Classification using Minimax Distance

Madhav yadav

Indian Institute of Technology Hyderabad

Under the supervision of Dr.Balasubramaniam Jayaram

June 6, 2023

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

What is minimax distance?

Entire Dataset is represented in the form of graph G(O,D) where O is the set objects representing data points and D is the set of edges.

- Each Data point has two characteristics:
  - The vector
  - The Label.
- D_{ij} represents the pairwise distance 'd' between nodes i and j

contd.

The base distance 'd' needs to satisfy the following conditions:

- d_{ii} =0
- $\mathrm{d}_{ij} \geq 0$
- $d_{ij} = d_{ji}$

Note: The base distance need not be a metric since it need not satisfy the Triangle inequality.

contd.

$$d_{ij}{}^{MM} = \min_{r \in R_{ij}(G)} (\max_{1 \leq l \leq |r|-1} d_{r(l)r(l+1)})$$

where:
$R_{ij}(G)$ *is the set of all possible paths between* i *and* j
r *is the sequence of object indices*
$r(l)$ *is the* l^{th} *object in the path*

Plan of Action

1. First we will compute the pairwise minimax distance between all the objects i and j.
2. Then we will compute an embedding of these points in a new vector space such that the pairwise squared Euclidean distance in the new space is equal to the pairwise minimax distance in the original space.

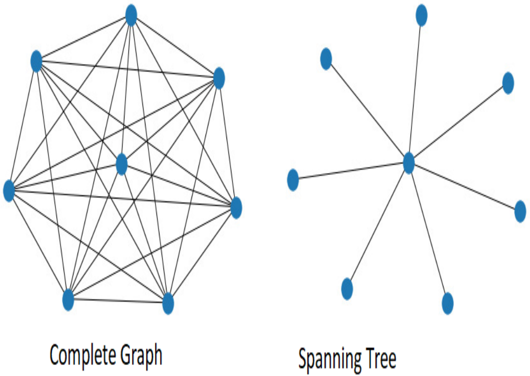
Motivation

Computing Minimax distance would be tedious if we consider each and every path. Does the MST contains all the edges that contributes to minimax distance?

Spanning Tree

Given a graph G=(O,D), a sub-graph of G that is connects all of the vertices and is a tree is called a spanning tree.

Example:

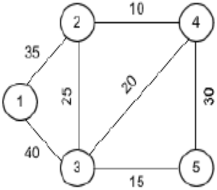


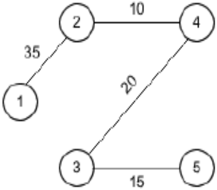
Minimum spanning tree:
A special kind of tree that minimizes the lengths (or “weights”) of the edges of the tree is called Minimum Spanning Tree(MST).

How to achieve MST?

Reverse Delete Algorithm.

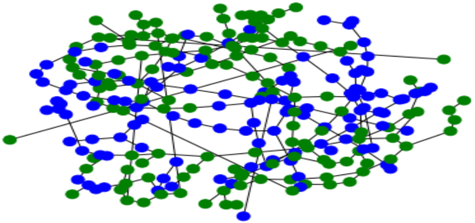
1. Sort all edges of graph in non-increasing order of edge weights.
2. Initialize MST as original graph and remove extra edges using step 3.
3. Pick highest weight edge from remaining edges and check if deleting the edge disconnects the graph or not. If disconnects, then we don't delete the edge. Else we delete the edge and continue



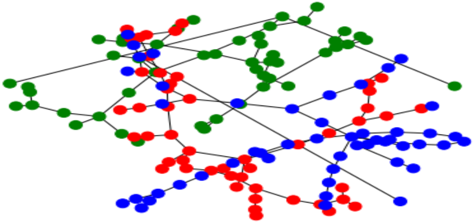


Some MST of different Data sets with Manhatten distance as base distance:

Annular dataset:



iris dataset:



Minimax distance over a graph is equivalent to its MST

Theorem

Given a graph G(O,D), for every pair of objects i,j belongs to O their minimax distance over G is identical to their Minimax distance

Proof:

Suppose the two arbitrary nodes $i, j \in O$

let n be the number of possible paths between them

let $O_1, O_2, ...O_n$ be the largest edge weights in each of the respective paths.

We know that only 1 path exists in MST.

$\implies$ we have to remove the n-1 highest weighted edges.

$\implies$ we will still be left with the minimum of those maximum edge weights.

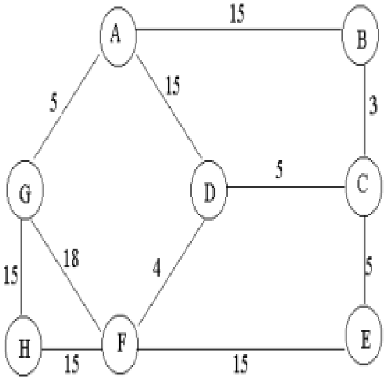
Hence, MST will contain all the edges that contributes to the minimax distance.

Now the Minimax distance for the Minimum spanning tree is

$$d_{ij}{}^{MM} = \max_{1 \leq l \leq |r|-1} d_{r(l)r(l+1)}$$

Now we can find the minimax distance between two nodes by simply getting the maximum edge weight between the path between them.

for example:



Here the minimax distance between F and G is 15
Note: the direct distance between F and G is 18.

Embedding

In this step, we are given D^{MM} matrix of the pairwise minimax distance between the objects.

Aim: *To find an embedding of the objects into a vector space such that the pairwise minimax distance in the original space is equal to the pairwise squared Euclidean distance in the new space.*

Before finding out the embedding, we must check whether an L_2^2 embedding exists or not.

Theorem: *Given the pairwise distances D^{MM}, the matrix of Minimax distances D^{MM} induces an L_2^2 embedding*

Proof:
It has been shown that every ultrametric induces an L_2^2 *embedding (Deza and Laurent 1992).*

Ultrametric

A distance measure d is said to be ultrametric if it satisfies the following properties:

1. $\forall i, j : d_{ij} = 0 \iff i{=}j$
2. $\forall i, j : d_{ij} \geq 0$
3. $\forall i, j : d_{ij} = d_{ji}$
4. $\forall i, j : d_{ij} \leq max(d_{ik}, d_{jk})$

Claim: D^{MM} *is an ultrametric.*

contd.

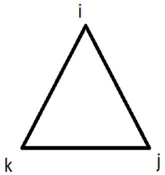
1. $D_{ij}^{MM} = 0 \iff i{=}j$
Given: $D_{ij}^{MM} = 0$;To show: $i{=}j$
$D_{ij}^{MM} = 0 \implies D_{ij} = 0 \implies i = j (\because d_{ij} = 0 \iff i = j)$
$Conversely, D_{ii}^{MM} = D_{ii} = 0$

2. $D_{ij}^{MM} \geq 0$
$\because$ All the edge weights are positive hence $D_{ij}^{MM} \geq 0$ hence
$D_{ij}^{MM} \geq 0$

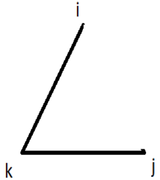
3.$D_{ij}^{MM} = D_{ij}^{MM}$
By our assumption, D is symmetric hence any path from i to j will also be a path from j to i and vice versa.

4. $D_{ij} \leq max(D_{ik}, D_{jk})$
Let us assume that $D_{ij} > max(D_{ik}, D_{jk})$

Consider 3 points i,j,k of the triangle



Case 1: d_{ij} *is the maximum*
$\Longrightarrow$ *MST doesn't contain ij*



$d_{ij}^{MM} = max(d_{ik}, d_{kj})$
$d_{ik}^{MM} = d_{ik}$

Multi Dimensional Scaling

Multidimensional scaling (MDS) is a technique for visualizing distances between objects, where the distance is known between pairs of the objects.

Types of MDS

1. Metric Multidimensional Scaling
2. Non Metric Multidimensional Scaling
3. Individual Differences Scaling
4. Multidimensional Analysis of Preference

Metric Multidimensional Scaling

- Also Known as Principal Coordinate Analysis
- Input: Distance Matrix, Output: Scatter plot which in d dimension showing the points.
- Used for Perceptual Mapping and Product Development
- It converts the Distance Matrix into a double centred Distance Matrix which is followed by **Singular Value Decomposition**.

Non Metric Multidimensional Scaling

- Almost Similar to Metric MDS
- The only difference is Metric MDS is applied for interval data scale while Non metric MDS is applied for ordinal data.

Suppose X is a given data matrix consisting of n objects with respect to p variables

$$X_{n,p} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,p} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{m,p} \end{pmatrix}$$

From the given matrix we can say that n objects are present in p dimensional space.

Aim: To obtain n objects in r dimensional space where $r < p$.
i.e. we have to find
$\tilde{X}=(\tilde{x}_{ij}), \quad i = 1, 2, 3.....n \quad and \quad j = 1, 2, 3, ...r$

let $\tilde{X}$ be the n*r matrix holding the same dissimilarity among the objects as in X
let d_{ij} be the dissimilarity between nodes i and j and it is represented as:

$$d_{ij} = \sqrt{\sum_{\lambda=1}^{r} \left(\tilde{x}_{i\lambda} - \tilde{x}_{j\lambda}\right)^2} + \epsilon_{ij} \tag{1}$$

$where \tilde{x}_{i\lambda}$ is a point of of an object i with respect to dimension λ in r $(r < p)$

$$d_{ij} = \sqrt{\sum_{a=1}^{p} \left(x_{ia} - x_{ja}\right)^2} \tag{2}$$

$where$ x_{ia} is a point of of an object i with respect to dimension a in p

In order to preserve the dissimilarity in both the dimension $\epsilon_{ij} = 0 \forall i, j$

Equation 1 can be written as:

$$D^2 = 1.1^{'} diag(\tilde{X}.\tilde{X}^{'}) - 2(\tilde{X}.\tilde{X}^{'}) + diag(\tilde{X}.\tilde{X}^{'})1.1^{'} \qquad (3)$$

where:
D^2 is an n*n matrix whose (i, j) elements are $(d_{i_j}^2)$
$'1'$ is a vector of length n (all entries as 1)

This can be further transformed as:

$$P = \frac{-1}{2} J D^2 J = \tilde{X}\tilde{X}^{'} \qquad (4)$$

Where:
P=(p_{ij}), i,j=1,2,3,....n
J is Symmetric Matrix with diagonal entries 1-$\frac{1}{n}$ and non diagonal entries as -$\frac{1}{n}$

What does the matrix P actually do?

P is a centering matrix which is symmetric and idempotent matrix, which when multiplied with a vector has the same effect as subtracting the mean of the components of the vector from every component of that vector.

This means for any given n*n matrix M

- M.P will have all row summing up to zero
- P.M will have all column summing up to zero.

$\therefore \quad 1.1' diag(\tilde{X}.\tilde{X}')$ and $diag(\tilde{X}.\tilde{X}')1.1'$ vanishes in equation 4.

Now we need to find out $\tilde{X}$.

Singular Value Decomposition(SVD)

What is SVD?

let Y be any m*n positive semi definite matrix then it can be expressed in terms of the product of three matrices.

$$Y = U\Lambda V^*$$

where
U is an $m \times m$ unitary matrix,
$\boldsymbol{\Lambda}$ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal,
V is an $n \times n$ complex unitary matrix.

$\boldsymbol{\Lambda}$ = diag(λ_1,λ_2,λ_3,..........λ_n)
where λ_1,λ_2,λ_3,..........λ_n are the Eigen values of Y in decreasing order.

What is positive semi definite matrix?

Let M be $n \times n$ Matrix then M is positive semi definite iff:

1. M is symmetric.
2. $v^T M v \geq 0$ where v is any vector of length n.

Now we will check whether P matrix is a positive semi definite. let x be any vector of length n.

$$\begin{aligned} x^T P x &= x^T \tilde{X} \tilde{X}^T x \\ &= (x^T \tilde{X})(\tilde{X}^T x) \\ &= (x^T \tilde{X})(x^T \tilde{X})^T \\ &= \left\| x^T \tilde{X} \right\|^2 \geq 0 \end{aligned}$$

$\therefore$ P is a positive semi definite matrix.

Now we can perform SVD on P.

$$
\begin{aligned}
P &= H\Lambda H^{'} \\
&= H\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}H^{'} \\
&= (H\Lambda^{\frac{1}{2}})(H\Lambda^{\frac{1}{2}})^{'} = \tilde{X}\tilde{X}^{'}
\end{aligned}
$$

$$\implies \tilde{X} = H\Lambda^{\frac{1}{2}}$$

Given: D^{MM} (Minimax distance matrix)
To compute: d dimensional vectors in the new space where minimax distance = square euclidean distance.

Step 1 Centering the distance matrix.

$$W^{MM} = \frac{-1}{2} J D^{MM} J \tag{5}$$

Step 2 Singular Value Decomposition of W^{MM}

$$W^{MM} = V \Lambda V' \tag{6}$$

Step 3 Calculating Minimax vectors

$$Y_d^{MM} = V_d (\Lambda_d)^{\frac{1}{2}} \tag{7}$$

Frame Title