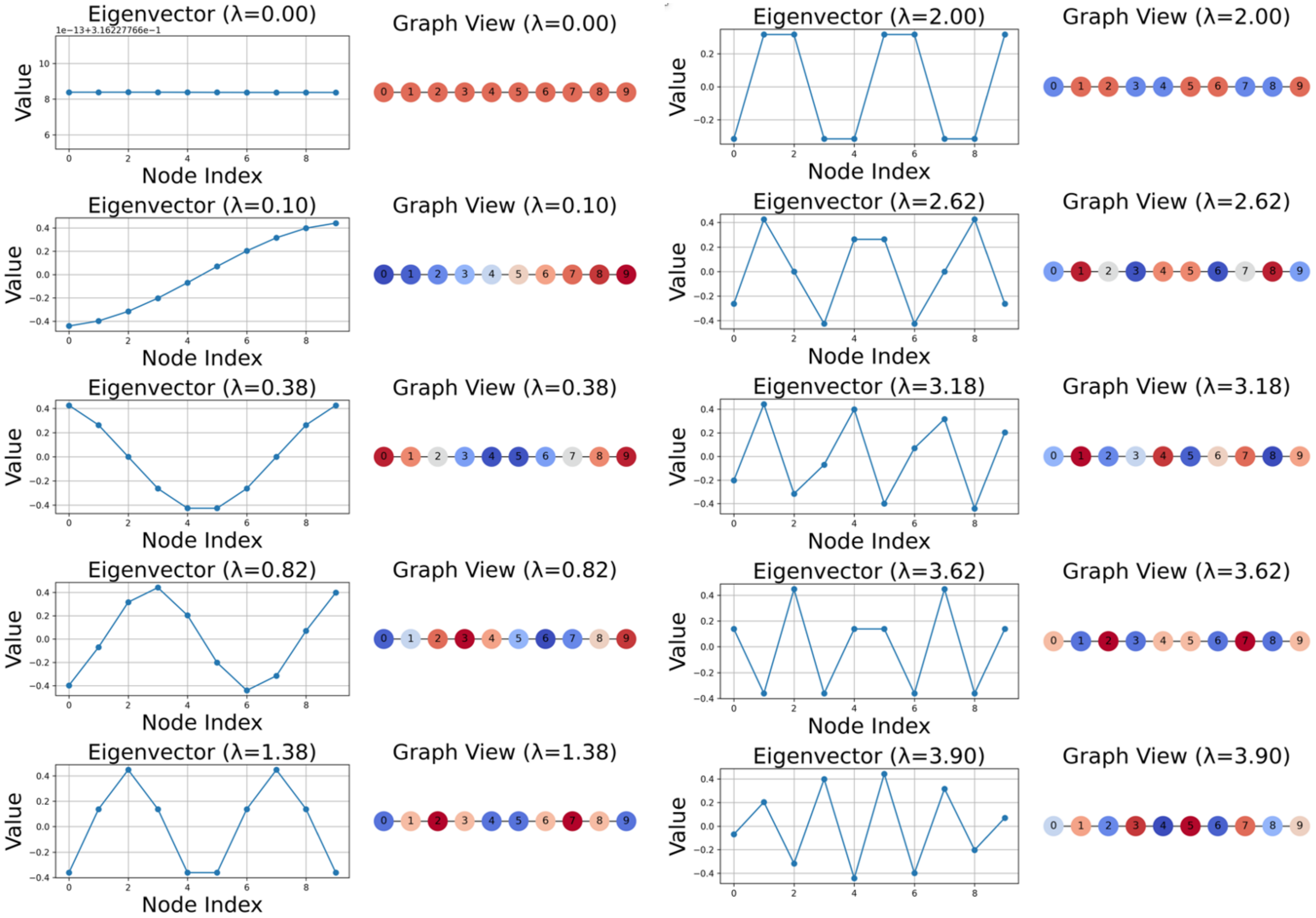
Scan the QR to record your attendance

Quick review – An illustration of EVD on L

A line graph with 10 nodes

A → L=D-A → EVD

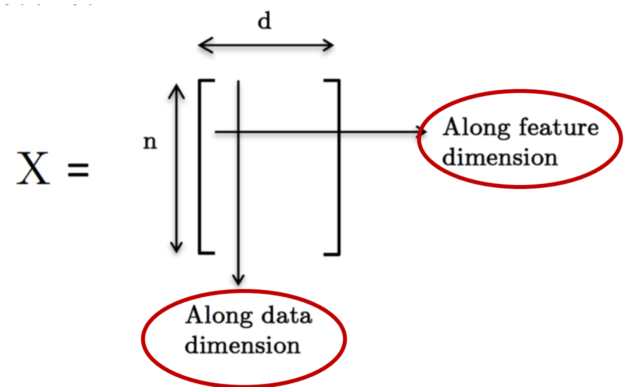
→ $\lambda \in R^{10}$ & 10 * u $\in R^{10}$



Quick review -- Introduction to Graph Science

- Unnormalized / Normalized / Random Walk Graph Laplacian
- EVD on L
  - u_k : Fourier functions, i.e. vibration modes of the graph.
  - λ_k : frequencies of the Fourier functions, i.e. how fast u_k vibrate.

- Distance
  - Euclidean/L2, Cosine/dot product, KL divergence, Wasserstein distance…
- Pre-processing
  - Center, Z-scoring, Project on L_2 sphere, Min-Max normalization



Quick review -- Graph Clustering

- k-means: partition the dataset into k clusters, minimize the least-squares
  - $L = \frac{1}{kn}\sum_{q=1}^{k}\sum_{i\in S_q}\left|\left|x_i - m_q\right|\right|^2$
- EM for k-means: cluster update (expectation) ←→ mean update (maximization)
  - Monotonic loss, Guaranteed convergence, Speed complexity O(n·d·k· n_i)
  - NP-hard, Good initialization
- Non-liner k-means: $x \to \phi(x)$, higher dim, more separable, $L = \frac{1}{kn}\sum_{q=1}^{k}\sum_{i\in S_q}\theta_i\left|\left|\phi(x_i) - m_q\right|\right|^2$
- Indicator function: $F \in \{0,1\}^{n\times k}$ → $L = \frac{1}{kn}tr(F^{\top}\Theta D), \Theta = diag(\theta_1, \dots, \theta_n) \in R^{n\times n}, D \in R^{n\times k}$ distance
  - $D = diag(K)1_K^{\top} - 2K\Theta FZ + 1_n diag(ZF^{\top}\Theta K\Theta FZ) \in R^{n\times k}$
  - $K = \phi(x)\phi(x)^{\top} \in R^{n\times n}$ (mapping, similarity) → direct similarity → $K(x_i, x_j)$ linear, Gaussian, Polynomial…
  - Now, EM becomes
    - $F^0 \to D^0 \to F^1 \to D^1 \to \cdots$
    - No m_q in D → No maximization !!!

Quick review -- Graph Clustering

$$D = diag(K)1_K^\top - \mathbf{2K\Theta FZ} \quad + \quad \mathbf{1_n diag}(\boldsymbol{ZF^\top \Theta K \Theta F Z})$$

$$\frac{\overset{K}{\phi(x_i)^\top} \overset{F}{\sum_{j\in S_q}} \overset{\Theta}{\theta_j}\,\overset{K}{\phi(x_j)}}{\underbrace{\sum_{j\in S_q}\theta_j}_{Z^{-1} = \mathrm{diag}(1_n^T \Theta F)}}$$

$$\overset{K}{\left(\frac{\overset{F}{\sum_{j\in S_q}} \overset{\Theta}{\theta_j}\phi(x_j)}{\underbrace{\sum_{j\in S_q}\theta_j}_{Z^{-1}}}\right)\left(\frac{\overset{F}{\sum_{j\in S_q}} \overset{\Theta}{\theta_j}\phi(x_j)}{\underbrace{\sum_{j\in S_q}\theta_j}_{Z^{-1}}}\right)^\top}$$

$$D_{iq} = \left|\left|\phi(x_i) - m_q\right|\right|^2 = K_{ii} \quad - \quad \underbrace{2A_{iq}}_{\phi(x_i)^\top m_q} \quad + \quad \underbrace{B_{qq}}_{m_q m_q^\top}$$

Quick review -- Graph Clustering

- Non-liner k-means: $L = \frac{1}{kn}\sum_{q=1}^{k}\sum_{i\in S_q}\theta_i\left|\left|\phi(x_i) - m_q\right|\right|^2$ → $m_q \in R^{k\times d}$ EM means
- Spectral mean $m_{\mathrm{i}} \in R^{\mathrm{n}\times d}$: cluster-centric → sample-centric
  - $L = \frac{1}{kn}\sum_{q=1}^{k}\sum_{i\in S_q}\theta_i\left|\left|\phi(x_i) - m_q\right|\right|^2 = \frac{1}{kn}\sum_{q=1}^{k}\sum_{i\in S_q}\left|\left|\theta_i^{1/2}\phi(x_i) - \theta_i^{1/2}m_q\right|\right|^2 = \frac{1}{kn}\left|\left|\theta^{\frac{1}{2}}\phi - \theta^{\frac{1}{2}}\boldsymbol{M}\right|\right|^2$, $M = FZF^{\top}\Theta\phi$
  - Indicator $Y = \Theta^{1/2}FZ^{1/2}$ → $L = \frac{1}{kn}\left|\left|\theta^{\frac{1}{2}}\phi - \boldsymbol{YY}^{\top}\theta^{\frac{1}{2}}\phi\right|\right|^2$ → $\max_{Y\in R^{n\times k}} tr(Y^{\top}\Theta^{1/2}K\Theta^{1/2}Y)$ → $\max_{Y\in R^{n\times k}} tr(Y^{\top}\boldsymbol{A}Y)$
  - A is symmetric & positive semi-definite → Y are the k largest eigenvectors, $\max_{Y\in R^{n\times k}} tr(Y^{\top}\boldsymbol{A}Y) = \sum_{q=1}^{k}\lambda_q$
- Spectral clustering
  - A=$\Theta^{1/2}K\Theta^{1/2}$ → EVD → pick the largest k eigenvectors as Y^* → binarize Y^* (standard k-means on Y^*)
  - Independent of initialization, O(n^2k)

Core API

- Plot scatter
  - plt.scatter(x, y, c, s), c is label / value …
- Compute pairwise distance
  - sklearn.metrics.pairwise.pairwise_distances(X, metric=‘euclidean’)
- Numpy
  - sort(a, axis=-1), argsort(a, axis=-1), ascending order