# Lecture: Dimensionality Reduction

http://bicmr.pku.edu.cn/~wenzw/bigdata2018.html

Acknowledgement: Some of these slides are based on Prof. Jure Leskovec's and Prof. Yinyu Ye's lecture notes

# Why Reduce Dimensions?

Why reduce dimensions?

- Discover hidden correlations/topics
  - Words that occur commonly together
- Remove redundant and noisy features
  - Not all words are useful
- Interpretation and visualization
- Easier storage and processing of the data

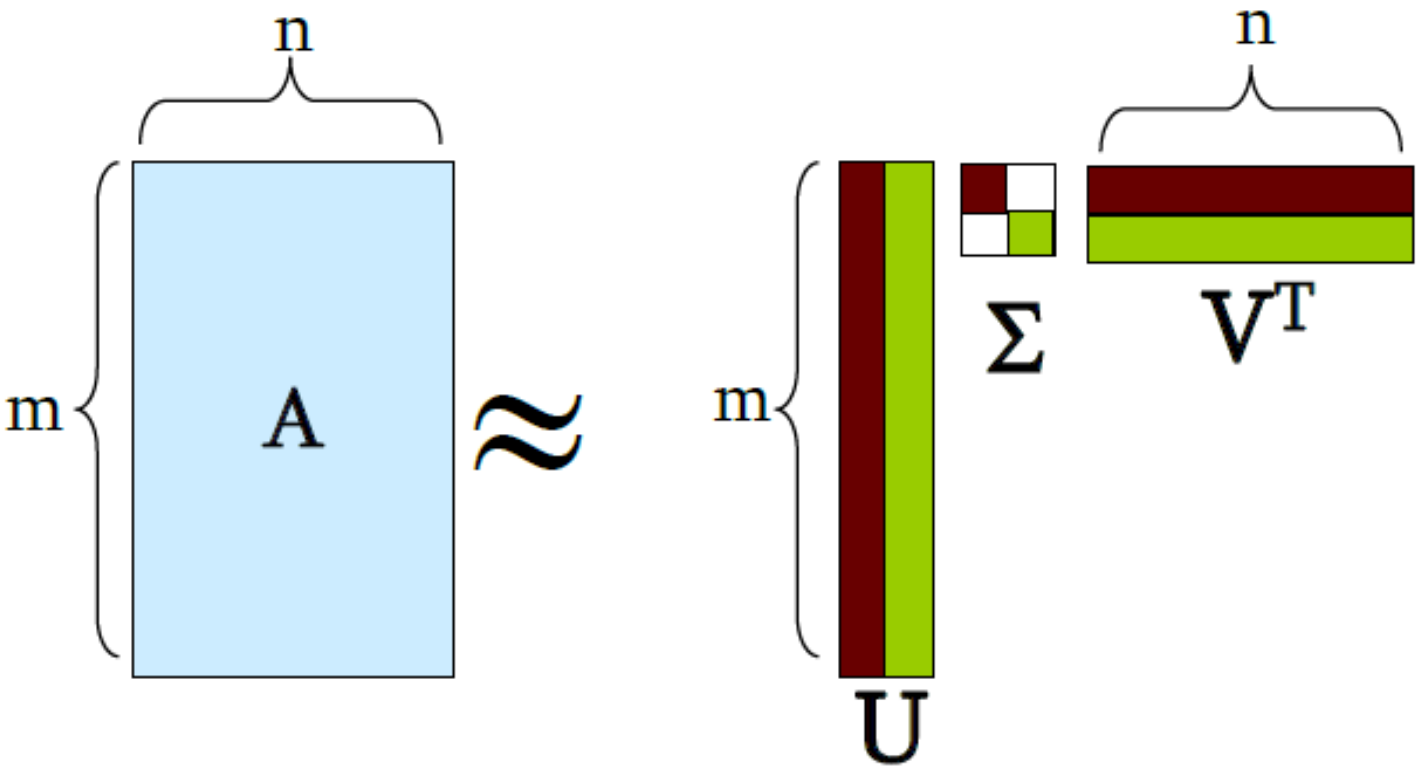
# SVD - Definition

$$A_{[m\times n]} = U_{[m\times r]}\Sigma_{[r\times r]}V^{\top}_{[r\times n]}$$

- $A$: Input data matrix
  - $m \times n$ matrix (e.g., $m$ documents, $n$ terms)
- $U$: Left singular vectors: $U^T U = I$
  - $m \times r$ matrix ($m$ documents, $r$ concepts)
- $\Sigma$: Singular values
  - $r \times r$ diagonal matrix (strength of each 'concept') ($r$ : rank of the matrix $A$)
- $V$: Right singular vectors: $V^T V = I$
  - $n \times r$ matrix ($n$ terms, $r$ concepts)

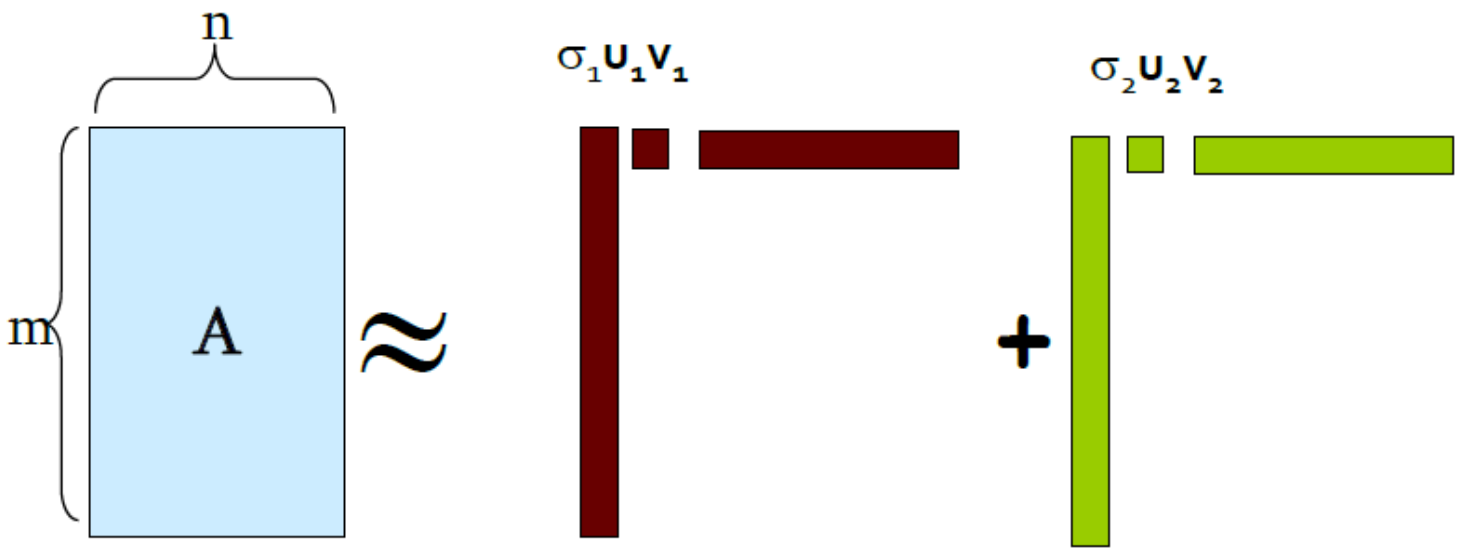
# SVD

$$A \approx U\Sigma V^\top = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$



# SVD

$$A \approx U\Sigma V^\top = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$



# SVD - Properties

## Theorem: SVD

If $A$ is a real $m$-by-$n$ matrix, then there exits

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{m \times m} \text{ and } V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$$

such that $U^T U = I$, $V^T V = I$ and

$$U^T A V = \text{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n}, \quad p = \min(m, n),$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$.

- Proof: Let $V_1 \in \mathbb{R}^{n \times r}$ has orthonormal columns, then exits $V_2 \in \mathbb{R}^{n \times (n-r)}$ such that $V = [V_1, V_2]$ is orthogonal.
- Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ be unit 2-norm vectors: $Ax = \sigma y$ with $\sigma = \|A\|_2$. Then exists $V_2 \in \mathbb{R}^{n \times (n-1)}$ and $U_2 \in \mathbb{R}^{m \times (m-1)}$ so $V = [x, V_2] \in \mathbb{R}^{n \times n}$ and $U = [y, U_2] \in \mathbb{R}^{m \times m}$ are orthogonal.

- Then it can be proved that $U^TAV$ has the following structure

$$U^TAV = \begin{pmatrix} \sigma & w^T \\ 0 & B \end{pmatrix} \equiv A_1.$$

Since

$$\left\| A_1 \begin{pmatrix} \sigma \\ w \end{pmatrix} \right\|_2^2 \geq (\sigma^2 + w^T w)^2,$$

we have $\|A_1\|_2^2 \geq (\sigma^2 + w^T w)$. But $\sigma^2 = \|A\|_2^2 = \|A_1\|_2^2$, and so we must have $w = 0$. An induction gives the proof.

Properties:

- $AV = U\Sigma$, $A^TU = V\Sigma^T$: $Av_i = \sigma u_i$, $A^Tu_i = \sigma_i v_i$, $i = 1, \ldots, p$.
- $rank(A) = r$, $null(A) = span\{v_{r+1}, \ldots, v_n\}$, $ran(A) = span\{u_1, \ldots, u_r\}$
- $A = \sum_{i=1}^r \sigma_i u_i v_i^T$
- $\|A\|_F^2 = \sigma_1^2 + \ldots + \sigma_p^2$, $\|A\|_2 = \sigma_1$

# SVD - Best Low Rank Approximation

## Theorem

Let the SVD of $A \in \mathbb{R}^{m\times n}$ be given in Theorem: SVD. If $k < r = rank(A)$ and $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, then

$$\min_{rank(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}.$$

- Proof: Since $U^T A_k V = \text{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)$ it follows that $rank(A_k) = k$ and $U^T(A - A_k)V = \text{diag}(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_p)$. Hence $\|A - A_k\|_2 = \sigma_{k+1}$.
- Suppose $rank(B) = k$ for some $B \in \mathbb{R}^{m\times n}$. We can find orthonormal vectors $x_1, \ldots, x_{n-k}$ so $null(B) = span\{x_1, \ldots, x_{n-k}\}$. A dimension argument shows:

$$span\{x_1, \ldots, x_{n-k}\} \cap span\{v_1, \ldots, v_{k+1}\} \neq \{0\}$$

- Let $z$ be a unit 2-norm vector in this intersection. Since $Bz = 0$ and

$$Az = \sum_{i=1}^{k+1} \sigma_i (v_i^T z) u_i,$$

we have

$$\|A - B\|_2^2 \geq \|(A - B)z\|_2^2 = \|Az\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 (v_i^T z)^2 \geq \sigma_{k+1}^2$$

Comments:

- So zeroing small $\sigma_i$ introduces less error
- How many $\sigma$s to keep? Rule of thumb: keep 80-90% of 'energy' $(= \sum \sigma_i^2)$

# SVD - Complexity

- To compute SVD: $O(nm^2)$ or $O(n^2m)$
- But:
  - Less work, if we just want singular values
  - or if we want first k singular vectors
  - or if the matrix is sparse
- Implemented in linear algebra packages like
  - Dense matrix: LAPACK
  - Sparse Matrix: ARPACK, PROPACK
  - High Level Software packages: Matlab, SPlus, Mathematica ...

# Relation to Eigen-decomposition

- SVD gives us $A = U\Sigma V^\top$
- Eigen-decomposition: $A = X\Lambda X^\top$
  - $A$ is symmetric
  - $U, V, X$ are orthonormal
  - $\Lambda, \Sigma$ are diagonal
- $AA^\top = U\Sigma\Sigma^\top U^\top$
- $A^\top A = V\Sigma\Sigma^\top V^\top$
- $\lambda_i(A^\top A) = \sigma_i^2(A)$

# Outline

# Dimensionality Reduction

- Assume we have a dataset represented in an $n \times D$ matrix $\mathbf{X}$ consisting of $n$ data vectors $\mathbf{x}_i$ with dimensionality $D$.
  $\mathbf{x}_i$ is the $i$th row of $\mathbf{X}$.
- Assume further that this dataset has intrinsic dimensionality $d$ (often $d \ll D$).
- Dimensionality reduction techniques transform dataset $\mathbf{X}$ with dimensionality $D$ into a new dataset $\mathbf{Y}$ with dimensionality $d$, while retaining the geometry of the data as much as possible.

# Principal Component Analysis (PCA)

- Principal component analysis (PCA) constructs a low-dimensional representation of the data that describes as much of the variance in the data as possible.
- Let $x \in \mathbb{R}^D$ be a random vector with mean $\mu$ and covariance $\Sigma$.
- Find $y = \sum_{i=1}^{D} z_i x_i = z^T x$ such that the variance of $y$ is maximized.

$$\begin{aligned} Var(y) &= E[(z^T x - E(z^T x))(z^T x - E(z^T x))] \\ &= E(z^T x - z^T Ex)(z^T x - z^T Ex)^T \\ &= z^T E[(x - Ex)(x - Ex)^T]z = z^T \text{cov}(\mathbf{X})z \end{aligned}$$

- Model problem:

$$\max \; z^T \text{cov}(\mathbf{X})z, \qquad \text{s.t. } \|z\|_2 = 1.$$

Find the maximal eigenpair $(\lambda_1, v_1)$ of $\text{cov}(\mathbf{X})$.

# Principal Component Analysis (PCA)

- find a linear mapping $\mathbf{M}$ that maximizes data variance

$$\begin{aligned} \max_{\mathbf{M}} \quad & \mathrm{Tr}(\mathbf{M}^\top \mathrm{cov}(\mathbf{X})\mathbf{M}) \\ \text{s.t.} \quad & \mathbf{M}^\top \mathbf{M} = I \end{aligned}$$

- Lagrangian function:

$$L(\mathbf{M}, \Lambda) = \mathrm{Tr}(\mathbf{M}^\top \mathrm{cov}(\mathbf{X})\mathbf{M}) - \left\langle \Lambda, \mathbf{M}^\top \mathbf{M} - I \right\rangle$$

- Stationary point (KKT condition) at

$$\begin{aligned} \mathrm{cov}(\mathbf{X})\mathbf{M} &= \mathbf{M}\Lambda \\ \mathbf{M}^\top \mathbf{M} &= I \end{aligned}$$

- Thus PCA essentially requires eigenvalue decomposition

# PCA

- Let $\bar{\mathbf{X}} = \mathbf{X} - \frac{1}{n} 1 1^T \mathbf{X}$, where $1$ is a column vector of all ones
- SVD: $\bar{\mathbf{X}} = U \Sigma V^T$, $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$.
- PCA computes

$$\max_{\mathbf{M}} \quad \text{Tr}(\mathbf{M}^\top \text{cov}(\mathbf{X}) \mathbf{M}), \text{ s.t.} \quad \mathbf{M}^\top \mathbf{M} = I,$$

where

$$\begin{aligned} \text{cov}(\mathbf{X}) &= \frac{1}{n-1} (\mathbf{X} - E\mathbf{X})^T (\mathbf{X} - E\mathbf{X}) \\ &= \frac{1}{n-1} \left( (X - \frac{1}{n} 1 1^T X)^T (I - \frac{1}{n} 1 1^T X) \right) = \bar{\mathbf{X}}^T \bar{\mathbf{X}} \\ &= V \Lambda V^\top, \text{ where } V^T V = I, \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n) \end{aligned}$$

- PCA takes $d$ columns of $\bar{X} V = U \Sigma V^T V = U \Sigma$:

$$\tilde{X} = \bar{X} [v_1, \ldots, v_d] = [u_1 \sigma_1, \ldots, u_d \sigma_d] = U_d \Sigma_d$$

# Classical MDS

- Known data points $\mathbf{x}_i \in \mathbb{R}^D$, $i = 1, \ldots, n$.
  compute pairwise distance matrix $D(X)$ with

$$d_{ij}(\mathbf{X}) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

- Let $D_2(\mathbf{X}) = (d_{ij}^2(\mathbf{X}))$. MDS: Find $\mathbf{y}_i \in \mathbb{R}^d$ such that

$$\min_{\mathbf{Y}} \quad \|HD_2(\mathbf{X})H - HD_2(\mathbf{Y})H\|_F^2$$

**Lemma**

Let $D_2(\mathbf{X}) = (d_{ij}^2(\mathbf{X}))$ of $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \ldots \\ \mathbf{x}_n \end{pmatrix}$, $H = I_n - \frac{1}{n}11^T$. Let $B = -\frac{1}{2}HD_2(\mathbf{X})H$ and $\bar{\mathbf{X}} = \mathbf{X} - \frac{1}{n}11^T X$. Then we have $B = \bar{\mathbf{X}}\bar{\mathbf{X}}^T$.

# Classical MDS

- Proof: Define $K = \mathbf{X}\mathbf{X}^T$, $K_{ij} = \mathbf{x}_i\mathbf{x}_j^T$. Then

$$D_2(\mathbf{X})_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T = \mathbf{x}_i\mathbf{x}_i^T + \mathbf{x}_j\mathbf{x}_j^T - 2\mathbf{x}_i\mathbf{x}_j^T.$$

Define $w = \mathrm{diag}(K)$. Then

$$D_2(\mathbf{X}) = w1^T + 1w^T - 2K.$$

Thus

$$B = -\frac{1}{2}H(w1^T + 1w^T - 2K)H = H\mathbf{X}\mathbf{X}^T H = \bar{\mathbf{X}}\bar{\mathbf{X}}^T,$$

since $H1w^T = 1w^T - \frac{1^T 1}{n}1w^T = 0$ and $1w^T H = 0$.

# Classical MDS

- Let $B = \bar{\mathbf{X}}\bar{\mathbf{X}}^T$ and SVD: $\bar{\mathbf{X}} = U\Sigma V^T$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$. The eigenvalue decomposition: $B = U\Lambda U^T$, $\lambda_i = \sigma_i^2$.
- Assume that $\mathbf{Y}$ is centered, i.e., $-\frac{1}{2}HD_2(Y)H = YY^T$, then Classical MDS is equivalent to

$$\min_{\mathbf{Y}} \quad \|B - \mathbf{Y}\mathbf{Y}^T\|_F^2$$

- The optimal solution $\mathbf{Y} = U\Lambda^{\frac{1}{2}}$
- Classical MDS: take $\tilde{\mathbf{X}} = U_d\Lambda_d^{\frac{1}{2}} = U_d\Sigma_d$, where $U_d = [u_1, \ldots, u_d]$, $\Lambda_d = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ are the largest $d$ eigenvalues and $u_1, \ldots, u_d$ are the corresponding eigenvectors
- PCA and classical MDS are equivalent!

# Extension of MDS

- Known data points $\mathbf{x}_i \in \mathbb{R}^D$, $i = 1, \ldots, n$.
  Given norms $\|\cdot\|_x, \|\cdot\|_y$, compute pairwise distance matrix

$$d_{ij}(\mathbf{X}) = \|\mathbf{x}_i - \mathbf{x}_j\|_x, \quad d_{ij}(\mathbf{Y}) = \|\mathbf{y}_i - \mathbf{y}_j\|_y$$

- Find $\mathbf{y}_i \in \mathbb{R}^d$ such that

$$\min_{\mathbf{Y}} \quad \|D_x(\mathbf{X}) - D_y(\mathbf{Y})\|_F^2$$

  or with centering matrix $H = I_n - \frac{1}{n}11^T$ and $D_{2x}(\mathbf{X}) = (d_{ij}^2(\mathbf{X}))$:

$$\min_{\mathbf{Y}} \quad \|HD_{2x}(\mathbf{X})H - HD_{2y}(\mathbf{Y})H\|_F^2$$

- Kernel PCA: $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|_x^2 &= \|\mathbf{x}_i\|_x^2 + \|\mathbf{x}_j\|_x^2 - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle_x \\ (D_x(\mathbf{X})_{ij})^2 &= K_{ii} + K_{jj} - 2K_{ij} \end{aligned}$$

- Isomap: geodesic distance for $\mathbf{X}$ and F-norm for $\mathbf{Y}$

# Outline

# Maximum variance unfolding

- PCA and MDS are linear dimensionality reduction methods that compute mappings to preserve Euclidean distances between all pairs of data points
- Based on the notion of *isometry*, maximum variance unfolding (MVU) considers the much larger class of nonlinear transformations that only preserve the geometric properties of local neighborhoods
- We say that the data sets are $k$**-locally isometric** if for every point $x_i$, there exists a rotation and translation that maps $\mathbf{x}_i$ and its $k$ nearest neighbors $\{\mathbf{x}_{j1}, \mathbf{x}_{j2}, \ldots, \mathbf{x}_{jk}\}$ precisely onto the points $\mathbf{y}_i$ and $\{\mathbf{y}_{j1}, \mathbf{y}_{j2}, \ldots, \mathbf{y}_{jk}\}$
- The notion of isometry can be translated into various sets of equality constraints of inputs $\{\mathbf{x}_i\}_{i=1}^n$ and outputs $\{\mathbf{y}_i\}_{i=1}^n$

# MVU - constraints

- Inputs $\{\mathbf{x}_i\}_{i=1}^n$ and outputs $\{\mathbf{y}_i\}_{i=1}^n$ are locally isometric if whenever $\mathbf{x}_i$ and $\mathbf{x}_j$ are themselves neighbors or common neighbors of another point in the data set, we have:

$$\|\mathbf{y}_i - \mathbf{y}_j\|^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

- We also constrain the outputs $\mathbf{y}_i$ to be centered on the origin:

$$\sum_i \mathbf{y}_i = 0$$

this simply removes a translational degree of freedom from the final solution

# MVU - objective function

- Any "fold" between two points on a manifold serves to decrease the Euclidean distance between the points
- This suggests an optimization that we can perform to compute the outputs $\mathbf{y}_i$ that unfold a manifold sampled by inputs $\mathbf{x}_i$.
- Maximize the sum of pairwise squared distances between outputs:

$$\Phi = \frac{1}{2n} \|\mathbf{y}_i - \mathbf{y}_j\|^2$$

- We pull the outputs as far apart as possible, subject to the constraints in the previous slide
- It can be verified that the objective function is indeed bounded
- Intuitively, the constraints to preserve local distances prevent a divergence to pull the outputs infinitely far apart

# MVU - optimization

- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the graph formed by pairwise connecting each input to all of its $k$-nearest neighbors
- Then, in terms of the squared distance matrix $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$, the optimization can be written as:

$$\begin{aligned} \max \quad & \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ \text{s.t.} \quad & \sum_i \mathbf{y}_i = 0, \\ & \|\mathbf{y}_i - \mathbf{y}_j\|^2 = D_{ij}, \forall (i,j) \in \mathcal{E} \end{aligned}$$

- This problem is not convex, as it involves maximizing a quadratic form subject to quadratic equality constraints

# MVU - reformulation

- The inner product matrix $K_{ij} = \mathbf{y}_i \cdot \mathbf{y}_j$ determines the outputs up to rotation
- Expanding the square in the distance constraint we obtain

$$K_{ii} - 2K_{ij} + K_{jj} = D_{ij}$$

- Likewise, the centering constraint can be expressed as

$$0 = \| \sum_i \mathbf{y}_i \|^2 = \sum_{ij} \mathbf{y}_i \cdot \mathbf{y}_j = \sum_{ij} K_{ij}$$

- Both are now linear equality constraints on the elements of $K$
- We may view our original problem as an optimization over inner product matrices $K_{ij}$ rather than vectors $\mathbf{y}_i$
- Only symmetric matrices with nonnegative eigenvalues can be interpreted as inner product matrices, therefore

$$K = K^\top \succeq 0$$

# MVU - reformulation

- For the objective function:

$$\begin{aligned}\Phi &= \frac{1}{2n}\sum_{ij}(\|\mathbf{y}_i\|^2 + \|\mathbf{y}_j\|^2 + 2\mathbf{y}_i \cdot \mathbf{y}_j) \\ &= \sum_i \|\mathbf{y}_i\|^2 \\ &= \sum_i K_{ii} \\ &= \mathrm{Tr}(K).\end{aligned}$$

- We will obtain a low dimensional embedding by maximizing the trace of the inner product matrix

# MVU - reformulation

- We rewrite the MVU problem as:

$$\begin{aligned} \max \quad & \operatorname{Tr}(K) \\ \text{s.t.} \quad & K = K^\top \succeq 0, \\ & \mathbf{1}^\top K \mathbf{1} = 0, \\ & K_{ii} - 2K_{ij} + K_{jj} = D_{ij}, \forall (i,j) \in \mathcal{E} \end{aligned}$$

- This is a semidefinite program
  - The domain is the cone of psd matrices intersected with hyperplanes (equality constraints)
  - The objective function is bounded above and linear
  - The problem is guaranteed to be feasible because the constraints are trivially satisfied by the Gram matrix $G_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$
- The reformulation into SDP not only allows global and efficient solution of the MVU problem, but also gives the extra capability of estimating the intrinsic dimension

# MVU - reformulation

- From the inner product matrix $K^*$ solved by this SDP, the outputs $\mathbf{y}_i$ can be recovered by applying MDS on $K^*$
- Let $V_{\alpha i}$ denote the $i$th element of the $\alpha$th eigenvector, with eigenvalue $\lambda_\alpha$
- Then the inner product matrix can be written as

$$K^*_{ij} = \sum_{\alpha=1}^{n} \lambda_\alpha V_{\alpha i} V_{\alpha j}$$

- The embedding is obtained by identifying the $\alpha$th element of the output $\mathbf{y}_i$ as

$$Y_{\alpha i} = \sqrt{\lambda_\alpha} V_{\alpha i}$$

- A low dimensional embedding that is approximately locally isometric is given by truncating the elements of $\mathbf{y}_i$

# The dual MVU problem

- Examining the dual of an optimization problem often gives further insight and offers theoretical and computational advantages
- The MVU problem is no exception
- For notational convenience, write the last set of equality constraints as

$$\mathrm{Tr}(KE^{\{i,j\}}) = D_{ij}, \quad (i,j) \in \mathcal{E}$$

  where the $n \times n$ matrix $E^{\{i,j\}}$ has only four nonzero elements: $E^{\{i,j\}}_{ii} = E^{\{i,j\}}_{jj} = 1, E^{\{i,j\}}_{ij} = E^{\{i,j\}}_{ji} = -1$
- In forming the Lagrangian, we associate the dual variables $Z = Z^\top \succeq 0$, $\nu \in \mathbb{R}$ and $W_{ij}$ for $\forall (i,j) \in \mathcal{E}$

# The dual MVU problem

- The Lagrangian:

$$\begin{aligned} L(K,Z,\nu,W) &= \mathrm{Tr}(K) + \mathrm{Tr}(KZ) - \nu \mathbf{1}^\top K \mathbf{1} \\ &\quad - \sum_{(i,j)\in\mathcal{E}} W_{ij}(\mathrm{Tr}(KE^{\{i,j\}}) - D_{ij}) \\ &= \mathrm{Tr}[K(I + Z - \nu \mathbf{1}\mathbf{1}^\top - \sum_{(i,j)\in\mathcal{E}} W_{ij}E^{\{i,j\}})] \\ &\quad + \sum_{(i,j)\in\mathcal{E}} D_{ij}W_{ij} \end{aligned}$$

- The dual function is obtained as

$$\begin{aligned} & g(Z,\nu,W) = \sup_{K=K^\top} L(K,Z,\nu,W) \\ = & \begin{cases} \sum_{(i,j)\in\mathcal{E}} D_{ij}W_{ij} & \text{if } I + Z - \nu \mathbf{1}\mathbf{1}^\top - \sum_{(i,j)\in\mathcal{E}} W_{ij}E^{\{i,j\}} = 0 \\ +\infty & \text{otherwise} \end{cases} \end{aligned}$$

# The dual MVU problem

- Eliminating $Z$ from the equality, the feasibility condition in the above equation becomes

$$I - \nu \mathbf{1}\mathbf{1}^\top - L \preceq 0, \quad L = \sum_{(i,j)\in\mathcal{E}} W_{ij} E^{\{i,j\}}$$

- Note that L is a weighted Laplacian of the graph $\mathcal{G}$. The above linear matrix inequality is equivalent to

$$\nu \geq \frac{1}{n}, \quad \lambda_{n-1}(L) \geq 1$$

where $\lambda_{n-1}$ denotes the second smallest eigenvalue of a symmetric matrix (Here $\lambda_n(L) = 0$ with associated eigenvector $\mathbf{1}$)

# The dual MVU problem

- The dual MVU problem is:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in\mathcal{E}} D_{ij}W_{ij} \\ \text{s.t.} \quad & \lambda_{n-1}(L) \geq 1, \\ & L = \sum_{(i,j)\in\mathcal{E}} W_{ij}E^{\{i,j\}} \end{aligned}$$

- This is a convex optimization problem because the function $\lambda_{n-1}(L)$ is concave under the implicit constraint $\lambda_n(L) = 0$
- Note that the dual variable $\nu$ does not appear in the problem

# Duality

- The following duality results hold for the primal MVU problem and the dual MVU problem
  - **Weak duality**: For any primal feasible $K$ and any dual feasible $W$, we have

$$\mathrm{Tr}(K) \leq \sum_{i,j} D_{ij} W_{ij}$$

    (Note that $W_{ij} = 0$ if $(i,j) \notin \mathcal{E}$) This can be seen by checking the duality gap
  - **Strong duality**: There exist a primal-dual feasible pair $(K^*, W^*)$ with zero duality gap, i.e.

$$\mathrm{Tr}(K^*) = \sum_{i,j} D_{ij} W^*_{ij}$$

    Strong duality follows from Slater's condition for constraint qualification

# Optimality conditions

- A pair $(K^*, W^*)$ is primal-dual optimal iff they satisfy the following KKT conditions
  - primal feasibility

$$K^* \succeq 0, \quad \mathbf{1}^\top K^* \mathbf{1} = 0$$
$$K^*_{ii} - 2K^*_{ij} + K^*_{jj} = D_{ij}, \forall (i,j) \in \mathcal{E}$$

  - dual feasibility

$$L^* = \sum_{(i,j)\in\mathcal{E}} W^*_{ij} E^{\{i,j\}}, \quad \lambda_{n-1}(L^*) \geq 1$$

  - complementary slackness

$$L^* K^* = K^* \quad \text{(why?)}$$

# Optimality conditions

- Note that we always have $\lambda_{n-1}(L^*) \geq 1$, thus the complementary slackness condition $L^*K^* = K^*$ means that the range of $K^*$ lies in the eigenspace (e.s.) of $L^*$ associated with $\lambda_{n-1}$
- Since $K^*$ is a dense Gram matrix while $L^*$ is a sparse weighted Laplacian, $L^*K^* = K^*$ means precisely

$$\text{top e.s. of dense } K^* \subseteq \text{ bottom e.s. of sparse } L^*$$

  where "bottom e.s." means the eigenspace associated with $\lambda_{n-1}$ (discard the eigenvector $\mathbf{1}$ and $\lambda_n = 0$)

- Another direct consequence is

$$r \leq \operatorname{rank}(K^*) \leq \text{ multiplicity of } \lambda_{n-1}(L^*)$$

  where $r$ is the dimension of the low dimensional representations obtained by performing MDS on $K^*$

# Outline

# Graph Realization and Sensor Network Localization Problems

- **Input**: $m$ known points (anchors) $a_k \in \mathbb{R}^2, k = 1, \ldots, m$, and $n$ unknown points (sensors or targets) $x_j \in \mathbb{R}^2, j = 1, \ldots, n$. For some pair of two points, we have a Euclidean distance measure $d_{kj}$ between $a_k$ and $x_j$ , or distance measure $d_{ij}$ between $x_i$ and $x_j$
- **Output**: Position estimation for all unknown points, and confidence measures on reliability of each position estimation
- **Objective**: Robust, fast and accurate

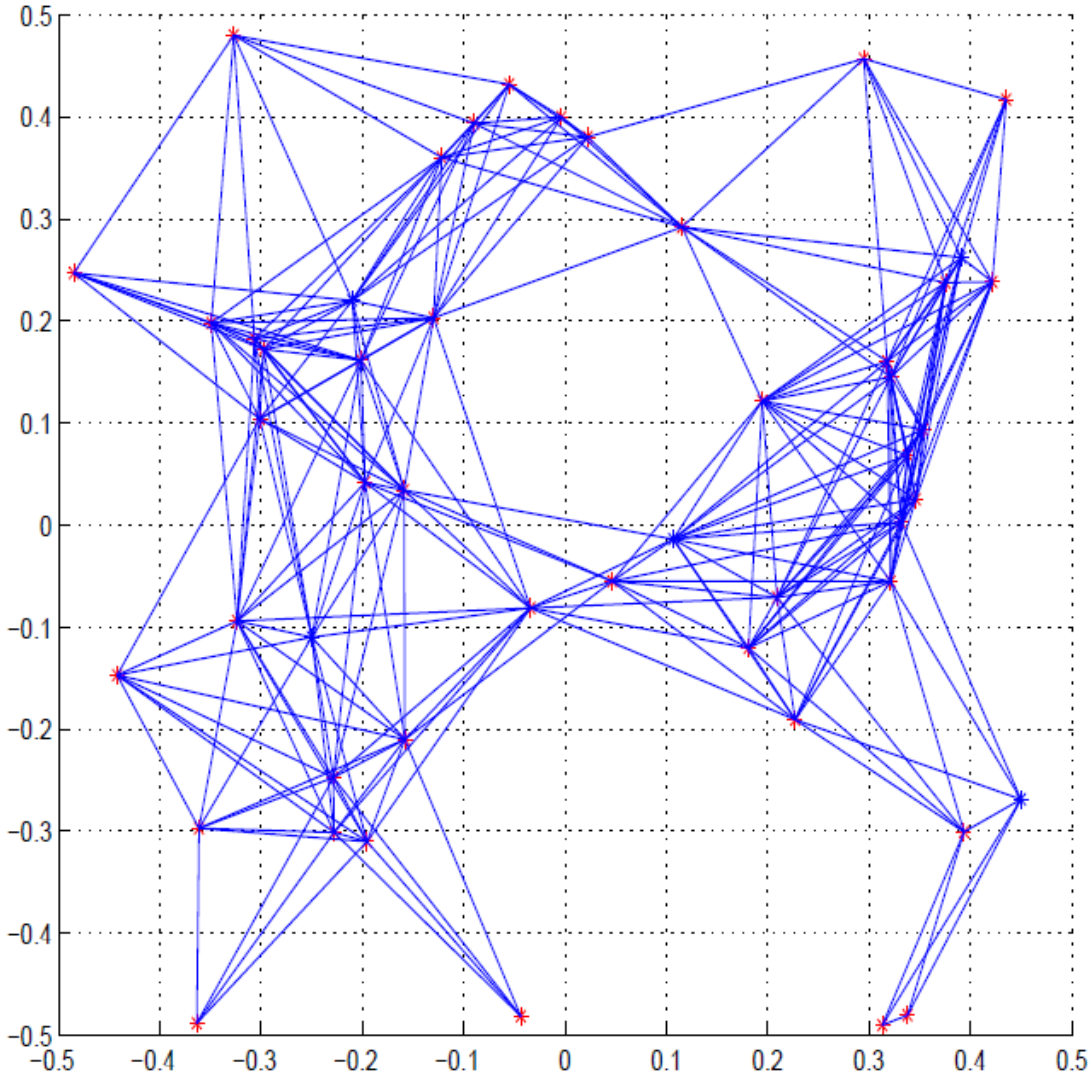



Figure: 50-Sensor Network with Radio Range .3

# Euclidean Distance Geometry Model

$$\|x_i - x_j\|^2 = d_{ij}^2, \forall (i,j) \in \mathcal{N}_x, i < j,$$
$$\|a_k - x_j\|^2 = d_{kj}^2, \forall (k,j) \in \mathcal{N}_a$$

$d_{ij}^2 (d_{kj}^2)$ connects $x_i$ to $x_j$ ($a_k$ to $x_j$) with an edge whose length is $d_{ij} (d_{kj})$

- Does the system has a localization or realization of all $x_j$s?
- Is the localization unique?
- Is the localization reliable or trustworthy?
- Is the system partially localizable?

# Global and Nonlinear Least Squares

$$\min \sum_{i,j \in \mathcal{N}_x} (\|x_i - x_j\|^2 - d_{ij}^2)^2 + \sum_{k,j \in \mathcal{N}_a} (\|a_k - x_j\|^2 - d_{kj}^2)^2$$

$$\min \sum_{i,j \in \mathcal{N}_x} (\|x_i - x_j\| - d_{ij})^2 + \sum_{k,j \in \mathcal{N}_a} (\|a_k - x_j\| - d_{kj})^2$$

For example, Moré and Wu (1997)

# Matrix representation

Let $X = [x_1, x_2, \ldots, x_n]$ be the $2 \times n$ matrix that needs to be determined. Then

$$\|x_i - x_j\|^2 = (e_i - e_j)^\top X^\top X (e_i - e_j),$$
$$\|a_k - x_j\|^2 = (a_k; -e_j)^\top [I \quad X]^\top [I \quad X](a_k; -e_j)$$

where $e_j$ is the vector of all zero except 1 at the $j$th position

$$(e_i - e_j)^\top Y (e_i - e_j) = d_{ij}^2, \forall i, j \in \mathcal{N}_x, i < j,$$
$$(a_k; -e_j)^\top \begin{pmatrix} I & X \\ X^\top & Y \end{pmatrix} (a_k; -e_j) = d_{kj}^2, \forall k, j \in \mathcal{N}_a,$$
$$Y = X^\top X$$

# SDP relaxation and analysis

- Change $Y = X^\top X$ to $Y \succeq X^\top X$, which is equivalent to (e.g., Boyd and Vandenberghe 2005)

$$Z = \begin{pmatrix} I & X \\ X^\top & Y \end{pmatrix} \succeq 0$$

Find a symmetric matrix $Z \in \mathbb{R}^{(2+n)\times(2+n)}$ such that

$$\begin{aligned}
&Z_{1:2,1:2} = I, \\
&(\mathbf{0}; (e_i - e_j))(\mathbf{0}; (e_i - e_j))^\top \bullet Z = d_{ij}^2, \forall (i,j) \in \mathcal{N}_x, i < j, \\
&(a_k; -e_j)(a_k; -e_j)^\top \bullet Z = d_{kj}^2, \forall (k,j) \in \mathcal{N}_a, \\
&Z \succeq 0.
\end{aligned}$$

- Any matrix solution for the SDP relaxation has rank at least 2
- If every sensor point is connected, directly or indirectly, to an anchor point, then the solution set must be bounded

# Outline

# Euclidean Distance Embedding

**Reference: Hou-Duo Qi, Xiaoming Yuan, Computing the nearest Euclidean distance matrix with low embedding dimensions, Math. Program., Ser. A, DOI 10.1007/s10107-013-0726-0**

- Suppose we have $n$ points $\{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^r$
- The primary information that is available for those points is the measured Euclidean distances $d_{ij} \approx \|x_i - x_j\|$ among them
- Those measurements may or may not be accurate
- The problem is to recover the $n$ points in $\mathbb{R}^r$ purely based on those available distances

# CMDS-based formulation

- In cMDS, a $n \times n$ matrix $D$ is called a EDM if there exist points $p_1, \ldots, p_n$ in $\mathbb{R}^r$ such that $D_{ij} = \|p_i - p_j\|^2$ for $i, j = 1, \ldots, n$ (note: the distance is squared).
- A $n \times n$ symmetric matrix $D$ is EDM if and only if

$$\mathrm{diag}(D) = 0, J(-D)J \succeq 0 \text{ and } J := I - \frac{ee^\top}{n}$$

- The cMDS tries to solve

$$\min_Y \|Y - \frac{JDJ}{2}\|^2, \text{ s.t. } Y \in \mathcal{S}_+^n(r)$$

  $\mathcal{S}_+^n(r)$ denote the set of positive semidefinite matrices whose ranks are not greater than r
- The corresponding embedding dimension is $r = \mathrm{rank}(JDJ)$
- The $n$ embedding points $p_1, \ldots, p_n$ in $\mathbb{R}^r$ are given by the columns of $P^\top$ , where $P \in \mathbb{R}^{n \times r}$ satisfies

$$-JDJ/2 = PP^\top$$

# CMDS-based formulation

- When $D$ is not a true EDM, the cMDS is often not robust as the nearest distance from $D$ is measured through the transformation $JDJ$ rather than on $D$ itself.
- A more natural "nearness" measurement is the following:

$$\min_{Y} \|Y - D\|^2/2, \text{ s.t. } Y \in \mathcal{E}^n(r)$$

where $\mathcal{E}^n(r)$ is the set of EDMs with embedding dimensions not greater than $r$

- A more general model:

$$\min_{Y} \|H \circ (Y - D)\|^2/2, \text{ s.t. } Y \in \mathcal{E}^n(r)$$

where $H \in \mathcal{S}^n$ is the weight matrix ($H_{ij} \geq 0$) and $\circ$ is the Hadamard product

- Both problems are nonconvex, have no closed-form solutions and have to rely on iterative algorithms for their optimal solutions.

# Convex relaxations

- The nonconvexity is caused by a rank constraint since

$$Y \in \mathcal{E}^n(r) \Longleftrightarrow Y \in \mathcal{S}_h^n, J(-D)J \succeq 0 \text{ and } \operatorname{rank}(JDJ) \leq r$$

  where $\mathcal{S}_h^n = \{A \in \mathcal{S}^n : \operatorname{diag}(A) = 0\}$ is the *hollow* subspace in $\mathcal{S}^n$

- Ignoring the rank constraint leads to the convex relaxation:

$$\min_Y \|Y - D\|^2/2, \text{ s.t. } Y \in \mathcal{S}_h^n \text{ and } -JYJ \succeq 0$$

  where $\mathcal{E}^n(r)$ is the set of EDMs with embedding dimensions $\leq r$

- The feasible region is now actually the convex cone of all $n \times n$ EDMs, denoted by $\mathcal{E}^n$

- A serious drawback: (generalized) Slater condition does not hold because $JYJ$ has eigenvalue 0

- Two important convex reformulations emerged to rectify this

# Convex reformulation I

- One reformulation is based on the fact that there exists a one-to-one linear transformation $\mathcal{L} : \mathcal{S}_+^{n-1} \mapsto \mathcal{E}^n$
- The problem is equivalent to

$$\min_X \|\mathcal{L}(X) - D\|^2/2, \text{ s.t. } X \in \mathcal{S}_+^{n-1}$$

- This reformulation establishes an important link to SDP
- It was further studied as a prototype of convex quadratic SDPs, where a regularization term is added to encourage a low-rank solution

# Convex reformulation II

- The other reformulation is based on the fact that

$$-JYJ \succeq 0 \Longleftrightarrow -Y \in \mathcal{K}_+^n := \{A \in \mathcal{S}^n : x^\top Ax \geq 0, x \in e^\perp\}$$

  where $e^\perp := \{x \in \mathbb{R}^n : e^\top x = 0\}$

- The problem is equivalent to

$$\min_Y \|Y - D\|^2/2, \text{ s.t. } Y \in \mathcal{S}_h^n \text{ and } -Y \in \mathcal{K}_+^n$$

- A nice feature of this reformulation is that it can be treated as a projection problem onto the intersection of the subspace $S_n^h$ and the closed convex cone $-\mathcal{K}_+^n$

- Hence, the Method of Alternating Projections (MAP) of Dykstra-Han type is a choice

# Outline

# Matrix factorization

- Matrix Decompositions has a long history and generally centers around a set of known factorizations such as LU, QR, SVD and eigendecompositions
- More recent factorizations have seen the light of the day with constraints on the factors as in NMF, k-means and related algorithms
- Many sparse optimization problems can be formulated as matrix factorization

# Clustering

- An important method for data compression and classification is to organize data points in *clusters*
- A cluster is a subset of the set of data points that are close together in some distance measure
- Minimize distance within clusters, maximize distance between clusters

# K-means

We have n data points $(x_1, \ldots, x_n)$ and wish to partition them into $k$ disjoint clusters $C_1, \ldots, C_k$

1. Choose at random $k$ cluster centers (centroids) $\mathbf{m}_j, j = 1, \ldots, k$
2. While changes in clusters $C_j$ happen
   - form clusters: assign all points closest to $\mathbf{m}_j$ to the cluster $C_j$
   - compute new centroids: $\mathbf{m}_j$ = mean of all points in $C_j$

# K-means/K-median

- Matrix factorization form: $A = DX$ with unknown $D$ and $X$, solve for $XX^T = I$ and binary $X_{ij}$

$$\begin{aligned} \min_{\mathbf{D},\mathbf{X}} \quad & \|\mathbf{A} - \mathbf{DX}\|^2 \\ \text{s.t.} \quad & \mathbf{XX}^\top = \mathbf{I} \\ & \mathbf{X}_{ij} \in \{0, 1\} \end{aligned}$$

- K-means: Frobenius-norm minimization
  - if $\mathbf{D}^\top\mathbf{D} = \mathbf{I}$ while ignoring the binary constraint, we get SVD
- K-median: $\ell_1$-norm ($\sum |\cdot|$) minimization

# Spectral clustering

We have n data points $(x_1, \ldots, x_n)$ and wish to partition them into $k$ disjoint clusters $C_1, \ldots, C_k$

1. Form affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by

$$A_{ij} = \begin{cases} \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) & if\ i \neq j \\ 0 & if\ i = j \end{cases}$$

2. Define $D$ to be the diagonal matrix with $D_{ii} = \sum_j A_{ij}$, i.e. the sum of $A$'s $i$th row, and construct the matrix

$$L = D^{-1/2} A D^{-1/2}$$

3. Find the eigenvectors $v_j$ of $L$ corresponding to the $k$ largest eigenvalues, and form the matrix $V = [v_1 v_2 \ldots v_k] \in \mathbb{R}^{n \times k}$
4. Form the matrix $Y$ from $V$ by renormalizing each of $V$'s rows to have unit length
5. Treating each row of $Y$ as a point in $\mathbb{R}^k$, cluster them into $k$ clusters via k-means (or any other clustering algorithm)

# Spectral clustering

- Use methods from spectral graph partitioning to do clustering
- Needed: pairwise distances between data points
- Can be thought of as weights of links in a graph: clustering problem becomes a graph partitioning problem
- Unlike k-means, clusters need not be convex
- Matrix factorization form: $A = DX$ with unknown $D$ and $X$, solve for sparse $X$ and $X_i = 0$ or 1

# Subspace clustering

- Subspace clustering: self-representation of data

$$\begin{aligned} \min_{\mathbf{X},\mathbf{E}} \quad & \operatorname{rank}(\mathbf{X}) + \lambda \|\mathbf{E}\|_0 \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{A}\mathbf{X} + \mathbf{E} \end{aligned}$$

- Convex relaxation

$$\begin{aligned} \min_{\mathbf{X},\mathbf{E}} \quad & \|\mathbf{X}\|_* + \lambda \|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{A}\mathbf{X} + \mathbf{E} \end{aligned}$$

- Essentially: $A = AX$ with unknown $X$, solve for sparse/other conditions on $X$

# Graph matching

- $A = XBX^T$ with unknown X, B solve for X as a permutation

$$\min_X \quad \|A - XBX^T\|_F^2, \text{ s.t. } X \text{ is a permutation}$$

# Non-Negative matrix factorization (NMF)

- Non-negative matrix factorization (NMF): $A = DX$ with unknown $D$ and $X$, solve for elements of $D, X > 0$

$$\begin{aligned} \min_{\mathbf{D},\mathbf{X}} \quad & \|\mathbf{A} - \mathbf{DX}\|^2 \\ \text{s.t.} \quad & d_{ij} \geq 0 \\ & x_{ij} \geq 0 \end{aligned}$$

- Nonconvex problem; alternating minimization

# Matrix completion

- Matrix completion (MC) as matrix factorization: $A = M \circ X$ with a known mask $M$; solve for $X$ with lowest rank possible

$$\begin{aligned} \min_{\mathbf{X}} \quad & \operatorname{rank}(\mathbf{X}) \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{M} \circ \mathbf{X} \end{aligned}$$

- Convex relaxation

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{X}\|_* \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{M} \circ \mathbf{X} \end{aligned}$$

# Robust PCA

- Robust principal component analysis (RPCA)

$$\begin{aligned} \min_{\mathbf{X},\mathbf{E}} \quad & \operatorname{rank}(\mathbf{X}) + \lambda \|\mathbf{E}\|_0 \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{X} + \mathbf{E} \end{aligned}$$

- Matrix factorization view: $A = X + E$ with a low rank component $X$ and a sparse component $E$
- Convex relaxation

$$\begin{aligned} \min_{\mathbf{X},\mathbf{E}} \quad & \|\mathbf{X}\|_* + \lambda \|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{X} + \mathbf{E} \end{aligned}$$

# Stable Principle Component Pursuit (SPCP)/ Noisy Robust PCA

- RPCA is limited to the low-rank component being exactly low-rank and the sparse component being exactly sparse
- In real world applications the observations are often corrupted by noise
- Introducing entry-wise noise $N$: $A = X + E + N$ with $X, E, N$ unknown, solve for $X$ low rank, $E$ sparse
- Optimization problem

$$\begin{aligned} \min_{\mathbf{X},\mathbf{E}} \quad & \|\mathbf{X}\|_* + \lambda \|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \|\mathbf{A} - \mathbf{X} + \mathbf{E}\|_F \leq \delta \end{aligned}$$

# Sparse PCA

- $A = DX$ with unknown $D$ and $X$, solve for sparse $D$
- Sparsity in number of basis vectors
- Optimization form

$$\begin{aligned} \min_{\mathbf{D},\mathbf{X}} \quad & \|\mathbf{A} - \mathbf{DX}\|_F^2 + \lambda \|\mathbf{D}\|_1 \\ \text{s.t.} \quad & \|\mathbf{x}_i\|_2 \leq 1 \end{aligned}$$

# Dictionary learning

- $A = DX$ with unknown $D$ and $X$, solve for sparse $X$
- Sparsity in representation coefficients
- Optimization form

$$\begin{aligned} \min_{\mathbf{D},\mathbf{X}} \quad & \|\mathbf{A} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_1 \\ \text{s.t.} \quad & \|\mathbf{d}_i\|_2 \leq 1 \end{aligned}$$

# Matrix Compressive Sensing (MCS)

- Find a rank-r matrix L such that $A(L) = b$ or $A(L+S) = b$
- Optimization form

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{X}\|_* \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{X}) = \mathbf{b} \end{aligned}$$

# Other applications

- Y = A X with unknown X and rows of X are sparse
- Compressive sensing is a special case of MMV called Single Measurement Vector (SMV)
- Y = A X with unknown X and rows of X are sparse, X is one column
- Y = A X with unknown A and X and statistical independence between columns of X or subspaces of columns of X

# Summary

- Matrix factorization: $\mathbf{A} = \mathbf{DX}$
- Utilizing different structural properties on $\mathbf{D}$ and/or $\mathbf{X}$
  - Low-rank factorization: $\mathbf{D}$ and $\mathbf{X}$ have few columns/rows
  - Dictionary Learning / Sparse PCA: either $\mathbf{D}$ or $\mathbf{X}$ has few non-zeros
  - Clustering: sparse binary $\mathbf{X}$
  - nonnegative matrix factorization: element-wise positivity
  - ...
- Algorithms
  - Convex relaxation: (A)PG, ALM, ADMM
  - Jointly nonconvex problems: alternating minimization, BCD
  - ...