Similarity Graphs in Machine Learning

MLSS 2007

Practical Session on Graph Based Algorithms for Machine Learning

Matthias Hein and Ulrike von Luxburg

August 2007

Graphs in machine learning

Many machine learning algorithms build on graphs:

- Clustering algorithms, e.g. spectral clustering
- Dimensionality reduction algorithms based on manifolds (LLE, Isomap)
- Semi-supervised learning algorithms, e.g. label propagation
- Ranking algorithms

What is so nice about graphs?

Graphs in machine learning (2)

Many data sets have a natural graph structure:

- Web pages and the hyperlink structure
- Protein-interaction networks
- Social networks
- Citation graphs
- ...

Many of those graphs have very particular properties (for example, they are "scale free").
In this tutorial we don't talk about those "natural graphs".

Graphs in machine learning (3)

Many data sets can be transformed to a graph representation by simple means: $\rightsquigarrow$ similarity graphs.

Given:

- data "points" $X_1, ..., X_n$
- similarity values $s(X_i, X_j)$ or distance values $d(X_i, X_j)$

Construct graph:

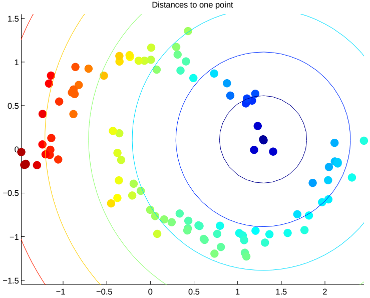
- Data points are vertices of the graph
- Connect points which are "close"
- Intuition: Graph captures local neighborhoods

Why could this be useful?

Graphs in machine learning (4)

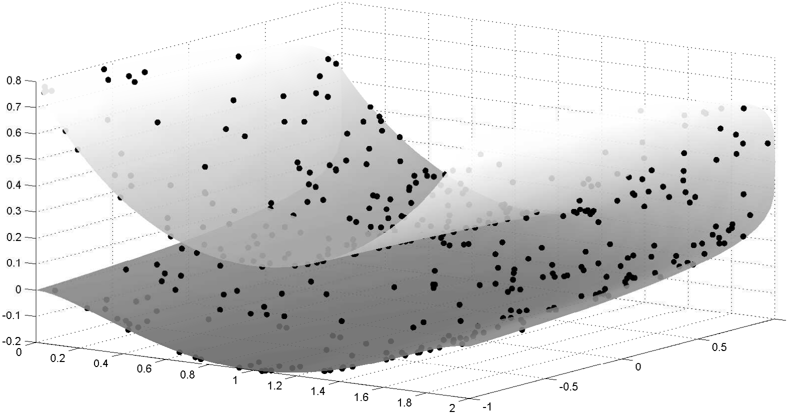
Look at similarity values:

- Usually, the similarity values are very reliable in encoding "local structure"
- Can reliably indicate which points are "close" or "similar"
- The global structure induced by a similarity or distance function often does not capture the true global structure of the data



Graphs in machine learning (5)

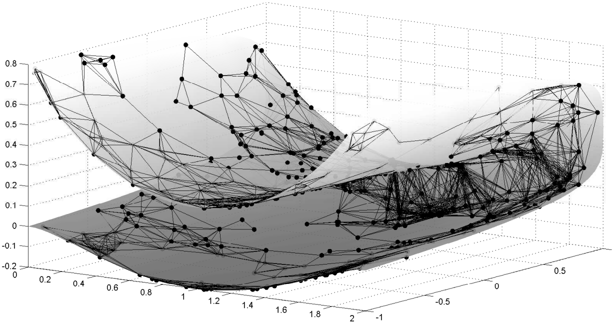
Another example for misleading global distances:



Graphs in machine learning (6)

Now idea:

- Only rely on local information provided by similarity
- Construct graph based on this local information
- Machine learning algorithm should discover global structure by itself



Graphs in machine learning (7)

Further advantages of graph-based data representations:

- They are ideally suited to represent data based on pairwise information of objects (such as similarities, distances, relations)
- They are an efficient way of encoding data (sparse)
- Graphs are omnipresent in computer science, have been studied a lot, and for many tasks efficient algorithms are known

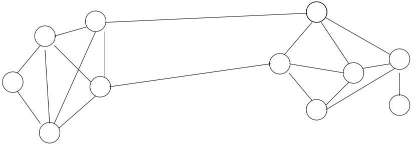
Recap: distances and similarities

A similarity score between two objects is “high” if the objects are “very similar”.
Most prominent example in $\mathbb{R}^d$: Gaussian kernel:
$s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$

A distance score between two objects is “small” if the objects are “close” to each other.
Most prominent example in $\mathbb{R}^d$: Eulidean distance:
$d(x_i, x_j) = \|x_i - x_j\|$

- Distances and similarities are “inverse” to each other:
  similarity high $\iff$ distance low
- In the following, only talk about similarities, everything also works with distances!

Recap: basic graph notation



- A graph consists of vertices and edges.
- Edges can be directed or undirected, and weighted or unweighted.
- The adjacency matrix (weight matrix) W describes the graph:
  $w_{ij} = 0$ if vertices i and j are not connected
  $w_{ij} =$ weight of the edge, if they are connected
- The degree of a vertex is the sum of all adjacent edge weights:
  $d_i = \sum_j w_{ij}$
- All vertices which can reached from each other by a path form a connected component

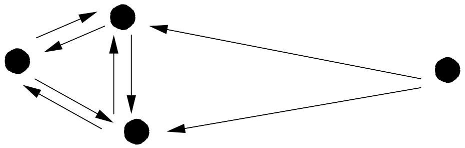
Directed k-nearest neighbor graph

- Given data objects and their pairwise similarities s_{ij}
- Connect each point to its k nearest neighbors
- Weight the edges by the similarity score

Note:

- Resulting graph is directed
- Graph is not symmetric (as neighborhood relationship is not symmetric)!!!

Two nearest neigbors:



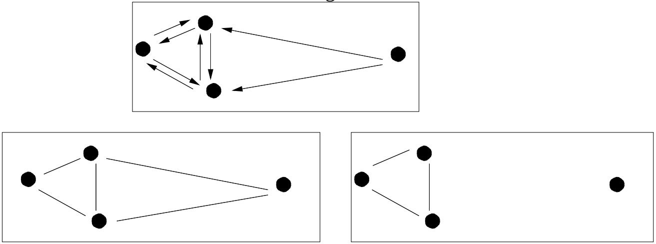
Undirected k-nearest neighbor graphs

Make directed graph undirected: either using “or” or “and” operation on directed edges

“The” kNN graph (other names: symmetric kNN graph):
connects A with B if $A \leftarrow B$ or $A \rightarrow B$

The mutual kNN graph:
connects A with B if $A \leftarrow B$ and $A \rightarrow B$

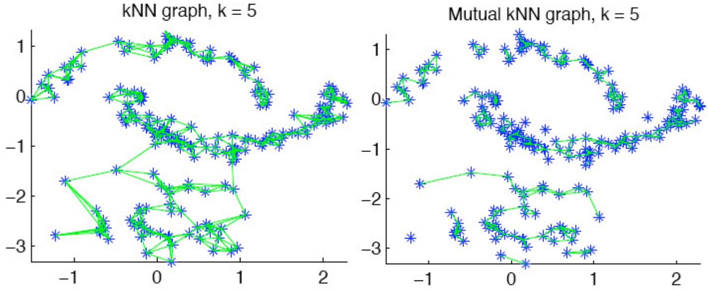
Directed nearest neigbors:



The (symmetric) kNN graph

The mutual kNN graph

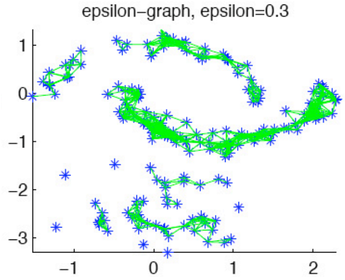
Undirected k-nearest neighbor graphs (2)



Note: by construction, the mutual kNN-graph is a subset of the symmetric kNN graph.

ε-neighborhood graph

- Given data objects and their pairwise distances d_{ij}
- Connect each point to all other points which have distance d_{ij} smaller than a threshold ε
- Either use unweighted graph
- or additionally transform distances to similarities and use similarities as weights



DemoSimilarityGraphs

Now want to explore graph properties in a demo:

- Go to `http://agbs.kyb.tuebingen.mpg.de/wikis/mlss07`
- Download file DemoSimilarityGraphs.zip
- Unzip it in some convenient folder
- Start matlab, go to the folder DemoSimilarityGraphs
- Start the demo by typing DemoSimilarityGraphs in matlab

DemoSimilarityGraphs (2)

Tuning the similarity function:

As similarity function, demo uses the Gaussian kernel:
$s(x, y) = \exp(-\|x - y\|^2/(2\sigma^2))$
Want to select a good parameter σ.

- Data set: two moons
- Noise dimensions: 0
- Choose different values for the kernel parameter σ and press "Update Data Plots"
- Look at top panel only
- How do you know whether a certain σ is useful?

DemoSimilarityGraphs (3)

Effect of noise on the similarity function:

- Same data set as before
- Now increase the number of noise dimensions.

- Try to re-adjust sigma.
- What can you observe?
- Do you have an explanation?

DemoSimilarityGraphs (4)

What happens is: distances in high-dimensional spaces become less meaningful, they mainly model noise:

Let X and Y be points drawn from two d-dim Gaussians:
$X \sim N(\mu_1, \sigma_1^2 I)$
$Y \sim N(\mu_2, \sigma_2^2 I)$
Then their expected distance satisfies

$$\begin{aligned} E\|X - Y\|^2 &= E \sum_{i=1}^d |X_i - Y_i|^2 \\ &= \sum_{i=1}^d [\ \mathrm{Var}(X_i - Y_i) + (E(X_i - Y_i))^2\ ] \\ &= d(\sigma_1^2 + \sigma_2^2) + \|\mu_1 - \mu_2\|^2 \end{aligned}$$

If d is large, the noise term $d(\sigma_1^2 + \sigma_2^2)$ will always dominate the "informative term" $\|\mu_1 - \mu_2\|^2$!!!

DemoSimilarityGraphs (5)

Comparing symmetric and mutual kNN graph:

- Choose data set Gaussians unbalanced
- 500 data points
- Noise dimensions: 0
- First adjust a reasonable σ
- In the two graph panels, choose symmetric kNN and mutual kNN

- Now try to find the smallest parameter k for which both graphs are connected (have one connected component).
- What can you observe?
- How do both graphs look like for a k which is just a bit smaller?
- And if k is much higher?

DemoSimilarityGraphs (6)

Symmetric and mutal kNN graph for high noise:

- Data set Gaussians different variance
- 150 data points
- Slowly increase the number of noise dimensions.

- What happens for 200 noise dimensions?

Any explanation?

DemoSimilarityGraphs (7)

Have already seen:

$X \sim N(\mu_1, \sigma_1^2 I),\; Y \sim N(\mu_2, \sigma_2^2 I)$

$\implies$

$E\|X - Y\|^2 = \|\mu_1 - \mu_2\|^2 + d\sigma_1^2 + d\sigma_2^2$

Assume d is large and $\sigma_1 < \sigma_2$. Then:

$$\begin{aligned} & E\|X - X'\|^2 = 2d\sigma_1^2 \\ \leq & E\|X - Y\|^2 = \|\mu_1 - \mu_2\|^2 + d\sigma_1^2 + d\sigma_2^2 \\ \leq & E\|Y - Y'\|^2 = 2d\sigma_2^2 \end{aligned}$$

- Points Y in the low-density cluster are closer to points X in the high-density cluster than to points Y' in their own cluster!

DemoSimilarityGraphs (8)

Symmetric kNN graph vs. ε-neighborhood graph:

- Data set Gaussians different variance
- 100 data points
- 0 noise dimensions

- Try to adjust ε for the ε-graph such that the graph is connected.
- What can you observe?
- What happens if ε is smaller than this?
- How does the symmetric kNN graph behave compared to this?

DemoSimilarityGraphs (9)

The degrees of the graph vertices

- Any data set
- 0 noise dimensions
- All three graphs

- Look at the plot of the degrees of the graph. How are the graph degrees related to the data set?