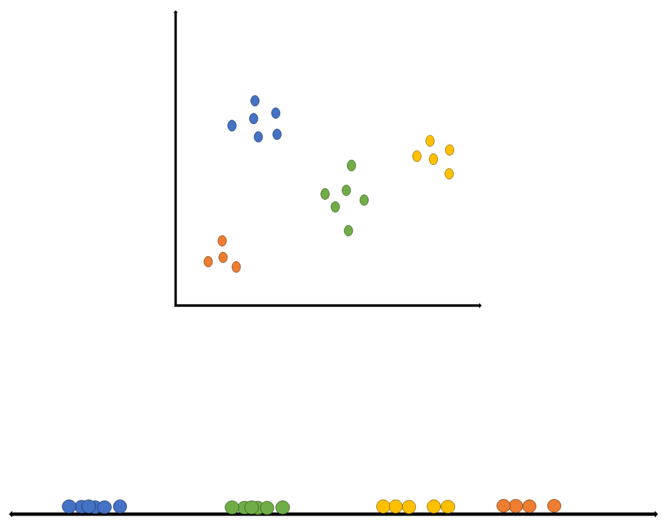
# UMAP: Uniform Manifold Approximation and Projection


C L Srinivas

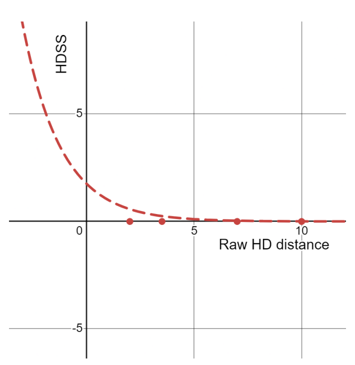
National Institute of Science Education and Research


## Why UMAP? What does it do?

1. **HD Distance matrix:-** Calculate the pairwise HD distance between data points and store it in a matrix.
2. **Calculate and symmetrize HDSS:-** Calculate HDSS for each point with respect to all the other points (even though the only non-zero scores will be those with respect to other points in it's own HD cluster).
3. **Initialize a LD graph:-** Initialize a low dimension graph using spectral embedding and the calculated HDSS.
4. **Augmenting the LD graph:-**
   1. **Randomly choose pairs of points to adjust:-** Choose three points, say a,b and c such that a and b belong to the same HD cluster and a and c belong to different HD clusters.
   2. **Calculate LDSS and adjust the points:-** Calculate the LDSS between a and b and between a and c. Adjust the position of b with respect to a and c, use the relevant cost function to minimize loss (SGD).



## High dimensional Similarity Score (HDSS)




$HDSS(p_1, p_2) = e^{-(x-d_n)/\sigma}$

$x$ = HD Euclidian distance between the two points
$d_n$ = distance between $p_1$ and it's nearest neighbour
$\sigma$ = hyperparameter

$$\sum_{i=1}^{HDN-1} HDSS(p_i) = \log_2(HDN)$$

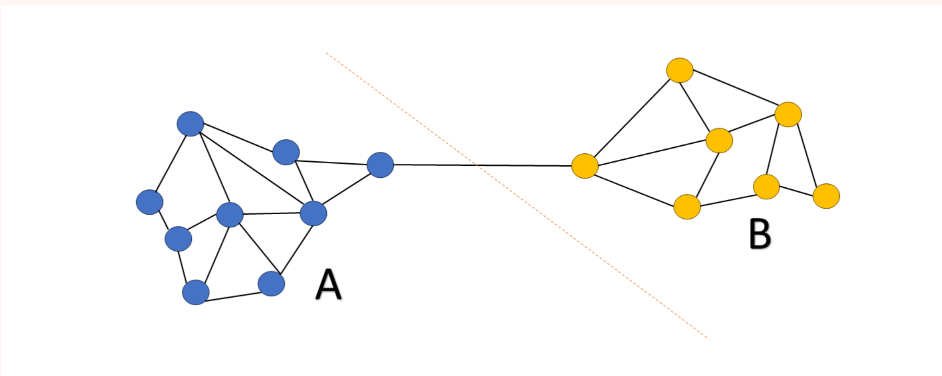
$HDN$ = number of High Dimensional Neighbours
$p_i = i^{th}$ neighbour of $p_1$

### Fuzzy Union Operation

$$HDSS(p_1, p_2) = HDSS(p_1, p_2) + HDSS(p_2, p_1) - HDSS(p_1, p_2).HDSS(p_2, p_1)$$

## Spectral Embedding

For a given matrix, the set of eigenvalues and their corresponding eigenvectors arranged in ascending order is called a "spectrum".
Using this idea, we will see how we can embed, vertices of a multi-cluster graph into a one dimensional-number line.




## Modularity and Adjacency Matrices

$$\phi(A) = \frac{\text{mod } \{(i,j) \in E; i \in A, j \notin A\}}{\min\{\text{vol}(A), 2m - \text{vol}(A)\}}$$

$AX = Y, y_i = \sum_{j=1}^{n} A_{ij} \cdot x_j = \sum_{(i,j)\in E} x_j$

*for a graph with two d-regular components*

$AX = \lambda X,$

$$X = \begin{cases} x_i = 1 & \text{if } i \in A, \\ x_i = 0 & \text{if } i \in B. \end{cases}$$

$\lambda = d$

$$\begin{array}{c|cccc} & v_1 & v_2 & \dots & v_n \\ \hline v_1 & 0 & 1 & \dots & 0 \\ v_2 & 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_n & 0 & 1 & \dots & 0 \end{array}$$

## Graph Laplacian and Eigenvectors

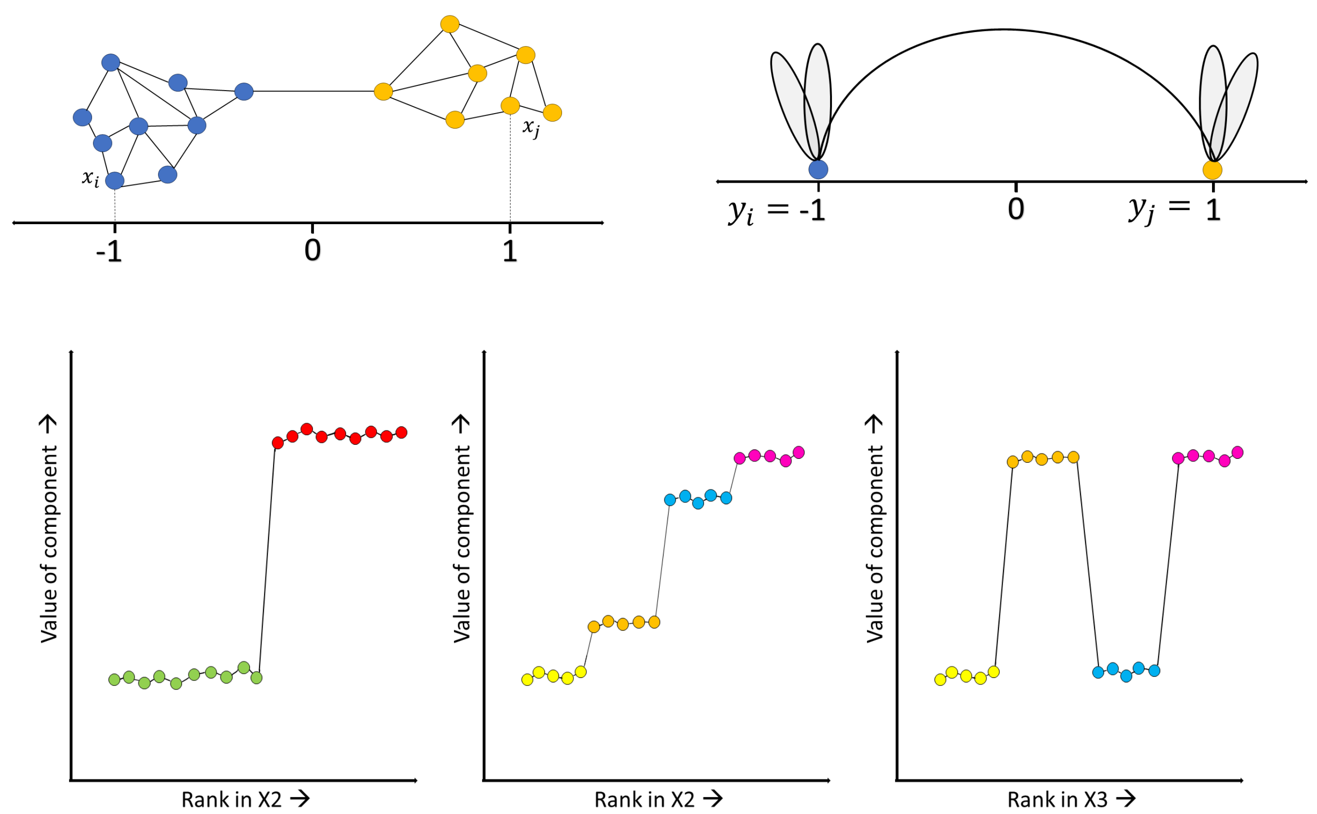
$$\lambda_2 = \min_x \frac{x^T M x}{x^T x}$$

$$\lambda_2 = \min \sum_{(i,j)\in E} (x_i - x_j)^2$$

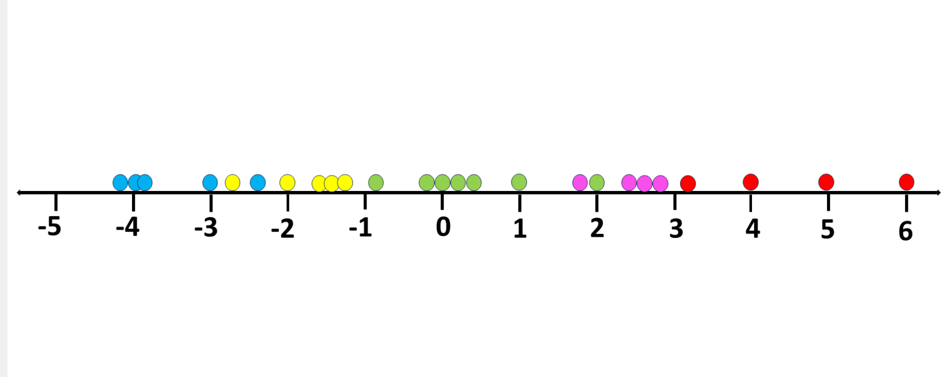
### Graph Laplacian

$$\begin{array}{c|cccc} & v_1 & v_2 & \dots & v_n \\ \hline v_1 & d_{v1} & -1 & \dots & 0 \\ v_2 & -1 & d_{v2} & \dots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_n & 0 & -1 & \dots & d_{vn} \end{array}$$

## Application of Embedding




## Initializing Low-Dimensional graph



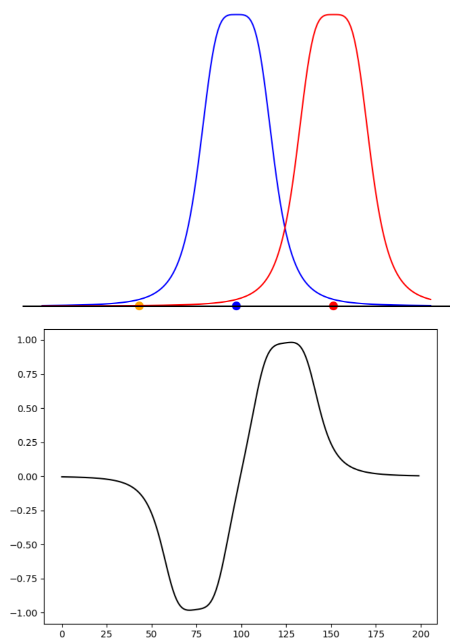

## Computing Low-Dimension Similarity Scores (LDSS) and Shifting the Points in the LD Graph

$$LDSS(p_i, p_j) = \frac{1}{1 + \alpha d(p_i, p_j)^{2\beta}}$$

$\alpha = parameter$

$d(p_i, p_j) = LD\ distance\ between\ p_i\ and\ p_j$

$\beta = parameter$



$$cost = \log\left(\frac{1}{s_n}\right) + \log\left(\frac{1}{1 - s_{nn}}\right)$$

$s_n = neighbour\ similarity\ score$

$s_{nn} = not\ neighbour\ similarity\ score$

## References


[1] Amr Elsayyad.
Spectral clustering - stanford university.
YouTube video playlist, 2022.

[2] StatQuest with Josh Starmer.
Umap dimension reduction, main ideas!!!, March 2022.

[3] StatQuest with Josh Starmer.
Umap: Mathematical details (clearly explained!!!), March 2022.