Machine Learning I: Fractal 2

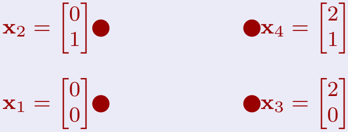
Rajendra Nagar

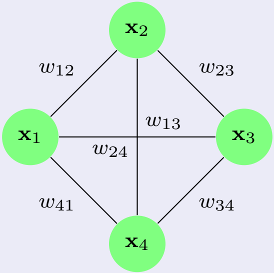
Assistant Professor
Department of Electircal Engineering
Indian Institute of Technology Jodhpur
http://home.iitj.ac.in/∼rn/

These slides are prepared from the following book:
Shalev-Shwartz, Shai, and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.

Spectral Clustering

Construct a graph where a vertex represents a data point, and every two vertices are connected by an edge with weight $\mathbf{W}_{i,j} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}}$. Partition the graph such that the edges between different groups have low weights and the edges within a group have high weights.





Let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be the clustering and $\mathbf{H} \in \mathbb{R}^{n \times k}$ be a matrix such that

$$\mathbf{H}_{i,j} = \frac{1}{\sqrt{|\mathcal{C}_j|}} \mathbb{1}_{[i \in \mathcal{C}_j]}.$$

The matrix $\mathbf{H}$ is an orthogonal matrix, i.e., $\mathbf{H}^\top \mathbf{H} = \mathbf{I}$, and satisfies the below relation.

$$\text{RatioCut}(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \sum_{i=1}^{k} \frac{1}{|\mathcal{C}_i|} \sum_{r \in \mathcal{C}_i} \sum_{s \notin \mathcal{C}_i} \mathbf{W}_{r,s} = \text{trace}(\mathbf{H}^\top \mathbf{L} \mathbf{H}).$$

Problem Formulation

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \text{RatioCut}(\mathcal{C}_1,\ldots,\mathcal{C}_k) \Leftrightarrow \min_{\mathbf{H}\in\mathbb{R}^{n\times k},\mathbf{H}^\top\mathbf{H}=\mathbf{I}} \text{trace}(\mathbf{H}^\top\mathbf{L}\mathbf{H}).$$

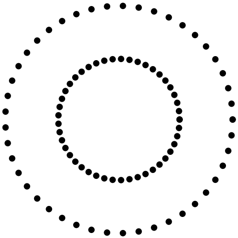
Let $\mathbf{L}\mathbf{u}_i = \lambda_i\mathbf{u}_i, \forall i \in \{1,2,\ldots,n\}$ be the EVD of the matrix $\mathbf{L}$. Here, we assume that the eigenvalues are such that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then, the solution to the above problem is $\mathbf{H}^\star = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_k\end{bmatrix}$.

Rayleigh quotient

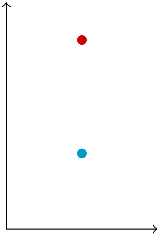
$$\arg\min_{\mathbf{v}_1,\ldots,\mathbf{v}_k} \sum_{i=1}^{k} \mathbf{v}_i^\top\mathbf{L}\mathbf{v}_i \text{ subject to } \mathbf{v}_i^\top\mathbf{v}_j = 1 \text{ if } i=j \text{ and } \mathbf{v}_i^\top\mathbf{v}_j = 0 \text{ if } i\neq j.$$

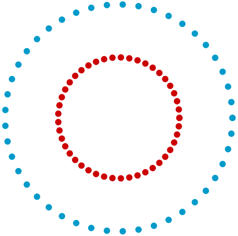
$$\begin{aligned}
f(\mathbf{v}_1,\ldots,\mathbf{v}_k) &= \sum_{i=1}^{k}\mathbf{v}_i^\top\mathbf{L}\mathbf{v}_i + \sum_{i=1}^{k}\lambda_i(1-\mathbf{v}_i^\top\mathbf{v}_i) \\
\nabla_{\mathbf{v}_i} f &= 2\mathbf{L}\mathbf{v}_i - 2\lambda\mathbf{v}_i \\
\mathbf{L}\mathbf{v}_i &= \lambda_i\mathbf{v}_i \\
\mathbf{v}_i^\top\mathbf{L}\mathbf{v}_i &= \lambda_i
\end{aligned}$$

Therefore, we have to minimize $\sum_{i=1}^{k}\lambda_i$ such that $\mathbf{L}\mathbf{v}_i = \lambda\mathbf{v}_i$ and $\mathbf{v}_i^\top\mathbf{v}_j = 0$ if $i \neq j$. Hence, $\mathbf{v}_i^\star$ = eigenvector of the matrix $\mathbf{L}$ corresponding to the i^{th} smallest eigenvalue $= \mathbf{v}_i$.



$$\mathbf{H} = \begin{bmatrix} 1 & 0.5 \\ 1 & 0.5 \\ \vdots & \vdots \\ 1 & 0.5 \\ 1 & -0.5 \\ 1 & -0.5 \\ \vdots & \vdots \\ 1 & -0.5 \end{bmatrix}$$



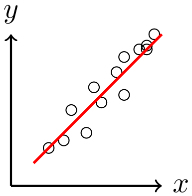


Spectral Clustering Algorithm

1: **Input:** $\mathbf{W} \in \mathbb{R}^{n \times n}$, Number of clusters k.
2: **Initialize:** Compute the graph Laplacian $\mathbf{L}$.
3: $\mathbf{H} \leftarrow$ matrix whose columns are the eigenvectors of $\mathbf{L}$ corresponding to the k-smallest eigenvalues.
4: $\mathbf{r}_1, \ldots, \mathbf{r}_n$ be the rows of $\mathbf{H}$.
5: Cluster the points $\mathbf{r}_1, \ldots, \mathbf{r}_n$ using k-means algorithm.
6: **Output:** Clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of the k-means algorithm.

Dimensionality Reduction

- Dimensionality reduction is the process of mapping the input data into a new space whose dimensionality is much smaller.
- High dimensional data impose computational challenges.
- Dimensionality reduction can be used for interpretability of the data, finding meaningful structure of the data, and illustration purpose.



Principal Component Analysis Algorithm

- Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be an input dataset where each data point $\mathbf{x}_i \in \mathbb{R}^d$.
- We would like to reduce the dimensionality of these vectors using a linear transformation.

$$\mathbf{x} \in \mathbb{R}^d \longrightarrow \mathbf{W} \in \mathbb{R}^{k \times d} \longrightarrow \mathbf{W}\mathbf{x} \in \mathbb{R}^k \longrightarrow \mathbf{U} \in \mathbb{R}^{d \times k} \longrightarrow \mathbf{U}\mathbf{W}\mathbf{x} \in \mathbb{R}^d$$

- A matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$, where $k < d$, induces a mapping $\mathbf{x} \mapsto \mathbf{W}\mathbf{x}$, where $\mathbf{W}\mathbf{x} \in \mathbb{R}^k$ is the lower dimensionality representation of $\mathbf{x}$.
- Then, a second matrix $\mathbf{U} \in \mathbb{R}^{d \times k}$ can be used to recover the each original vector $\mathbf{x}$ from its compressed version.
- In PCA, we find the compression matrix $\mathbf{W}$ and the recovering matrix $\mathbf{U}$ so that the total squared distance between the original and recovered vectors is as minimum as possible:

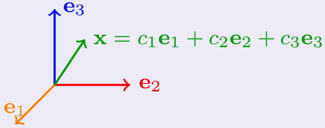
$$\mathbf{W}^\star, \mathbf{U}^\star = \underset{\mathbf{W}, \mathbf{U}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{U}\mathbf{W}\mathbf{x}_i\|_2^2.$$

- That is, for a compressed vector $\mathbf{y} = \mathbf{W}\mathbf{x}$, where $\mathbf{y}$ is in the low dimensional space $\mathbb{R}^k$, we can find $\hat{\mathbf{x}} = \mathbf{U}\mathbf{y}$, so that $\hat{\mathbf{x}}$ is the recovered version of $\mathbf{x}$ and resides in the original high dimensional space $\mathbb{R}^d$.
- **Claim:** *Let $(\mathbf{U}, \mathbf{W})$ be a solution. Then the columns of $\mathbf{U}$ are orthonormal and $\mathbf{W} = \mathbf{U}^\top$.*

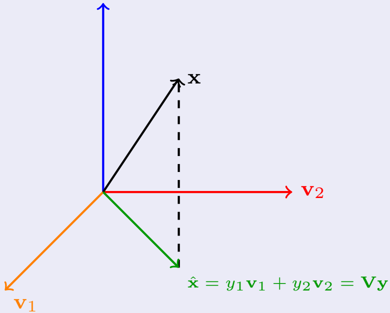
Basis Vector Representation

Let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d\}$ be a set of d unit norm orthogonal vectors, in $\mathbb{R}^d$. Then, any vector $\mathbf{x}$ in $\mathbb{R}^d$ can be written as a linear combination of these basis vectors for some constants $c_1, \ldots, c_d$ as:

$$\begin{aligned}
\mathbf{x} &= c_1\mathbf{e}_1 + c_2\mathbf{e}_2 + \cdots + c_d\mathbf{e}_d \\
c_i &= \mathbf{x}^\top \mathbf{e}_i \\
\mathbf{x} &= \begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_d\end{bmatrix}\begin{bmatrix}c_1 \\ \vdots \\ c_d\end{bmatrix} = \mathbf{E}\mathbf{c}.
\end{aligned}$$



Subspace Projection



Let $\mathcal{R}$ be a k dimensional subspace of $\mathbb{R}^d$. Let $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_k\end{bmatrix} \in \mathbb{R}^{d\times k}$ be an orthonormal matrix containing the basis vectors of $\mathcal{R}$. Then, the closest vector $\mathbf{x}^\star \in \mathcal{R}$ to a vector $\mathbf{x} \in \mathbb{R}^d$ can be found by solving the below optimization problem.

$$\mathbf{x}^\star = \underset{\hat{\mathbf{x}} \in \mathcal{R}}{\text{argmin}} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2.$$

We know that $\hat{\mathbf{x}} = \mathbf{V}\mathbf{y}$ for some $\mathbf{y} \in \mathbb{R}^k$, hence

$$\mathbf{y}^\star = \underset{\mathbf{y} \in \mathbb{R}^k}{\text{argmin}} \|\mathbf{x} - \mathbf{V}\mathbf{y}\|_2^2$$

Then, $\mathbf{y}^\star = \mathbf{V}^\top\mathbf{x} \Rightarrow \mathbf{x}^\star = \mathbf{V}\mathbf{V}^\top\mathbf{x}$.

Solution

$$
\begin{aligned}
&\underset{\mathbf{U}\in\mathbb{R}^{d\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}}{\text{argmin}} && \sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{U}\mathbf{U}^\top\mathbf{x}_i\|_2^2\\
&\|\mathbf{x}_i-\mathbf{U}\mathbf{U}^\top\mathbf{x}_i\|_2^2 &=& (\mathbf{x}_i-\mathbf{U}\mathbf{U}^\top\mathbf{x}_i)^\top(\mathbf{x}_i-\mathbf{U}\mathbf{U}^\top\mathbf{x}_i)\\
&&=& (\mathbf{x}_i^\top-\mathbf{x}_i^\top\mathbf{U}\mathbf{U}^\top)(\mathbf{x}_i-\mathbf{U}\mathbf{U}^\top\mathbf{x}_i)\\
&&=& \mathbf{x}_i^\top\mathbf{x}_i-2\mathbf{x}_i^\top\mathbf{U}\mathbf{U}^\top\mathbf{x}_i+\mathbf{x}_i^\top\mathbf{U}\mathbf{U}^\top\mathbf{U}\mathbf{U}^\top\mathbf{x}_i\\
&&=& \mathbf{x}_i^\top\mathbf{x}_i-\mathbf{x}_i^\top\mathbf{U}\mathbf{U}^\top\mathbf{x}_i
\end{aligned}
$$

$$
\max_{\mathbf{U}\in\mathbb{R}^{d\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}}\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{U}\mathbf{U}^\top\mathbf{x}_i \quad\Leftrightarrow\quad \max_{\mathbf{U}\in\mathbb{R}^{d\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}}\text{Trace}(\mathbf{U}^\top\mathbf{X}\mathbf{X}^\top\mathbf{U})
$$

Let $\mathbf{X}\mathbf{X}^\top\mathbf{u}_i=\lambda_i\mathbf{u}_i, \forall i\in\{1,2,\ldots,n\}$ be the EVD of the matrix $\mathbf{X}\mathbf{X}^\top$. Here, we assume that the eigenvalues are such that $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_n$. Then, the solution to the above problem is $\mathbf{U}^\star=\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_k\end{bmatrix}$.

PCA Algorithm

1: **Input:** Let $\mathbf{X}=\begin{bmatrix}\mathbf{x}_1 & \cdots & \mathbf{x}_n\end{bmatrix}$ be a set of input points.
2: Let $\mathbf{X}\mathbf{X}^\top\mathbf{u}_i=\lambda_i\mathbf{u}_i$ be the EVD of $\mathbf{X}\mathbf{X}^\top$ with $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_n$.
3: $\mathbf{U}\leftarrow\begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_k\end{bmatrix}$.
4: $\hat{\mathbf{x}}_i\leftarrow\mathbf{U}^\top\mathbf{x}_i$.