Module 6: Dimensionality Reduction

Lecture 31: FastMap

The Lecture Contains:

Module 6: Dimensionality Reduction

Lecture 31: FastMap

FastMap

- Efficient way of dimensionality reduction
- Designed for Euclidean spaces
- Works for any metric space
- SVD finds axis where variance is maximum
- FastMap tries to find axis where range (or spread) is maximum
- Basic idea
    - Find axis with largest range
    - Project points onto that
    - Obtain (reduced) dimensions one by one by repeating above steps

Module 6: Dimensionality Reduction

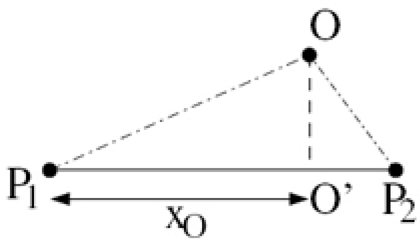
Lecture 31: FastMap

Prev topic Next topic Prev page
Next page

Axis with largest range

- Find objects (pivots) whose distance is the largest
- Use the line joining them as an approximation for the axis with largest range
    - Corresponds to diameter of the dataset
- Choosing pivots
    - Cannot look at all $O(m^2)$ distances, where m is the number of points in the dataset
- Randomly grab a point, say P_1
- Find P_2 which is at the largest distance from P_1
- Now, again find P_1 which is at the largest distance from P_2
- Repeat until convergence or for a fixed number of steps
- Analysis
    - Convergence guaranteed only after $O(m^2)$ steps
    - May not find the object pair with largest distance
    - Running time is $O(m)$ for fixed number of steps

Projection



- Pivots are P_1 and P_2
- Projection of object O onto the P_1P_2 line is O'
- O' has a distance x_O from P_1
- x_O is the new coordinate for O

$$d(P_1,O)^2 - d(P_1,O')^2 = d(O,O')^2 = d(P_2,O)^2 - d(P_2,O')^2$$

or, $$d(P_1,O)^2 - (x_O)^2 = d(P_2,O)^2 - (d(P_1,P_2) - x_O)^2$$

or, $$x_O = \frac{d(P_1,P_2)^2 + d(P_1,O)^2 - d(P_2,O)^2}{2d(P_1,P_2)}$$

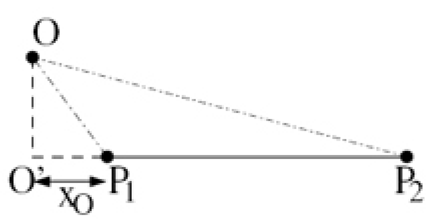
Prev topic Next topic Prev page
Next page

Analysis of projection

- Utilizes Pythagoras' theorem
- Assumes distance function d is a metric distance
- Works fine for the first coordinate in the reduced dimensionality
- Need a way to extend and generalize this
- Assume an original dimensionality of n
- After first step project everything onto a $(n-1)$-hyperdimensional plane which is perpendicular to P_1P_2 line
- Modify distance d to work for $n-1$ dimensions
- Modified distance d' may not be a metric distance any more
- Will be a metric if in Euclidean space

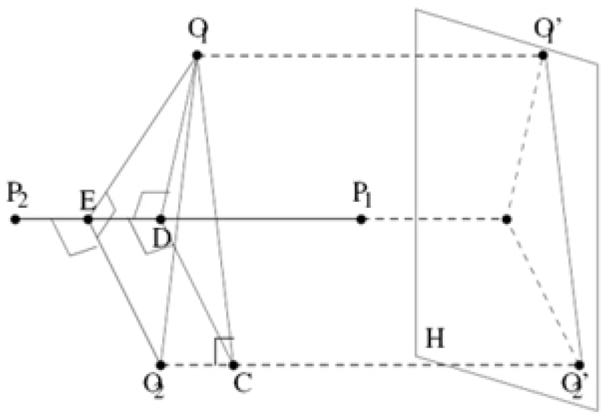
Analysis of projection

- Utilizes Pythagoras' theorem
- Assumes distance function d is a metric distance
- Works fine for the first coordinate in the reduced dimensionality
- Need a way to extend and generalize this
- Assume an original dimensionality of n
- After first step project everything onto a $(n-1)$-hyperdimensional plane which is perpendicular to P_1P_2 line
- Modify distance d to work for $n-1$ dimensions
- Modified distance d' may not be a metric distance any more
- Will be a metric if in Euclidean space
- Coordinates may be negative
    - Possible but unlikely since O and P_2 should have been pivots



Prev topic Next topic Prev page
Next page

Subsequent dimensions



- H is the hyperplane perpendicular to P_1P_2
- O_1', O_2' are projections of O_1, O_2 respectively onto H
- $d'(O_1', O_2')$ needs to be computed where d' is the modified distance function in $(n-1)$ dimensions, i.e., for H
- Once this is done, the algorithm can proceed assuming H to be the original space and d' to be the original distances
- D and E are projections of O_1 and O_2 respectively onto P_1P_2 line
- C is projection of O_1 onto O_2O_2' line

$$d(O_1, O_2)^2 = d(O_1, C)^2 + d(O_2, C)^2 = d'(O_1', O_2')^2 + (|x_1 - x_2|)^2$$

$$\therefore d'(O_1', O_2')^2 = d(O_1, O_2)^2 - (|x_1 - x_2|)^2$$

Generalization and properties

- The i^{th} coordinate for object O is given by

$$x_O^i = \frac{d_i(P_1^i, P_2^i)^2 + d_i(P_1^i, O)^2 - d_i(P_2^i, O)^2}{2d_i(P_1^i, P_2^i)}$$

where, $d_i(O_a, O_b)^2 = d_{i-1}(O_a, O_b)^2 - (|x_{O_a}^{i-1} - x_{O_b}^{i-1}|)^2$

and, $d_1(O_a, O_b)^2 = d(O_a, O_b)^2$

- P_1^i, P_2^i are the pivots chosen in the i^{th}step

Generalization and properties

- The i^{th} coordinate for object O is given by

$$x_O^i = \frac{d_i(P_1^i, P_2^i)^2 + d_i(P_1^i, O)^2 - d_i(P_2^i, O)^2}{2d_i(P_1^i, P_2^i)}$$

where, $d_i(O_a, O_b)^2 = d_{i-1}(O_a, O_b)^2 - (|x_{O_a}^{i-1} - x_{O_b}^{i-1}|)^2$

and, $d_1(O_a, O_b)^2 = d(O_a, O_b)^2$

- P_1^i, P_2^i are the pivots chosen in the i^{th}step

- Running time

  - Choosing pivots takes $O(m)$ time

  - Distance computation takes $O(n)$ time

  - Therefore, projection of m objects takes $O(m \times n)$ time

  - Needs to be iterated for k steps where k is the reduced dimensionality

  - Hence, total running time is $O(mnk)$

Generalization and properties

- The i^{th} coordinate for object O is given by

$$x_O^i = \frac{d_i(P_1^i, P_2^i)^2 + d_i(P_1^i, O)^2 - d_i(P_2^i, O)^2}{2d_i(P_1^i, P_2^i)}$$

where, $d_i(O_a, O_b)^2 = d_{i-1}(O_a, O_b)^2 - (|x_{O_a}^{i-1} - x_{O_b}^{i-1}|)^2$

and, $d_1(O_a, O_b)^2 = d(O_a, O_b)^2$

- P_1^i, P_2^i are the pivots chosen in the i^{th}step
- Running time
  - Choosing pivots takes $O(m)$ time
  - Distance computation takes $O(n)$ time
  - Therefore, projection of m objects takes $O(m \times n)$ time
  - Needs to be iterated for k steps where k is the reduced dimensionality
  - Hence, total running time is $O(mnk)$
- May not work for non-metric distances
  - May yield negative distance squares
- Contractive for Euclidean spaces