Spectral Clustering and Kernel Methods

Robert Jenssen

Email: robert.jenssen@uit.no

- UiT Machine Learning Group: http://site.uit.no/ml
- Norsk Regnesentral NORWEGIAN COMPUTING CENTER Norwegian Computing Center

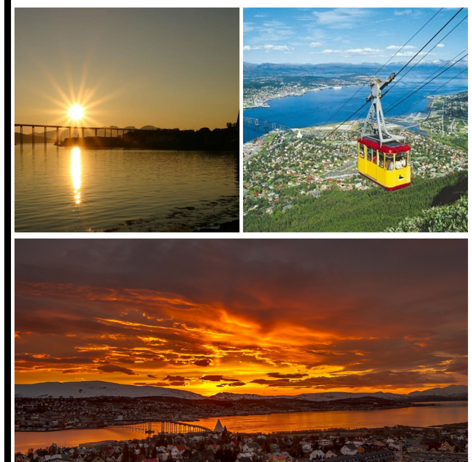
Geilo Winter School on eScience 2017

SCIA 2017

20th Scandinavian Conference on Image Analysis 12-14 June, 2017, Tromsø, Norway

SCIA 2017 12-14 JUNE

The 20th conference in the long tradition of Scandinavian Conferences on Image Analysis will take place in Tromsø, Norway on June 12-14, 2017.



Photos: www.visittromso.no

Paper Submission

The submissions will be reviewed by three anonymous reviewers. Papers will be accepted for oral or poster presentations.

Tutorials and Workshops

There will be tutorials and workshops in addition to the main program. We invite proposals for the tutorials and workshops in topics related to the main conference.

More info: www.scia2017.org

Conference Topics

The conference invites paper submissions presenting original high quality work within the following topics:

- **3D vision**
- **Color and multispectral image analysis**
- **Computational imaging and graphics**
- **Faces and gestures**
- **Feature extraction and segmentation**
- **Human-centered computing**
- **Matching, registration and alignment**
- **Medical and biomedical image analysis**
- **Motion analysis**
- **Object and scene recognition**
- **Machine learning and pattern recognition**
- **Remote sensing image analysis**
- **Robot vision**
- **Video and multimedia analysis**
- **Vision systems and applications**

Important dates:

Submission of full papers:
January 14, 2017

Proposals for tutorials/workshops:
January 14, 2017

Notification of acceptance:
March 10, 2017

Camera-ready paper:
March 24, 2017

Registration for paper presenters:
March 24, 2017

Proceedings published in Springer Lecture Notes in Computer Science

Springer

Lecture Notes in Computer Science
LNCS LNAI LNBI

Agenda

- An example
- Basic idea of kernel methods
  - Support vector machine
  - Kernel PCA
- Spectral clustering – graph view & embedding view
- Kernel entropy component analysis

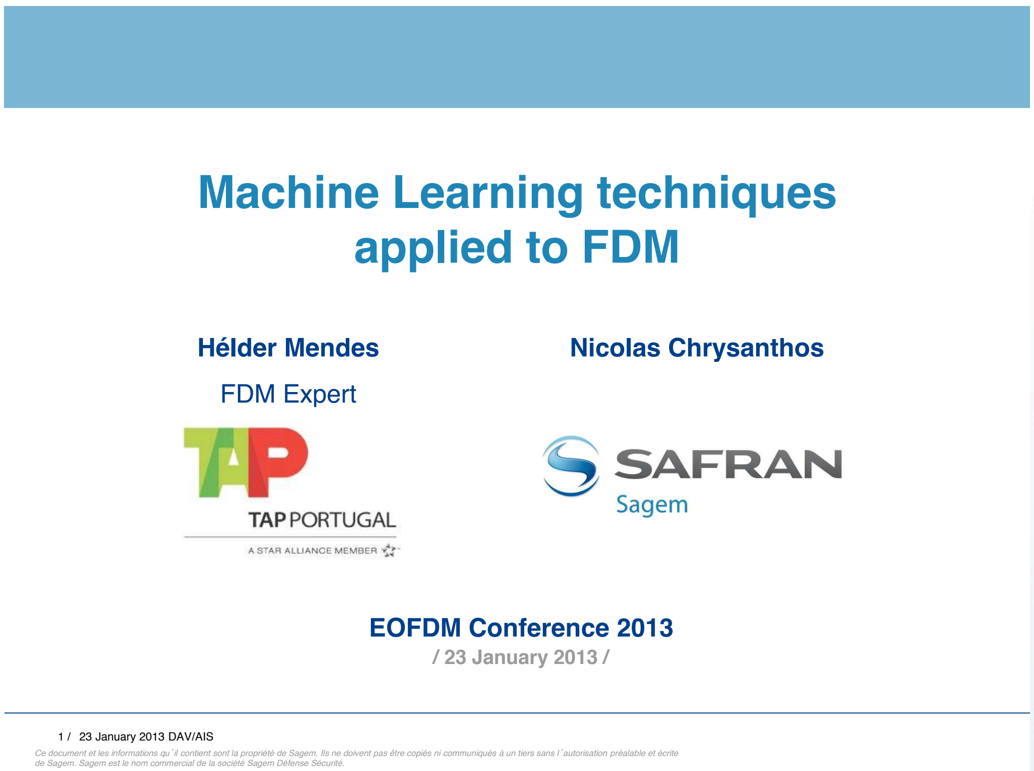
Slide inspirations

- N. Cristianini: Kernel methods for pattern analysis
- V. Zografos and K. Nordberg: Introduction to spectral clustering
- A. Singh: Spectral clustering, Carnegie Mellon
- D. Hamad and P. Biela: Introduction to spectral clustering
- M. Hein and U. Luxburg: Short introduction to spectral clustering
- J. Gao: Lecture on spectral methods, SUNY Buffalo

European Operators Flight Data Monitoring (EOFDM) Conference

EASA
European Aviation Safety Agency

UiT
THE ARCTIC UNIVERSITY OF NORWAY
MACHINE LEARNING GROUP



Safran is an international high-technology group and tier-1 supplier of systems and equipment in its core markets of Aerospace, Defense and Security.

Operating worldwide, Safran has over 70,000 employees and logged sales of 17.4 billion euros in 2015.

European Operators Flight Data Monitoring (EOFDM) Conference

EASA
European Aviation Safety Agency

UiT
THE ARCTIC UNIVERSITY OF NORWAY
MACHINE LEARNING GROUP

Machine Learning techniques applied to FDM

Hélder Mendes
FDM Expert

TAP PORTUGAL
A STAR ALLIANCE MEMBER

FLIGHT DATA MONITORING

SAFRAN
Sagem

EOFDM Conference 2013
/ 23 January 2013 /

1 / 23 January 2013 DAV/AIS

Safran is an international high-technology group and tier-1 supplier of systems and equipment in its core markets of Aerospace, Defense and Security.

Operating worldwide, Safran has over 70,000 employees and logged sales of 17.4 billion euros in 2015.

Detecting atypical flights

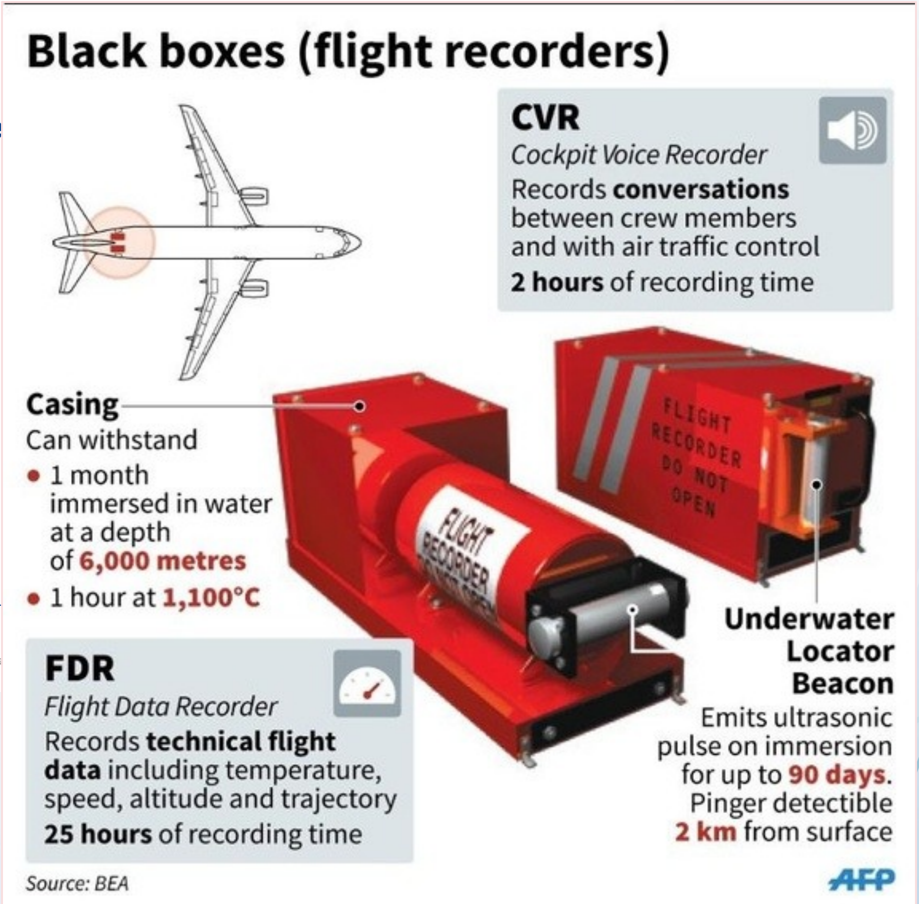
➔ « When considering a population of flights, an **atypical flight** is a flight which is in a sense different from the majority of the other flights ».

➔ Atypical flights may present **operational or safety issues** and thus need to be studied by an FDM expert!

Example study

- We have studied 721 flights from Porto to Orly 26, from **Transavia France**, same aircraft, **from approach till touch down** (10000 feet to 0).

- 14 parameters, in first pass we studied the
  - Position (latitude, longitude)
  - Altitudes, heading,
  - Roll, Pitch
  - Accelerations (angular and along axes)
  - Speeds (vertical and longitudinal)
  - N1.



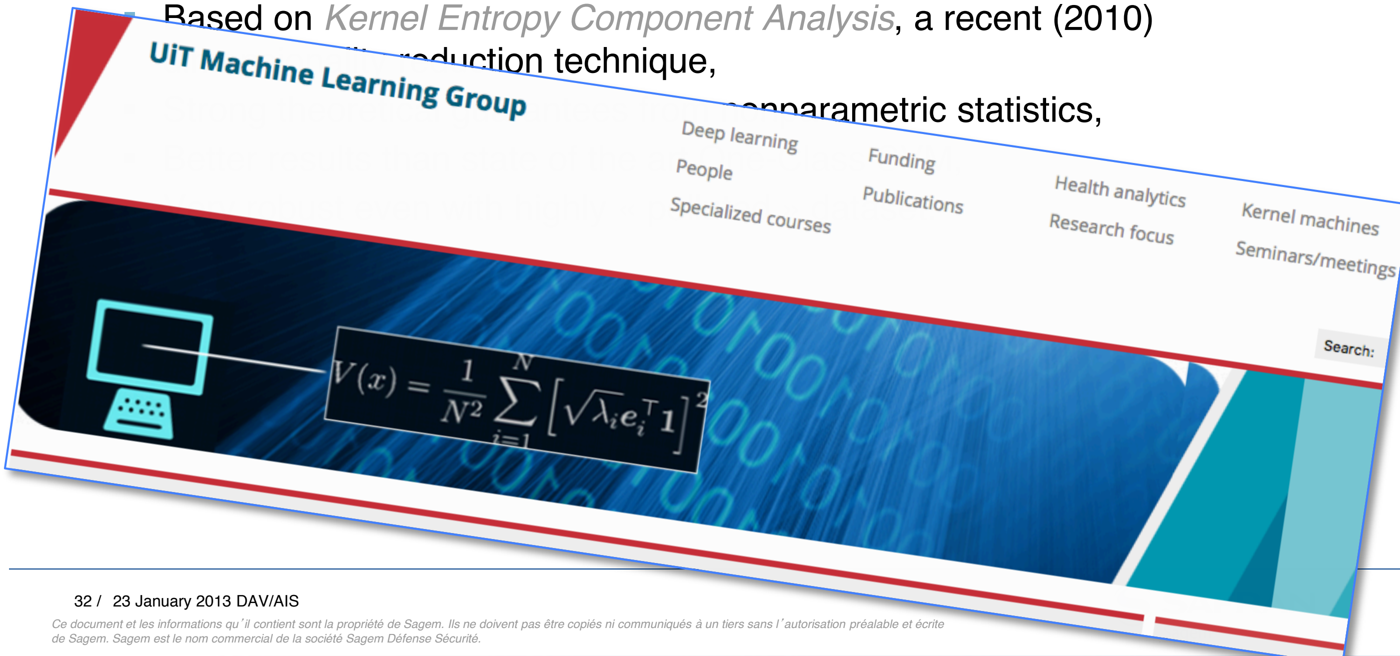
A word on the mathematics

➔ We have used our own detection method:

- Based on *Kernel Entropy Component Analysis*, a recent (2010) dimensionality reduction technique,
- Strong theoretical guarantees from nonparametric statistics,
- Better results than state of the art One-Class SVM,
- Very robust even with highly « polluted » dataset.

A word on the mathematics

- We have used our own detection method:
  - Based on *Kernel Entropy Component Analysis*, a recent (2010)
    ...reduction technique,
    ...nonparametric statistics,

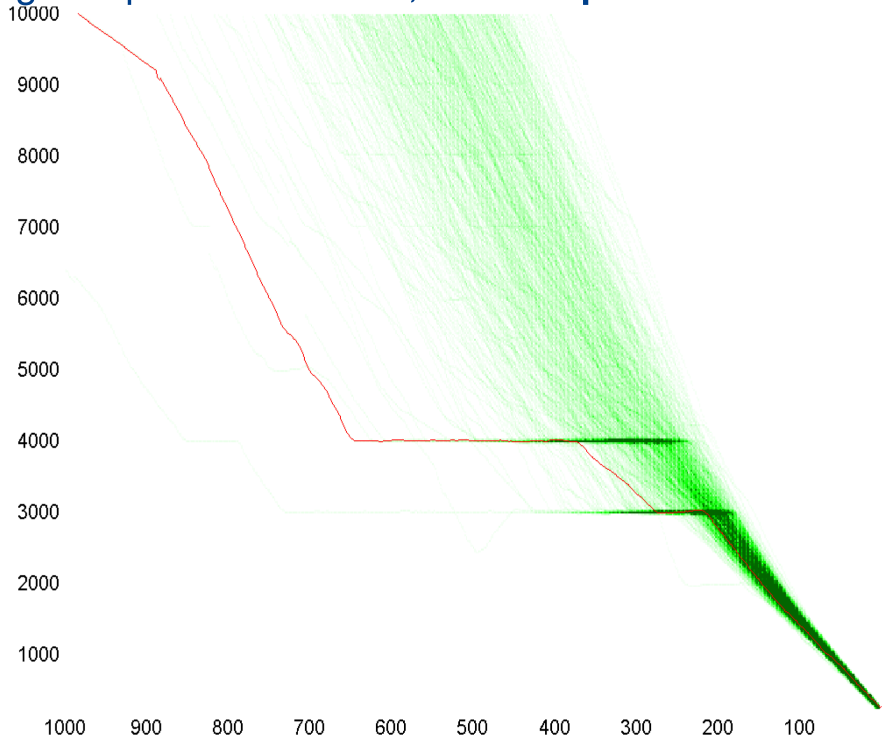


R. Jenssen, Kernel entropy component analysis, IEEE Trans. Pattern Analysis and Machine Intelligence, 2010
R. Jenssen, Entropy-relevant dimensions in kernel feature space, IEEE Signal Processing Magazine, 2013
R. Jenssen et al., Kernel maximum entropy data transformation, NIPS 2007

Example of atypical flights

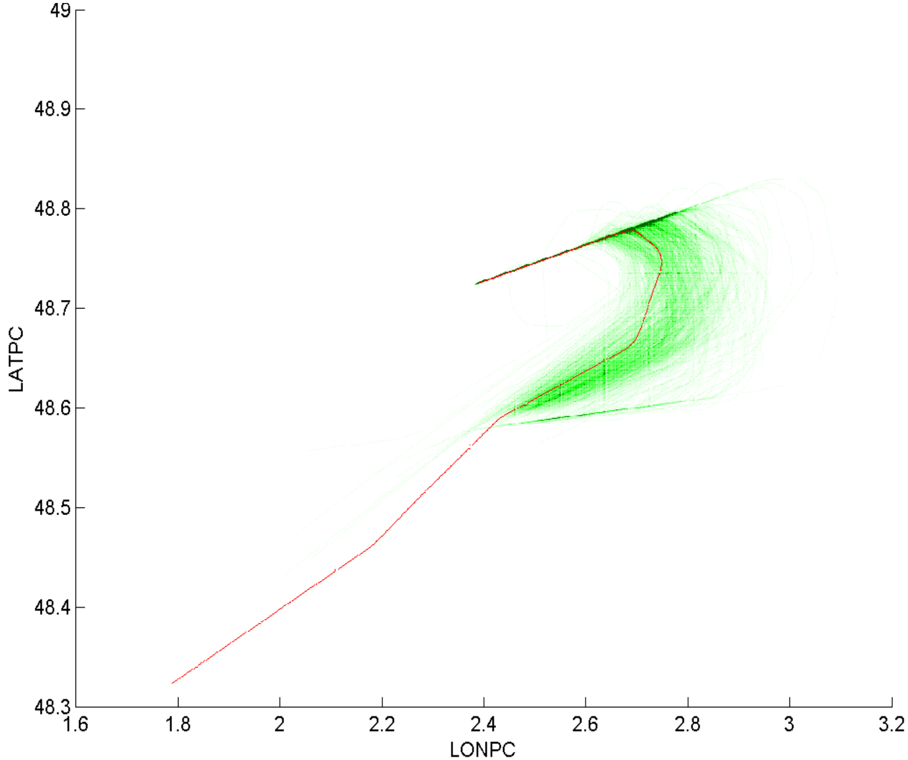
UiT
THE ARCTIC
UNIVERSITY
OF NORWAY
MACHINE LEARNING
GROUP

- Atypical flight 2: pvalue = 0.0004, **altitude plot**



Example of atypical flights

➔ Atypical flight 2: pvalue = 0.0004, **trajectory plot**



Example of atypical flights

UiT THE ARCTIC UNIVERSITY OF NORWAY MACHINE LEARNING GROUP

➔ Atypical flight 2: pvalue = 0.0004

- Classical analysis:
  - **No event detected** with the classical analysis.
- Diagnostic:
  - Meteo: thunder; cumulonimbus clouds, towering cumulus clouds observed
  - Meteorological constraints: the pilot had to lower his altitude to avoid the cumulonimbus cloud.

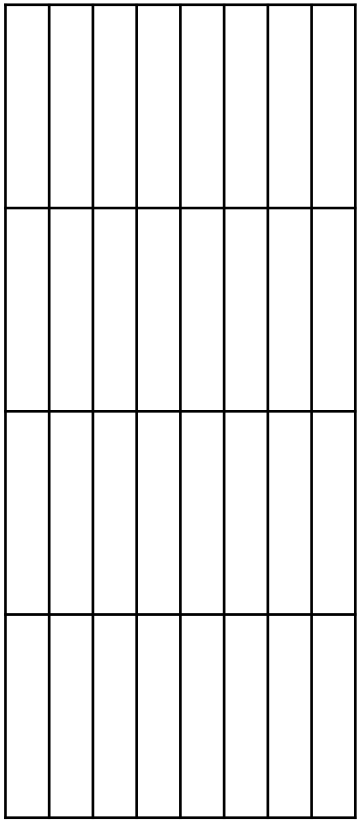
Results

➔ Of the 721 flights, **35 are detected**: each flight is given a **pvalue:**

- « A pvalue is the probability that, **under normal conditions**, a flight at least as extreme could occur by chance alone »

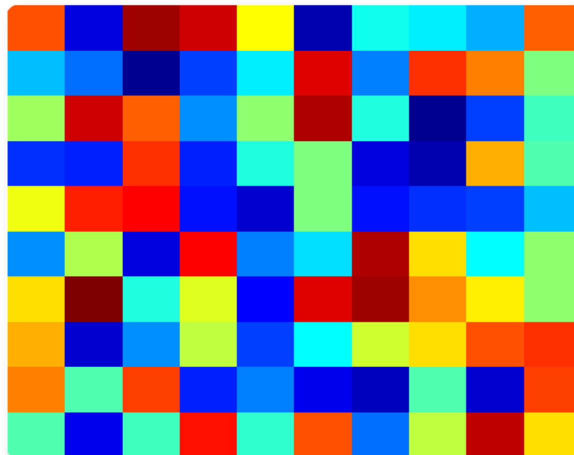
- A flight with a pvalue<0.01 is considered very likely to be an atypical flight,
- A flight with a pvalue<0.001 is considered extremely likely to be an atypical flight.

Data



k(x,z)

$\mathbf{K}$



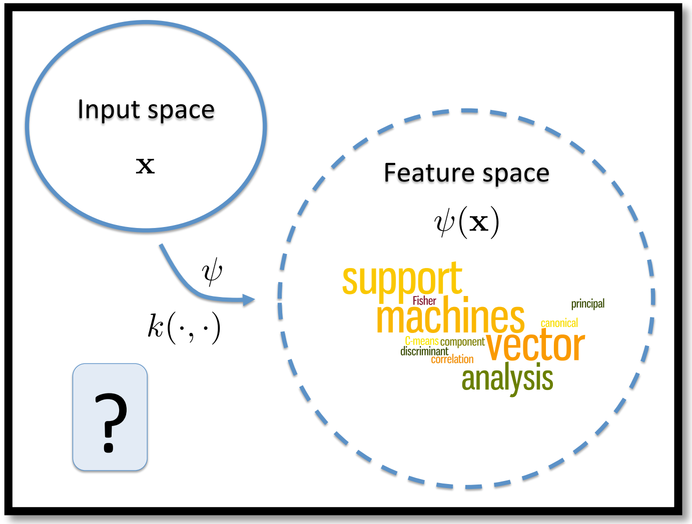
Kernel matrix
(Affinity matrix)
(Covariance matrix)

ML

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

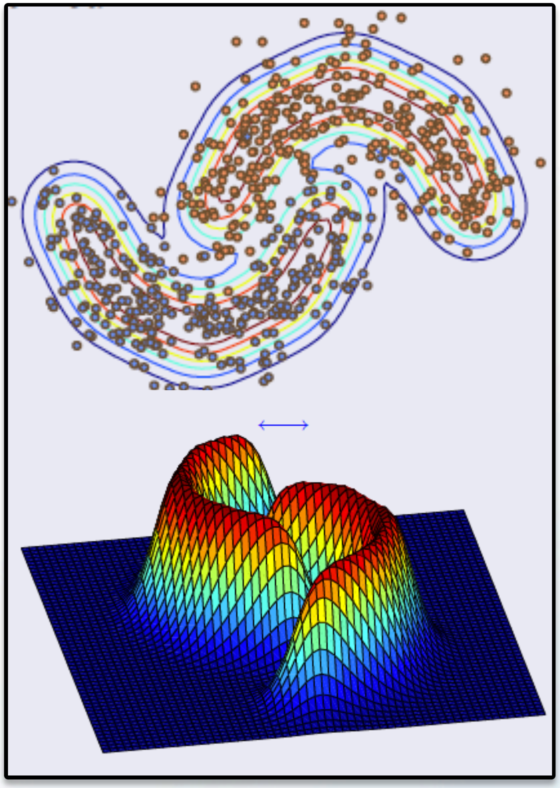
Function for

- Classification
- Prediction
- Embedding

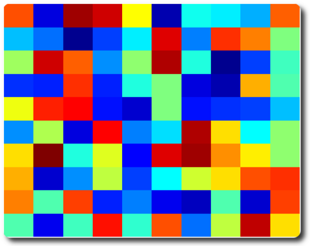


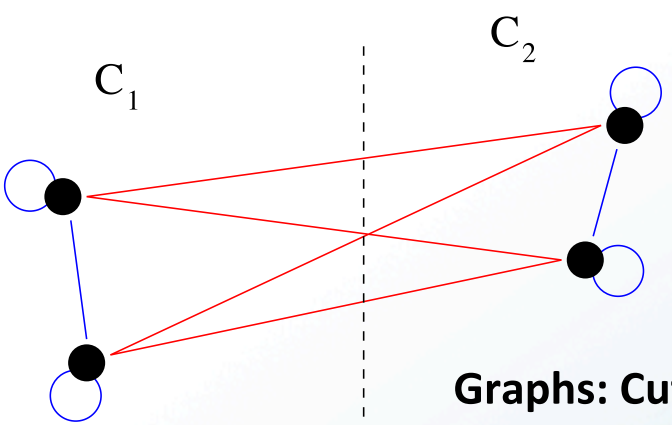
Kernel methods:
Support vector machine (SVM)
Kernel PCA ++

Information theoretic:
Kernel ECA



$\mathbf{K}$





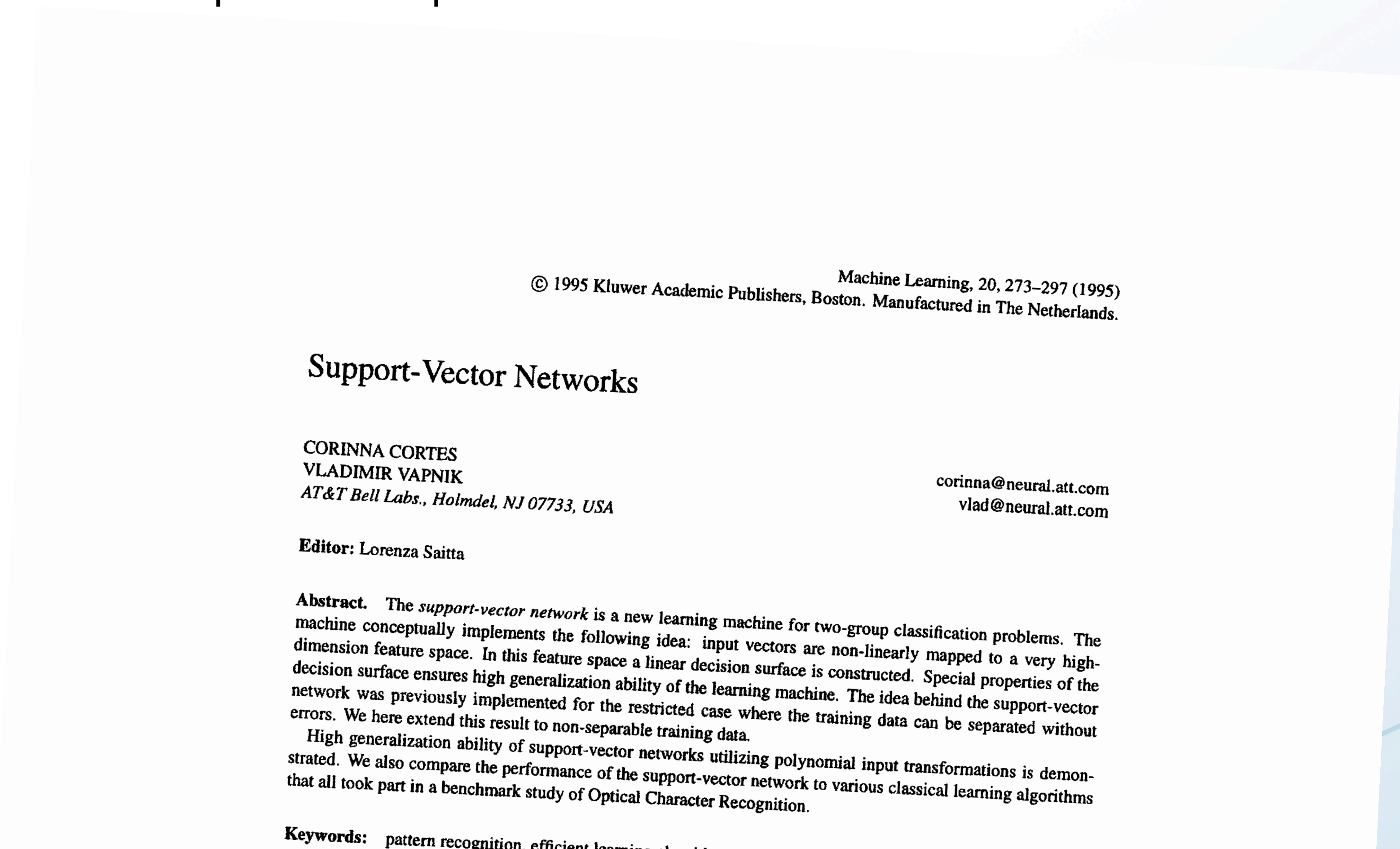
Graphs: Cuts, random walks

Bayesian:
Gaussian processes

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

Kernel methods: SVM

- One of the most used classifiers over the last 10-15 years
- Convex optimization problem



Machine Learning, 20, 273–297 (1995)

Support-Vector Networks

CORINNA CORTES corinna@neural.att.com
VLADIMIR VAPNIK vlad@neural.att.com
AT&T Bell Labs., Holmdel, NJ 07733, USA

Editor: Lorenza Saitta

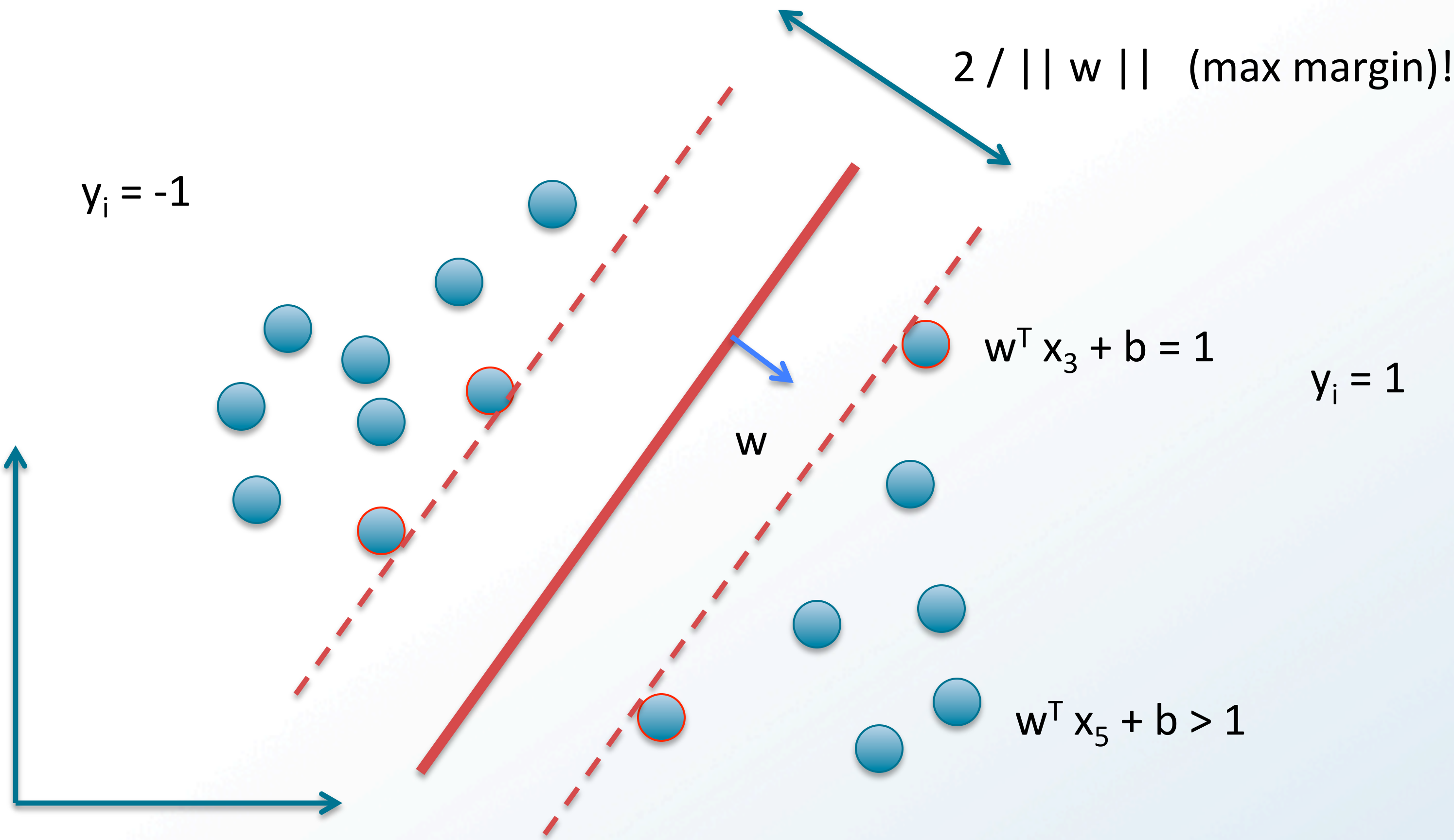
Abstract. The *support-vector network* is a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensures high generalization ability of the learning machine. The idea behind the support-vector network was previously implemented for the restricted case where the training data can be separated without errors. We here extend this result to non-separable training data.

High generalization ability of support-vector networks utilizing polynomial input transformations is demonstrated. We also compare the performance of the support-vector network to various classical learning algorithms that all took part in a benchmark study of Optical Character Recognition.

Keywords: pattern recognition, efficient

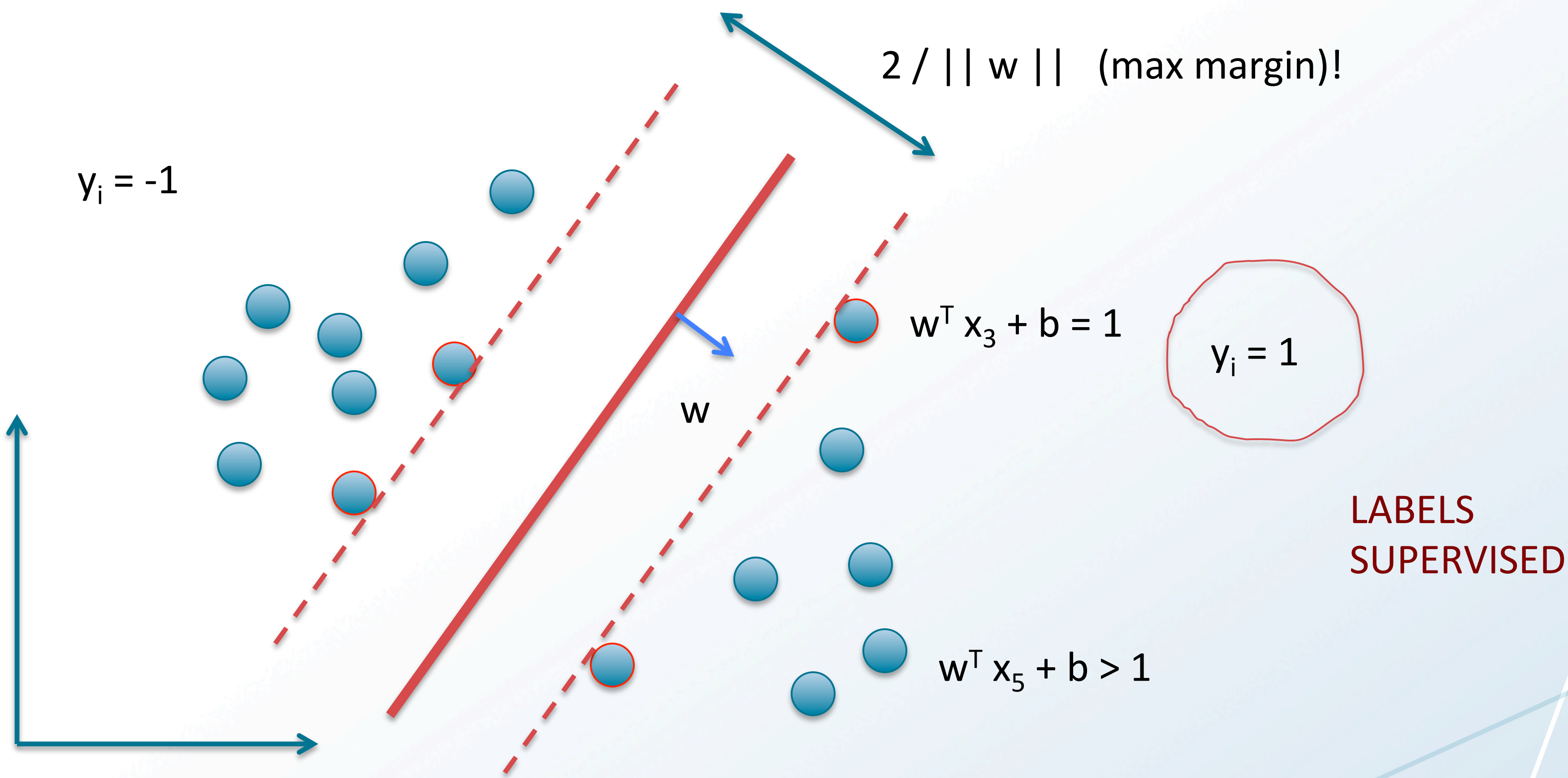
SVM

- SVM: Create a classifier that puts a linear decision boundary “midway” between the classes!



SVM

- SVM: Create a classifier that puts a linear decision boundary “midway” between the classes!



SVM

- Want to solve:

$$\min_{\mathbf{w},b} ||\mathbf{w}||^2$$

such that

for $y_i = 1$: $\mathbf{w}^\top \mathbf{x}_i + b \geq 1$

for $y_i = -1$: $\mathbf{w}^\top \mathbf{x}_i + b \leq -1$

- This is a constrained optimization problem → Need to know Lagrange optimization theory!

R. Jenssen, M. Kloft, A. Zien, S. Sonnenburg and K.-R Muller, "A scatter-based prototype framework and multi-class extension of support vector machines," PLoS ONE, 2012.

SVM

- Lagrange multipliers only active for constraints corresponding to data points "on the margin": Support vectors (SV)

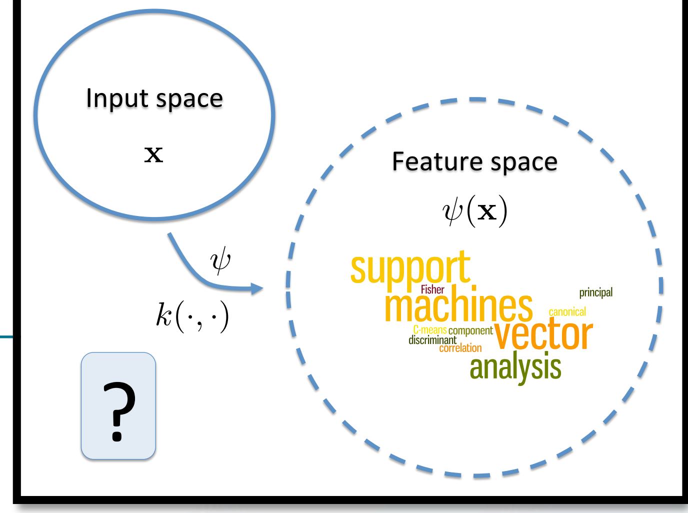
- At solution: $\mathbf{w} = \sum_{\mathbf{x}_i \in SV} \lambda_i y_i \mathbf{x}_i$

- Insert into primal, get dual

$$\max_{\boldsymbol{\lambda}} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \underbrace{\mathbf{x}_i^\top \mathbf{x}_j}_{\text{Wow! Inner-product}} \right)$$

Quadprog

SVM



- Inner-product dependency is key.
- Certain functions can realize $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle$
- Doing this enables the SVM to be trained in *feature space* yielding a <u>nonlinear</u> classifier since the mapping is nonlinear.
- How to classify unknown **x**? Check:

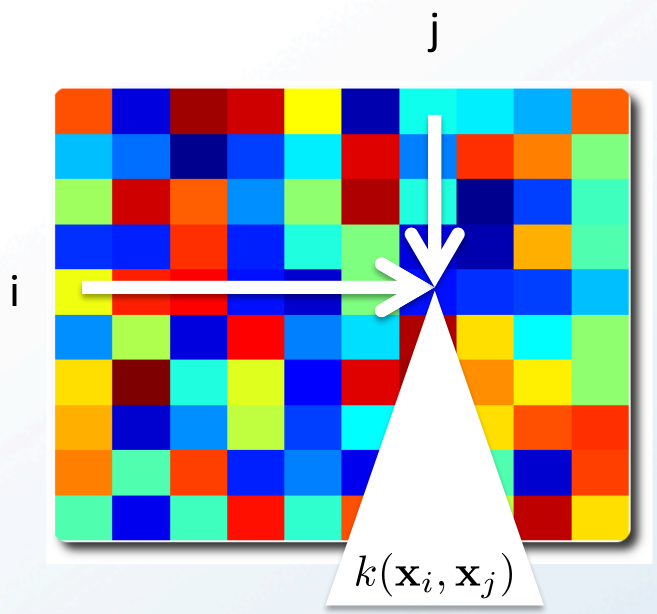
$$\mathbf{w}^\top \mathbf{x} + b = \sum_{\mathbf{x}_i \in SV} \lambda_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

Kernel matrix

- What functions k(.,.) can be used?
- Any function that makes

$$\mathbf{K}$$

Positive semidefinite

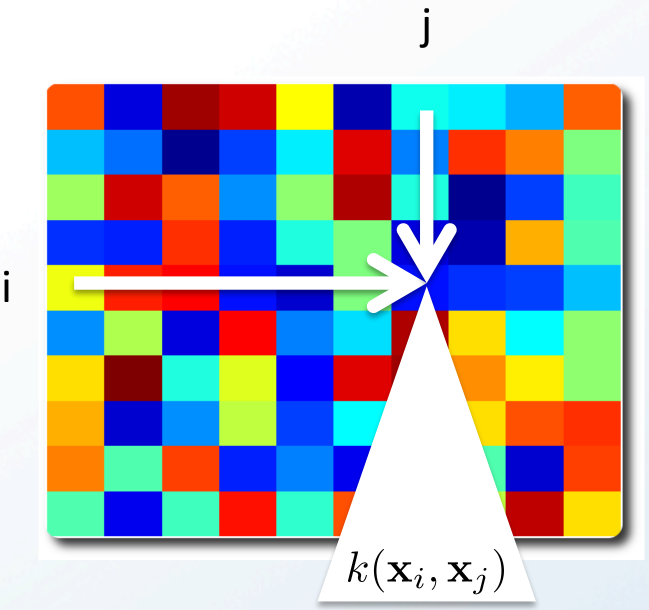


Kernel matrix

- What functions k(.,.) can be used?
- Any function that makes

$$\mathbf{K}$$

Positive semidefinite



How to check it?

$$\mathbf{K}\mathbf{e}_i = \delta_i \mathbf{e}_i$$

Kernel function

- Kernels are functions that return inner products between the images of data points in some space.

- By replacing inner products with kernels in linear algorithms, we obtain very flexible representations

- Choosing k is equivalent to choosing the embedding map

- Very often

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$$

Note: Similarity measure, affinity

Kernel methods

Nonlinear

- Kernel SVM
- Kernel Ridge regression
- Kernel Canonical correlation analysis
- Kernel Fisher discriminant analysis
- Kernel K-means
- Kernel PCA
- Etc (inner-product)

More important than nonlinearity

- Kernels can be defined on general data types
- Classical algorithms can then work on non-vectorial data!
- Sequences
- Trees
- Graphs
- Kernels over pdfs
- Etc

Kernel methods

- Now quite standard and many libraries exist
- LibSVM
- Kernlab
- Shogun
- Weka
- Matlab
  - Statistics and Machine Learning Toolbox
  - Neural Networks Toolbox

Machine learning in Python

http://scikit-learn.org

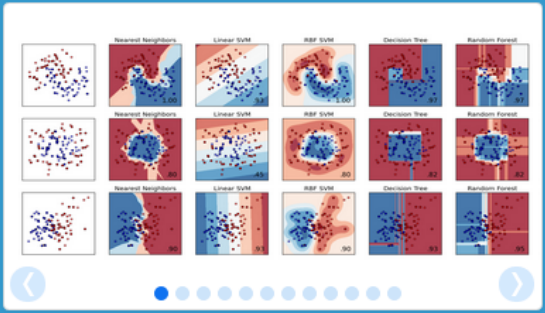
UiT THE ARCTIC UNIVERSITY OF NORWAY
MACHINE LEARNING GROUP

scikit learn

Home Installation Documentation Examples

Google™ Custom Search Search

Fork me on GitHub



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*. — *Examples*

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*. — *Examples*

NR Norsk Regnesentral NORWEGIAN COMPUTING CENTER

NR (Norwegian Computing Center) project: Detection of seals in aerial images

Courtesy Arnt-Børre Salberg

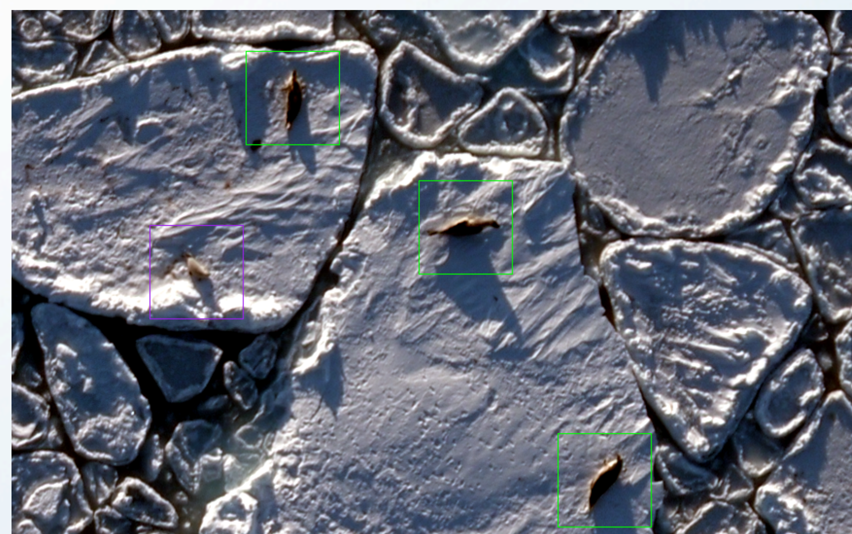
To estimate the population of harp and hooded seals, the Institute of Marine Research counts the number of seals pups regularly.

Aerial photos are aquired, and the animals are counted.

Currently the process is manually and very time-consuming.

Goal

Develop an algorithm that automatically counts the number of harp and hooded seals in aerial images.



Detection of seals using CNNs and SVMs

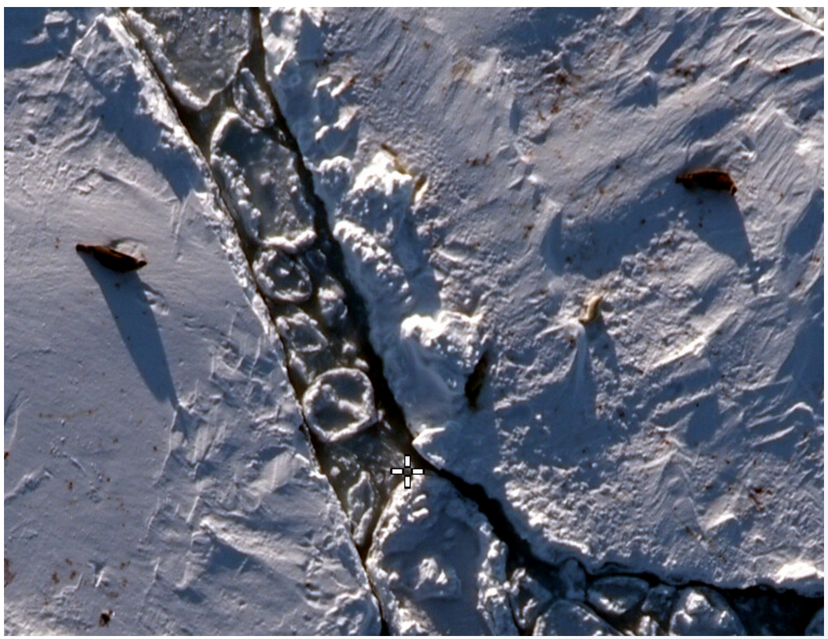
Two step approach:

1. Detection of potential objects. These objects define the set of candidate detections available to the classifier.
2. Feature extraction. This is based on a deep CNN that extracts fixed-length feature vector corresponding to the image patch that covers each potential object.
3. Classification of potential objects. The classifier is based on a SVM classifier that classifies the feature vectors into the desired classes.

Detection of seals using CNNs and SVMs

Detection of potential objects

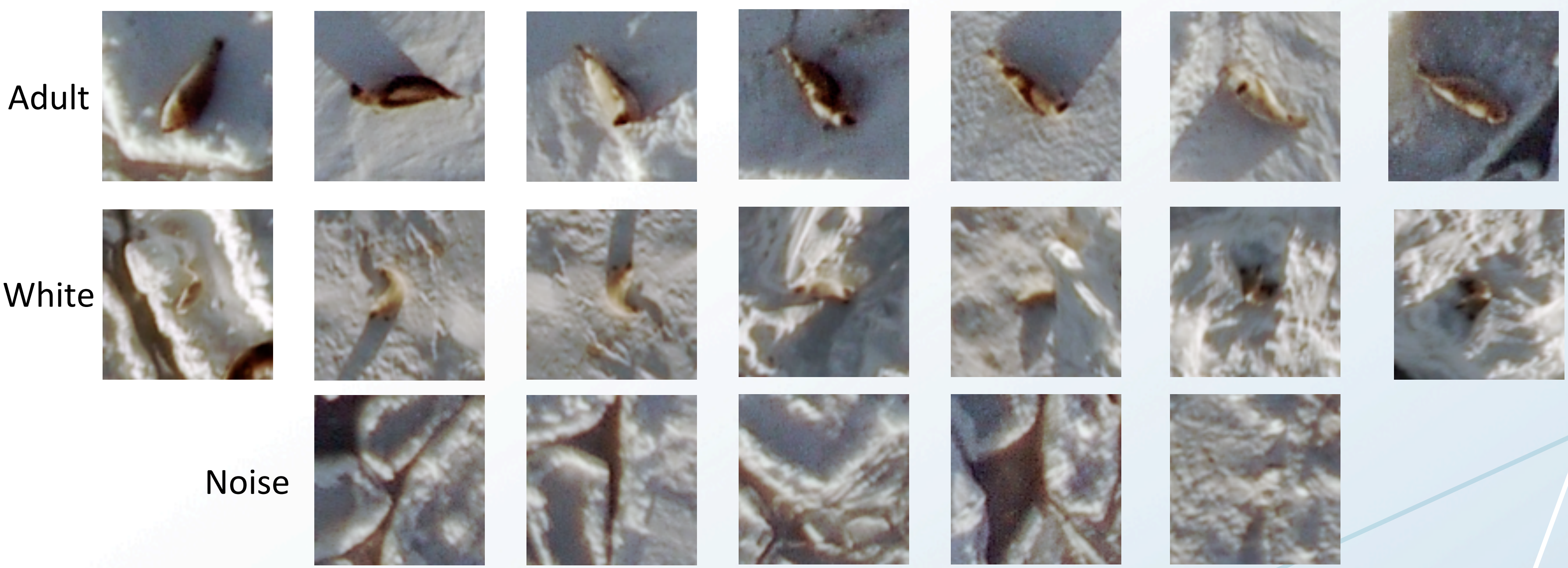
Potential objects were detected using the constrained energy minimization (CEM) classification methodology.



Detection of seals using CNNs and SVMs

Feature extraction

- A 97x97 sub-image covering each detection is extracted.
- This is then rescaled to 256x256 and sent into the CNN (ImageNet 2012 winner network).
- The 4096 element CNN feature vector of each sub-image is stored.



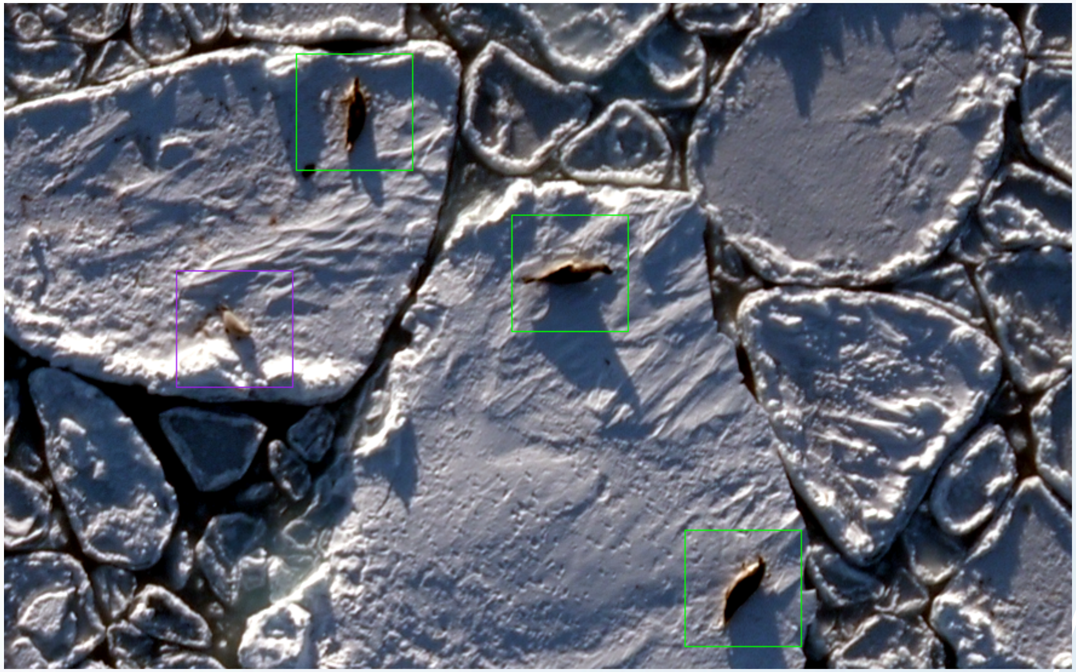
Detection of seals using CNNs and SVMs

Classification

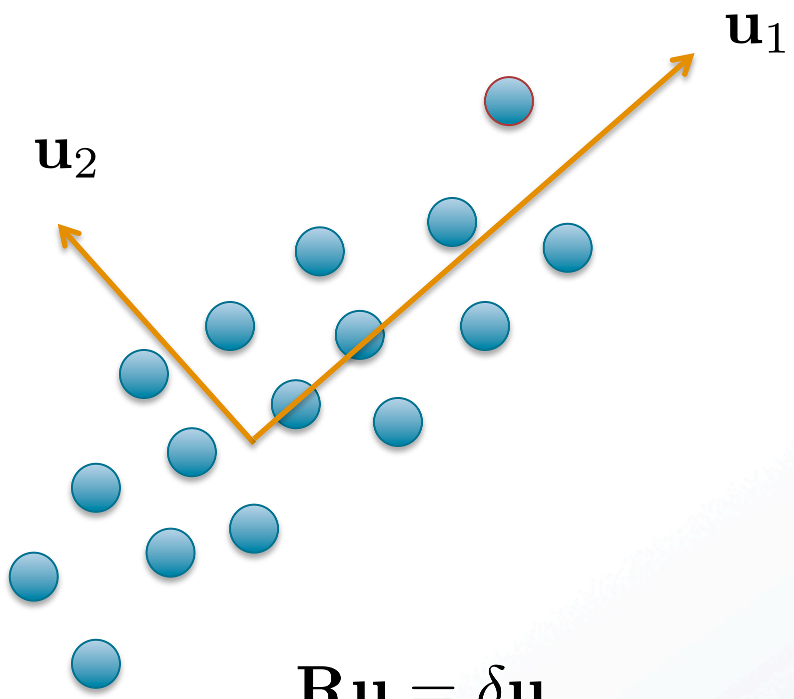
- Classes: Adult, pup and non-seal
- A SVM is trained on the 4096 element feature vectors using R library

Total Accuracy > **95%**

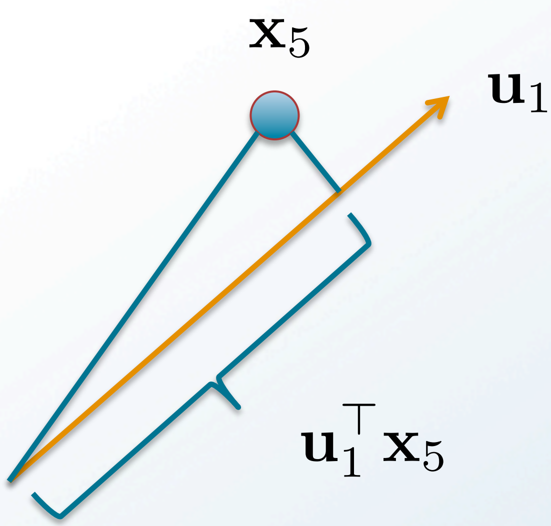
(20-fold cross-validation on training data)



Going spectral! Kernel PCA



UNSUPERVISED !!

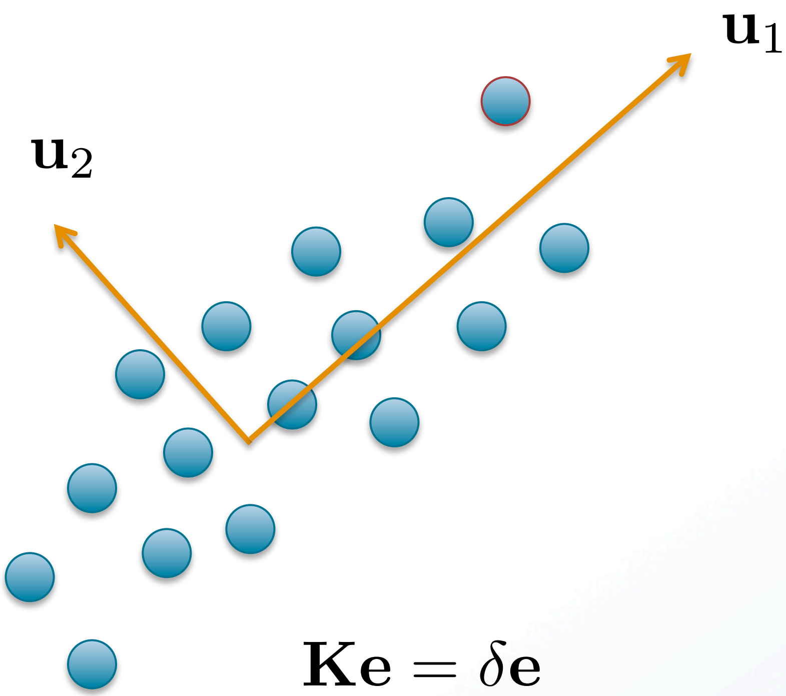


$$\mathbf{R}\mathbf{u} = \delta \mathbf{u}$$

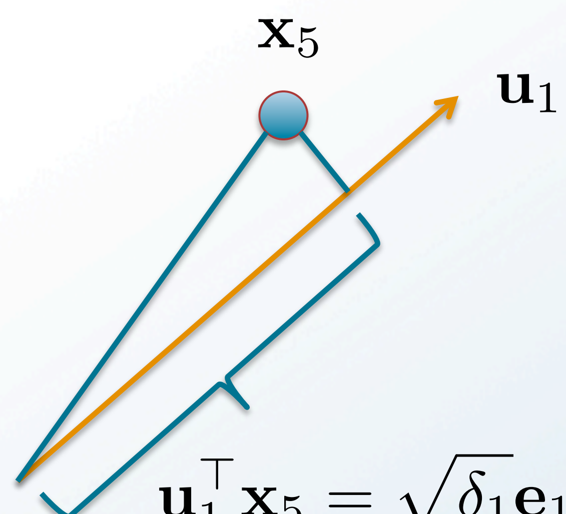
Correlation/Covariance matrix

Scholkopf, Smola, Muller, Neural Computation, 1998

Kernel PCA



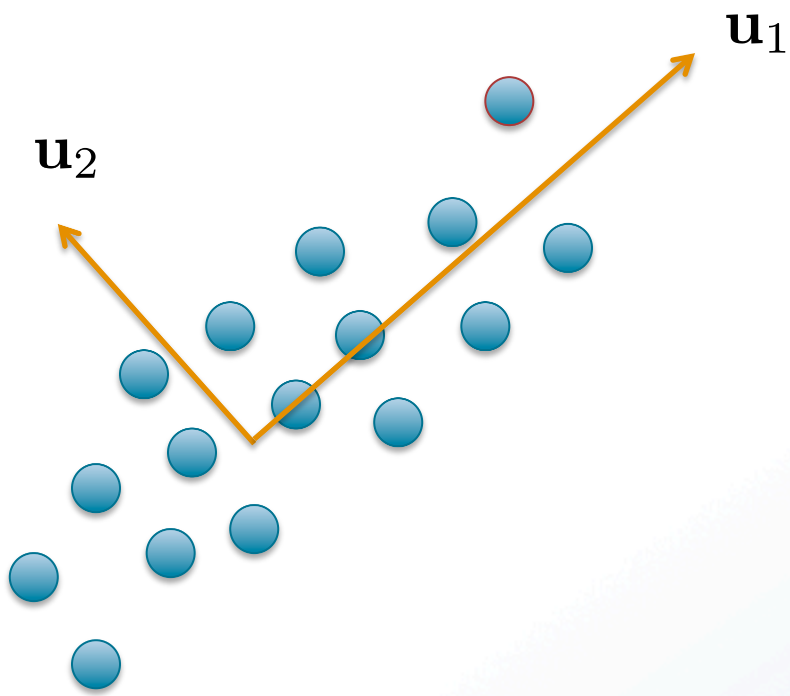
$$\mathbf{K}\mathbf{e} = \delta\mathbf{e}$$

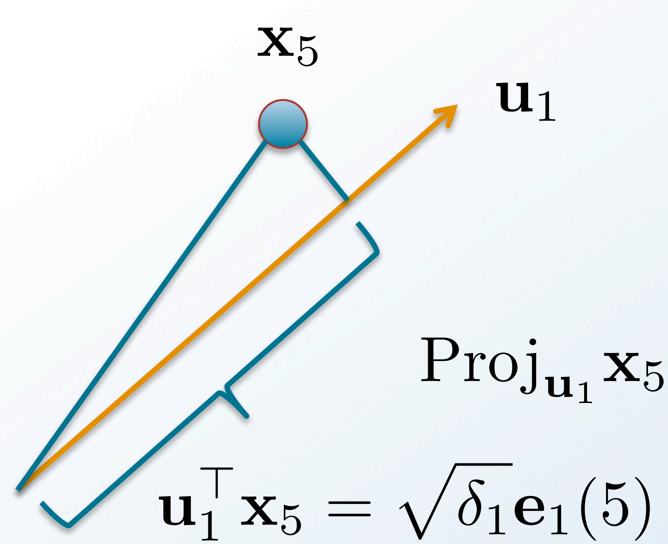


$$\mathbf{u}_1^\top \mathbf{x}_5 = \sqrt{\delta_1}\mathbf{e}_1(5) \quad !$$

Inner-product matrix

Kernel PCA

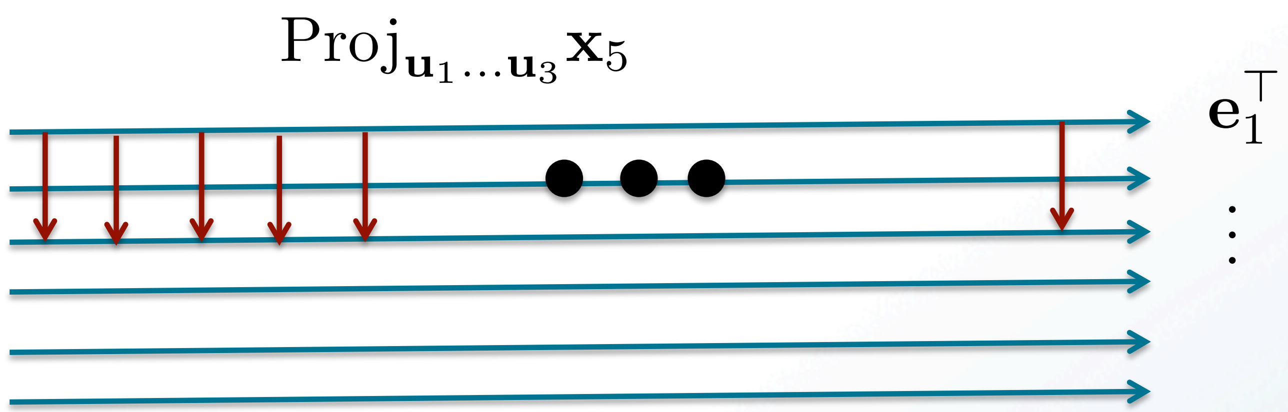




Wow!

$$\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_N] \qquad \mathrm{Proj}_{\mathbf{u}_1}\mathbf{X} = \sqrt{\delta_1}\mathbf{e}_1$$

Kernel PCA

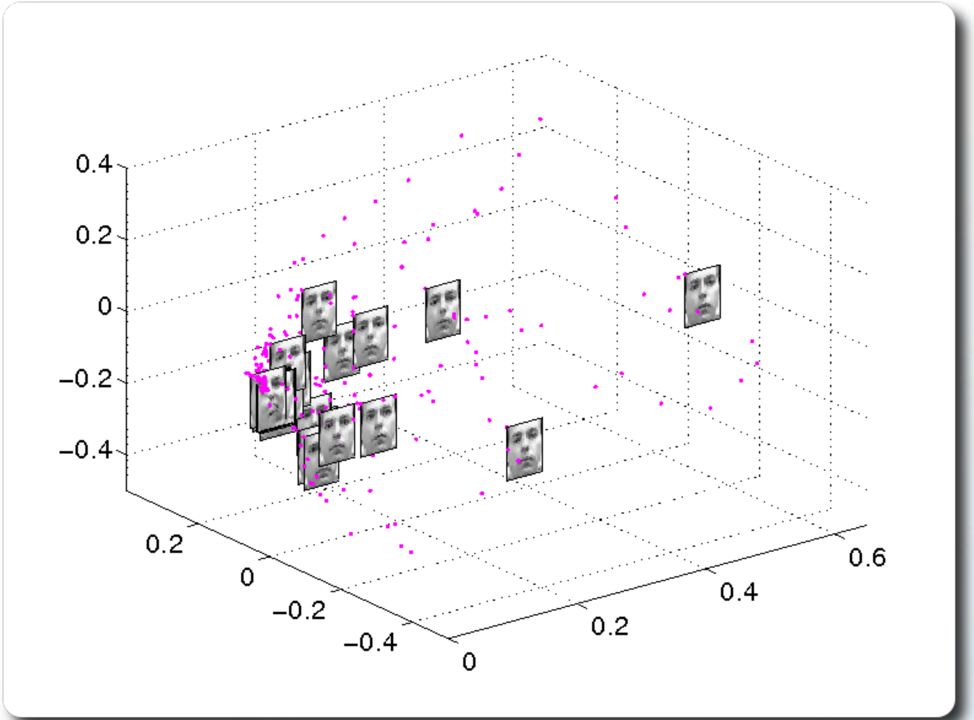


Empirical kernel map!

Creates a new representation

New data set of possibly lower dimensionality

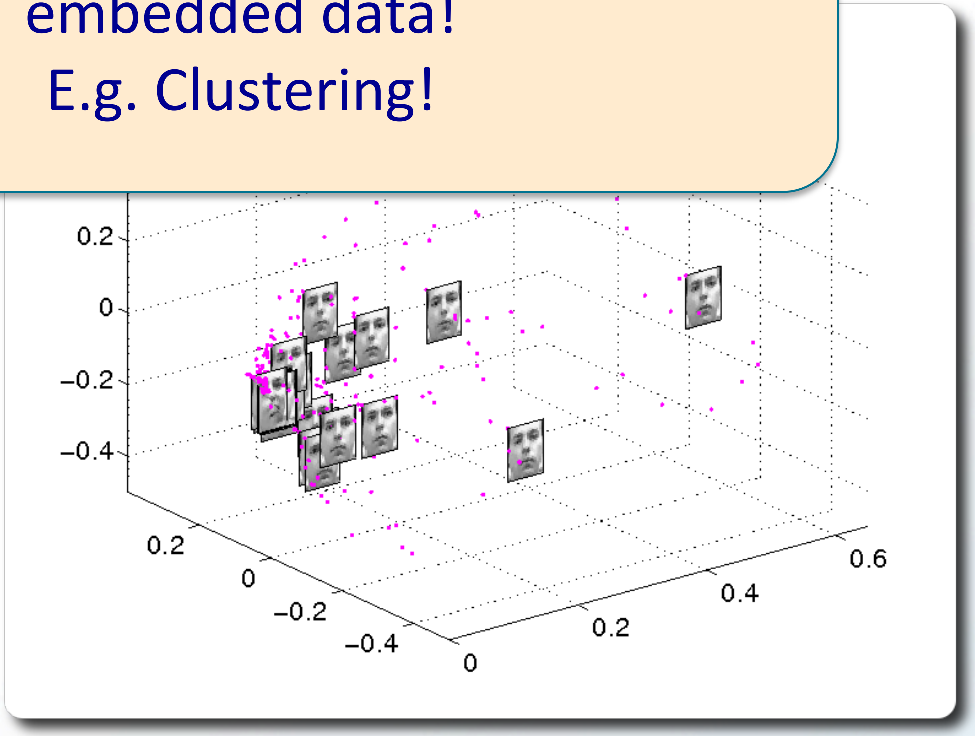
Kernel PCA



Frey faces example

Kernel PCA: Looking ahead

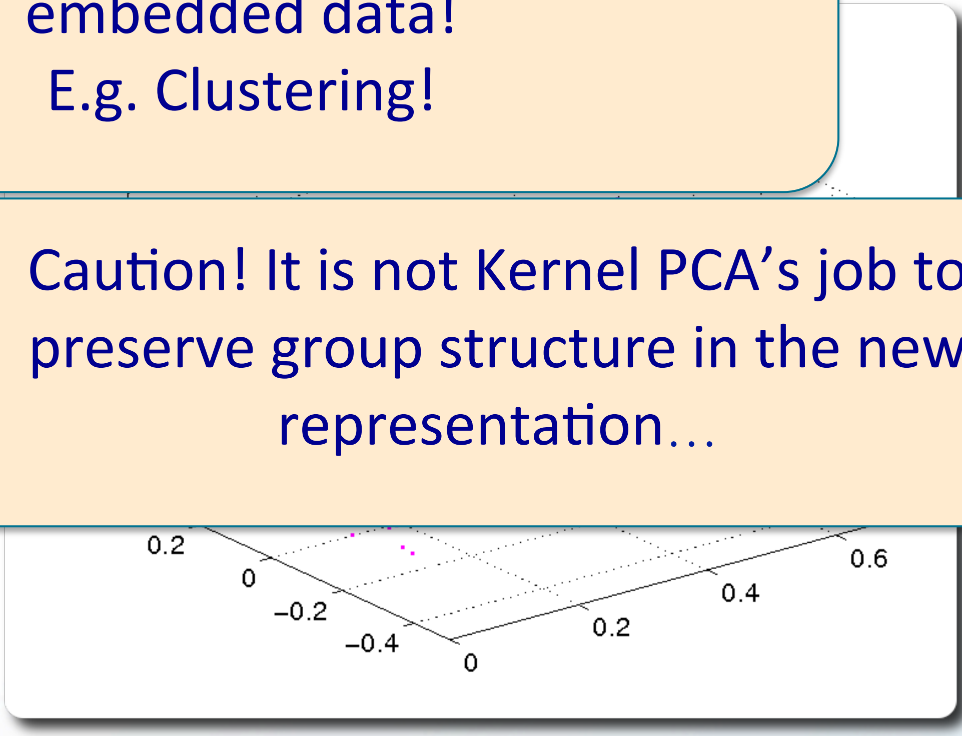
You can do whatever you want on these embedded data!
E.g. Clustering!



Frey faces example

Kernel PCA: Looking ahead

You can do whatever you want on these embedded data!
E.g. Clustering!

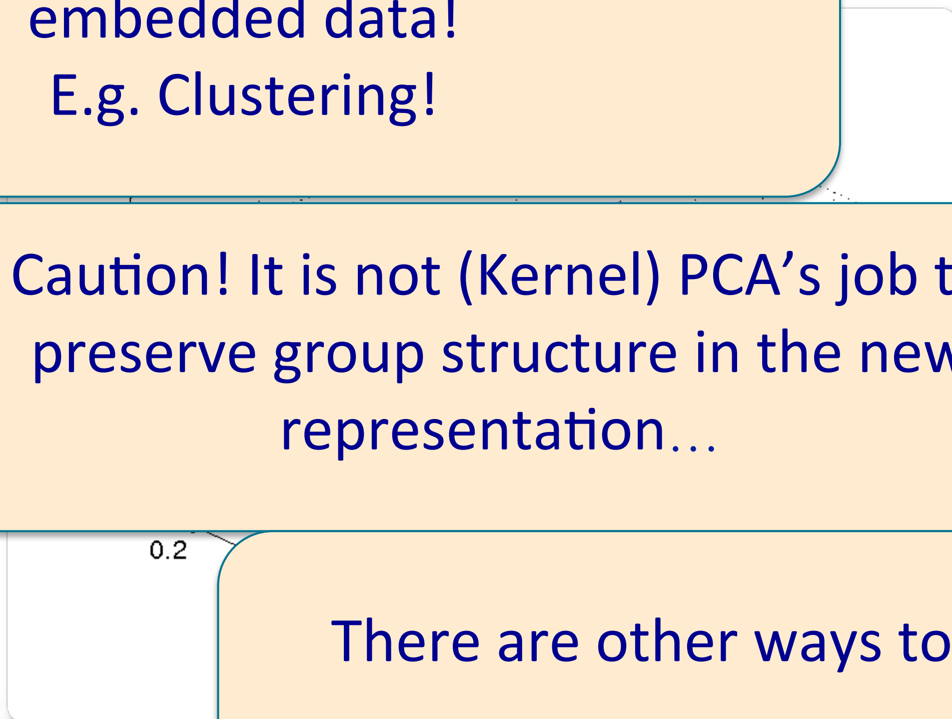


Caution! It is not Kernel PCA's job to preserve group structure in the new representation…

Frey faces example

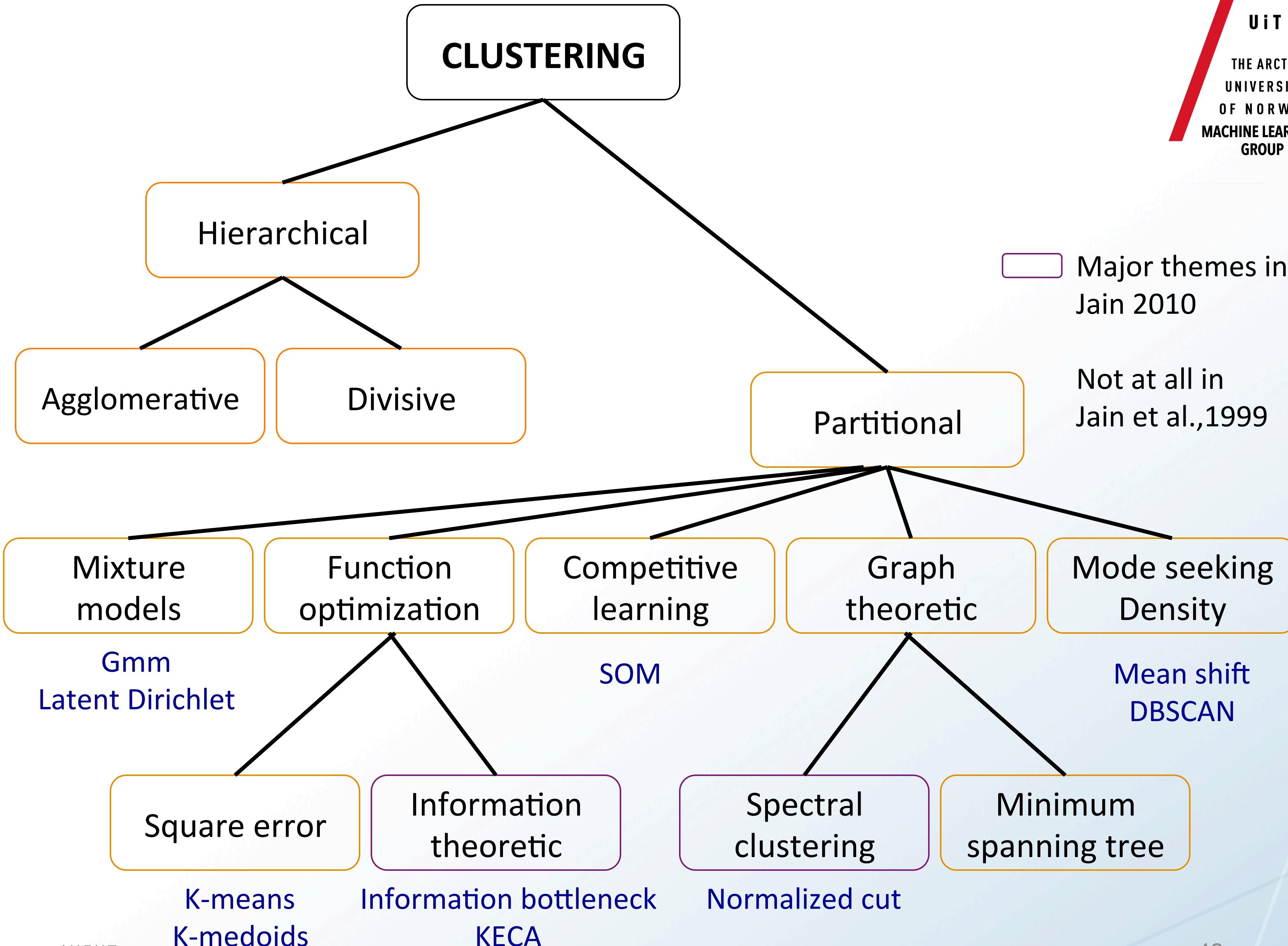
Kernel PCA: Looking ahead

You can do whatever you want on these embedded data!
E.g. Clustering!



Caution! It is not (Kernel) PCA's job to preserve group structure in the new representation…

There are other ways to embed data!


CLUSTERING
Hierarchical
Agglomerative
Divisive
Partitional
Major themes in Jain 2010
Not at all in Jain et al.,1999
Mixture models
Function optimization
Competitive learning
Graph theoretic
Mode seeking Density
Gmm
Latent Dirichlet
SOM
Mean shift
DBSCAN
Square error
Information theoretic
Spectral clustering
Minimum spanning tree
K-means
K-medoids
Information bottleneck
KECA
Normalized cut

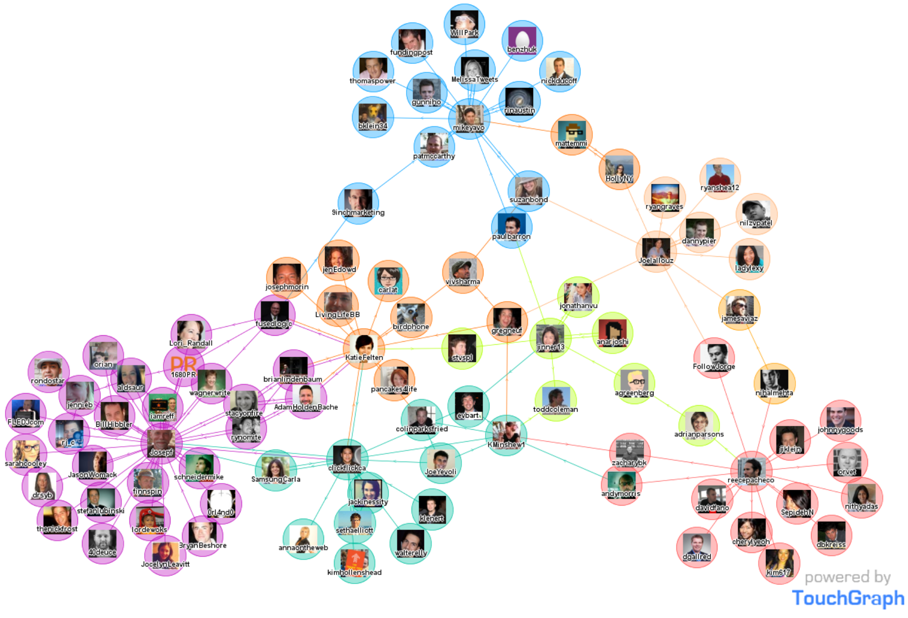
Spectral clustering

- Treats clustering as a graph partitioning problem
- Makes no assumptions on the form of clusters
- Cluster points using eigenvalues and eigenvectors of matrices derived from data
- Embed or map data to a low-dimensional space and do the clustering there (e.g. by k-means)

Dominant direction in modern clustering!

Y. Han, M. Filippone, Mini-batch spectral clustering, 2016
C. Boutsidis et al., Spectral clustering via the power method − provably, ICML 2015
E. Izquierdo-Verdiguier, R. Jenssen et al., Spectral clustering with the probabilistic cluster kernel, Neurocomputing, 2015

Graphs



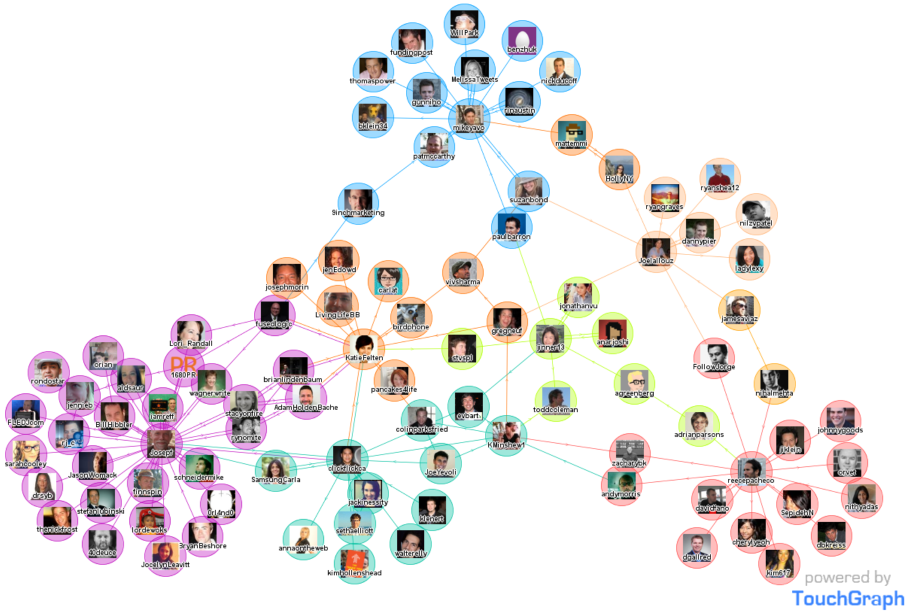
- Natural graph structure very common
  - Web pages, links (PageRank)
  - Protein structures
  - Citation graphs

- Other data sets can be easily transformed into <u>similarity</u>, or <u>affinity</u>, graphs
  - Affinity encode local structure in data

- Represents data by <u>pairwise</u> relationships

- A positive and symmetric matrix is equivalent to a graph

Graphs



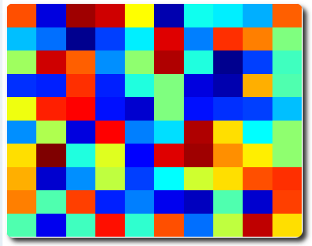
- Natural graph structure very common
  - Web pages, links (PageRank)
  - Protein structures
  - Citation graphs

- Other data sets can be easily transformed into <u>similarity</u>, or <u>affinity</u>, graphs
  - Affinity encode local structure in data

- Represents data by <u>pairwise</u> relationships

- A positive and symmetric matrix is equivalent to a graph

$\mathbf{K}$



To give a hint

Segmentation of cDNA Microarray Images using Parallel Spectral Clustering

Sandrine Mousset[a], Ronan Guivarch[b], Joseph Noailles[b], Daniel Ruiz[b]

[a] University of Toulouse - UPS - IRIT,

[b] University of Toulouse - INPT(ENSEEIHT) - IRIT.

KEYWORD

Spectral Clustering

Domain Decomposition

Image Segmentation

Microarray Image

ABSTRACT

Microarray technology generates large amounts of expression level of genes to be analyzed simultaneously. This analysis implies microarray image segmentation to extract the quantitative information from spots. Spectral clustering is one of the most relevant unsupervised methods able to gather data without a priori information on shapes or locality. We propose and test on microarray images a parallel strategy for the
tion with a criterion to

1 Introduction

Image segmentation in microarray analysis is a crucial step to extract quantitative information from the spots [RUEDA, 2009], [USLAN, 2010], [CHEN, 2011]. Clustering methods are used to separate the pixels that belong to the spot from the pixels of the background and noise. Among these, some methods imply some restrictive assumptions on the shapes of the spots [YANG, 2001], [RUEDA, 2005]. Due to the fact that the most of spots in a microarray image have irregular-shapes, the clustering based-method should be adaptive to arbitrary shape of spots such as fuzzy clustering [GLEZ-PENA, 2009], but it should also not depend on many input parameters. To address these requirements, the spectral methods, and in particular the spectral clustering algorithm introduced by Ng-Jordan-Weiss [NG, 2002], are useful to partition subsets of data with no a priori on the shapes. Spectral clustering exploits eigenvectors of a Gaussian affinity matrix in order to define a low dimensional space in which data points can be easily clustered. But when very large data sets are considered, the

Superpixel Segmentation using Linear Spectral Clustering

Zhengqin Li Jiansheng Chen

Department of Electronic Engineering, Tsinghua University, Beijing, China

li-zq12@mails.tsinghua.edu.cn jschenthu@mail.tsinghua.edu.cn

Abstract

We present in this paper a superpixel segmentation algorithm called Linear Spectral Clustering (LSC), which produces compact and uniform superpixels with low computational costs. Basically, a normalized cuts formulation of the superpixel segmentation is adopted based on a similarity metric that measures the color similarity and space proximity between image pixels. However, instead of using the traditional eigen-based algorithm, we approximate the similarity metric using a kernel function leading to an explicitly mapping of pixel values ...

Figure 1. Images [13] segmented into 1000/500/200 superpixels using the proposed LSC algorithm.

...g properties of superpixel segmentation are generally de-...rable. First, superpixels should adhere well to the natural ...age boundaries and each superpixel should not overlap ...ith multiple objects. Second, as a preprocessing technique ...r improving efficiency of computer vision tasks, super-...xel segmentation should be of low complexity itself. Last ...t not the least, global image information which is impor-...t for human vision cognition should be considered ap-...propriately. It is critical for a segmentation process to uti-...e the perceptually important non-local clues to group un-...ited image pixels into semantically meaningful regions. ...vertheless, considering global relationship among pixels ...ally lead to substantial increases in computational com-...ity. A typical example is the eigen-based solution to normalized cuts (Ncuts) based superpixel segmentation ...rithm proposed in [17]. As a result, most practical su-...ixel segmentation algorithms, such as [5][21][11], are ...ly based on the analysis of local image information. These methods may fail to correctly segment image ...ns with high intensity variability [8].

...address this issue, we propose a superpixel segmen-... algorithm, *Linear Spectral Clustering* (LSC), which ...ly captures perceptually important global image prop-..., but also runs in linear complexity with high memory ...ency. In LSC, we map each image pixel to a point ...en dimensional feature space in which weighted K-... is applied for segmentation. Non-local information ...licitly preserved due to the equivalence between the ...ted K-means clustering in this ten dimensional feature ...and normalized cuts in the original pixel space. Sim-...ighted K-means clustering in the feature space can be

www.nature.com/scientificreports

SCIENTIFIC REPORTS

OPEN

A multi-similarity spectral clustering method for community detection in dynamic networks

Received: 28 January 2016

Accepted: 15 July 2016

Published: 16 August 2016

Xuanmei Qin[1], Weidi Dai[2], Pengfei Jiao[2], Wenjun Wang[2] & Ning Yuan[3]

Community structure is one of the fundamental characteristics of complex networks. Many methods have been proposed for community detection. However, most of these methods are designed for static networks and are not suitable for dynamic networks that evolve over time. Recently, the evolutionary clustering framework was proposed for clustering dynamic data, and it can also be used for community detection in dynamic networks. In this paper, a multi-similarity spectral (MSSC) method is proposed as an improvement to the former evolutionary clustering method. To detect the community structure in dynamic networks, our method considers the different similarity metrics of networks. First, multiple similarity matrices are constructed for each snapshot of dynamic networks. Then, a dynamic co-training algorithm is proposed by bootstrapping the clustering of different similarity measures. Compared with a number of baseline models, the experimental results show that the proposed MSSC method has better performance on some widely used synthetic and real-world datasets with ground-truth community structure that change over time.

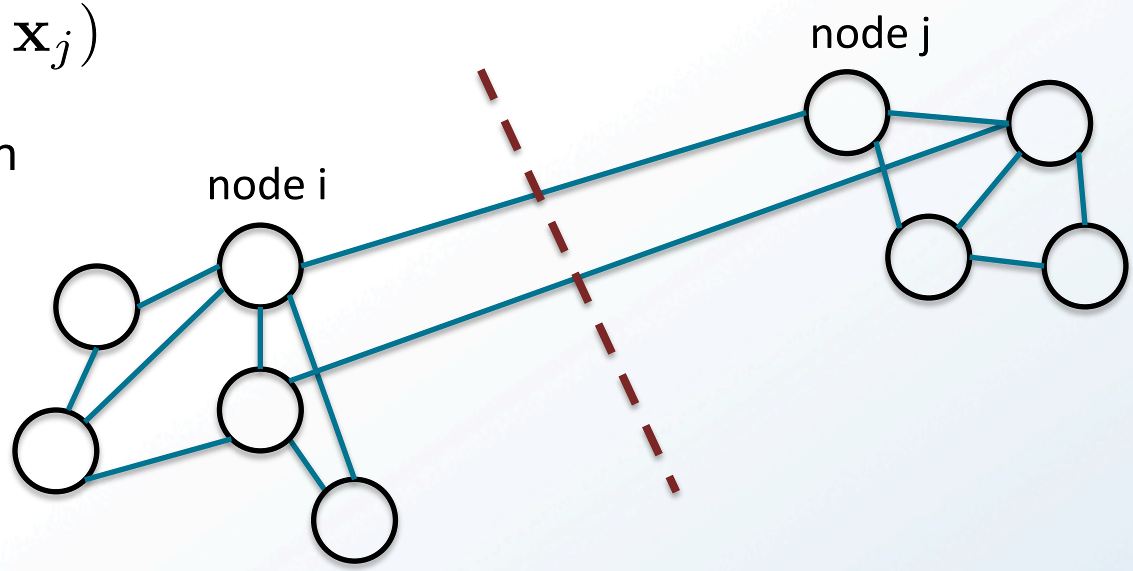
Complex networks have been studied in many domains, such as genomic networks, social networks, communica-

Spectral clustering – graph view

- Given data points $\mathbf{x}_1 \ldots \mathbf{x}_N$ and pairwise affinities

$$k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

- Build similarity graph



- Clustering by finding a cut through the graph
  - Define cut cost function
  - Solve it (find groups)

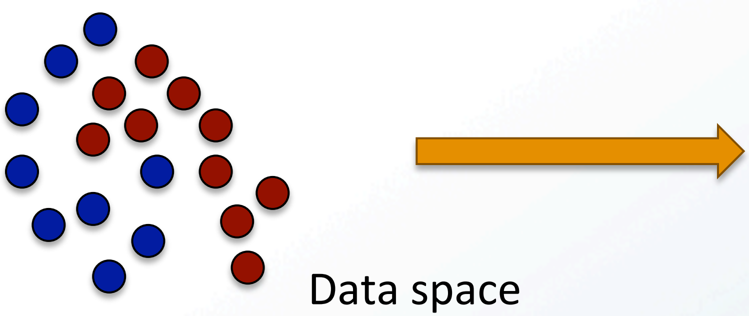
Spectral clustering – embedding view

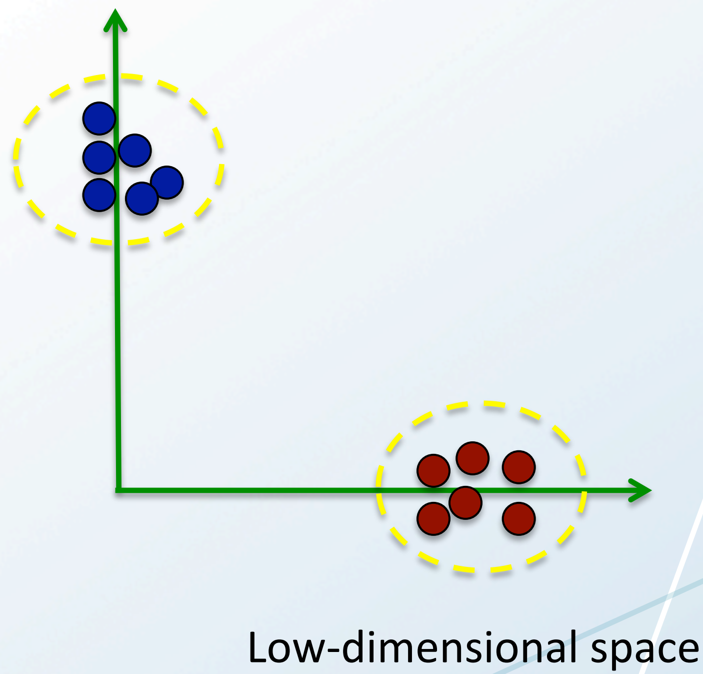
- Given data points $\mathbf{x}_1 \ldots \mathbf{x}_N$ and <u>pairwise affinities</u>

$$k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

- Find a low-dimensional embedding

- Project data points to new space





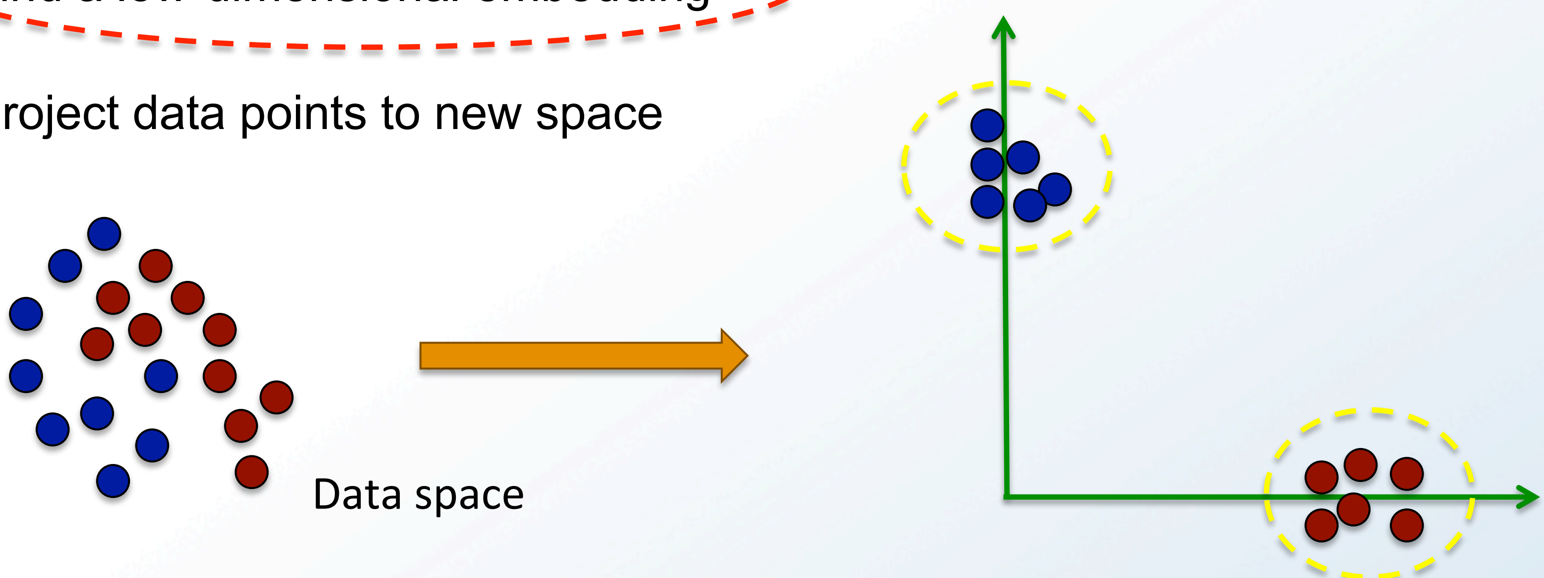
- Cluster using favorite algorithm!

Spectral clustering – embedding view

- Given data points $\mathbf{x}_1 \ldots \mathbf{x}_N$ and <u>pairwise affinities</u>

$$k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

- Find a low-dimensional embedding
- Project data points to new space



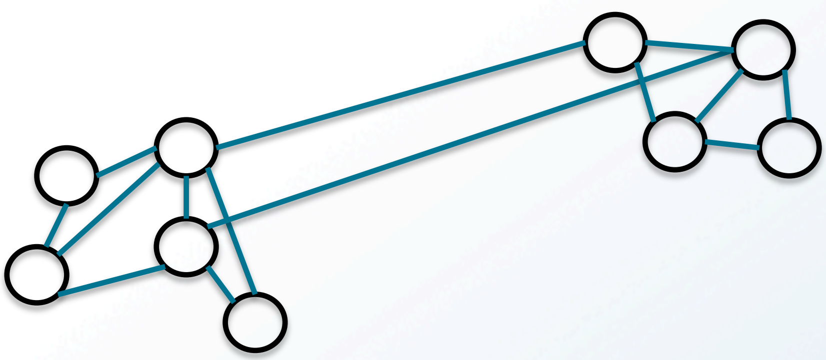
- Cluster using favorite algorithm!

Graph basics

Adjacency matrix **W**

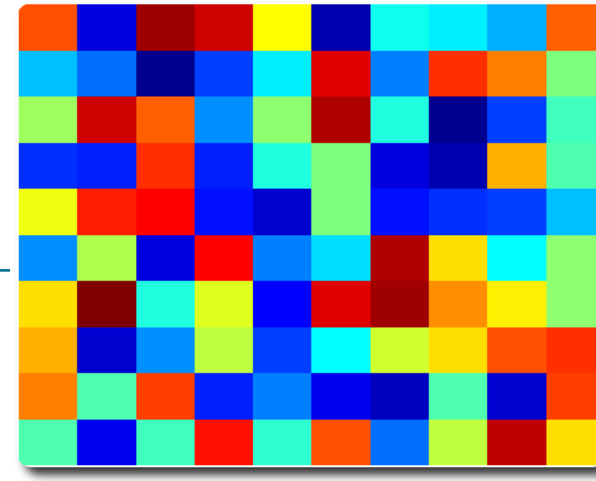
- NxN symmetric and binary
- Rows and columns represent vertices and entries represent presence of edges in the graph

w(i,j) = 1 if i,j are connected
w(i,j) = 0 if i,j are not connected



0	1	0	0	0	1	1	0	1
1	0	1	1	1	0	0	0	0
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	1	0	0	0	1	1	0
1	0	0	0	0	1	0	1	1
0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	1	0	0

Graph basics



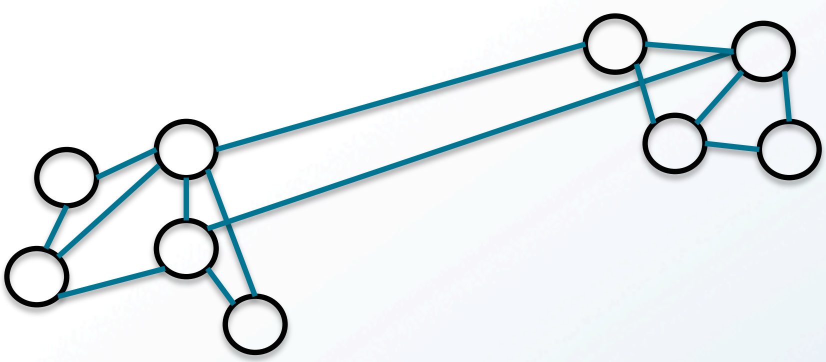
Affinity matrix **K**

- NxN symmetric and positive
- Weighted adjacency matrix

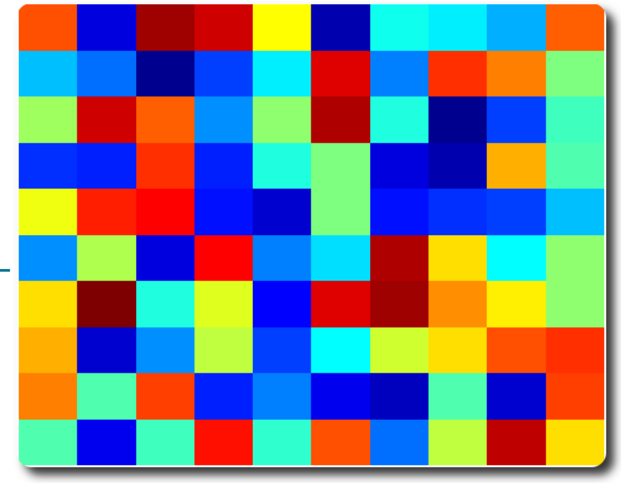
.*

0	1	0	0	0	1	1	0	1
1	0	1	1	1	0	0	0	0
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	1	0	0	0	1	1	0
1	0	0	0	0	1	0	1	1
0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	1	0	0

a(i,j) = k(i,j)   if i,j are connected!
a(i,j) = 0   if i,j are not connected



Graph basics



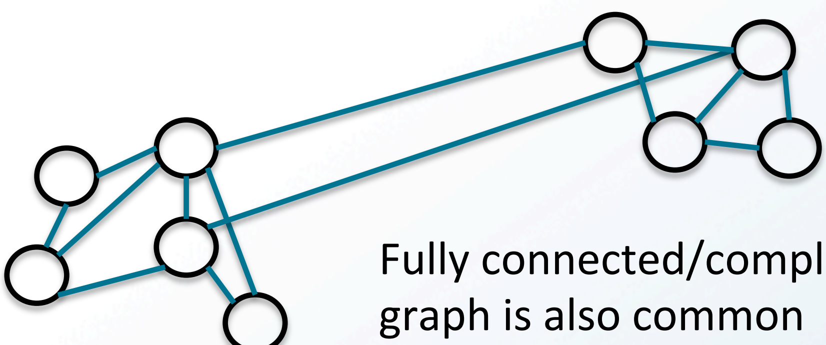
Affinity matrix **K**

- NxN symmetric and positive
- Weighted adjacency matrix

.*

0	1	0	0	0	1	1	0	1
1	0	1	1	1	0	0	0	0
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	1	0	0	0	1	1	0
1	0	0	0	0	1	0	1	1
0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	1	0	0

a(i,j) = k(i,j)   if i,j are connected!
a(i,j) = 0   if i,j are not connected



Fully connected/complete graph is also common

Graph basics

UiT THE ARCTIC UNIVERSITY OF NORWAY MACHINE LEARNING GROUP

Degree matrix **D**

- NxN diagonal

$d(i,j) = \text{degree}(v_i) \quad \text{if } i=j$
$\text{degree}(v_i) = \text{sum } i^{th} \text{ row}$

0	1	0	0	0	1	1	0	1
1	0	1	1	1	0	0	0	0
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	1	0	0	0	1	1	0
1	0	0	0	0	1	0	1	1
0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	1	0	0

4	0	0	0	0	0	0	0	0
0	4	0	0	0	0	0	0	0
0	0	3	0	0	0	0	0	0
0	0	0	3	0	0	0	0	0
0	0	0	0	3	0	0	0	0
0	0	0	0	0	4	0	0	0
0	0	0	0	0	0	4	0	0
0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	2

Graph basics

Laplacian matrix **L = D - K**

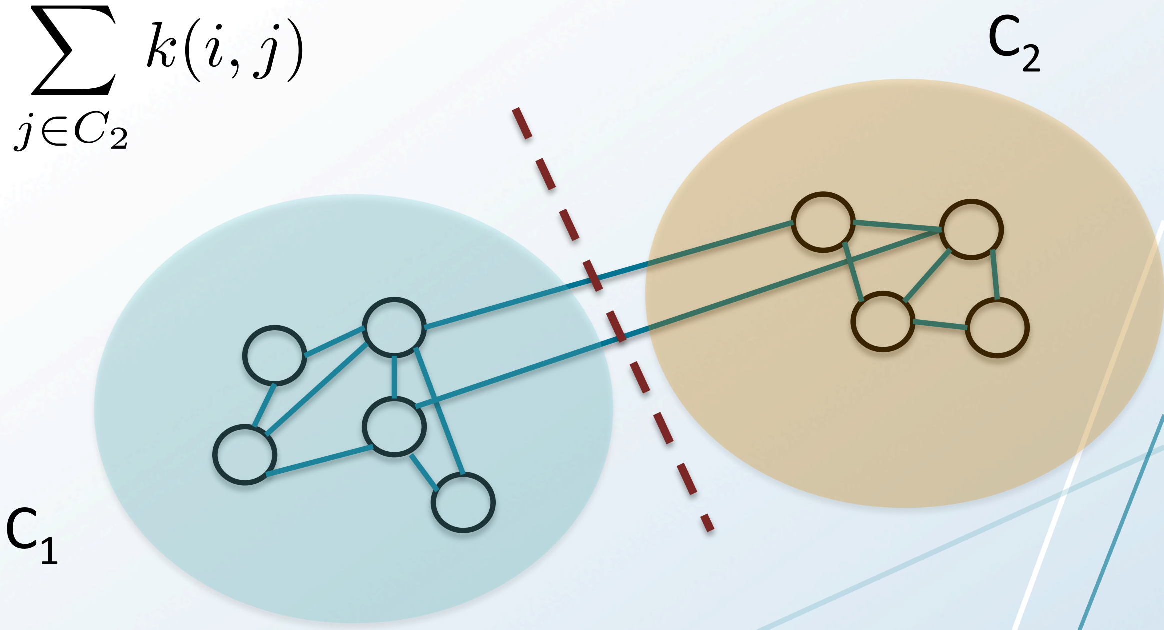
- NxN symmetric and positive semi-definite (real and positive eigenvalues)
- The smallest eigenvalue is 0 and the corresponding eigenvector is constant
- The eigenvector corresponding to the second smallest eigenvector is special: *Fiedler vector*. It is related to graph cuts!

Many spectral clustering methods use the Laplacian matrix (or a version of it) to embed data for then to perform clustering!

The Laplacian and graph cuts

- Min-cut problem: Find C_1 and C_2 such that the *cut* is minimized

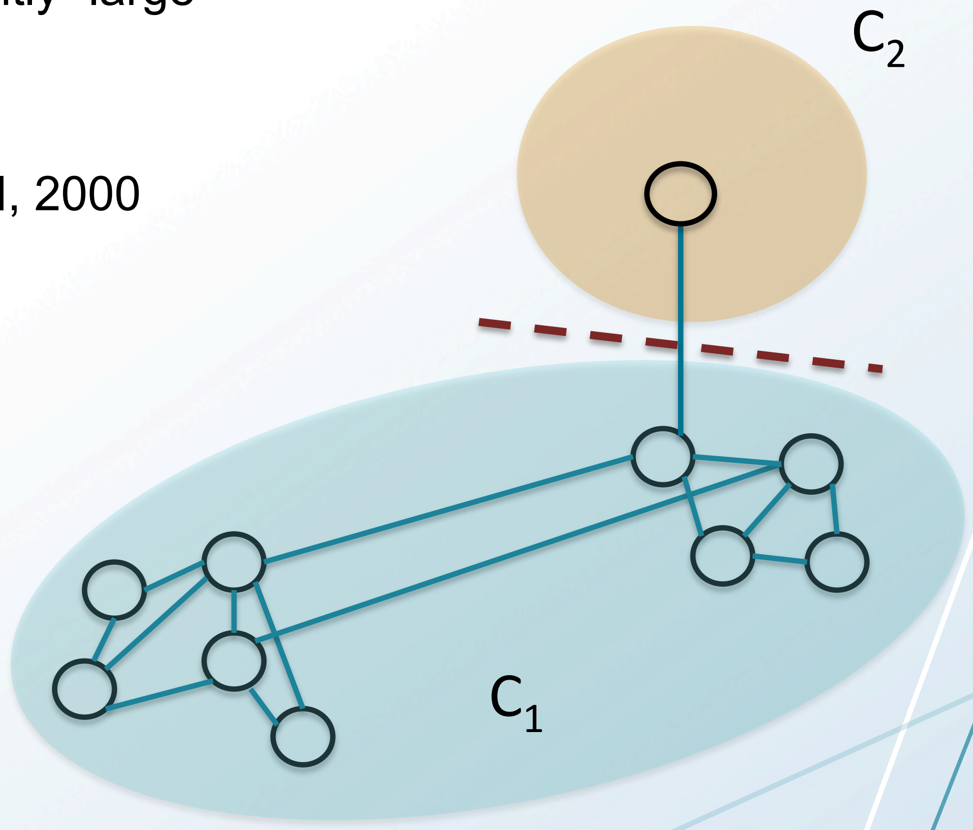
$$\mathrm{CUT}(C_1, C_2) = \sum_{i \in C_1} \sum_{j \in C_2} k(i,j)$$



The Laplacian and graph cuts

- Does not always lead to reasonable results if the connected components are imbalanced
- Ensure that clusters are sufficiently “large”

→ Normalized cut
J. Shi, J. Malik, IEEE TPAMI, 2000

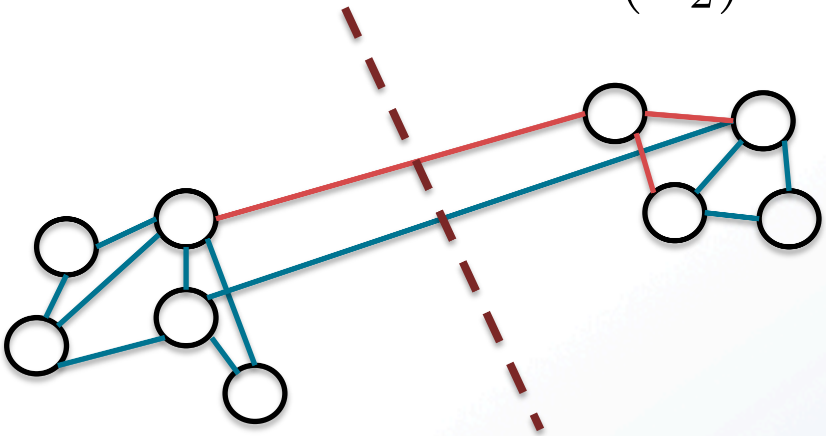


M. Meila, J. Shi, A random walks view of spectral segmentation, AISTATS, 2001
R. Jenssen et al., The Laplacian PDF distance, NIPS 2005
U. Luxburg, A tutorial on spectral clustering, Statistics and Computing, 2007

Normalized cut

- Normalize the *cut* by the *volumes* of the sub-graphs

$$\mathrm{VOL}(C_2) = \sum_{j \in C_2} \mathrm{degree}_j$$

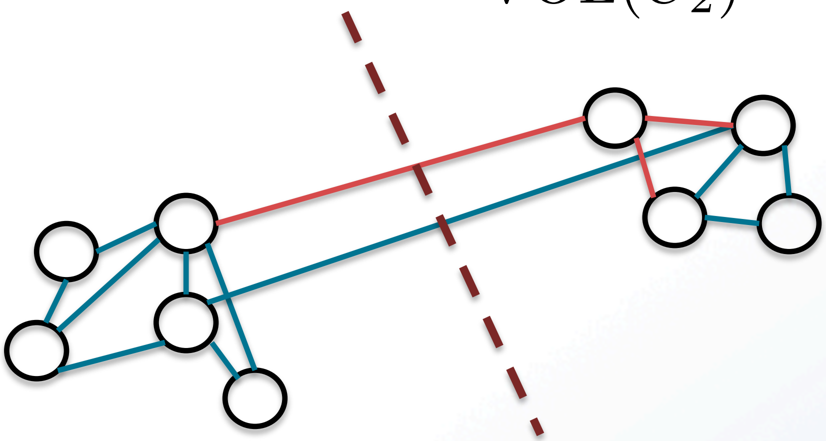


$$\mathrm{NCUT}(C_1, C_2) = \frac{\mathrm{CUT}(C_1, C_2)}{\mathrm{VOL}(C_1)} + \frac{\mathrm{CUT}(C_1, C_2)}{\mathrm{VOL}(C_2)}$$

Normalized cut

- Normalize the *cut* by the *volumes* of the sub-graphs

$$\mathrm{VOL}(C_2) = \sum_{j \in C_2} \mathrm{degree}_j$$



Find C_1 and C_2 such that NCUT is minimized!

$$\mathrm{NCUT}(C_1, C_2) = \frac{\mathrm{CUT}(C_1, C_2)}{\mathrm{VOL}(C_1)} + \frac{\mathrm{CUT}(C_1, C_2)}{\mathrm{VOL}(C_2)}$$

Solution (relaxed – graph view) to NCUT

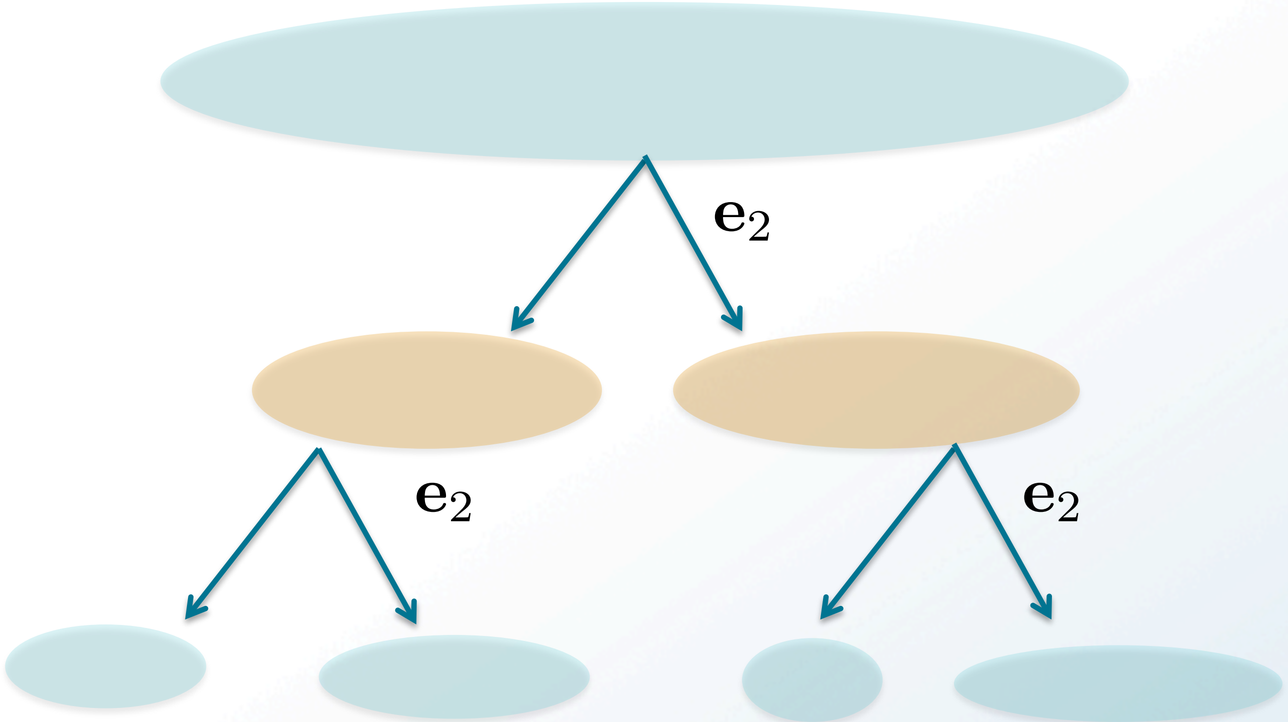
- Strangely:
  - Form: $\mathbf{L}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$
  - Compute: $\mathbf{L}_{\text{sym}}\mathbf{e}_i = \delta_i \mathbf{e}_i$
  - Second largest eigenvector: $\mathbf{e}_2$

$$[0.2 \quad 0.21 \quad 0.23 \quad -0.34 \quad 0.25 \quad -0.33 \quad -0.4 \quad \ldots]$$

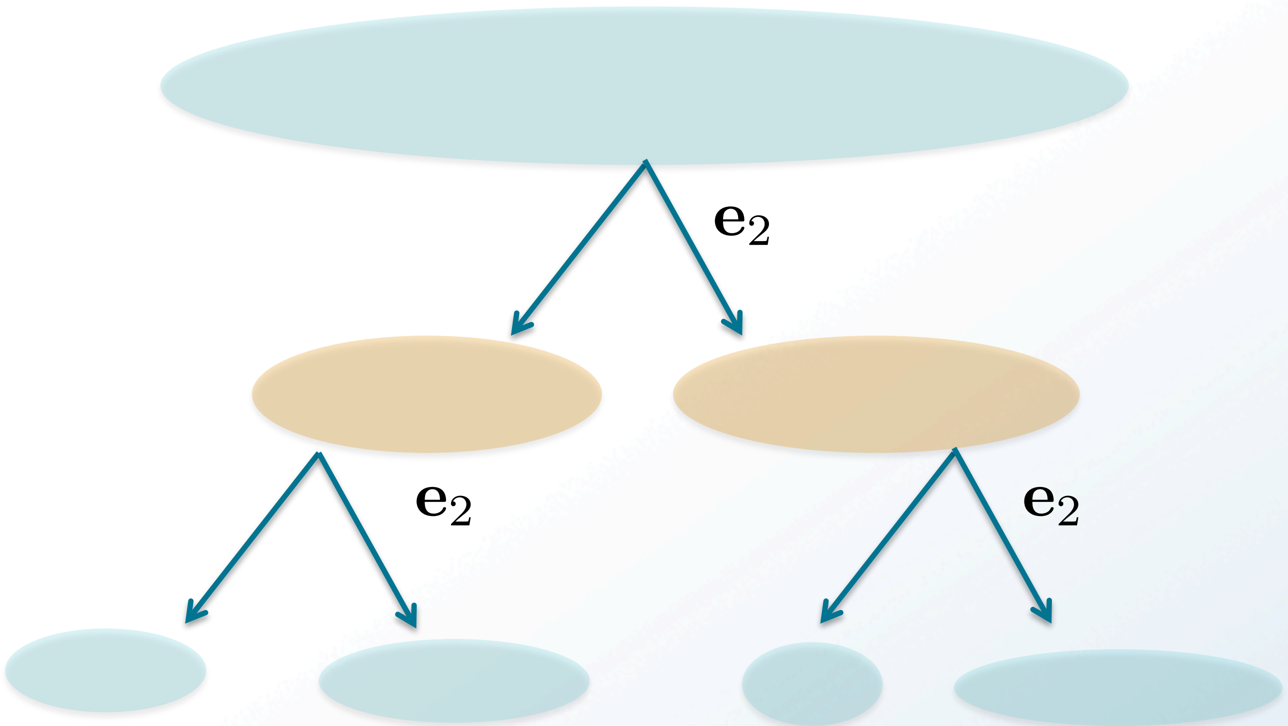
positive: $\mathbf{x}_1 \to C_1$

negative: $\mathbf{x}_6 \to C_2$

Graph view cont'd



Graph view cont'd

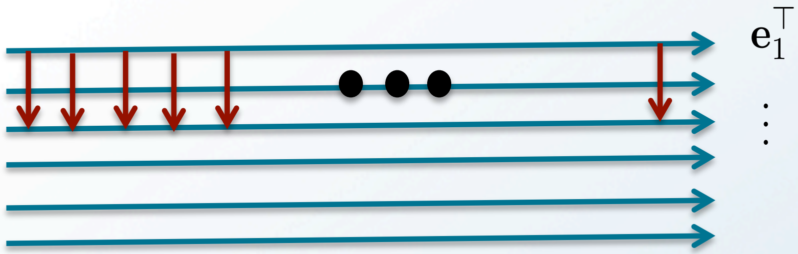


OR use *k* eigenvectors, and embed the data into a k-dimensional space!

NCUT (embedding view)

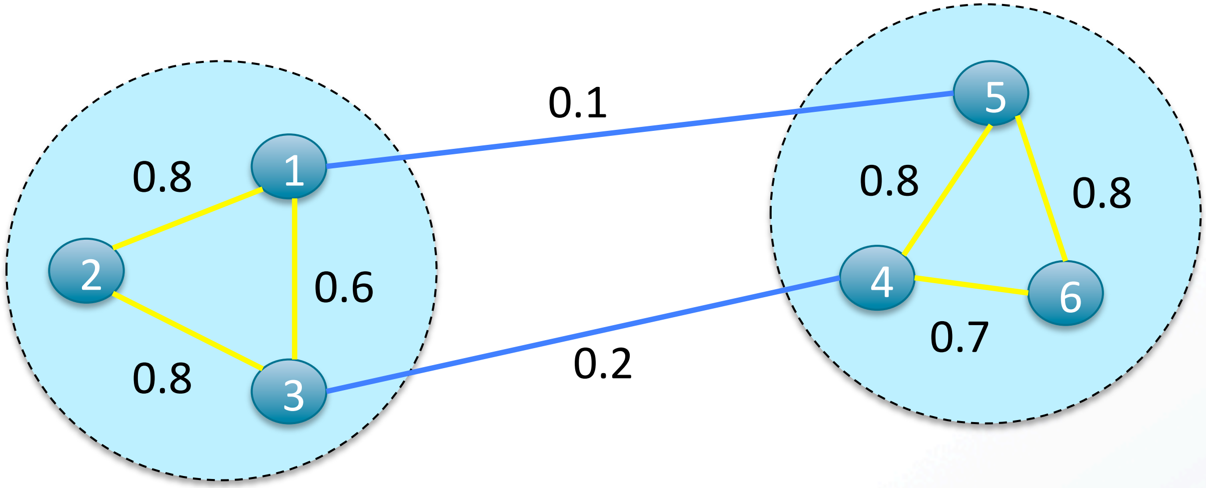
UiT
THE ARCTIC UNIVERSITY OF NORWAY
MACHINE LEARNING GROUP

- Do
  - Form: $\mathbf{L}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$
  - Compute: $\mathbf{L}_{\text{sym}}\mathbf{e}_i = \delta_i \mathbf{e}_i$
  - Select the *k* largest eigenvectors and store them as rows in a matrix $\mathbf{E}_k$
  - Let $\mathbf{y}_i$ be the vector corresponding to the ith column of $\mathbf{E}_k$
  - Cluster $\mathbf{y}_i$ for i=1,…,N with e.g. k-means

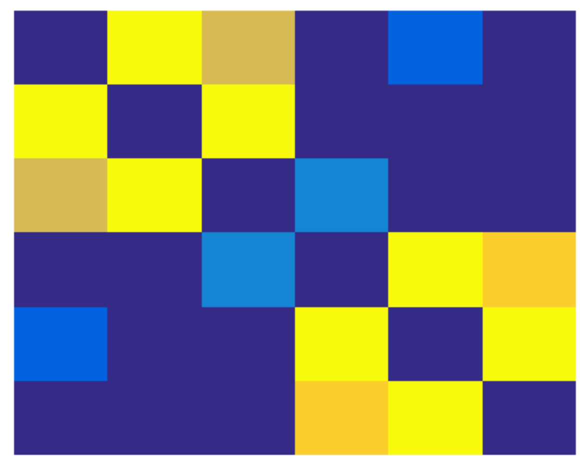


A. Ng, M. Jordan, Y. Weiss, On spectral clustering, NIPS 2002

Example



	x_1	x_2	x_3	x_4	x_6	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0



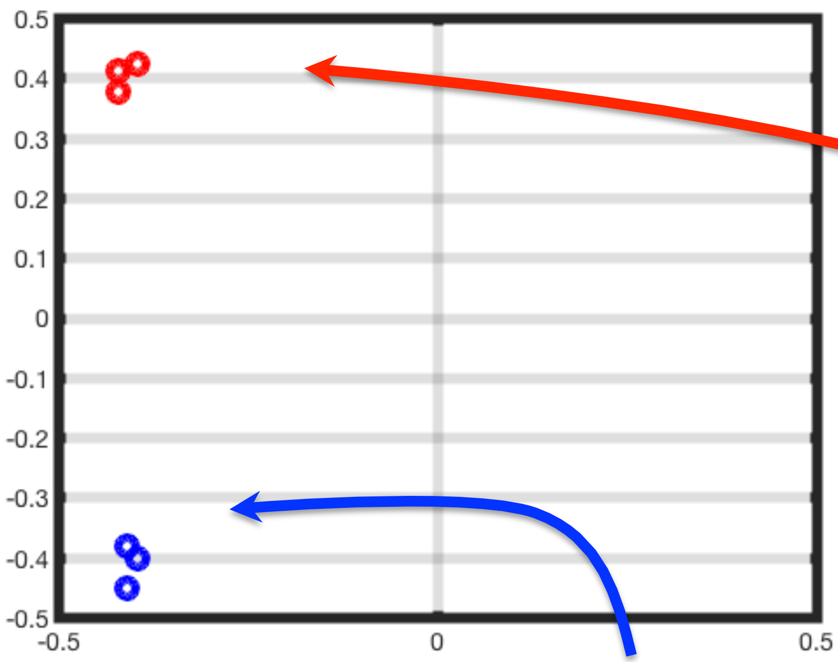
Example

	x_1	x_2	x_3	x_4	x_6	x_6
x_1	0	0.51	0.39	0	0.06	0
x_2	0.52	0	0.50	0	0	0
x_3	0.39	0.50	0	0.12	0	0
x_4	0	0	0.12	0	0.47	0.44
x_5	0.06	0	0	0.47	0	0.50
x_6	0	0	0	0.44	0.50	0

$$\mathbf{L}_{\mathrm{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$$

$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_3$	$\mathbf{y}_4$	$\mathbf{y}_5$	$\mathbf{y}_6$	
−0.39	−0.40	−0.40	−0.42	−0.42	−0.39	$\mathbf{e}_1^\top$
−0.40	−0.45	−0.38	0.37	0.41	0.42	$\mathbf{e}_2^\top$

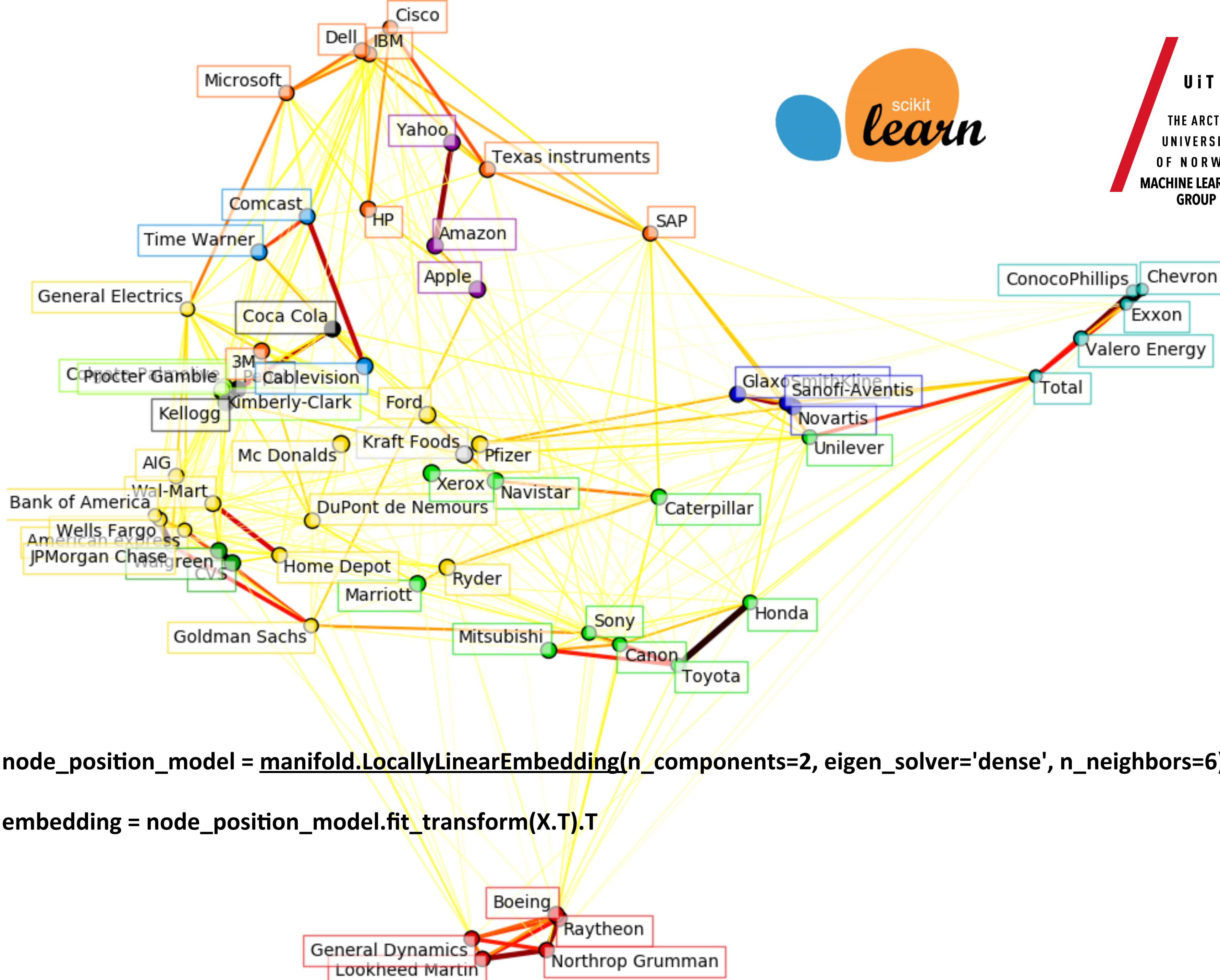
Example



MATLAB

```
Lsym=D^(-.5)*K*D^(-.5)
[E,Delta]=eigs(Lsym,2);
y=E';
```

−0.39	−0.40	−0.40	−0.42	−0.42	−0.39
−0.40	−0.45	−0.38	0.37	0.41	0.42



scikit learn

UiT THE ARCTIC UNIVERSITY OF NORWAY
MACHINE LEARNING GROUP

node_position_model = <u>manifold.LocallyLinearEmbedding(</u>n_components=2, eigen_solver='dense', n_neighbors=6)

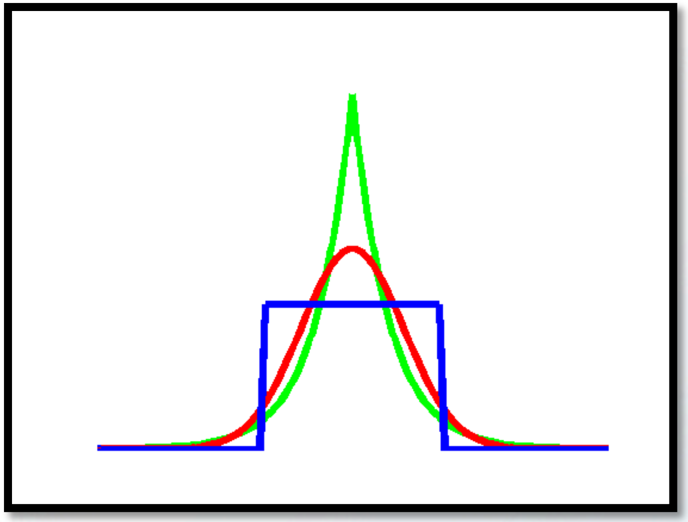
embedding = node_position_model.fit_transform(X.T).T

Entropy

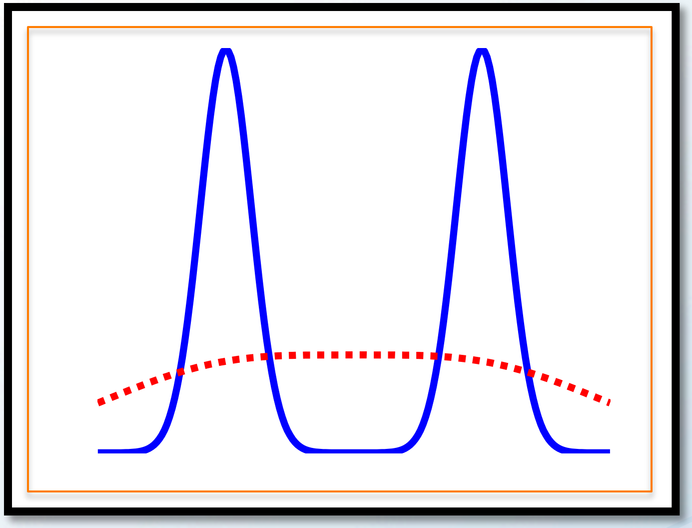
A measure of the uncertainty, or *information*, associated with a random variable described by a probability distribution.

$$H_{R_2} = -\log \int p^2(\mathbf{x}) d\mathbf{x}$$

Shape



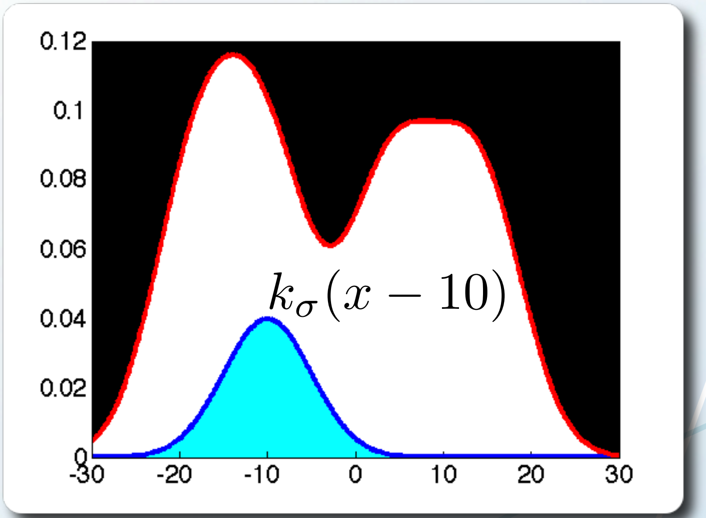
Modes, clusters



Kernel Entropy Component Analysis (KECA)

$$\hat{p}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} k_\sigma(\mathbf{x} - \mathbf{x}_i)$$

Gaussian,
Epanechnikov ++



$$\int p^2(\boldsymbol{x})d\boldsymbol{x} \approx \frac{1}{N^2}\underbrace{\mathbf{1}^\top \boldsymbol{K}\mathbf{1}}_{}$$

Sum all elements

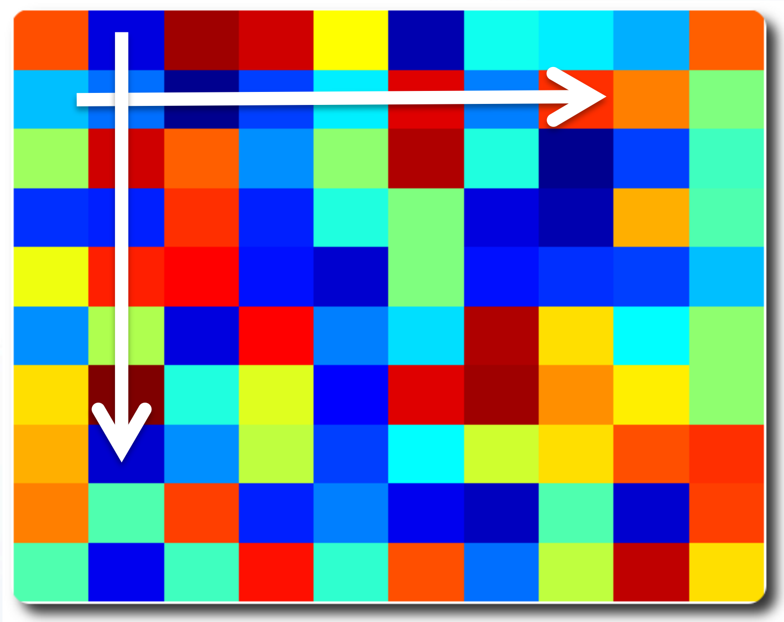
KECA

Sum of elements

$k_\sigma(\mathbf{x}_1, \mathbf{x}_9)$

$$\int p^2(\boldsymbol{x})d\boldsymbol{x} \approx \frac{1}{N^2}\sum_{i=1}^{N}\left[\sqrt{\lambda_i}\boldsymbol{e}_i^\top \mathbf{1}\right]^2$$

$\mathbf{K}$



$$\mathbf{K} = \mathbf{EDE}^\top$$

$\sqrt{\lambda_i}\boldsymbol{e}_i^\top$ is a spectral *transformation*, or mapping, of data based on $\mathbf{K}$

KECA measures contribution of each dimension to entropy.

Represent data in lower dimensions by selecting entropy-preserving features!

KECA

UiT THE ARCTIC UNIVERSITY OF NORWAY
MACHINE LEARNING GROUP

- Select the kernel function [rule-of-thumb!] and create $\mathbf{K}$
- Eigendecompose $\mathbf{K}$ and compute entropy values
- Represent input data using (lower dimensional) features corresponding to high entropy values

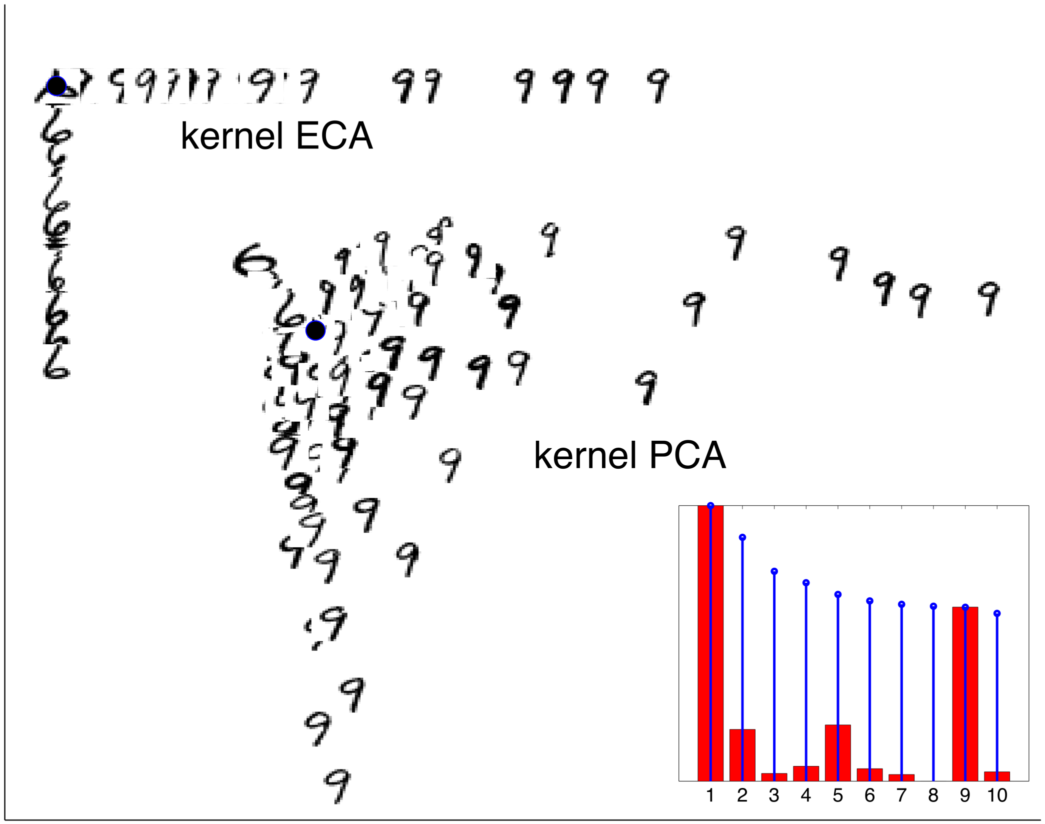
Selected dimensions depend both on eigenvalue and on structure of eigenvector!

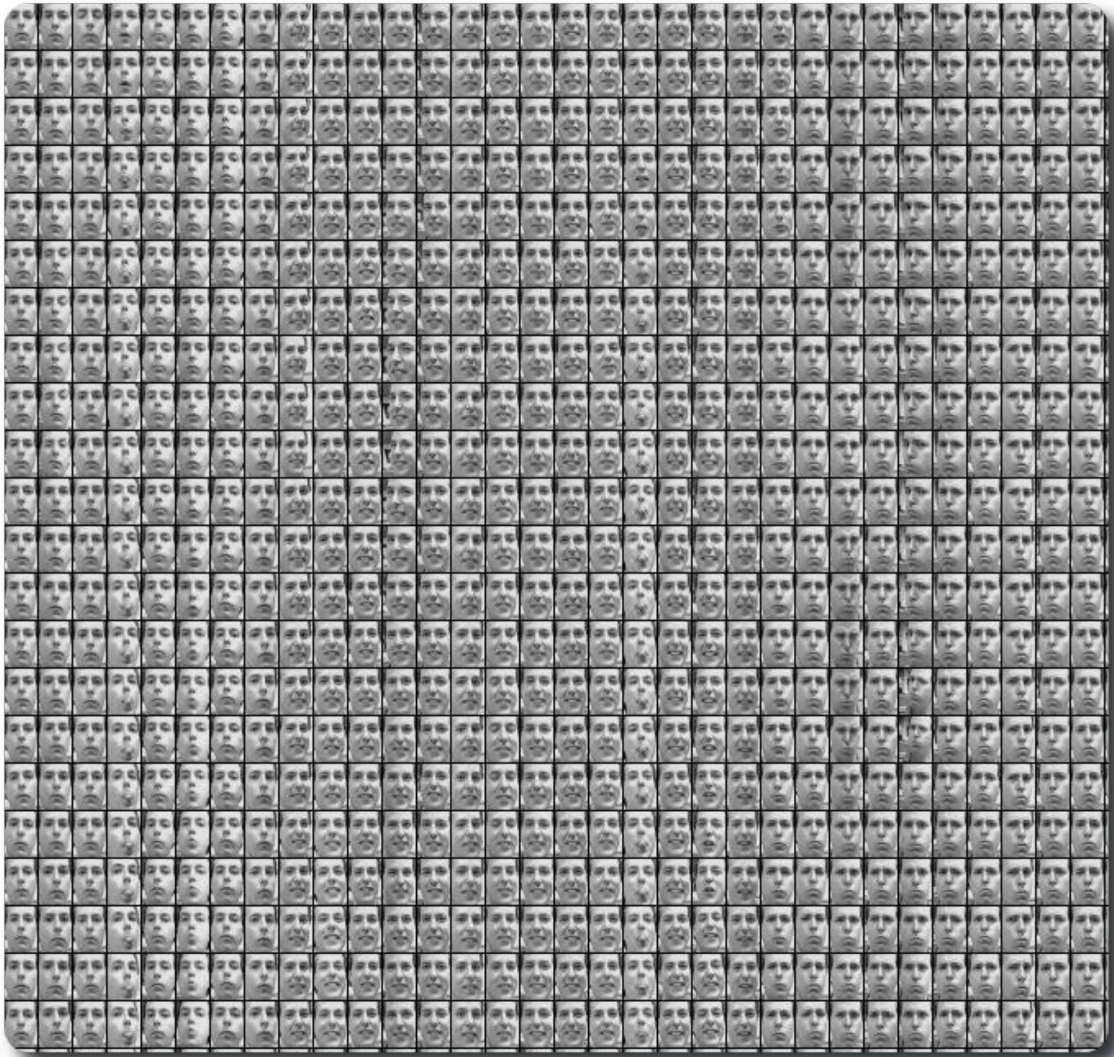
Different from "all" other methods!

$$\int p^2(\boldsymbol{x})d\boldsymbol{x} \approx \frac{1}{N^2}\sum_{i=1}^{N}\Big[\underbrace{\sqrt{\lambda_i}\boldsymbol{e}_i^\top \mathbf{1}}_{\text{Entropy values}}\Big]^2$$

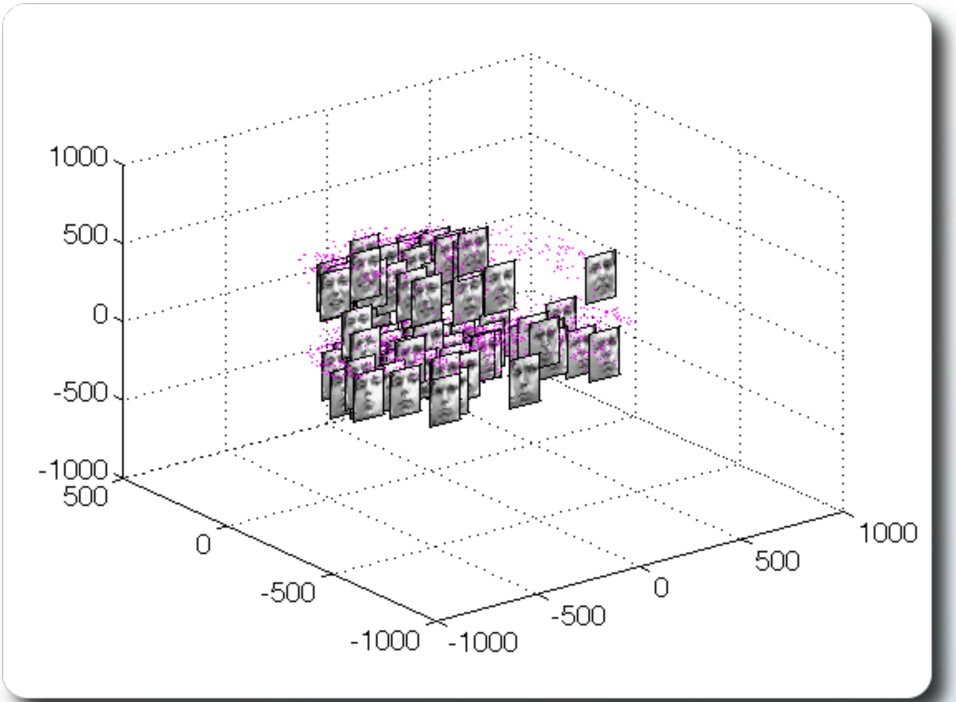
E. Izquierdo-Verdiguier, R. Jenssen et al., Optimized kernel entropy components, IEEE Trans. Neural Networks and Learning Systems, 2016

KECA



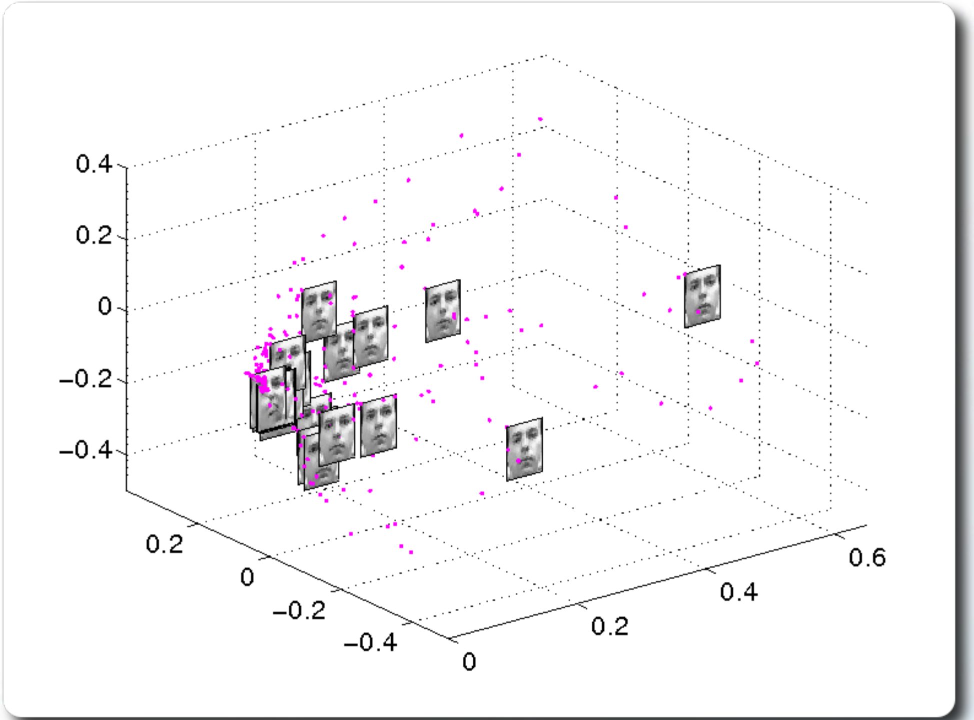


Faces



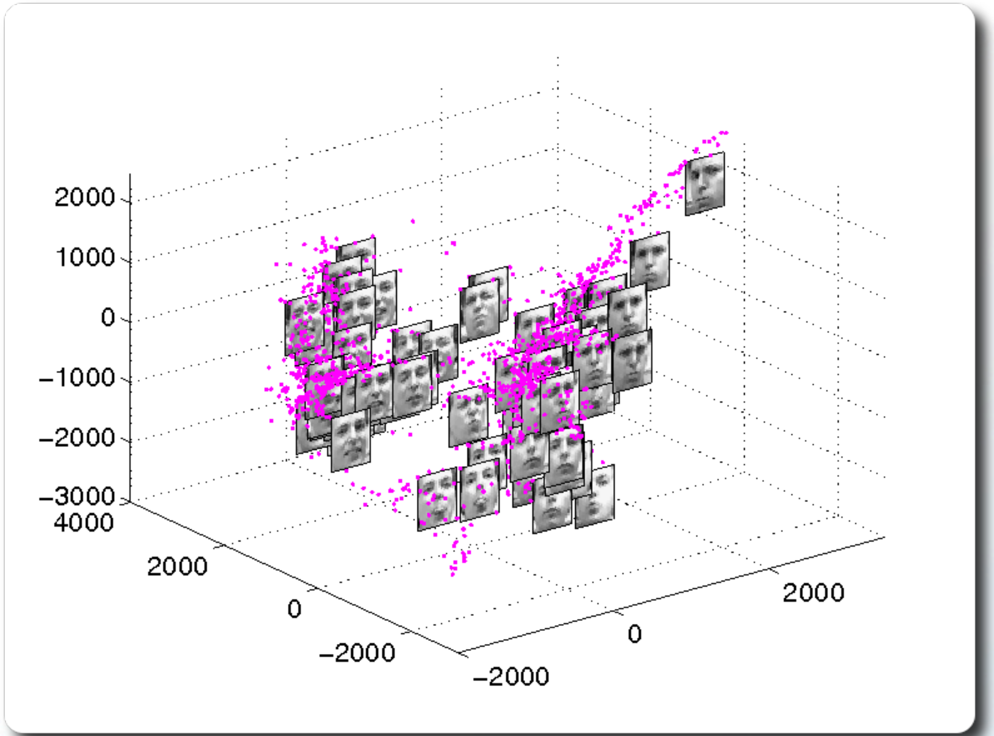
PCA

Faces



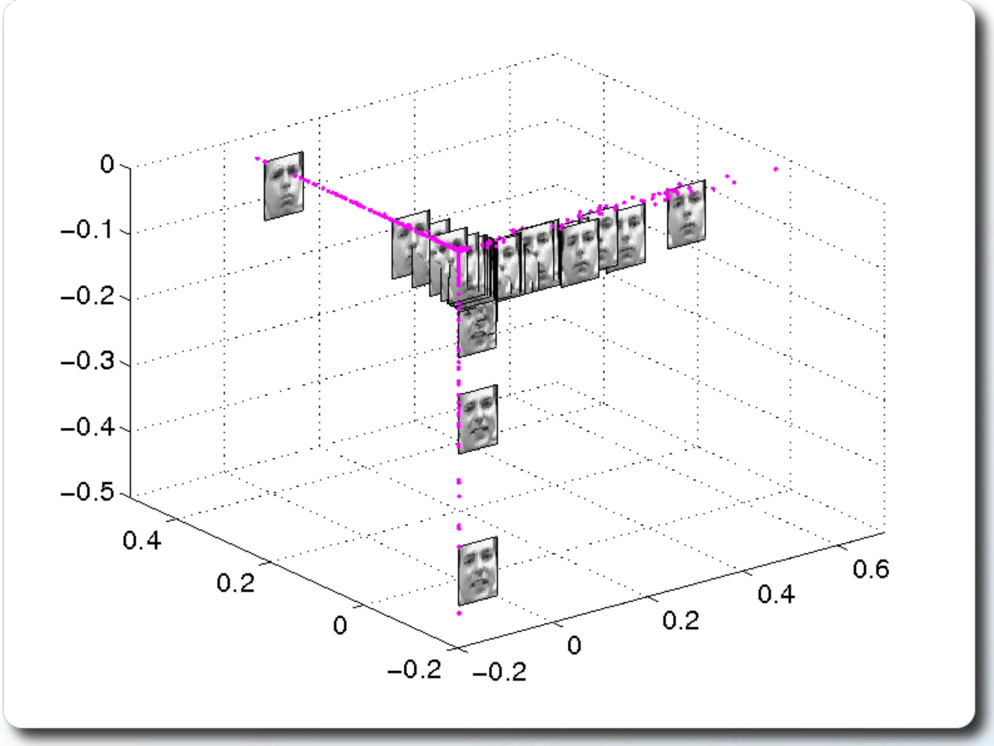
KPCA [Schölkopf et al.]

Faces



Isomap [Tenenbaum et al.]

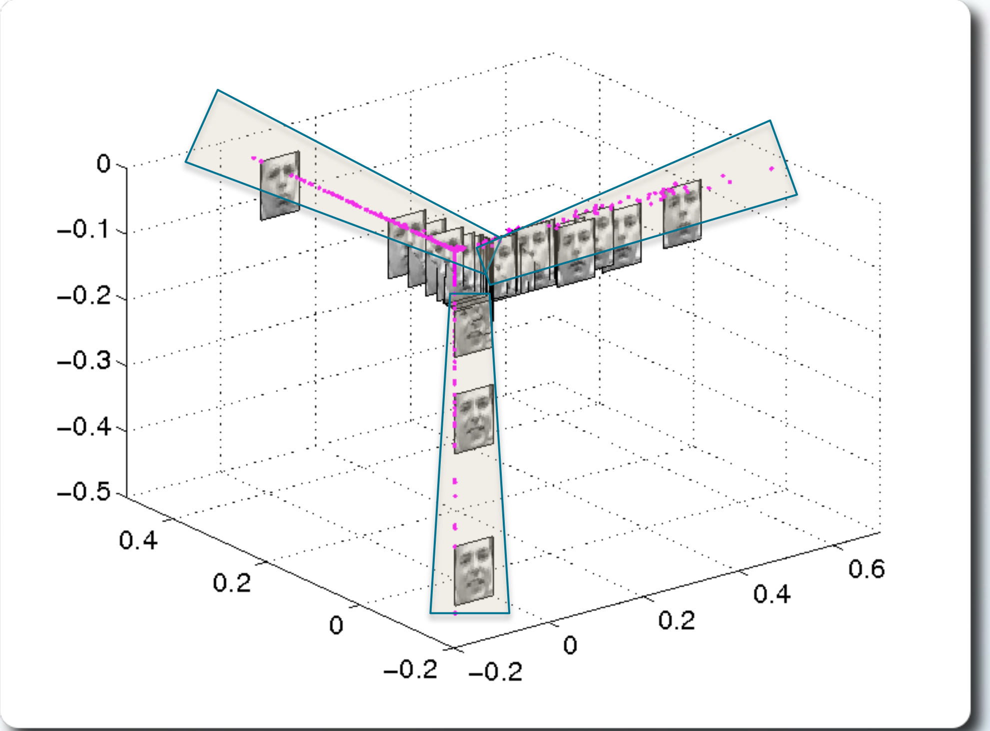
Faces



KECA [1, 4, 10]

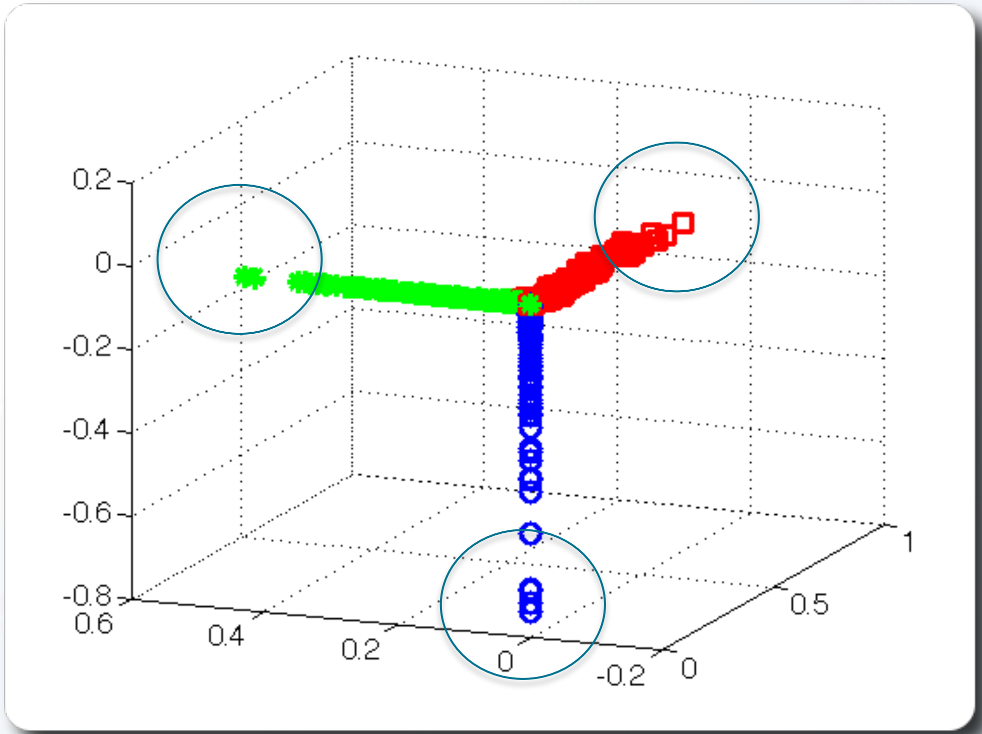
Faces

- KECA Clustering
  - Cosine k-means
  - Max divergence!
  - Initialization

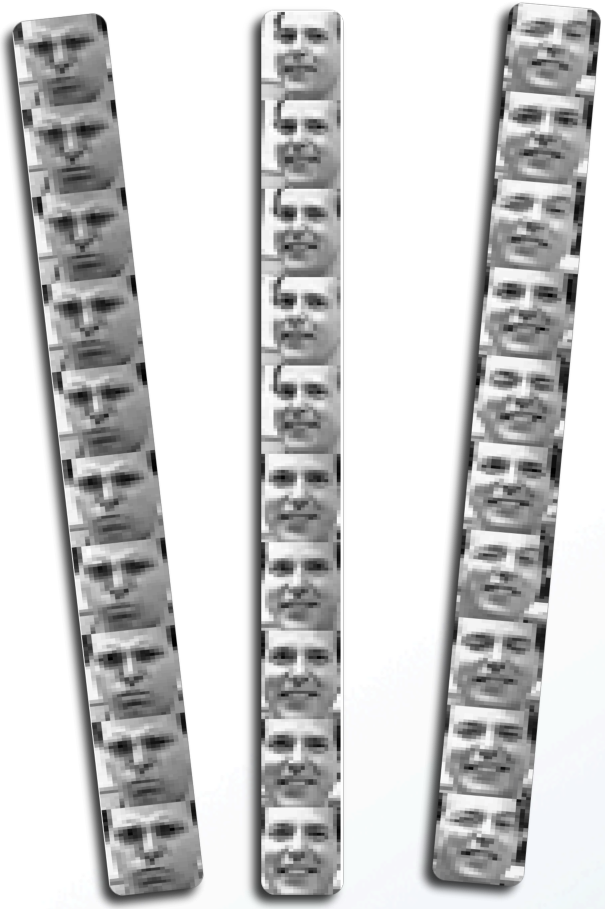


KECA

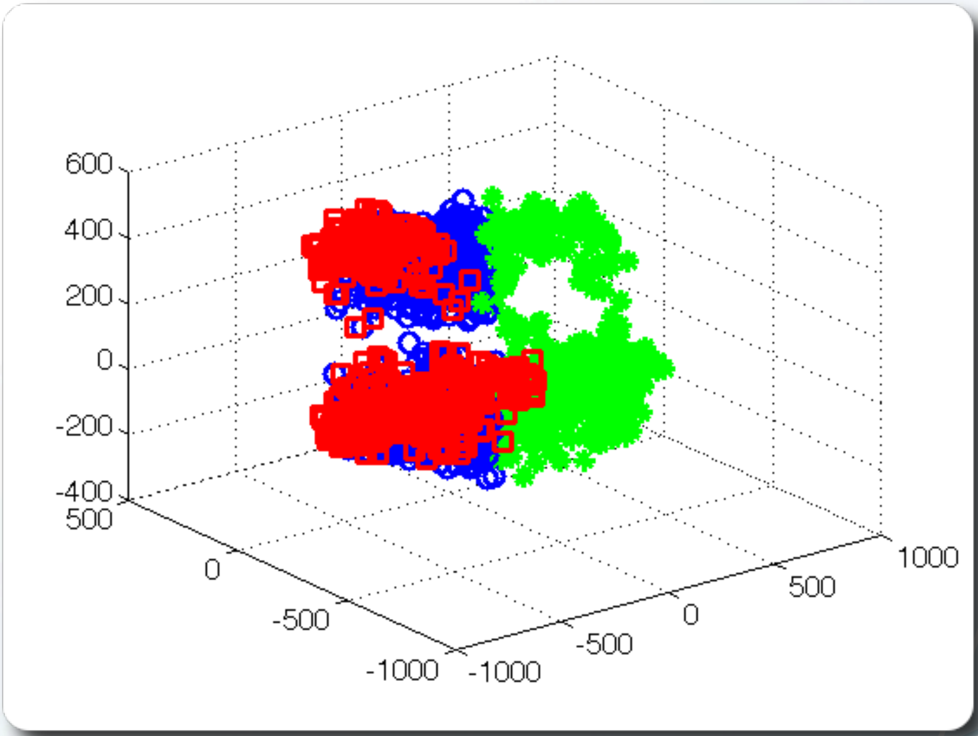
Faces



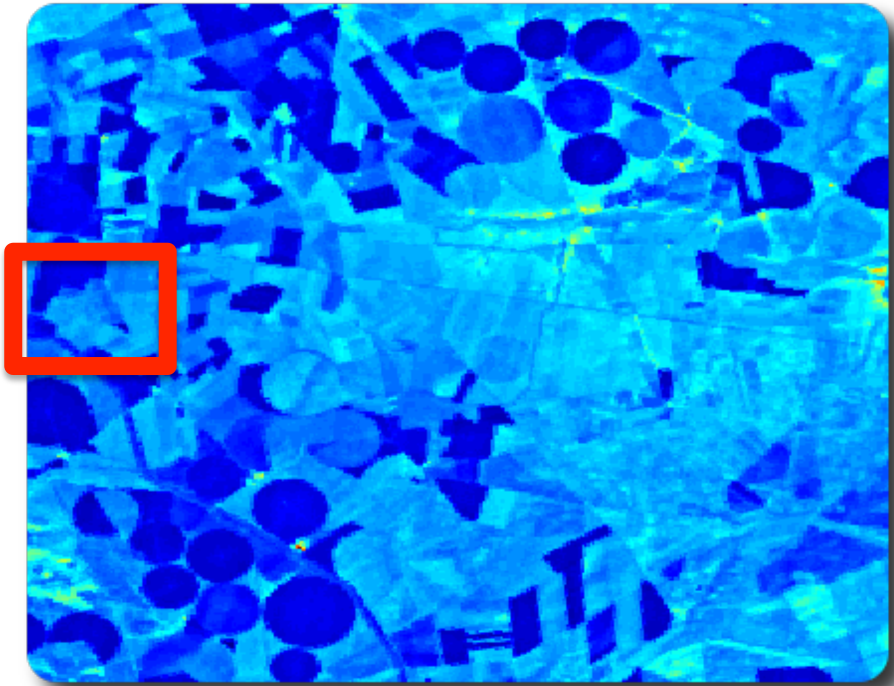
Faces

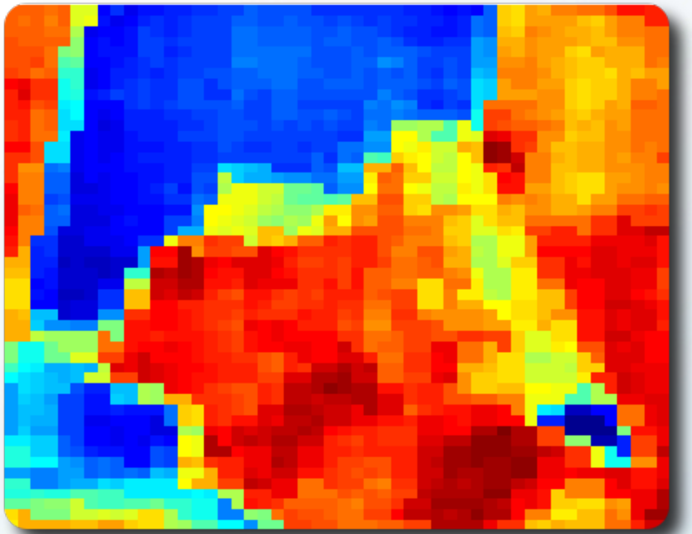


PCA + k-means

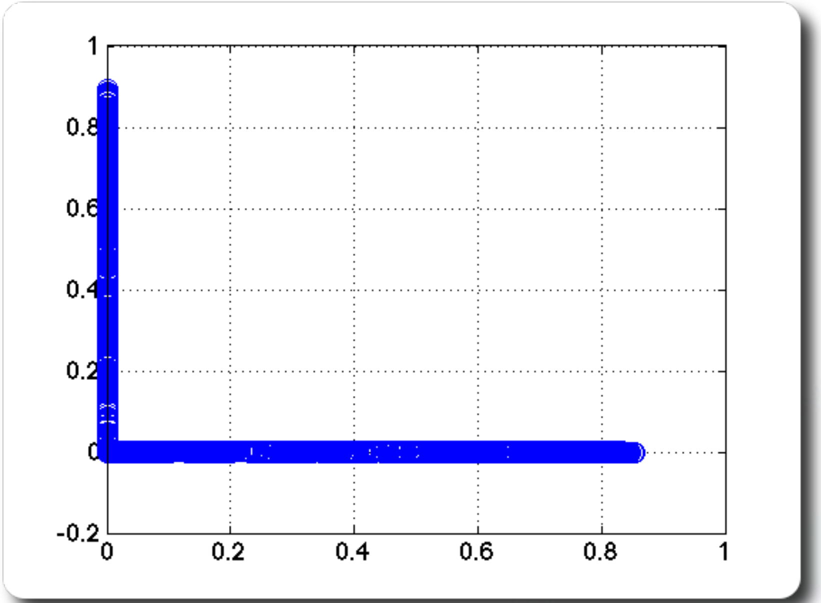


Hyperspectral

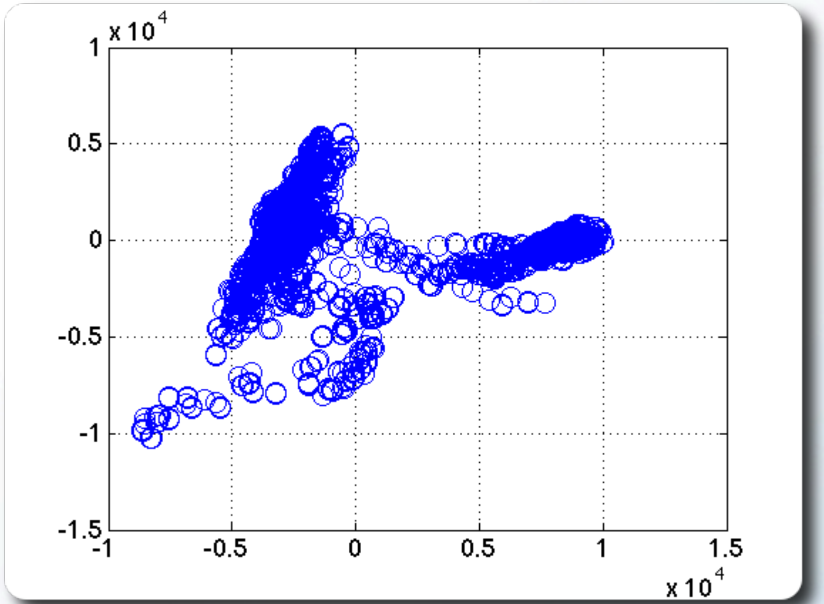




52 bands
Chlorophyll
Spain (G. Camps-Valls)



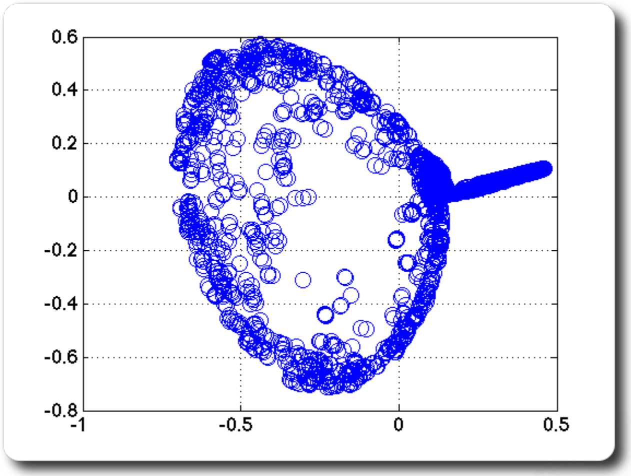
KECA



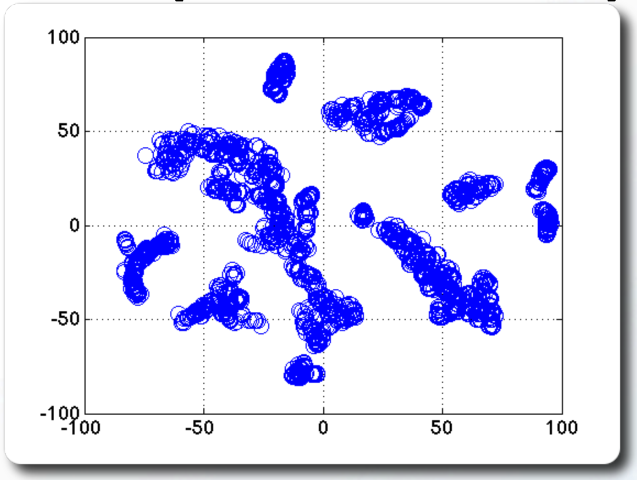
PCA

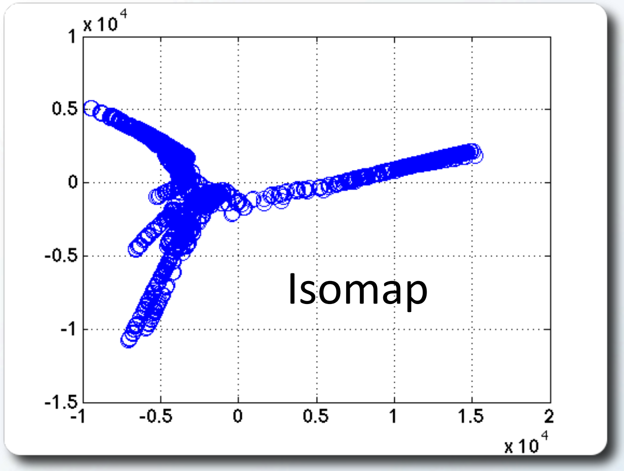
Hyperspectral

KPCA

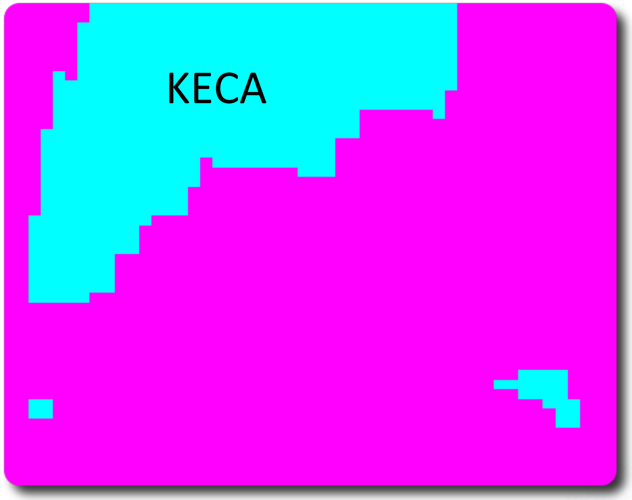


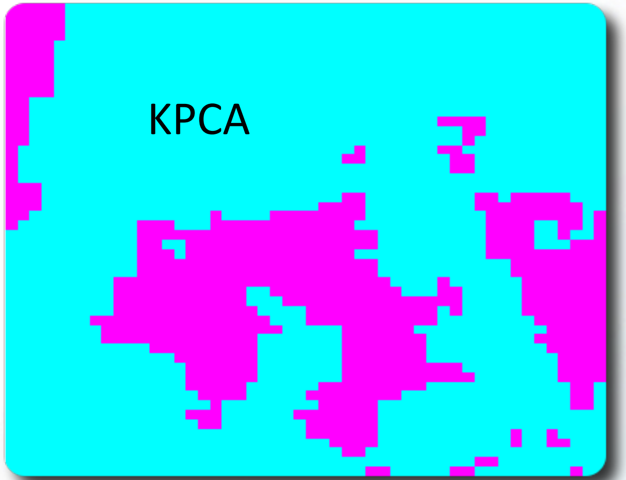
t-SNE [van Maaten & Hinton]

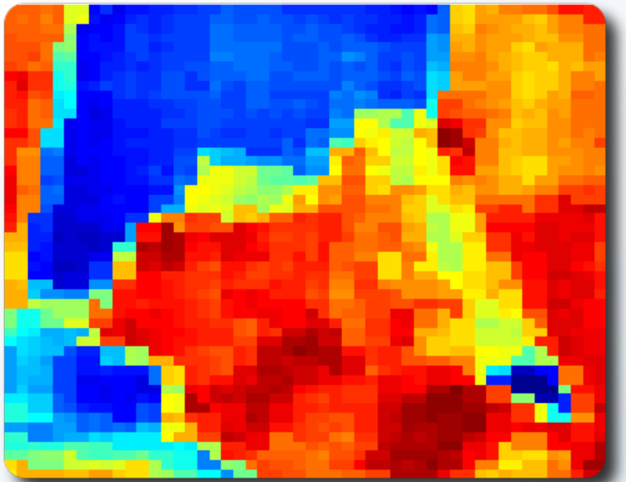




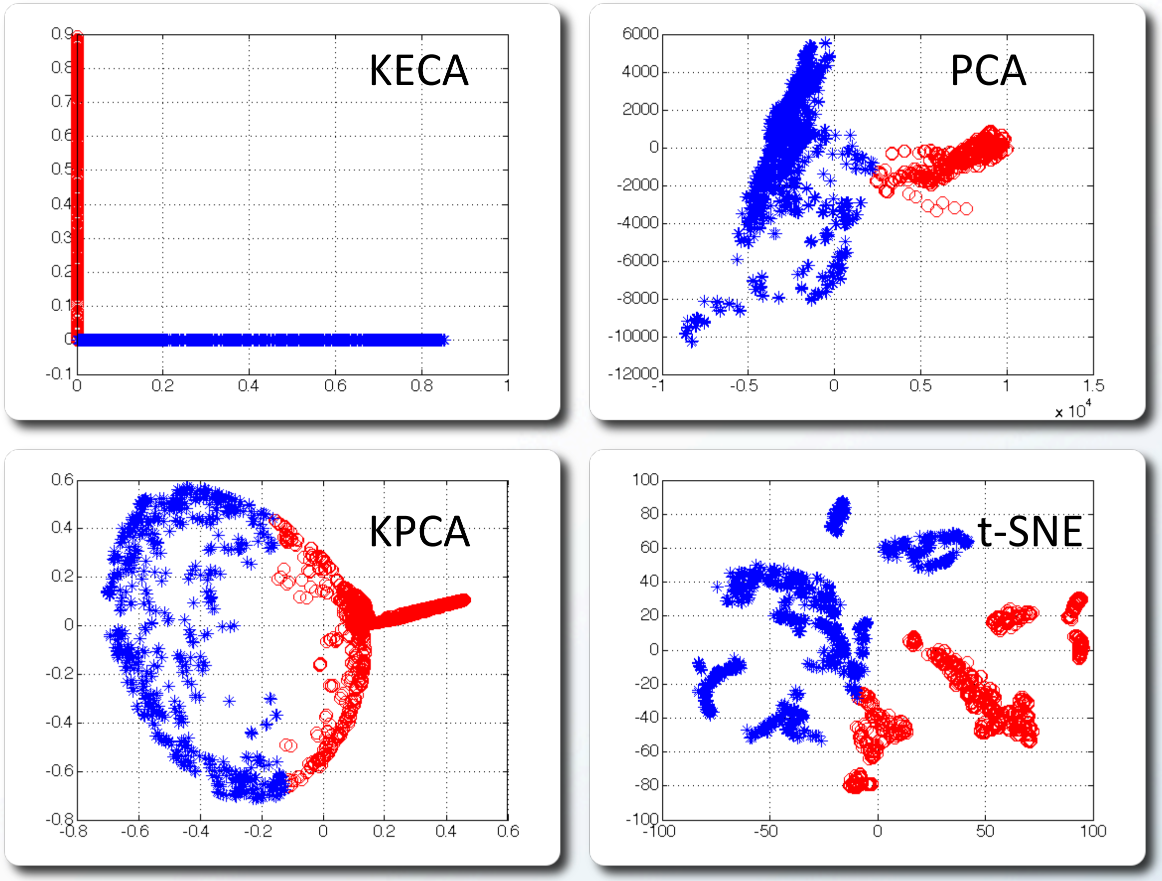
Hyperspectral



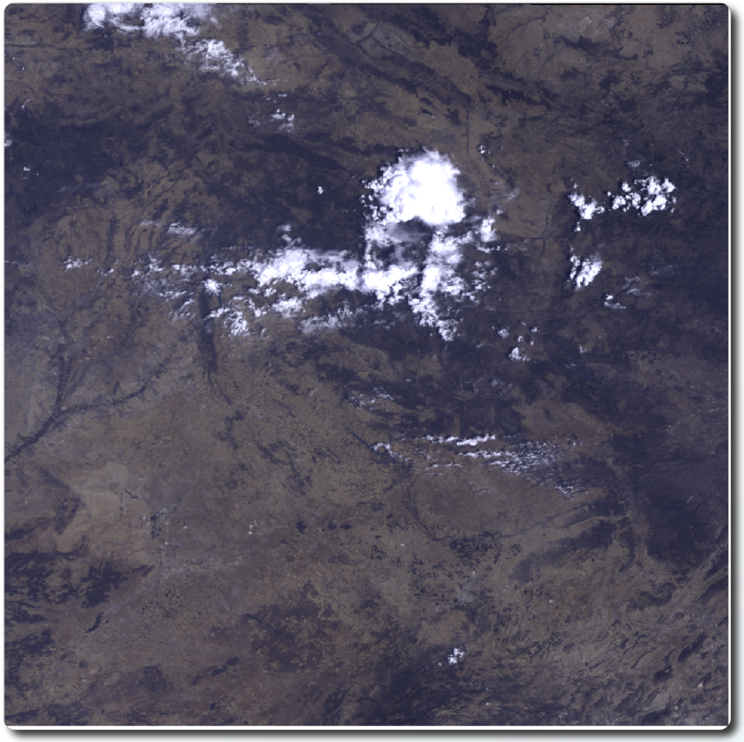




Hyperspectral



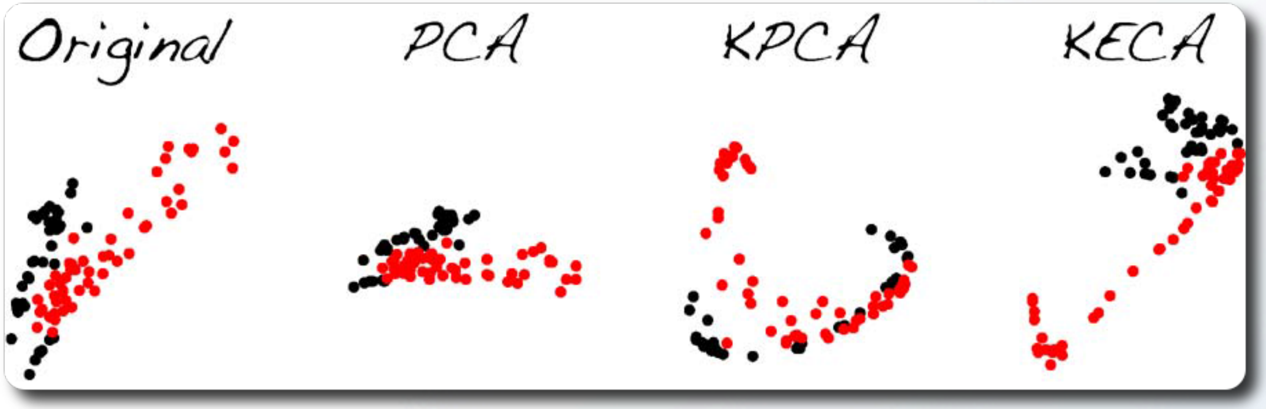
Cloud screening



- Morphological features (22)
- Cloud vs. no-cloud

L. Gomez-Chova, R. Jenssen and Camps-Valls, Kernel entropy component analysis for remote sensing image clustering
IEEE Geoscience and Remote Sensing Society Best Paper Award, 2013

Cloud screening



FINAL

UiT THE ARCTIC UNIVERSITY OF NORWAY MACHINE LEARNING GROUP

Cloud screening