Lecture 19 – Matrix rank minimization

Semidefinite programming

A reminder

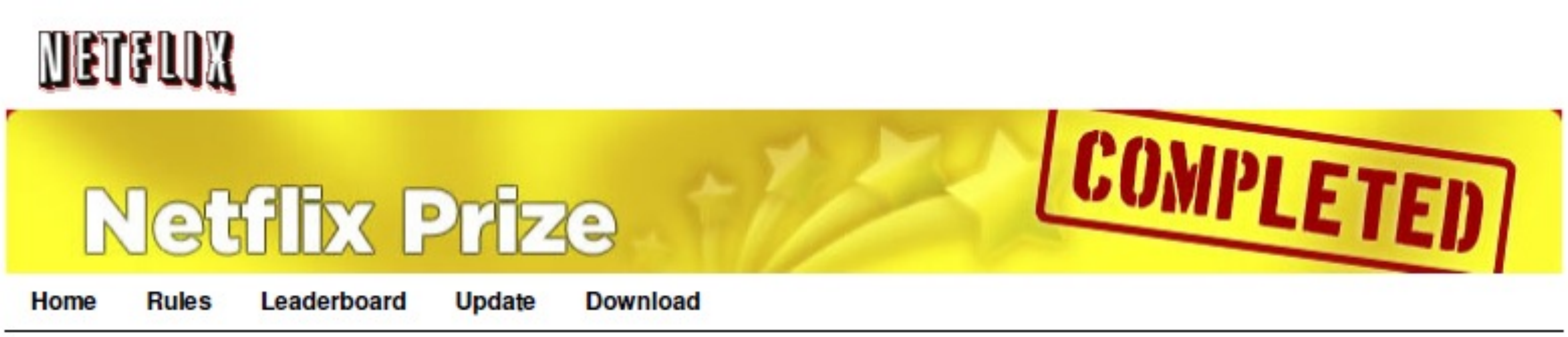
Primal Semidefinite Programming Problem

$$\begin{array}{ll} \min & \mathrm{trace(CX)}, \\ \text{s.t.} & \mathrm{trace(A_iX) = b_i,\ \ i = 1, \ldots, m} \\ & X \in \mathbf{S}^n \ \ X \succeq 0 \\ & C, A_i \in \mathbf{S}^n, b \in \mathbf{R}^m. \end{array}$$

SDP cone $K = \{x \in \mathbf{S}^n :\ X \succeq 0\}$ - self dual.

Dual Semidefinite Programming Problem

$$\begin{array}{ll} \max & b^T y, \\ \text{s.t.} & \sum_{i=1}^{m} y_i A_i + S = C \\ & S \succeq 0. \end{array}$$



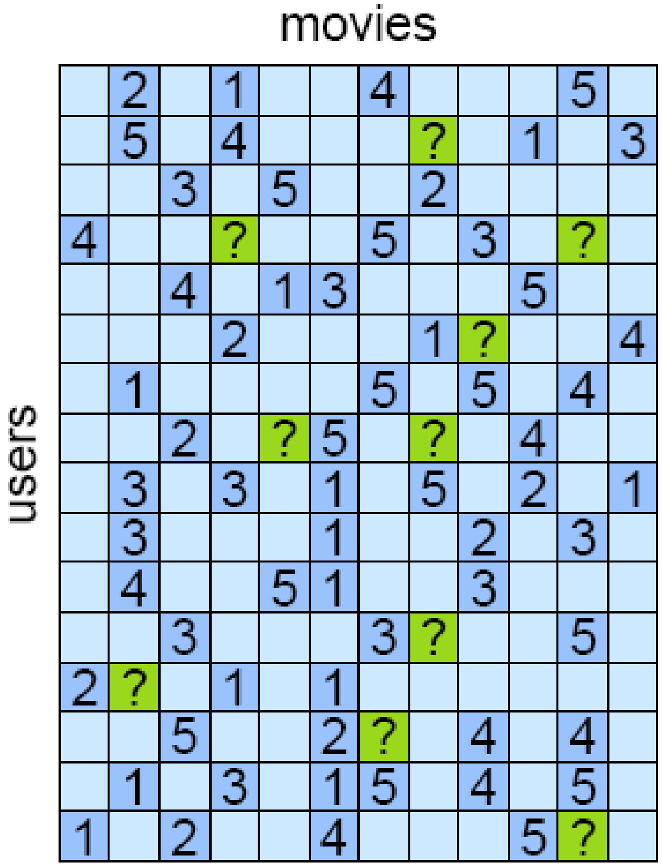
- Some users rate some movies they watched (or didn’t!)
- Predict the rating (1..5) for each user/movie pair.
- Use this prediction to recommend users the movies that they would like

Matrix completion problem, collaborative filtering

Collaborative filtering: famous Netflix challenge

Will user i like movie j?

Complete the matrix based on partially filled information.

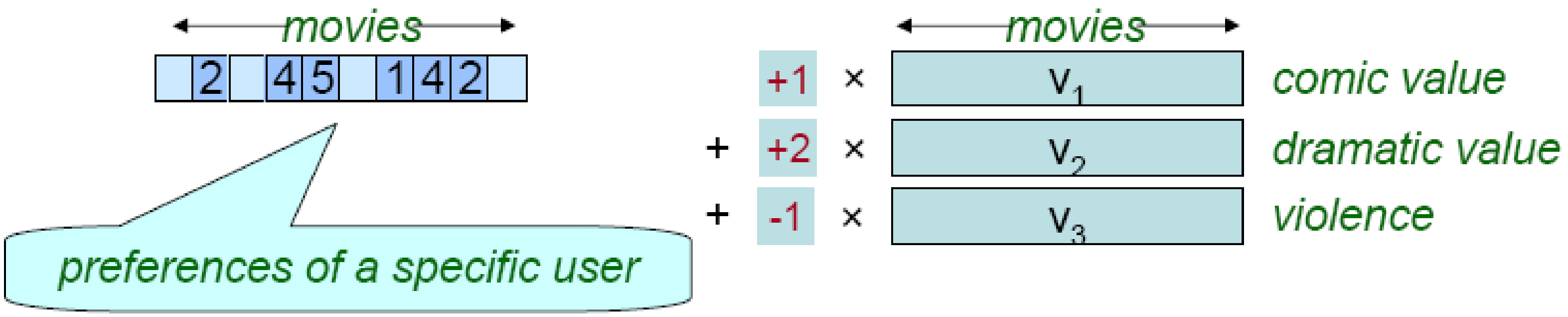


movies

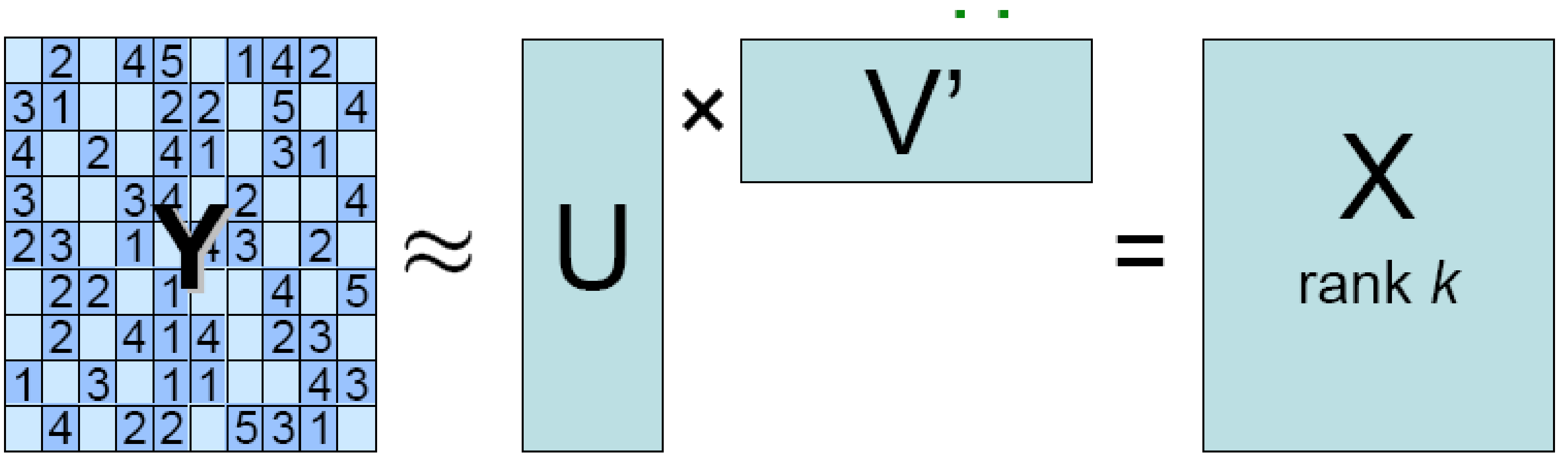
users

	2		1			4				5	
	5		4				?		1		3
		3		5			2				
4			?			5		3		?	
		4		1	3				5		
			2				1	?			4
	1					5		5		4	
		2		?	5		?		4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3	?			5	
2	?		1		1						
		5			2	?		4		4	
	1		3		1	5		4		5	
1		2			4				5	?	

Linear factor model



	2		4	5		1	4	2	



	2		4	5		1	4	2	
3	1			2	2		5		4
4		2		4	1		3	1	
3			3	4		2			4
2	3		1		4	3		2	
	2	2		1			4		5
	2		4	1	4		2	3	
1		3		1	1			4	3
	4		2	2		5	3	1	

Convex relaxation via nuclear norm

- Given the values for a subset of entries, find the matrix with these entries and the smallest (or given) rank.

$$\begin{array}{ll} \min_{X \in \mathbf{R}^{m \times n}} & \mathbf{rank}(X) \\ \text{s.t.} & X_{ij} = M_{ij}, \ (i,j) \in I \end{array}$$

- NP-hard problem.

$\mathbf{rank}(X) = \|\sigma(X)\|_0,$
where $\sigma(X)$ is the vector of the singular values.

$\| \cdot \|_0, \Rightarrow \| \cdot \|_1$ - the tightest convex relaxation.

Convex relaxation: $\|\sigma(X)\|_1 = \sum_{i=1}^{n} \sigma_i(X)$

Under suitable randomness hypothesis

ACMRM

- Recht, Fazel and Parrilo, 2007:
  For fixed $0 < \delta < 1$, when $p = O((m+n)r\log(mn))$, with high probability, $\mathcal{A}$ satisfies the Restricted Isometry Property (RIP):

$$(1-\delta_r(\mathcal{A}))\|X\|_F \leq \|\mathcal{A}X\|_2 \leq (1+\delta_r(\mathcal{A}))\|X\|_F,$$

  with $\delta_r(\mathcal{A}) \leq \delta$ for all matrices X of rank r.

Matrix Completion

- Candès and Recht, 2008: $O(n^{1.2}r\log n)$
- Candès and Tao, 2009: $O(nr\text{poly}\log n)$

Convex relaxation via nuclear norm

- Given the values for a subset of entries, find the matrix with these entries and the smallest "nuclear norm".

$$\begin{array}{ll}\min_{X \in \mathbf{R}^{m\times n}} & \|X\|_* \\ \text{s.t.} & X_{ij} = M_{ij},\ (i,j) \in I\end{array}$$

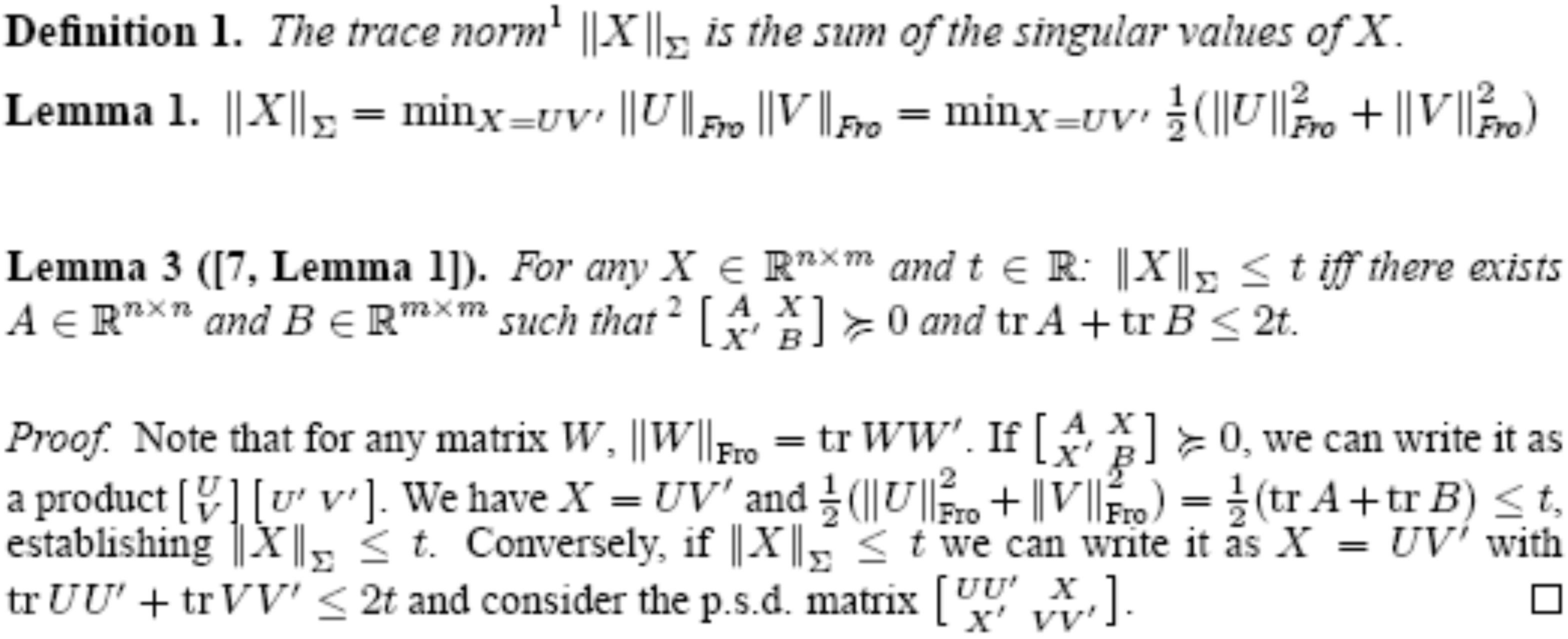
- Convex problem

$$\|X\|_* = \|\sigma(X)\|_1 = \sum_{i=1}^{n} \sigma_i(X)$$

- Convex Cone

$$\|X\|_* \le t$$

Trace norm properties



Definition 1. *The trace norm*[1] $\|X\|_\Sigma$ *is the sum of the singular values of* X.

Lemma 1. $\|X\|_\Sigma = \min_{X=UV'} \|U\|_{Fro} \|V\|_{Fro} = \min_{X=UV'} \frac{1}{2}(\|U\|^2_{Fro} + \|V\|^2_{Fro})$

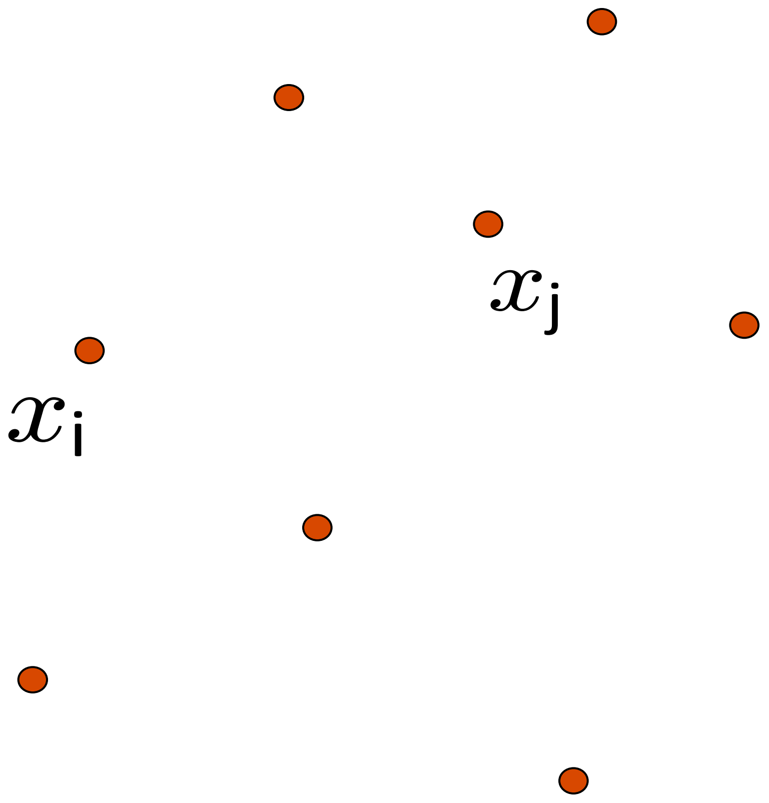
Lemma 3 ([7, Lemma 1]). *For any* $X \in \mathbb{R}^{n\times m}$ *and* $t \in \mathbb{R}$: $\|X\|_\Sigma \leq t$ *iff there exists* $A \in \mathbb{R}^{n\times n}$ *and* $B \in \mathbb{R}^{m\times m}$ *such that*[2] $\left[\begin{smallmatrix} A & X \\ X' & B \end{smallmatrix}\right] \succcurlyeq 0$ *and* $\operatorname{tr} A + \operatorname{tr} B \leq 2t$.

Proof. Note that for any matrix W, $\|W\|_{\text{Fro}} = \operatorname{tr} WW'$. If $\left[\begin{smallmatrix} A & X \\ X' & B \end{smallmatrix}\right] \succcurlyeq 0$, we can write it as a product $\left[\begin{smallmatrix} U \\ V \end{smallmatrix}\right] \left[\begin{smallmatrix} U' & V' \end{smallmatrix}\right]$. We have $X = UV'$ and $\frac{1}{2}(\|U\|^2_{\text{Fro}} + \|V\|^2_{\text{Fro}}) = \frac{1}{2}(\operatorname{tr} A + \operatorname{tr} B) \leq t$, establishing $\|X\|_\Sigma \leq t$. Conversely, if $\|X\|_\Sigma \leq t$ we can write it as $X = UV'$ with $\operatorname{tr} UU' + \operatorname{tr} VV' \leq 2t$ and consider the p.s.d. matrix $\left[\begin{smallmatrix} UU' & X \\ X' & VV' \end{smallmatrix}\right]$. □

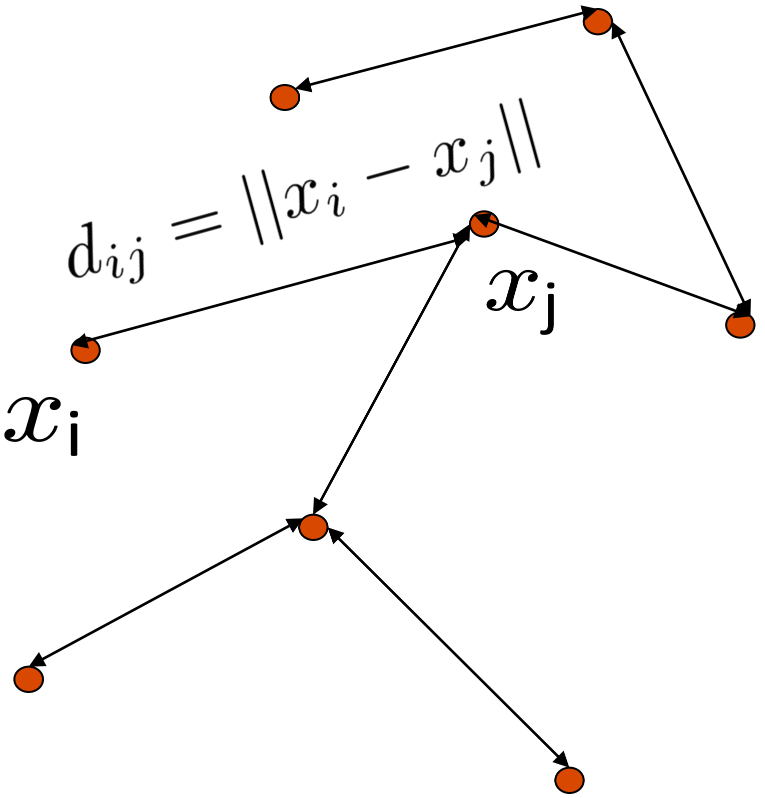
Matrix Completion formulation

$$\begin{array}{ll}
\min & \operatorname{trace}\begin{pmatrix} W_1 & X \\ X^\top & W_2 \end{pmatrix} \\
\text{s.t.} & X_{ij} = M_{ij}, \;\; (i,j) \in I \\
 & \begin{pmatrix} W_1 & X \\ X^\top & W_2 \end{pmatrix} \succeq 0 \\
 & X \in \mathbf{R}^{m\times n}, \\
 & W_1 \in \mathbf{R}^{m\times m}, \;\; W_2 \in \mathbf{R}^{n\times n}
\end{array}$$

Sensor network localization



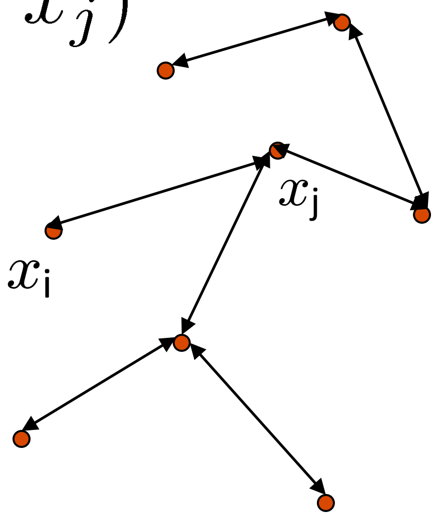
Sensor network localization



SDP relaxation of sensor network localization problem

- Given partial information on pair-wise distances find all distances and the exact locations of sensors.

$$d_{ij} = \|x_i - x_j\|^2 = (x_i - x_j)^\top (x_i - x_j)$$



Looking for matrix $X \in \mathbf{R}^{n \times 3}$
such that for all pairs (i, j)
for which d_{ij} is known
$\|x_i - x_j\|^2 = \|x_i\|^2 + \|x_j\|^2 - 2{x_i}^\top x_j$

Find $Y = XX^\top$: $Y_{ij} = {x_i}^\top x_j$
linear constaints $Y_{ii} + Y_{jj} - 2Y_{ij} = d_{ij}$, Y-low rank

Convex relaxation via trace

- Given the distances between some elements, find the matrix with a given rank (2 or 3)

$$\begin{aligned} & \mathbf{rank}(Y) = 2\ (3) \\ \text{s.t.}\quad & Y_{ii} + Y_{jj} - 2Y_{ij} = d_{ij},\ (i,j) \in I \\ & Y \succeq 0 \end{aligned}$$

- NP-hard problem.

$\mathbf{rank}(Y) = \|\lambda(Y)\|_0,$
where $\lambda(Y)$ is the vector of eignevalues values of Y.

Convex relaxation: $\|\lambda(Y)\|_1 = \sum_{i=1}^{n} \lambda_i(Y) = \text{trace}(Y)$

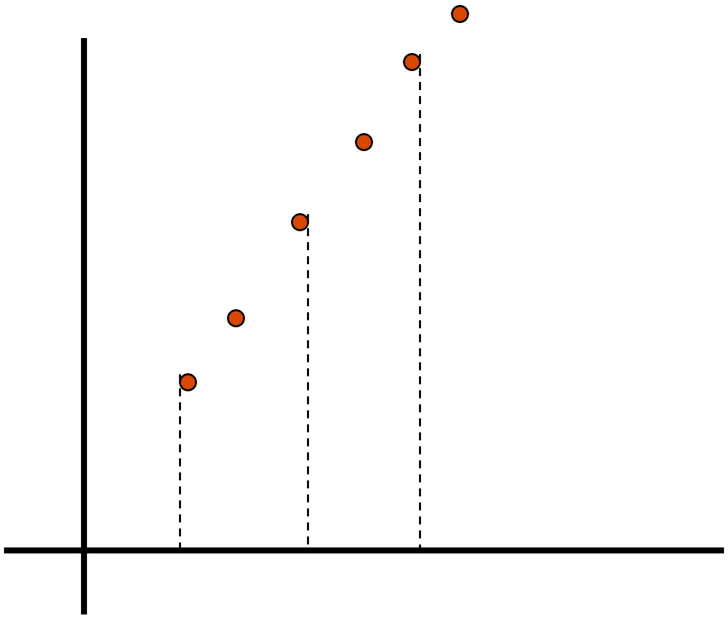
SDP relaxation for sensor network localization

$$\begin{array}{ll} \min & \operatorname{trace}(Y) \\ \text{s.t.} & Y_{ii} + Y_{jj} - 2Y_{ij} = d_{ij}, \ (i,j) \in I \\ & Y \succeq 0 \\ & Y \in \mathbf{R}^{n \times n} \end{array}$$

Metric multidimensional scaling

Let us take three points in $\mathbf{R}^2$:

$$\begin{aligned} x_1 &= (2,1) \\ x_2 &= (4,2) \\ x_3 &= (6,2) \end{aligned}$$



$$Y = XX^\top = \begin{bmatrix} 5 & 10 & 15 \\ 10 & 20 & 30 \\ 15 & 30 & 45 \end{bmatrix} = \begin{bmatrix} \sqrt{5} \\ 2\sqrt{5} \\ 3\sqrt{5} \end{bmatrix} \begin{bmatrix} \sqrt{5} & 2\sqrt{5} & 3\sqrt{5} \end{bmatrix}$$

Preserving the inner products $x_i^\top x_j$ we preserve distances $||x_i - x_j||^2$

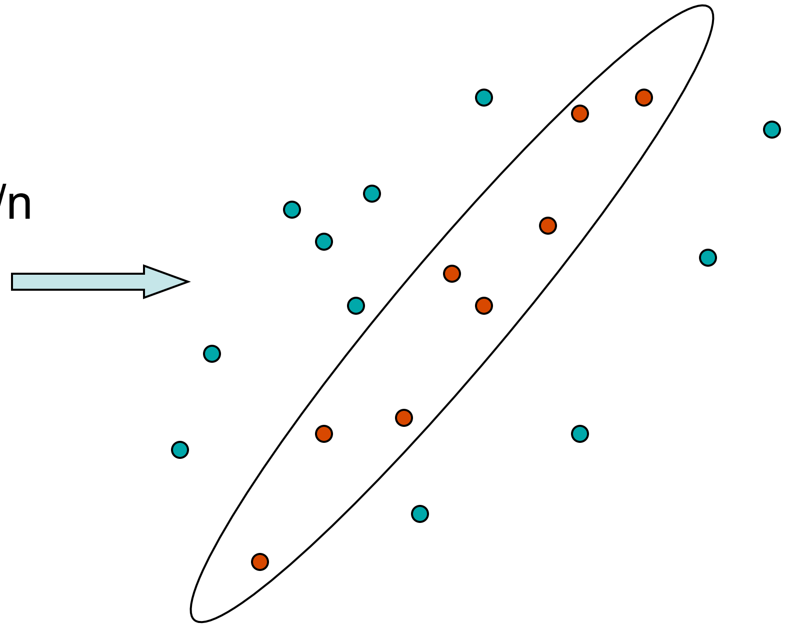
Dimensionality reduction

Principal Component Analysis

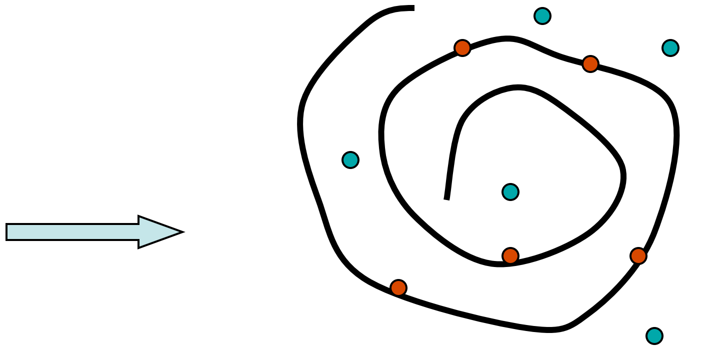
Select few largest eigenvectors of $Y=X^TX/n$

Multidimensional Metric Scaling

Select few largest eigenvectors of $Y=XX^T$



?



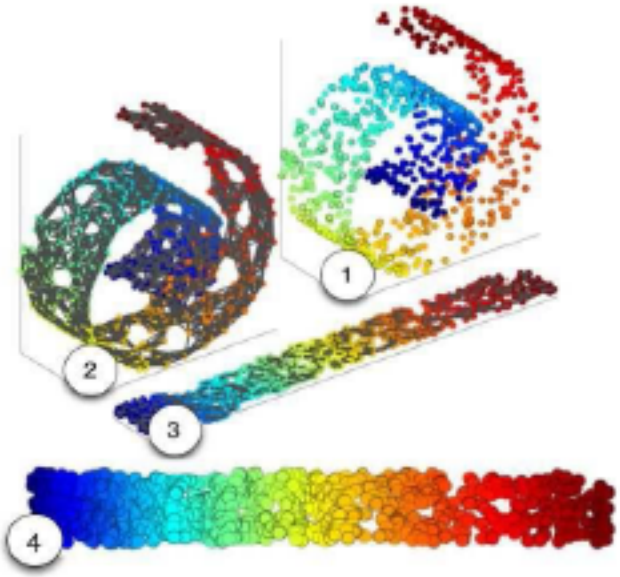


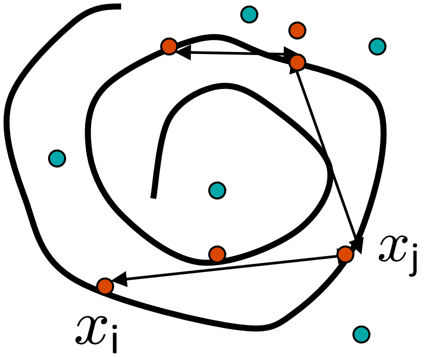
Figure 1. The problem of manifold learning, illustrated for $N = 800$ data points sampled from a "Swiss roll". (1). A discretized manifold is revealed by connecting each data point and its $k = 6$ nearest neighbors (2). An unsupervised learning algorithm unfolds the Swiss roll while preserving the local geometry of nearby data points (3). Finally, the data points are projected onto the two dimensional subspace that maximizes their variance, yielding a faithful embedding of the original manifold (4).

Nonlinear dimensionality reduction

- Preserve pair-wise distances between neighboring points.

$$d_{ij} = \|x_i - x_j\|^2 = (x_i - x_j)^\top (x_i - x_j), \text{ for } (i,j) \in I$$

Looking for matrix $Z \in \mathbf{R}^{n \times k}$ (for some k) such that for all pairs $(i,j) \in I$
$\|z_i - z_j\|^2 = \|z_i\|^2 + \|z_j\|^2 - 2{z_i}^\top z_j$



Find $Y = ZZ^\top$: $Y_{ij} = {z_i}^\top z_j$
constaints $Y_{ii} + Y_{jj} - 2Y_{ij} = d_{ij}$, $Y \succeq 0$

SDP formulation for nonlinear dimensionality reduction

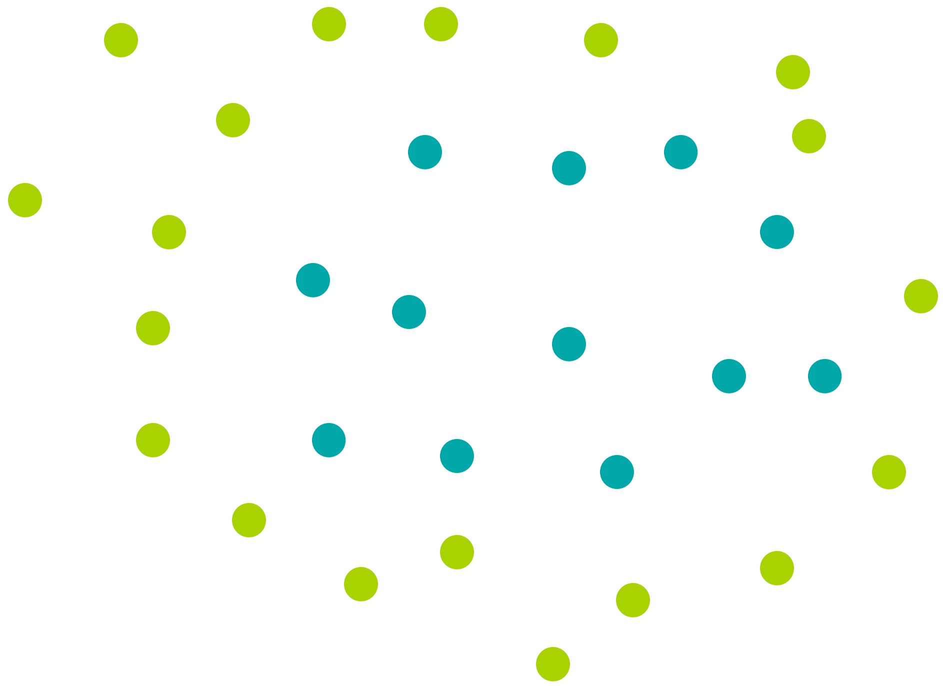
$$\begin{array}{ll}
\max & \mathrm{trace}(Y) \\
 & \sum_{ij} Y_{ij} = 0 \\
\mathrm{s.t.} & Y_{ii} + Y_{jj} - 2Y_{ij} = d_{ij}, \;\; (i,j) \in I \\
 & Y \succeq 0 \\
 & Y \in \mathbf{R}^{n \times n}
\end{array}$$

Distance metric learning

Gaussian Kernel

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma}}$$

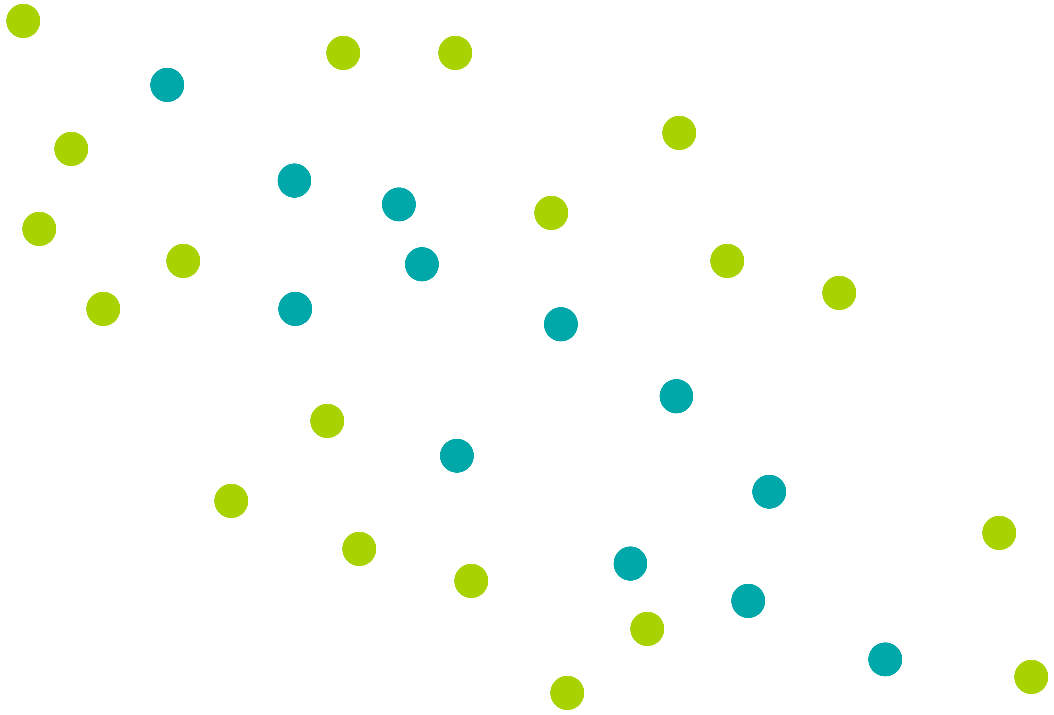
Euclidean distance $\|x_i - x_j\|^2 = (x_i - x_j)^\top (x_i - x_j)$



Gaussian Kernel

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma}}$$

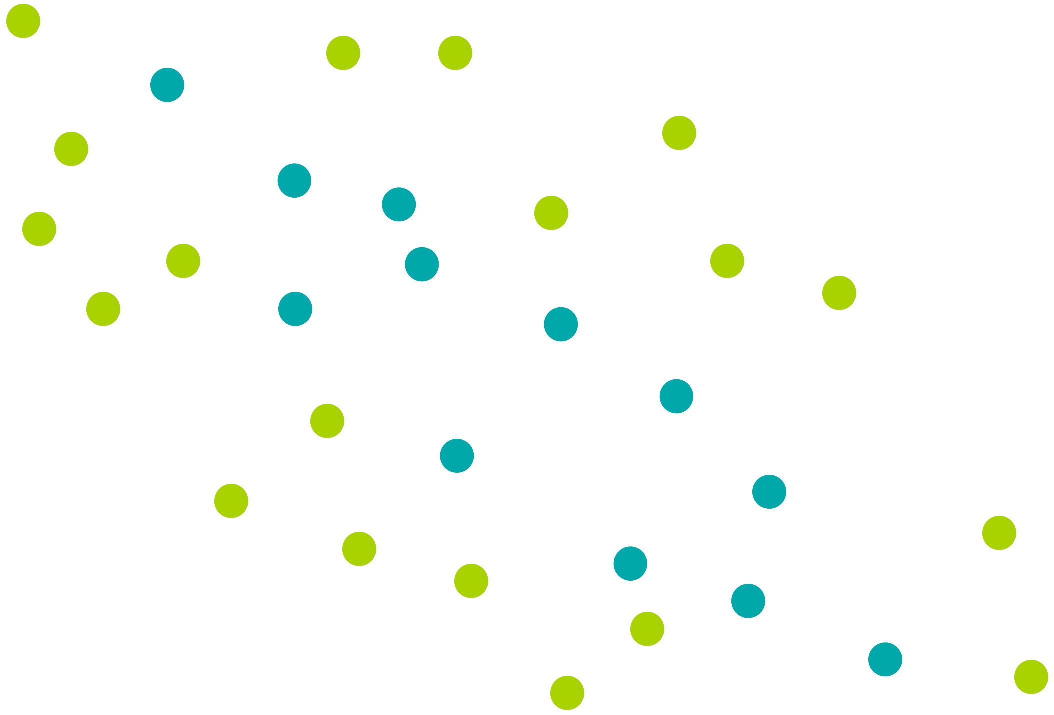
Euclidean distance $\|x_i - x_j\|^2 = (x_i - x_j)^\top (x_i - x_j)$



Gaussian Kernel

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|_M^2}{2\sigma}}$$

Mahalanobis distance $\|x_i - x_j\|_M^2 = (x_i - x_j)^\top M (x_i - x_j)$



Distance Metric Learning

$$\begin{aligned} \max_M \quad & \sum_{(i,j)\in D} (x_i - x_j)^\top M (x_i - x_j) \\ \text{s.t.} \quad & \sum_{(i,j)\in S} (x_i - x_j)^\top M (x_i - x_j) \leqslant 1 \\ & M \succeq 0 \end{aligned}$$

S – the set of similarly labeled examples, card(S)~ O(n^2)

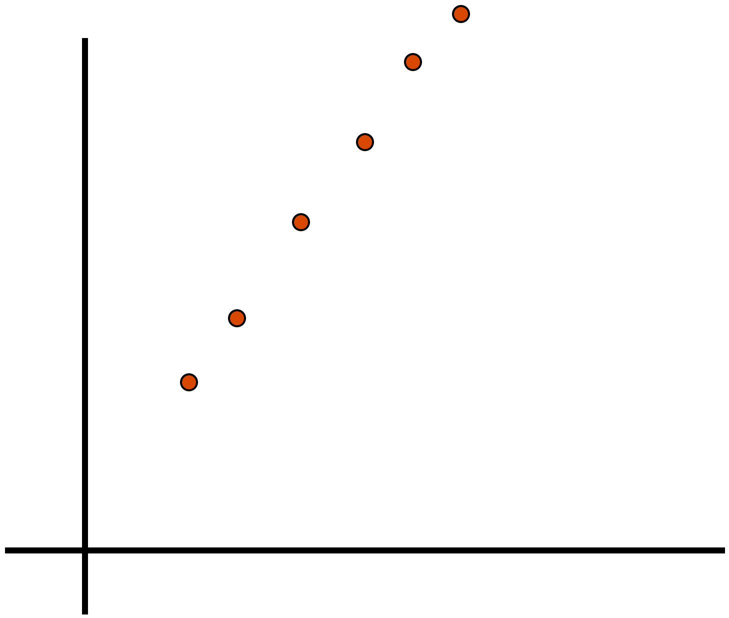
D – the set of differently labeled examples, card(D)~O(n^2)

IPM is too expensive, need a first order method approach

Principal component analysis

Let us take three points in $\mathbf{R}^2$:

$$\begin{aligned} x_1 &= (2, 1) \\ x_2 &= (4, 2) \\ x_3 &= (6, 2) \end{aligned}$$

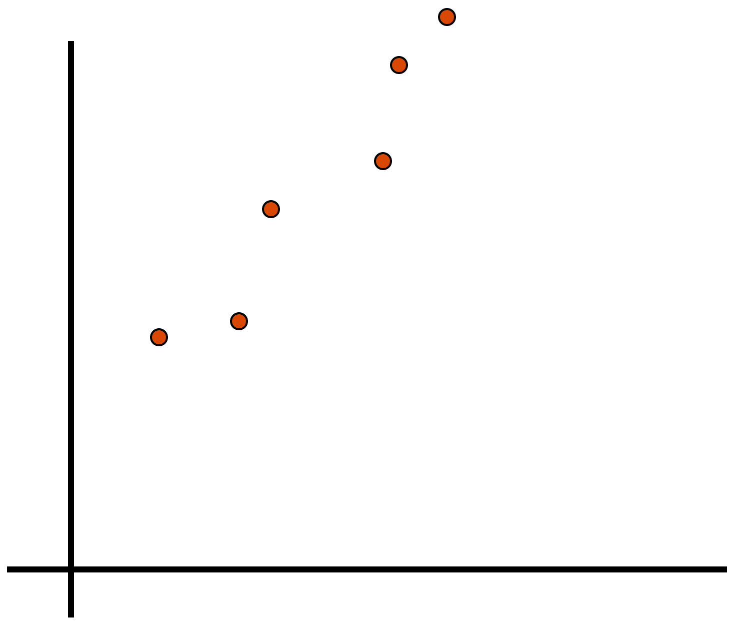


$$Y = \frac{1}{3}\sum_{i=1}^{3} x_i {x_i}^\top = \frac{1}{3}\left(\left[\begin{array}{cc} 4 & 2 \\ 2 & 1 \end{array}\right] + \left[\begin{array}{cc} 16 & 8 \\ 8 & 4 \end{array}\right] + \left[\begin{array}{cc} 36 & 18 \\ 18 & 9 \end{array}\right]\right)$$

$$Y = \frac{1}{3}\sum_{i=1}^{3} x_i {x_i}^\top = \frac{1}{3}\left[\begin{array}{cc} 56 & 28 \\ 28 & 14 \end{array}\right] = \frac{14}{3}\left[\begin{array}{c} 2 \\ 1 \end{array}\right]\left[\begin{array}{cc} 2 & 1 \end{array}\right]$$

Principal component analysis

Find direction of largest variance



Let us take three points in $\mathbf{R}^2$:

$$\begin{aligned} y_1 &= (2,1) + (O(\epsilon), O(\epsilon)) \\ y_2 &= (4,2) + (O(\epsilon), O(\epsilon)) \\ y_3 &= (6,3) + (O(\epsilon), O(\epsilon)) \end{aligned}$$

$$A = \frac{14}{\sqrt{3}} \left[\begin{array}{c} 2/\sqrt{3} \\ 1/\sqrt{3} \end{array} \right] \left[\begin{array}{cc} 2/\sqrt{3} & 1/\sqrt{3} \end{array} \right] + \left[\begin{array}{cc} O(\epsilon) & O(\epsilon) \\ O(\epsilon) & O(\epsilon) \end{array} \right]$$

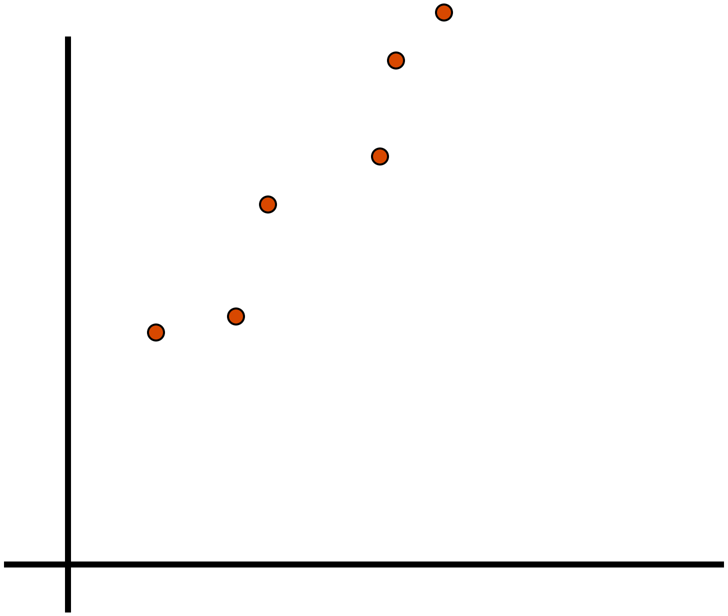
$$\left[\begin{array}{c} 2/\sqrt{3} \\ 1/\sqrt{3} \end{array} \right] = \mathrm{argmax}_{x \in \mathbf{R}^2,\ \|x\|=1} x^\top A x$$

Sparse principal component analysis

Find a **sparse** direction of largest variance

Let us take three points in $\mathbf{R}^2$:

$$\begin{aligned} y_1 &= (2,1) + (O(\epsilon), O(\epsilon)) \\ y_2 &= (4,2) + (O(\epsilon), O(\epsilon)) \\ y_3 &= (6,3) + (O(\epsilon), O(\epsilon)) \end{aligned}$$

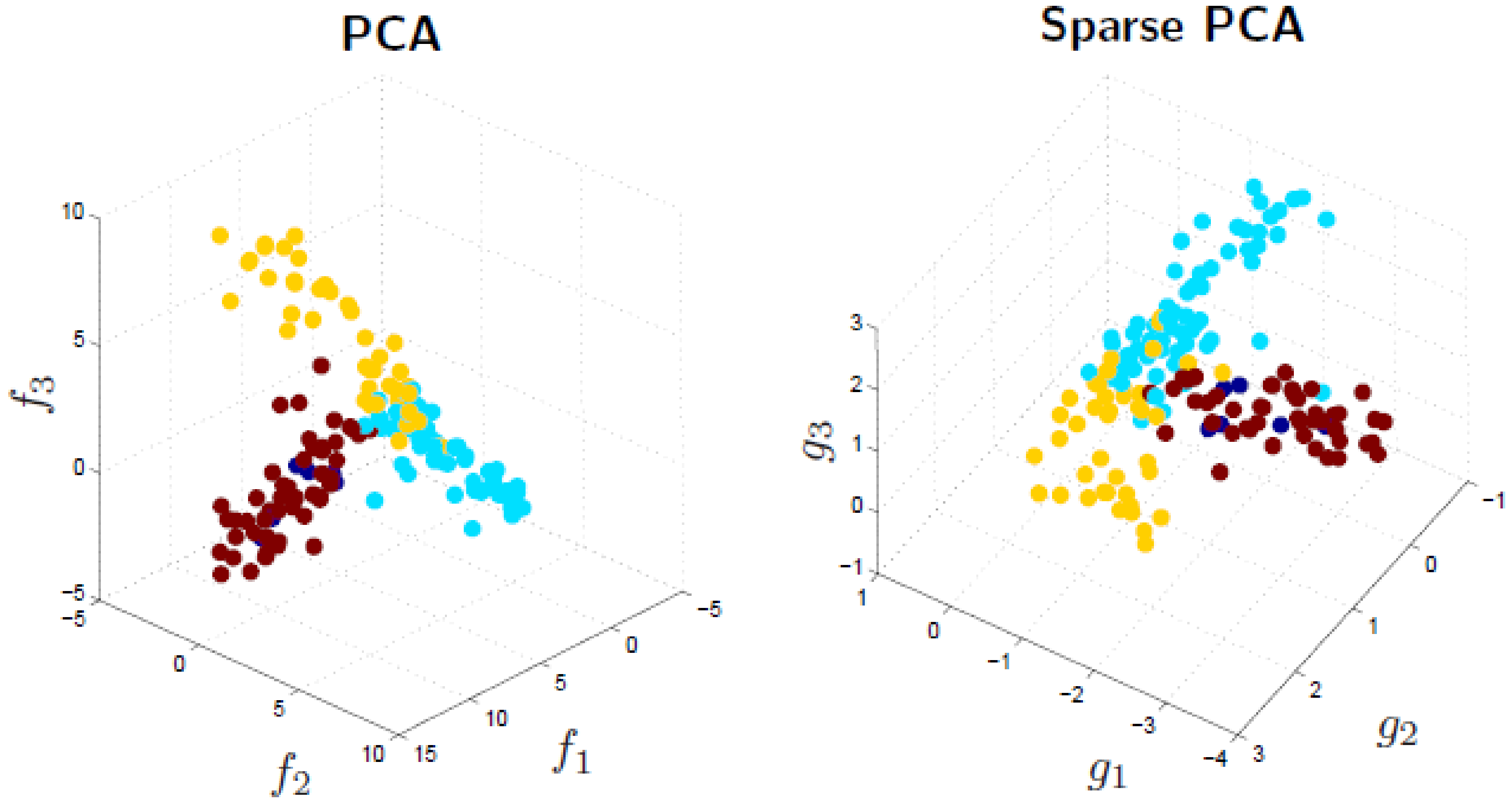


$$A = \frac{1}{3}\begin{bmatrix} 56+O(\epsilon) & 28+O(\epsilon) \\ 28+O(\epsilon) & 14+O(\epsilon) \end{bmatrix} \approx \frac{56}{3}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix} + \begin{bmatrix} O(\epsilon) & O(\epsilon) \\ O(\epsilon) & O(\epsilon) \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mathrm{argmax}_{x\in\mathbf{R}^2,\ \|x\|=1,\ \mathrm{card}(x)=1}\, x^\top A x$$

Introduction

Clustering of gene expression data in PCA versus sparse PCA, on 500 genes.


PCA

Sparse PCA

The PCA factors f_i on the left are dense and each use all 500 genes.
The sparse factors g_1, g_2 and g_3 on the right involve 6, 4 and 4 genes respectively.

Sparse PCA

Given a set $Y \in \mathbf{R}^m \times n$ compute empirical covariance matrix $A = \frac{1}{m} Y^\top Y$

Principal component analysis

Maximize the variance explained by factor x

$$\begin{array}{ll} \max\limits_{x \in \mathbf{R}^n} & x^\top A x \\ \text{s.t.} & \|x\|_2 = 1 \end{array}$$

Sparse principal component analysis

Maximize the variance explained by a factor x with bounded cardinality

$$\begin{array}{ll} \max_{x \in \mathbf{R}^n} & x^\top A x \\ \text{s.t.} & card(x) = k \\ & \|x\|_2 = 1 \end{array}$$

Semidefinite relaxation

Start from:

$$\begin{array}{ll} \text{maximize} & x^T A x \\ \text{subject to} & \|x\|_2 = 1 \\ & \mathbf{Card}(x) \leq k, \end{array}$$

where $x \in \mathbf{R}^n$. Let $X = xx^T$ and write everything in terms of the matrix X:

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{Card}(X) \leq k^2 \\ & X = xx^T, \end{array}$$

Replace $X = xx^T$ by the equivalent $X \succeq 0, \;\; \mathbf{Rank}(X) = 1$:

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{Card}(X) \leq k^2 \\ & X \succeq 0, \;\; \mathbf{Rank}(X) = 1, \end{array}$$

again, this is the same problem.

Semidefinite relaxation

We have made some progress:

- The objective $\mathbf{Tr}(AX)$ is now linear in X
- The (non-convex) constraint $\|x\|_2 = 1$ became a linear constraint $\mathbf{Tr}(X) = 1$.

But this is still a hard problem:

- The $\mathbf{Card}(X) \leq k^2$ is still non-convex.
- So is the constraint $\mathbf{Rank}(X) = 1$.

We still need to relax the two non-convex constraints above:

- If $u \in \mathbf{R}^p$, $\mathbf{Card}(u) = q$ implies $\|u\|_1 \leq \sqrt{q}\|u\|_2$. So we can replace $\mathbf{Card}(X) \leq k^2$ by the weaker (but convex): $\mathbf{1}^T|X|\mathbf{1} \leq k$.
- We simply drop the rank constraint

Semidefinite Programming

Semidefinite relaxation:

$$\begin{array}{ll} \text{maximize} & x^T A x \\ \text{subject to} & \|x\|_2 = 1 \\ & \mathbf{Card}(x) \leq k, \end{array} \qquad \textbf{becomes} \qquad \begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{1}^T |X| \mathbf{1} \leq k \\ & X \succeq 0, \end{array}$$

- This is a **semidefinite program** in the variable $X \in \mathbf{S}^n$...
- Solve small problems (a few hundred variables) using IP solvers, etc.
- Dimensionality reduction apps: solve very large instances.

Solution: use first order algorithm...