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

Eckart & Young, 1936

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

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

Outline

Dimensionality Reduction

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

PCA

- Assume $p < N$. For a collection of data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in R^p$, we want to represent them by a rank q linear model

$$\mathbf{u}(\mu, \lambda, \mathbf{M}) = \mu + \mathbf{M}\mathbf{y}, \mathbf{M} \in R^{p \times q}, \forall q \leq p,$$

  where μ is local vector and $\mathbf{M}^\top \mathbf{M} = I$.

- We write $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in R^{p \times N}$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in R^{q \times N}$. The new representation of the data is:

$$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N] \in R^{p \times N}, \quad \mathbf{u}_i = \mu + \mathbf{M}\mathbf{y}_i.$$

- Consider the model

$$\min_{\mu, \mathbf{y}_i, \mathbf{M}} F(\mu, \mathbf{y}_i, \mathbf{M}) = \sum_{i=1}^{N} \|\mathbf{x}_i - \mu - \mathbf{M}\mathbf{y}_i\|_2^2, \text{ s.t. } \mathbf{M}^\top \mathbf{M} = I.$$

PCA

- Let $\bar{\mathbf{x}} = \frac{1}{N}\sum_i \mathbf{x}_i$ and $\bar{\mathbf{y}} = \frac{1}{N}\sum_i \mathbf{y}_i$.
- $\frac{\partial F}{\partial \mu} = 0 \Longrightarrow \mu = \bar{\mathbf{x}} - \mathbf{M}\bar{\mathbf{y}} \qquad (1)$
- $\frac{\partial F}{\partial \mathbf{y}_i} = 0 \Longrightarrow \mathbf{y}_i = \mathbf{M}^\top(\mathbf{x}_i - \mu) \Longrightarrow \bar{\mathbf{y}} = \mathbf{M}^\top(\bar{\mathbf{x}} - \mu) \qquad (2)$
- Hence, we must have $\mu = \bar{\mathbf{x}}$ and have $\mathbf{y}_i = \mathbf{M}^\top(\mathbf{x}_i - \bar{\mathbf{x}})$.
- After partially optimizing $\mathbf{y}$ and μ, the origianl problem reduce to

$$\min_{\mathbf{M}} F(\mathbf{M}) = \sum_{i=1}^{N} \|(\mathbf{x}_i - \bar{\mathbf{x}}) - \mathbf{M}\mathbf{M}^\top(\mathbf{x}_i - \bar{\mathbf{x}})\|_2^2, \text{ s.t. } \mathbf{M}^\top\mathbf{M} = I.$$

PCA

- Let $\bar{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ and $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_N]$. In fact,

$$\bar{\mathbf{X}} = \mathbf{X} - \frac{1}{N}\mathbf{X}\mathbf{1}\mathbf{1}^\top.$$

Then, we have

$$\begin{aligned} F(\mathbf{M}) &= \sum_{i=1}^{N} \|\bar{\mathbf{x}}_i - \mathbf{M}\mathbf{M}^\top \bar{\mathbf{x}}_i\|_2^2 \\ &= \|\bar{\mathbf{X}} - \mathbf{M}\mathbf{M}^\top \bar{\mathbf{X}}\|_F^2 \\ &= \mathrm{Tr}(\bar{\mathbf{X}}^\top \bar{\mathbf{X}}) - \mathrm{Tr}(\bar{\mathbf{X}}^\top \mathbf{M}\mathbf{M}^\top \bar{\mathbf{X}}) \\ &= \mathrm{Tr}(\bar{\mathbf{X}}\bar{\mathbf{X}}^\top) - \mathrm{Tr}(\mathbf{M}^\top \bar{\mathbf{X}}\bar{\mathbf{X}}^\top \mathbf{M}). \end{aligned}$$

- Let $\mathrm{cov}(\mathbf{X}) = \bar{\mathbf{X}}\bar{\mathbf{X}}^\top$. Thus the problem is equivalent to

$$\max_{\mathbf{M} \in R^{p \times q}} \mathrm{Tr}(\mathbf{M}^\top \mathrm{cov}(\mathbf{X})\mathbf{M}), \text{ s.t. } \mathbf{M}^\top \mathbf{M} = I.$$

PCA

- PCA constructs a low-dimensional representation of the data that describes as much of the variance in the data as possible.
- Let $\mathbf{x} \in \mathbb{R}^p$ be a random vector with mean μ and covariance Σ.
- Find $y = z^T\mathbf{x}$ such that the variance of y is maximized.

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

- Model problem:

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

Find the maximal eigenpair (λ_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 $\bar{\mathbf{X}} = \mathbf{X} - \frac{1}{N}\mathbf{X}11^\top$, where 1 is a column vector of all ones
- SVD: $\bar{\mathbf{X}} = U\Sigma V^T$, $\Sigma = [\text{diag}(\sigma_1, \ldots, \sigma_p), 0]$, $U^TU = I$, $V^TV = I$.
  Eigen: $\bar{\mathbf{X}}\bar{\mathbf{X}}^\top = U\Lambda U^\top$, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p)$, and $\lambda_i = \sigma_i^2$
- PCA computes

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

  Then the optimal solution $\mathbf{M} = U_q$.
- Therefore, we have: $\mathbf{y}_i = \mu + \mathbf{M}\mathbf{M}^\top(\mathbf{x}_i - \bar{\mathbf{x}})$, i.e.,

$$\mathbf{Y} = \mu 1^\top + \mathbf{M}\mathbf{M}^\top\bar{\mathbf{X}} = \mu 1^\top + \mathbf{M}\Sigma_q V_q^\top.$$

- The matrix $\mathbf{M}^\top\bar{\mathbf{X}} = \Sigma_q V_q^\top \in R^{q\times N}$ is called the first q principal components of $\bar{\mathbf{X}}$.

Classical MDS

- Known data points $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, N$.
  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 $\mathbf{y}_i \in \mathbb{R}^q$ 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 $\mathbf{X} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_N\right)$, $H = I_N - \frac{1}{N}11^T$. Let $B = -\frac{1}{2}HD_2(\mathbf{X})H$ and $\bar{\mathbf{X}} = \mathbf{X} - \frac{1}{N}\mathbf{X}11^T$. Then we have $B = \bar{\mathbf{X}}^\top\bar{\mathbf{X}}$.

Classical MDS

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

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

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

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

Thus

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

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

Classical MDS

- Let $B = \bar{\mathbf{X}}^\top \bar{\mathbf{X}}$ and SVD: $\bar{\mathbf{X}} = U\Sigma V^\top$, $\Sigma = [\mathrm{diag}(\sigma_1, \ldots, \sigma_p), 0]$. The eigenvalue decomposition: $B = V\Lambda V^\top$, $\lambda_i = \sigma_i^2$.
- Assume that $\mathbf{Y}$ is centered, i.e., $-\frac{1}{2}HD_2(Y)H = Y^\top Y$, then Classical MDS is equivalent to

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

- The optimal solution $\mathbf{Y} = \Lambda^{\frac{1}{2}} V^\top$
- Classical MDS: take $\tilde{\mathbf{X}} = \Lambda_q^{\frac{1}{2}} V_q^\top = \Sigma_q V_q^\top$
- PCA and classical MDS are equivalent!

Extension of MDS

- Known data points $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, N$.
  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 $\mathbf{y}_i \in \mathbb{R}^q$ such that

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

  or with centering matrix $H = I_N - \frac{1}{N} 1 1^T$ 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:

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

- 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 ith element of the αth eigenvector, with eigenvalue λ_α
- 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 αth 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 λ_{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 ν 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 λ_{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 λ_{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

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

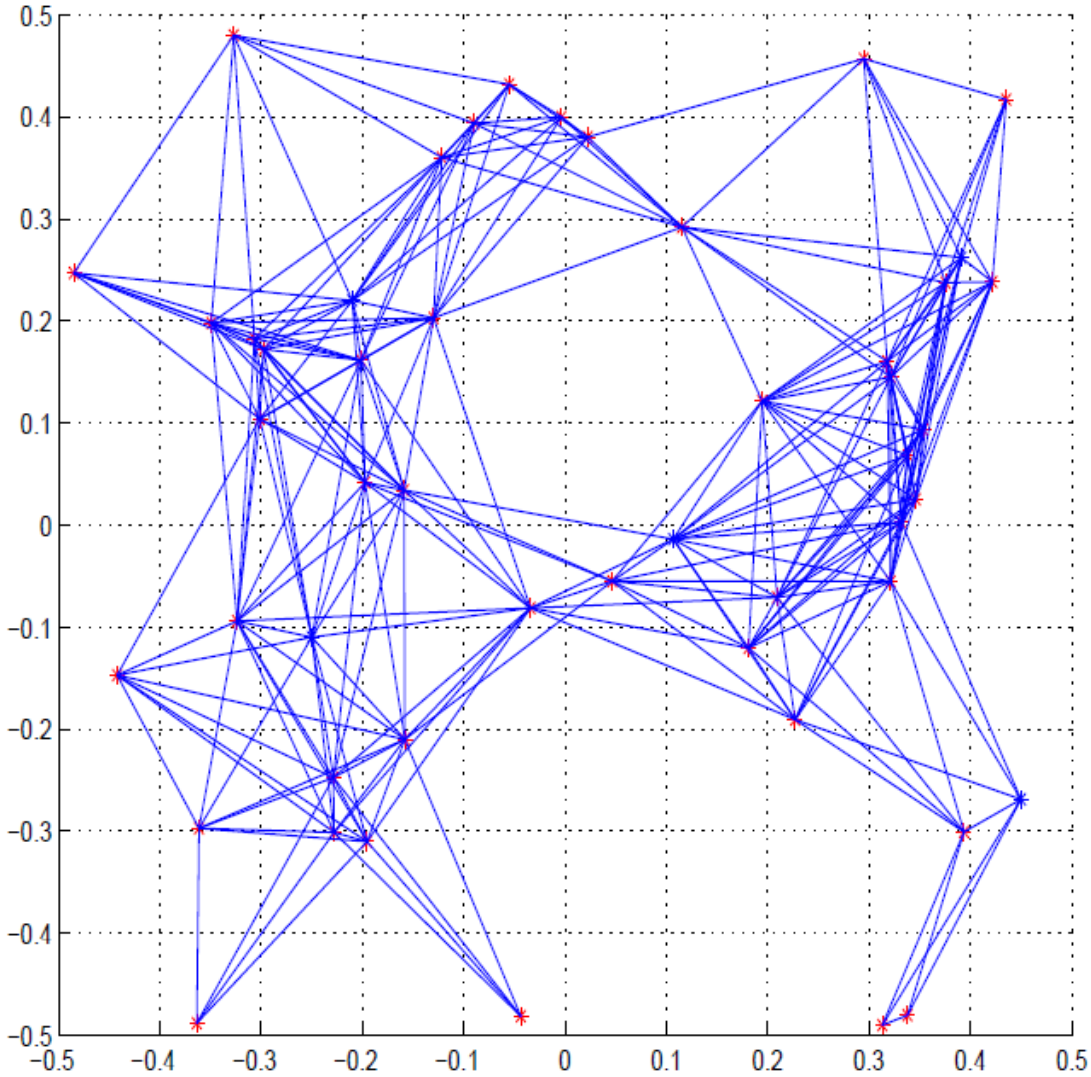


Figure: 50-Sensor Network with Radio Range .3

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_js?
- 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 jth 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

Transformer

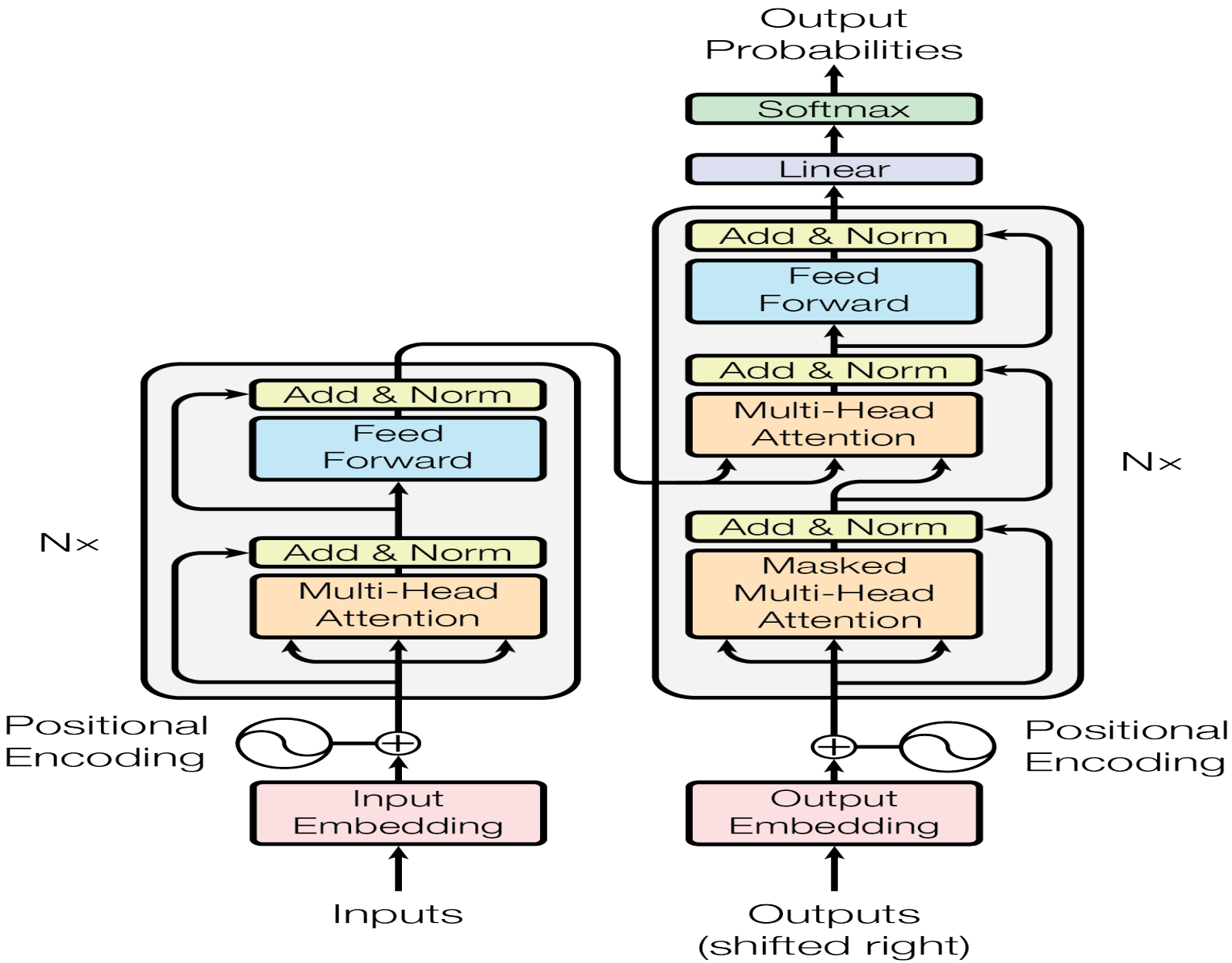
- Transformer架构具有编码器和解码器两个部分. 并且用到的自注意力相关结构大体上可分为三个部分：输入多头自注意力，输出多头自注意力及输入-输出多头自注意力.
- 设编码器部分的输出为$X^{[1]}$，解码器部分中经过掩盖多头自注意力机制处理的部分为$Y^{[1]}$，则在输入-输出多头自注意力机制结构中，自注意力机制的结构变为

$$Q = X^{[1]}W_Q, K = X^{[1]}W_K, V = Y^{[1]}W_V, \tag{1}$$

即查询矩阵Q与键矩阵K的信息是来自于$X^{[1]}$，值矩阵来自于$Y^{[1]}$.

- 在图中，$N\times$表示这个模块进行N次，并且在自注意力输出部分往往需要做与输入相加并且层归一化的操作. 最后，在训练好Transformer模型后，我们输入起始信息Y_P^0，得到输出Y_P^1，依此进行下去，直到得到终止信号，这样就完成了整个输出过程.

Transformer



Why PETuning and LoRA?

When we want to develop large models, like language model of GPT, we need to do **large scale pre-training**.
Problems and observations:

- It is very time-consuming to full fine-tuning the model by calculate gradient for each parameter when training.
- Experiments shows that even if we do full fine-tuning, the result of fine-tuning only have significant impact on a low rank subspace of the parameter space.

$\Rightarrow$ Only fine-tuning on a low rank subspace.

Such kind of method is called **Parameter-Efficient Tuning (PETuning)**.

LoRA: A typical way

- **LoRA**: a kind of PETuning.

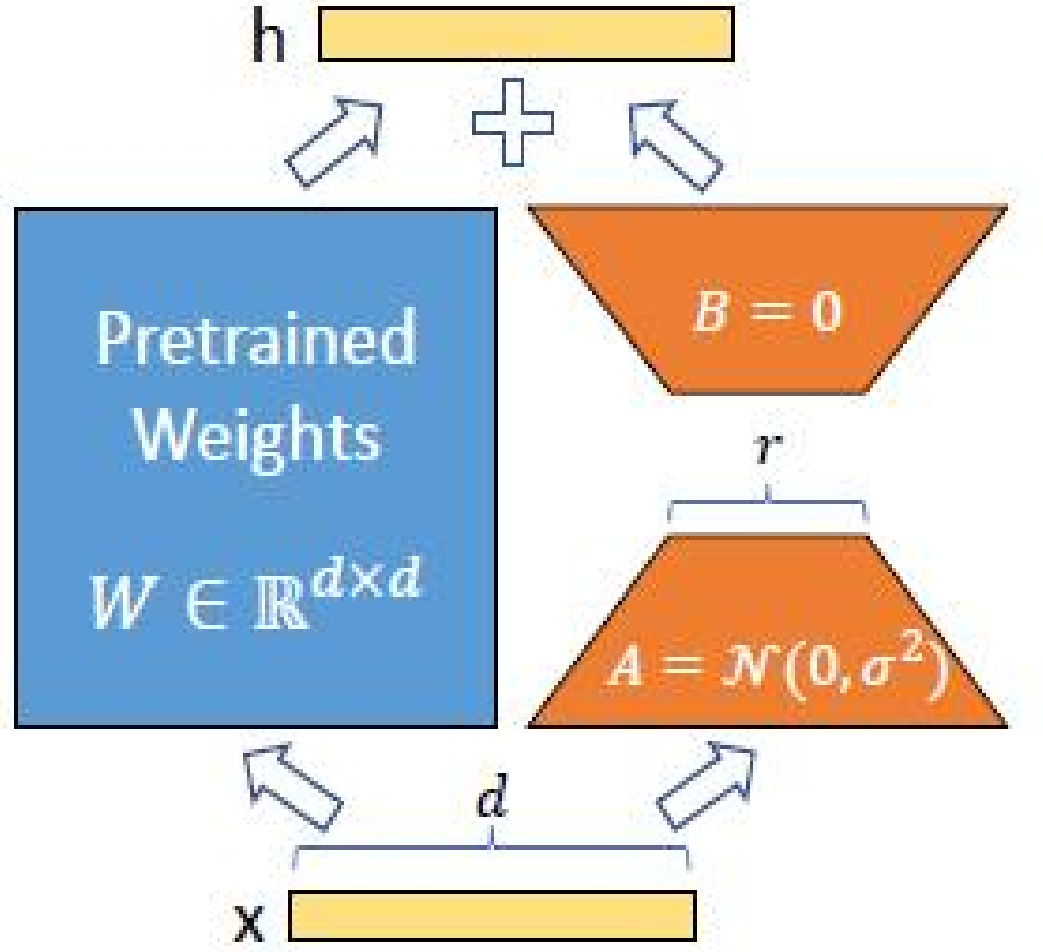
Basic form:
Suppose W_0 is a pre-trained weight matrix, the traditional way is to update it by $W_0 + \Delta W$, where $\Delta W \in \mathbb{R}^{d \times k}$ and can be trained. In LoRA, we update it by $W_0 + BA$, where $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$, and the rank $r \ll \min(d, k)$. A and B are trainable.

- Reduce the amount of trainable parameter to $r \times (d + k)$

Initialization of training:

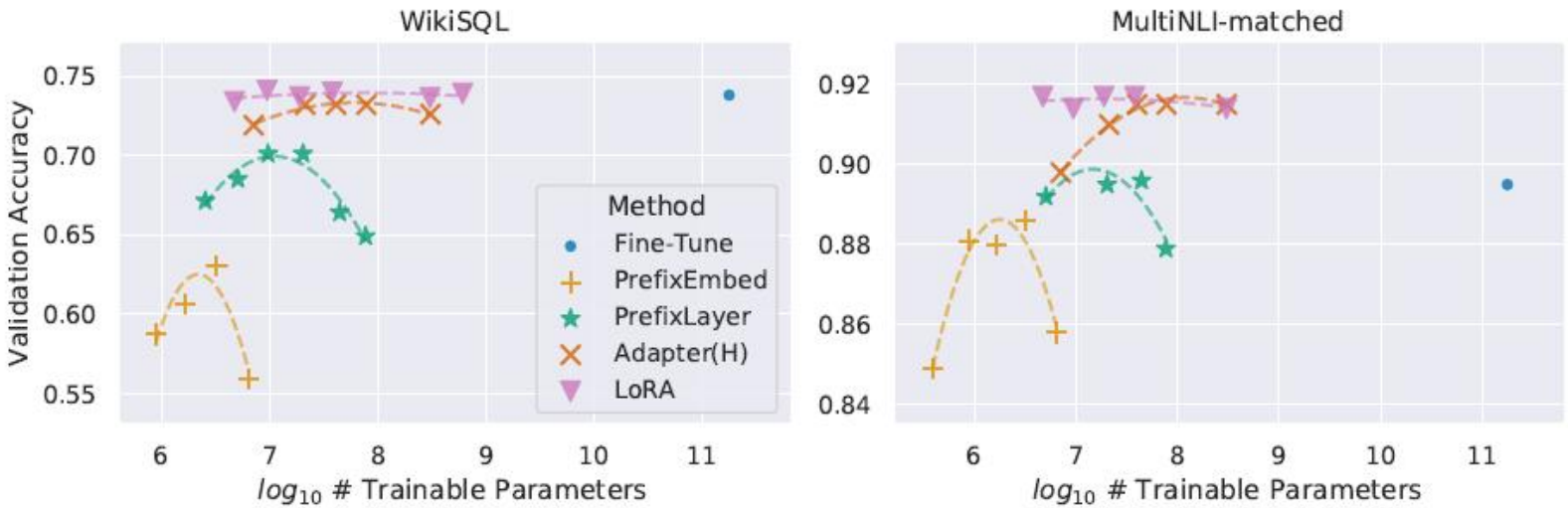
- A: Random Gaussian
- B: Zero matrix

LoRA: A typical method



LoRA: Benifits

Empirical Result:



- Reduction in memory and storage.
- higher efficiency.
- Switch between task at lower cost.
- higher interpretability

Limitations and Improvement of LoRA

Limitations:

- Not able to batch inputs to different tasks with different A and B in a single forward pass
- The choice of initialization of A and B: Gaussian and zero may not be good.
- The choice of r: we do not know which r is the best.
- Theoretical explanation of why the method works.

Improvements:

- PST: to further reduce the calculation, we adapt sparsity method to LoRA - In each iteration we use sparsity approximation to the matrix W.
- PVP: if $D' \subset D$ is the dataset for a specific problem, we can pre-train A and B with datas in D.

Outline

K-均值聚类

聚类(clustering)不同于分类(classification)，在聚类问题中我们仅仅知道数据点本身，而不知道每个数据点具体的标签。聚类分析的任务就是将一些无标签的数据点按照某种相似度来进行归类，进而从数据点本身来学习其内蕴的类别特征。

给定p 维空间中n 个数据点$a_1, a_2, \cdots, a_n$，聚类问题就是要寻找k个不相交的非空集合$S_1, S_2, \cdots, S_k$，使得

$$\{a_1, a_2, \cdots, a_n\} = S_1 \cup S_2 \cup \cdots \cup S_k,$$

并且使得组内距离平方和最小，即

$$\begin{aligned} \min_{S_1, S_2, \cdots, S_k} \quad & \sum_{i=1}^{k} \sum_{a \in S_i} \|a - c_i\|^2, \\ \text{s.t.} \quad & S_1 \cup S_2 \cup \cdots \cup S_k = \{a_1, a_2, \cdots, a_n\}, \\ & S_i \cap S_j = \emptyset, \quad \forall i \neq j, \end{aligned} \tag{2}$$

K-均值聚类的等价表述一

为了表示聚类方式$S_1, S_2, \cdots, S_k$，一个很自然的想法是使用一个向量$\phi_i \in \mathbb{R}^k$来表示点a_i所处的类别．具体地，定义ϕ_i为

$$(\phi_i)_j = \begin{cases} 1, & a_i \in S_j, \\ 0, & a_i \notin S_j, \end{cases}$$

则聚类问题可以等价描述为

$$\begin{aligned} \min_{\Phi, H} \quad & \|A - \Phi H\|_F^2, \\ \text{s.t.} \quad & \Phi \in \mathbb{R}^{n\times k}, \text{每一行只有一个元素为1，其余为0}, \\ & H \in \mathbb{R}^{k\times p}. \end{aligned} \tag{3}$$

在这里Φ的第i行的向量就是ϕ_i^{T}.

K-均值聚类的等价表述二

首先定义$\mathbf{1}_{S_t}$, $1 \le t \le k$ 为n 维空间中每个分量取值0或1 的向量，且

$$\mathbf{1}_{S_t}(i) = \begin{cases} 1, & a_i \in S_t, \\ 0, & a_i \notin S_t. \end{cases}$$

可以证明，第t 类S_t 中每个点到其中心点的距离平方和可以写成$\frac{1}{2n_t}\operatorname{tr}(D\mathbf{1}_{S_t}\mathbf{1}_{S_t}^{\mathrm{T}})$，其中$D \in \mathbb{R}^{n\times n}$ 的元素为$D_{ij} = \|a_i - a_j\|^2$.
因此，我们将问题(2) 转化为

$$\begin{aligned} \min_{S_1,S_2,\cdots,S_k} \quad & \frac{1}{2}\operatorname{tr}(DX), \\ \text{s.t.} \quad & X = \sum_{t=1}^{k} \frac{1}{n_t}\mathbf{1}_{S_t}\mathbf{1}_{S_t}^{\mathrm{T}}, \\ & S_1 \cup S_2 \cup \cdots \cup S_k = \{a_1, a_2, \cdots, a_n\}, \\ & S_i \cap S_j = \emptyset, \quad \forall i \ne j. \end{aligned} \tag{4}$$

K-均值聚类的等价表述二

对半定矩阵X进行分解$X = YY^{\mathrm{T}}, Y \in \mathbb{R}^{n\times k}$，我们可以进一步得到如下矩阵优化问题（这里$\mathbf{1}$ 是n维向量且分量全为1）：

$$\begin{aligned} \min_{Y\in\mathbb{R}^{n\times k}} \quad & \mathrm{tr}(Y^{\mathrm{T}}DY), \\ \text{s.t.} \quad & YY^{\mathrm{T}}\mathbf{1} = \mathbf{1}, \\ & Y^{\mathrm{T}}Y = I_k,\ Y \geq 0. \end{aligned} \tag{5}$$

ONMF

- $S_+^{n,k} := \{X \in \mathbb{R}^{n\times k} : X^\top X = I_k,\ X \geq 0\}$
- Orthogonal NMF (ONMF), Ding, Li, Peng & Park (2006)
  - Data matrix $A \in \mathbb{R}_+^{n\times r}$, n data samples, each with r features, k clusters
  - $A \approx XY^\top, X \in S_+^{n,k}, Y \in \mathbb{R}_+^{r\times k}$
  - $X_{ij} \begin{cases} > 0, & \text{if sample } i \in \text{cluster } j \\ = 0, & \text{otherwise} \end{cases}$
  - ONMF model

$$\min_{X\in S_+^{n,k}, Y\in\mathbb{R}_+^{r\times k}} \|A - XY^\top\|_F^2 \tag{6}$$

- Orthonormal projective NMF (OPNMF) model, Yang & Oja (2010)

$$\min_{X\in S_+^{n,k}} \|A - XX^\top A\|_F^2 \tag{7}$$

- K-indicators model, Chen, Yang, Xu, Zhang & Zhang (2019)

$$\min_{X\in S_+^{n,k}, Y^\top Y=I} \|UY - X\|_F^2 \quad \text{s.t.} \quad \|X_{i,:}\|_0 = 1, i \in [n], \tag{8}$$

where U is the features matrix extracted from the data matrix A.

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

Non-Negative matrix factorization (NMF)

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

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

- Nonconvex problem; alternating minimization

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

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

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
  - ...