Spectral methods

Some methods:

- Kernel PCA
- MDS
- Spectral Clustering
- Isomap and Locally Linear Embedding (LLE)

Common to all approaches:

- Embedding methods.
- No immediate out-of-sample extension.

Observation: All methods can be regarded as spectral decompositions of a kernel matrix.

Proposed approach: Generic out-of-sample extension by application of Nyström formula to kernel operator.

Principal Component Analysis – PCA

Find components that are most useful for representing data based on the covariance (scatter) matrix C.

Eigenvectors sorted by decreasing Eigenvalues: provide direction of highest variance.

Select top N Eigenvectors to project input data into reduced space that best minimizes the squared-error.

Linear system.

Kernel PCA

Map data onto a (higher dimension, possibly infinite) feature space via a data independent function $\tilde{\phi}$

$$\tilde{\phi} : \mathbf{x} \rightarrow \tilde{\phi}(\mathbf{x})$$

Perform PCA on higher feature space to yield the new (lower dimensional) space.

Compute (empirical) feature space covariance matrix:

$$\tilde{C} = \hat{\mathsf{E}}\left[\tilde{\phi}(\mathbf{x})\tilde{\phi}(\mathbf{x})^T\right]$$

Kernel PCA

Problem 1: Centering

$$\phi_m(\mathbf{x}) = \tilde{\phi}(\mathbf{x}) - \frac{1}{m}\Sigma_{i=1}^{m}\tilde{\phi}(\mathbf{x}_i)$$

New data dependent function $\phi_m(x)$

$$C = \hat{\mathsf{E}}\left[\phi_m(\mathbf{x})\phi_m(\mathbf{x})^T\right]$$

Eigen-decomposition to eigenvectors w_r and associated eigen-values λ_r

Kernel PCA

Problem 2: High dimension: computationally expensive

Solution: Perform all vector operations by kernel trick.

$$k(\mathbf{s}, \mathbf{t}) = \langle \phi(\mathbf{s}) | \phi(\mathbf{t}) \rangle$$

Centering in feature space

Data independent Kernel $\tilde{k}(\mathbf{x}, \mathbf{y}) = \left\langle \tilde{\phi}(\mathbf{x}) | \tilde{\phi}(\mathbf{y}) \right\rangle$

Extend $\tilde{k}(\mathbf{x}, y)$ to data dependent Kernel corresponding to ϕ_m.

$$
\begin{aligned}
k_m(\mathbf{x}, \mathbf{y}) &= \langle \phi_m(\mathbf{x}) | \phi_m(\mathbf{y}) \rangle \\
&= \tilde{k}(\mathbf{x}, \mathbf{y}) - \hat{\mathsf{E}}_{\mathbf{x}'}[\tilde{k}(\mathbf{x}', \mathbf{y})] - \hat{\mathsf{E}}_{\mathbf{y}'}[\tilde{k}(\mathbf{x}, \mathbf{y}')] \\
&\quad + \hat{\mathsf{E}}_{\mathbf{x}', \mathbf{y}'}[\tilde{k}(\mathbf{x}', \mathbf{y}')]
\end{aligned}
$$

Kernel PCA

Gram matrix: K with $K_{ij} := \langle \mathbf{x}_i | \mathbf{x}_j \rangle = k_m(\mathbf{x}_i, \mathbf{x}_j)$

Covariance matrix: $C = \hat{\mathrm{E}} \left[\phi_m(\mathbf{x}) \phi_m(\mathbf{x})^T \right]$

Data matrix in feature space: $\Phi = (\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_m))^t$

Note:

$$K = \Phi \Phi^t \qquad C = \frac{1}{m} \Phi^t \Phi$$

Eigenvector relation

Eigensystems: $K\mathbf{v}_r = l_r\mathbf{v}_r \qquad C\mathbf{w}_r = \lambda_r\mathbf{w}_r$

Relation: If $\mathbf{v}$ is an eigenvector of K,

$$C(\Phi^t\mathbf{v}) = \frac{1}{m}\Phi^t\Phi\Phi^t\mathbf{v} = \frac{1}{m}\Phi^t K\mathbf{v} = \frac{1}{m}\Phi^t l\mathbf{v} = \frac{l}{m}(\Phi^t\mathbf{v})$$

$\Rightarrow \mathbf{w} := \Phi^t\mathbf{v}$ eigenvector of C.

Consequence: Eigenvectors of C (size $H \times H$) can be computed indirectly via eigenvectors of K (size $m \times m$).

Kernel PCA

Compute projection $P(\mathbf{x}) = (P_1(\mathbf{x}), \ldots, P_N(\mathbf{x}))^t$ for point $\mathbf{x}$:

$$
\begin{aligned}
P_r(\mathbf{x}) &= \langle w_r | \phi_m(\mathbf{x}) \rangle = \left\langle \frac{1}{\sqrt{l_r}} \Phi^t \mathbf{v}_r \middle| \phi_m(\mathbf{x}) \right\rangle \\
&= \left\langle \frac{1}{\sqrt{l_r}} \sum_{i=1}^{m} \phi_m(\mathbf{x}_i) v_{ri} \middle| \phi_m(\mathbf{x}) \right\rangle \\
&= \frac{1}{\sqrt{l_r}} \sum_{i=1}^{m} v_{ri} \langle \phi_m(\mathbf{x}_i) | \phi_m(\mathbf{x}) \rangle \\
&= \frac{1}{\sqrt{l_r}} \sum_{i=1}^{m} v_{ri} k_m(\mathbf{x}_i, \mathbf{x})
\end{aligned}
$$

Spectral clustering

Problem: Clustering (N clusters) for non-blob data.

Idea: Use a "proximity kernel": $k(\mathbf{x}, \mathbf{y})$ large $\Leftrightarrow$ $\mathbf{x}, \mathbf{y}$ close

Resulting K: Proximity table.

Spectral decomposition: Decorrelated components $\leftrightarrow$ non-proximate points

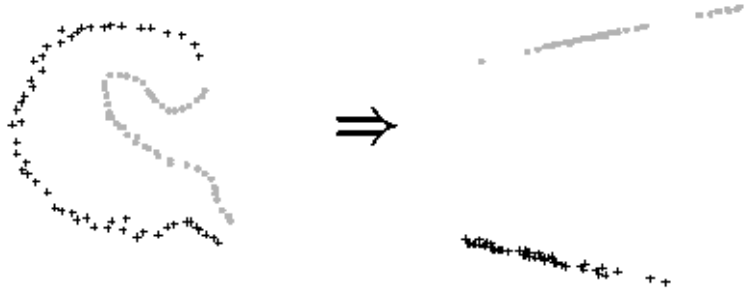
Clustering: First N eigenvectors should correspond to the N clusters.

Spectral clustering

Algorithm:

1. Compute K.
2. Compute first N eigenvectors.
3. Normalize.
4. Perform standard clustering algorithm.

Example: Input data, data after step (2):



Integral kernels

Integral equations: $Tf = g$, where f, g are functions.

Integral operator T defined by means of *integral kernel* k:

$$(Tf)(\mathbf{y}) := \int_\Omega k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) d\mu(\mathbf{x}) = \int_\Omega k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

Properties:

- T is linear (since integral linear)
- k is assumed to be symmetric, i.e. $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$.

Comparison: Linear algebra

Finite-dim. lin. operator: Represented as matrix $Mv = u$, $v, u \in \mathbb{R}^d$.
Component i of u: $u_i = \sum_j M_{ij} v_j$ (*)

Functions instead of vectors: T "infinitely large square matrix"

Infinite-dim. case:

indices i, j	$\rightarrow$	variables $\mathbf{x}, \mathbf{y}$
M_{ij}	$\rightarrow$	$k(\mathbf{x}, \mathbf{y})$
$\sum_i$	$\rightarrow$	$\int dx$

Analogue to sum (*): $g(\mathbf{y}) = \int_\Omega k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) d\mathbf{x}$

Eigensystem of T

Eigenfunctions: $T\psi = \lambda\psi$ with $\lambda \in \mathbb{R}$, $\psi : \mathbb{R}^d \to \mathbb{R}$.

Scalar product:

$$\langle f|g\rangle_p := \int_\Omega f(\mathbf{x})g(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

Orthogonality: f, g are p-orthogonal iff $\langle f|g\rangle_p = 0$.

Analogue to symm. matrices:

- all eigenvalues $\lambda \in \mathbb{R}$
- eigenfunctions are p-orthonormal: $\forall i, j :$ $\langle \psi_i|\psi_j\rangle_p = \delta_{ij}$

Nyström's method

Approximating T: Given data $\mathbf{x}_1, ..., \mathbf{x}_m$, substitute p_{emp} for p:

$$(\hat{T}f)(\mathbf{y}) := \int_\Omega k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) p_{\text{emp}}(\mathbf{x}) d\mathbf{x} = \frac{1}{m} \sum_{i=1}^{m} k(\mathbf{x}_i, \mathbf{y}) f(\mathbf{x}_i)$$

Approximating eigenfunctions: Assuming that $\hat{T} \approx T$, for eigenfunction ψ:

$$\lambda\psi(\mathbf{y}) = (T\psi)(\mathbf{y}) \approx (\hat{T}\psi)(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} k(\mathbf{x}_i, \mathbf{y}) \psi(\mathbf{x}_i)$$

Interpolation formula for $\psi(\mathbf{y})$!

Nyström's method

Approximate p-orthogonality:

$$\delta_{ij} = \langle \psi_i | \psi_j \rangle_p \approx \int_\Omega \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) p_{\text{emp}}(\mathbf{x}) d\mathbf{x} = \frac{1}{m} \sum_k \psi_i(\mathbf{x}_k) \psi_j(\mathbf{x}_k)$$

Spectral decomposition: Kernel represented by eigensystem:

$$k(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{\infty} \lambda_l \psi_l(\mathbf{x}) \psi_l(\mathbf{y}) \approx \sum_{l=1}^{N} \lambda_l \psi_l(\mathbf{x}) \psi_l(\mathbf{y})$$

(Linear algebra analogue: $K = V \Lambda V^t$.)

Summary: Discretized spectrum

Define: $\hat{\psi}_l := (\psi_l(\mathbf{x}_1), ...\psi_l(\mathbf{x}_j))^t$

We know:

$$
\begin{aligned}
K\hat{\psi}_l &\approx \lambda\hat{\psi}_l \\
\left\langle \hat{\psi}_i | \hat{\psi}_j \right\rangle &\approx \delta_{ij} \\
\psi_l(\mathbf{y}) &\approx \frac{1}{m\lambda_l} \sum_{i=1}^{m} k(\mathbf{x}_i, \mathbf{y}) \hat{\psi}_{li} \\
k(\mathbf{x}, \mathbf{y}) &\approx \sum_{i=1}^{N} \lambda_l \psi_l(\mathbf{x}) \psi_l(\mathbf{y})
\end{aligned}
$$

Out-of-sample extension for spectral methods

Given: Embedding of $\mathbf{x}_1, \ldots, \mathbf{x}_m$, new point $\mathbf{x}_{m+1}$.

Idea. If $\mathbf{x}_{m+1}$ had been included in training: All eigenvectors would contain additional component $\hat{\psi}_{l,m+1}$.

Approximation property: $\hat{\psi}_{l,m+1} \approx \psi_l(\mathbf{x}_{m+1})$.

With interpolation formula:

$$\hat{\psi}_{l,m+1} \approx \psi_l(\mathbf{x}_{m+1}) \approx \frac{1}{m\hat{\lambda}_l} \sum_{k=1}^{m} k(\mathbf{x}_j, \mathbf{x}_{m+1}) \hat{\psi}_{lj}$$

Proposed learning criterion

Consider matrix analogue first, for matrix $A \in \mathbb{R}^{m \times m}$.

Property utilized: Spectral decomposition $A = \sum_{l=1}^{m} \lambda_l \mathbf{v}_l \mathbf{v}_l^t$.

Use for successive approximation:

$$\underset{\mathbf{v}}{\operatorname{argmin}} \|A - \mathbf{v}\mathbf{v}^t\|_2$$

will recover $\mathbf{v} = \lambda_1 \mathbf{v}_1 \Rightarrow$ eigenpair: $\lambda_1 := \|\mathbf{v}\|, \mathbf{v}_1 := \frac{1}{\lambda_1}\mathbf{v}$.

Iterate: If first $(N-1)$ eigenpairs known,

$$\underset{\mathbf{v}}{\operatorname{argmin}} \|A - \mathbf{v}\mathbf{v}^t - \sum_{l=1}^{N-1} \lambda_l \mathbf{v}_l \mathbf{v}_l^t\|_2$$

Proposed learning criterion

For kernels: If we could actually optimize w.r.t. a function,

$$\operatorname*{argmin}_{\psi} \left\| k(\mathbf{x},\mathbf{y}) - \psi(\mathbf{x})\psi(\mathbf{y}) - \sum_{l=1}^{N-1} \lambda_l \psi_l(\mathbf{x})\psi_l(\mathbf{y}) \right\|_2$$

Approximation on sample:

$$\operatorname*{argmin}_{\mathbf{v}} \frac{1}{m^2} \sum_{i,j} \left(K_{ij} - v_i v_j - \sum_{l=1}^{N-1} \hat{\lambda}_l \hat{\psi}_{li} \hat{\psi}_{lj} \right)^2$$

Theoretical results

Prop. 1: For $\mathbf{y} \in \{\mathbf{x}_1, ..., \mathbf{x}_m\}$, the approximation

$$\psi_l(\mathbf{y}) \approx \frac{1}{m\lambda_l} \sum_{i=1}^{m} k(\mathbf{x}_i, \mathbf{y}) \hat{\psi}_{il}$$

is exact.

Prop. 2: Convergence of eigenfunctions. If

1. k not data-dependent
2. k bounded
3. (geometric) multiplicity of λ_l is 1 (and $\lambda_l \neq 0$)

then: approximate eigensystem converges to real one.

Theoretical results

Data-dependent case: Additionally require $k_m \longrightarrow k$ uniformly.

Prop. 3: Learning criterion.

1. Optimization of learning criterion equivalent to computation of corresponding eigendecomposition.
2. Approximate criterion asymptotically converges to exact one.

Novelty of results:

Result	Novelty
Kernel rep. of spectral methods	Few are new.
Common framework	Novel.
Nyström interpolation & prediction	Williams & Seeger, 2001
Prop. 1 & 2: Eigensystem appr.	e.g. Anselone (*)
Prop. 3: Learning criterion	Basic result in LA/FA.

Previous publication: Neural Comp. 16, 2197-2219, 2004.

*) P. M. Anselone: “Collectively compact operator approximation theory and applications to integral equations” (1971)