CSI 436/536
Introduction to Machine Learning

Dimension reduction: MDS & ISOMAP

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York
finding nonlinear manifold
how to recover nonlinear manifold

• what is a manifold
  - a differential geometric entity that is locally described with linear space (tangent space)
  - think about manifold as a warped subspace embedded in Euclidean space

• what is distinct about manifold is it is not flat
how to recover nonlinear manifold

- manifold is not flat = it has non-zero curvatures
- the shortest distance between two points on a manifold is not given by the straight line
- it is given by a curve connecting the two points on the manifold (known as geodesic)
ISOMAP

- obtaining geodesic distances between two points
  - first construct a graph using a k-nearest neighbor of a data point
  - using Floyd algorithm to compute the pairwise shortest distances between any two points on the graph, this distance is used as geodesic distance
Floyd algorithm

• also known as Floyd-Warshall algorithm
• find the lengths (summed weights) of the shortest paths between all pairs of vertices
• running time $O(n^3)$ for $n$ being the number of nodes in a graph
• essentially a dynamic programming algorithm

```
Initialize
for k=1 to n
    for i=1 to n
        for j=1 to n
            if Dist[i,j] > Dist[i,k] + Dist[k,j]
                then Dist[i,j] ← Dist[i,k] + Dist[k,j]
```
Floyd algorithm

Initialize
for k=1 to n
  for i=1 to n
    for j=1 to n
      if Dist[i,j] > Dist[i,k] + Dist[k,j]
        then Dist[i,j] ← Dist[i,k] + Dist[k,j]

Pivot k:  
k=1: nothing
k=2: 1 → 3: 3, 1 → 5: 7
distance matrix and Gram matrix

- Distance matrix: \( D_{ij} = \) squared Euclidean distance between two vectors \( x_i \) and \( x_j \)
- Gram matrix: \( G = X^T X \), or \( G_{ij} = x_i^T x_j \), inner products between two vectors \( x_i \) and \( x_j \)
- Relation between distance matrix and Gram matrix
  \[
  D = \text{diag}(G)1^T + 1\text{diag}(G)^T - 2G
  \]
- Then we can obtain
  \[
  G = -\frac{1}{2} \left( I - \frac{1}{n} 11^T \right) D \left( I - \frac{1}{n} 11^T \right)
  \]
- this procedure is called **double centering**, i.e., it centers a matrix across both rows and columns
*Derivations

- From definition, \( D_{ij} = (x_i - x_j)^T(x_i - x_j) = x_i^T x_i - 2x_i^T x_j + x_j^T x_j \)
- So we have \( D_{ij} = G_{ii} - 2G_{ij} + G_{jj} \)
- Or \( D = \text{diag}(G)1^T + 1\text{diag}(G)^T - 2G \)(*)
- Multiply both sides by vector 1 and assume \( G1 = 0 \)
  \( D1 = \text{diag}(G)1^T1 + 1\text{diag}(G)^T1 = n\text{diag}(G) + \text{diag}(G)^T11 \)
- Multiply by vector 1 on the left \( 1^T D1 = 2n1^T \text{diag}(G) \)
- Put this back
  \( D1 = \text{diag}(G)1^T1 + 1\text{diag}(G)^T1 = n\text{diag}(G) + \frac{1}{2n}1^T D11 \)
- Now we have \( \text{diag}(G) = \frac{1}{n}D1 - \frac{1}{2n^2}1^T D11 \)
- Put this back to (*) and with some algebraic manipulation shows the result
MDS

• we obtain a matrix of pairwise geodesic distances and want to obtain low dim representation from it

distance matrix $\Rightarrow$ Gram matrix
$\Rightarrow$ covariance matrix $\Rightarrow$ apply PCA

• this process is known as multi-dimensional scaling (MDS)
PCA with Gram matrix

• PCA is based on the eigen-decomposition of the covariance matrix.

• Covariance matrix and Gram matrix share eigenvalues and eigenvectors are related.

• Gram matrix only cares about “relations between data, not data themselves.”

• procedure
  \[ G = XTX = U\Gamma^{1/2}\Gamma^{1/2}UT \]
  \[ X = \Gamma^{1/2}UT \]
MDS on Swiss swirl data

- an example showing the procedure
summary

• MDS finds nonlinear low dimensional manifold of a high dimensional dataset

• the basic steps are
  • first, compute geodesic distances between data points and form the distance (metric) matrix
  • then, obtain the Gram (inner product) matrix from the distance matrix by double centering
  • last, run PCA on the Gram matrix to obtain low dimensional manifold

• advantage: theoretical guarantee of performance

• drawback: sensitivity to parameter choices (degree of neighbors)
CSI 436/536
Introduction to Machine Learning

Spectral clustering

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York
Spectral clustering

- K-means clustering focuses on the closeness of elements **within** the same cluster
- Spectral clustering focuses on distinctiveness of elements **across** different clusters
  - represents relation between data using an undirected weighted graph (similarity graph)
  - weights on the graph correspond to data similarities

\[ W_{ij} = \exp\left(-d(x_i, x_j)^2/\sigma^2\right) \]
Spectral clustering for two clusters

- use a binary indicator $v_i$ for each vertex, $v_i = 1$ if vertex $i$ is in cluster 1, $v_i = 0$ if vertex $i$ is in cluster 2
- for any edge connecting vertex $i$ and vertex $j$, $W_{ij}(v_i - v_j)^2$ measures the cost of putting them into different clusters
Spectral clustering for two clusters

- total cost of a bi-section (cut) of the graph is then
  \[
  \text{cut}(C_1, C_2) = \frac{1}{2} \sum_{i,j} W_{ij} (v_i - v_j)^2
  \]

- two vertices in the same cluster has no cost

- we aim to minimize this cost by searching optimal assignments for \( v_i \)

- this is a NP-hard problem if solved precisely
Spectral clustering for two clusters

- Expand the min-cut cost
\[
\frac{1}{2} \sum_{i,j} W_{ij}(v_i - v_j)^2 = \frac{1}{2} \left( \sum_{ij} W_{ij} v_i^2 - 2 \sum_{ij} W_{ij} v_i v_j + \sum_{ij} W_{ij} v_j^2 \right) = \sum_{i} v_i^2 \sum_{j} W_{ij} - \sum_{ij} W_{ij} v_i v_j
\]

- Introduce \( \nu = (\nu_1, \ldots, \nu_n)^\top \) and a diagonal matrix \( D = \text{diag}(W1) \) as \( D_{ii} = \sum_j W_{ij} \)

- The min-cut cost becomes \( \min_{\nu} \nu^\top L \nu, \text{ s.t. } \nu_i \in \{0,1\} \)
  - where \( L = D - W \) is the graph Laplacian matrix
  - an integer-programming problem
  - we find approximate solution by relaxation
Graph Laplacian

• Definition $L = \text{diag}(W1) - W$ Where 1 is the all one vector, it has the following properties
  • L is symmetric and positive definite
  • any constant vector is an eigenvector with eigenvalue zero
• Graph Laplacian can be understood as the differential operator for functions on a graph
  • It is a very useful tool for graph data analysis
  • # of zero eigenvalues = # of connected components in a graph
  • smallest non-zero eigenvalue is known as the Fiedler number of the graph (spectral gap)
Relaxation of the min-cut problem

- Original problem \( \min_v v^\top L v, \) \( \text{s.t.} \) \( v_i \in \{0,1\} \) is intractable (exponential number of possible solutions)

- Approximation by relaxation
  - Solve \( \min_v v^\top L v, \) \( \text{s.t.} \) \( \|v\| = 1, v \neq 1 \)
  - Thresholding the obtained \( v \) into binary vector
  - The approximate solution is an upper-bound of the actual objective
Solving the relaxed objective

• This is a constrained optimization

\[ \min_{\nu} \nu^T L \nu, \quad \text{s.t.} \quad \| \nu \| = 1, \nu \neq 1 \]

• Solve it by introducing Lagrangian multiplier

\[ 0 = \frac{\partial}{\partial \nu} (\nu^T L \nu - 2 \lambda (\nu^T \nu - 1)) \Rightarrow L \nu = \lambda \nu \]

• So optimal solution is necessarily an eigenvector of matrix L

• Selecting the eigenvector corresponding to the smallest non-zero eigenvalue (Fiedler number) to be the optimal \( \nu \)
CSI 436/536
Introduction to Machine Learning

classification: LDA

Professor Siwei Lyu
Computer Science
University at Albany, State University of New York
class specific dimension reduction

- data are not created equal, we are interested in classes (categories of data)
- these class specific information is important for many applications
- we would like to use dimension reduction to data with classes (we will discuss general classification problem later)
PCA for classification

removing irrelevant dimension with PCA in this case does not affect classification
class specific dimension reduction

- PCA is designed for signal representation
- there is no class difference in the definition of PCA
- PCA feature may not be relevant for classification
- those discarded PCs may contain important information for classification, even though they have little overall information contributed to represent all the data
LDA

• stands for *linear discriminant analysis*, also known as Fisher linear discriminant analysis (FLD) after its inventor, Sir Rowan Fisher (1854-1927)

• find low dimensional representation for class specific data set
  • dimensionality is determined by the number of classes
  • LDA is solved as a generalized eigenvalue problem
notations

- we assume data are in two classes with class labels as \{-1, +1\}, known as *binary classification*, examples include
  - face detection: +1: face, -1: not face
  - diagnosis: +1: positive, -1: negative
- data matrix: \( X = [X^+ \ X^-] \in R^{dxn} \)
  - positive labeled data \( X^+ = (x_1^+, \ldots, x_{n^+}) \in R^{dxn^+} \)
  - negative labeled data \( X^- = (x_1^-, \ldots, x_{n^-}) \in R^{dxn^-} \)
- \( n^+ + n^- = n \), total number of data
LDA

- basic idea (for binary classification)
  find a projection $v$, such that after project data on $v$, the two classes are best separated
LDA

• find a projection such that
  • two classes are best separated: means (centers) of two classes are far
  • data of same class are concentrated: variance of each classes are small
scatterings

• within class scattering: variances of projections of each class

• between class scattering: distance between the projected centers of two classes

• An optimal classifier needs to consider both mean and spread
between class scattering

- mean of positive data \( \mu_+ = \frac{1}{n_+} \sum_{i=1}^{n_+} x_i^+ = \frac{1}{n_+} X^+ 1_{n_+} \)

- mean of negative data \( \mu_- = \frac{1}{n_-} \sum_{i=1}^{n_-} x_i^- = \frac{1}{n_-} X^- 1_{n_-} \)

- Mean of all data \( \mu = \frac{1}{n} \sum_{i} x_i = \frac{n_+}{n} \mu_+ + \frac{n_-}{n} \mu_- \)

- square difference of means on the projection \( v \)
  - \( (v^T(\mu_+ - \mu_-))^2 = v^T(\mu_+ - \mu_-)(\mu_+ - \mu_-)^T v \)

- between class scattering matrix:
  \[
  S_b = \frac{n_+ n_-}{n^2} (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T
  \]
within class scattering

- covariance matrix of positive data
  \[ S_+ = \frac{1}{n_+} \sum_i (x_i^+ - \mu_+)(x_i^+ - \mu_+)^T = \frac{1}{n_+} \sum_i x_i^+ x_i^{+T} - \mu_+\mu_+^T \]

- covariance matrix of negative data
  \[ S_- = \frac{1}{n_-} \sum_i (x_i^- - \mu_-)(x_i^- - \mu_-)^T = \frac{1}{n_-} \sum_i x_i^- x_i^{-T} - \mu_-\mu_-^T \]

- within class scattering matrix:
  \[ S_w = \frac{n_+}{n} S_+ + \frac{n_-}{n} S_- = \frac{1}{n} \sum_i x_ix_i^T - \frac{n_+}{n} \mu_+\mu_+^T - \frac{n_-}{n} \mu_-\mu_-^T \]

- variance of data on the projection \( v \)
  - positive data: \( v^T S_+ v \), negative data: \( v^T S_- v \)
  - \( (n_+/n)v^T S_+ v + (n_-/n)v^T S_- v = v^T S_w v \) measures overall scattering
Relation

- Covariance of all data
  \[ S = \frac{1}{n} \sum_{i} (x_i - \mu)(x_i - \mu)^T = \frac{1}{n} \sum_{i} x_ix_i^T - \mu\mu^T \]

- Relation of \( S \) (overall data scattering matrix) with \( S_b \) and \( S_w \) (between and within scattering matrices, respectively)
  \[ S = S_w + S_b \]

- Need to show that
  \[ \frac{n_+}{n}\mu_+\mu_+^T + \frac{n_-}{n}\mu_-\mu_-^T - \frac{n_+n_-}{n^2}(\mu_+ - \mu_-)(\mu_+ - \mu_-)^T = \mu\mu^T \]

- Also, we have
  \[ S_b = (\mu_+ - \mu)(\mu_+ - \mu)^T + (\mu_- - \mu)(\mu_- - \mu)^T, \] which is useful when extend to the multi-class classification case
Proof

- Reformulate the between class scattering matrix

\[
\begin{align*}
\mu_+ - \mu &= \mu_+ - \frac{n_+}{n} \mu_+ - \frac{n_-}{n} \mu_- = \frac{n_-}{n} (\mu_+ - \mu_-) \\
\mu_- - \mu &= \mu_- - \frac{n_+}{n} \mu_+ - \frac{n_-}{n} \mu_- = \frac{n_+}{n} (\mu_- - \mu_+) \\
(\mu_+ - \mu)(\mu_+ - \mu)^T &= \frac{n_-^2}{n^2} (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T \\
(\mu_- - \mu)(\mu_- - \mu)^T &= \frac{n_+^2}{n^2} (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T \\
\frac{1}{n_+} (\mu_+ - \mu)(\mu_+ - \mu)^T &= \frac{n_-^2}{n^2 n_+} (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T \\
\frac{1}{n_-} (\mu_- - \mu)(\mu_- - \mu)^T &= \frac{n_+^2}{n^2 n_-} (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T \\
\frac{1}{n_+} (\mu_+ - \mu)(\mu_+ - \mu)^T + \frac{1}{n_-} (\mu_- - \mu)(\mu_- - \mu)^T &= (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T = S_b
\end{align*}
\]