

Web Mining — Lecture 20080409

© 2007-2008 Mauro Brunato and Elisa Cilia

Facoltà di Scienze
Università di Trento

Academic Year 2007-2008, second semester
Last revision: April 15, 2008

Self Organizing Maps (SOMs)

Multidimensional scaling

▶ FastMap

Latent Semantic Indexing (LSI)

Multidimensional scaling

- ▶ k -means and Kohonen maps require document placement in (vector) space, mutual distances are not enough.
- ▶ What if only distance (or similarity) is available?
- ▶ Useful for incorporating user feedback (“ i is quite similar to j but not to k ”).

Given data

A matrix of mutual distances d_{ij} between documents.

Goal

Embed documents in a low-dimensional space (just like Kohonen maps) so that mutual distances \hat{d}_{ij} differ as little as possible from those specified in the original matrix.

Multidimensional scaling

$$\text{stress} = \frac{\sum_{ij} (\hat{d}_{ij} - d_{ij})^2}{\sum_{ij} d_{ij}^2}.$$

How do we minimize the stress?

Iterative relaxation

1. Place all points at random (or by means of an external heuristic)
2. Iterate
 - 2.1 Take a point
 - 2.2 Move it slightly in a direction so that stress is reduced

$O(n)$ distances must be evaluated to move a point.

Multidimensional scaling: FastMap

- ▶ FastMap, Faloutsos and Lin [1995]

Idea

Pretend that objects are indeed points in some unknown n -dimensional space and project these points on k mutually orthogonal directions.

Algorithm

Recursively

1. Project the objects on a carefully selected line (to obtain a coordinate)
2. Project the points to a hyperplane perpendicular to the line until we obtain a k -coordinate representation for each object

FastMap: Final Considerations

- ▶ At the end we obtain a vector (x_1, \dots, x_k) for each point x in the original data set
- ▶ FastMap runs in $O(nk)$
 - ▶ for visualization tasks becomes linear in the size of the point set

FastMap: Key points

1. How to find a good direction or line
 2. How to "project" the original points onto the line
 3. How to project the points on the hyperplane
1. choose two pivot points a and b and the line passing through them
 2. projection of the points on the line are computed using *cosine law*

$$d_{b,x}^2 = d_{a,x}^2 + d_{a,b}^2 - 2x_1 d_{a,b} \Rightarrow x_1 = \frac{d_{a,x}^2 + d_{a,b}^2 - d_{b,x}^2}{2d_{a,b}}$$

3. Determine the distance function between two projections on the hyperplane

$$d'_{x',y'} = \sqrt{d_{x,y}^2 - (x_1 - y_1)^2}$$

Projections and Subspaces

Similarity computation in clustering algorithms:

- ▶ significant fraction of the running time is spent in it
- ▶ is proportional to the total number of non-zero components of the two vectors involved

Truncation

Only the largest components of the document vectors are retained

Examples

- ▶ a fixed number of components
- ▶ the smallest number of components that make up at least 90% of the norm of the original vector

Cutting down from 10^4 to 50 dims has no significant negative impact on clustering quality [Schutze, Silverstein]

Projections and Subspaces

How many dimensions are enough?

- ▶ Orthogonal Subspace Projection \Rightarrow look at the clustering outcome
- ▶ Non-orthogonal Subspace Projection

$$k \geq \frac{4}{\epsilon^2/2 - \epsilon^3/3} \ln n \quad \text{for } 0 < \epsilon < 1$$

grants that

$$(1 - \epsilon)\|\vec{x} - \vec{y}\|^2 \leq \|f(\vec{x}) - f(\vec{y})\|^2 \leq (1 + \epsilon)\|\vec{x} - \vec{y}\|^2$$

where n is the number of documents and $f : \mathcal{R}^d \rightarrow \mathcal{R}^k$

Data-sensitive random projections

1. select k^3 docs uniformly at random
2. find k^2 clusters (using partitioning approaches)
3. note the k^2 centroid vectors
4. for each doc d , find the projection of \vec{d} onto each of the centroid vectors
5. use the k^2 -real number vector as a representation of d
6. run a clustering algorithm on the k^2 -dimensional representation