Unsupervised Learning

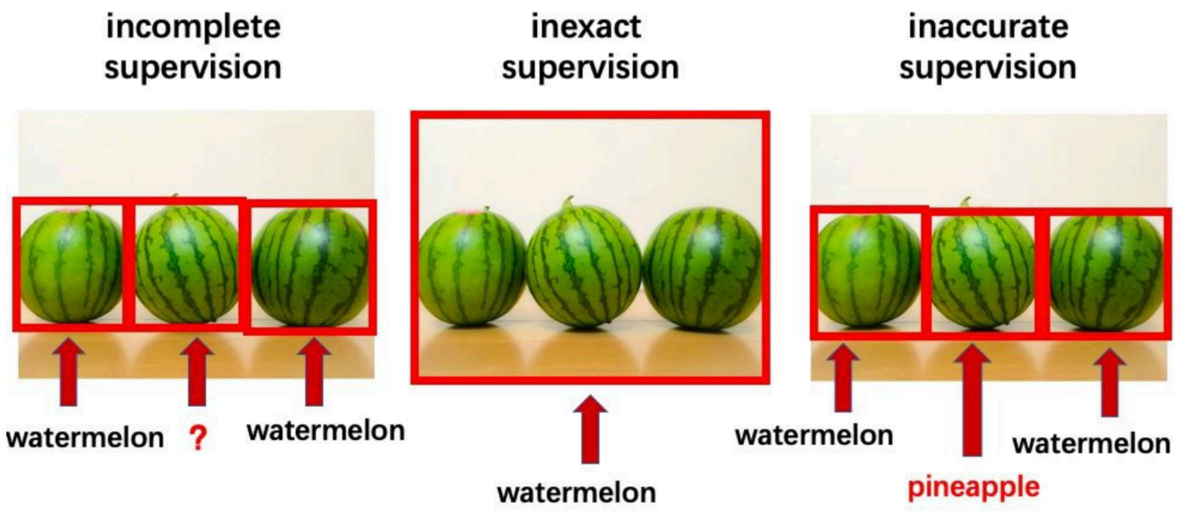
Jiachun Jin
jinjch@shanghaitech.edu.cn

Overview of unsupervised learning

> **"Machine Learning is far from fitting something."**
>
> —— 鲁迅

From supervised, weakly supervised to unsupervised

- Supervised learning: easy, just fit some functions, take the generalization into consideration
  - Regression
  - Classification
- Weakly supervised learning: label is expensive, a modern research topic in the ML community, a lot new learning settings, **nice for your course projects**



  - Incomplete supervision (不完全监督): 一个数据集，有的样本有标注，有的没有
    - Semi-supervised learning
      - Inductive
      - Transductive
    - Active learning: 算法可以挑一些没有标注的样本让标注员去标一下(有限的标注预算)
  - Inexact supervision (不确切监督): 标注比较粗糙，不精细
    - Multi-instance learning: 视频标注
    - Partial label learning, the label of x_i is a set $S_i \subseteq \{y_1, \dots, y_k\}$
    - Confused multi-task learning(ICML 2020) [1] :样本点来源于不同的回归任务，但是不知道每一个点具体属于哪一个任务
  - Inaccurate supervision (不精确监督)
    - Learning with noise label: 有标签标错了
  - 还有一些别的弱监督学习的设定
    - Positive unlabeled learning: 只有正标签，用户只告诉你他喜欢什么
    - One-bit supervision learning(NIPS 2020) [2] :
      - $\mathcal{D} = \mathcal{D}^S \cup \mathcal{D}^O \cup \mathcal{D}^U$, where $\mathcal{D}^S$ denote common supervised dataset, $\mathcal{D}^U$ denote the unsupervised dataset, while $\mathcal{D}^O$ denote a fixed set(here different from active learning) for **one-bit supervision**
      - One-bit supervision: labeler tell whether the image belongs to the specific label, only allowed once(once labeled y_n^-, no further supervision can be obtained)
    - etc
- Unsupervised learning: difficult
  - Goal of unsupervised learning: to discover “interesting structure” in the data, **knowledge discovery**
  - In probabilistic view, supervised learning we build models of the form $p(y_i|\mathbf{x}_i, \theta)$, while in unsupervised learing, we build models of form $p(\mathbf{x}_i|\theta)$, known as "**density estimation**"
    - Gaussian Mixture Model

$$p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

- Probabilistic PCA

$$p(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}\left(\mathbf{x} \mid \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}\right)$$

Self-supervised learning

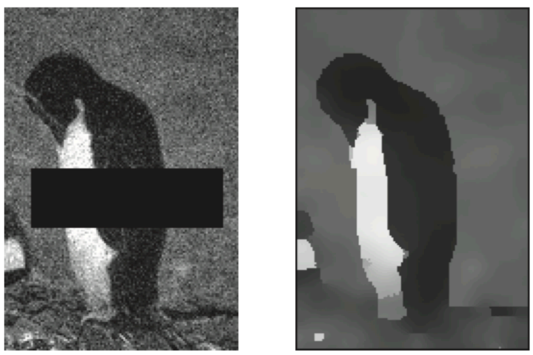
- Create proxy supervised tasks from unlabeled data
  - 把一段文字中间扣掉几个词，让语言模型去预测扣掉的词是什么(BERT)
  - 把一些没有标注的图片随机旋转一下，让模型去预测旋转的角度

Clustering

- First goal, estimate the distribution over the number of clusters K, $p(K|\mathcal{D})$, $\hat{K} = \arg\max_K p(K|\mathcal{D})$
- Second goal, estimate which cluster each point belongs to, z_i: latent varible(never observed) denotes which cluster point x_i belongs to, $\hat{z}_i = \arg\max_k p(z_i = k|\mathbf{x_i}, \mathcal{D})$

Matrix completion

- Image inpainting



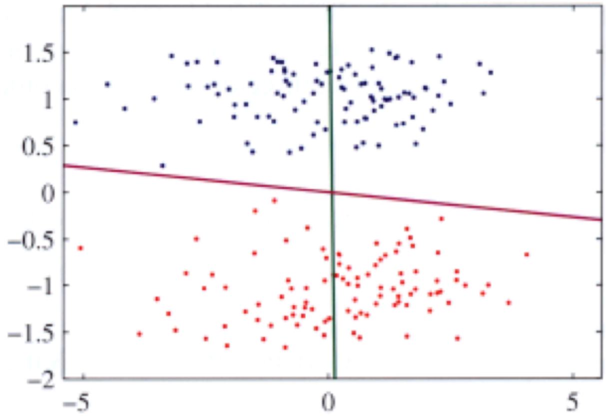
(a) (b)

- The Netflix competition
  - 1 million USD prize
  - 18,000 movies, 500,000 users, sparse ratings from 1 to 5
- Low-rankness assumption
- Active matrix completion

Dimension reduction???

- We often take this as unsupervised learning, but there are also supervised dimension reduction algorithms, e.g. LDA
- Difference between LDA and PCA

- Data from 2 classes
- Red line: PCA choose the direction of maximum variance
- Green line: LDA takes account of the data labels, it choose the green line to give a good class separation



Spectral Clustering[3]

Problem formulation

- $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{D\times N}$: data matrix
-
  - We can construct a similarity matrix of all the data points, $W = W^\top \in \mathbb{R}^{N\times N}$
    - KNN graph
    - $\epsilon-$neighbourhood graph
    - fully connected graph: $s_(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/(2\sigma^2))$
- Construct a graph for all the data points: $G = (V, E, W)$
  - V: vertices, in this case each data point is a vertex
  - E: edges between 2 vertices
  - W: weighted adjacency matrix, w_{ij} denotes the weight of the vertex between v_i and v_j
    - nonnegative
    - symmetric
  - Take the clustering problem as a graph cut problem

Graph Laplacian

- D: degree matrix, diagonal, $D_{ii} = \sum_{j=1}^N w_{ij}$
- Graph Laplacian matrix: $L = D - W$

> **Properties of graph Laplacian matrix**
>
> 1. $\forall f \in \mathbb{R}^N, f^\top L f = \frac{1}{2}\sum_{i,j=1}^N w_{ij}(f_i - f_j)^2$

Proof:

$$
\begin{aligned}
f^\top L f &= f^\top D f - f^\top W f \\
&= \sum_{i=1}^N D_{ii} f_i^2 - \sum_{i,j=1}^N w_{ij} f_i f_j \\
&= \frac{1}{2}\left[\sum_{i=1}^N D_{ii} f_i^2 - 2\sum_{i,j=1}^N w_{ij} f_i f_j + \sum_{j=1}^N D_{jj} f_j^2\right] \\
&= \frac{1}{2}\left[\sum_{i=1}^N (\sum_{j=1}^N w_{ij}) f_i^2 - 2\sum_{i,j=1}^N w_{ij} f_i f_j + \sum_{j=1}^N (\sum_{i=1}^N w_{ji}) f_j^2\right] \\
&= \frac{1}{2}\sum_{i,j=1}^N w_{ij}(f_i - f_j)^2
\end{aligned}
$$

2. L is symmetric and positive semi-definite

   Obvious by property 1.

3. L's smallest eigenvalue is 0, the eigenvector is the all one vector $\mathbf{1}$.

   Obvious by property 2.

The null space of L

G: an undirected graph, with non-negative weights

- $A_1, \ldots, A_k$: the connected components of G
- $\mathbf{1}_{A_1}, \mathbf{1}_{A_2}, \ldots, \mathbf{1}_{A_k}$: indicator vector of $A_1, \ldots, A_k$

Proposition

The null space of L has dimension k(the same as the number of connected conponents of G), and is spanned by $\{\mathbf{1}_{A_1}, \mathbf{1}_{A_2}, \ldots, \mathbf{1}_{A_k}\}$.

Proof:

We first prove $f^\top L f = 0 \Rightarrow f \in \mathcal{N}(L)$:

Since L is positive semi-definite, then we can write $L = BB^\top$, $f^\top L f = f^\top BB^\top f = \|B^\top f\|_2^2 \geq 0$, $x^\top L x = 0 \Rightarrow f \in \mathcal{N}(B^\top) \Rightarrow f \in \mathcal{N}(L)$.

This gives us that $f \in \mathcal{N}(L) \Leftrightarrow f^\top L f = 0 \Leftrightarrow \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2 = 0 \Leftrightarrow f_i = f_j, \forall x_i, x_j$ stays in the same connected component $A_s, \forall s = 1, 2, \ldots, k$.

> 说人话：找一根N维的向量f_{A_1}，每一个元素代表一个样本点（图G上的一个顶点），f_{A_1}中如果点x_i属于A_1这个connected component，那么f_{A_1}的第i个元素为1，不然为0。注意到，只有$f_{A_1}, f_{A_2}, \ldots, f_{A_k}$能使得$f^\top Lf = 0 \Leftrightarrow \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2 = 0$。因此得证。

所以我们只要：

1. 根据数据矩阵X构造出他的graph Laplacian L
2. 找出$\mathcal{N}(L)$的一组basis(eigen decomposition)

就能知道：

1. 数据中的cluster个数($\dim \mathcal{N}(L)$)
2. 每个数据点所属的cluster
    - naive!

Assign points to clusters

Compute the basis of $\mathcal{N}(L)$ with not produce $[\mathbf{1}_{A_1}, \mathbf{1}_{A_2}, \ldots, \mathbf{1}_{A_k}]$, denote the computed bases as $B = [b_1, b_2, \ldots, b_k]$, then $B = [\mathbf{1}_{A_1}, \mathbf{1}_{A_2}, \ldots, \mathbf{1}_{A_k}]\,\Theta$, where Θ is an invertible $k \times k$ matrix.

Let's transpose it: $Y = B^\top = \Theta^\top \begin{bmatrix} \mathbf{1}_{A_1}^\top \\ \vdots \\ \mathbf{1}_{A_k} \end{bmatrix} \in \mathbb{R}^{k \times N}$, and now each column of Y denotes a point, and x_i, x_j lies in the same cluster if and only if $Y_i = Y_j$.

- We call Y_i as an embedding of x_i
- The same embedding, stay in the same cluster

Proof:

1. x_i, x_j stay in the same cluster $\Rightarrow Y_i = Y_j$:

    $Y = \Theta^\top \begin{bmatrix} \mathbf{1}_{A_1}^\top \\ \vdots \\ \mathbf{1}_{A_k} \end{bmatrix}$, so $Y_i = \Theta^\top i^{th}$ column of $\begin{bmatrix} \mathbf{1}_{A_1}^\top \\ \vdots \\ \mathbf{1}_{A_k} \end{bmatrix}$, and i^{th} column of $\begin{bmatrix} \mathbf{1}_{A_1}^\top \\ \vdots \\ \mathbf{1}_{A_k} \end{bmatrix}$ is the same as the j^{th} column of $\begin{bmatrix} \mathbf{1}_{A_1}^\top \\ \vdots \\ \mathbf{1}_{A_k} \end{bmatrix}$

    denote $E = \begin{bmatrix} \mathbf{1}_{A_1}^\top \\ \vdots \\ \mathbf{1}_{A_k} \end{bmatrix}$, x_i, x_j stay in the same cluster $\Rightarrow E_i = E_j$ (E的每一列都只有一个元素是1，其余都是0) $\Rightarrow Y_i = Y_j$

2. $Y_i = Y_j \Rightarrow x_i, x_j$ stay in the same cluster:

   Suppose $Y_i = Y_j$ but x_i, x_j stay in different clusters, then we have $\Theta^\top E_i = \Theta^\top E_j$ and $E_i, E_j \neq 0$, so $\Theta^\top \neq 0$, contradiction.

> In practice, all the embeddings may not be exactly the same, we run K-means to figure out the clusters.

More on PCA

The view of subspace projection

> **Prerequisites**
>
> There is a d-dimensional subspace S stay in $\mathbb{R}^D$, Use $U = [u_1, u_2, \dots, u_d]$ to denote a set of basis of S, then the project matrix onto S is: $U(U^\top U)^{-1}U^\top$
>
> - This means $\forall x \in \mathbb{R}^D, U(U^\top U)^{-1}U^\top x$ lies in the subspace S, note that once U is orthogonal, the project matrix becomes $UU^\top$ since $U^\top U = I_d$
>
> - If you are not familiar with this, read the Appendix 1 for a more detailed explanation.
> - $AA^\dagger$ projects a vector onto $\mathcal{R}(A)$, $A^\dagger A$ projects a vector onto $\mathcal{R}(A^\top)$.

$X = [x_1, \dots, x_n] \in \mathbb{R}^{D\times n}$, suppose it has been centerized

We want to find a d-dimensional subspace in $\mathbb{R}^D$ (a set of its orthogonal basis) such that the projection of the data onto this subspace is most close to the original data X:

$$\min_{U\in\mathbb{R}^{D\times d}, U^\top U = I_d} \|X - UU^\top X\|_F^2 \tag{1}$$

Let's do some derivation:

note that $Trace(ABC) = Trace(CAB) = Trace(BCA)$

$$\begin{aligned}
\|X - UU^\top X\|_F^2 &= Trace((X - UU^\top X)(X - UU^\top X)^\top) \\
&= Trace(XX^\top - XX^\top UU^\top - UU^\top XX^\top + UU^\top XX^\top UU^\top) \\
&= Trace(XX^\top - XX^\top UU^\top - UU^\top XX^\top + UU^\top XX^\top UU^\top) \\
&= \|X\|_F^2 - 2Trace(XX^\top UU^\top) + Trace(UU^\top UU^\top XX^\top) \\
&= \|X\|_F^2 - 2Trace(XX^\top UU^\top) + Trace(UU^\top XX^\top) \\
&= \|X\|_F^2 - Trace(XX^\top UU^\top)
\end{aligned}$$

So the original optimization problem becomes:

$$\max_{U \in \mathbb{R}^{D \times d}, U^\top U = I_d} Trace(U^\top X X^\top U) \tag{2}$$

Note that

$$Trace(U^\top X X^\top U) = Trace(X X^\top U U^\top) \underbrace{\leq}_{\text{Von-Neumann inequality}} \sum_{i=1}^{D} \lambda_i(XX^\top)\lambda_i(UU^\top) = \sum_{i=1}^{d} \lambda_i(XX^\top)$$

So it is sufficient to show that take $\hat{U}$ to be the top d eigenvectors of $XX^\top$ maximizes the objective function.

> **Von-Neumann inequality**
>
> For the details and proof of Von-Neumann inequality, read 7.4.1 and 8.7.6 of this book [4].

The view of rank minimization

We now want to seek a low rank matrix, which can approximate the data matrix X best:

$$\min_{\substack{A \in \mathbb{R}^{D \times n} \\ rank(A) \leq d}} \|X - A\|_F^2 \tag{3}$$

$$\begin{aligned} \|X - A\|_F^2 &= Trace((X - A)(X - A)^\top) \\ &= Trace(XX^\top - XA^\top - AX^\top + AA^\top) \\ &= \|X\|_F^2 - 2Trace(XA^\top) + \|A\|_F^2 \end{aligned}$$

write $X = U_X \Sigma_X V_X^\top$, $A = U_A \Sigma_A V_A^\top$, then we further have:

$$
\begin{aligned}
\|X\|_F^2 - 2Trace(XA^\top) + \|A\|_F^2 &= \sum_{i=1}^{D} \sigma_i^2(X) + \sum_{i=1}^{D} \sigma_i^2(A) - 2Trace(XA^\top) \\
&\geq \sum_{i=1}^{D} \sigma_i^2(X) + \sum_{i=1}^{D} \sigma_i^2(A) - \sum_{i=1}^{D} 2\sigma_i(X)\sigma_i(A) \\
&= \sum_{i=1}^{D} (\sigma_i(X) - \sigma_i(A))^2 \\
&= \sum_{i=1}^{d} (\sigma_i(X) - \sigma_i(A))^2 + \sum_{i=d+1}^{D} \sigma_i(X)^2 \\
&\geq \sum_{i=d+1}^{D} \sigma_i(X)^2
\end{aligned}
$$

the first inequality comes again from Von Neumann inequality, and equality holds when we make $U_A = U_X, V_A = V_X$, and the last inequality holds once we make the first d singular values of A be the corresponding ones of X. So the finally optimal low rank matrix $\hat{A}$ is:

$$
\hat{A} = U_X \begin{bmatrix} \sigma_1(X) & & & \\ & \ddots & & \\ & & \sigma_d(X) & \\ & & & 0 \end{bmatrix} V_X^\top \tag{4}
$$

Robust PCA

- Noise in X
- Missing entries in X
- Outliers in X

Probabilistic PCA

- Introduce an explicit latent variable z corresponding to the subspace
- $x = Wz + \mu + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I)$
- Prior over z: $p(z) = \mathcal{N}(z|0, I)$
- Conditional distribution of the observed variable(data point) x: $p(x|z) = \mathcal{N}(x|Wz + \mu, \sigma^2 I)$
- Learn W, μ, σ with MLE or EM
- Use Bayes Theorem to do dimension reduction: $p(z|x) = \mathcal{N}(z|M^{-1}W^\top(x - \mu), \sigma^{-2}M)$, where $M = W^\top W + \sigma^2 I$

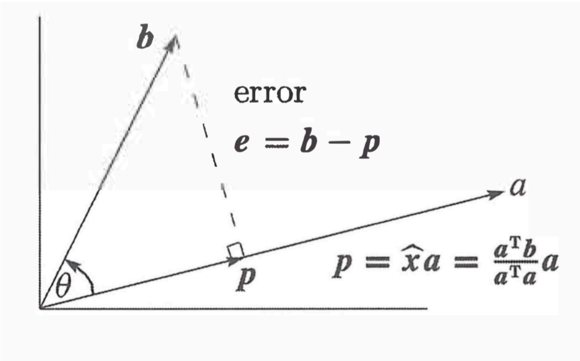
> Read PRML 12.2 for more details.

Generalized PCA(subspace clustering)

- In PCA, we assume that all the data points come from one low-dimensional subspace.
- Subspace clustering deals with the problem that the data points comes from a union of subspace
  - With unknown number of subspaces
  - With unknown dimensions of each subspace

Appendix 1

How to project a vector b onto a line with direction a?



> **The key point**
>
> p is b's projection onto a, and we know $p = \hat{x}a$, $\hat{x}$ is an unknown number. Note that $b-p$ is perpendicular to a:
>
> $$\begin{aligned} a^\top(b-p) &= 0 \\ a^\top b &= a^\top \hat{x} a \\ a^\top b &= \hat{x} a^\top a \\ \Rightarrow \quad \hat{x} &= \frac{a^\top b}{a^\top a} \\ \Rightarrow \quad p &= \frac{a^\top b}{a^\top a} a \end{aligned}$$

How to project a vector b onto a subspace with dimension n, which's basis is in the columns of $A = [a_1, .., a_n]$?

Problem: find $\hat{x}_1, \ldots, \hat{x}_n$, such that $p = \hat{x}_1 a_1 + \ldots + \hat{x}_n a_n = [a_1, a_2, \ldots, a_n] \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix} = A\hat{\boldsymbol{x}}$, and $b - p$ is perpendicular to the subspace spanned by A, in other words, $b - p$ is perpendicular to $a_1, \ldots, a_n$.

The key point again

$$
\begin{aligned}
\boldsymbol{a}_1^{\mathrm{T}}(b - A\widehat{\boldsymbol{x}}) &= 0 \\
&\vdots \\
\boldsymbol{a}_n^{\mathrm{T}}(b - A\widehat{\boldsymbol{x}}) &= 0
\end{aligned} \tag{5}
$$

Write in matrix form: $\Rightarrow A^\top(b - A\hat{\boldsymbol{x}}) = 0$, which gives $\hat{\boldsymbol{x}} = (A^\top A)^{-1}A^\top b$, and $p = A\hat{\boldsymbol{x}} = A(A^\top A)^{-1}A^\top b$.

References

1. Su, Xin, et al. "Task Understanding from Confusing Multi-task Data." *International Conference on Machine Learning*. PMLR, 2020. ↩

2. Hu, Hengtong, et al. "One-bit Supervision for Image Classification." *arXiv preprint arXiv:2009.06168* (2020). ↩

3. Von Luxburg, Ulrike. "A tutorial on spectral clustering." *Statistics and computing* 17.4 (2007): 395-416.APA ↩

4. Horn, Roger A., and Charles R. Johnson. *Matrix analysis*. Cambridge university press, 2012. ↩