Kernel-based Methods for Unsupervised Learning

LEAR project-team, INRIA

Zaid Harchaoui

Grenoble, July 30th 2010
Outline

1 Introduction

2 Kernel methods and feature space

3 Mean element and covariance operator

4 Kernel PCA

5 Kernel CCA

6 Spectral Clustering
Outline

1. Introduction
2. Kernel methods and feature space
3. Mean element and covariance operator
4. Kernel PCA
5. Kernel CCA
6. Spectral Clustering
Unsupervised learning

Dimension reduction

According to media reports, a pair of hackers said on Saturday that the Firefox Web browser, commonly perceived as the safer and more customizable alternative to market leader Internet Explorer, is critically flawed. A presentation on the flaw was shown during the ToorCon hacker conference in San Diego.

Zambian President Levy Mwanawasa has won a second term in office in an election his challenger Michael Sata accused him of rigging, official results showed on Monday.

face images
documents
gene expression data
MEG readings
Unsupervised learning

Dimension reduction

- Computational efficiency: space and time savings
- Statistical performance: fewer dimensions $\rightarrow$ regularization
- Visualization: discover underlying structure of the data

$\rightarrow$ PCA and KPCA
Unsupervised learning

Feature extraction

A view from Idyllwild, California, with pine trees and snow capped Marion Mountain under a blue sky.

Learn kernelized projections that relate both spaces.

Harcaoui (FR)

VRML

Grenoble 6 / 66
Unsupervised learning

Feature extraction

- Multimodality: leverage the correlation between the modalities
- Statistical performance: take advantage of both views of the data
- Putting in relation: discover underlying relations between the modalities

→ CCA and KCCA
Unsupervised learning

Clustering
Unsupervised learning

Clustering

- Semantics: grouping datapoints in meaningful clusters
- Statistical performance: intrinsic degrees of freedom of the data
- Visualization: discover groupings between datapoints

→ spectral clustering, temporal segmentation, and regularized clustering (DIFFRAC)
Unsupervised learning

Detection problems
Unsupervised learning

Detection problems

- Balance risks: control detection rate with a guaranteed false alarm probability
- Power: detect differences not only in mean or covariance

→ homogeneity testing, change detection
Outline

1. Introduction
2. Kernel methods and feature space
3. Mean element and covariance operator
4. Kernel PCA
5. Kernel CCA
6. Spectral Clustering
Kernel methods

Machine Learning methods taking $\mathbf{K} = [k(X_i, X_j)]_{i,j=1,...,n}$ (Gram matrix as input for processing a sample $\{X_1, \ldots, X_n\}$, where $k(x, y)$ is a similarity measure between $x$ and $y$ defining a positive definite kernel.

Strengths of Kernel Methods

- **Minimal assumptions** on data types (vectors, strings, trees, graphs, etc.)
- **Interpretation of** $k(x, y)$ as a dot product $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ in a reproducing kernel Hilbert space $\mathcal{H}$ where the observations are mapped via $[\phi : \mathcal{X} \to \mathcal{H}]$ the feature map $\phi(\bullet) = k(\bullet, \cdot)$
How does the feature space look like?

Example: space of shapes of birds
How does the feature space look like?

Feature map?

How does the feature map look like?

$$k ( \cdot , \cdot )$$
How does the feature space look like?

Feature map?

The feature map is a function whose values span the whole range of shapes with varying magnitudes.
Examples of Kernels

Kernels on vectors

- **Polynomial** \( k(x, y) = (c + (x, y))^d \)
- **Laplace** \( k(x, y) = \exp(-\|x - y\|_1/\sigma) \)
- **RBF** \( k(x, y) = \exp(-\|x - y\|_2^2/\sigma^2) \)
Examples of Kernels

Kernels on histograms

Kernels built on top of divergence between probability distributions

\[
\psi_{JD}(\theta, \theta') = h\left(\frac{\theta + \theta'}{2}\right) - \frac{h(\theta) + h(\theta')}{2},
\]

\[
\psi_{\chi^2}(\theta, \theta') = \sum_i \frac{(\theta_i - \theta'_i)^2}{\theta_i + \theta'_i}, \quad \psi_{TV}(\theta, \theta') = \sum_i |\theta_i - \theta'_i|,
\]

\[
\psi_{H_2}(\theta, \theta') = \sum_i |\sqrt{\theta_i} - \sqrt{\theta'_i}|^2, \quad \psi_{H_1}(\theta, \theta') = \sum_i |\sqrt{\theta_i} - \sqrt{\theta'_i}|.
\]

\[
k(\theta, \theta') = \exp\left(-\psi(\theta, \theta')/\sigma^2\right).
\]
Kernel methods and feature space

The kernel jungle

Kernels on histograms

- Pyramid match kernels (Grauman and Darrell, 2005)
- Multiresolution (nested histograms) kernels (Cuturi, 2006)
- Walk and tree-walk kernels (Ramon & Gaertner, 2004; Harchaoui & Bach, 2007; Mahe et al., 2007)

Kernels from statistical generative models

- Mutual Information Kernels (Seeger, 2002)
- Fisher kernels (see Shawe-Taylor & Cristianini, 2004)

Other kernels

- Kernels of shapes and point clouds (Bach, 2007)
- Kernels on time series (Cuturi, 2007)
How does the feature space look like?

Classical kernel trick
- Describes what happens to pairs of examples
- Focuses on the *pointwise* effect of the feature map on an example

“Remixed” kernel trick
- Describes what happens to a random sample from a probability distribution
- Focuses on the *global* effect of the feature map on a sample
Outline

1 Introduction

2 Kernel methods and feature space

3 Mean element and covariance operator

4 Kernel PCA

5 Kernel CCA

6 Spectral Clustering
Coordinate-free definitions of mean and covariance

Usual definitions

- need explicit basis to define quantities
  → tricky in high-dimensional/$\infty$-dimensional feature spaces

Coordinate-free definitions

- define quantities through their projections along any direction
  → allow direct application of the *reproducing property*
Mean vector and mean element

Empirical mean element

Empirical mean vector $\hat{\mu}$ of $X_1, \ldots, X_m \sim \mathbb{P}$

$$\forall w \in \mathcal{X}, \quad (\hat{\mu}, w) \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} (x_\ell, w)$$

Empirical mean element $\hat{\mu}$ of $X_1, \ldots, X_m \sim \mathbb{P}$

$$\forall f \in \mathcal{H}, \quad \langle \hat{\mu}, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} \langle \phi(x_\ell), f \rangle_{\mathcal{H}}$$
Mean vector and mean element

Empirical mean element

Empirical *mean element* $\hat{\mu}$ of $x_1, \ldots, x_m \sim \mathbb{P}$

\[
\forall f \in \mathcal{H},
\langle \hat{\mu}, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} \langle \phi(x_{\ell}), f \rangle_{\mathcal{H}}
\]

\[
\langle \hat{\mu}, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} \langle k(x_{\ell}, \cdot), f \rangle_{\mathcal{H}}
\]

\[
\overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} f(x_{\ell}) \text{ (reproducing property)}
\]

\[
\overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} \sum_{j=1}^{n} \alpha_j k(x_j, x_{\ell}) , \text{ if } f(\cdot) = \sum_{j=1}^{n} \alpha_j k(x_j, \cdot)
\]
Centering in feature space

Gram matrix
\[ \mathbf{K} = [k(X_i, X_j)]_{i,j=1,...,n} \] of all evaluations of the kernel \( k(\cdot, \cdot) \) on the sample \( x_1, \ldots, x_n \).

Centering in feature space
To center all \( \phi(x_1), \ldots, \phi(x_n) \) simultaneously, do

\[ \mathbf{K} \leftarrow \tilde{\mathbf{K}} = \Pi \mathbf{K} \Pi, \]

where

\[ \Pi = \mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T. \]
Empirical covariance operator

Empirical covariance matrix $\hat{\Sigma}$ of $x_1, \ldots, x_m \sim \mathcal{P}$

$$\forall w, v \in \mathcal{X},$$

$$(w, \hat{\Sigma}v) = \frac{1}{m} \sum_{\ell=1}^{m} (w, \tilde{x}_\ell)(\tilde{x}_\ell, v)$$

$$\tilde{x}_\ell = x_\ell - \hat{\mu}.$$
Covariance operator

Empirical covariance operator $\hat{\Sigma}$ of $x_1, \ldots, x_m \sim \mathbb{P}$

\[
\forall f, g \in \mathcal{H},
\langle f, \hat{\Sigma} g \rangle = \frac{1}{m} \sum_{\ell=1}^{m} \langle f, \tilde{\phi}(x_{\ell}) \rangle \langle \tilde{\phi}(x_{\ell}), g \rangle
= \frac{1}{m} \sum_{\ell=1}^{m} \{ f(x_{\ell}) - \langle \hat{\mu}, f \rangle_{\mathcal{H}} \} \{ f(x_{\ell}) - \langle \hat{\mu}, g \rangle_{\mathcal{H}} \}.
\]
Computing variance along a direction in feature space

Gram matrix

\[ \mathbf{K} = [k(X_i, X_j)]_{i,j=1,...,n} \] of all evaluations of the kernel \( k(\cdot, \cdot) \) on \( x_1, \ldots, x_n \).

Covariance along two directions

\[
\langle f, \hat{\Sigma} g \rangle = \frac{1}{m} \alpha^T \tilde{\mathbf{K}} \tilde{\mathbf{K}} \beta ,
\]

where

\[
f(\cdot) = \sum_{j=1}^{n} \alpha_j k(x_j, \cdot),
\]

\[
g(\cdot) = \sum_{j=1}^{n} \beta_j k(x_j, \cdot).
\]
Mean element and covariance operator

Population mean element and covariance operator

Population mean element \( \mu \) and population covariance operator \( \Sigma \) of \( \mathbf{x} \sim \mathbb{P} \)

\[
\langle \mu, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \mathbb{E}[f(\mathbf{x})], \quad \forall f \in \mathcal{H}
\]

\[
\langle f, \Sigma g \rangle_{\mathcal{H}} \overset{\text{def}}{=} \text{Cov}[f(\mathbf{x}), g(\mathbf{x})], \quad \forall f, g \in \mathcal{H}
\]

Empirical mean element and covariance operator

Empirical mean element \( \hat{\mu} \) and empirical covariance operator \( \hat{\Sigma} \) of \( \mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbb{P} \)

\[
\langle \hat{\mu}, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} f(\mathbf{x}_\ell), \quad \forall f \in \mathcal{H}
\]

\[
\langle f, \hat{\Sigma} g \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} \left\{ f(\mathbf{x}_\ell) - \langle \hat{\mu}, f \rangle_{\mathcal{H}} \right\} \left\{ f(\mathbf{x}_\ell) - \langle \hat{\mu}, g \rangle_{\mathcal{H}} \right\}, \quad \forall f, g \in \mathcal{H}
\]
Some casual considerations before the real stuff

Supervised learning

- least-square regression, kernel ridge regression, multilayer-perceptron
  → tackled through (possibly a sequence of) linear of systems
- Operation \ in Matlab/Octave

Unsupervised learning

- (kernel) principal component analysis, (kernel) canonical correlation analysis, spectral clustering
  → tackled through (possibly a sequence of) eigenvalue problems
- Function eigs in Matlab/Octave
Outline

1 Introduction

2 Kernel methods and feature space

3 Mean element and covariance operator

4 Kernel PCA

5 Kernel CCA

6 Spectral Clustering
Principal Component Analysis (PCA)

A brief refresher

- Let $x_1, \ldots, x_n$ a dataset of points in $\mathbb{R}^d$
- PCA is a classical method in multivariate statistics to define a set of orthogonal directions, called *principal components*, that capture the maximum variance
- Projection along the first 2-3 principal components allows to visualize the dataset
Refresher on Principal Component Analysis

Computational aspects

- Maximum variance criterion corresponds to a Rayleigh quotient
- PCA boils down to an eigenvalue problem on the centered covariance matrix $\hat{\Sigma}$ of the dataset, i.e. the principal components $w_1, \ldots, w_d$ are the eigenvectors of $\hat{\Sigma}$ (assuming $n > d$)
- Computational complexity: $O(ndc)$ in time with a *Singular Value Decomposition* (SVD; see eigs in Matlab/Octave), with $n$ the number of points, $d$ the dimension, $c$ the number of principal components retained; stochastic approximation version for nonstationary/large-scale datasets.
Variance along a direction and Rayleigh quotients

**Variance along a direction**

PCA seeks for directions $w_1, \ldots, w_c$ such that

$$w_j = \arg\max_{w \in \mathbb{R}^d; w_j \perp \{w_1, \ldots, w_{j-1}\}} \frac{\text{Var}_{\text{emp}}(w, x)}{(w, w)}$$

$$= \arg\max_{w \in \mathbb{R}^d; w_j \perp \{w_1, \ldots, w_{j-1}\}} \frac{1}{m} \sum_{i=1}^{m} \frac{(w, x_i)^2}{(w, w)}$$

$$= \arg\max_{w \in \mathbb{R}^d; w_j \perp \{w_1, \ldots, w_{j-1}\}} \frac{(w, \hat{\Sigma} w)}{(w, w)} .$$

Principal components $w_1, \ldots, w_c$ are the first $c$ eigenvectors of $\hat{\Sigma}$. 
Kernel PCA

Variance along a direction and Rayleigh quotients

Variance along a direction

KPCA seeks for directions $f_1, \ldots, f_c$ such that

$$f_j = \arg\max_{f \in \mathcal{H}; f_j \perp \{f_1, \ldots, f_{j-1}\}} \text{Var}_{\text{emp}} \frac{\langle f, \phi(x) \rangle}{\langle f, f \rangle}$$

$$= \arg\max_{f \in \mathcal{H}; f_j \perp \{f_1, \ldots, f_{j-1}\}} \frac{1}{m} \sum_{i=1}^{m} \frac{\langle f, \phi(x_i) \rangle^2}{\langle f, f \rangle}$$

$$= \arg\max_{f \in \mathcal{H}; f_j \perp \{f_1, \ldots, f_{j-1}\}} \frac{\langle f, \hat{\Sigma} f \rangle}{\langle f, f \rangle}.$$ Rayleigh quotient

Principal components $f_1, \ldots, f_c$ are the first $c$ eigenvectors of $\hat{\Sigma}$. Is that it?
Rescue theorems

Properties of covariance operators
RKHS Covariance operators are (Zwald et al., 2005, Harchaoui et al., 2008)
- self-adjoint (∞-dimensional counterpart of symmetric)
- positive
- trace-class

Consequence
The covariance operator $\hat{\Sigma}$ and the centered Gram matrix $\tilde{K}$ share the same eigenvalues on the nonzero part of their spectra, and their eigenvectors are related by a simple relation.
Kernel Principal Component Analysis

KPCA algorithm

- Center the Gram matrix
- Performs an SVD on $\tilde{K}$ to get the first $c$ eigenvector/eigenvalue pairs $(e_j, \lambda_j)_{j=1,...,c}$.
- Normalize the eigenvector $\tilde{e}_j \leftarrow e_j / \lambda_j$
- Projections onto the $j$-th eigenvectors is given by $\tilde{K}\tilde{e}_j$
Computational aspects of KPCA

Computational aspects

- Maximum variance in feature space corresponds to a Rayleigh quotient.
- KPCA boils down to an eigenvalue problem involving the centered auto-covariance matrices $\tilde{K}$.
- Computational complexity: $O(cn^2)$ in time with a *Singular Value Decomposition* (SVD; see `eigs` in Matlab/Octave), with $n$ the number of points, $c$ the number of principal components retained; stochastic approximation version for nonstationary/large-scale datasets.
Low-dimensional representation with KPCA

Human body pose representation

- Walking sequence of length 400 (containing about 3 walking cycles) obtained from the CMU MoCap database
- Data: silhouette images of size (160 100) taken at a side view

Human body pose representation (Kim & Pavlovic, 2008)
Low-dimensional representation with KPCA

Human body pose representation
Low-dimensional representation with KPCA

Human body pose representation

Figure: Central subspaces for silhouette images from walking motion
Super-resolution with KPCA (Kim et al., 2005)

Super-resolution

![Super-resolution figures](fig: Kim et al., PAMI 2005.)
KPCA+n : unsupervised alignment (de la Torre & Nguyen, 2009)

Unsupervised alignment

KPCA + Rigid motion model

Figure: original USPS (a), Parametrized KPCA (b), congealing (c)
Kernel PCA

Applications

Popular

- Image denoising (digits, faces, etc.)
- Visualization of bioinformatics data (strings, proteins, etc.)
- Dimension-reduction of high-dimensional features (appearance, interest points, etc.)

Not so well-know property of KPCA

- Regularization in supervised learning can be enforced by projection → careful not to regularize twice!
- Useful in settings where ridge-regularization is impractical (Zwald et al., 2009; Harchaoui et al., 2009; Guillaumin et al., 2010)
Outline

1. Introduction
2. Kernel methods and feature space
3. Mean element and covariance operator
4. Kernel PCA
5. Kernel CCA
6. Spectral Clustering
Kernel Canonical Correlation Analysis (Shawe-Taylor & Cristianini, 2004)

Canonical Correlation Analysis (CCA)

A brief refresher

- Let \((x_1, y_1), \ldots, (x_n, y_n)\) a dataset of points in \(\mathbb{R}^d \times \mathbb{R}^p\), for which two views are available: the "x-view" and the "y-view"

- CCA is a classical method from multivariate statistics to define a set of pairs of orthogonal directions, called canonical variates, that capture the maximum correlation between the two views.

- Projection along the first 2-3 pairs of canonical variates resp. of "x-view" and the "y-view" allows to visualize the components dataset maximizing the correlation between the two views.
Computational aspects

- Maximum correlation criterion corresponds to a generalized Rayleigh quotient.
- CCA boils down to a generalized eigenvalue problem involving the (centered) auto-covariance matrices $\hat{\Sigma}_{xx}$ and $\hat{\Sigma}_{yy}$ and on the (centered) cross-covariance matrix $\hat{\Sigma}_{xy}$.
- Computational complexity: $O(n(d + p)c)$ in time with a Singular Value Decomposition (SVD; see `eigs` in Matlab/Octave), with $n$ the number of points, $d$ the dimension, $c$ the number of canonical variates retained; stochastic approximation version for nonstationary/large-scale datasets.
Cross-covariance matrix and cross-covariance operator

Empirical cross-covariance matrix

Empirical cross-covariance matrix $\hat{\Sigma}_{xy}$ of $x_1, \ldots, x_m \sim P_x$ and $y_1, \ldots, y_m \sim P_y$

$$\forall w, v \in \mathcal{X}, \mathcal{Y}$$

$$(w, \hat{\Sigma}_{xy} v) = \frac{1}{m} \sum_{\ell=1}^{m} (w, \tilde{x}_\ell)(\tilde{y}_\ell, v)$$

$$\tilde{x}_\ell = x_\ell - \hat{\mu}_x$$

$$\tilde{y}_\ell = y_\ell - \hat{\mu}_y.$$
Covariance along two directions and generalized Rayleigh quotients

Covariance along two directions
CCA seeks for directions \((w_1, v_1)\) such that\(^1\)

\[
(w_1, v_1) = \arg\max_{(w, v) \in \mathbb{R}^d \times \mathbb{R}^p} \frac{\text{Cov}((w, x), (v, y))}{\text{Var}^{1/2}((w, x)) \text{Var}^{1/2}((v, y))} = \arg\max_{(w, v) \in \mathbb{R}^d \times \mathbb{R}^p} \frac{(w, \hat{\Sigma}_{xy} v)}{(w, \hat{\Sigma}_{xx} w)^{1/2} (v, \hat{\Sigma}_{yy} v)^{1/2}}.
\]

---

1. focus here on the first pair of canonical variates
Covariance along two directions and generalized Rayleigh quotients

Generalized Rayleigh quotient
Canonical variates \((w_1, v_1), \ldots, (w_c, v_c)\) are the first \(c\) pairs of vectors solutions of the generalized eigenvalue problem

\[
\begin{bmatrix}
0 & \hat{\Sigma}_{xy} \\
\hat{\Sigma}_{xy} & 0
\end{bmatrix}
\begin{bmatrix}
w \\
v
\end{bmatrix}
= \rho
\begin{bmatrix}
\hat{\Sigma}_{xx} & 0 \\
0 & \hat{\Sigma}_{yy}
\end{bmatrix}
\begin{bmatrix}
w \\
v
\end{bmatrix}.
\]
Covariance along two directions and generalized Rayleigh quotients

**Covariance along two directions**

Kernel CCA seeks for directions \((f_1, g_1)\) such that \(^2\)

\[
(f_1, g_1) = \arg\max_{(f, g) \in \mathcal{H} \times \mathcal{H}} \frac{\text{Cov}(\langle f, \phi(x) \rangle, \langle g, \psi(y) \rangle)}{\{\text{Var} \langle f, \phi(x) \rangle + \epsilon \langle f, f \rangle\}^{1/2} \{\text{Var} \langle g, \psi(x) \rangle + \epsilon \langle g, g \rangle\}^{1/2}}
\]

\[
= \arg\max_{(f, g) \in \mathcal{H} \times \mathcal{H}} \frac{\langle f, \hat{\Sigma}_{xy} g \rangle}{\langle f, (\hat{\Sigma}_{xx} + \frac{\epsilon n}{2}) g \rangle^{1/2} \langle f, (\hat{\Sigma}_{yy} + \frac{\epsilon n}{2}) g \rangle^{1/2}}.
\]

---

2. focus here on the first pair of canonical variates
Correlation along two directions

Generalized eigenvalue problem

Coefficients of canonical variates \((\alpha_1, \beta_1), \ldots, (\alpha_c, \beta_c)\) are the first \(c\) pairs of vectors solutions of the generalized eigenvalue problem

\[
\begin{bmatrix}
0 & \tilde{K}_x \tilde{K}_y \\
\tilde{K}_x \tilde{K}_y & 0
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta
\end{bmatrix}
= \rho
\begin{bmatrix}
\tilde{K}_x \tilde{K}_x & 0 \\
0 & \tilde{K}_y \tilde{K}_y
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta
\end{bmatrix}.
\]
Computational aspects of KCCA

Computational aspects

- Maximum correlation in feature space corresponds to a Rayleigh quotient.
- KCCA boils down to a generalized eigenvalue problem involving the squared centered Gram matrices $\tilde{K}_x^2 \tilde{K}_y^2$ and the product of the Gram matrices $\tilde{K}_x \tilde{K}_y$.
- Computational complexity: $O(cn^2)$ in time with a Singular Value Decomposition (SVD; see eigs in Matlab/Octave), with $n$ the number of points, $c$ the number of principal components retained; stochastic approximation version for nonstationary/large-scale datasets.
Multimedia content based image retrieval with KCCA

Multimedia

- Multimedia content → multi-view data
- Images with text captions: text → “x”-view, image → “y”-view

Multimedia content based image retrieval (Hardoon et al, 2004)

<table>
<thead>
<tr>
<th>Image</th>
<th>Label</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>Sports</td>
<td>position, college, weight, born, lbs, height, guard</td>
</tr>
<tr>
<td>$I_2$</td>
<td>Aviation</td>
<td>na, air, convair, wing</td>
</tr>
<tr>
<td>$I_3$</td>
<td>Paintball</td>
<td>check, darkside, force, gog, strike, odt</td>
</tr>
</tbody>
</table>
Outline

1 Introduction
2 Kernel methods and feature space
3 Mean element and covariance operator
4 Kernel PCA
5 Kernel CCA
6 Spectral Clustering
Spectral clustering (von Luxburg, 2007)

Overview

- Let $x_1, \ldots, x_n$ a dataset of points in $\mathbb{R}^d$, along with pairwise similarities $s(x_i, x_j), 1 \leq i, j \leq n$.
- Build similarity graph, with data points as vertices and similarities as edge lengths.
- Spectral clustering finds the best cut through the graph.
Laplacian matrix and spectral clustering

Laplacian matrix

Spectral clustering relies on the spectrum of the Laplacian matrix $L$

$$L = D - S,$$

where

$$D = \text{Diag}(\text{deg}(x_1), \ldots, \text{deg}(x_n))$$

and

$$\text{deg}(x_i) = \sum_{j=1}^{n} s(x_i, x_j).$$
Laplacian matrix and the Laplace-Beltrami operator

Laplacian matrix

The Laplacian matrix measures the discrete variation of $f$ along the graph

$$\forall f \in \mathbb{R}^d, f^T \mathbf{L} f = \frac{1}{2} \sum_{j=1}^{n} s(x_i, x_j) (f_i - f_j)^2 ,$$

$$f^T \mathbf{L} f \approx \frac{1}{2} \sum_{j=1}^{n} \frac{(f_i - f_j)^2}{d(x_i, x_j)^2} , \text{ if } s(x_i, x_j) \approx \frac{1}{d(x_i, x_j)^2} .$$

Laplacian operator

The Laplacian matrix is the discrete counterpart of the Laplace$^3$ operator

$$\forall f \in \mathbb{R}^d, \langle f, \Delta f \rangle = \int |\nabla f|^2 dx .$$

3. Laplace-Beltrami generalizes the Laplace operator to manifold data.
Rescue theorems

Properties of Laplacian operators
Laplacian matrices are (von Luxburg et al., 2005, Gine and Koltchinskii, 2008)

- symmetric
- positive definite
- smallest eigenvalue is 0, and associated eigenvector 1

Interpretation

- Multiplicity of eigenvalue 0 is the number of connected components of the graph $A_1, \ldots, A_k$
- Eigenspace spanned by the characteristic functions $1_{A_1}, \ldots, 1_{A_k}$ on those components (so all eigenvectors are piecewise constants)
Normalization

Normalized graph Laplacians

Graph Laplacian matrices can be normalized in two ways:\(^4\)

\[
L_{rw} = D^{-1}L \quad \text{random walk normalization},
\]

\[
L_{sym} = D^{-1/2}LD^{-1/2} \quad \text{symmetrized normalization}.
\]

Interpretation

- \(L_{rw}\) and \(L_{sym}\) share similar spectral properties with \(\Lambda\)
- Normalized graph Laplacians are better understood theoretically and are consistent under general assumptions in large-sample settings
- Un-normalized ones are still used (!) despite their lack of consistency in some cases in large-sample settings.

\(^4\) Caution: eigenspace of \(L_{rw}\) spanned by the \(1_{A_1}, \ldots, 1_{A_k}\); eigenspace of \(L_{sym}\) spanned by the \(D^{1/2}1_{A_1}, \ldots, D^{1/2}1_{A_k}\).
Spectral clustering algorithm

- Build similarity graph
- Performs an SVD on $L_{rw}$ or $L_{sym}$ to get the first $k$ eigenvector/eigenvalue pairs $(v_j, \lambda_j)_{j=1,...,c}$.
- Build the matrix $V = [v_1, \ldots, v_k]$ stacking the $k$ eigenvectors as columns
- Launch your *favourite clustering algorithm* on the $n$ rows of $V$
Example

2D example with 3 clusters
Example

Projections onto eigenvectors
Example

Clustering obtained with $k$-means as the favourite clustering algorithm
Spectral clustering for image segmentation

Image segmentation algorithm

Figure 6: Automatic image segmentation. Fully automatic intensity based image segmentation results using our algorithm.

More experiments and results on real data sets can be found on our web-page: http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html

Discussion & Conclusions

Spectral clustering practitioners know that selecting good parameters to tune the clustering process is an art requiring skill and patience. Automating spectral clustering was the main motivation for this study. The key ideas we introduced are three: (a) using a local scale, rather than a global one, (b) estimating the scale from the data, and (c) rotating the eigenvectors to create the maximally sparse representation. We proposed an automated spectral clustering algorithm based on these ideas: it computes automatically the scale and the number of groups and it can handle multi-scale data which are problematic for previous approaches.

Some of the choices we made in our implementation were motivated by simplicity and are perfectible. For instance, the local scale might be better estimated by a method which relies on more informative local statistics. Another example: the cost function in Eq. (3) is reasonable, but by no means the only possibility (e.g. the sum of the entropy of the rows might be used instead).

Acknowledgments:

Finally, we wish to thank Yair Weiss for providing us his code for spectral clustering.

This research was supported by the MURI award number SA3318 and by the Center of Neuromorphic Systems Engineering award number EEC-9402726.

References

GrabCut and foreground extraction

Interactive foreground extraction algorithm
Kernel-based Methods for Unsupervised Learning

LEAR project-team, INRIA

Zaid Harchaoui

Grenoble, July 30th 2010
Outline

1. Discriminative clustering

2. Temporal Segmentation
Outline

1 Discriminative clustering

2 Temporal Segmentation
Summary

- **Discriminative clustering** = find labels that maximize linear separability
- **Multiclass square loss** for classification = cost function in closed form
- Optimization of the labels by **convex relaxation**
- Efficient optimization algorithm by **partial dualization**
- Application in **semi-supervised learning**
Discriminative clustering

Classification with square loss

- $n$ points $x_1, \ldots, x_n$ in $\mathbb{R}^d$, represented in a matrix $X \in \mathbb{R}^{n \times d}$.
- Labels = partitions of $\{1, \ldots, n\}$ into $k > 1$ clusters, represented by indicator matrices

$$y \in \{0, 1\}^{n \times k} \text{ such that } y 1_k = 1_n$$

- Regularized linear regression problem of $y$ given $X$:

$$J(y, X, \kappa) = \min_{w \in \mathbb{R}^{d \times k}, b \in \mathbb{R}^{1 \times k}} \frac{1}{n} \|y - Xw - 1_n b\|_F^2 + \kappa \text{ Tr } w^\top w,$$

- Multi-label classification problems with square loss functions
- Solution in closed form (with $\Pi_n = I_n - \frac{1}{n} 1_n 1_n^\top$):

$$w^* = (X^\top \Pi_n X + n\kappa I_n)^{-1} X^\top \Pi_n y \quad \text{and} \quad b^* = \frac{1}{n} 1_n^\top (y - Xw^*)$$
Discriminative clustering consists in finding labels such that they lead to best linear separation by a discriminative classifier (Xu et al., 2004, 2005)

- Use square loss for multi-class classification
- Main advantages
  - minimizing the regularized cost in closed form
  - including a bias term by simply centering the data
- Optimal value equal to $J(y, X, \kappa) = \text{Tr} \; yy^\top A(X, \kappa)$, where

$$A(X, \kappa) = \frac{1}{n} \Pi_n (I_n - X(X^\top \Pi_n X + n\kappa I)^{-1} X^\top) \Pi_n$$
Optimization problem: minimize $\text{Tr}yy^\top A(X, \kappa)$ with respect to $y$ (indicator matrices)

The cost function only involves the matrix $M = yy^\top \in \mathbb{R}^{n \times n} = k$-class equivalence matrix $\in \{0, 1\}^{n \times n}$

Convex outer approximation for $M$

- $M$ is positive semidefinite (denoted as $M \succeq 0$)
- the diagonal of $M$ is equal to $1_n$ (denoted as $\text{diag}(M) = 1_n$)
- if $M$ corresponds to at most $k$ clusters, we have $M \succeq \frac{1}{k}1_n1_n^\top$

Convex set:

$$C_k = \{M \in \mathbb{R}^{n \times n}, M = M^\top, \text{diag}(M) = 1_n, M \succeq 0, M \succeq \frac{1}{k}1_n1_n^\top\}$$
Minimum cluster sizes

- Avoid trivial solution by imposing a minimum size $\lambda_0$ for each cluster, through:
  - **Row sums**: $M1_n \geq \lambda_0 1_n$ and $M1_n \leq (n - (k - 1)\lambda_0)1_n$ (same constraint as Xu et al., 2005).
  - **Eigenvalues**: The sizes of the clusters are exactly the $k$ largest eigenvalues of $M \Rightarrow$ constraint equivalent to $\sum_{i=1}^{n} \lambda_i(M) \geq \lambda_0 \geq k$, where $\lambda_1(M), \ldots, \lambda_n(M)$ are the $n$ eigenvalues of $M$.
    - Non convex constraint
    - Relaxed as $\sum_{i=1}^{n} \phi_{\lambda_0}(\lambda_i(M)) \geq k$, where $\phi_{\lambda_0}(\kappa) = \min\{\kappa/\lambda_0, 1\}$

- **Final convex relaxation**: minimize $\text{Tr}A(X, \kappa)M$ such that
  $M = M^\top$, $\text{diag}(M) = 1_n$, $M \succeq 0$, $M \succeq 1/\kappa 1_n 1_n^\top$,
  $\sum_{i=1}^{n} \phi_{\lambda_0}(\lambda_i(M)) \geq k$
Comparison with K-means

- **DIFFRAC** \((\kappa = 0)\) : minimize
  \[
  \text{Tr } \Pi_n(I_n - X(X^\top \Pi_n X)^{-1}X^\top)\Pi_n y y^\top
  \]

  \[
  \min_{\mu \in \mathbb{R}^{k \times d}} \|X - y\mu\|_F^2 = \text{Tr}(I_n - y(y^\top y)^{-1}y^\top)(\Pi_n X)(\Pi_n X)^\top
  \]
The matrix $A(X, \kappa)$ can be expressed only in terms of the Gram matrix $K = XX^\top$. 

$$A(K, \kappa) = \kappa \Pi_n (\tilde{K} + n\kappa I_n)^{-1} \Pi_n$$

where $\tilde{K} = \Pi_n K \Pi_n$ is the “centered Gram matrix” of the points $X$.

Additional relaxation to kernel PCA:

1. relaxing the constraints $M \succeq \frac{1}{k} 1_n 1_n^\top$ into $M \succeq 0$
2. relaxing $\text{diag}(M) = 1_n$ into $\text{Tr} M = n$
3. removing the constraint $M \succeq 0$ and the constraints on the row sums.

Important constraint: $\text{diag}(M) = 1_n$
Optimization problem:

$$\min \text{Tr} AM \quad \text{such that} \quad M = M^\top, \ M \succeq 0, \ \text{Tr} M = n$$

$$\Phi_{\lambda_0}(M) = \sum_{i=1}^{n} \phi_{\lambda_0}(\lambda_i(M)) \geq k$$

$$\text{diag}(M) = 1_n$$

$$M1_n \leq (n - (k - 1)\lambda_0)1_n, \ M1_n \geq \lambda_01_n$$

$$M \succeq 0$$

$$M \succeq \frac{1_n1_n^\top}{k}$$

Partial dualization of constraints:

- Kept constraints lead to simple spectral problem.
Lagrangian equal to $\text{Tr} B(\beta) M - b(\beta)$ with

$$B(\beta) = A + \text{Diag}(\beta_1) - \frac{1}{2}(\beta_2 - \beta_3)1^\top - \frac{1}{2}1(\beta_2 - \beta_3)^\top - \beta_4 + \frac{1}{2}\frac{\beta_5\beta_5^\top}{\beta_6}$$

$$b(\beta) = \beta_1^\top 1 - (n - (k - 1)\lambda_0)\beta_2^\top 1 + \lambda_0\beta_3^\top 1 + k\beta_6/2 + \beta_5^\top 1$$

- Primal variable $M$, dual variables $\beta_1$, $\beta_2$, $\beta_3$, $\beta_4$, $(\beta_5, \beta_6)$

- Dual problem: $\max_{\beta} \left\{ \min_{M \succeq 0, \text{Tr} M = n, \Phi_{\lambda_0}(M) \geq k} \text{Tr} B(\beta) M - b(\beta) \right\}$

- Minimization with respect to $M$ leads to convex non differentiable spectral function in $\beta$

- Maximization with respect to $\beta$ by projected subgradient or projected gradient (after smoothing)
Constant times the matrix-vector operation with the matrix $A$

- **Linear complexity** in the number $n$ of data points.
- For linear kernels with dimension $d : O(d^2n)$
- For general kernels : $O(n^3)$ or $O(m^2n)$ using an incomplete Cholesky decomposition of rank $m$

- **Rounding**
  - After the convex optimization, we obtain a low-rank matrix $M \in C_k$ which is pointwise nonnegative with unit diagonal
  - Spectral clustering algorithm on the matrix $M$ (Ng & al., 2001)
  - NB : Diffrac works better than just doing spectral clustering on $A$ or $K$!
Semi-supervised learning

- Equivalence matrices $M$ allow simple inclusion of prior knowledge (Xu et al., 2004, De Bie and Cristianini, 2006)
- “must-link” constraints (positive constraints) : $M_{ij} = 1$
  - With a square loss $\Rightarrow$ equivalent to grouping into chunks
- “must-not-link” constraints (negative constraints) : $M_{ij} = 0$

![Graph showing clustering error vs noise dimension for K-means and diffrac methods at 20% and 40% noise levels.](image)
Simulations

- Clustering classification datasets
  - Performance measured by clustering error between 0 and $100(k - 1)$
  - Comparison with K-means and RCA (Bar-Hillel et al., 2003)
  - Different amount of labelled data (0 to 40%)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>K-means</th>
<th>Dirfrac</th>
<th>RCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mnist-linear 0%</td>
<td>5.6 ± 0.1</td>
<td>6.0 ± 0.4</td>
<td>3.0 ± 0.2</td>
</tr>
<tr>
<td>Mnist-linear 20%</td>
<td>4.5 ± 0.3</td>
<td>3.6 ± 0.3</td>
<td>1.8 ± 0.4</td>
</tr>
<tr>
<td>Mnist-linear 40%</td>
<td>2.9 ± 0.3</td>
<td>2.2 ± 0.2</td>
<td></td>
</tr>
<tr>
<td>Mnist-RBF 0%</td>
<td>5.6 ± 0.2</td>
<td>4.9 ± 0.2</td>
<td></td>
</tr>
<tr>
<td>Mnist-RBF 20%</td>
<td>4.6 ± 0.0</td>
<td>1.8 ± 0.4</td>
<td>4.1 ± 0.2</td>
</tr>
<tr>
<td>Mnist-RBF 40%</td>
<td>4.9 ± 0.0</td>
<td>0.9 ± 0.1</td>
<td>2.9 ± 0.1</td>
</tr>
<tr>
<td>Isolet-linear 0%</td>
<td>12.1 ± 0.6</td>
<td>12.3 ± 0.3</td>
<td></td>
</tr>
<tr>
<td>Isolet-linear 20%</td>
<td>10.5 ± 0.2</td>
<td>7.8 ± 0.8</td>
<td>9.5 ± 0.4</td>
</tr>
<tr>
<td>Isolet-linear 40%</td>
<td>9.2 ± 0.5</td>
<td>3.7 ± 0.2</td>
<td>7.0 ± 0.4</td>
</tr>
<tr>
<td>Isolet-RBF 0%</td>
<td>11.4 ± 0.4</td>
<td>11.0 ± 0.3</td>
<td></td>
</tr>
<tr>
<td>Isolet-RBF 20%</td>
<td>10.6 ± 0.0</td>
<td>7.5 ± 0.5</td>
<td>7.8 ± 0.5</td>
</tr>
<tr>
<td>Isolet-RBF 40%</td>
<td>10.0 ± 0.0</td>
<td>3.7 ± 1.0</td>
<td>6.9 ± 0.6</td>
</tr>
</tbody>
</table>
Simulations

- Semi-supervised classification
  - Diffrac “works” with any amount of supervision
  - Comparison with LDS (Chapelle & Zien, 2004)

![Learning curve on Coil100](image_url)
Extension to images co-segmentation (Joulin et al., 2010)

Natural images
Extension to images co-segmentation (Joulin et al., 2010)

Cycles and horses
Outline

1. Discriminative clustering

2. Temporal Segmentation
Temporal segmentation (clustering with temporal consistency)

Change-in-mean model

Time series of independent r.v. \( \{Y_t\}_{t=1,...,n} \) such that

\[
Y_t \sim \mathcal{N}(\mu^*_k, \sigma^2), \quad t_{k-1} + 1 \leq t \leq t^*_k, \quad k = 1, \ldots, K^* + 1, \quad (1)
\]
Temporal Segmentation

Change-in-mean-element model

Time series of independent r.v. \( \{Y_t\}_{t=1,...,n} \) such that

\[
\mathbb{E}[k(Y_t, \cdot)] = \mu^*_k, \quad t_{k-1}^* + 1 \leq t \leq t_k^*, \quad k = 1, \ldots, K^* + 1.
\]
Temporal segmentation with kernels

Classical least-squares formulation

Minimize $t_1, \ldots, t_{K^*}$

$$\sum_{k=1}^{K^*+1} \sum_{t=t_{k-1}+1}^{t_k} (Y_t - \overline{Y}(t_{k-1}, t_k))^2$$

Kernel-based version in $\mathcal{H}$

Minimize $t_1, \ldots, t_{K^*}$

$$\sum_{k=1}^{K^*+1} \sum_{t=t_{k-1}+1}^{t_k} \|k(Y_t, \cdot) - \hat{\mu}[t_{k-1}; t_k]\|^2_{\mathcal{H}}$$
Massaging the objective function

Intra-segment scatter

Minimize

\[
\sum_{k=1}^{K-1} \hat{V}(Y_{t_k+1}, \ldots, Y_{t_k+1})
\]

with \( \hat{V}(Y_{t+1}, \ldots, Y_{t+s}) = \| k(Y_t, \cdot) - \hat{\mu}_{[t+1:t+s]} \|^2_H \)
Forward-backward recursions

Forward recursions

\[ I_k(t) = \min_{t_1, \ldots, t_{k-1}; t_k = t} \sum_{k=1}^{K-1} \hat{V}(Y_{tk+1}, \ldots, Y_{tk+1}) \]

\[ = \min_{t_{k-1}; t_k = t} \min_{t_1, \ldots, t_{k-2}} \sum_{k=1}^{K-1} \hat{V}(Y_{tk+1}, \ldots, Y_{tk+1}) \]

\[ = \min_{t_{k-1}} (I_{k-1}(t_{k-1}) + \hat{V}(Y_{tk-1}, \ldots, Y_t)) . \]

Dynamic programming

Dynamic programming algorithm working on submatrices of the Gram matrix, leading to a time-complexity of \( O(Kn^2) \).
Kernel-based Methods for Unsupervised Learning

LEAR project-team, INRIA

Zaid Harchaoui

Grenoble, July 30th 2010
Outline

1 Introduction

2 Homogeneity testing

3 Change-point Analysis
Outline

1. Introduction

2. Homogeneity testing

3. Change-point Analysis
Kernel methods

Machine Learning methods taking $\mathbf{K} = [k(X_i, X_j)]_{i,j=1,...,n}$ (Gram matrix as input for processing a sample $\{X_1, \ldots, X_n\}$, where $k(x, y)$ is a similarity measure between $x$ and $y$ defining a positive definite kernel.

Strengths of Kernel Methods

- Minimal assumptions on data types (vectors, strings, trees, graphs, etc.)
- Interpretation of $k(x, y)$ as a dot product $k(x, y) = \langle \phi(x), \phi(y) \rangle_\mathcal{H}$ in a reproducing kernel Hilbert space $\mathcal{H}$ where the observations are mapped via $[\phi: \mathcal{X} \rightarrow \mathcal{H}]$ (feature map)
Mean element and covariance operator

Population mean element and covariance operator

Population mean element $\mu$ and population covariance operator $\Sigma$ of $X \sim \mathbb{P}$

$$\langle \mu, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \mathbb{E}[f(X)], \quad \forall f \in \mathcal{H}$$

$$\langle f, \Sigma g \rangle_{\mathcal{H}} \overset{\text{def}}{=} \text{Cov}[f(X), g(X)], \quad \forall f, g \in \mathcal{H}$$

Empirical mean element and covariance operator

Empirical mean element $\hat{\mu}$ and empirical covariance operator $\hat{\Sigma}$ of $X_1, \ldots, X_m \sim \mathbb{P}$

$$\langle \hat{\mu}, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} f(X_\ell), \quad \forall f \in \mathcal{H}$$

$$\langle f, \hat{\Sigma} g \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{m} \sum_{\ell=1}^{m} \{f(X_\ell) - \langle \hat{\mu}, f \rangle_{\mathcal{H}}\}\{f(X_\ell) - \langle \hat{\mu}, g \rangle_{\mathcal{H}}\} \quad \forall f, g \in \mathcal{H}$$
Test for homogeneity

Homogeneity of two samples

- Two samples $X^{(1)}_1, \ldots, X^{(1)}_{n_1} \sim P^{(1)}$ and $X^{(2)}_1, \ldots, X^{(2)}_{n_2} \sim P^{(2)}$ independent
- Problem: decide between

$$H_0 : P^{(1)} = P^{(2)}$$

$$H_A : P^{(1)} \neq P^{(2)}$$
Test statistic

Empirical mean elements $\hat{\mu}_1$ and $\hat{\mu}_2$, and empirical covariance operators $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ resp. $\{X^{(1)}_1, \ldots, X^{(1)}_{n_1}\}$ et $\{X^{(2)}_1, \ldots, X^{(2)}_{n_2}\}$

$\{X^{(1)}_1, \ldots, X^{(1)}_{n_1}\} \hookrightarrow (\hat{\mu}_1, \hat{\Sigma}_1)$ and $\{X^{(2)}_1, \ldots, X^{(2)}_{n_2}\} \hookrightarrow (\hat{\mu}_2, \hat{\Sigma}_2)$.

Regularized Fisher ratio

$$
\text{KFDR}_{n_1,n_2;\gamma}(X^{(1)}_1, \ldots, X^{(1)}_{n_1}; X^{(2)}_1, \ldots, X^{(2)}_{n_2}) \triangleq \frac{n_1n_2}{n_1 + n_2} \left\| \left( \frac{n_1}{n} \hat{\Sigma}_1 + \frac{n_2}{n} \hat{\Sigma}_2 + \gamma I \right)^{-1/2} (\hat{\mu}_2 - \hat{\mu}_1) \right\|_{\mathcal{H}}^2.
$$

Hotelling’s $T^2$: homogeneity of two normal probability distributions with different means and unknown covariance matrices

$$
\frac{n_1n_2}{n_1 + n_2} \left\| \left( \frac{n_1}{n} \hat{\Sigma}_1 + \frac{n_2}{n} \hat{\Sigma}_2 \right)^{-1/2} (\hat{\mu}_2 - \hat{\mu}_1) \right\|_{\mathbb{R}^d}^2.
$$
Large-sample distribution under $H_0$: regime $\gamma_n \equiv \gamma$

Proposition

Assume the kernel is bounded and that for $a = 1, 2$ the eigenvalues
$\{\lambda_p(\Sigma_a)\}_{p \geq 1}$ satisfy $\sum_{p=1}^{\infty} \lambda_p^{1/2}(\Sigma_a) < \infty$. Assume also that $P_1$ and $P_2$ are equal i.e. $P_1 = P_2 = P$, and that $\gamma_n \equiv \gamma > 0$. Then,

$$
\text{KFDR}_{n_1,n_2;\gamma} - d_{1,n_1,n_2;\gamma}(\hat{\Sigma}_n^{W})
\sqrt{2} d_{2,n_1,n_2;\gamma}(\hat{\Sigma}_n^{W})
\xrightarrow{D}
\frac{1}{\sqrt{2} d_{2,n_1,n_2;\gamma}(\Sigma_W)} \sum_{p=1}^{\infty} \frac{\lambda_p(\Sigma_W)}{\lambda_p(\Sigma_W) + \gamma \left( Z_p^2 - 1 \right)} + \frac{\lambda_p(\Sigma_W)}{\lambda_p(\Sigma_W) + \gamma \left( \chi^2_1 \right)}
$$

Remarks

$d_{1,n_1,n_2;\gamma}(\hat{\Sigma}_n^{W}) \overset{\text{def}}{=} \text{Tr}((\hat{\Sigma}_n + \gamma I)^{-1}\hat{\Sigma}_n)$

renormalization

$d_{2,n_1,n_2;\gamma}(\hat{\Sigma}_n^{W}) \overset{\text{def}}{=} [\text{Tr}((\hat{\Sigma}_n + \gamma I)^{-2}\hat{\Sigma}_n^2)]^{1/2}$

recentering
Homogeneity testing

Large-sample distribution under $H_0$: regime $\gamma_n \rightarrow 0$

**Proposition**

Assume the kernel is bounded and that for $a = 1, 2$ the eigenvalues $\{\lambda_p(\Sigma_a)\}_{p \geq 1}$ satisfy $\sum_{p=1}^{\infty} \lambda_p^{1/2}(\Sigma_a) < \infty$. Assume in addition that $P_1 = P_2 = P$, and that $\{\gamma_n\}$ is such that

$$\gamma_n + \frac{d_{1,n_1,n_2;\gamma_n}(\Sigma W)}{d_{2,n_1,n_2;\gamma_n}(\Sigma W)} \gamma_n^{-1} n^{-1/2} \rightarrow 0 .$$

Then,

$$\frac{\text{KFDR}_{n_1,n_2;\gamma_n} - d_{1,n_1,n_2;\gamma_n}(\hat{\Sigma} W_{n_1,n_2})}{\sqrt{2} d_{2,n_1,n_2;\gamma_n}(\hat{\Sigma} W_{n_1,n_2})} \overset{D}{\rightarrow} \mathcal{N}(0, 1) .$$

**Remarks**

- **Typical situation** $\gamma_n \rightarrow 0$ slower than $1/\sqrt{n}$
- **Case** $\lambda_p = p^{-2m}$: $d_{1,n_1,n_2;\gamma_n} \sim \gamma_n^{-1/2m}$ et $d_{2,n_1,n_2;\gamma_n} \sim \gamma_n^{-1/4m}$
Total sample size $n_1 + n_2 = 500$, Gaussian RBF kernel with $\sigma = 1$, $\mathbb{P}^{(1)} = \mathbb{P}^{(2)}$ normal probability distributions
Homogeneity testing

Consistency in power

Proposition
Assume the kernel is bounded and that for \( a = 1, 2 \) the eigenvalues \( \{ \lambda_p(\Sigma_a) \}_{p \geq 1} \) satisfy \( \sum_{p=1}^{\infty} \lambda_p^{1/2}(\Sigma_a) < \infty \), and that the RKHS \( \mathcal{H} \) is dense in \( L^2(\mathbb{P}) \) for all \( \mathbb{P} \). Let \( \mathbb{P}_1 \) and \( \mathbb{P}_2 \) two probability distributions such that \( \mathbb{P}_2 \neq \mathbb{P}_1 \). In both regimes (\( \gamma_n \equiv \gamma \) and \( \gamma_n \to 0 \)), for all \( 0 < \alpha < 1 \)

\[
\mathbb{P}_{\mathcal{H}_A} \left( \frac{\text{KFD}_{n_1,n_2;\gamma_n} - d_{1,n_1,n_2;\gamma}(\hat{\Sigma}_{n_1,n_2}^W)}{\sqrt{2} d_{2,n_1,n_2;\gamma}(\hat{\Sigma}_{n_1,n_2}^W)} > c_{1-\alpha} \right) \to 1. \tag{1}
\]

Remarks
Universal density of the RKHS satisfied for translation-invariant kernels \( k(x, y) = k(x - y) \) such as the Gaussian RBF kernel (Steinwart, 2006; Sriperumbudur et al., 2008).
Consistency against local alternatives

- Framework of local alternatives

\[ H_0 : \ P_1 = P_2^n \]
\[ H_A : \ P_1 \neq P_2^n \]

where \( P_1 \) and \( P_2^n \) get closer as \( n \to \infty \), meaning that the \( \chi^2 \)-divergence

\[ D_{\chi^2}(P_1, P_2^n) \leq \eta_n , \quad \text{as} \ n \to \infty . \]
Illustration: uniform vs. uniform+high-frequency contamination with spline kernels

Figure: Comparison of change in power of KFDA versus MMD as $\gamma = 1, 10^{-1}, \ldots, 10^{-9}$, for local alternatives spanned by the $q$-ième component (from left to right) with $q = 1, 5, 9$. 
Homogeneity testing

Computational aspects

Computation

\[ \left\| (\hat{\Sigma}_W + \gamma_n I)^{-1/2} (\hat{\mu}_2 - \hat{\mu}_1) \right\|_H^2 = \gamma^{-1} \left\{ m_n^T K_n m_n - n^{-1} m_n^T K_n N_n (\gamma I + n^{-1} N_n K_n N_n)^{-1} N_n K_n m_n \right\}. \]

\( K_n = [k(x_i, x_j)]_{i,j=1,...,n} \) is the Gram matrix, \( N_n \) is that intra-class re-centering matrix (each block re-centers each sample), and \( m_n = (m_{n,i})_{1 \leq i \leq n} \) stand for the “vector of mean difference” with \( m_{n,i} = -n_1^{-1} \) pour \( i = 1, \ldots, n_1 \) et \( m_{n,i} = n_2^{-1} \) for \( i = n_1 + 1, \ldots, n_1 + n_2 \)

Computational complexity

\( O((n_1 + n_2)^2) \) is space and \( O((n_1 + n_2)^3) \) in time.
Homogeneity testing

Application: speaker verification

- 8 speakers from the NIST evaluation 2004
- descriptors: MFCC

![ROC Curves Comparison](image)

**Figure:** Comparison ROC curves for speaker verification
Application: audio segmentation

“Grand echiquier” TV-shows archives

- Semantic segmentation (coarse segmentation): applause/film/music/interview
- Speaker segmentation (fine segmentation): Coluche/J. Chancel/F.-R. Duchable/etc.

<table>
<thead>
<tr>
<th></th>
<th>Nb. of sections</th>
<th>Mean duration (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>applause</td>
<td>84</td>
<td>14</td>
</tr>
<tr>
<td>film</td>
<td>29</td>
<td>155</td>
</tr>
<tr>
<td>music</td>
<td>38</td>
<td>194</td>
</tr>
<tr>
<td>speech</td>
<td>188</td>
<td>70</td>
</tr>
<tr>
<td>spk turns</td>
<td>962</td>
<td>6</td>
</tr>
</tbody>
</table>

Table: Data description
Experiments in audio segmentation

Experiences

- sliding-window along the signal
- super-descriptors built from cepstral coefficients
- comparison with unsupervised approaches MMD (Gretton et al., 2004), KCD (Desobry et al., 2005), and supervised HMM (Rabiner et al., 2007)

<table>
<thead>
<tr>
<th></th>
<th>Semantic seg.</th>
<th>Spk seg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>KFDR</td>
<td>0.72</td>
<td>0.63</td>
</tr>
<tr>
<td>MMD</td>
<td>0.71</td>
<td>0.58</td>
</tr>
<tr>
<td>KCD</td>
<td>0.65</td>
<td>0.63</td>
</tr>
<tr>
<td>HMM</td>
<td>0.73</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Table: Precision and recall
Outline

1. Introduction
2. Homogeneity testing
3. Change-point Analysis
Change-point Analysis

Assumption

Time series $X_1, \ldots, X_n$ of independent observations

Change-point Problem

Detection

1) Decide between

$H_0 : \ P_{X_1} = \cdots = P_{X_\theta} = \cdots = P_{X_n}$

$H_A : \text{there exists } 1 < k^* < n \text{ such that}$

$P_{X_1} = \cdots = P_{X_{k^*}} \neq P_{X_{k^*+1}} = \cdots = P_{X_n}$

Estimation

2) Estimate $k^*$ from the sample $\{X_1, \ldots, X_n\}$ if $H_A$ is true.

Change-point Analysis = Change-point Detection + Estimation
Running Maximum Strategy for change-point detection run along the series of observations $X_1, \ldots, X_n$, scanning all change-point candidates $k \in ]1, n[$, in order to catch the true change-point instant $k^*$, for which the segment before change and the segment after change have minimum homogeneity.
Building block for the test statistic: finite-dimensional case

- Time series $X_1, \ldots, X_n \in \mathbb{R}^d$ of independent observations
- For any interval $[i, j] \subset \{2, \ldots, n - 1\}$, define resp. the mean vector $\hat{\mu}_{i:j}$ and the covariance matrix $\hat{\Sigma}_{i:j}$.
- For any instant $k \in \{2, \ldots, n - 1\}$,

$$T_{n,k}(X_1, \ldots, X_n) \overset{\text{def}}{=} \frac{k(n-k)}{n} \left\| \left( \frac{k}{n} \hat{\Sigma}_{1:k} + \frac{n-k}{n} \hat{\Sigma}_{k+1:n} \right)^{-1/2} (\hat{\mu}_{k+1:n} - \hat{\mu}_{1:k}) \right\|_2^2.$$

- Null distribution

$$\max_{a_n < k < b_n} T_{n,k}(X_1, \ldots, X_n) \xrightarrow{\text{D}} \max_{u < t < v} \frac{\sum_{p=1}^{d} B_p^2(t)}{t(1-t)}$$

- Consistency in Power (see James, James, Siegmund, 1987)
Building block for the test statistic: kernelized case

- Time series $X_1, \ldots, X_n$ of independent observations
- For any interval $[i, j] \subset \{2, \ldots, n - 1\}$, define for all $f, g \in \mathcal{H}$
  \[
  \langle \hat{\mu}_{i:j}, f \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{j - i + 1} \sum_{\ell=i}^{j} f(X_\ell)
  \]
  \[
  \langle f, \hat{\Sigma}_{i:j} g \rangle_{\mathcal{H}} \overset{\text{def}}{=} \frac{1}{j - i + 1} \sum_{\ell=i}^{j} \{f(X_\ell) - \langle \hat{\mu}_{i:j}, f \rangle_{\mathcal{H}}\} \{g(X_\ell) - \langle \hat{\mu}_{i:j}, g \rangle_{\mathcal{H}}\}
  \]
- For any instant $k \in \{2, \ldots, n - 1\}$,
  \[
  \text{KFDR}_{n,k;\gamma}(X_1, \ldots, X_n) \text{ (maximum) Kernel Fisher Discriminant Ratio}
  \]
  \[
  \overset{\text{def}}{=} \frac{k(n - k)}{n} \left\| \left( \frac{k}{n} \hat{\Sigma}_{1:k} + \frac{n - k}{n} \hat{\Sigma}_{k+1:n} + \gamma I \right)^{-1/2} \left( \hat{\mu}_{k+1:n} - \hat{\mu}_{1:k} \right) \right\|_{\mathcal{H}}^2.
  \]
Kernel Change-point Analysis (KCpA)

**KCpA Test statistic**

\[ T_{n;\gamma_n} = \max_{a_n < k < b_n} \frac{\text{KFDR}_{n,k;\gamma_n} - d_{1,n,k;\gamma_n}(\hat{\Sigma}_{n,k})}{\sqrt{2} \ d_{2,n,k;\gamma_n}(\hat{\Sigma}_{n,k})} \]

with

\[ d_{1,n,k;\gamma}(\hat{\Sigma}_{n,k}) \overset{\text{def}}{=} \text{Tr}((\hat{\Sigma}_{n,k} + \gamma I)^{-1}\hat{\Sigma}_{n,k}) \quad \text{recentering} \]

\[ d_{2,n,k;\gamma}(\hat{\Sigma}_{n,k}) \overset{\text{def}}{=} \left[ \text{Tr}((\hat{\Sigma}_{n,k} + \gamma I)^{-2}(\hat{\Sigma}_{n,k})^2) \right]^{1/2} \quad \text{rescaling} \]

**Change-point Detection**

\[ T_{n;\gamma_n} \leq t_{1-\alpha} \quad \text{no change occurred} \]
\[ T_{n;\gamma_n} > t_{1-\alpha} \quad \text{a change occurred} \]

with \( t_{1-\alpha} \) the \( \alpha \)-significance threshold.

**Change-point Estimation**

\[ \hat{k}_n = \arg\max \frac{\text{KFDR}_{n,k;\gamma_n} - d_{1,n,k;\gamma_n}(\hat{\Sigma}_{n,k})}{\sqrt{2} \ d_{2,n,k;\gamma_n}(\hat{\Sigma}_{n,k})} \]

if a change has indeed occurred \((H_A)\), and where \( \hat{k}_n \) is the change-point estimator.
Change-point Analysis

Limiting distribution under $H_0 : \gamma_n \to 0$ regime

**Proposition**

Assume that the kernel is bounded and that for $a = 1, 2$ the eigenvalues $\{\lambda_p(\Sigma_a)\}_{p \geq 1}$ of the covariance operator $\Sigma$ satisfy $\sum_{p=1}^{\infty} \lambda_p^{1/2}(\Sigma_a) < \infty$. Assume in addition $H_0$, i.e. $P_{X_i} = P$ for all $1 \leq i \leq n$, and that $\{\gamma_n\}_{n \geq 1}$ is such that

$$\gamma_n + \frac{d_{1,n;\gamma_n}(\Sigma)}{d_{2,n;\gamma_n}(\Sigma)} \gamma_n^{-1} n^{-1/2} \to 0,$$

Then,

$$\max_{a_n < k < b_n} T_{n;\gamma_n}(k) \xrightarrow{\mathcal{D}} \sup_{u < t < v} \frac{B(t)}{\sqrt{t(1 - t)}},$$

where $a_n/n \to u > 0$ and $b_n/n \to v < 1$ as $n \to \infty$, and $\{B_p(t)\}_t$ is a brownian bridge.

**Remark**

- Typically : $\gamma_n \to 0$ slower than $1/\sqrt{n}$
- Case $\lambda_p = p^{-2m}$ : $d_{1,n_1,n_2;\gamma_n} \sim \gamma_n^{-1/2m}$ et $d_{2,n_1,n_2;\gamma_n} \sim \gamma_n^{-1/4m}$
Consistency in power

Proposition
Assume that the kernel is bounded and that for $a = 1, 2$ the eigenvalues $\{\lambda_p(\Sigma_a)\}_{p \geq 1}$ satisfy $\sum_{p=1}^{\infty} \lambda_p^{1/2}(\Sigma_a) < \infty$, and that the RKHS is dense in $L^2(\mathbb{P})$ for all $\mathbb{P}$, and $H_A$, i.e. $u < \theta^* < v$ with $u > 0$ and $v < 1$ such that $\mathbb{P}X_{[n\theta^*]} \neq \mathbb{P}X_{[n\theta^*]+1}$ for all $1 \leq i \leq n$. Then, in either regularization scheme, for all $0 < \alpha < 1$,

$$\mathbb{P}_{H_A} \left( \max_{a_n < k < b_n} \frac{\text{KFDR}_{n,k;\gamma} - d_{1,n,k;\gamma}(\hat{\Sigma}_{n,k}^W)}{\sqrt{2} d_{2,n,k;\gamma}(\hat{\Sigma}_{n,k}^W)} > t_{1-\alpha} \right) \to 1, \quad \text{as } n \to \infty,$$

(2)

where $a_n/n \to u > 0$ and $b_n/n \to v < 1$ as $n \to \infty$.

Remark
Universal density of RKHS satisfied for most translation-invariant kernels $k(x, y) = k(x - y)$, such as the gaussian kernel (Steinwart, 2006; Sriperumbudur et al., 2008).
Mental task segmentation: comparison with supervised methods

Dataset

- Data: 3 normal subjects during 4 non-feedback sessions
- 3 tasks: imagination of repetitive self-paced left hand movements or right hand movements, and generation of words beginning with the same random letter
- Features: based on Power Spectral Density

Experimental results

<table>
<thead>
<tr>
<th></th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCpA</td>
<td>79%</td>
<td>74%</td>
<td>61%</td>
</tr>
<tr>
<td>SVM</td>
<td>76%</td>
<td>69%</td>
<td>60%</td>
</tr>
</tbody>
</table>
Mental task segmentation: comparison with unsupervised methods

Dataset

- Data: 3 normal subjects during 4 non-feedback sessions
- 3 tasks: imagination of repetitive self-paced left hand movements or right hand movements, and generation of words beginning with the same random letter
- Features: based on Power Spectral Density

Experimental results

Figure: Comparison of performances in terms of ROC curves
Conclusion

Kernel learning and regularization

- Extension of mean element/covariance operator analysis to varying-kernel/multiple kernel settings
- Importance of regularization in unsupervised learning (see discriminative clustering and detection problems)

Computational efficiency

- Efficient large-scale versions of kernel-based unsupervised learning algorithms
- Low-rank approximation suited for particular unsupervised learning tasks