Change detection and Canonical correlation analysis

Karina Nielsen (karni@space.dtu.dk) and Allan Aasbjerg Nielsen

November, 2022

Overview of module 5

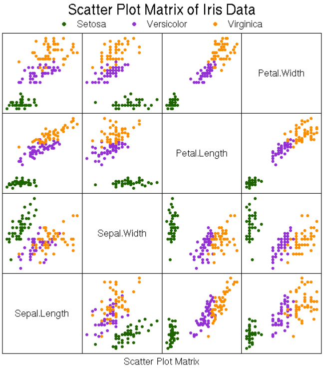
- Repetition of unsupervised Classification
- Canonical correlation analysis for change detection
- Exercises (Report, not mandatory, but strongly recommended to do)
- Next time Applications
  - Thesis Presentations by Simon and Magdalene
  - Geust lecturer Michael Schultz Rasmussen
  - Change detection project presentations (more on this later, important part of the formal assessment)
  - Evaluation of optical remote sensing part

Classification

- classification is the process of grouping observations (pixels or regions) into classes intended to represent different physical objects or types
- here, the production of a **thematic map** from (image) data with digital numbers representing for example reflected or emitted EM-radiation in different wavelength bands
- very many classification methods ranging from quite simple to highly advanced
- two major groups of methods: supervised and unsupervised
  - supervised: ideally physical classes but not necessarily statistically distinct
  - unsupervised: statistically distinct but not necessarily physical classes
- In remote sensing:
  - supervised classification: Here we obtain information classes
  - unsupervised classification: Here we obtain spectral classes

Feature space

- p variables
  C classes
  N observations (or samples)
- $\boldsymbol{x}_i, \ i = 1, \ldots, N, \quad p \times 1$
  is a point (or vector) in p-dimensional **feature space**
- figure shows all possible pairwise projections on original variables



K-means

We need to decide how many classes we have

- choose C
- assign C class centres $\boldsymbol{\mu}_c$
- calculate distance, e.g., $D^2_{Eic} = (\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{x}_i - \boldsymbol{\mu}_c)$ for all observations to all class centres, $i = 1, \ldots, N, \; c = 1, \ldots, C$
- assign class c to $\boldsymbol{x}_i$ if distance smallest for class c
- compute new class centres $\boldsymbol{\mu}_c$ (include only obs in class c)
- iterate from third step

Initialization of μ_c

- random observations within range of data
- first C 'different enough' observations
- based on PCA, e.g., uniformly distributed along first PC axis, or in plane spanned by two first PC axes
- ...

Fuzzy c-means

1. choose C
2. assign C class centres $\boldsymbol{\mu}_c$
3. calculate distance, e.g., $D^2_{Eic} = (\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{x}_i - \boldsymbol{\mu}_c)$ for all observations to all class centres
4. assign degree of membership u_{ic} to $\boldsymbol{x}_i$ for all classes, e.g., $u_{ic} = (1/D^2_{Eic})/\sum_{j=1}^{C} 1/D^2_{Eij}$ leading to $\sum_{c=1}^{C} u_{ic} = 1$
5. compute new class centres (include all obs weighted by u_{ic}) $\boldsymbol{\mu}_c = \sum_{i=1}^{N} u_{ic}\boldsymbol{x}_i / \sum_{i=1}^{N} u_{ic}$
6. iterate from third step

Gaussian mixture models, GMM

- Bayes' rule: $P(\omega_c|\boldsymbol{x}_i) = K\ P(\boldsymbol{x}_i|\omega_c)P(\omega_c)$ with $1/K = \sum_{j=1}^{C} P(\boldsymbol{x}_i|\omega_j)P(\omega_j)$
- **GMM**: Given some $u_{ic} = P(\omega_c|\boldsymbol{x}_i)$ with $\sum_{c=1}^{C} u_{ic} = 1$, calculate
- $P(\omega_c) = \frac{1}{N}\sum_{i=1}^{N} u_{ic}$ (interpreted as proportion of class c)
  $\boldsymbol{\mu}_c = \frac{1}{NP(\omega_c)}\sum_{i=1}^{N} u_{ic}\boldsymbol{x}_i$
  $\boldsymbol{\Sigma}_c = \frac{1}{NP(\omega_c)}\sum_{i=1}^{N} u_{ic}(\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^T$
- $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ define $P(\boldsymbol{x}_i|\omega_c)$ which with $P(\omega_c)$ via Bayes' rule give a new $u_{ic} = P(\omega_c|\boldsymbol{x}_i)$ which in turn gives a new $P(\omega_c)$: iterate
- example on Expectation Maximization (EM) algorithm
  E-step: calculate $P(\omega_c) = \cdots,\ \boldsymbol{\mu}_c = \cdots,\ \boldsymbol{\Sigma}_c = \cdots$
  M-step: calculate $P(\omega_c|\boldsymbol{x}_i)$ in Bayes' rule

Initialization of μ_c and Σ_c

1. select observations at random as initial means
   mixing proportions are uniform
   initial covariance matrices are diagonal, elements on the diagonal are the variances
2. start with result from k-means or fuzzy c-means
3. ...

Examples of change detection

- Deforestation
- Flooding and Surface water extent variation
- Urban development
- Seasonal change
- Wild fire detection
- Drought
- Change in Glaciers
- Change in sea ice
- ...

Change detection methods

- Visual
- Image difference
- Principal component analysis
- Classification methods supervised and unsupervised
- Multivariate alternation detection (MAD)
    - Iteratively reweighted MAD

Visual

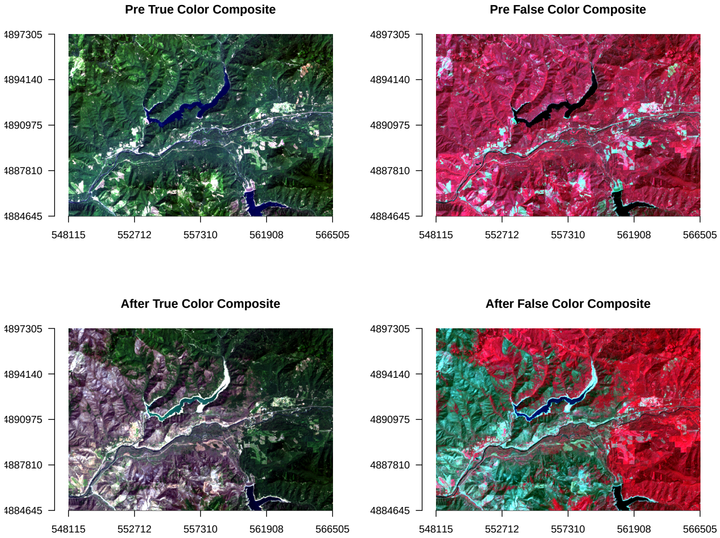

Pre True Color Composite
Pre False Color Composite
After True Color Composite
After False Color Composite
4897305
4894140
4890975
4887810
4884645
548115
552712
557310
561908
566505

Image difference

- Assume we have 2 multi-spectral images with k bands taken at different times. Considering one location we have $\boldsymbol{X} = [X_1, \ldots, X_k]$ and $\boldsymbol{Y} = [Y_1, \ldots, Y_k]$
- We can take the simple difference between the individual bands
    - Data must be well calibrated and referenced to a common level and on the same scale

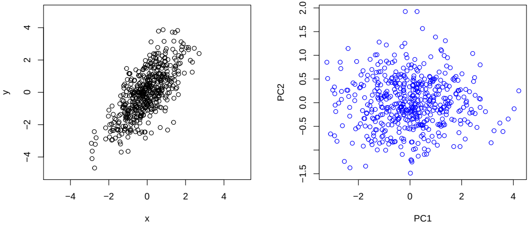
$$\boldsymbol{X} - \boldsymbol{Y} = [X_1 - Y_1, \cdots, X_k - Y_k]^T, \quad \mu_X = \mu_Y = 0$$

Change detection by PCA

- Remember in PCA we perform a linear transformation of the data to the directions where the variance is maximal

$$Y = ZA$$

- Z is an $n \times m$ matrix consisting of m band differences between the two images and n is the number of observations.
- A is an $m \times m$ matrix



Change detection by PCA

$$VA = AD$$

- This can be recognized as the eigenvalue equation, where D is the eigenvalues, A is the matrix containing the eigenvectors and V is the data covariance matrix.
- Here the data covariance matrix is constructed from the band differences.
- Hence
  1. standardize the original images
  2. Construct image differences of original band values
  3. Standardize differences
  4. Perform PCA
  5. Investigate the PCs for change
- Limitations
  - Scale dependent
  - The two images need to have the same number of bands

Related to exercise: why we detect changes with the PC2 and lower? Because PC1 basically represent the main image.

Change detection via classifications

- First form a combined image stack based on the images from time 1 and 2.
- Extract training data
    - display false color composites that enhance the change
    - The training data should represent areas of change and no-change
        - class 1 forest no forest   class 0 : forest forest
    - class 2 water water
    - class 3 water no water
    - class 4 ...
- Next perform classification on the combined image
    - Using Supervised/unsupervised classification
        - Maximum likelihood classification

For more information you can follow this detailed tutorial by NASA

Multivariate alternation detection (MAD)

- A more parameter rich transformation that allows for different coefficients for X and Y and a different number of bands p and q in the two images is (again we consider one location) We take image, we create linear combination, same for the second image

$$\boldsymbol{a}^T\boldsymbol{X} = a_1X_1 + \cdots + a_pX_p$$
$$\boldsymbol{b}^T\boldsymbol{Y} = b_1Y_1 + \cdots + b_qY_q$$

- To obtain change we can form the differences $\boldsymbol{a}^T\boldsymbol{X} - \boldsymbol{b}^T\boldsymbol{Y}$ and maximize $\mathrm{Var}(\boldsymbol{a}^T\boldsymbol{X} - \boldsymbol{b}^T\boldsymbol{Y})$, with the constraint that $\mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X}\} = \mathrm{Var}\{\boldsymbol{b}^T\boldsymbol{Y}\} = 1$

$$\mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X}-\boldsymbol{b}^T\boldsymbol{Y}\} = \mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X}\}+\mathrm{Var}\{\boldsymbol{b}^T\boldsymbol{Y}\}-2\mathrm{Cov}\{\boldsymbol{a}^T\boldsymbol{X},\boldsymbol{b}^T\boldsymbol{Y}\} = 2(1-\mathrm{Corr}\{\boldsymbol{a}^T\boldsymbol{X},\boldsymbol{b}^T\boldsymbol{Y}\})$$

- With $\mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X}\} = \mathrm{Var}\{\boldsymbol{b}^T\boldsymbol{Y}\} = 1.$ We further request that $\boldsymbol{a}^T\boldsymbol{X}, \boldsymbol{b}^T\boldsymbol{Y}$ are positively correlated
- To perform this transformation we must determine the coefficient $\boldsymbol{a}$ and $\boldsymbol{b}$
- The last term can be recognized as the canonical correlation
- The MAD (and subsequent additions) method was developed by Allan AAsbjerg Nielsen DTU Compute
- Implemented in google earth engine see here

Canonical correlation analysis

- Canonical correlation analysis CCA is a means of assessing the relationship between two sets of variables.
- The idea is to study the correlation between a linear combination of the variables in one set and a linear combination of the variables in another set.
- Let us have a look at the canonical correlation

$\boldsymbol{X} = (X_1, \ldots, X_p)$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_q)$ denote random vectors with mean vectors μ_X and μ_Y and covariance matrices Σ_X and Σ_Y

let $\boldsymbol{Z}$=($\boldsymbol{X}$,$\boldsymbol{Y}$) , $\boldsymbol{Z} \sim (\mu, \Sigma)$, $\mu = (\mu_X, \mu_Y)$

$$\Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix}$$

- Σ_{XX} is a $p \times p$ covariance matrix obtained from $\boldsymbol{X}$
- Σ_{YY} is a $q \times q$ covariance matrix obtained from $\boldsymbol{Y}$
- Σ_{XY} is a $p \times q$ covariance matrix obtained from $\boldsymbol{X}$ and $\boldsymbol{Y}$
- $\Sigma_{YX} = \Sigma_{XY}^T$

Canonical correlation analysis

Introduce the linear combinations of $\boldsymbol{X}$ and $\boldsymbol{Y}$

$$U = \boldsymbol{a}^T \boldsymbol{X}$$
$$V = \boldsymbol{b}^T \boldsymbol{Y}$$

Where $\text{Var}(U) = \boldsymbol{a}^T \Sigma_{XX} \boldsymbol{a}$ and $\text{Var}(V) = \boldsymbol{b}^T \Sigma_{YY} \boldsymbol{b}$ and $\text{Cov}(U, V) = \boldsymbol{a}^T \Sigma_{XY} \boldsymbol{b}$

Now find a and b $\text{Corr}(U, V)$ gets as large as possible under the constraint that $\text{Var}(U) = \text{Var}(V) = 1$

$$\text{Corr}(U, V) = \frac{\text{Cov}(U, V)}{\sqrt{\text{Var}(U)}\sqrt{\text{Var}(V)}} = \frac{\boldsymbol{a}^T \Sigma_{xy} \boldsymbol{b}}{\sqrt{\boldsymbol{a}^T \Sigma_{xx} \boldsymbol{a}}\sqrt{\boldsymbol{b}^T \Sigma_{yy} \boldsymbol{b}}}$$

Canonical correlation some definitions

- Here U_1 and V_1 is the firs set of canonical variables having unit variances and which maximize the correlation

$$\text{Corr}(U, V) = \frac{\boldsymbol{a}^T \Sigma_{xy} \boldsymbol{b}}{\sqrt{\boldsymbol{a}^T \Sigma_{xx} \boldsymbol{a}} \sqrt{\boldsymbol{b}^T \Sigma_{yy} \boldsymbol{b}}}$$

- The second pair of canonical variables is the pair of linear combinations U_2 and V_2 having unit variances, which maximize the correlation among all choices that are uncorrelated with the first pair of canonical variables.
- and so on
- To find the parameters that maximize the correlation we calculate the partial derivative of $\rho = \text{Corr}(U, V)$ with respect to $\boldsymbol{a}$ and $\boldsymbol{b}$ and set these to zero

We want to align these 2 different images, thus we want to maximise the correlation

Canonical correlations: solution

- We will not cover the proof here but please see e.g. the appendix in Nielsen, A.A 2007 (available via Learn)
- The pair of canonical varaibale U_k and V_k are given by

$$U_k = \underbrace{e_k^T \Sigma_{xx}^{-1/2}}_{\boldsymbol{a}^T} \boldsymbol{X} \quad \text{and}\, V_k = \underbrace{f_k^T \Sigma_{yy}^{-1/2}}_{\boldsymbol{b}^T} \boldsymbol{Y}$$

- To ensure unit variance we redefine $\boldsymbol{a}$ and $\boldsymbol{b}$, remember that $\text{Var}(z\boldsymbol{X}) = \boldsymbol{z}^T \Sigma_{XX} \boldsymbol{z}$, and $AA^{-1} = I$
- We have that $\text{Var}(U_k) = \text{Var}(V_k) = 1$
- and $\text{Cov}(U_k, V_k) = \rho_k$, where $\rho_1^2 \geq \rho_2^2, \geq \ldots \geq \rho_p^2$ and $(\rho_k^2, \boldsymbol{e}_k)$ and $(\rho_k^2, \boldsymbol{f}_k)$ are the eigenvalue-eigenvector pair of the systems, respectively

$$\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} \Sigma_{xx}^{-1/2}$$

$$\Sigma_{yy}^{-1/2} \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy} \Sigma_{yy}^{-1/2}$$

Canonical correlations: solution

The solution can also be written as

$$\Sigma_{XX}^{-1}\Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}\boldsymbol{a} = \rho^2\boldsymbol{a}$$

$$\Sigma_{YY}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}\boldsymbol{b} = \rho^2\boldsymbol{b}$$

- We only need to solve one of the above to determine $\boldsymbol{a}$ or $\boldsymbol{b}$ the remaining can be found from

$$\boldsymbol{a} = \frac{1}{\rho}\Sigma_{XX}^{-1}\Sigma_{XY}\boldsymbol{b}$$

$$\boldsymbol{b} = \frac{1}{\rho}\Sigma_{YY}^{-1}\Sigma_{YX}\boldsymbol{a}$$

Multivariate alternation detection (MAD)

$$\mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X} - \boldsymbol{b}^T\boldsymbol{Y}\} = \mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X}\} + \mathrm{Var}\{\boldsymbol{b}^T\boldsymbol{Y}\} - 2\mathrm{Cov}\{\boldsymbol{a}^T\boldsymbol{X}, \boldsymbol{b}^T\boldsymbol{Y}\} = 2(1 - \mathrm{Corr}\{\boldsymbol{a}^T\boldsymbol{X}, \boldsymbol{b}^T\boldsymbol{Y}\})$$

With $\mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X}\} = \mathrm{Var}\{\boldsymbol{b}^T\boldsymbol{Y}\} = 1$

Last term can be recognized as the canonical correlation

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix} \to \begin{bmatrix} \boldsymbol{a}_1^T\boldsymbol{X} - \boldsymbol{b}_1^T\boldsymbol{Y} \\ \vdots \\ \boldsymbol{a}_p^T\boldsymbol{X} - \boldsymbol{b}_p^T\boldsymbol{Y} \end{bmatrix}$$

$$\mathrm{Var}\{\boldsymbol{a}^T\boldsymbol{X} - \boldsymbol{b}^T\boldsymbol{Y}\} = 2(\boldsymbol{I} - \boldsymbol{R})$$

, where $\boldsymbol{R}$ contains the canonical correlations in the diagonal (large to small)

$$\sigma^2_{MAD_i} = 2(1 - \rho_i)$$

Multivariate alternation detection (MAD)

- The MAD variates will be approximately Gaussian distributed
- If there is no change in a given pixel j then MAD_{ij} has mean 0
- Ideally, the sum of standardized MAD variates follow a χ^2 distribution
- We can construct a change/no-change map via the following relation

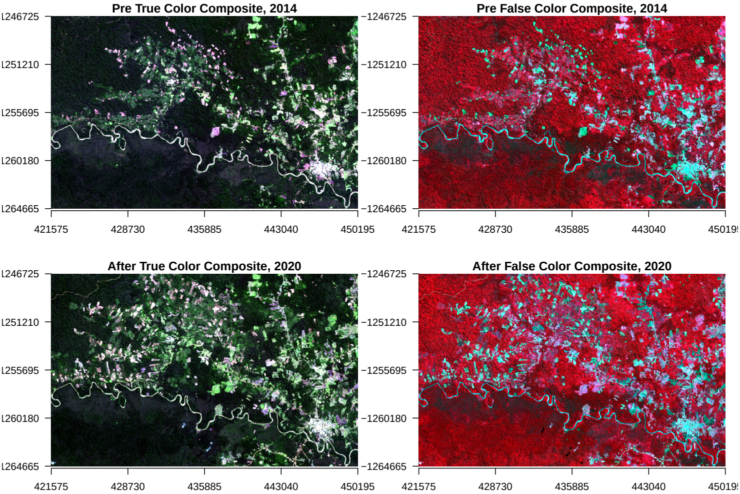
$$T_j = \sum_{i=1}^{p} \left(\frac{MAD_{ij}}{\sigma_{MAD_i}} \right)^2 \in \chi^2(p)$$

- Here σ_{MAD_i} ideally are the standard deviation of the the no change observations
- In MAD space the orthogonal variables are order by similarity (measured by linear correlation)

How to perform CCA/MAD in R

- In R there are different packages for automatically calculating the Canonical correlation ρ and pairs (U, V)
- From "R base" we use the function `cancor()`
- From the R package "CCA" use the function `cc()`
- You can also do your own implementation see slide 20 and 21
- In the following example we will use both approaches

An example: Deforestation of the Amazonian forest



An example: Deforestation of the Amazonian forest

```r
#library(CCA)
# read a raster stack
#old <- stack("L8_brazil2014.tif")
#new <- stack("L8_brazil2020.tif")
#standardize data
X<-old[]
Y<-new[]
XX <- apply(X, 2, function(x){(x - mean(x))/sd(x)})
YY <- apply(Y, 2, function(x){(x - mean(x))/sd(x)})
CCA<-cc(XX,YY)
#the canonical correlations
CCA$cor
```

```
## [1] 0.75906151 0.72103573 0.49301962 0.31438743 0.23645527 0.09047075
```

An example: Deforestation of the Amazonian forest

```
# The coefficients a
CCA$xcoef
```

```
##          [,1]       [,2]       [,3]       [,4]         [,5]        [,6]
## B2  0.5008440 -0.7763340  0.3249526  5.2593949 -0.003719877  0.71158273
## B3 -0.1264030  2.1697837  1.0488248 -0.6628310  1.800796753 -2.78378574
## B4  0.3904895 -0.7213187 -2.5600902 -4.0662417 -1.017745633  4.02253643
## B5 -0.2675876 -0.5799503 -1.7467104  0.1294203 -0.098842475  0.04442054
## B6 -0.8038629 -0.2097875  2.7436958  0.8467020  1.125400920  3.34996403
## B7 -0.4664680  0.3389098 -1.4852880 -1.0634667 -2.149188127 -4.80694650
```

```
# The coefficients b
CCA$ycoef
```

```
##          [,1]       [,2]       [,3]        [,4]       [,5]        [,6]
## B2  0.6231246 -0.9139935  0.1029989  5.15051942 -0.4796939  0.77526779
## B3 -0.6634723  2.4386655  0.9494022 -0.38889120  2.2678327 -2.08060375
## B4  1.0400001 -0.8108397 -2.2747078 -4.26809722 -1.2896550  3.10183739
## B5 -0.1786463 -0.4885270 -1.5813530  0.04276403 -0.1799364  0.03480602
## B6 -0.9906122 -0.7188572  2.7796888  0.93603262  0.8327673  2.55601603
## B7 -0.5617261  0.6578700 -1.7261408 -1.19911657 -1.7237212 -3.93702334
```

An example: Deforestation of the Amazonian forest

- Now we can check that $\text{Var}(\boldsymbol{a}^T \boldsymbol{X}) = 1$ and that $\text{Corr}(U, V) = \rho$

```
varU<-t(CCA$xcoef)%*%cov(XX)%*%CCA$xcoef
round(varU,4)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    1    0    0    0
## [4,]    0    0    0    1    0    0
## [5,]    0    0    0    0    1    0
## [6,]    0    0    0    0    0    1
```

```
U<-CCA$scores$xscores
V<-CCA$scores$yscores
round(cor(U,V),4)
```

```
##        [,1]  [,2]  [,3]   [,4]   [,5]   [,6]
## [1,] 0.7591 0.000 0.000 0.0000 0.0000 0.0000
## [2,] 0.0000 0.721 0.000 0.0000 0.0000 0.0000
## [3,] 0.0000 0.000 0.493 0.0000 0.0000 0.0000
## [4,] 0.0000 0.000 0.000 0.3144 0.0000 0.0000
## [5,] 0.0000 0.000 0.000 0.0000 0.2365 0.0000
## [6,] 0.0000 0.000 0.000 0.0000 0.0000 0.0905
```

An example: Deforestation of the Amazonian forest

- Let us now try to do this manually let us follow the approach on slide 21

```
covX<-cov(XX)
covY<-cov(YY)
covXY<-cov(XX,YY)
covYX<-t(covXY)
# eigenvalue decomposition to obtain the correlation and a
out<-eigen(solve(covX)%*%covXY%*%solve(covY)%*%covYX)
# the canonical correlations
rho<-sqrt(out$values)
rho
```

```
## [1] 0.75906151 0.72103573 0.49301962 0.31438743 0.23645527 0.09047075
```

```
# we see that these are the same as the output from the function cc
# the coefficients a are however not the same, why?
a<-out$vectors
a
```

```
##            [,1]        [,2]        [,3]        [,4]         [,5]         [,6]
## [1,] -0.4303102  0.30865848 -0.07169202  0.77127910  0.001166226  0.092823191
## [2,]  0.1086017 -0.86267265 -0.23139487 -0.09720275 -0.564571467 -0.363133993
## [3,] -0.3354969  0.28678524  0.56481479 -0.59630571  0.319075512  0.524724189
## [4,]  0.2299032  0.23057932  0.38536451  0.01897922  0.030988306  0.005794486
## [5,]  0.6906550  0.08340828 -0.60532243  0.12416706 -0.352826740  0.436989742
## [6,]  0.4007753 -0.13474531  0.32768870 -0.15595514  0.673796358 -0.627047425
```

An example: Deforestation of the Amazonian forest

- In R the eigenvectors are scaled to unity, meaning that they will not have the correct length to fulfill $\text{Var}(\boldsymbol{a}^T\boldsymbol{X}) = 1$ (this might be different in matlab and python)
- So we must rescale the eigenvectors in $\boldsymbol{a}$ to their correct length

```
s<-t(a[,1])%*%covX%*%a[,1] # the variance of U
s<-1/sqrt(s) # scaling parameter to obtain unit variance
a1<-a[,1]*s
```

```
## Warning in a[, 1] * s: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
# let us check
a1
```

```
## [1] -0.5008440  0.1264030 -0.3904895  0.2675876  0.8038629  0.4664680
```

```
CCA$xcoef[,1]
```

```
##         B2         B3         B4         B5         B6         B7
##  0.5008440 -0.1264030  0.3904895 -0.2675876 -0.8038629 -0.4664680
```

- The only difference is the sign witch like PCA is arbitrary

An example: Deforestation of the Amazonian forest

- We can now obtain $\boldsymbol{b} = \frac{1}{\rho}\Sigma_{YY}^{-1}\Sigma_{YX}\boldsymbol{a}$

```
b1<-1/rho[1]*solve(covY)%*%covYX%*%a1
t(b1)
```

```
##              B2        B3    B4        B5        B6        B7
## [1,] -0.6231246 0.6634723 -1.04 0.1786463 0.9906122 0.5617261
```

```
CCA$ycoef[,1]
```

```
##         B2         B3         B4         B5         B6         B7
##  0.6231246 -0.6634723  1.0400001 -0.1786463 -0.9906122 -0.5617261
```

```
a1
```

```
## [1] -0.5008440  0.1264030 -0.3904895  0.2675876  0.8038629  0.4664680
```

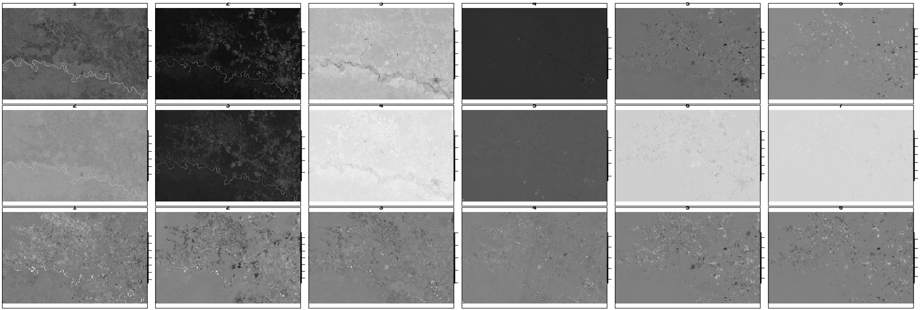
- We see the b_1 also is the same

An example: Deforestation of the Amazonian forest

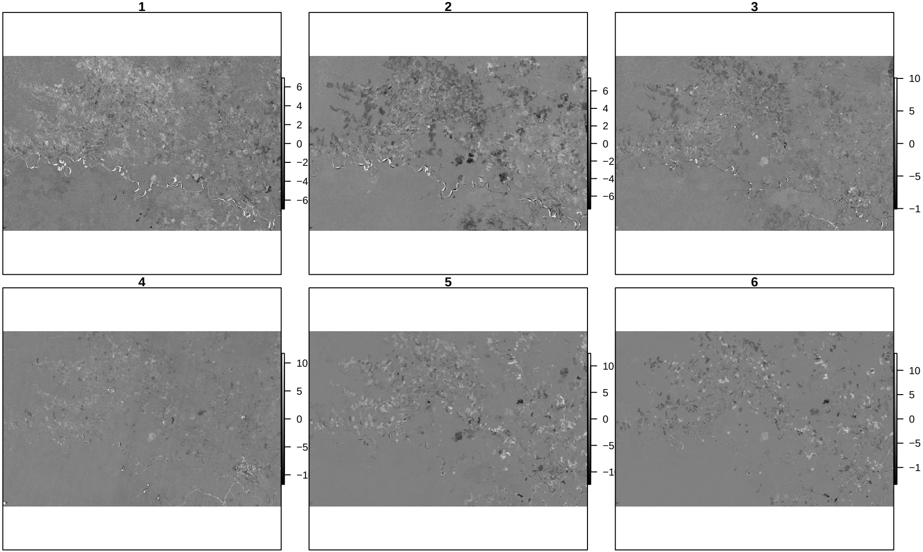
In MAD it doesn't matter if pictures have different scale !!!
For example images taken from 2 different satellite ...

- We can calculate the MAD variates and plot those to investigate the change

```
MAD<-U-V
```



An example: Deforestation of the Amazonian forest

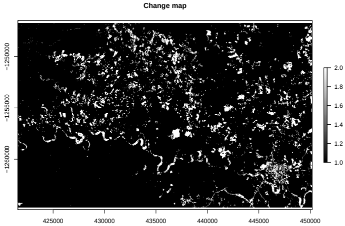


An example: Deforestation of the Amazonian forest

- We can also construct the change map

```
lim<-qchisq(0.99, df=6)
sdMAD<-sqrt(2*(1-rho))  Stand. dev. MAD

ZZsd <- lapply(1:6,function(i){(MAD[,i] - mean(MAD[,i]))/sdMAD[i]})
ZZsd<-do.call(cbind,ZZsd)
mychi<-rowSums(ZZsd^2)
change<-ifelse(mychi > lim ,2,1)
mychange <- raster(old[[1]])
mychange <- setValues(mychange, change)
plot(mychange,main='Change map',col = gray(0:100 / 100))
```



Packages in Matlab and Python

- In matlab the function `canoncorr` can be used
- sklearn.cross_decomposition import CCA

Change detection Presentation for next time (earliest 1/11)

- Form groups of 2-3 persons
- Find two images that cover a change event. You can find inspiration here
- Select 1 or more methods to detect change
- present the method and results in a small 5-10 minutes presentation

Exercise

- Write a code to carry out CCA and change detection (without the iterative bit described in the paper)
- apply it to the six band Copenhagen data from 1986 and 2005 on Learn (originally from saccess.dk) and or some of the other data set pairs.
- As usual write a small report with ample description and illustrations of data and results; include a program listing.
- Additionally
  - You can try to perform change detection via PCA and/or classification