Manifold Learning and Classification for EEG Analysis

Zoltán Szabó
(CMAP, École Polytechnique)

Summer School on Mathematical and Computational Methods for Life Sciences

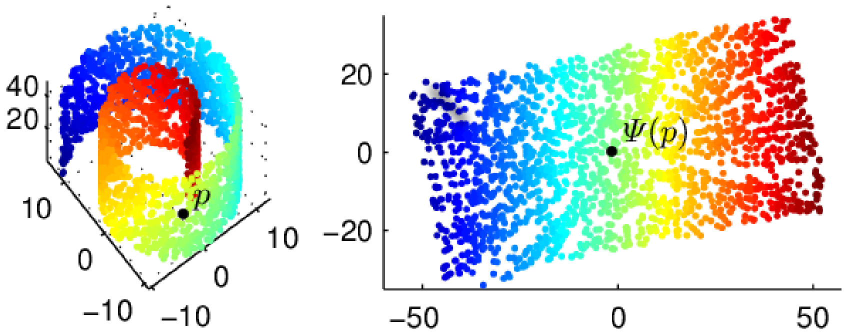
July 27, 2017

Contents

- Manifold learning (visualization).
- Classification (prediction).

Object of interest

Manifold learning

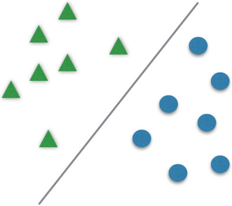


Manifold learning: intuition

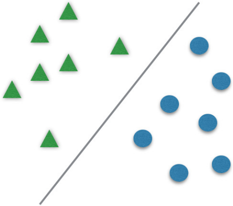
- Given: a set of observations $X = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^D$.
- Goal: find $X' = \{\mathbf{x}'_i\}_{i=1}^n \subset \mathbb{R}^d$ 'preserving' the geometry of X.
- $d \ll D$: compression (images, music, ...).

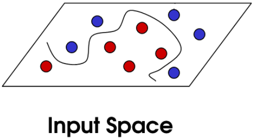
Classification

- Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{-1, 1\}$.
- Goal: find an f classifier such that $f(\mathbf{x}) \approx y$.

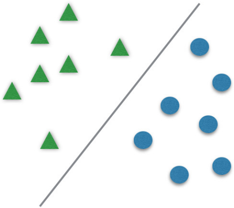


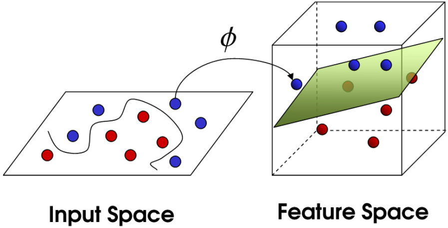
- Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{-1, 1\}$.
- Goal: find an f classifier such that $f(\mathbf{x}) \approx y$.



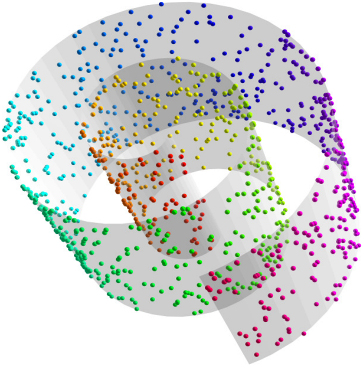


- Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{-1, 1\}$.
- Goal: find an f classifier such that $f(\mathbf{x}) \approx y$.





Manifold learning



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

Manifold learning

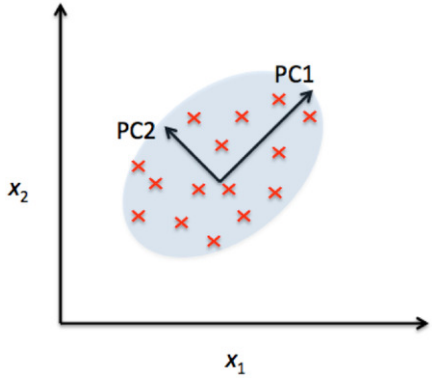
- In the following:
  - PCA (in detail).
  - MDS, ISOMAP, Sammon mapping.
  - MVU, LLE.

Manifold learning

- In the following:
  - PCA (in detail).
  - MDS, ISOMAP, Sammon mapping.
  - MVU, LLE.
- Toolbox
  [van der Maaten and Hinton, 2008, van der Maaten et al., 2009]:
  - `https://lvdmaaten.github.io/drtoolbox/`
  - 34 methods.

PCA: intuition

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

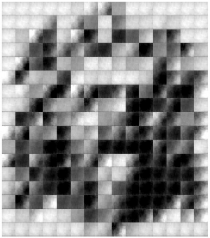


PCA example: 100%

(A)

PCA example: 100% → 1%

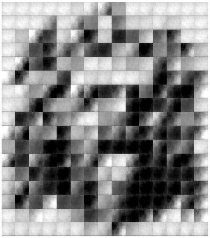
(A)



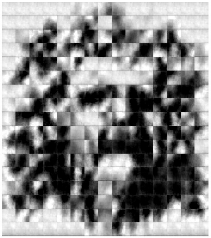
(B)

PCA example: 100% $\rightarrow$ 2%

(A)

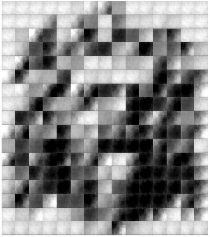


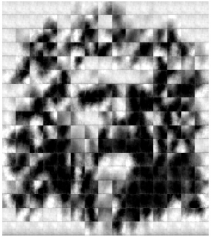
(B)



(C)


(A)

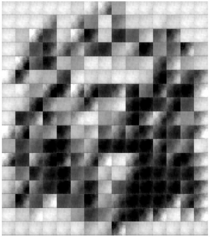

(B)


(C)

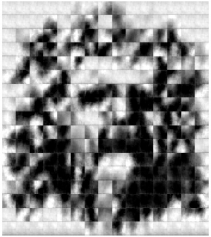

(D)

PCA example: 100% $\rightarrow$ 10%

(A)



(B)



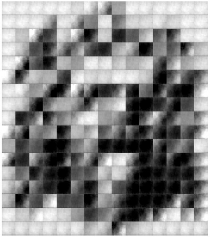
(C)

(D)

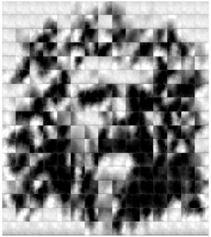
(E)

PCA example: 100% $\rightarrow$ 20%

(A)



(B)



(C)

(D)

(E)

(F)

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

PCA: min residual $\Leftrightarrow$ max squared projection

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

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:

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

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 $\Leftrightarrow$ max squared projection

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

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

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

Solution

maximizes the mean squared projection.

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

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

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

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

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

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

To sum up:

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

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

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

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

To sum up:

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

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

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

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

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

To sum up:

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

PCA: Optimization

By the bilinearity of *cov*:

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

PCA: Optimization

By the bilinearity of *cov*:

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

PCA: Optimization

By the bilinearity of *cov*:

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

Lagrange function, solving for 'derivatives $=0$':

By the bilinearity of *cov*:

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

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*:

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

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*:

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

Lagrange function, solving for 'derivatives = 0':

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

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

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

PCA: Optimization

By the bilinearity of *cov*:

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

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: $d \geqslant 1$

PCA ($d \geqslant 1$): basis, approximation

- 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:

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

PCA ($d \geqslant 1$): min residual $\Leftrightarrow$ max squared projection

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \left\|\mathbf{x} - \mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_2^2$$

PCA ($d \geqslant 1$): min residual $\Leftrightarrow$ max squared projection

$$\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2 = \left\|\mathbf{x}-\mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_2^2 = \mathbf{x}^T \underbrace{\left(\mathbf{I}-\mathbf{W}\mathbf{W}^T\right)\left(\mathbf{I}-\mathbf{W}\mathbf{W}^T\right)}_{=\mathbf{I}-2\mathbf{W}\mathbf{W}^T+\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T=\mathbf{I}-\mathbf{W}\mathbf{W}^T} \mathbf{x}$$

PCA ($d \geqslant 1$): min residual $\Leftrightarrow$ max squared projection

Using $\mathbf{W}^T\mathbf{W} = \mathbf{I}$

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \left\|\mathbf{x} - \mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_2^2 = \mathbf{x}^T \underbrace{\left(\mathbf{I} - \mathbf{W}\mathbf{W}^T\right)\left(\mathbf{I} - \mathbf{W}\mathbf{W}^T\right)}_{=\mathbf{I}-2\mathbf{W}\mathbf{W}^T+\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T=\mathbf{I}-\mathbf{W}\mathbf{W}^T} \mathbf{x}$$

$$= \|\mathbf{x}\|_2^2 - \left\|\mathbf{W}^T\mathbf{x}\right\|_2^2,$$

Using $\mathbf{W}^T\mathbf{W} = \mathbf{I}$

$$\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2 = \left\|\mathbf{x}-\mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_2^2 = \mathbf{x}^T \underbrace{\left(\mathbf{I}-\mathbf{W}\mathbf{W}^T\right)\left(\mathbf{I}-\mathbf{W}\mathbf{W}^T\right)}_{=\mathbf{I}-2\mathbf{W}\mathbf{W}^T+\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T=\mathbf{I}-\mathbf{W}\mathbf{W}^T} \mathbf{x}$$

$$= \|\mathbf{x}\|_2^2 - \left\|\mathbf{W}^T\mathbf{x}\right\|_2^2,$$

$$\mathbb{E}\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2 = \underbrace{\mathbb{E}\|\mathbf{x}\|_2^2}_{\text{independent of } \mathbf{W}} - \mathbb{E}\left\|\mathbf{W}^T\mathbf{x}\right\|_2^2.$$

PCA ($d \geqslant 1$): min residual $\Leftrightarrow$ max squared projection

Using $\mathbf{W}^T\mathbf{W} = \mathbf{I}$

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \left\|\mathbf{x} - \mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_2^2 = \mathbf{x}^T \underbrace{\left(\mathbf{I} - \mathbf{W}\mathbf{W}^T\right)\left(\mathbf{I} - \mathbf{W}\mathbf{W}^T\right)}_{=\mathbf{I}-2\mathbf{W}\mathbf{W}^T+\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T=\mathbf{I}-\mathbf{W}\mathbf{W}^T} \mathbf{x}$$

$$= \|\mathbf{x}\|_2^2 - \left\|\mathbf{W}^T\mathbf{x}\right\|_2^2,$$

$$\mathbb{E}\,\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \underbrace{\mathbb{E}\,\|\mathbf{x}\|_2^2}_{\text{independent of } \mathbf{W}} - \mathbb{E}\left\|\mathbf{W}^T\mathbf{x}\right\|_2^2.$$

Thus $\min_{\mathbf{W}} \mathbb{E}\,\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \Leftrightarrow \max_{\mathbf{W}} \mathbb{E}\left\|\mathbf{W}^T\mathbf{x}\right\|_2^2$.

PCA ($d \geqslant 1$): max squared projection $\Leftrightarrow$ max variance of projection

Let $\mathbf{y} = \mathbf{W}^T\mathbf{x}$:

$$\mathbb{E}\,\|\mathbf{y}\|_2^2 - \|\mathbb{E}\mathbf{y}\|_2^2 = var(\mathbf{y})?$$

PCA ($d \geqslant 1$): max squared projection $\Leftrightarrow$ max variance of projection

Let $\mathbf{y} = \mathbf{W}^T\mathbf{x}$:

$$\mathbb{E}\left\|\mathbf{y}\right\|_2^2 - \left\|\mathbb{E}\mathbf{y}\right\|_2^2 = var(\mathbf{y})?$$

$$= \mathbb{E}\left[\sum_i y_i^2\right] - \sum_i (\mathbb{E}y_i)^2 = \sum_i var(y_i) \Rightarrow$$

PCA ($d \geqslant 1$): max squared projection $\Leftrightarrow$ max variance of projection

Let $\mathbf{y} = \mathbf{W}^T\mathbf{x}$:

$$\begin{aligned}\mathbb{E}\left\|\mathbf{y}\right\|_2^2 - \left\|\mathbb{E}\mathbf{y}\right\|_2^2 &= var(\mathbf{y})?\\ &= \mathbb{E}\left[\sum_i y_i^2\right] - \sum_i (\mathbb{E}y_i)^2 = \sum_i var(y_i) \Rightarrow\end{aligned}$$

$$\mathbb{E}\left\|\mathbf{W}^T\mathbf{x}\right\|_2^2 - \Big\|\underbrace{\mathbb{E}[\mathbf{W}^T\mathbf{x}]}_{=\mathbf{W}^T\mathbb{E}\mathbf{x}=\mathbf{0}}\Big\|_2^2 = \sum_i var\left(\left(\mathbf{W}^T\mathbf{x}\right)_i\right) \to \max_{\mathbf{W}}.$$

PCA: $d \geqslant 1$

- The d principal components:

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

PCA: $d \geqslant 1$

- The d principal components:

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

- $\mathbf{\Sigma}$: symmetric, positive semi-definite $\Rightarrow$ $\{\mathbf{w}_i\}$: ONS, $\lambda_i \geqslant 0$.

PCA: $d \geqslant 1$

- The d principal components:

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

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

PCA: $d \geqslant 1$

- The d principal components:

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

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

$$R^2 = R^2(d) := \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{D} \lambda_i} \in [0,1].$$

PCA: $d \geqslant 1$

- The d principal components:

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

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

$$R^2 = R^2(d) := \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{D} \lambda_i} \in [0, 1].$$

- In practice: choose d such that $R^2 \approx 0.8 - 0.9$.

PCA/subspace alternatives

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. $\mathbf{x}'_i = \sqrt{\lambda_i}\mathbf{v}_i$.

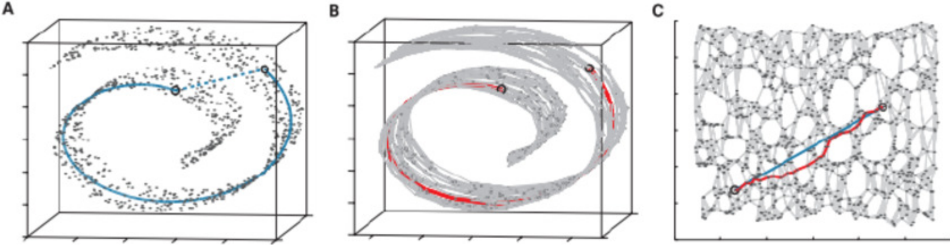
Multidimensional scaling (MDS)

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

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

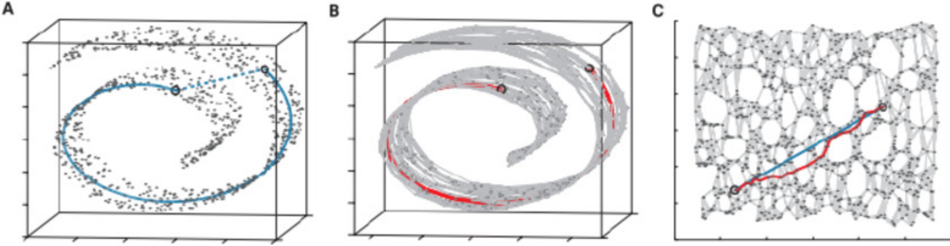
- Solution: $\mathbf{G} = \mathbf{X}^T\mathbf{X} = [\langle \mathbf{x}_i, \mathbf{x}_j \rangle]_{i,j=1}^{n}$ Gram matrix.
  1. Top d eigenvalues, eigenvectors of $\mathbf{G}$: λ_i, $\mathbf{v}_i$ $(i = 1, \ldots, d)$.
  2. $\mathbf{x}_i' = \sqrt{\lambda_i}\mathbf{v}_i$.
- Expensive computationally.

ISOMAP [Tenenbaum et al., 2000] $\Leftarrow$ MDS

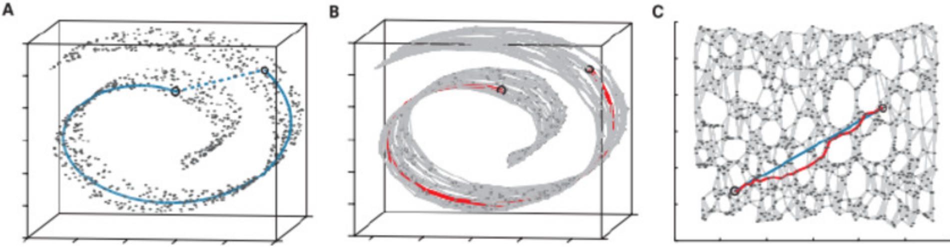


- Idea: For curved manifold rely on neighborhoods.

ISOMAP [Tenenbaum et al., 2000] $\Leftarrow$ MDS

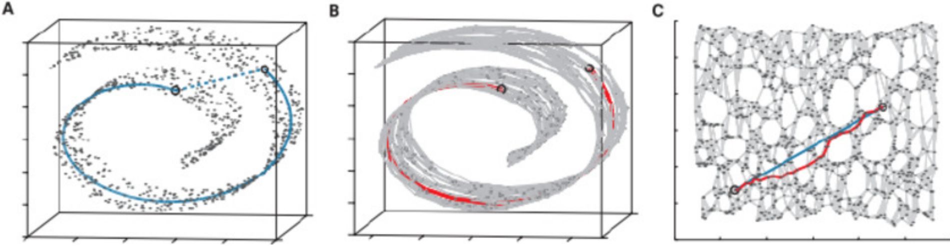


- Idea: For curved manifold rely on neighborhoods.
- Steps:
  1. $\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)$ = shortest path of $\mathbf{x}_i$ and $\mathbf{x}_j$ on kNN graph. (Dijkstra/Floyd's alg.)



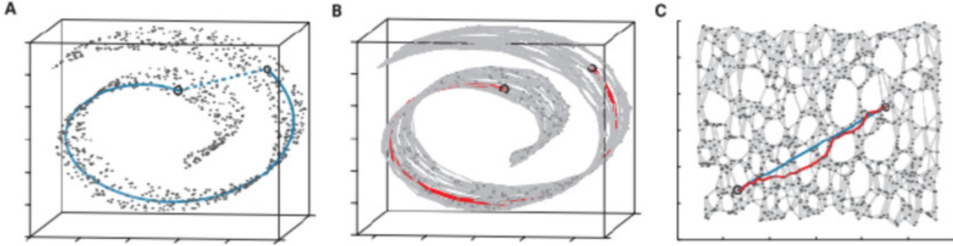
- Idea: For curved manifold rely on neighborhoods.
- Steps:
  1. $\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)$ = shortest path of $\mathbf{x}_i$ and $\mathbf{x}_j$ on kNN graph. (Dijkstra/Floyd's alg.)
  2. $\mathbf{D} := \left[\hat{d}_{\text{geodesic}}(\mathbf{x}_i, \mathbf{x}_j)\right]$.

ISOMAP [Tenenbaum et al., 2000] $\Leftarrow$ MDS



- Idea: For curved manifold 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 [Tenenbaum et al., 2000] $\Leftarrow$ MDS



- Idea: For curved manifold 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 [Torgerson, 1952]

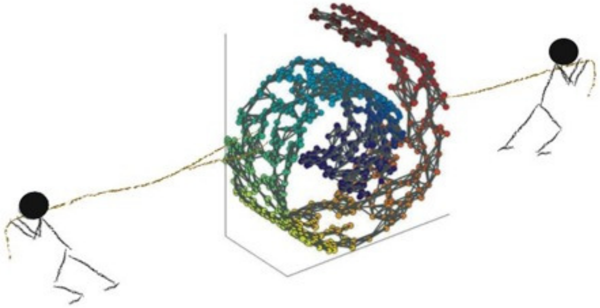
- Recall (MDS):

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

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

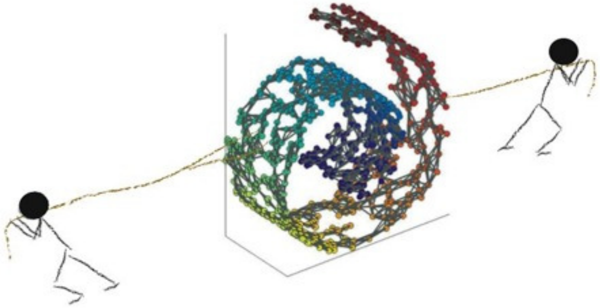
$$\min_{\mathbf{X}'} \frac{1}{\sum_{i \neq j} d_{ij}} \sum_{i \neq j} \frac{\left(d_{ij} - \left\|\mathbf{x}_i' - \mathbf{x}_j'\right\|_2\right)^2}{d_{ij}}.$$

MVU [Weinberger et al., 2004] = MDS & explicit unfolding



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

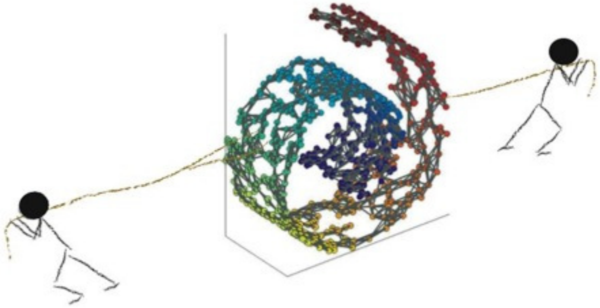
MVU [Weinberger et al., 2004] = MDS & explicit unfolding



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

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

MVU [Weinberger et al., 2004] = 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.

Locally linear embedding (LLE) [Roweis and Saul, 2000]

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

Locally linear embedding (LLE) [Roweis and Saul, 2000]

- 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:

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

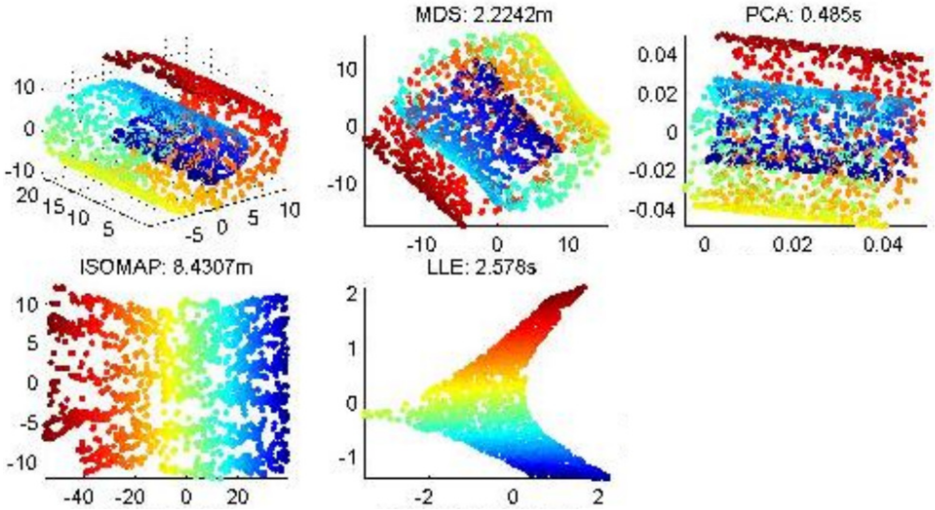
Locally linear embedding (LLE) [Roweis and Saul, 2000]

- 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:

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

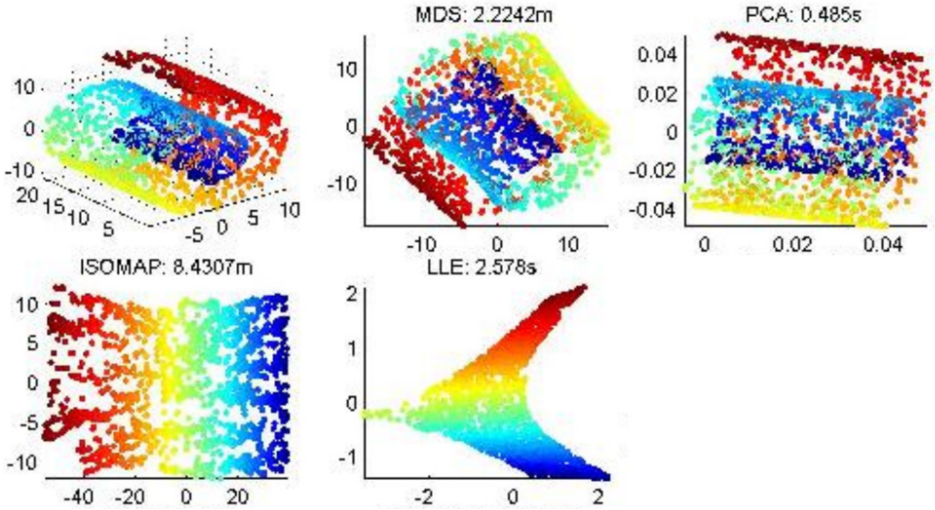
- Solution: from eigensystem of $(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$, $\mathbf{W} = 1 - \chi_G$.

Manifold embedding: demo[†]



[†]Todd Wittman

Manifold embedding: demo†



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

†Todd Wittman

Manifold embedding: summary

- PCA: linear subspace.
- 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.

Classification

Program

- kNN classifier.
- Sparse coding, structured sparse coding.
- SVM: linear, non-linear.

Classification: kNN

Task: recap

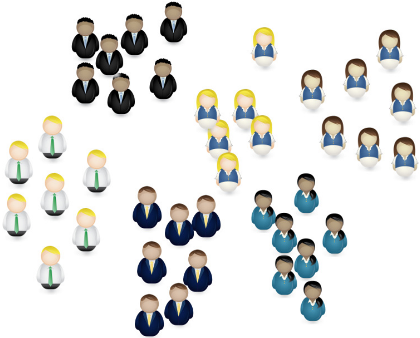
- Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{-1, 1\}$.
- Goal: find an f classifier s.t. $f(\mathbf{x}) \approx y$.

Task: recap

- Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{-1, 1\}$.
- Goal: find an f classifier s.t. $f(\mathbf{x}) \approx y$.
- In the EEG example:
  - $y_i = 1$ (calm), $y_i = -1$ (traumatic),
  - $\mathbf{x}_i$: i^{th} patient.

kNN classification

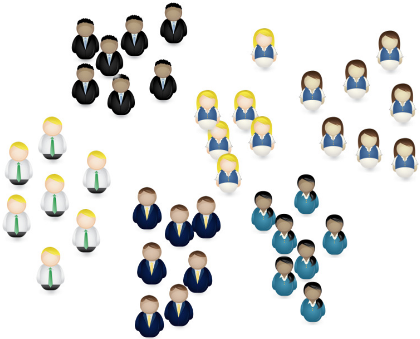
- Simplest decision rule[†]:



[†] kNN illustration credit: scikit-learn.

kNN classification

- Simplest decision rule[†]:



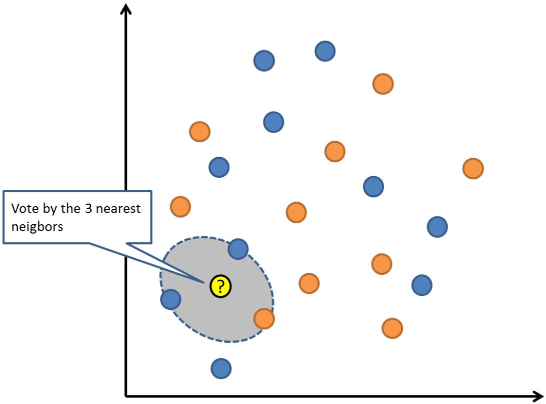
- Let $k = 1$. For a test $\mathbf{x}$, we predict the label of the closest point:

$$i^* := \arg\min_i \rho(\mathbf{x}, \mathbf{x}_i), \qquad \rho(\mathbf{x}, \mathbf{x}') = \left\|\mathbf{x} - \mathbf{x}'\right\|_2,$$
$$\hat{y} := y_{i*}.$$

[†]kNN illustration credit: scikit-learn.

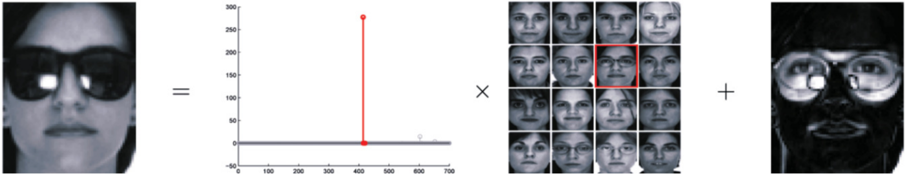
kNN classification: $k \geqslant 1$

- Generalization of the 1-NN idea.
- Majority vote of the k-nearest neighbors.



Classification: (structured) sparse coding.

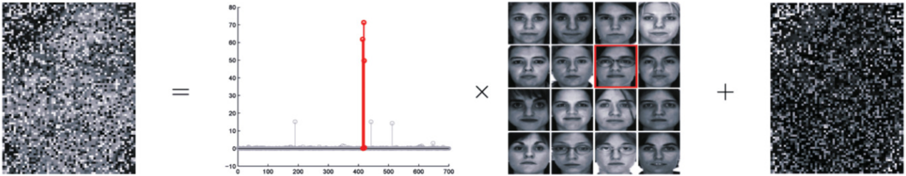
Demo: face recognition.



Idea [Wright et al., 2009, Wagner et al., 2009]:

- test image = sparse linear combination of the training set + error
- error = corruption/occlusion.

Classification as sparse coding – continued



- Nice performance despite severe corruption.

Problem formulation

Objective function (Lasso problem, EEG: $K = 2$):

$$\mathbf{A} := [\mathbf{x}_1, \ldots, \mathbf{x}_n],$$
$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{c}\|_2^2}_{\text{good approximation}} + \lambda \underbrace{\|\mathbf{c}\|_1}_{\text{sparsity}} \to \min_{\mathbf{c}} \quad (\lambda > 0).$$

Problem formulation

Objective function (Lasso problem, EEG: $K = 2$):

$$\mathbf{A} := [\mathbf{x}_1, \ldots, \mathbf{x}_n],$$
$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{c}\|_2^2}_{\text{good approximation}} + \lambda \underbrace{\|\mathbf{c}\|_1}_{\text{sparsity}} \to \min_{\mathbf{c}} \quad (\lambda > 0).$$

- Optimal $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$: non-zero in the relevant class.

Problem formulation

Objective function (Lasso problem, EEG: $K = 2$):

$$\mathbf{A} := [\mathbf{x}_1, \ldots, \mathbf{x}_n],$$
$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{c}\|_2^2}_{\text{good approximation}} + \lambda \underbrace{\|\mathbf{c}\|_1}_{\text{sparsity}} \to \min_{\mathbf{c}} \quad (\lambda > 0).$$

- Optimal $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$: non-zero in the relevant class.
- Decision rule with $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_K]$:

$$i^* = \arg\max_{i=1,\ldots,K} \|\mathbf{c}_i\|_2, \qquad \hat{y} = y_{i*}$$

Problem formulation

Objective function (Lasso problem, EEG: $K = 2$):

$$\mathbf{A} := [\mathbf{x}_1, \ldots, \mathbf{x}_n],$$
$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|_2^2}_{\text{good approximation}} + \lambda \underbrace{\left\|\mathbf{c}\right\|_1}_{\text{sparsity}} \to \min_{\mathbf{c}} \quad (\lambda > 0).$$

- Optimal $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$: non-zero in the relevant class.
- Decision rule with $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_K]$:

$$i^* = \arg\max_{i=1,\ldots,K} \left\|\mathbf{c}_i\right\|_2, \qquad \hat{y} = y_{i^*}, \text{ or}$$
$$i^* = \arg\min_{i=1,\ldots,K} \left\|\mathbf{x} - \mathbf{A}_i\mathbf{c}_i\right\|_2, \qquad \hat{y} = y_{i^*}.$$

Lasso solution: ISTA/FISTA [Beck and Teboulle, 2009]

$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\left\|\mathbf{x} - \mathbf{Ac}\right\|_2^2}_{=:f(\mathbf{c})} + \underbrace{\lambda\left\|\mathbf{c}\right\|_1}_{=:g(\mathbf{c})} \to \min_{\mathbf{c}}.$$

Lasso solution: ISTA/FISTA [Beck and Teboulle, 2009]

$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|_2^2}_{=:f(\mathbf{c})} + \underbrace{\lambda\left\|\mathbf{c}\right\|_1}_{=:g(\mathbf{c})} \to \min_{\mathbf{c}}.$$

- f: smooth convex,

$$\left\|\nabla f(\mathbf{a}) - \nabla f(\mathbf{b})\right\|_2 \leqslant \underbrace{L}_{>0}\left\|\mathbf{a} - \mathbf{b}\right\|_2 \quad \forall \mathbf{a}, \mathbf{b}.$$

Lasso solution: ISTA/FISTA [Beck and Teboulle, 2009]

$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|_2^2}_{=:f(\mathbf{c})} + \underbrace{\lambda\left\|\mathbf{c}\right\|_1}_{=:g(\mathbf{c})} \rightarrow \min_{\mathbf{c}}.$$

- f: smooth convex,

$$\left\|\nabla f(\mathbf{a}) - \nabla f(\mathbf{b})\right\|_2 \leqslant \underbrace{L}_{>0}\left\|\mathbf{a} - \mathbf{b}\right\|_2 \quad \forall \mathbf{a}, \mathbf{b}.$$

Example: $f(\mathbf{c}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|_2^2$, smallest $L = \lambda_{max}\left(\mathbf{A}^T\mathbf{A}\right)$.

Lasso solution: ISTA/FISTA [Beck and Teboulle, 2009]

$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\left\|\mathbf{x} - \mathbf{Ac}\right\|_2^2}_{=:f(\mathbf{c})} + \underbrace{\lambda\left\|\mathbf{c}\right\|_1}_{=:g(\mathbf{c})} \to \min_{\mathbf{c}}.$$

- f: smooth convex,

$$\left\|\nabla f(\mathbf{a}) - \nabla f(\mathbf{b})\right\|_2 \leqslant \underbrace{L}_{>0}\left\|\mathbf{a} - \mathbf{b}\right\|_2 \quad \forall \mathbf{a}, \mathbf{b}.$$

  Example: $f(\mathbf{c}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{Ac}\right\|_2^2$, smallest $L = \lambda_{max}\left(\mathbf{A}^T\mathbf{A}\right)$.
- g: continuous, convex, often nonsmooth.

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{c}_t = \mathbf{c}_{t-1} - \delta_t \nabla f(\mathbf{c}_{t-1}) \Leftrightarrow$$

$$\mathbf{c}_t = \arg\min_{\mathbf{c}} \left[f(\mathbf{c}_{t-1}) + \langle \mathbf{c} - \mathbf{c}_{t-1}, \nabla f(\mathbf{c}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{c} - \mathbf{c}_{t-1}\|_2^2 \right].$$

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{c}_t = \mathbf{c}_{t-1} - \delta_t \nabla f(\mathbf{c}_{t-1}) \Leftrightarrow$$

$$\mathbf{c}_t = \arg\min_{\mathbf{c}} \left[f(\mathbf{c}_{t-1}) + \langle \mathbf{c} - \mathbf{c}_{t-1}, \nabla f(\mathbf{c}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{c} - \mathbf{c}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$\widehat{(f+g)}_L(\mathbf{c}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{c} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{c} - \mathbf{y}\|_2^2 + g(\mathbf{c}). \Rightarrow$$

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{c}_t = \mathbf{c}_{t-1} - \delta_t \nabla f(\mathbf{c}_{t-1}) \Leftrightarrow$$

$$\mathbf{c}_t = \arg\min_{\mathbf{c}} \left[f(\mathbf{c}_{t-1}) + \langle \mathbf{c} - \mathbf{c}_{t-1}, \nabla f(\mathbf{c}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{c} - \mathbf{c}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$(\widehat{f+g})_L(\mathbf{c}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{c} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{c} - \mathbf{y}\|_2^2 + g(\mathbf{c}). \Rightarrow$$

$$p_L(\mathbf{y}) := \arg\min_{\mathbf{c}} (\widehat{f+g})_L(\mathbf{c}, \mathbf{y})$$

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{c}_t = \mathbf{c}_{t-1} - \delta_t \nabla f(\mathbf{c}_{t-1}) \Leftrightarrow$$

$$\mathbf{c}_t = \arg\min_{\mathbf{c}} \left[f(\mathbf{c}_{t-1}) + \langle \mathbf{c} - \mathbf{c}_{t-1}, \nabla f(\mathbf{c}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{c} - \mathbf{c}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$\left(\widehat{f+g}\right)_L(\mathbf{c}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{c} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{c} - \mathbf{y}\|_2^2 + g(\mathbf{c}). \Rightarrow$$

$$p_L(\mathbf{y}) := \arg\min_{\mathbf{c}} \left(\widehat{f+g}\right)_L(\mathbf{c}, \mathbf{y})$$

$$= \arg\min_{\mathbf{c}} \left[g(\mathbf{c}) + \frac{L}{2} \left\| \mathbf{c} - \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right) \right\|_2^2 \right]$$

ISTA '$=$' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{c}_t = \mathbf{c}_{t-1} - \delta_t \nabla f(\mathbf{c}_{t-1}) \Leftrightarrow$$

$$\mathbf{c}_t = \arg\min_{\mathbf{c}} \left[f(\mathbf{c}_{t-1}) + \langle \mathbf{c} - \mathbf{c}_{t-1}, \nabla f(\mathbf{c}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{c} - \mathbf{c}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$\widehat{(f+g)}_L(\mathbf{c}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{c} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{c} - \mathbf{y}\|_2^2 + g(\mathbf{c}). \Rightarrow$$

$$\begin{aligned} p_L(\mathbf{y}) &:= \arg\min_{\mathbf{c}} \widehat{(f+g)}_L(\mathbf{c}, \mathbf{y}) \\ &= \arg\min_{\mathbf{c}} \left[g(\mathbf{c}) + \frac{L}{2} \left\| \mathbf{c} - \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right) \right\|_2^2 \right] \\ &= prox_{\frac{1}{L} g} \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right). \end{aligned}$$

ISTA: L given

for all $t = 1 : T$ **do**
    $\mathbf{c}_t = p_L\left(\mathbf{c}_{t-1}\right) \leftarrow$ gradient descent $+$ 'projection'

ISTA: L given

for all $t = 1 : T$ **do**

$\quad \mathbf{c}_t = p_L\left(\mathbf{c}_{t-1}\right) \leftarrow$ gradient descent $+$ 'projection'

Notes:

- L: does not have to be known – backtracking.

ISTA: L given

1: **for all** $t = 1 : T$ **do**
2: $\quad \mathbf{c}_t = p_L\left(\mathbf{c}_{t-1}\right) \leftarrow$ gradient descent + 'projection'

Notes:

- L: does not have to be known – backtracking.
- Convergence: $O\left(\frac{1}{T}\right)$ in F-sense.

FISTA: L given

Init: $\mathbf{y}_1 = \mathbf{c}_0$, $\delta_1 = 1$
for all $t = 1 : T$ **do**
$\quad \mathbf{c}_t = p_L\left(\mathbf{y}_t\right)$
$\quad \delta_{t+1} = \frac{1+\sqrt{1+4\delta_t^2}}{2}$
$\quad \mathbf{y}_{t+1} = \mathbf{c}_t + \frac{\delta_t - 1}{\delta_{t+1}}(\mathbf{c}_t - \mathbf{c}_{t-1})$

FISTA: L given

Init: $\mathbf{y}_1 = \mathbf{c}_0$, $\delta_1 = 1$
for all $t = 1 : T$ **do**
  $\mathbf{c}_t = p_L\left(\mathbf{y}_t\right)$
  $\delta_{t+1} = \frac{1+\sqrt{1+4\delta_t^2}}{2}$
  $\mathbf{y}_{t+1} = \mathbf{c}_t + \frac{\delta_t - 1}{\delta_{t+1}}\left(\mathbf{c}_t - \mathbf{c}_{t-1}\right)$

Notes:

- L: not needed – backtracking.

FISTA: L given

Init: $\mathbf{y}_1 = \mathbf{c}_0$, $\delta_1 = 1$
for all $t = 1 : T$ **do**
$\quad \mathbf{c}_t = p_L\left(\mathbf{y}_t\right)$
$\quad \delta_{t+1} = \frac{1+\sqrt{1+4\delta_t^2}}{2}$
$\quad \mathbf{y}_{t+1} = \mathbf{c}_t + \frac{\delta_t - 1}{\delta_{t+1}}(\mathbf{c}_t - \mathbf{c}_{t-1})$

Notes:

- L: not needed – backtracking.
- Convergence: $O\left(\frac{1}{T^2}\right)$ in F-sense.

Local summary: ISTA/FISTA

We can solve

$$J(\mathbf{c}) = \underbrace{\frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{c}\|_2^2}_{=:f(\mathbf{c})} + \underbrace{\lambda\|\mathbf{c}\|_1}_{=:g(\mathbf{c})} \to \min_{\mathbf{c}}$$

type sparse coding problems quickly if

$$\nabla_f : \checkmark,$$

$$prox_g(\mathbf{v}) = \arg\min_{\mathbf{y}} \left[g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{v}\|_2^2 \right] \checkmark.$$

Prox: generalization of projection

$prox_g$ = Euclidean projection onto C if

$$g(\mathbf{y}) = I_C(\mathbf{y}) = \begin{cases} 0 & \mathbf{y} \in C, \\ \infty & \mathbf{y} \notin C. \end{cases}$$

Prox: properties

Our case: $g(\mathbf{y}) = \sum_m |y_m|$.

- Separable g: for $g(\mathbf{y}) = \sum_{m=1}^{M} g_m(\mathbf{y}_m)$

$$prox_g(\mathbf{y}_1, \ldots, \mathbf{y}_M) = \left[prox_{g_1}(\mathbf{y}_1); \ldots; prox_{g_M}(\mathbf{y}_M)\right].$$

Prox: properties

Our case: $g(\mathbf{y}) = \sum_m |y_m|$.

- Separable g: for $g(\mathbf{y}) = \sum_{m=1}^M g_m(\mathbf{y}_m)$

$$prox_g(\mathbf{y}_1, \ldots, \mathbf{y}_M) = \left[prox_{g_1}(\mathbf{y}_1); \ldots; prox_{g_M}(\mathbf{y}_M)\right].$$

- For $g(y) = |y|$

$$prox_{\kappa g}(y) = \begin{cases} y - \kappa & y \geqslant \kappa, \\ 0 & |y| \leqslant \kappa \\ y + \kappa & y \leqslant -\kappa. \end{cases}$$

Convex structured sparse coding ($\lambda > 0$):

$$J(\mathbf{c}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|_2^2 + \lambda \left\| \left(\|\mathbf{c}_G\|_2\right)_{G\in\mathcal{G}} \right\|_1 \to \min_{\mathbf{c}},$$

$\mathcal{G}$: group structure on $\{1,\ldots,d_c\}$, $\{1,\ldots,d_c\} = \cup_{G\in\mathcal{G}}$.

Lasso: extensions

Convex structured sparse coding ($\lambda > 0$):

$$J(\mathbf{c}) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{c}\|_2^2 + \lambda \left\| \left(\|\mathbf{c}_G\|_2 \right)_{G \in \mathcal{G}} \right\|_1 \rightarrow \min_{\mathbf{c}},$$

$\mathcal{G}$: group structure on $\{1, \ldots, d_c\}$, $\{1, \ldots, d_c\} = \cup_{G \in \mathcal{G}}$.

Traditional group Lasso:

- $\mathcal{G}$ = partition [EEG: $G_i = i^{th}$ emotion].

Lasso: extensions

Convex structured sparse coding ($\lambda > 0$):

$$J(\mathbf{c}) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{c}\|_2^2 + \lambda \left\| \left(\|\mathbf{c}_G\|_2\right)_{G \in \mathcal{G}} \right\|_1 \rightarrow \min_{\mathbf{c}},$$

$\mathcal{G}$: group structure on $\{1, \ldots, d_c\}$, $\{1, \ldots, d_c\} = \cup_{G \in \mathcal{G}}$.

Traditional group Lasso:

- $\mathcal{G}$ = partition [EEG: $G_i = i^{th}$ emotion].
- prox: block soft-thresholding.

Greedy methods: UoS models

- Greedy techniques: faster, weaker guarantees.

Greedy methods: UoS models

- Greedy techniques: faster, weaker guarantees.
- Union-of-subspace (UoS) models:

$$\mathbf{c} \in \mathcal{U} = \cup_{m=1}^{M} \mathcal{U}_i, \quad \mathcal{U}_i \subseteq \mathbb{R}^{d_c}: \text{ subspace.}$$

Greedy methods: UoS models

- Greedy techniques: faster, weaker guarantees.
- Union-of-subspace (UoS) models:

$$\mathbf{c} \in \mathcal{U} = \cup_{m=1}^{M} \mathcal{U}_i, \quad \mathcal{U}_i \subseteq \mathbb{R}^{d_c}\text{: subspace.}$$

- prox $\rightarrow \mathcal{P}_{\mathcal{U}}$: $\mathcal{P}_{\mathcal{U}}(\mathbf{u}) = \arg\min_{\mathbf{v} \in \mathcal{U}} \|\mathbf{u} - \mathbf{v}\|_2$.

Greedy methods: UoS models

- Greedy techniques: faster, weaker guarantees.
- Union-of-subspace (UoS) models:

$$\mathbf{c} \in \mathcal{U} = \cup_{m=1}^{M} \mathcal{U}_i, \quad \mathcal{U}_i \subseteq \mathbb{R}^{d_c}\text{: subspace.}$$

- prox $\rightarrow \mathcal{P}_{\mathcal{U}}$: $\mathcal{P}_{\mathcal{U}}(\mathbf{u}) = \arg\min_{\mathbf{v}\in\mathcal{U}} \|\mathbf{u}-\mathbf{v}\|_2$.
- s-block-sparse model: s blocks with largest energy

$$\sum_{i\in G_j} |c_i|^2 \leftarrow \text{Hard block-thresholding.}$$

Greedy methods: UoS models

- Greedy techniques: faster, weaker guarantees.
- Union-of-subspace (UoS) models:

$$\mathbf{c} \in \mathcal{U} = \cup_{m=1}^{M} \mathcal{U}_i, \quad \mathcal{U}_i \subseteq \mathbb{R}^{d_c}: \text{ subspace.}$$

- prox $\rightarrow \mathcal{P}_{\mathcal{U}}$: $\mathcal{P}_{\mathcal{U}}(\mathbf{u}) = \arg\min_{\mathbf{v}\in\mathcal{U}} \|\mathbf{u} - \mathbf{v}\|_2$.
- s-block-sparse model: s blocks with largest energy

$$\sum_{i \in G_j} |c_i|^2 \leftarrow \text{Hard block-thresholding.}$$

- Methods: IHT, CoSaMP, SP [Blumensath and Davies, 2009a, Blumensath and Davies, 2009b, Baraniuk et al., 2010].

Structured sparse coding on time-series

$$J(\mathbf{c}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|^2 + \lambda\left\|\left(\left\|\mathbf{c}_G\right\|_2\right)_{G\in\mathcal{G}}\right\|_1 \to \min_{\mathbf{c}}.$$

Idea [Jeni et al., 2014]:

- Columns of **A** are time-series of possibly different length.

Structured sparse coding on time-series

$$J(\mathbf{c}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|^2 + \lambda\left\|\left(\left\|\mathbf{c}_G\right\|_2\right)_{G\in\mathcal{G}}\right\|_1 \to \min_{\mathbf{c}}.$$

Idea [Jeni et al., 2014]:

- Columns of **A** are time-series of possibly different length.
- Example:
  - $\mathbf{x}_i$: evolution of facial muscle activities.

Structured sparse coding on time-series

$$J(\mathbf{c}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|^2 + \lambda\left\|\left(\left\|\mathbf{c}_G\right\|_2\right)_{G\in\mathcal{G}}\right\|_1 \to \min_{\mathbf{c}}.$$

Idea [Jeni et al., 2014]:

- Columns of $\mathbf{A}$ are time-series of possibly different length.
- Example:
  - $\mathbf{x}_i$: evolution of facial muscle activities.
  - y_i: emotion of the user.

Structured sparse coding on time-series

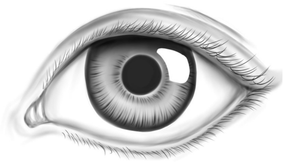
$$J(\mathbf{c}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{A}\mathbf{c}\right\|^2 + \lambda \left\| \left(\left\| \mathbf{c}_G \right\|_2 \right)_{G \in \mathcal{G}} \right\|_1 \to \min_{\mathbf{c}}.$$

Idea [Jeni et al., 2014]:

- Columns of **A** are time-series of possibly different length.
- Example:
  - $\mathbf{x}_i$: evolution of facial muscle activities.
  - y_i: emotion of the user.
  - $\mathbf{a}_i = \varphi(\mathbf{x}_i)$, $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle$.

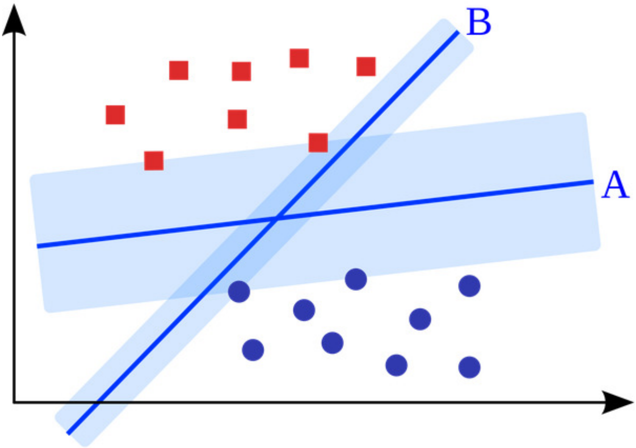
+Info on (structured) sparse coding, compressed sensing

- Papers, blog:
  - https://sites.google.com/site/igorcarron2/cs,
  - http://nuit-blanche.blogspot.com/search/label/CS.
- Code:
  - SLEP: http://www.yelab.net/software/SLEP/
  - SPAMS: http://spams-devel.gforge.inria.fr/

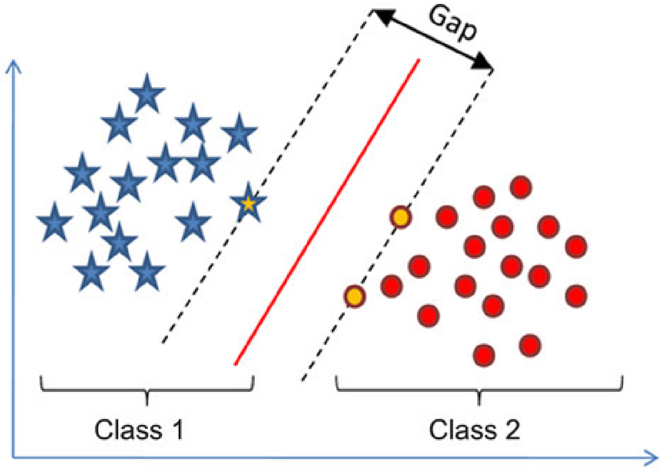


Classification: SVM

Which separating line is the 'best'?



SVM answer: the one with the largest margin.



SVM formulation: hard classification

- Hyperplane: $f_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.
  - $\mathbf{w}$: normal vector, b: offset.

SVM formulation: hard classification

- Hyperplane: $f_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.
  - $\mathbf{w}$: normal vector, b: offset.
- Goal:

$$\max_{\mathbf{w},b} \underbrace{\frac{2}{\|\mathbf{w}\|_2}}_{\text{margin}} \Leftrightarrow \min \|\mathbf{w}\|_2^2, \text{ s.t. } \underbrace{\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geqslant 1 & \text{if } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leqslant -1 & \text{otherwise.} \end{cases}}_{\text{correction classification}}$$

SVM formulation: hard classification

- Hyperplane: $f_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.
  - $\mathbf{w}$: normal vector, b: offset.
- Goal:

$$\max_{\mathbf{w},b} \underbrace{\frac{2}{\|\mathbf{w}\|_2}}_{\text{margin}} \Leftrightarrow \min \|\mathbf{w}\|_2^2, \text{ s.t. } \underbrace{\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geqslant 1 & \text{if } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leqslant -1 & \text{otherwise.} \end{cases}}_{\text{correction classification}}$$

- Shortly,

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2^2, \text{ s.t. } y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) \geqslant 1, \forall i.$$

SVM formulation: hard classification

- Hyperplane: $f_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.
  - $\mathbf{w}$: normal vector, b: offset.
- Goal:

$$\max_{\mathbf{w},b} \underbrace{\frac{2}{\|\mathbf{w}\|_2}}_{\text{margin}} \Leftrightarrow \min \|\mathbf{w}\|_2^2, \text{ s.t. } \underbrace{\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geqslant 1 & \text{if } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leqslant -1 & \text{otherwise.} \end{cases}}_{\text{correction classification}}$$

- Shortly,

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2^2, \text{ s.t. } y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) \geqslant 1, \forall i.$$

- Decision: $\hat{y} = sign \left(\langle \mathbf{w}, \mathbf{x} \rangle + b \right)$.

SVM formulation: soft classification

- Hard classification objective:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2^2 \text{ s.t. } y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geqslant 1, \forall i.$$

There might not be solution! (non-linearly separable case)

SVM formulation: soft classification

- Hard classification objective:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2^2 \;\text{ s.t. }\; y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geqslant 1, \forall i.$$

There might not be solution! (non-linearly separable case)

- Soft classification objective:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \;\text{ s.t. }\; y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geqslant 1 - \xi_i, \forall i.$$

Linear penalty on misclassification.

SVM formulation: soft classification

Soft classification objective:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i, \text{ s.t. } y_i\left(\langle\mathbf{w},\mathbf{x}_i\rangle + b\right) \geqslant 1 - \xi_i \quad (\forall i).$$

Using Lagrangian multipliers

SVM formulation: soft classification

Soft classification objective:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i, \text{ s.t. } y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geqslant 1 - \xi_i \quad (\forall i).$$

Using Lagrangian multipliers: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$ and

$$\max_{\boldsymbol{\alpha}} \underbrace{\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j}_{\text{quadratic in } \boldsymbol{\alpha}}, \text{ s.t. } \underbrace{0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^{n}\alpha_i y_i = 0}_{\text{linear in } \boldsymbol{\alpha}}.$$

SVM formulation: soft classification

Soft classification objective:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i, \text{ s.t. } y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geqslant 1 - \xi_i \quad (\forall i).$$

Using Lagrangian multipliers: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$ and

$$\max_{\boldsymbol{\alpha}} \underbrace{\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j}_{\text{quadratic in } \boldsymbol{\alpha}}, \text{ s.t. } \underbrace{0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^{n}\alpha_i y_i = 0}_{\text{linear in } \boldsymbol{\alpha}}.$$

- $b \Leftarrow y_i(\mathbf{w}_i^T\mathbf{x}_i + b) = 1 \Leftarrow \alpha_i \in (0, C)$.

SVM formulation: soft classification

Soft classification objective:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \xi_i, \text{ s.t. } y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geqslant 1 - \xi_i \quad (\forall i).$$

Using Lagrangian multipliers: $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ and

$$\max_{\boldsymbol{\alpha}} \underbrace{\sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j}_{\text{quadratic in } \boldsymbol{\alpha}}, \text{ s.t. } \underbrace{0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^n \alpha_i y_i = 0}_{\text{linear in } \boldsymbol{\alpha}}.$$

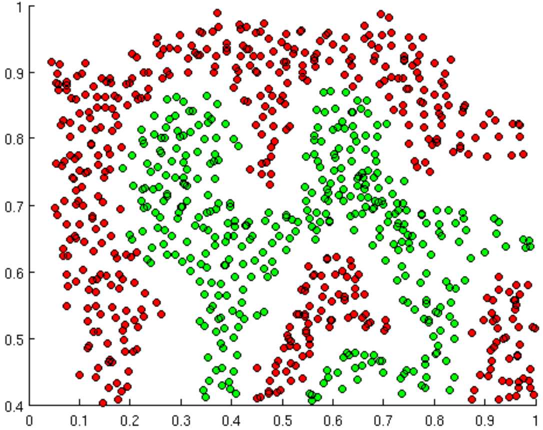
- $b \Leftarrow y_i(\mathbf{w}_i^T\mathbf{x}_i + b) = 1 \Leftarrow \alpha_i \in (0, C)$.
- QP: solvers are available.

If linear separability does not hold

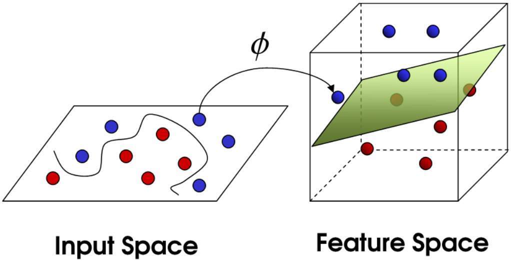
- Until this point:
  - (almost) linearly separable case.

If linear separability does not hold

- Until this point:
  - (almost) linearly separable case.
- Now:



If linear separability does not hold: kernel trick



Nonlinear SVM

- Linear SVM (dual):

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \text{ s.t. } 0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^{n} \alpha_i y_i = 0, \forall i.$$

Nonlinear SVM

- Linear SVM (dual):

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \text{ s.t. } 0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^{n} \alpha_i y_i = 0, \forall i.$$

- Nonlinear SVM (dual):

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \text{ s.t. } 0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^{n} \alpha_i y_i = 0, \forall i.$$

Nonlinear SVM

- Linear SVM (dual):

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \text{ s.t. } 0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^{n} \alpha_i y_i = 0, \forall i.$$

- Nonlinear SVM (dual):

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \text{ s.t. } 0 \leqslant \alpha_i \leqslant C, \sum_{i=1}^{n} \alpha_i y_i = 0, \forall i.$$

- Nonlinear SVM (primal):

$$\min_{f \in \mathcal{H}_k, \boldsymbol{\xi}} \frac{1}{2} \|f\|_{\mathcal{H}_k}^2 + C \sum_{i=1}^{n} \xi_i, \text{ s.t. } y_i f(\mathbf{x}_i) \geqslant 1 - \xi_i, \forall i.$$

Kernel: similarity between features

- Given: x and $x' \in \mathcal{X}$ objects (images, texts, ...).

Kernel: similarity between features

- Given: x and $x' \in \mathcal{X}$ objects (images, texts, ...).
- Question: how similar they are?

Kernel: similarity between features

- Given: x and $x' \in \mathcal{X}$ objects (images, texts, ...).
- Question: how similar they are?
- Define features of the objects:

$$\mathcal{H} \ni \varphi(x) : \text{features of } x,$$
$$\mathcal{H} \ni \varphi(x') : \text{features of } x'.$$

Kernel: similarity between features

- Given: x and $x' \in \mathcal{X}$ objects (images, texts, ...).
- Question: how similar they are?
- Define features of the objects:

$$\mathcal{H} \ni \varphi(x) : \text{features of } x,$$
$$\mathcal{H} \ni \varphi(x') : \text{features of } x'.$$

- Kernel: inner product of these features

$$k(x, x') := \left\langle \varphi(x), \varphi(x') \right\rangle_{\mathcal{H}}.$$

Kernel: similarity between features

- Given: x and $x' \in \mathcal{X}$ objects (images, texts, ...).
- Question: how similar they are?
- Define features of the objects:

$$\mathcal{H} \ni \varphi(x) : \text{features of } x,$$
$$\mathcal{H} \ni \varphi(x') : \text{features of } x'.$$

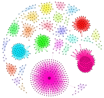
- Kernel: inner product of these features

$$k(x, x') := \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}.$$

defines a $\mathcal{H}_k = \{\mathcal{X} \to \mathbb{R} : \dots\}$ function space.

Kernel examples



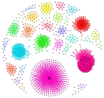


- $\mathcal{X} = \mathbb{R}^d$:

$$k_p(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + \gamma)^p, \qquad k_G(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|_2^2},$$

$$k_e(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|_2}, \qquad k_C(\mathbf{x}, \mathbf{y}) = 1 + \frac{1}{\gamma \|\mathbf{x}-\mathbf{y}\|_2^2}.$$

Kernel examples



- $\mathcal{X} = \mathbb{R}^d$:

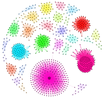
$$k_p(\mathbf{x},\mathbf{y}) = (\langle \mathbf{x},\mathbf{y}\rangle + \gamma)^p, \qquad k_G(\mathbf{x},\mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|_2^2},$$

$$k_e(\mathbf{x},\mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|_2}, \qquad k_C(\mathbf{x},\mathbf{y}) = 1 + \frac{1}{\gamma\|\mathbf{x}-\mathbf{y}\|_2^2}.$$

- $\mathcal{X}$ = texts, strings:
  - bag-of-word kernel,
  - r-spectrum kernel: # of common $\leq r$-substrings.

Kernel examples





- $\mathcal{X} = \mathbb{R}^d$:

$$k_p(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + \gamma)^p, \qquad k_G(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|_2^2},$$

$$k_e(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|_2}, \qquad k_C(\mathbf{x}, \mathbf{y}) = 1 + \frac{1}{\gamma \|\mathbf{x}-\mathbf{y}\|_2^2}.$$

- $\mathcal{X}$ = texts, strings:
  - bag-of-word kernel,
  - r-spectrum kernel: $\#$ of common $\leqslant r$-substrings.
- $\mathcal{X}$ = time-series: dynamic time-warping.

RKHS definition(s)

Given: $\mathcal{X}$ set.

- Kernel: $k(a, b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.

RKHS definition(s)

Given: $\mathcal{X}$ set.

- Kernel: $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.
- Reproducing kernel of an $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ Hilbert space,
  1. $\underbrace{k(\cdot, b)}_{=\varphi(b)} \in \mathcal{H}$,

RKHS definition(s)

Given: $\mathcal{X}$ set.

- Kernel: $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.
- Reproducing kernel of an $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ Hilbert space,
  1. $\underbrace{k(\cdot,b)}_{=\varphi(b)} \in \mathcal{H}$,
  2. $\langle f, k(\cdot,b) \rangle_{\mathcal{H}} = f(b)$. $\Rightarrow k(a,b) = \langle \underbrace{k(\cdot,a)}_{=\varphi(a)}, \underbrace{k(\cdot,b)}_{=\varphi(b)} \rangle_{\mathcal{H}}$.

RKHS definition(s)

Given: $\mathcal{X}$ set.

- Kernel: $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.
- Reproducing kernel of an $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ Hilbert space,
  1. $\underbrace{k(\cdot,b)}_{=\varphi(b)} \in \mathcal{H}$,
  2. $\langle f, k(\cdot,b) \rangle_{\mathcal{H}} = f(b)$. $\Rightarrow k(a,b) = \langle \underbrace{k(\cdot,a)}_{=\varphi(a)}, \underbrace{k(\cdot,b)}_{=\varphi(b)} \rangle_{\mathcal{H}}$.
- $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ sym. is pd. if $\mathbf{G} = [k(x_i,x_j)]_{i,j=1}^n \succeq 0$.

Looking back to nonlinear SVM

$$\min_{f \in \mathcal{H}_k, \boldsymbol{\xi}} \frac{1}{2} \|f\|_{\mathcal{H}_k}^2 + C \sum_{i=1}^{n} \xi_i, \text{ s.t. } y_i f(x_i) \geqslant 1 - \xi_i.$$

- Linear classification on $\varphi(x) = k(\cdot, x)$ $[\Leftarrow f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}_k}]$.

Looking back to nonlinear SVM

$$\min_{f\in\mathcal{H}_k,\boldsymbol{\xi}} \frac{1}{2}\|f\|_{\mathcal{H}_k}^2 + C\sum_{i=1}^n \xi_i, \text{ s.t. } y_i f(x_i) \geqslant 1-\xi_i.$$

- Linear classification on $\varphi(x) = k(\cdot, x)$ [$\Leftarrow f(x) = \langle f, k(\cdot, x)\rangle_{\mathcal{H}_k}$].
- Access to $k(x, x')$ is enough.

Looking back to nonlinear SVM

$$\min_{f\in\mathcal{H}_k,\boldsymbol{\xi}} \frac{1}{2}\|f\|_{\mathcal{H}_k}^2 + C\sum_{i=1}^n \xi_i, \text{ s.t. } y_i f(x_i) \geqslant 1-\xi_i.$$

- Linear classification on $\varphi(x) = k(\cdot, x)$ $[\Leftarrow f(x) = \langle f, k(\cdot,x)\rangle_{\mathcal{H}_k}]$.
- Access to $k(x,x')$ is enough.
- Representation theorem $\Rightarrow$ finiteD problem:

$$f^* = \sum_{i=1}^n \alpha_i k(\cdot, x_i).$$

Representer theorem

- Given: $\{(x_i, y_i)\}_{i=1}^n$, say classification/regression.
- Goal:

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|_{\mathcal{H}_k}^2\right) \to \min_{\mathcal{H}_k},$$

r : monotically increasing.

Representer theorem

- Given: $\{(x_i, y_i)\}_{i=1}^n$, say classification/regression.
- Goal:

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|^2_{\mathcal{H}_k}\right) \to \min_{\mathcal{H}_k},$$

r : monotically increasing.

- Example:

$$V(\ldots) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}_{y_i f(x_i) < 0} \text{ (classification)},$$

$$V(\ldots) = \frac{1}{n}\sum_{i=1}^n [f(x_i) - y_i]^2 \text{ (regression)}.$$

...then

- $\exists$ solution in the form:

$$f^* = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i).$$

- r: strictly increasing $\Rightarrow$ $\forall$ solution is of this form.
- Example: $r(z) = \lambda z$, $\lambda > 0$.

Representer theorem – proof

Objective

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|^2_{\mathcal{H}_k}\right) \to \min_{\mathcal{H}_k}.$$

Decompose & Pythagorean theorem:

$$\begin{aligned} S &= span\left(k(\cdot, x_i), i = 1, \ldots, n\right), \\ f &= f_S + f_\perp, \\ \|f\|^2_{\mathcal{H}_k} &= \|f_S\|^2_{\mathcal{H}_k} + \underbrace{\|f_\perp\|^2_{\mathcal{H}_k}}_{\geqslant 0} \geqslant \|f_S\|^2_{\mathcal{H}_k}. \end{aligned}$$

Representer theorem – proof

Objective

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|_{\mathcal{H}_k}^2\right) \to \min_{\mathcal{H}_k}.$$

Decompose & Pythagorean theorem:

$$\begin{aligned} S &= span\left(k(\cdot, x_i), i = 1, \ldots, n\right), \\ f &= f_S + f_\perp, \\ \|f\|_{\mathcal{H}_k}^2 &= \|f_S\|_{\mathcal{H}_k}^2 + \underbrace{\|f_\perp\|_{\mathcal{H}_k}^2}_{\geqslant 0} \geqslant \|f_S\|_{\mathcal{H}_k}^2. \end{aligned}$$

In J

- 1st term: depends on f_S only.

Representer theorem – proof

Objective

$$J(f) = V(x_1, y_1, f(x_1), \dots, x_n, y_n, f(x_n)) + r\left(\|f\|^2_{\mathcal{H}_k}\right) \to \min_{\mathcal{H}_k}.$$

Decompose & Pythagorean theorem:

$$\begin{aligned} S &= span\left(k(\cdot, x_i), i = 1, \dots, n\right), \\ f &= f_S + f_\perp, \\ \|f\|^2_{\mathcal{H}_k} &= \|f_S\|^2_{\mathcal{H}_k} + \underbrace{\|f_\perp\|^2_{\mathcal{H}_k}}_{\geqslant 0} \geqslant \|f_S\|^2_{\mathcal{H}_k}. \end{aligned}$$

In J

- 1st term: depends on f_S only.
- 2nd term: can only decrease by neglecting $f_\perp$ ($r \nearrow$).

M-fold cross-validation [$\theta := (C, \sigma)$]:

1. Split data:
    - training set (X_{tr}, Y_{tr}): $X_{val,i}, Y_{val,i}$, $i = 1, \ldots, M$.
    - test set: X_{te}, Y_{te}.

M-fold cross-validation [$\theta := (C, \sigma)$]:

1. Split data:
   - training set (X_{tr}, Y_{tr}): $X_{val,i}, Y_{val,i}$, $i = 1, \ldots, M$.
   - test set: X_{te}, Y_{te}.
2. For fixed θ: evaluate the average error while
   - trained on: $X_{tr} \backslash X_{val,i}$, $Y_{tr} \backslash Y_{val,i}$,
   - tested on: $X_{val,i}$, $Y_{val,i}$.

Parameter selection $\rightarrow$ cost: C, kernel parameter: σ

M-fold cross-validation $[\theta := (C, \sigma)]$:

1. Split data:
   - training set (X_{tr}, Y_{tr}): $X_{val,i}, Y_{val,i}$, $i = 1, \ldots, M$.
   - test set: X_{te}, Y_{te}.
2. For fixed θ: evaluate the average error while
   - trained on: $X_{tr} \backslash X_{val,i}$, $Y_{tr} \backslash Y_{val,i}$,
   - tested on: $X_{val,i}$, $Y_{val,i}$.
3. $\theta^* :=$ minimizer of CV error.

Parameter selection $\rightarrow$ cost: C, kernel parameter: σ

M-fold cross-validation $[\theta := (C, \sigma)]$:

1. Split data:
   - training set (X_{tr}, Y_{tr}): $X_{val,i}, Y_{val,i}$, $i = 1, \ldots, M$.
   - test set: X_{te}, Y_{te}.
2. For fixed θ: evaluate the average error while
   - trained on: $X_{tr} \backslash X_{val,i}$, $Y_{tr} \backslash Y_{val,i}$,
   - tested on: $X_{val,i}$, $Y_{val,i}$.
3. $\theta^* :=$ minimizer of CV error.
4. Report: performance of θ^* on X_{te}, Y_{te}.

Classification: summary

- kNN: simple.

Classification: summary

- kNN: simple.
- Sparse, structured sparse coding:
  - Lasso, group Lasso, UoS models.
  - soft (ISTA, FISTA) / hard thresholding (greedy methods).
  - kernel: ✓

Classification: summary

- kNN: simple.
- Sparse, structured sparse coding:
  - Lasso, group Lasso, UoS models.
  - soft (ISTA, FISTA) / hard thresholding (greedy methods).
  - kernel: ✓
- SVM:
  - linearly separable,
  - nonlinear (kernel, flexible).

Classification: summary

- kNN: simple.
- Sparse, structured sparse coding:
  - Lasso, group Lasso, UoS models.
  - soft (ISTA, FISTA) / hard thresholding (greedy methods).
  - kernel: ✓
- SVM:
  - linearly separable,
  - nonlinear (kernel, flexible).
- Parameter selection: cross-validation.

Summary

- Manifold learning:
  - PCA.
  - MDS, ISOMAP, Sammon mapping; MVU, LLE.

Summary

- Manifold learning:
  - PCA.
  - MDS, ISOMAP, Sammon mapping; MVU, LLE.
- Classification:
  - kNN methods.
  - (Structured) sparse coding.
  - SVM.

Thank you for the attention!

Baraniuk, R. G., Cevher, V., Duarte, M. F., and Hegde, C. (2010).
Model-based compressive sensing.
IEEE Transactions on Information Theory, 56(4):1982 – 2001.

Beck, A. and Teboulle, M. (2009).
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM Journal on Imaging Sciences, 2(1):183–202.

Blumensath, T. and Davies, M. E. (2009a).
Iterative hard thresholding for compressed sensing.
Applied and Computational Harmonic Analysis, 27:265–274.

Blumensath, T. and Davies, M. E. (2009b).
Sampling theorems for signals from the union of finite-dimensional linear subspaces.
IEEE Transactions on Information Theory, 55(4):1872–1882.

Jeni, L., Lörincz, A., Szabó, Z., Cohn, J. F., and Kanade, T. (2014).
Spatio-temporal event classification using time-series kernel based structured sparsity.
In *European Conference on Computer Vision (ECCV)*, pages 135–150.

Roweis, S. and Saul, L. (2000).
Nonlinear dimensionality reduction by locally linear embedding.
Science, 290(5500):2323–2326.

Tenenbaum, J., de Silva, V., and Langford, J. (2000).
A global geometric framework for nonlinear dimensionality reduction.
Science, 290:2319–2323.

Torgerson, W. (1952).
Multidimensional scaling: Theory and methods.
Psychometrika, 17:401–419.

van der Maaten, L. and Hinton, G. (2008).
Visualizing high-dimensional data using t-sne.
Journal of Machine Learning Research, 9:2579–2605.

van der Maaten, L., Postma, E., and van den Herik, H. (2009).
Dimensionality reduction: A comparative review.
Technical report, Tilburg University.

Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., and Ma, Y. (2009).
Toward a practical face recognition: Robust pose and illumination via sparse representation.
In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 597–604.

Weinberger, K. Q., Sha, F., and Saul, L. K. (2004).
Learning a kernel matrix for nonlinear dimensionality reduction.

In *International Conference on Machine Learning (ICML)*, pages 106–113.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009).
Robust face recognition via sparse representation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2):210–227.