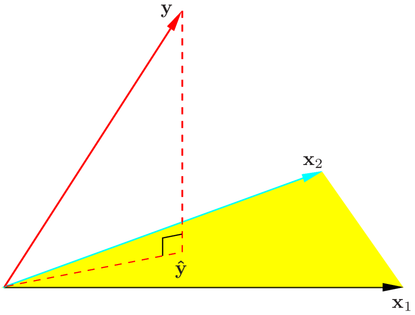
Dimensionality Reduction

Zoltán Szabó – CMAP, École Polytechnique

Data Science @ HEC Paris
May 10, 2019

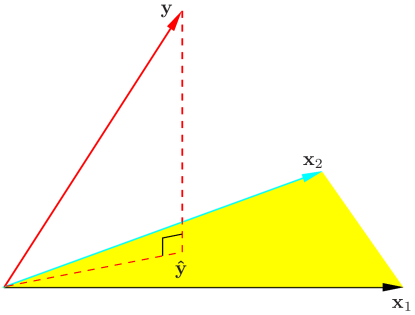
Recall from Tuesday

- We projected to a fixed subspace, $span(\{\mathbf{x}_i\}_{i=1}^n)$:



Recall from Tuesday

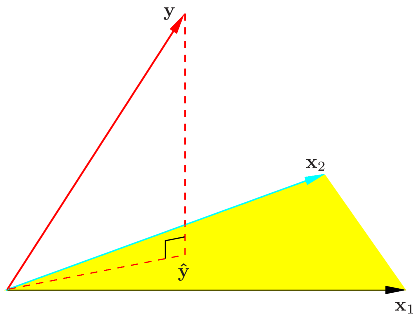
- We projected to a fixed subspace, $span(\{\mathbf{x}_i\}_{i=1}^n)$:



- Non-linear extensions:
  - $\varphi(x)$: explicit,

Recall from Tuesday

- We projected to a fixed subspace, $span(\{\mathbf{x}_i\}_{i=1}^n)$:

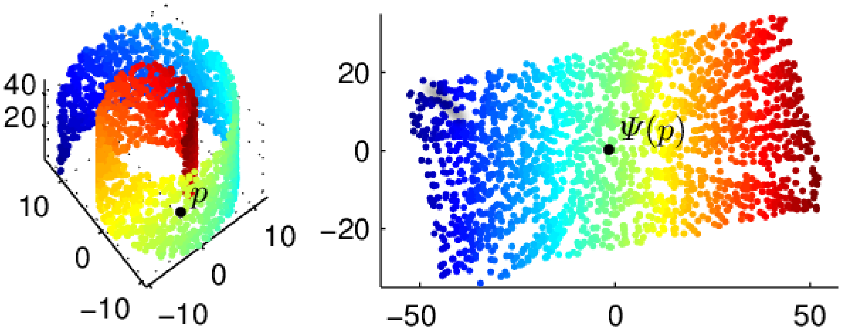


- Non-linear extensions:
  - $\varphi(x)$: explicit,
  - $k(x,x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}_k}$
    - implicit usage of features,
    - $\mathcal{H}_k = \overline{\left\{\sum_{i=1}^n \alpha_i \varphi(x_i)\right\}}$.

Today: dimensionality reduction

- Given: a set of observations $X = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^D$.
- Goal: find $X' = \{\mathbf{x}'_i\}_{i=1}^n \subset \mathbb{R}^d$ 'preserving' the geometry of X.
- $d \ll D$: compression (images, music, ...).

Dimensionality reduction = manifold learning



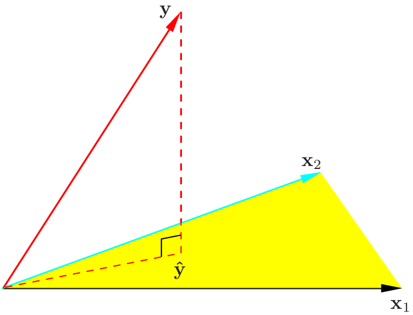
Why?

- Visualization, computational reason, noise reduction.

Why?

- Visualization, computational reason, noise reduction.
- Simplest example:

  We optimize the subspace of projection (PCA).



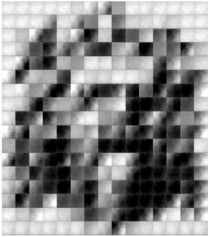
Principal Component Analysis (PCA)

PCA example: 100%

(A)

PCA example: 100% $\rightarrow$ 1%

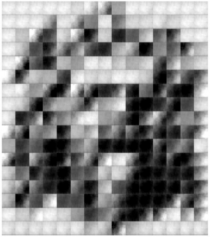
(A)



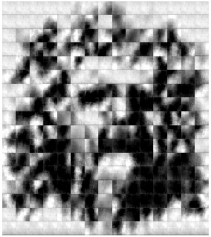
(B)

PCA example: 100% → 2%

(A)



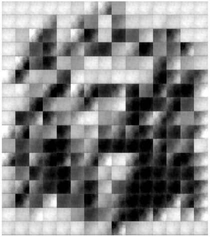
(B)



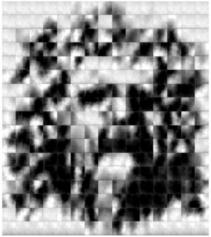
(C)

PCA example: 100% $\rightarrow$ 5%

(A)



(B)

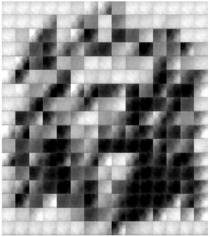


(C)

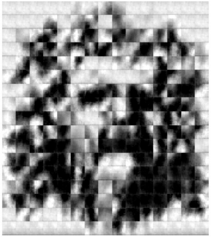
(D)

PCA example: 100% $\rightarrow$ 10%

(A)



(B)

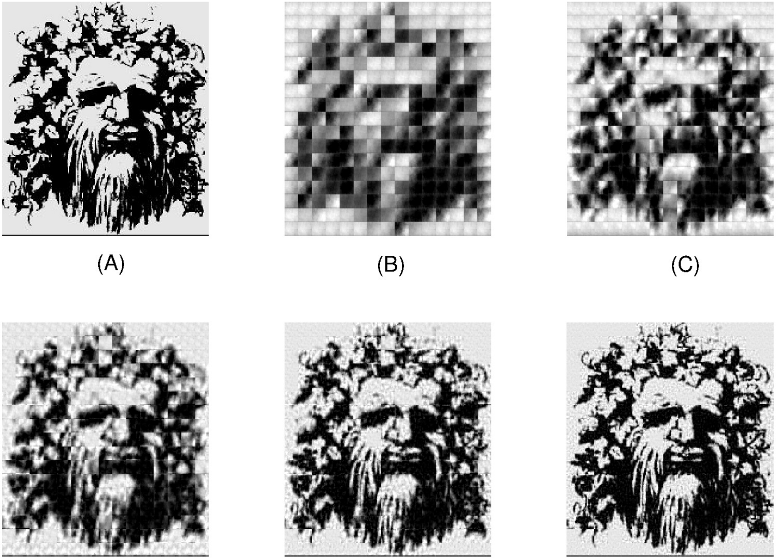


(C)

(D)

(E)

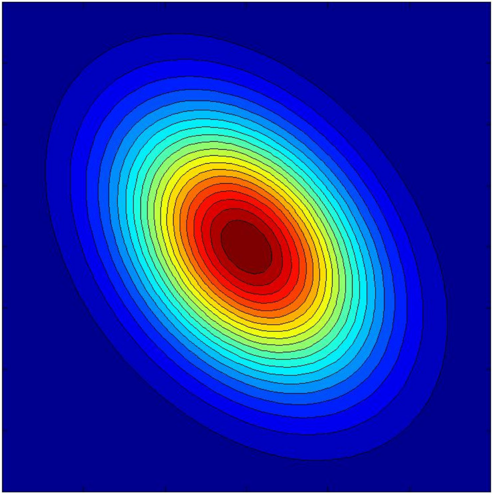
PCA example: 100% → 20%



(A) (B) (C)

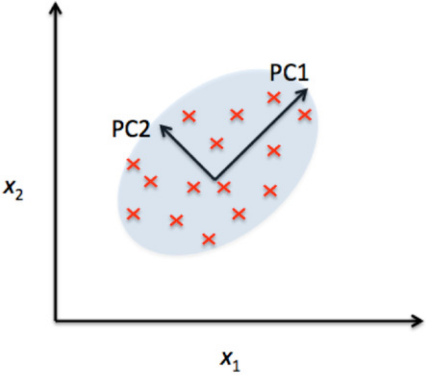
(D) (E) (F)

Conjecture? Most important direction?



PCA: intuition

Task: find the best d-dimensional subspace approximating $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^D$.



Cov, var, corr: properties – recall

- Covariance:

$$\mathrm{cov}(x, y) = \mathbb{E}_{xy}[(x - \mathbb{E}x)(y - \mathbb{E}y)].$$

Cov, var, corr: properties – recall

- Covariance: → values? $\mathrm{cov}(ax, by) = ?$

$$\mathrm{cov}(x, y) = \mathbb{E}_{xy}[(x - \mathbb{E}x)(y - \mathbb{E}y)].$$

Cov, var, corr: properties – recall

- Covariance:

$$\mathrm{cov}(x, y) = \mathbb{E}_{xy}[(x - \mathbb{E}x)(y - \mathbb{E}y)].$$

- Variance:

$$\mathrm{var}(x) = \mathrm{cov}(x, x) = \mathbb{E}\left(x^2\right) - \mathbb{E}^2(x)$$

Cov, var, corr: properties – recall

- Covariance:

$$\mathrm{cov}(x,y) = \mathbb{E}_{xy}[(x-\mathbb{E}x)(y-\mathbb{E}y)].$$

- Variance, std: $\rightarrow$ values? min?

$$\mathrm{var}(x) = \mathrm{cov}(x,x) = \mathbb{E}\left(x^2\right) - \mathbb{E}^2(x), \quad \sigma(x) = \sqrt{\mathrm{var}(x)}.$$

Cov, var, corr: properties – recall

- Covariance:

$$\mathrm{cov}(x, y) = \mathbb{E}_{xy}[(x - \mathbb{E}x)(y - \mathbb{E}y)].$$

- Variance, std:

$$\mathrm{var}(x) = \mathrm{cov}(x, x) = \mathbb{E}\left(x^2\right) - \mathbb{E}^2(x), \quad \sigma(x) = \sqrt{\mathrm{var}(x)}.$$

- Correlation:

$$\mathrm{corr}(x, y) = \frac{\mathrm{cov}(x, y)}{\sigma(x)\sigma(y)}.$$

Cov, var, corr: properties – recall

- Covariance:

$$\mathrm{cov}(x,y) = \mathbb{E}_{xy}[(x-\mathbb{E}x)(y-\mathbb{E}y)].$$

- Variance, std:

$$\mathrm{var}(x) = \mathrm{cov}(x,x) = \mathbb{E}\left(x^2\right) - \mathbb{E}^2(x), \quad \sigma(x) = \sqrt{\mathrm{var}(x)}.$$

- Correlation: $\to$ intuition? values? max? zero?

$$\mathrm{corr}(x,y) = \frac{\mathrm{cov}(x,y)}{\sigma(x)\sigma(y)}.$$

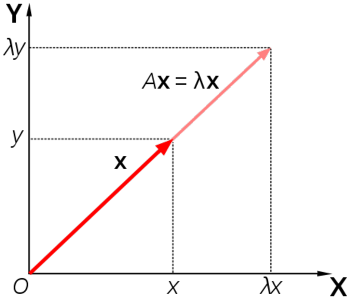
Eigenvectors, eigenvalues – recall

- Simplest transformation: scaling.

Eigenvectors, eigenvalues – recall

- Simplest transformation: scaling.
- $\mathbf{x} \neq \mathbf{0}$ is an eigenvector of $\mathbf{A}$ with eigenvalue $\lambda \in \mathbb{R}$ if

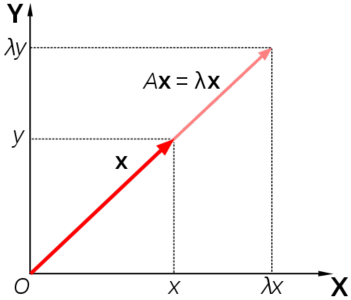
$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}.$$



Eigenvectors, eigenvalues – recall

- Simplest transformation: scaling.
- $\mathbf{x} \neq \mathbf{0}$ is an eigenvector of $\mathbf{A}$ with eigenvalue $\lambda \in \mathbb{R}$ if

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}.$$



- Size of $\mathbf{A}$?

Eigensystems: continued

Examples:

- Identity: $\mathbf{A} = \mathbf{I}$.

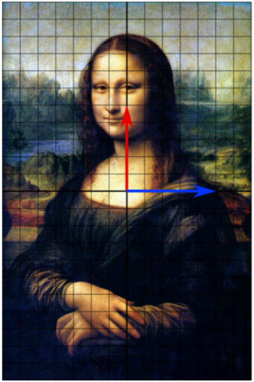
Eigensystems: continued

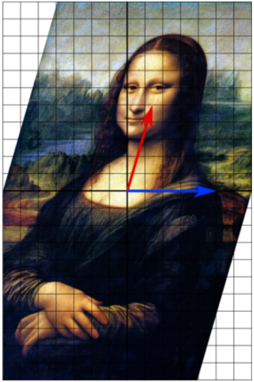
Examples:

- Identity: $\mathbf{A} = \mathbf{I}$.
- Diagonal matrix: $\mathbf{A} = diag(a_i)$, spec: reflection.

Examples:

- Identity: $\mathbf{A} = \mathbf{I}$.
- Diagonal matrix: $\mathbf{A} = diag(a_i)$, spec: reflection.
- Shear mapping on Mona Lisa:





Symmetric matrices are nice

- Diagonal matrix: we saw that the eigensystem is orthogonal.

Symmetric matrices are nice

- Diagonal matrix: we saw that the eigensystem is orthogonal.
- A symmetric $\mathbf{A}$ ($\mathbf{A} = \mathbf{A}^T$) behaves similarly:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T,$$

where $\mathbf{\Sigma} = diag(\lambda_i)$, $\mathbf{U}$: orthogonal.

Let us apply these observations in PCA!

PCA formulation: $d = 1$

- We are looking for the best one-dimensional projection.

- $\mathbb{E}$:= empirical/population expectation: $\mathbb{E}\mathbf{x} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$.
- Assumption: $\mathbb{E}\mathbf{x} = \mathbf{0}$.

PCA formulation: $d = 1$

- We are looking for the best one-dimensional projection.

- $\mathbb{E}$:= empirical/population expectation: $\mathbb{E}\mathbf{x} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$.
- Assumption: $\mathbb{E}\mathbf{x} = \mathbf{0}$.
  - centering: $\mathbf{x} \to \mathbf{x} - \mathbb{E}\mathbf{x}$.

PCA: projection

Projection ($\|\mathbf{w}\|_2 = 1$):

- $\hat{\mathbf{x}} = \langle \mathbf{w}, \mathbf{x} \rangle \mathbf{w}$.
- zero mean: $\mathbf{0} \stackrel{?}{=} \mathbb{E}\hat{\mathbf{x}} = \mathbb{E}\left[\langle \mathbf{w}, \mathbf{x} \rangle \mathbf{w}\right]$

PCA: projection

Projection ($\|\mathbf{w}\|_2 = 1$):

- $\hat{\mathbf{x}} = \langle \mathbf{w}, \mathbf{x} \rangle \mathbf{w}$.
- zero mean: $\mathbf{0} \stackrel{?}{=} \mathbb{E}\hat{\mathbf{x}} = \mathbb{E}\left[\langle \mathbf{w}, \mathbf{x} \rangle \mathbf{w}\right] = \Big\langle \mathbf{w}, \underbrace{\mathbb{E}\mathbf{x}}_{=\mathbf{0}} \Big\rangle \mathbf{w}$.

PCA: min residual $\Leftrightarrow$ max squared projection

- Goal: $\mathbb{E}\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 \to \min_{\mathbf{w}}$.

PCA: min residual $\Leftrightarrow$ max squared projection

- Goal: $\mathbb{E}\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 \rightarrow \min_{\mathbf{w}}$.
- Residual $\Rightarrow$ objective:

$$\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 = \left\|\mathbf{x}-\langle\mathbf{w},\mathbf{x}\rangle\,\mathbf{w}\right\|_2^2$$

PCA: min residual $\Leftrightarrow$ max squared projection

- Goal: $\mathbb{E}\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 \to \min_{\mathbf{w}}$.
- Residual $\Rightarrow$ objective:

$$\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 = \left\|\mathbf{x}-\left\langle \mathbf{w},\mathbf{x}\right\rangle \mathbf{w}\right\|_2^2 \overset{\|\mathbf{w}\|_2^2=1}{=} \left\|\mathbf{x}\right\|_2^2 - \left\langle \mathbf{w},\mathbf{x}\right\rangle^2 \Rightarrow$$

PCA: min residual $\Leftrightarrow$ max squared projection

- Goal: $\mathbb{E}\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 \to \min_{\mathbf{w}}$.
- Residual $\Rightarrow$ objective:

$$\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 = \left\|\mathbf{x}-\langle\mathbf{w},\mathbf{x}\rangle\mathbf{w}\right\|_2^2 \overset{\|\mathbf{w}\|_2^2=1}{=} \left\|\mathbf{x}\right\|_2^2 - \langle\mathbf{w},\mathbf{x}\rangle^2 \Rightarrow$$

$$\mathbb{E}\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 = \mathbb{E}\left[\left\|\mathbf{x}\right\|_2^2 - \langle\mathbf{w},\mathbf{x}\rangle^2\right] = \underbrace{\mathbb{E}\left\|\mathbf{x}\right\|_2^2}_{\text{independent of } \mathbf{w}} - \mathbb{E}\langle\mathbf{w},\mathbf{x}\rangle^2 \Leftrightarrow$$

PCA: min residual ⇔ max squared projection

- Goal: $\mathbb{E}\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 \to \min_{\mathbf{w}}$.
- Residual ⇒ objective:

$$\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 = \left\|\mathbf{x}-\langle\mathbf{w},\mathbf{x}\rangle\mathbf{w}\right\|_2^2 \overset{\|\mathbf{w}\|_2^2=1}{=} \left\|\mathbf{x}\right\|_2^2 - \langle\mathbf{w},\mathbf{x}\rangle^2 \Rightarrow$$

$$\mathbb{E}\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2^2 = \mathbb{E}\left[\left\|\mathbf{x}\right\|_2^2 - \langle\mathbf{w},\mathbf{x}\rangle^2\right] = \underbrace{\mathbb{E}\left\|\mathbf{x}\right\|_2^2}_{\text{independent of } \mathbf{w}} - \mathbb{E}\langle\mathbf{w},\mathbf{x}\rangle^2 \Leftrightarrow$$

Solution

maximizes the mean squared projection.

PCA: max squared projection ⇔ max variance of projection

By using $\mathbb{E}y^2 = (\mathbb{E}y)^2 + \mathrm{var}(y)$:

$$\max_{\mathbf{w}} \leftarrow \mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle^2 = \Big(\underbrace{\mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle}_{=0} \Big)^2 + \mathrm{var}(\left\langle \mathbf{w}, \mathbf{x} \right\rangle).$$

PCA: max squared projection ⇔ max variance of projection

By using $\mathbb{E}y^2 = (\mathbb{E}y)^2 + \mathrm{var}(y)$:

$$\max_{\mathbf{w}} \leftarrow \mathbb{E}\langle \mathbf{w}, \mathbf{x} \rangle^2 = \Big(\underbrace{\mathbb{E}\langle \mathbf{w}, \mathbf{x} \rangle}_{=0} \Big)^2 + \mathrm{var}(\langle \mathbf{w}, \mathbf{x} \rangle).$$

To sum up:

$$\text{Minimize MSE of the residual} : \min_{\mathbf{w}} \mathbb{E}\left\| \mathbf{x} - \hat{\mathbf{x}} \right\|_2^2 \Leftrightarrow$$

By using $\mathbb{E}y^2 = (\mathbb{E}y)^2 + \mathrm{var}(y)$:

$$\max_{\mathbf{w}} \leftarrow \mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle^2 = \Big(\underbrace{\mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle}_{=0}\Big)^2 + \mathrm{var}(\langle \mathbf{w}, \mathbf{x} \rangle).$$

To sum up:

$$\text{Minimize MSE of the residual} : \min_{\mathbf{w}} \mathbb{E}\left\|\mathbf{x} - \hat{\mathbf{x}}\right\|_2^2 \Leftrightarrow$$

$$\text{Maximize mean squared projection} : \max_{\mathbf{w}} \mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle^2 \Leftrightarrow$$

PCA: max squared projection $\Leftrightarrow$ max variance of projection

By using $\mathbb{E}y^2 = (\mathbb{E}y)^2 + \mathrm{var}(y)$:

$$\max_{\mathbf{w}} \leftarrow \mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle^2 = \Big(\underbrace{\mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle}_{=0}\Big)^2 + \mathrm{var}(\left\langle \mathbf{w}, \mathbf{x} \right\rangle).$$

To sum up:

$$\text{Minimize MSE of the residual} : \min_{\mathbf{w}} \mathbb{E}\left\| \mathbf{x} - \hat{\mathbf{x}} \right\|_2^2 \Leftrightarrow$$
$$\text{Maximize mean squared projection} : \max_{\mathbf{w}} \mathbb{E}\left\langle \mathbf{w}, \mathbf{x} \right\rangle^2 \Leftrightarrow$$
$$\text{Maximize variance of the projection} : \max_{\mathbf{w}} \mathrm{var}(\left\langle \mathbf{w}, \mathbf{x} \right\rangle).$$

PCA: optimization

By the bilinearity of cov:

$$\operatorname{var}(\langle \mathbf{w}, \mathbf{x} \rangle) = \operatorname{cov}\left(\mathbf{w}^T\mathbf{x}, \mathbf{w}^T\mathbf{x}\right)$$

PCA: optimization

By the bilinearity of cov:

$$\operatorname{var}\left(\langle \mathbf{w}, \mathbf{x} \rangle\right) = \operatorname{cov}\left(\mathbf{w}^T \mathbf{x}, \mathbf{w}^T \mathbf{x}\right) = \mathbf{w}^T \operatorname{cov}(\mathbf{x}) \mathbf{w} =: \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \rightarrow \max_{\|\mathbf{w}\|_2 = 1}.$$

PCA: optimization

By the bilinearity of cov:

$$\operatorname{var}(\langle \mathbf{w}, \mathbf{x} \rangle) = \operatorname{cov}\left(\mathbf{w}^T\mathbf{x}, \mathbf{w}^T\mathbf{x}\right) = \mathbf{w}^T \operatorname{cov}(\mathbf{x})\mathbf{w} =: \mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} \to \max_{\|\mathbf{w}\|_2=1}.$$

Lagrange function, solving for 'derivatives = 0':

PCA: optimization

By the bilinearity of cov:

$$\mathrm{var}\left(\langle \mathbf{w}, \mathbf{x} \rangle\right) = \mathrm{cov}\left(\mathbf{w}^T\mathbf{x}, \mathbf{w}^T\mathbf{x}\right) = \mathbf{w}^T \mathrm{cov}(\mathbf{x})\mathbf{w} =: \mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} \to \max_{\|\mathbf{w}\|_2 = 1}.$$

Lagrange function, solving for 'derivatives = 0':

$$L(\mathbf{w}, \lambda) = \underbrace{\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}}_{=\text{objective}} - \lambda(\underbrace{\mathbf{w}^T\mathbf{w} - 1}_{=\text{condition}}) \Rightarrow$$

PCA: optimization

By the bilinearity of cov:

$$\mathrm{var}\left(\langle \mathbf{w}, \mathbf{x} \rangle\right) = \mathrm{cov}\left(\mathbf{w}^T\mathbf{x}, \mathbf{w}^T\mathbf{x}\right) = \mathbf{w}^T \mathrm{cov}(\mathbf{x})\mathbf{w} =: \mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} \to \max_{\|\mathbf{w}\|_2=1}.$$

Lagrange function, solving for 'derivatives = 0':

$$L(\mathbf{w}, \lambda) = \underbrace{\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}}_{=\text{objective}} - \lambda(\underbrace{\mathbf{w}^T\mathbf{w} - 1}_{=\text{condition}}) \Rightarrow$$

$$0 = \frac{\partial L(\mathbf{w}, \lambda)}{\partial \lambda} = -(\mathbf{w}^T\mathbf{w} - 1),$$

PCA: optimization

By the bilinearity of cov:

$$\mathrm{var}\left(\langle \mathbf{w}, \mathbf{x} \rangle\right) = \mathrm{cov}\left(\mathbf{w}^T\mathbf{x}, \mathbf{w}^T\mathbf{x}\right) = \mathbf{w}^T \mathrm{cov}(\mathbf{x})\mathbf{w} =: \mathbf{w}^T\mathbf{\Sigma}\mathbf{w} \to \max_{\|\mathbf{w}\|_2=1}.$$

Lagrange function, solving for 'derivatives = 0':

$$L(\mathbf{w},\lambda) = \underbrace{\mathbf{w}^T\mathbf{\Sigma}\mathbf{w}}_{=\text{objective}} - \lambda(\underbrace{\mathbf{w}^T\mathbf{w} - 1}_{=\text{condition}}) \Rightarrow$$

$$0 = \frac{\partial L(\mathbf{w},\lambda)}{\partial \lambda} = -(\mathbf{w}^T\mathbf{w} - 1),$$

$$\mathbf{0} = \frac{\partial L(\mathbf{w},\lambda)}{\partial \mathbf{w}} = 2\mathbf{\Sigma}\mathbf{w} - 2\lambda\mathbf{w} \Rightarrow$$

PCA: optimization

By the bilinearity of cov:

$$\mathrm{var}\left(\langle \mathbf{w}, \mathbf{x}\rangle\right) = \mathrm{cov}\left(\mathbf{w}^T\mathbf{x}, \mathbf{w}^T\mathbf{x}\right) = \mathbf{w}^T \mathrm{cov}(\mathbf{x})\mathbf{w} =: \mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} \to \max_{\|\mathbf{w}\|_2=1}.$$

Lagrange function, solving for 'derivatives = 0':

$$L(\mathbf{w},\lambda) = \underbrace{\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}}_{=\text{objective}} - \lambda(\underbrace{\mathbf{w}^T\mathbf{w} - 1}_{=\text{condition}}) \Rightarrow$$

$$0 = \frac{\partial L(\mathbf{w},\lambda)}{\partial \lambda} = -(\mathbf{w}^T\mathbf{w} - 1),$$

$$\mathbf{0} = \frac{\partial L(\mathbf{w},\lambda)}{\partial \mathbf{w}} = 2\boldsymbol{\Sigma}\mathbf{w} - 2\lambda\mathbf{w} \Rightarrow$$

Solution

$\mathbf{w}^*$: eigenvector associated to $\lambda_{\max}(\boldsymbol{\Sigma})$.

PCA ($d \geqslant 1$)

- Goal: approximate with a d-dimensional subspace.
- ONB in the subspace ($\mathbf{W}^T\mathbf{W} = \mathbf{I}$):

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_d] \in \mathbb{R}^{D \times d},$$

- Approximation:

$$\hat{\mathbf{x}} = \sum_{i=1}^{d} \langle \mathbf{w}_i, \mathbf{x} \rangle \mathbf{w}_i = \mathbf{W}\mathbf{W}^T\mathbf{x}.$$

After similar calculation than for $d = 1$ ...

PCA: $d \geqslant 1$

- The d principal components:

$$\{\mathbf{w}_i\}_{i=1}^{d} = \boxed{\text{top } d \text{ eigenvectors of } \mathrm{cov}(\mathbf{x})}.$$

PCA: $d \geqslant 1$

- The d principal components:

$$\{\mathbf{w}_i\}_{i=1}^{d} = \boxed{\text{top } d \text{ eigenvectors of } \operatorname{cov}(\mathbf{x})}.$$

- $\boldsymbol{\Sigma} := \operatorname{cov}(\mathbf{x})$: symmetric, positive semi-definite $\Rightarrow \{\mathbf{w}_i\}$: ONS, $\lambda_i \geqslant 0$.

PCA: $d \geqslant 1$

- The d principal components:

$$\{\mathbf{w}_i\}_{i=1}^d = \boxed{\text{top } d \text{ eigenvectors of } \operatorname{cov}(\mathbf{x})}.$$

- $\boldsymbol{\Sigma} := \operatorname{cov}(\mathbf{x})$: symmetric, positive semi-definite $\Rightarrow$ $\{\mathbf{w}_i\}$: ONS, $\lambda_i \geqslant 0$.
- Variance decomposition: $\operatorname{cov}(\mathbf{x}) = \sum_{i=1}^D \lambda_i \mathbf{w}_i \mathbf{w}_i^T$.

PCA: $d \geqslant 1$

- The d principal components:

$$\{\mathbf{w}_i\}_{i=1}^d = \boxed{\text{top } d \text{ eigenvectors of } \mathrm{cov}(\mathbf{x})}.$$

- $\mathbf{\Sigma} := \mathrm{cov}(\mathbf{x})$: symmetric, positive semi-definite $\Rightarrow$ $\{\mathbf{w}_i\}$: ONS, $\lambda_i \geqslant 0$.
- Variance decomposition: $\mathrm{cov}(\mathbf{x}) = \sum_{i=1}^D \lambda_i \mathbf{w}_i \mathbf{w}_i^T$.
- Energy preserved using d components: $\sum_{i=1}^d \lambda_i \Rightarrow$

$$R = R(d) := \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^D \lambda_i} \in [0, 1].$$

PCA: $d \geqslant 1$

- The d principal components:

$$\{\mathbf{w}_i\}_{i=1}^d = \boxed{\text{top } d \text{ eigenvectors of } \mathrm{cov}(\mathbf{x})}.$$

- $\boldsymbol{\Sigma} := \mathrm{cov}(\mathbf{x})$: symmetric, positive semi-definite $\Rightarrow$ $\{\mathbf{w}_i\}$: ONS, $\lambda_i \geqslant 0$.
- Variance decomposition: $\mathrm{cov}(\mathbf{x}) = \sum_{i=1}^D \lambda_i \mathbf{w}_i \mathbf{w}_i^T$.
- Energy preserved using d components: $\sum_{i=1}^d \lambda_i \Rightarrow$

$$R = R(d) := \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^D \lambda_i} \in [0, 1].$$

- In practice: choose d such that $R \approx 0.8 - 0.9$.

Non-linear PCA

Non-linear PCA: idea

- PCA:
  - objective: maximize the variance of the projection.
  - solution: leading eigenvectors of $\boldsymbol{\Sigma} = \mathrm{cov}(\mathbf{x})$.

Non-linear PCA: idea

- PCA:
  - objective: maximize the variance of the projection.
  - solution: leading eigenvectors of $\mathbf{\Sigma} = \mathrm{cov}(\mathbf{x})$.
- Non-linear PCA:
  - Take $\varphi(\mathbf{x})$.

Non-linear PCA: idea

- PCA:
  - objective: maximize the variance of the projection.
  - solution: leading eigenvectors of $\boldsymbol{\Sigma} = \mathrm{cov}(\mathbf{x})$.
- Non-linear PCA:
  - Take $\varphi(\mathbf{x})$.
  - What is $\boldsymbol{\Sigma} := \mathrm{cov}(\varphi(\mathbf{x}))$?

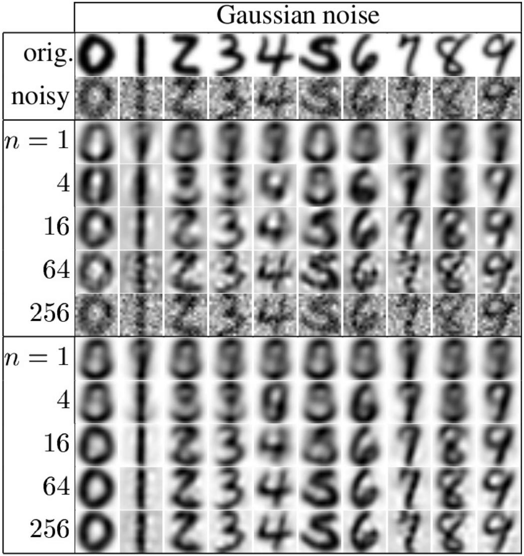
Non-linear PCA: idea

- PCA:
  - objective: maximize the variance of the projection.
  - solution: leading eigenvectors of $\boldsymbol{\Sigma} = \mathrm{cov}(\mathbf{x})$.
- Non-linear PCA:
  - Take $\varphi(\mathbf{x})$.
  - What is $\boldsymbol{\Sigma} := \mathrm{cov}(\varphi(\mathbf{x}))$?
  - Eigenvectors of an operator?

Non-linear PCA: idea

- PCA:
  - objective: maximize the variance of the projection.
  - solution: leading eigenvectors of $\mathbf{\Sigma} = \mathrm{cov}(\mathbf{x})$.
- Non-linear PCA:
  - Take $\varphi(\mathbf{x})$.
  - What is $\mathbf{\Sigma} := \boxed{\mathrm{cov}(\varphi(\mathbf{x}))}$?
  - Eigenvectors of an operator?
  - Computational tractability?

In denoising application: PCA vs non-linear PCA



Kernel PCA: idea for '$d = 1$' $\leftrightarrow f$

Let $\mathcal{H} = \mathcal{H}_k$.

- Objective function:

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}\left\langle f, \underbrace{\varphi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\varphi(x_j)}_{=:\tilde{\varphi}(x_i)}\right\rangle^2 = \mathrm{var}(f) \to \max_{f:\|f\|_{\mathcal{H}}\leqslant 1}.$$

Kernel PCA: idea for '$d = 1$' $\leftrightarrow f$

Let $\mathcal{H} = \mathcal{H}_k$.

- Objective function:

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}\left\langle f, \underbrace{\varphi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\varphi(x_j)}_{=:\tilde{\varphi}(x_i)}\right\rangle^2 = \mathrm{var}(f) \to \max_{f:\|f\|_{\mathcal{H}}\leqslant 1}.$$

- The solution can be searched in the form ($\mathcal{H} \ni f \leftrightarrow \mathbf{a} \in \mathbb{R}^n$):

$$f = \sum_{i=1}^{n} a_i \tilde{\varphi}(x_i)$$

since component $\perp$ *span* $\left(\{\tilde{\varphi}(x_i)\}_{i=1}^n\right)$ has no contribution.

Kernel PCA: idea for '$d = 1$' $\leftrightarrow f$

Let $\mathcal{H} = \mathcal{H}_k$.

- Objective function:

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}\left\langle f, \underbrace{\varphi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\varphi(x_j)}_{=:\tilde{\varphi}(x_i)}\right\rangle^2 = \mathrm{var}(f) \to \max_{f:\|f\|_{\mathcal{H}}\leqslant 1}.$$

- The solution can be searched in the form ($\mathcal{H} \ni f \leftrightarrow \mathbf{a} \in \mathbb{R}^n$):

$$f = \sum_{i=1}^{n} a_i \tilde{\varphi}(x_i)$$

since component $\perp$ *span* $\left(\{\tilde{\varphi}(x_i)\}_{i=1}^{n}\right)$ has no contribution.

- We will get an eigenvalue problem for **a**.

(Empirical) covariance operator

$$C := \frac{1}{n}\sum_{i=1}^{n} \tilde{\varphi}(x_i) \otimes \tilde{\varphi}(x_i).$$

$c \otimes d$ is the analogue of cd^T:

$$(c \otimes d)(e) = c \langle d, e \rangle_{\mathcal{H}}.$$

(Empirical) covariance operator

$$C := \frac{1}{n}\sum_{i=1}^{n} \tilde{\varphi}(x_i) \otimes \tilde{\varphi}(x_i).$$

$c \otimes d$ is the analogue of cd^T:

$$(c \otimes d)(e) = c \langle d, e \rangle_{\mathcal{H}}.$$

Similarly to the finite-dimensional case:

$$Cf_j = \lambda_j f_j.$$

Challenge

How do we solve this eigenvalue problem?

Computation of Cf_j

Assume j is fixed ($Cf = \lambda f$):

$$Cf = \left[\frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\otimes\tilde{\varphi}(x_i)\right]f$$

Computation of Cf_j

Assume j is fixed ($Cf = \lambda f$):

$$Cf = \left[\frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\otimes\tilde{\varphi}(x_i)\right]f$$

$$\overset{\otimes}{=}^{\text{def}} \frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\left\langle\tilde{\varphi}(x_i),\sum_{j=1}^{n}a_j\tilde{\varphi}(x_j)\right\rangle_{\mathcal{H}}$$

Assume j is fixed ($Cf = \lambda f$):

$$Cf = \left[\frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\otimes\tilde{\varphi}(x_i)\right]f$$

$$\overset{\otimes}{=}^{\text{def}} \frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\left\langle\tilde{\varphi}(x_i),\sum_{j=1}^{n}a_j\tilde{\varphi}(x_j)\right\rangle_{\mathcal{H}} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\sum_{j=1}^{n}a_j\tilde{k}(x_i,x_j)$$

with $\tilde{\mathbf{G}} = \mathbf{HGH} = \left[\tilde{k}(x_i,x_j)\right]_{i,j=1}^{n}$, $\mathbf{H} = \mathbf{I}_n - \frac{\mathbf{E}_n}{n}$, $\mathbf{E}_n = [1] \in \mathbb{R}^{n\times n}$.

Computation of Cf_j

Assume j is fixed ($Cf = \lambda f$):

$$Cf = \left[\frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\otimes\tilde{\varphi}(x_i)\right] f$$

$$\overset{\otimes}{=}^{\text{def}} \frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\left\langle\tilde{\varphi}(x_i), \sum_{j=1}^{n} a_j\tilde{\varphi}(x_j)\right\rangle_{\mathcal{H}} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\varphi}(x_i)\sum_{j=1}^{n} a_j\tilde{k}(x_i, x_j)$$

with $\tilde{\mathbf{G}} = \mathbf{HGH} = \left[\tilde{k}(x_i, x_j)\right]_{i,j=1}^{n}$, $\mathbf{H} = \mathbf{I}_n - \frac{\mathbf{E}_n}{n}$, $\mathbf{E}_n = [1] \in \mathbb{R}^{n\times n}$.

Since $f = \sum_{j=1}^{n} a_j \tilde{\varphi}(x_j)$

multiplying by $\tilde{\varphi}(x_r)$ $[r = 1, \ldots, n]$ gives expressions in terms of $\tilde{\mathbf{G}}$.

Eigenvalue problem

- We want to solve $Cf = \lambda f$; Cf and f: functions of $\tilde{\varphi}(x_i)$.
- By multipling with $\tilde{\varphi}(x_r)$:

$$\langle \tilde{\varphi}(x_r), \lambda f \rangle_{\mathcal{H}} = \lambda(\tilde{\mathbf{G}}\mathbf{a})_r,$$
$$\langle \tilde{\varphi}(x_r), Cf \rangle_{\mathcal{H}} = \frac{1}{n}(\tilde{\mathbf{G}}^2\mathbf{a})_r.$$

- Eigenvalue problem: $\tilde{\mathbf{G}}^2\mathbf{a} = n\lambda\tilde{\mathbf{G}}\mathbf{a}$, i.e. $\tilde{\mathbf{G}}\mathbf{a} = (n\lambda)\mathbf{a}$.

Orthogonal eigenvectors in kernel PCA

Taking two eigenvectors:

$$f_1 = \sum_{i=1}^{n} a_{1i}\tilde{\varphi}(x_i), \qquad \tilde{\mathbf{G}}\mathbf{a}_1 = \lambda_1 \mathbf{a}_1,$$

$$f_2 = \sum_{j=1}^{n} a_{2j}\tilde{\varphi}(x_j), \qquad \tilde{\mathbf{G}}\mathbf{a}_2 = \lambda_2 \mathbf{a}_2.$$

one has

$$0 \stackrel{?}{=} \langle f_1, f_2 \rangle_{\mathcal{H}}$$

Orthogonal eigenvectors in kernel PCA

Taking two eigenvectors:

$$f_1 = \sum_{i=1}^{n} a_{1i}\tilde{\varphi}(x_i), \qquad \tilde{\mathbf{G}}\mathbf{a}_1 = \lambda_1 \mathbf{a}_1,$$

$$f_2 = \sum_{j=1}^{n} a_{2j}\tilde{\varphi}(x_j), \qquad \tilde{\mathbf{G}}\mathbf{a}_2 = \lambda_2 \mathbf{a}_2.$$

one has

$$0 \stackrel{?}{=} \langle f_1, f_2 \rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{n} a_{1i}\tilde{\varphi}(x_i), \sum_{j=1}^{n} a_{2j}\tilde{\varphi}(x_j) \right\rangle_{\mathcal{H}}$$

Orthogonal eigenvectors in kernel PCA

Taking two eigenvectors:

$$f_1 = \sum_{i=1}^{n} a_{1i}\tilde{\varphi}(x_i), \qquad \tilde{\mathbf{G}}\mathbf{a}_1 = \lambda_1 \mathbf{a}_1,$$

$$f_2 = \sum_{j=1}^{n} a_{2j}\tilde{\varphi}(x_j), \qquad \tilde{\mathbf{G}}\mathbf{a}_2 = \lambda_2 \mathbf{a}_2.$$

one has

$$0 \stackrel{?}{=} \langle f_1, f_2 \rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{n} a_{1i}\tilde{\varphi}(x_i), \sum_{j=1}^{n} a_{2j}\tilde{\varphi}(x_j) \right\rangle_{\mathcal{H}} = \mathbf{a}_1^T \tilde{\mathbf{G}}\mathbf{a}_2$$

Orthogonal eigenvectors in kernel PCA

Taking two eigenvectors:

$$f_1 = \sum_{i=1}^{n} a_{1i}\tilde{\varphi}(x_i), \qquad \tilde{\mathbf{G}}\mathbf{a}_1 = \lambda_1 \mathbf{a}_1,$$

$$f_2 = \sum_{j=1}^{n} a_{2j}\tilde{\varphi}(x_j), \qquad \tilde{\mathbf{G}}\mathbf{a}_2 = \lambda_2 \mathbf{a}_2.$$

one has

$$0 \stackrel{?}{=} \langle f_1, f_2 \rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{n} a_{1i}\tilde{\varphi}(x_i), \sum_{j=1}^{n} a_{2j}\tilde{\varphi}(x_j) \right\rangle_{\mathcal{H}} = \mathbf{a}_1^T \tilde{\mathbf{G}}\mathbf{a}_2 = \mathbf{a}_1^T \lambda_2 \mathbf{a}_2.$$

Orthogonality ⇒ projection is easy

- Projection of a new x^* to the first d-PCs:

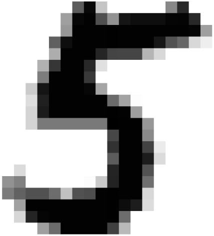
$$\Pi\left[\tilde{\varphi}\left(x^*\right)\right] = \sum_{j=1}^{d} \left\langle \tilde{\varphi}\left(x^*\right), f_j \right\rangle_{\mathcal{H}} f_j.$$

Orthogonality ⇒ projection is easy

- Projection of a new x^* to the first d-PCs:

$$\Pi\left[\tilde{\varphi}\left(x^*\right)\right] = \sum_{j=1}^{d} \left\langle \tilde{\varphi}\left(x^*\right), f_j \right\rangle_{\mathcal{H}} f_j.$$

- The pre-image problem we solved in denoising:



$$\widehat{x^*} = \arg\min_{x\in\mathcal{X}} \left\| \tilde{\varphi}(x) - \Pi\left[\tilde{\varphi}\left(x^*\right)\right] \right\|_{\mathcal{H}}^2.$$

Canonical Correlation Analysis (CCA)

CCA definition

- Given a pair of random variables: $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2d}$.
- Find the directions $(\mathbf{a} \in \mathbb{R}^d, \mathbf{b} \in \mathbb{R}^d)$ in which $\mathbf{x}$ and $\mathbf{y}$ are maximally correlated:

$$\mathrm{CCA}(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{a},\mathbf{b}} \mathrm{corr}_{\mathbf{x},\mathbf{y}} \left(\mathbf{a}^T \mathbf{x}, \mathbf{b}^T \mathbf{y}\right).$$

Examples

follow where dependence measures are useful!

Outlier-robust image registration

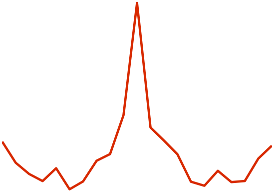
Given two images:

Goal: find the transformation which takes the right one to the left.

Outlier-robust image registration

Given two images:



Goal: find the transformation which takes the right one to the left.

Outlier-robust image registration: equations

- Reference image: $\mathbf{y}_{\text{ref}}$,
- test image: $\mathbf{y}_{\text{test}}$,
- possible transformations: $\boldsymbol{\Theta}$.

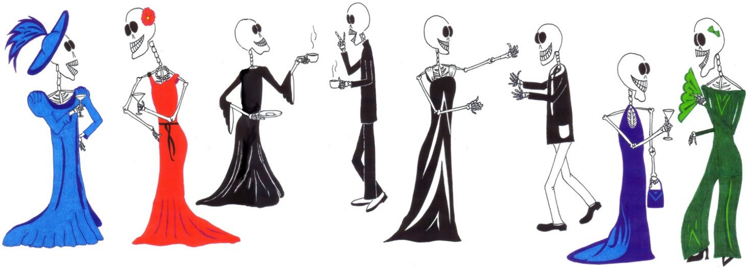
Objective:

$$J(\boldsymbol{\theta}) = \underbrace{\mathrm{I}(\mathbf{y}_{\text{ref}}, \mathbf{y}_{\text{test}}(\boldsymbol{\theta}))}_{\text{similarity}} \rightarrow \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}.$$

In the example: $\mathrm{I} =$ Non-linear CCA.

Cocktail party problem:

- independent groups of people / music bands,
- observation = mixed sources.



ISA equations

Observation:

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad \mathbf{s} = \left[\mathbf{s}^1; \ldots; \mathbf{s}^M\right].$$

Goal: $\hat{\mathbf{s}}$ from $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$. Assumptions:

- independent groups: $I\left(\mathbf{s}^1, \ldots, \mathbf{s}^M\right) = 0$,
- $\mathbf{s}^m$-s: non-Gaussian,
- $\mathbf{A}$: invertible.

ISA solution

Find **W** which makes the estimated components independent:

$$\mathbf{y} = \mathbf{W}\mathbf{x} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right],$$

$$J(\mathbf{W}) = \mathrm{I}\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) \to \min_{\mathbf{W}}.$$

Recall: feature selection

- **Goal**: find
  - the feature subset (# of rooms, criminal rate, local taxes)
  - most relevant for house price prediction (y).

Here we consider a non-linear alternative of Lasso.

Feature selection: equations

- Features: $x^1, \ldots, x^F$. Subset: $S \subseteq \{1, \ldots, F\}$.
- MaxRelevance - MinRedundancy principle:

$$J(S) = \frac{1}{|S|} \sum_{i \in S} \mathrm{I}\left(x^i, y\right) - \frac{1}{|S|^2} \sum_{i,j \in S} \mathrm{I}\left(x^i, x^j\right) \to \max_{S \subseteq \{1,\ldots,F\}}.$$

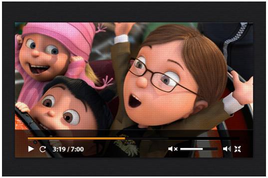
Example: independence testing-1

- We are given paired samples. Task: test independence.
- Examples:
  - (song, year of release) pairs

Example: independence testing-1

- We are given paired samples. Task: test independence.
- Examples:
  - (song, year of release) pairs

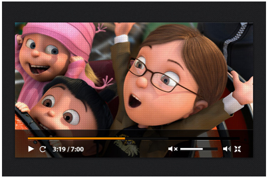
  - (video, caption) pairs



Example: independence testing-1

- We are given paired samples. Task: test independence.
- Examples:
  - (song, year of release) pairs

  - (video, caption) pairs



- $\{(x_i, y_i)\}_{i=1}^n \xrightarrow{?} \mathbb{P}_{xy} = \mathbb{P}_x \mathbb{P}_y$.

Example: independence testing-2

- How do we detect dependency? (paired samples)

x_1: Honourable senators, I have a question for the Leader of the Government in the Senate with regard to the support funding to farmers that has been announced. Most farmers have not received any money yet.

x_2: No doubt there is great pressure on provincial and municipal governments in relation to the issue of child care, but the reality is that there have been no cuts to child care funding from the federal government to the provinces. In fact, we have increased federal investments for early childhood development.

...

y_1: Honorables sénateurs, ma question s'adresse au leader du gouvernement au Sénat et concerne l'aide financiére qu'on a annoncée pour les agriculteurs. La plupart des agriculteurs n'ont encore rien reu de cet argent.

y_2: Il est évident que les ordres de gouvernements provinciaux et municipaux subissent de fortes pressions en ce qui concerne les services de garde, mais le gouvernement n'a pas réduit le financement qu'il verse aux provinces pour les services de garde. Au contraire, nous avons augmenté le financement fédéral pour le développement des jeunes enfants.

...

Example: independence testing-2

- How do we detect dependency? (paired samples)

x_1: Honourable senators, I have a question for the Leader of the Government in the Senate with regard to the support funding to farmers that has been announced. Most farmers have not received any money yet.

x_2: No doubt there is great pressure on provincial and municipal governments in relation to the issue of child care, but the reality is that there have been no cuts to child care funding from the federal government to the provinces. In fact, we have increased federal investments for early childhood development.

...

y_1: Honorables sénateurs, ma question s'adresse au leader du gouvernement au Sénat et concerne l'aide financiére qu'on a annoncée pour les agriculteurs. La plupart des agriculteurs n'ont encore rien reu de cet argent.

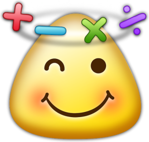
y_2: Il est évident que les ordres de gouvernements provinciaux et municipaux subissent de fortes pressions en ce qui concerne les services de garde, mais le gouvernement n'a pas réduit le financement qu'il verse aux provinces pour les services de garde. Au contraire, nous avons augmenté le financement fédéral pour le développement des jeunes enfants.

...

Are the French paragraphs translations of the English ones, or have nothing to do with it, i.e. $\mathbb{P}_{xy} = \mathbb{P}_x \mathbb{P}_y$?

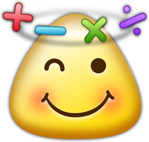
Towards non-linear CCA – History

- Given: random variable $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $(x, y) \sim \mathbb{P}_{xy}$.
- Goal: measure the dependence of x and y.



Towards non-linear CCA – History

- Given: random variable $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $(x, y) \sim \mathbb{P}_{xy}$.
- Goal: measure the dependence of x and y.
- Desiderata for a $Q(\mathbb{P}_{xy})$ independence measure:
  1. $Q(\mathbb{P}_{xy})$ is well-defined,
  2. $Q(\mathbb{P}_{xy}) \in [0, 1]$,
  3. $Q(\mathbb{P}_{xy}) = 0$ iff. $x \perp y$.
  4. $Q(\mathbb{P}_{xy}) = 1$ iff. $y = f(x)$ or $x = g(y)$.



- $Q(\mathbb{P}_{xy}) = \sup_{f,g} \mathrm{corr}(f(x), g(y))$ satisfies 1-4.

Independence measures

- $Q(\mathbb{P}_{xy}) = \sup_{f,g} \mathrm{corr}(f(x), g(y))$ satisfies 1-4.
- Too ambitious:
  - computationally intractable.
  - many functions.

Independence measures: restriction to continuous functions

- $C_b(\mathcal{X}) = \{f : \mathcal{X} \to \mathbb{R}$, bounded continuous$\}$ would also work.
- Still too large!

Independence measures: restriction to continuous functions

- $C_b(\mathcal{X}) = \{f : \mathcal{X} \to \mathbb{R}$, bounded continuous$\}$ would also work.
- Still too large!
- Idea:
  - certain $\mathcal{H}_k$ function classes are dense in $C_b(\mathcal{X})$.
  - computationally tractable.

KCCA: definition

- Given: $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.
- Associated:
  - feature maps $\varphi(x) = k(\cdot, x)$, $\psi(y) = \ell(\cdot, y)$,
  - RKHS-s $\mathcal{H}_k$, $\mathcal{H}_\ell$.

KCCA: definition

- Given: $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\ell: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.
- Associated:
  - feature maps $\varphi(x) = k(\cdot, x)$, $\psi(y) = \ell(\cdot, y)$,
  - RKHS-s $\mathcal{H}_k$, $\mathcal{H}_\ell$.
- KCCA measure of $(x, y) \in \mathcal{X} \times \mathcal{Y}$

$$\rho_{\text{KCCA}}(x, y; \mathcal{H}_k, \mathcal{H}_\ell) = \sup_{f \in \mathcal{H}_k, g \in \mathcal{H}_\ell} \mathrm{corr}(f(x), g(y)),$$

$$\mathrm{corr}(f(x), g(y)) = \frac{\mathrm{cov}_{xy}(f(x), g(y))}{\sqrt{\mathrm{var}_x f(x)\, \mathrm{var}_y g(y)}}.$$

KCCA: notes

- Optimization domain: $\mathcal{H}_k \times \mathcal{H}_\ell \ni (f, g)$.
- By reproducing property: we will get a finite-D task.
- k,ℓ linear: traditional CCA.

KCCA: notes

- Optimization domain: $\mathcal{H}_k \times \mathcal{H}_\ell \ni (f, g)$.
- By reproducing property: we will get a finite-D task.
- k,ℓ linear: traditional CCA.
- In practice:
  - we have $\{(x_n, y_n)\}_{n=1}^N$ samples from (x, y),

KCCA: notes

- Optimization domain: $\mathcal{H}_k \times \mathcal{H}_\ell \ni (f, g)$.
- By reproducing property: we will get a finite-D task.
- k,ℓ linear: traditional CCA.
- In practice:
  - we have $\{(x_n, y_n)\}_{n=1}^N$ samples from (x, y),
  - it is worth applying regularization

$$\hat{\rho}_{\text{KCCA}}(x, y; \mathcal{H}_k, \mathcal{H}_\ell, \kappa) = \sup_{f \in \mathcal{H}_k, g \in \mathcal{H}_\ell} \widehat{\text{corr}}(f(x), g(y); \kappa),$$

$$\widehat{\text{corr}}(f(x), g(y); \kappa) = \frac{\widehat{\text{cov}}_{xy}(f(x), g(y))}{\sqrt{\widehat{\text{var}}_x f(x) + \kappa \|f\|_{\mathcal{H}_k}^2}\sqrt{\widehat{\text{var}}_y g(y) + \kappa \|g\|_{\mathcal{H}_\ell}^2}}.$$

KCCA solution: one-page summary

- Representer theorem $\Rightarrow f = \sum_{i=1}^{N} c_i \tilde{\varphi}(x_i)$, $g = \sum_{i=1}^{N} d_i \tilde{\psi}(y_i)$.

KCCA solution: one-page summary

- Representer theorem $\Rightarrow f = \sum_{i=1}^{N} c_i \tilde{\varphi}(x_i)$, $g = \sum_{i=1}^{N} d_i \tilde{\psi}(y_i)$.
- Objective in terms of **c** and **d**:

$$\widehat{\rho_{\text{KCCA}}}(x,y) := \sup_{\mathbf{c}\in\mathbb{R}^N, \mathbf{d}\in\mathbb{R}^N} \frac{\mathbf{c}^T \tilde{\mathbf{G}}_x \tilde{\mathbf{G}}_y \mathbf{d}}{\sqrt{\mathbf{c}^T \left(\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N\right)^2 \mathbf{c}} \sqrt{\mathbf{d}^T \left(\tilde{\mathbf{G}}_y + \kappa \mathbf{I}_N\right)^2 \mathbf{d}}}.$$

KCCA solution: one-page summary

- Representer theorem $\Rightarrow f = \sum_{i=1}^{N} c_i \tilde{\varphi}(x_i)$, $g = \sum_{i=1}^{N} d_i \tilde{\psi}(y_i)$.
- Objective in terms of **c** and **d**:

$$\widehat{\rho_{\text{KCCA}}}(x,y) := \sup_{\mathbf{c}\in\mathbb{R}^N, \mathbf{d}\in\mathbb{R}^N} \frac{\mathbf{c}^T \tilde{\mathbf{G}}_x \tilde{\mathbf{G}}_y \mathbf{d}}{\sqrt{\mathbf{c}^T \left(\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N\right)^2 \mathbf{c}} \sqrt{\mathbf{d}^T \left(\tilde{\mathbf{G}}_y + \kappa \mathbf{I}_N\right)^2 \mathbf{d}}}.$$

- Stationary points of $\widehat{\rho_{\text{KCCA}}}(x,y)$:

$$\mathbf{0} = \frac{\partial \widehat{\rho_{\text{KCCA}}}(x,y)}{\partial \mathbf{c}}, \qquad \mathbf{0} = \frac{\partial \widehat{\rho_{\text{KCCA}}}(x,y)}{\partial \mathbf{d}}.$$

KCCA solution: one-page summary

- Representer theorem $\Rightarrow f = \sum_{i=1}^{N} c_i \tilde{\varphi}(x_i)$, $g = \sum_{i=1}^{N} d_i \tilde{\psi}(y_i)$.
- Objective in terms of $\mathbf{c}$ and $\mathbf{d}$:

$$\widehat{\rho_{\text{KCCA}}}(x,y) := \sup_{\mathbf{c}\in\mathbb{R}^N, \mathbf{d}\in\mathbb{R}^N} \frac{\mathbf{c}^T \tilde{\mathbf{G}}_x \tilde{\mathbf{G}}_y \mathbf{d}}{\sqrt{\mathbf{c}^T \left(\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N\right)^2 \mathbf{c}} \sqrt{\mathbf{d}^T \left(\tilde{\mathbf{G}}_y + \kappa \mathbf{I}_N\right)^2 \mathbf{d}}}.$$

- Stationary points of $\widehat{\rho_{\text{KCCA}}}(x,y)$:

$$\mathbf{0} = \frac{\partial \widehat{\rho_{\text{KCCA}}}(x,y)}{\partial \mathbf{c}}, \qquad \mathbf{0} = \frac{\partial \widehat{\rho_{\text{KCCA}}}(x,y)}{\partial \mathbf{d}}.$$

- We just need the maximal eigenvalues ($\mathbf{Az} = \lambda \mathbf{Bz}$) of

$$\begin{bmatrix} (\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N)^2 & \tilde{\mathbf{G}}_x \tilde{\mathbf{G}}_y \\ \tilde{\mathbf{G}}_y \tilde{\mathbf{G}}_x & (\tilde{\mathbf{G}}_\mathbf{y} + \kappa \mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \lambda \begin{bmatrix} (\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N)^2 & \mathbf{0} \\ \mathbf{0} & (\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}.$$

KCCA: M-variables

2-variables $[(x,y)]$:

$$\begin{bmatrix} (\tilde{\mathbf{G}}_{\mathbf{x}} + \kappa \mathbf{I}_N)^2 & \tilde{\mathbf{G}}_x \tilde{\mathbf{G}}_y \\ \tilde{\mathbf{G}}_y \tilde{\mathbf{G}}_x & (\tilde{\mathbf{G}}_{\mathbf{y}} + \kappa \mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \lambda \begin{bmatrix} (\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N)^2 & \mathbf{0} \\ \mathbf{0} & (\tilde{\mathbf{G}}_x + \kappa \mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}$$

KCCA: M-variables

2-variables $[(x,y)]$:

$$\begin{bmatrix} (\tilde{\mathbf{G}}_{\mathbf{x}}+\kappa\mathbf{I}_N)^2 & \tilde{\mathbf{G}}_x\tilde{\mathbf{G}}_y \\ \tilde{\mathbf{G}}_y\tilde{\mathbf{G}}_x & (\tilde{\mathbf{G}}_{\mathbf{y}}+\kappa\mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \lambda \begin{bmatrix} (\tilde{\mathbf{G}}_x+\kappa\mathbf{I}_N)^2 & \mathbf{0} \\ \mathbf{0} & (\tilde{\mathbf{G}}_x+\kappa\mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}$$

For M-variables (pairwise dependence):

$$\begin{bmatrix} (\tilde{\mathbf{G}}_1+\kappa\mathbf{I}_N)^2 & \tilde{\mathbf{G}}_1\tilde{\mathbf{G}}_2 & \dots & \tilde{\mathbf{G}}_1\tilde{\mathbf{G}}_M \\ \tilde{\mathbf{G}}_2\tilde{\mathbf{G}}_1 & (\tilde{\mathbf{G}}_2+\kappa\mathbf{I}_N)^2 & \dots & \tilde{\mathbf{G}}_2\tilde{\mathbf{G}}_M \\ \vdots & \vdots & & \vdots \\ \tilde{\mathbf{G}}_M\tilde{\mathbf{G}}_1 & \tilde{\mathbf{G}}_M\tilde{\mathbf{G}}_2 & \dots & (\tilde{\mathbf{G}}_M+\kappa\mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_M \end{bmatrix} =$$

$$\lambda \begin{bmatrix} (\tilde{\mathbf{G}}_1+\kappa\mathbf{I}_N)^2 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & (\tilde{\mathbf{G}}_2+\kappa\mathbf{I}_N)^2 & \dots & \mathbf{0} \\ \vdots & \vdots & & \\ \mathbf{0} & \mathbf{0} & \dots & (\tilde{\mathbf{G}}_M+\kappa\mathbf{I}_N)^2 \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_M \end{bmatrix}.$$

KCCA as an independence measure

If $x \perp y$, then $\rho_{\text{KCCA}}(x, y; \mathcal{H}_k, \mathcal{H}_\ell, \kappa) = 0$. Opposite direction:

- For 'rich' $\mathcal{H}_k$, $\mathcal{H}_\ell$.

KCCA as an independence measure

If $x \perp y$, then $\rho_{\text{KCCA}}(x, y; \mathcal{H}_k, \mathcal{H}_\ell, \kappa) = 0$. Opposite direction:

- For 'rich' $\mathcal{H}_k$, $\mathcal{H}_\ell$.
- Enough: universal kernel.

KCCA as an independence measure

If $x \perp y$, then $\rho_{\text{KCCA}}(x, y; \mathcal{H}_k, \mathcal{H}_\ell, \kappa) = 0$. Opposite direction:

- For 'rich' $\mathcal{H}_k$, $\mathcal{H}_\ell$.
- Enough: universal kernel.
- Example ($\gamma > 0$):
  - Gaussian: $k(x, x') = e^{-\gamma \|x - x'\|_2^2}$.
  - Laplacian kernel: $k(x, x') = e^{-\gamma \|x - x'\|_2}$.

Universality

Definition

Assume:

- $\mathcal{X}$: compact metric space.
- k: continuous kernel on $\mathcal{X}$.

k is called universal if $\mathcal{H}_k$ is dense in $(C_b(\mathcal{X}), \|\cdot\|_\infty)$.

Properties of universal kernels

If k is universal, then

- $k(x,x) > 0$ for all $x \in \mathcal{X}$.

Properties of universal kernels

If k is universal, then

- $k(x,x) > 0$ for all $x \in \mathcal{X}$.
- Every restriction of k to an $\mathcal{X}' \subseteq \mathcal{X}$ compact set is universal.

Properties of universal kernels

If k is universal, then

- $k(x,x) > 0$ for all $x \in \mathcal{X}$.
- Every restriction of k to an $\mathcal{X}' \subseteq \mathcal{X}$ compact set is universal.
- $\varphi(x) = k(\cdot, x)$ is injective, i.e.

$$\rho_k(x,y) = \|\varphi(x) - \varphi(y)\|_{\mathcal{H}_k}$$

  is a metric.

Properties of universal kernels

If k is universal, then

- $k(x,x) > 0$ for all $x \in \mathcal{X}$.
- Every restriction of k to an $\mathcal{X}' \subseteq \mathcal{X}$ compact set is universal.
- $\varphi(x) = k(\cdot, x)$ is injective, i.e.

$$\rho_k(x,y) = \|\varphi(x) - \varphi(y)\|_{\mathcal{H}_k}$$

  is a metric.
- The normalized kernel

$$\tilde{k}(x,y) := \frac{k(x,y)}{\sqrt{k(x,x)k(y,y)}}$$

  is universal.

Universal Taylor kernels

- For an $C^\infty \ni f : (-r, r) \to \mathbb{R}$

$$f(t) = \sum_{n=0}^{\infty} a_n t^n \quad t \in (-r, r), r \in (0, \infty].$$

Universal Taylor kernels

- For an $C^{\infty} \ni f : (-r, r) \to \mathbb{R}$

$$f(t) = \sum_{n=0}^{\infty} a_n t^n \quad t \in (-r, r), r \in (0, \infty].$$

- If $a_n > 0 \ \forall n$, then

$$k(x, y) = f(\langle x, y \rangle)$$

is universal on $\mathcal{X} := \left\{ x \in \mathbb{R}^d : \|x\|_2 \leqslant \sqrt{r} \right\}$.

Universal kernels, $\alpha > 0$

- $k(\mathbf{x}, \mathbf{y}) = e^{\alpha \langle \mathbf{x}, \mathbf{y} \rangle}$: previous result with $a_n = \frac{\alpha^n}{n!}$.

Universal kernels, $\alpha > 0$

- $k(\mathbf{x},\mathbf{y}) = e^{\alpha\langle \mathbf{x},\mathbf{y}\rangle}$: previous result with $a_n = \frac{\alpha^n}{n!}$.
- $k(\mathbf{x},\mathbf{y}) = e^{-\alpha\|\mathbf{x}-\mathbf{y}\|_2^2}$: exp. kernel & normalization.

Universal kernels, $\alpha > 0$

- $k(\mathbf{x}, \mathbf{y}) = (1 - \langle \mathbf{x}, \mathbf{y} \rangle)^{-\alpha}$ binomial kernel
  - on $\mathcal{X}$ compact $\subset \{x \in \mathbb{R}^d : \|\mathbf{x}\|_2 < 1\}$.
  - $f(t) = (1 - t)^{-\alpha} = \sum_{n=0}^{\infty} \underbrace{\binom{-\alpha}{n}(-1)^n}_{>0} t^n \; (|t| < 1)$,

where $\binom{b}{n} = \sum_{i=1}^{n} \frac{b-i+1}{i}$.

KCCA estimation: ITE

Artifacts of too much free time

https://bitbucket.org/szzoli/ite-in-python/

KCCA estimation: ITE

Artifacts of too much free time

https://bitbucket.org/szzoli/ite-in-python/

Import ITE, generate observations:

```
>>> import ite
>>> from numpy.random import randn
>>> from numpy import array
>>> ds = array([2, 3, 4])
>>> t = 1000
>>> y = randn(t, sum(ds))
```

KCCA estimation: ITE

Estimate KCCA:

```
>>> co = ite.cost.BIKCCA()
>>> kcca = co.estimation(y, ds)
```

KCCA estimation: ITE

Estimate KCCA:

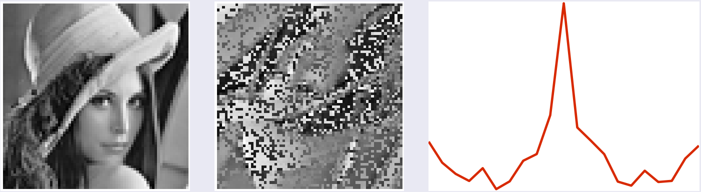
```
>>> co = ite.cost.BIKCCA()
>>> kcca = co.estimation(y, ds)
```

Alternative initialization:

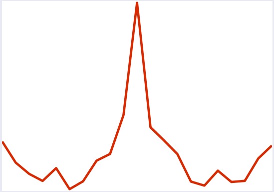
```
>>> co2 = ite.cost.BIKCCA(eta=1e-4, kappa=0.02)
>>> kcca2 = co2.estimation(y, ds)
```

where η: low-rank approximation, κ: regularization constant.

Recall: outlier-robust image registration (it was KCCA)



Recall: outlier-robust image registration (it was KCCA)



Can solving eigenvalue problems be avoided? Analytical solution?

CCA Alternative: HSIC

HSIC: intuition. $\mathcal{X}$: images, $\mathcal{Y}$: descriptions

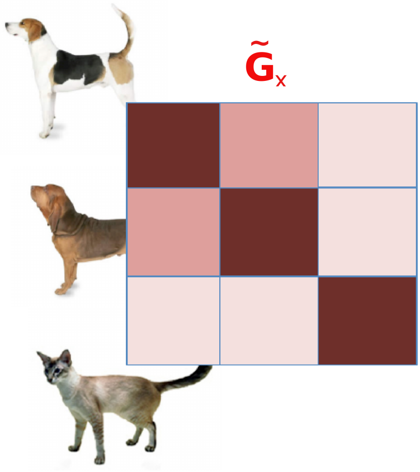
Their noses guide them through life, and they're never happier than when following an interesting scent. They need plenty of exercise, about an hour a day if possible.

A large animal who slings slobber, exudes a distinctive houndy odor, and wants nothing more than to follow his nose. They need a significant amount of exercise and mental stimulation.

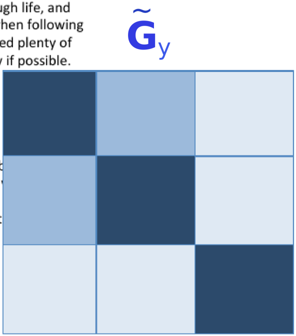
Known for their curiosity, intelligence, and excellent communication skills, the Javanese breed is perfect if you want a responsive, interactive pet, one that will blow in your ear and follow you everywhere.

Text from dogtime.com and petfinder.com

HSIC intuition: Gram matrices



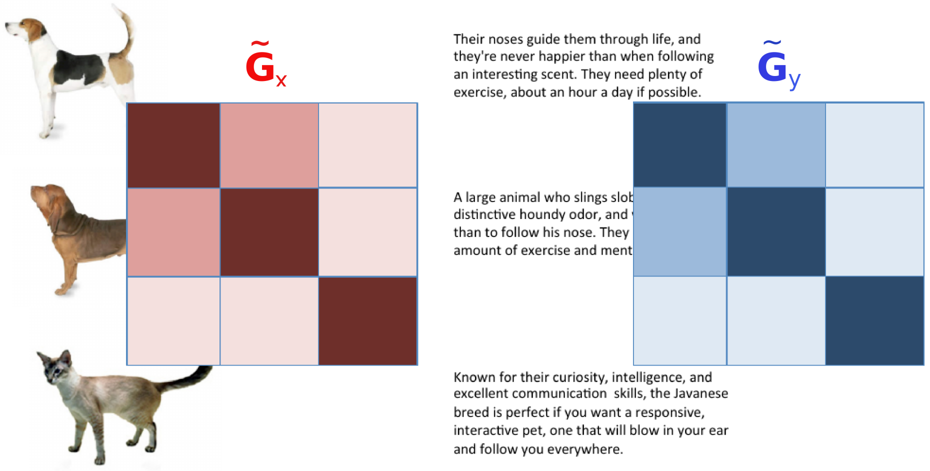
Their noses guide them through life, and they're never happier than when following an interesting scent. They need plenty of exercise, about an hour a day if possible.



A large animal who slings slob
distinctive houndy odor, and
than to follow his nose. They
amount of exercise and ment

Known for their curiosity, intelligence, and excellent communication skills, the Javanese breed is perfect if you want a responsive, interactive pet, one that will blow in your ear and follow you everywhere.

HSIC intuition: Gram matrices

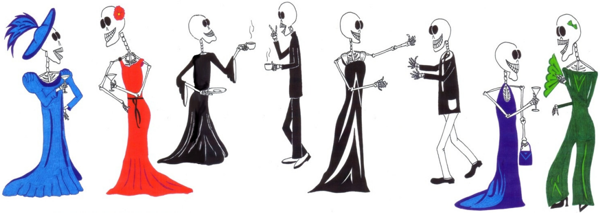


Empirical estimate†:

$$\widehat{\mathrm{HSIC}^2} = \frac{1}{n^2} \left\langle \tilde{\mathbf{G}}_x, \tilde{\mathbf{G}}_y \right\rangle_F . \quad \leftarrow \text{analytical!}$$

† Visual illustration credit: Arthur Gretton

Cocktail party: HSIC demo



ISA reminder

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \qquad \mathbf{s} = \left[\mathbf{s}^1; \ldots; \mathbf{s}^M\right],$$

where $\mathbf{s}^m$-s are non-Gaussian & independent.

- Goal: $\{\mathbf{x}_t\}_{t=1}^T \rightarrow \mathbf{W} = \mathbf{A}^{-1}, \{\mathbf{s}_t\}_{t=1}^T$,

ISA reminder

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \qquad \mathbf{s} = \left[\mathbf{s}^1; \ldots; \mathbf{s}^M\right],$$

where $\mathbf{s}^m$-s are non-Gaussian & independent.

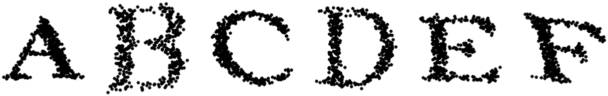
- Goal: $\{\mathbf{x}_t\}_{t=1}^T \rightarrow \mathbf{W} = \mathbf{A}^{-1}, \{\mathbf{s}_t\}_{t=1}^T$,
- Objective function:

$$\hat{\mathbf{s}} = \mathbf{W}\mathbf{x},$$

$$J(\mathbf{W}) = I\left(\hat{\mathbf{s}}^1, \ldots, \hat{\mathbf{s}}^M\right) \rightarrow \min_{\mathbf{W}}.$$

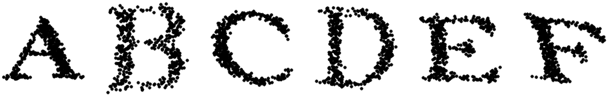
ISA: source, observation

- Hidden sources ($\mathbf{s}$):

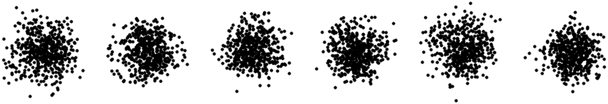


ISA: source, observation

- Hidden sources ($\mathbf{s}$):



- Observation ($\mathbf{x}$):



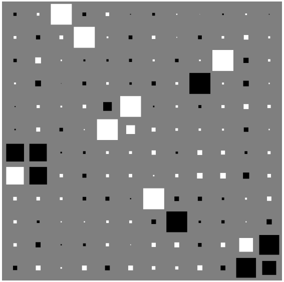
ISA: estimated sources using HSIC, ambiguity

- Estimated sources ($\hat{\mathbf{s}}$):

ISA: estimated sources using HSIC, ambiguity

- Estimated sources ($\hat{\mathbf{s}}$):

- Performance ($\hat{\mathbf{W}}\mathbf{A}$), ambiguity:

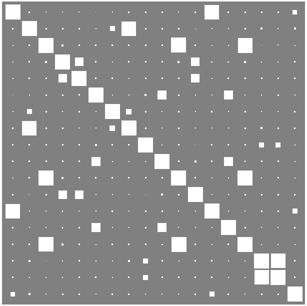


Conjecture: ISA separation theorem

- ISA = ICA + permutation.

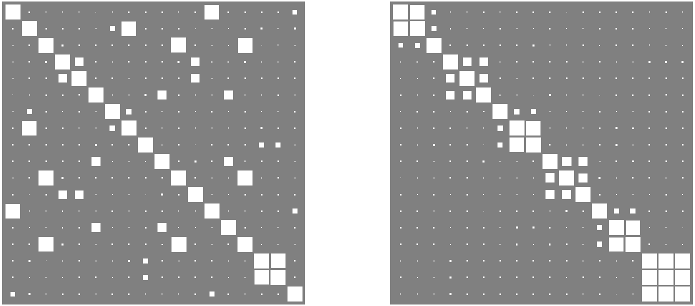
Conjecture: ISA separation theorem

- ISA = ICA + permutation. $\widehat{\text{HSIC}}(\hat{s}_i, \hat{s}_j)$. Here: $dim(\mathbf{s}^m) = 3$.



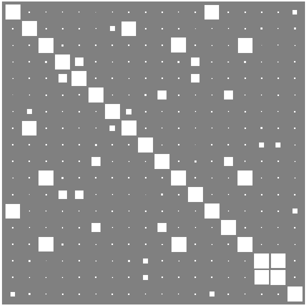
Conjecture: ISA separation theorem

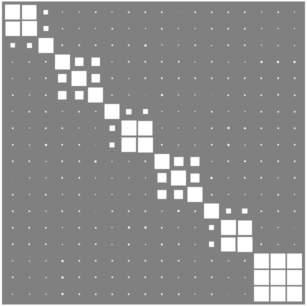
- ISA = ICA + permutation. $\widehat{\text{HSIC}}(\hat{s}_i, \hat{s}_j)$. Here: $dim(\mathbf{s}^m) = 3$.



Conjecture: ISA separation theorem

- ISA = ICA + permutation. $\widehat{\text{HSIC}}(\hat{s}_i, \hat{s}_j)$. Here: $dim(\mathbf{s}^m) = 3$.

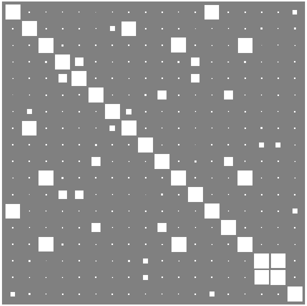


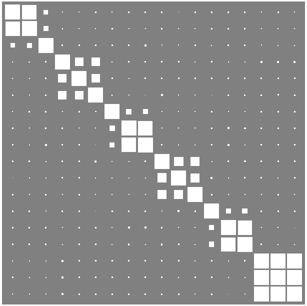


- Basis of the state-of-the-art ISA solvers.

Conjecture: ISA separation theorem

- ISA = ICA + permutation. $\widehat{\text{HSIC}}(\hat{s}_i, \hat{s}_j)$. Here: $dim(\mathbf{s}^m) = 3$.

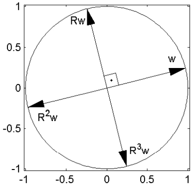


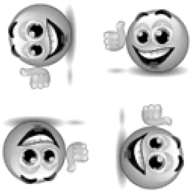


- Basis of the state-of-the-art ISA solvers.
- Sufficient conditions:
  - $\mathbf{s}^m$: spherical.

ISA separation theorem

For $dim(\mathbf{s}^m) = 2$: less is sufficient.



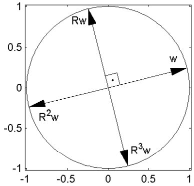


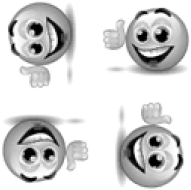
Invariance to

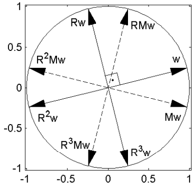
- 90° rotation: $f(u_1, u_2) = f(-u_2, u_1) = f(-u_1, -u_2) = f(u_2, -u_1)$.

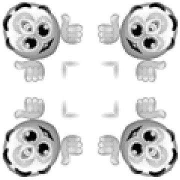
ISA separation theorem

For $dim(\mathbf{s}^m) = 2$: less is sufficient.







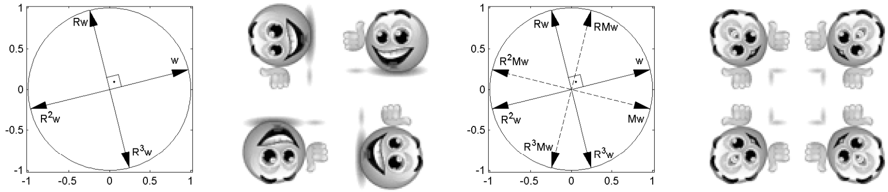


Invariance to

- 90° rotation: $f(u_1, u_2) = f(-u_2, u_1) = f(-u_1, -u_2) = f(u_2, -u_1)$.
- permutation and sign: $f(\pm u_1, \pm u_2) = f(\pm u_2, \pm u_1)$.

ISA separation theorem

For $dim(\mathbf{s}^m) = 2$: less is sufficient.



Invariance to

- 90° rotation: $f(u_1, u_2) = f(-u_2, u_1) = f(-u_1, -u_2) = f(u_2, -u_1)$.
- permutation and sign: $f(\pm u_1, \pm u_2) = f(\pm u_2, \pm u_1)$.
- L^p-spherical: $f(u_1, u_2) = h\left(\sum_i |u_i|^p\right) \quad (p > 0)$.

HSIC: view-1

Idea: $\mathbb{P}_{xy} \mapsto C_{xy}$.

- Covariance matrix

$$C_{xy} = \mathbb{E}_{xy}\left[(x - \mathbb{E}x)(y - \mathbb{E}y)^T\right]$$

HSIC: view-1

Idea: $\mathbb{P}_{xy} \mapsto C_{xy}$.

- Covariance matrix

$$C_{xy} = \mathbb{E}_{xy}\left[(x - \mathbb{E}x)(y - \mathbb{E}y)^T\right],$$

$$S = \|C_{xy}\|_F$$

HSIC: view-1

Idea: $\mathbb{P}_{xy} \mapsto C_{xy}$.

- Covariance matrix

$$C_{xy} = \mathbb{E}_{xy}\left[(x - \mathbb{E}x)(y - \mathbb{E}y)^T\right],$$

$$S = \|C_{xy}\|_F \stackrel{?}{=} 0$$

HSIC: view-1

Idea: $\mathbb{P}_{xy} \mapsto C_{xy}$.

- Covariance matrix

$$C_{xy} = \mathbb{E}_{xy}\left[(x - \mathbb{E}x)(y - \mathbb{E}y)^T\right],$$

$$S = \|C_{xy}\|_F \stackrel{?}{=} 0 \leftrightarrow \text{ linear dependence.}$$

HSIC: view-1

Idea: $\mathbb{P}_{xy} \mapsto C_{xy}$.

- Covariance matrix

$$C_{xy} = \mathbb{E}_{xy}\left[(x - \mathbb{E}x)(y - \mathbb{E}y)^T\right],$$

$$S = \left\|C_{xy}\right\|_F \overset{?}{=} 0 \leftrightarrow \text{ linear dependence.}$$

- Covariance operator: take features of x and y

$$C_{xy} = \mathbb{E}_{xy}\Big[\underbrace{(\varphi(x) - \mathbb{E}_x\varphi(x))}_{\text{centering in feature space}} \otimes (\psi(y) - \mathbb{E}_y\psi(y))\Big]$$

HSIC: view-1

Idea: $\mathbb{P}_{xy} \mapsto C_{xy}$.

- Covariance matrix

$$C_{xy} = \mathbb{E}_{xy}\left[(x - \mathbb{E}x)(y - \mathbb{E}y)^T\right],$$

$$S = \|C_{xy}\|_F \stackrel{?}{=} 0 \leftrightarrow \text{ linear dependence.}$$

- Covariance operator: take features of x and y

$$C_{xy} = \mathbb{E}_{xy}\Big[\underbrace{\left(\varphi(x) - \mathbb{E}_x\varphi(x)\right)}_{\text{centering in feature space}} \otimes \left(\psi(y) - \mathbb{E}_y\psi(y)\right)\Big],$$

$$S = \|C_{xy}\|_{HS} =: \text{HSIC}(\mathbb{P}_{xy}).$$

We capture non-linear dependencies via φ, ψ!

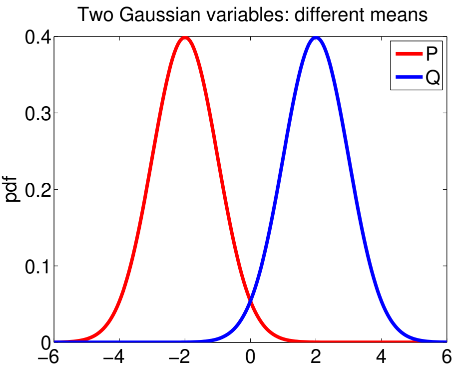
HSIC: view-2

- Independence: $\mathbb{P}_{xy} = \mathbb{P}_x \otimes \mathbb{P}_y$.

Questions

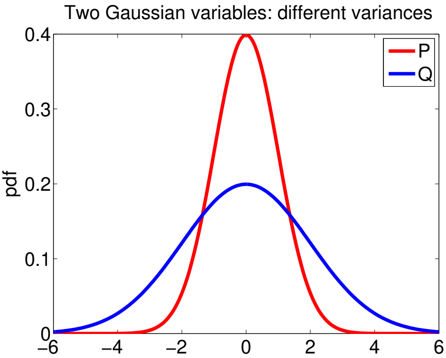
- How do we check this equality?
- How can distributions be represented?

- Given: 2 Gaussians with different means.
- Solution: *t*-test.



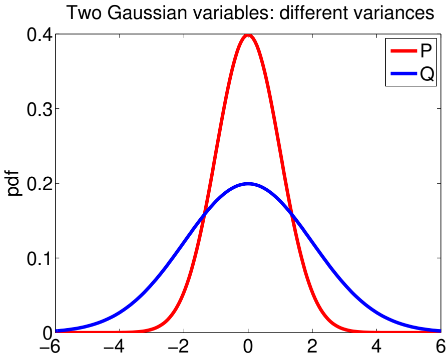
Representations of distributions: $\mathbb{E}X^2$

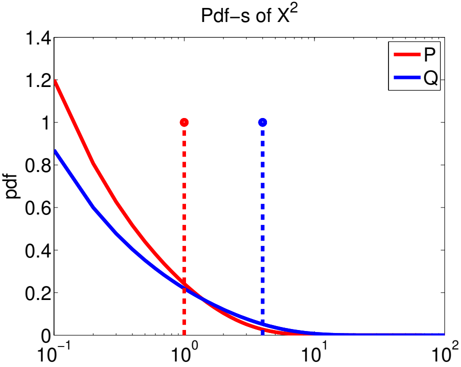
- Setup: 2 Gaussians; same means, different variances.
- Idea: look at the 2nd-order features of RVs.



Representations of distributions: $\mathbb{E}X^2$

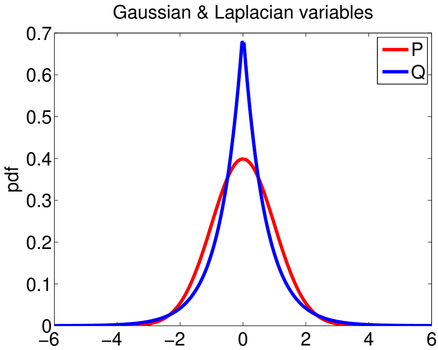
- Setup: 2 Gaussians; same means, different variances.
- Idea: look at the 2nd-order features of RVs.
- $\varphi(x) = x^2 \Rightarrow$ difference in $\mathbb{E}X^2$.





Representations of distributions: further moments

- Setup: a Gaussian and a Laplacian distribution.
- Challenge: their means *and* variances are the same.
- Idea: look at higher-order features.



$\varphi(\mathbf{x}) = e^{i\langle\cdot,\mathbf{x}\rangle}$: characteristic function, $\mathcal{X} = \mathbb{R}^d$.

Distribution representation via functions

$$\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x).$$

Distribution representation via functions

$$\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x).$$

- Cdf:

$$\mathbb{P} \mapsto \mathsf{F}_{\mathbb{P}}(z) = \mathbb{E}_{x \sim \mathbb{P}} \chi_{(-\infty, z)}(x).$$

Distribution representation via functions

$$\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x).$$

- Cdf:

$$\mathbb{P} \mapsto \mathsf{F}_{\mathbb{P}}(z) = \mathbb{E}_{x \sim \mathbb{P}} \chi_{(-\infty, z)}(x).$$

- Characteristic function:

$$\mathbb{P} \mapsto \mathsf{c}_{\mathbb{P}}(z) = \int e^{i\langle z, x\rangle} \mathrm{d}\mathbb{P}(x).$$

Distribution representation via functions

$$\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x).$$

- Cdf:

$$\mathbb{P} \mapsto \mathsf{F}_{\mathbb{P}}(z) = \mathbb{E}_{x \sim \mathbb{P}} \chi_{(-\infty,z)}(x).$$

- Characteristic function:

$$\mathbb{P} \mapsto \mathsf{c}_{\mathbb{P}}(z) = \int e^{i\langle z,x \rangle} \mathrm{d}\mathbb{P}(x).$$

- Moment generating function:

$$\mathbb{P} \mapsto \mathsf{M}_{\mathbb{P}}(z) = \int e^{\langle z,x \rangle} \mathrm{d}\mathbb{P}(x).$$

Distribution representation via functions

$$\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x).$$

- Cdf:

$$\mathbb{P} \mapsto \mathsf{F}_{\mathbb{P}}(z) = \mathbb{E}_{x \sim \mathbb{P}} \chi_{(-\infty, z)}(x).$$

- Characteristic function:

$$\mathbb{P} \mapsto \mathsf{c}_{\mathbb{P}}(z) = \int e^{i\langle z, x\rangle} \mathrm{d}\mathbb{P}(x).$$

- Moment generating function:

$$\mathbb{P} \mapsto \mathsf{M}_{\mathbb{P}}(z) = \int e^{\langle z, x\rangle} \mathrm{d}\mathbb{P}\left(x\right).$$

Trick

φ: on any kernel-endowed domain! $\varphi(x) := k(\cdot, x)$, $\mu_{\mathbb{P}} \in \mathcal{H}_k$.

We got

- Mean embedding:

$$\mu_{\mathbb{P}} := \int_{\mathcal{X}} \varphi(x)\, \mathrm{d}\mathbb{P}(x)$$

We got

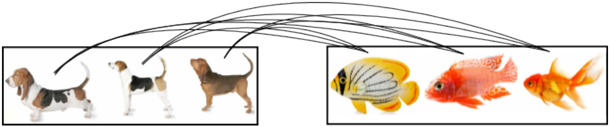
- Mean embedding:

$$\mu_{\mathbb{P}} := \int_{\mathcal{X}} \underbrace{\varphi(x)}_{k(\cdot,x)} \,\mathrm{d}\mathbb{P}(x) \in \mathcal{H}_k.$$

- Maximum mean discrepancy:

$$\text{MMD}_k(\mathbb{P},\mathbb{Q}) := \|\mu_k(\mathbb{P}) - \mu_k(\mathbb{Q})\|_{\mathcal{H}_k}.$$

Recall: $\langle \mu_k(\hat{\mathbb{P}}), \mu_k(\hat{\mathbb{Q}}) \rangle_{\mathcal{H}_k}$



We got

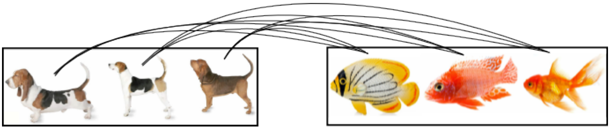
- Mean embedding:

$$\mu_{\mathbb{P}} := \int_{\mathcal{X}} \underbrace{\varphi(x)}_{k(\cdot,x)} \,\mathrm{d}\mathbb{P}(x) \in \mathcal{H}_k.$$

- Maximum mean discrepancy:

$$\mathrm{MMD}_k(\mathbb{P},\mathbb{Q}) := \|\mu_k(\mathbb{P}) - \mu_k(\mathbb{Q})\|_{\mathcal{H}_k}.$$

Recall: $\left\langle \mu_k(\hat{\mathbb{P}}), \mu_k(\hat{\mathbb{Q}}) \right\rangle_{\mathcal{H}_k}$



- Hilbert-Schmidt independence criterion, $k = \otimes_{m=1}^{M} k_m$:

$$\mathrm{HSIC}_k(\mathbb{P}) := \mathrm{MMD}_k\left(\mathbb{P}, \otimes_{m=1}^{M} \mathbb{P}_m\right).$$

MMD $\xrightarrow{\text{spec.}}$ HSIC

MMD with $k = \otimes_{m=1}^{M} k_m$:

$$k\left(x, x'\right) := \prod_{m=1}^{M} k_m\left(x_m, x'_m\right),$$

$$\text{HSIC}_k\left(\mathbb{P}\right) = \text{MMD}_k\left(\mathbb{P}, \otimes_{m=1}^{M} \mathbb{P}_m\right).$$

MMD with $k = \otimes_{m=1}^{M} k_m$:

$$k(x, x') := \prod_{m=1}^{M} k_m(x_m, x'_m),$$

$$\text{HSIC}_k(\mathbb{P}) = \text{MMD}_k\left(\mathbb{P}, \otimes_{m=1}^{M} \mathbb{P}_m\right).$$

Applications:

- blind source separation,
- feature selection, post selection inference,
- independence testing, causal inference.

MMD $\xrightarrow{\text{spec.}}$ HSIC

MMD with $k = \otimes_{m=1}^{M} k_m$:

$$k(x, x') := \prod_{m=1}^{M} k_m(x_m, x'_m),$$

$$\mathrm{HSIC}_k(\mathbb{P}) = \mathrm{MMD}_k\left(\mathbb{P}, \otimes_{m=1}^{M} \mathbb{P}_m\right).$$

Applications:

- blind source separation,
- feature selection, post selection inference,
- independence testing, causal inference.

The 2 views are equivalent; we estimated HSIC empirically.

Mean embedding, MMD: applications

Applications:

- two-sample testing,
- domain adaptation, -generalization,
- kernel Bayesian inference,
- approximate Bayesian computation, probabilistic programming,
- model criticism, goodness-of-fit,
- distribution classification, distribution regression,
- topological data analysis.

Critical in applications

When is

- $\text{MMD}_k(\mathbb{P},\mathbb{Q}) := \|\mu_k(\mathbb{P}) - \mu_k(\mathbb{Q})\|_{\mathcal{H}_k}$ a metric? In this case k is called characteristic.
- $\text{HSIC}_k(\mathbb{P})$ an independence measure?

Critical in applications

When is

- $\mathrm{MMD}_k(\mathbb{P},\mathbb{Q}) := \|\mu_k(\mathbb{P}) - \mu_k(\mathbb{Q})\|_{\mathcal{H}_k}$ a metric? In this case k is called characteristic.
- $\mathrm{HSIC}_k(\mathbb{P})$ an independence measure?

MMD: for continuous, bounded, shift-invariant k

- By the Bochner's theorem:

$$k(\mathbf{x},\mathbf{x}') = k_0(\mathbf{x}-\mathbf{x}') = \int_{\mathbb{R}^d} e^{i\langle \mathbf{x}-\mathbf{x}',\boldsymbol{\omega}\rangle} \mathrm{d}\Lambda(\boldsymbol{\omega}).$$

Critical in applications

When is

- $\mathrm{MMD}_k(\mathbb{P},\mathbb{Q}) := \|\mu_k(\mathbb{P}) - \mu_k(\mathbb{Q})\|_{\mathcal{H}_k}$ a metric? In this case k is called characteristic.
- $\mathrm{HSIC}_k(\mathbb{P})$ an independence measure?

MMD: for continuous, bounded, shift-invariant k

- By the Bochner's theorem:

$$k(\mathbf{x},\mathbf{x}') = k_0(\mathbf{x}-\mathbf{x}') = \int_{\mathbb{R}^d} e^{i\langle \mathbf{x}-\mathbf{x}',\boldsymbol{\omega}\rangle} \mathrm{d}\Lambda(\boldsymbol{\omega}).$$

- $\Rightarrow$ MMD in terms of characteristic functions:

$$\mathrm{MMD}_k^2(\mathbb{P},\mathbb{Q}) = \|c_{\mathbb{P}} - c_{\mathbb{Q}}\|^2_{L^2(\Lambda)}.$$

Simple description for shift-invariant kernels on $\mathbb{R}^d$

Theorem

k is characteristic iff. $supp(\Lambda) = \mathbb{R}^d$.

Simple description for shift-invariant kernels on $\mathbb{R}^d$

Theorem

k is characteristic iff. $supp(\Lambda) = \mathbb{R}^d$.

Example on $\mathbb{R}$:

kernel name	k_0	$\widehat{k_0}(\omega)$	$supp\left(\widehat{k_0}\right)$
Gaussian	$e^{-\frac{x^2}{2\sigma^2}}$	$\sigma e^{-\frac{\sigma^2\omega^2}{2}}$	$\mathbb{R}$
Laplacian	$e^{-\sigma\|x\|}$	$\sqrt{\frac{2}{\pi}}\frac{\sigma}{\sigma^2+\omega^2}$	$\mathbb{R}$
Sinc	$\frac{\sin(\sigma x)}{x}$	$\sqrt{\frac{\pi}{2}}\chi_{[-\sigma,\sigma]}(\omega)$	$[-\sigma, \sigma]$

Simple description for shift-invariant kernels on $\mathbb{R}^d$

Theorem

k is characteristic iff. supp$(\Lambda) = \mathbb{R}^d$.

Example on $\mathbb{R}$:

kernel name	k_0	$\widehat{k_0}(\omega)$	$supp\left(\widehat{k_0}\right)$
Gaussian	$e^{-\frac{x^2}{2\sigma^2}}$	$\sigma e^{-\frac{\sigma^2\omega^2}{2}}$	$\mathbb{R}$
Laplacian	$e^{-\sigma\|x\|}$	$\sqrt{\frac{2}{\pi}}\frac{\sigma}{\sigma^2+\omega^2}$	$\mathbb{R}$
Sinc	$\frac{\sin(\sigma x)}{x}$	$\sqrt{\frac{\pi}{2}}\chi_{[-\sigma,\sigma]}(\omega)$	$[-\sigma,\sigma]$

Note:

- universality $\Rightarrow$ characteristic.
- $k = \otimes_m k_m$: characteristic $\Rightarrow$ HSIC: ✓. How about in terms of k_m-s?

Description when HSIC is 'valid'

Proposition (characteristic property)

- $\otimes_{m=1}^{M} k_m$: *characteristic* $\Rightarrow$ $(k_m)_{m=1}^{M}$ *are characteristic.*
- $\not\Leftarrow$ $[|\mathcal{X}_m| = 2,\ k_m(x,x') = 2\delta_{x,x'} - 1]$

Description when HSIC is 'valid'

Proposition (characteristic property)

- $\otimes_{m=1}^{M} k_m$: *characteristic* $\Rightarrow$ $(k_m)_{m=1}^{M}$ *are characteristic.*
- $\not\Leftarrow$ $[|\mathcal{X}_m| = 2,\ k_m(x,x') = 2\delta_{x,x'} - 1]$

Proposition ($\mathcal{I}$-characteristic property)

- k_1, k_2: *characteristic* $\Rightarrow$ $k_1 \otimes k_2$: $\mathcal{I}$-*characteristic.*
- $\Leftarrow$: *for* $\forall M \geqslant 2$.
- k_1, k_2, k_3: *characteristic* $\not\Rightarrow$ $\otimes_{m=1}^{3} k_m$: $\mathcal{I}$-*characteristic [Ex].*

Description when HSIC is 'valid'

Proposition (characteristic property)

- $\otimes_{m=1}^M k_m$: *characteristic* $\Rightarrow$ $(k_m)_{m=1}^M$ *are characteristic.*
- $\not\Leftarrow$ *[*$|\mathcal{X}_m| = 2$, $k_m(x,x') = 2\delta_{x,x'} - 1$*]*

Proposition ($\mathcal{I}$-characteristic property)

- k_1, k_2: *characteristic* $\Rightarrow$ $k_1 \otimes k_2$: $\mathcal{I}$*-characteristic.*
- $\Leftarrow$: *for* $\forall M \geqslant 2$.
- k_1, k_2, k_3: *characteristic* $\not\Rightarrow$ $\otimes_{m=1}^3 k_m$: $\mathcal{I}$*-characteristic [Ex].*

Proposition ($\mathcal{X}_m = \mathbb{R}^{d_m}$, k_m: continuous, shift-invariant, bounded)

$(k_m)_{m=1}^M$*-s are characteristic* $\Leftrightarrow$ $\otimes_{m=1}^M k_m$: $\mathcal{I}$*-characteristic* $\Leftrightarrow$ $\otimes_{m=1}^M k_m$: *characteristic.*

Description when HSIC is 'valid'

Proposition (characteristic property)

- $\otimes_{m=1}^{M} k_m$: *characteristic* $\Rightarrow$ $(k_m)_{m=1}^{M}$ *are characteristic.*
- $\not\Leftarrow$ $[|\mathcal{X}_m| = 2,\ k_m(x,x') = 2\delta_{x,x'} - 1]$

Proposition ($\mathcal{I}$-characteristic property)

- k_1, k_2: *characteristic* $\Rightarrow$ $k_1 \otimes k_2$: $\mathcal{I}$*-characteristic.*
- $\Leftarrow$: *for* $\forall M \geqslant 2$.
- k_1, k_2, k_3: *characteristic* $\not\Rightarrow$ $\otimes_{m=1}^{3} k_m$: $\mathcal{I}$*-characteristic [Ex].*

Proposition ($\mathcal{X}_m = \mathbb{R}^{d_m}$, k_m: continuous, shift-invariant, bounded)

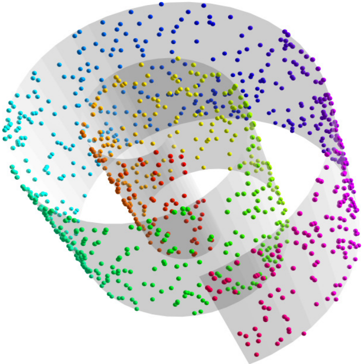
$(k_m)_{m=1}^{M}$*-s are characteristic* $\Leftrightarrow$ $\otimes_{m=1}^{M} k_m$: $\mathcal{I}$*-characteristic* $\Leftrightarrow$ $\otimes_{m=1}^{M} k_m$: *characteristic.*

Proposition (universality)

$\otimes_{m=1}^{M} k_m$: *universal* $\Leftrightarrow$ $(k_m)_{m=1}^{M}$ *are universal.*

Other dimensionality reduction techniques

Other non-linear methods



Goal: $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^D \overset{?}{\rightarrow} \{\mathbf{x}'_i\}_{i=1}^n \subset \mathbb{R}^d$, retaining the geometry of $\{\mathbf{x}_i\}_{i=1}^n$.

Multidimensional scaling (MDS)

- Given: $\mathbf{D} = [d_{ij}]_{i,j=1}^{n}$ distance matrix, $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$.

Multidimensional scaling (MDS)

- Given: $\mathbf{D} = [d_{ij}]_{i,j=1}^{n}$ distance matrix, $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$.
- Objective function:

$$\min_{\mathbf{X}'} \sum_{i,j} \underbrace{\left(d_{ij}^2 - \|\mathbf{x}_i' - \mathbf{x}_j'\|_2^2\right)}_{\text{preserve (large) distances}}, \text{ s.t. } \mathbf{x}_i' = \mathbf{W}\mathbf{x}_i, \|\mathbf{w}_i\|_2^2 = 1, \forall i.$$

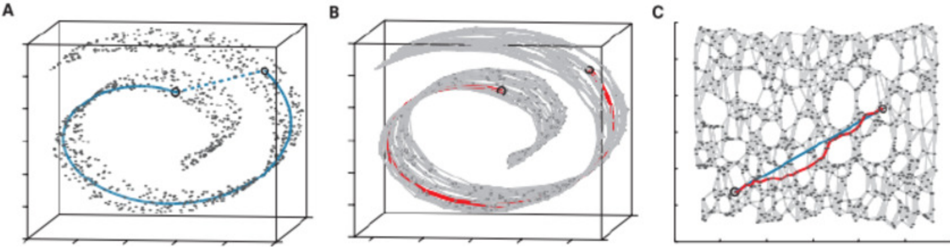
Multidimensional scaling (MDS)

- Given: $\mathbf{D} = [d_{ij}]_{i,j=1}^{n}$ distance matrix, $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$.
- Objective function:

$$\min_{\mathbf{X}'} \sum_{i,j} \underbrace{\left(d_{ij}^2 - \|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2\right)}_{\text{preserve (large) distances}}, \text{ s.t. } \mathbf{x}'_i = \mathbf{W}\mathbf{x}_i, \|\mathbf{w}_i\|_2^2 = 1, \forall i.$$

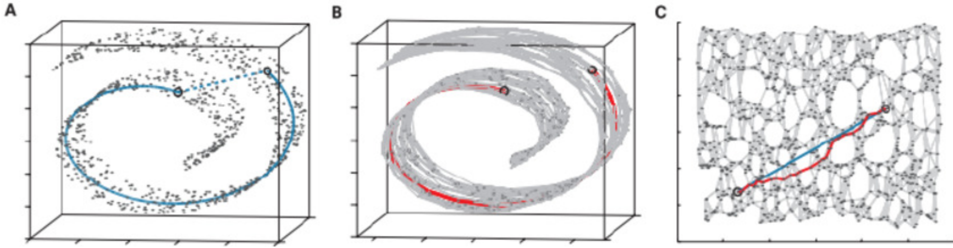
- Solution: $\mathbf{G} = \mathbf{X}^T\mathbf{X} = [\langle \mathbf{x}_i, \mathbf{x}_j \rangle]_{i,j=1}^{n}$ Gram matrix.
  1. Top d eigenvalues, eigenvectors of $\mathbf{G}$: λ_i, $\mathbf{v}_i$ $(i = 1, \ldots, d)$.
  2. $\mathbf{x}'_i = \sqrt{\lambda_i}\mathbf{v}_i$.

ISOMAP ⇐ MDS



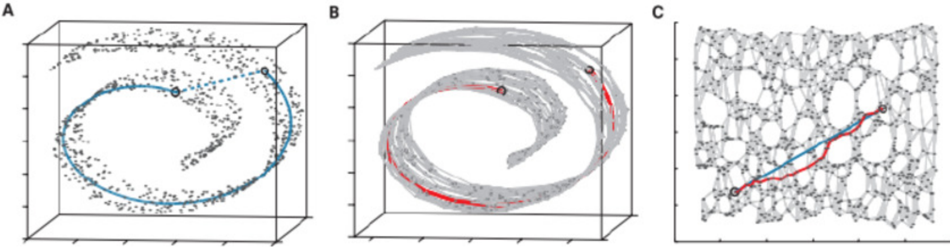
- Idea: For curved manifold let us rely on neighborhoods.

ISOMAP ⇐ MDS



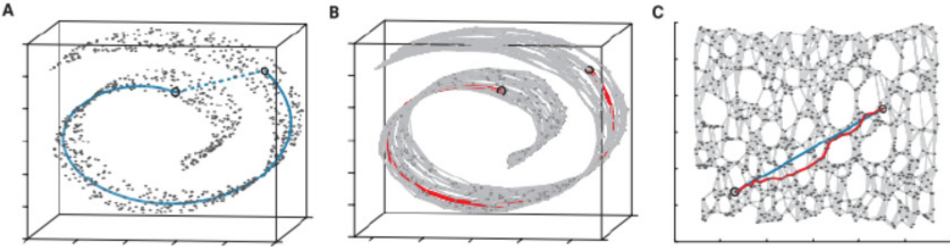
- Idea: For curved manifold let us rely on neighborhoods.
- Steps:
  1. $\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)$ = shortest path of $\mathbf{x}_i$ and $\mathbf{x}_j$ on kNN graph. (Dijkstra/Floyd's alg.)

ISOMAP $\Leftarrow$ MDS



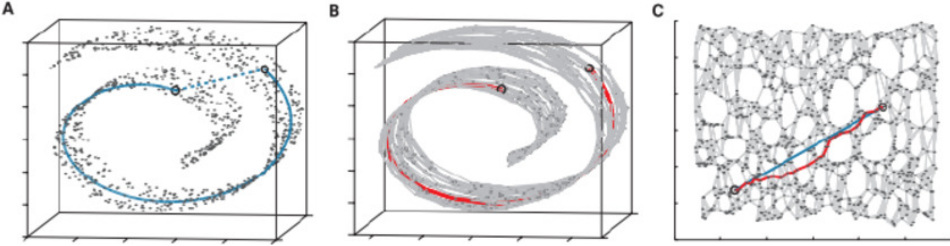
- Idea: For curved manifold let us rely on neighborhoods.
- Steps:
  1. $\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)$ = shortest path of $\mathbf{x}_i$ and $\mathbf{x}_j$ on kNN graph. (Dijkstra/Floyd's alg.)
  2. $\mathbf{D} := \left[\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)\right]$.

ISOMAP $\Leftarrow$ MDS



- Idea: For curved manifold let us rely on neighborhoods.
- Steps:
  1. $\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)$ = shortest path of $\mathbf{x}_i$ and $\mathbf{x}_j$ on kNN graph. (Dijkstra/Floyd's alg.)
  2. $\mathbf{D} := \left[\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)\right]$.
  3. Call MDS on $\mathbf{D}$.

ISOMAP ⇐ MDS



- Idea: For curved manifold let us rely on neighborhoods.
- Steps:
  1. $\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)$ = shortest path of $\mathbf{x}_i$ and $\mathbf{x}_j$ on kNN graph. (Dijkstra/Floyd's alg.)
  2. $\mathbf{D} := \left[\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)\right]$.
  3. Call MDS on $\mathbf{D}$.
- It can be slow.

Sammon mapping = MDS & local distance preservation

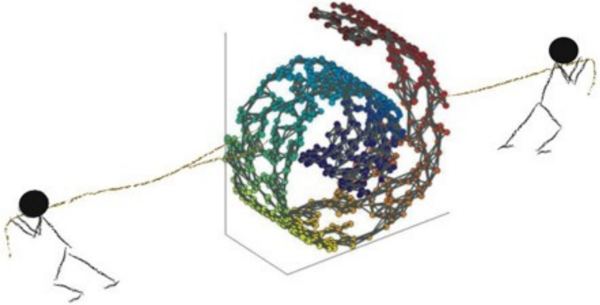
- Recall (MDS):

$$\min_{\mathbf{X}'} \sum_{i,j} \underbrace{\left(d_{ij}^2 - \left\|\mathbf{x}_i' - \mathbf{x}_j'\right\|_2^2\right)}_{\text{preserve (large) distances}}, \text{ s.t. } \mathbf{x}_i' = \mathbf{W}\mathbf{x}_i, \|\mathbf{w}_i\|_2^2 = 1, \forall i.$$

- MDS cares mostly about large distances.
- Sammon mapping: weights $:= \frac{1}{d_{ij}}$.

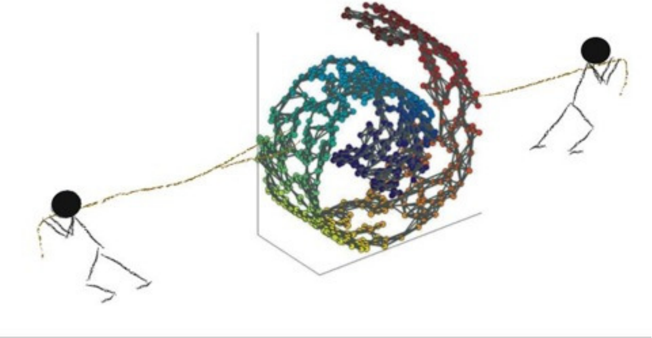
$$\min_{\mathbf{X}'} \frac{1}{\sum_{i\neq j} d_{ij}} \sum_{i\neq j} \frac{\left(d_{ij} - \left\|\mathbf{x}_i' - \mathbf{x}_j'\right\|_2\right)^2}{d_{ij}}.$$

MVU = MDS & explicit unfolding



$G :=$ kNN graph of $\{\mathbf{x}_i\}_{i=1}^n$.

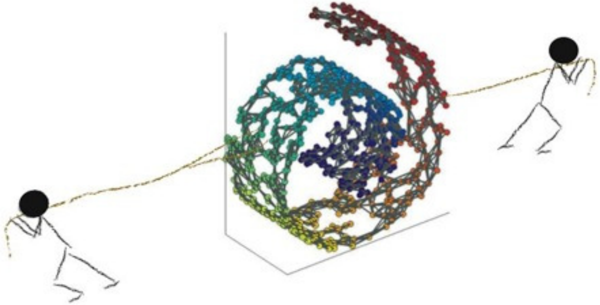
MVU = MDS & explicit unfolding



$G :=$ kNN graph of $\{\mathbf{x}_i\}_{i=1}^n$. Objective:

$$\max_{\mathbf{X}'} \sum_{ij} \left\|\mathbf{x}'_i - \mathbf{x}'_j\right\|_2^2 \text{ s.t. } \left\|\mathbf{x}'_i - \mathbf{x}'_j\right\|_2^2 = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \quad (i,j) \in G$$

MVU = MDS & explicit unfolding



$G :=$ kNN graph of $\{\mathbf{x}_i\}_{i=1}^n$. Objective:

$$\max_{\mathbf{X}'} \sum_{ij} \left\|\mathbf{x}'_i - \mathbf{x}'_j\right\|_2^2 \text{ s.t. } \left\|\mathbf{x}'_i - \mathbf{x}'_j\right\|_2^2 = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \quad (i,j) \in G$$

Leads to SDP: linear objective on positive-semidefinite matrices.

Locally linear embedding (LLE)

- Assumption: local linearity.
- Steps:
  1. G:=kNN graph $\Rightarrow \mathbf{x}_{i_j} := j^{th}$ NN of $\mathbf{x}_i$.

Locally linear embedding (LLE)

- Assumption: local linearity.
- Steps:
  1. G:=kNN graph $\Rightarrow \mathbf{x}_{i_j} := j^{th}$ NN of $\mathbf{x}_i$.
  2. $\mathbf{w}_i := \arg\min_{\mathbf{w}} \left\| \mathbf{x}_i - \sum_{j=1}^k w_{ij} \mathbf{x}_{i_j} \right\|_2$. Objective:

$$\min_{\mathbf{X}'} \sum_i \underbrace{\left\| \mathbf{x}'_i - \sum_j w_{ij} \mathbf{x}'_{i_j} \right\|_2^2}_{\text{local linearity preserving}} \quad \text{s.t.} \quad \underbrace{\left\| \mathbf{x}'^{(k)} \right\|_2^2 = 1, \forall k}_{\text{to avoid } \mathbf{X}' = \mathbf{0}}.$$

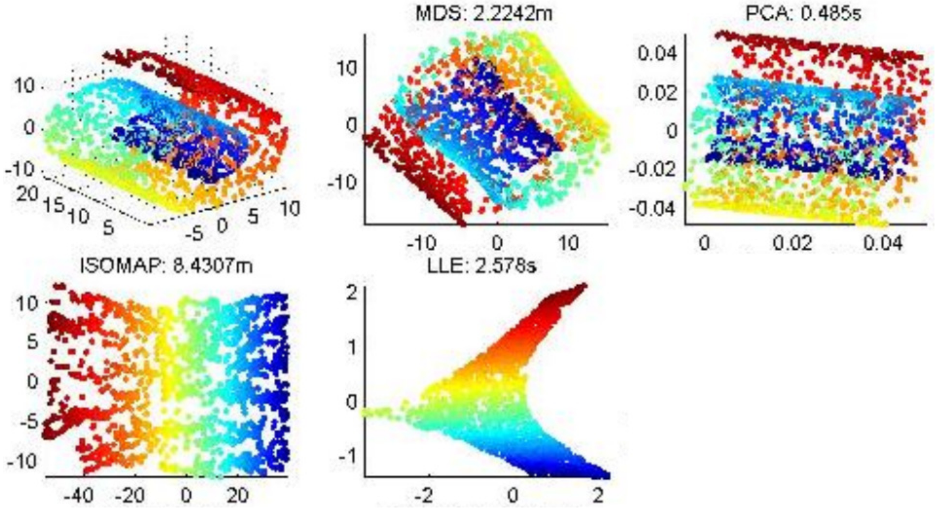
Locally linear embedding (LLE)

- Assumption: local linearity.
- Steps:
  1. G:=kNN graph $\Rightarrow \mathbf{x}_{i_j} := j^{th}$ NN of $\mathbf{x}_i$.
  2. $\mathbf{w}_i := \arg\min_{\mathbf{w}} \left\| \mathbf{x}_i - \sum_{j=1}^{k} w_{ij} \mathbf{x}_{i_j} \right\|_2$. Objective:

$$\min_{\mathbf{X}'} \sum_i \underbrace{\left\| \mathbf{x}'_i - \sum_j w_{ij} \mathbf{x}'_{i_j} \right\|_2^2}_{\text{local linearity preserving}} \quad \text{s.t.} \quad \underbrace{\left\| \mathbf{x}'^{(k)} \right\|_2^2 = 1, \forall k}_{\text{to avoid } \mathbf{X}' = \mathbf{0}}.$$

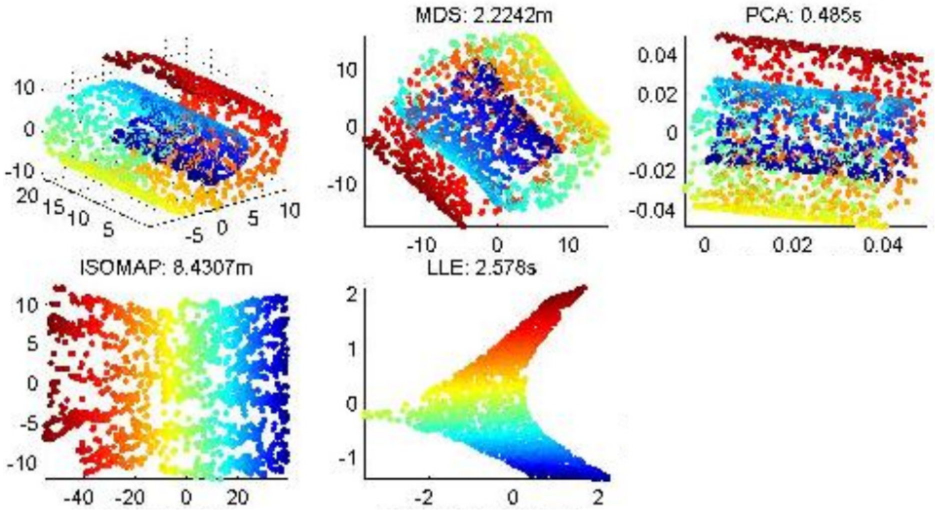
- Solution: from eigensystem of $(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$, $\mathbf{W} = 1 - \chi_G$.

Manifold embedding: demo[†]



[†] Todd Wittman

Manifold embedding: demo†



MDS, ISOMAP: slow. MDS, PCA: fail to unroll (no manifold info).

† Todd Wittman

Summary

Techniques:

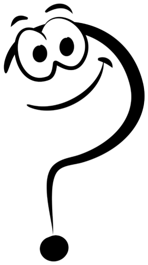
- PCA, KPCA: maximum variance projection.
- CCA, KCCA: maximally dependent projection.
- HSIC:
  - analytical KCCA alternative,
  - norm of covariance operator.
- MDS: (large) distance retaining.
- ISOMAP: geodesic distance preserving.
- Sammon mapping: distance retaining (including small ones).
- MVU: kNN distance preserving & explicit unrolling.
- LLE: local linearity preserving.

Summary

Applications:

- image compression & registration,
- non-linear feature selection,
- media annotation, translation testing,
- cocktail party (ISA).

Thank you for the attention!



Why do we get eigenvalue problems?

- $\mathbf{A} \in \mathbb{R}^{n \times n}$: symmetric matrix.
- Objective:

$$\max_{V \in \mathbb{R}^{n \times d}: \mathbf{V}^T \mathbf{V} = \mathbf{I}} \operatorname{Tr}\left(\mathbf{V}^T \mathbf{A} \mathbf{V}\right).$$

Why do we get eigenvalue problems?

- $\mathbf{A} \in \mathbb{R}^{n\times n}$: symmetric matrix.
- Objective:

$$\max_{V\in\mathbb{R}^{n\times d}:\mathbf{V}^T\mathbf{V}=\mathbf{I}} \operatorname{Tr}\left(\mathbf{V}^T\mathbf{A}\mathbf{V}\right).$$

- Optimal solution:
  - $\mathbf{V}^* = d$ leading eigenvectors of $\mathbf{A}$.
  - uniqueness up to subspace.

Why do we get generalized eigenvalue problems?

- $\mathbf{A} \in \mathbb{R}^{n \times n}$: symmetric matrix. $\mathbf{B} \in \mathbb{R}^{n \times n}$: positive definite.
- Objective:

$$\max_{V \in \mathbb{R}^{n \times d} : \mathbf{V}^T \mathbf{B} \mathbf{V} = \mathbf{I}} \operatorname{Tr}\left(\mathbf{V}^T \mathbf{A} \mathbf{V}\right).$$

Why do we get generalized eigenvalue problems?

- $\mathbf{A} \in \mathbb{R}^{n\times n}$: symmetric matrix. $\mathbf{B} \in \mathbb{R}^{n\times n}$: positive definite.
- Objective:

$$\max_{V\in\mathbb{R}^{n\times d}:\mathbf{V}^T\mathbf{B}\mathbf{V}=\mathbf{I}} \operatorname{Tr}\left(\mathbf{V}^T\mathbf{A}\mathbf{V}\right).$$

- Solution: $\mathbf{V}^* = d$ leading ($\mathbf{B}$-orthogonal) eigenvectors of the generalized eigenvalue problem

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}.$$