be defined as $\mathbf{S}_w = \sum_{i=1}^{g}\sum_{j=1}^{N_i}\left(\mathbf{x}_{i,j} - \overline{\mathbf{x}_i}\right)\left(\mathbf{x}_{i,j} - \overline{\mathbf{x}_i}\right)^T$ and a between-class scatter matrix $\mathbf{S}_b$ be defined as $\mathbf{S}_b = \sum_{i=1}^{g} N_i\left(\overline{\mathbf{x}_i} - \overline{\mathbf{x}}\right)\left(\overline{\mathbf{x}_i} - \overline{\mathbf{x}}\right)^T$, where g is the total number of groups (here g = 2), the vector $\mathbf{x}_{i,j}$ is the m-dimensional pattern j from the group π_i, N_i is the number of training patterns from group π_i, the vector $\overline{\mathbf{x}}_i$ is the unbiased sample mean of group π_i and $\overline{\mathbf{x}}$ is overall mean vector.
2. Find eigenvalues λ_j and eigenvectors Φ_j, j = 1,...,m, of a matrix $\mathbf{S}_p$, where $\mathbf{S}_p = \frac{\mathbf{S}_w}{N - g}$.
3. Calculate average eigenvalue $\overline{\lambda}$ of the matrix $\mathbf{S}_p$ by $\overline{\lambda} = \frac{tr\left(\mathbf{S}_p\right)}{m}$, where $tr(\mathbf{S}_p)$ is a trace of the matrix $\mathbf{S}_p$.
4. Construct a new matrix of eigenvalues based on the following largest dispersion criterion $\Lambda^* = diag\left[\max\left(\lambda_i, \overline{\lambda}\right), ..., \max\left(\lambda_m, \overline{\lambda}\right)\right]$.
5. Form the modified within-class scatter matrix $\mathbf{S}_w^*$ by $\mathbf{S}_w^* = \left(\Phi\Lambda^*\Phi^T\right)\left(N - g\right)$.
6. Calculate the projection matrix $\mathbf{V}_{MLDA}$ with column-wise eigenvectors of the matrix **S**, where $\mathbf{S} = \mathbf{S}_w^{*\,-1}\mathbf{S}_b$; the projection matrix $\mathbf{V}_{MLDA}$ maximizes the ratio of the determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix (Fishers's criterion).
7. Multiply the reduced matrix **Y** by the projection matrix $\mathbf{V}_{MLDA}$ to compute a matrix with the classification scores **Z** with the type of N x $(g - 1)$ by $\mathbf{Z} = \mathbf{Y} \cdot \mathbf{V}_{MLDA}$.

Classification into two groups is performed in the Recognition Toolbox. In two group data, pPCA and MLDA enables to reduce each 3-D input image into one number, a classification score. The classification scores are stored in a vector **z**.

A testing image is centred using subtraction of a mean image $\bar{\mathbf{x}}$ from the testing image. Afterwards, the testing image is reduced by the projection matrix $\mathbf{V}_{pPCA}$ and by the projection matrix $\mathbf{V}_{MLDA}$ into the classification score. The testing image is classified as a patient's image or healthy control's one according to the classification boundary. If the score is lower than the boundary, images are classified into the first group (patients); if the score is higher than the boundary, images are classified into the second group (healthy controls).

Thomaz et al. (2007a) compute the classification boundary as an arithmetic mean of a mean classification score of patients $\overline{z_1}$ and a mean classification score of healthy control subjets $\overline{z_2}$. In the Recognition Toolbox, the classification boundary of mMLDA z_{mMLDA} is based on computation of a mean which is weighted by standard deviations of classification scores of persons:

$$z_{mMLDA} = \frac{\frac{\overline{z_1}}{SD_1} + \frac{\overline{z_2}}{SD_2}}{\frac{1}{SD_1} + \frac{1}{SD_2}} = \frac{\overline{z_1} SD_2 + \overline{z_2} SD_1}{SD_1 + SD_2}, \tag{2}$$

where SD_1 is a standard deviation of classification scores of patients and SD_2 is a standard deviation of classification scores of controls. From the formula it is evident that the small standard deviation of a group indicates the high weight of the mean classification score of the group. The way of computation of weighted mean was used in analyses of microarray data (Culhane et al., 2002).

REFERENCES

Altman, D. G. 1999, *Practical Statistics for Medical Research*, Chapman and Hall/CRC, London.

Culhane, A. C., Perriere, G., Considine, E. C., Cotter, T. G. & Higgins, D. G. 2002. ‘Between-group analysis of microarray data‘, *Bioinformatics*, vol. 18, pp. 1600-1608.

Demirci, O., Clark, V. P., Magnotta, V. A., Andreasen, N. C., Lauriello, J., Kiehl, K. A., Pearlson, G. D. & Calhoun, V. D. 2008, ‘A Review of challenges in the use of fMRI for disease classification / characterization and a projection pursuit application from multi-site fMRI schizophrenia study‘, *Brain Imaging and Behavior*, vol. 2, pp. 147-226.

Fujita, A., Gomes, L. R., Sato J. R., Yamaguchi, R., Thomaz, C. E., Sogayar, M. C. & Miyano, S. 2008, ‘Multivariate gene expression analysis reveals functional connectivity changes between normal/tumoral prostates‘, *BMC Systems Biology*, vol. 2:106, pp. 1-14.

Fukunaga, K. 1990, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego.

Legendre, P., & Legendre, L. 1998, *Numerical ecology*, Elsevier Science, Amsterdam.

Thomaz, C. E., Boardman, J. P., Counsell, S., Hill, D. L. G., Hajnal, J. V., Edwards, A. D., Rutherford, M. A., Gillies, D. F. & Rueckert, D. 2007a, ‘A multivariate statistical analysis of the developing human brain in preterm infants‘, *Image and Vision Computing*, vol. 25, pp. 981-994.

Thomaz, C. E., Duran, F. L. S., Busatto, G. F., Gillies, D. F. & Rueckert, D. 2007b, ‘Multivariate statistical differences of MRI samples of the human brain’, *Journal of Mathematical Imaging and Vision*, vol. 29, pp. 95-106.