Distance Preservation - Part 2

Graph Distances

Niko Vuokko

October 9th 2007
NLDR Seminar

Outline

Outline

Distance along the manifold

- In a perfect embedding the distances you see are as if taken along the manifold.
- Therefore using these distances could be beneficial.
- These distances can then be instantly plugged in to MDS, NLM or CCA.
- The methods optimize the latent space variables x_i analytically or algebraically. Thus the data space distances $\delta_y(i,j)$ don't have to have any special characteristics.

Geodesic distance
Formal definition

- Simply defined as the shortest Euclidean distance along the manifold,

$$l = min_{p(z)} \int_{z(i)}^{z(j)} ||J_z m(p(z))|| dz.$$

- Simple? No.
- Solution? Discretize the manifold.

Graph distance - Practical simplification

Base construction

- In discretizing we have to construct an undirected graph out of the manifold.
- Node selection is like vector quantization, but all points may be used.
  - Enough nodes needed to explain the manifold.
  - Too much nodes means immense computation.
- Edge weights give the Euclidean distances between the edge's endpoints. Thus we have a Euclidean graph in our hands.
- How to select the edges?

Two rules for edge selection

1. Use the K closest neighbours.
2. Choose a "suitable" ε and all points closer than that.

- Too large a K or ε makes the graph very dense and computations hard. It may also let the graph jump across the void.
- Too small a K or ε may not connect the graph and gives insufficient information about the topology.
- This isn't too different from SOM...

Distances between data points

- The distance between x_i and x_j is the minimum sum of weights across any path from x_i to x_j in the graph.
- Now we need to find this minimum for all different point pairs (x_i, x_j).
- This is much easier than its analytical counterpart. Here we can efficiently calculate the distances with Dijkstra's algorithm.
- Theory says that in the ideal case these distances really give the optimal approximations for geodesic distances.

Recycling old ideas

- The old, fast linear methods with algebraic solutions are easily made nonlinear simply by replacing the Euclidean data space distances with graph distances.
- Result is a fast nonlinear method, where the nonlinearity comes from the distance used, not from the methods internals.
- Also inherently nonlinear methods such as NLM or CCA can be transformed this way to a new method.

Theory behind the transition
Developable manifold

- A manifold is called a *developable P-manifold* if its geodesic distances can be mapped to the embedding space Euclidean distances.
- After some calculus we find out that a manifold is developable iff there exists a parametric equation for the D-dimensional data set in which each coordinate depends on at most one latent variable.
- Mundanely: a manifold is developable if it is a twisted sheet of paper in space (or similar).
- Thus the swiss roll is developable, but the open box isn't.

Outline

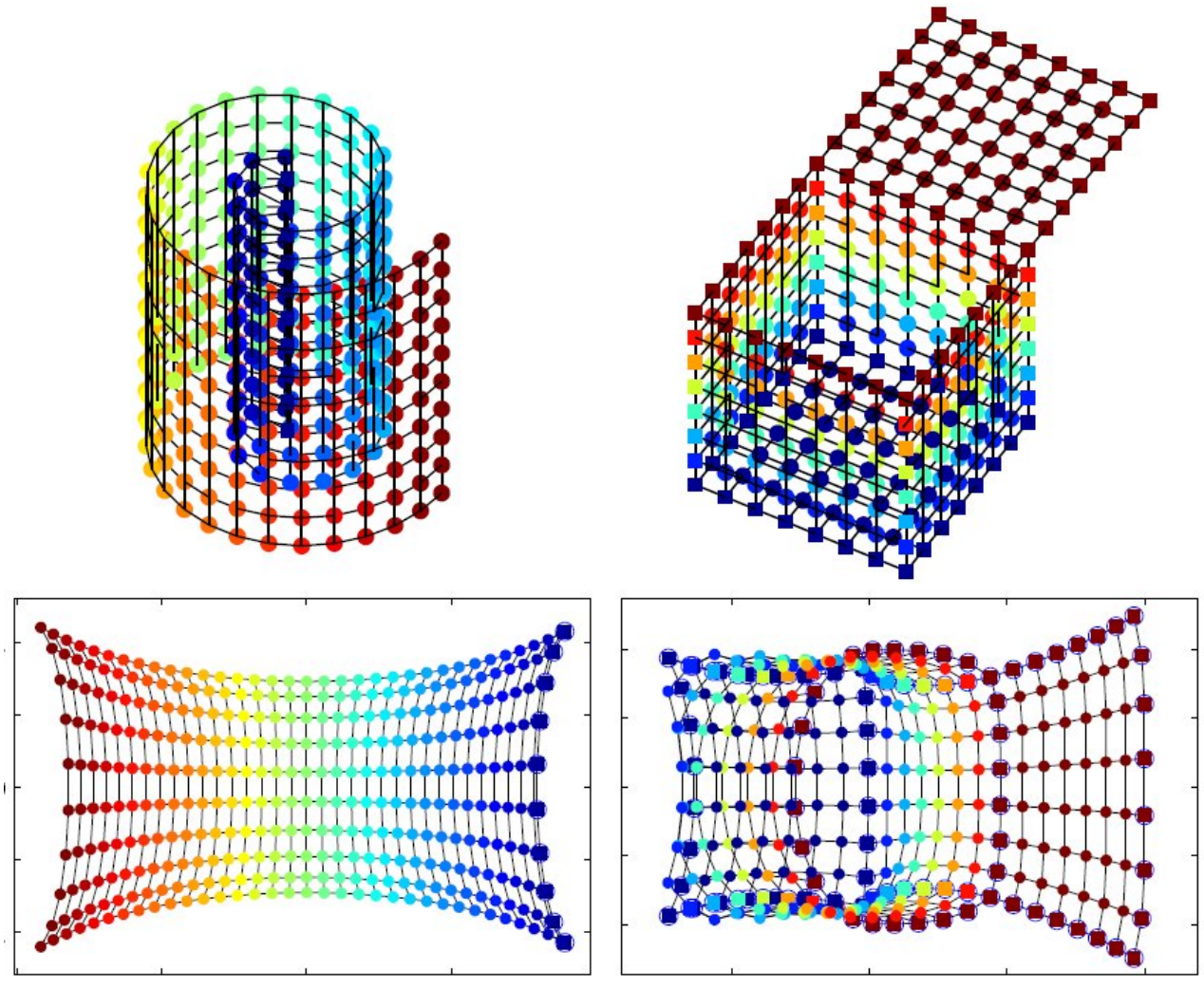
Isomap definition

- The core of the method is the same as in classical metric MDS.
- Only difference is that now the distance matrix $\mathbf{D}$ contains graph distances between the data points.
- Solution is once again $\hat{\mathbf{X}} = \mathbf{I}_{P\times N}\mathbf{\Lambda}^{1/2}\mathbf{U}^T$, where the Gram matrix $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ is calculated from $\mathbf{D}$ with double centering.
- Vector quantization, subsampling or "anchoring" the distances can be used to lower computational burden.
- The model works if the graph distances are close to the embedding space distances, i.e. if the manifold is developable.

Isomap properties

- Isomap has the same good properties as MDS: it's fast and the mappings can be done incrementally.
- One should note that the graph distance will never really explain any Euclidean configuration. Therefore some eigenvalues might be negative.
- The same elbow strategy still works when trying to figure out the intrinsic dimensionality.
- In the examples the graph distances are practically Manhattan distances. Therefore diagonal distances get overestimated and stretched.

Isomap example



Outline

Combining graph distances and NLM

- Similarly as with Isomap we merely use graph distances instead of Euclidean ones in $d_y(i,j)$ distances.
- Error function is

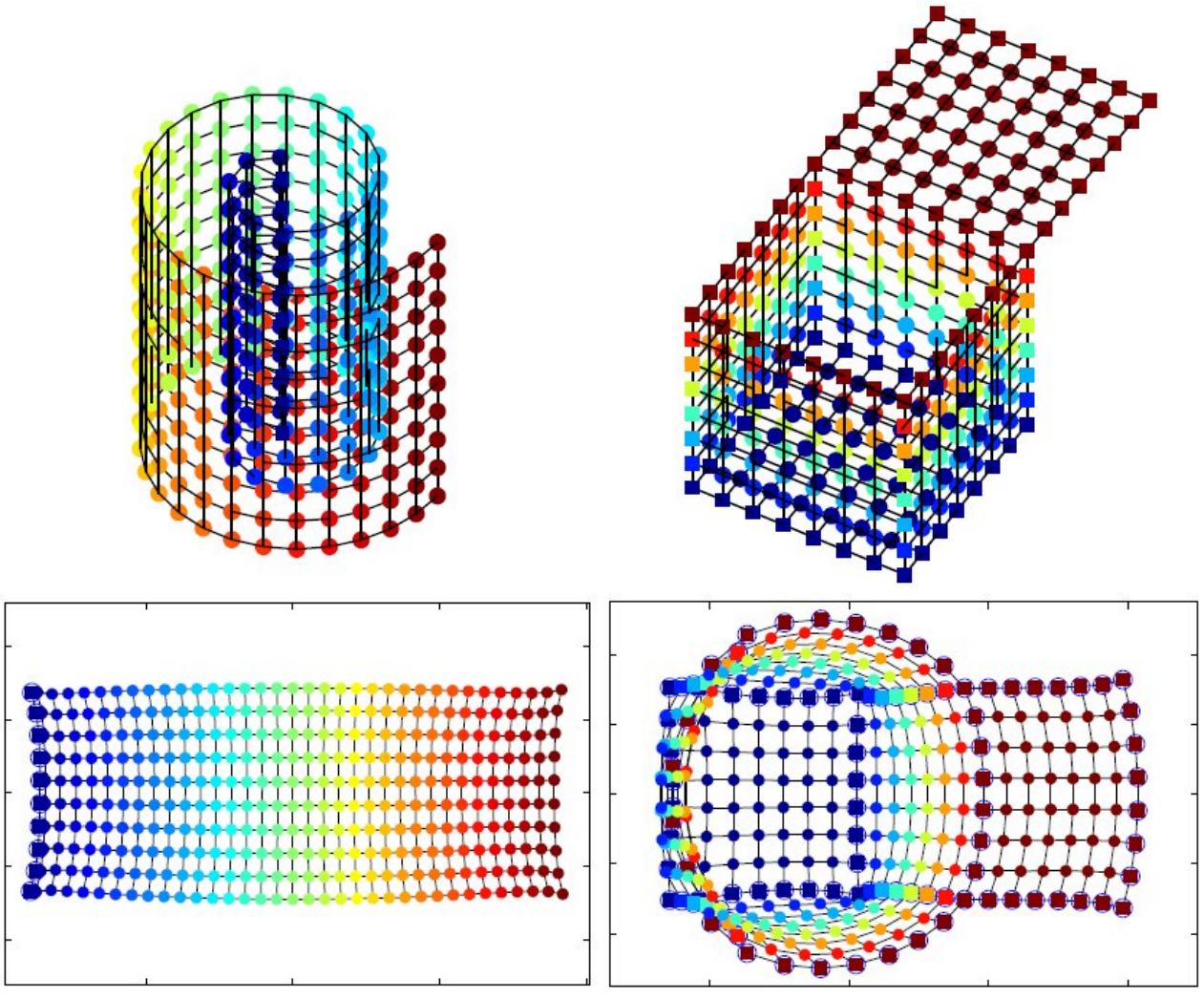
$$E_{GNLM} = \frac{1}{c} \sum_{\substack{i=1, \\ i<j}}^{N} \frac{(\delta_y(i,j) - d_x(i,j))^2}{\delta_y(i,j)},$$

  where δ_y is the graph distance.
- The same quasi-Newton update rule used in NLM is used also here.

GNLM properties

- GNLM doesn't depend on manifold developability and it moulds nonlinearity also in the optimization phase.
- This means that it should perform better than Isomap.
- Once again diagonals get longer than they should, but this can actually be beneficial in GNLM as they will then have less weight.

GNLM example



Outline

CCA + SOM = CDA

- CCA already made a clear point of looking only at the local structure.
- The weight function F_λ was used for this.
- CDA once again changes nothing but the data space distance to graph distance.
- If the manifold is developable, then the graph distances approximate the perfect embedding distances very well and F_λ has no use here.
- ...but in the real world manifolds may not be Euclidean and are definitely not when there is noise.
- Therefore F_λ is still a viable tuning component.

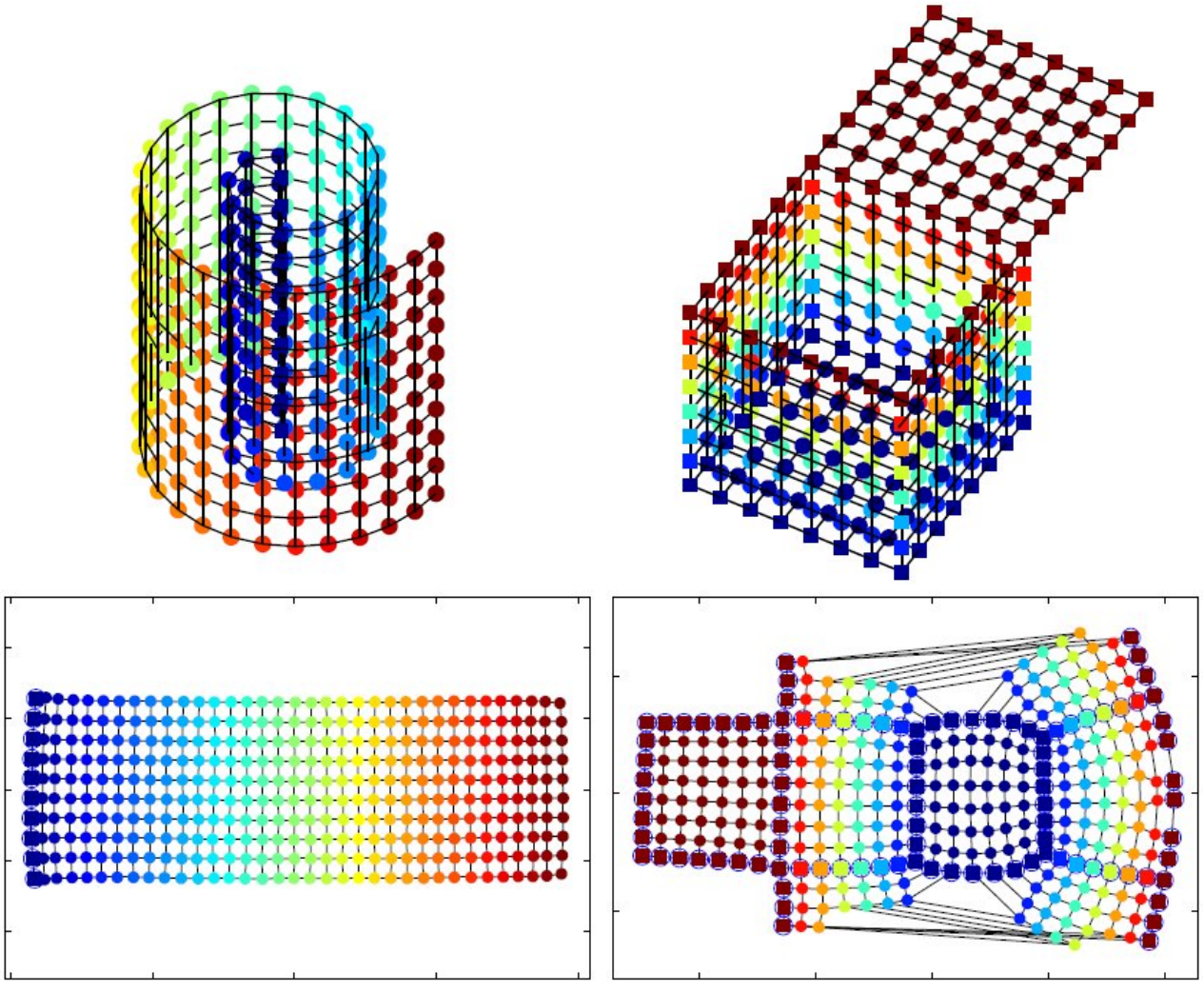
CCA + SOM = CDA

- CCA already made a clear point of looking only at the local structure.
- The weight function F_λ was used for this.
- CDA once again changes nothing but the data space distance to graph distance.
- If the manifold is developable, then the graph distances approximate the perfect embedding distances very well and F_λ has no use here.
- ...but in the real world manifolds may not be Euclidean and are definitely not when there is noise.
- Therefore F_λ is still a viable tuning component.

CDA properties

- Using the graph distance allows unwrapping even highly folded manifolds.
- Shortcuts across the “invisible forcefield“ are forbidden.
- Of course selecting λ still requires precision, but the method is more robust in this standpoint than CCA.
- We may use a neighbourhood proportion instead of λ. It can also vary during the process.
- CDA converges faster than CCA for approximately developable manifolds and its parameters need less attention.

CDA example



Outline

Large is small

- Usually DR methods try to find reductions that linearize variables to reduce dimensionality.
- Kernel PCA also tries to first linearize the data, but this time by growing the dimensionality from D to Q radically, maybe even to infinity(!)
- After the data is in the Q-dimensional space, basic MDS (as Q is large) is used to find the P-dimensional embedding.
- Usually $Q \gg N$, so MDS may (and often will) give us N non-zero eigenvalues.
- Here once again the linear MDS in extended by giving it nonlinear scalar products to work with.

The details

- The dimension-inflating mapping is $\phi : \mathbb{R}^D \to \mathbb{R}^Q$ and the Gram matrix used by MDS is $\mathbf{\Phi} = (\phi(\mathbf{y}_i) \cdot \phi(\mathbf{y}_j))_{ij}$.
- The Q-dimensional data $\phi(\mathbf{Y})$ must be centered before EVD, this can circumvented by double-centering the Gram matrix $\mathbf{\Phi}$ similarly to earlier double-centerings.
- Problem not yet discussed: What *is* ϕ? How to define it?
- Simple (?) answer: nobody really knows.
- As one can note, ϕ was used only to calculate the Gram matrix Φ.
- We can forget about ϕ and go straight to Φ by using a kernel function. Then the mapping ϕ or even Q is never actually considered explicitly.

The details

- The dimension-inflating mapping is $\phi : \mathbb{R}^D \to \mathbb{R}^Q$ and the Gram matrix used by MDS is $\mathbf{\Phi} = (\phi(\mathbf{y}_i) \cdot \phi(\mathbf{y}_j))_{ij}$.
- The Q-dimensional data $\phi(\mathbf{Y})$ must be centered before EVD, this can circumvented by double-centering the Gram matrix $\mathbf{\Phi}$ similarly to earlier double-centerings.
- Problem not yet discussed: What *is* ϕ? How to define it?
- Simple (?) answer: nobody really knows.
- As one can note, ϕ was used only to calculate the Gram matrix Φ.
- We can forget about ϕ and go straight to Φ by using a kernel function. Then the mapping ϕ or even Q is never actually considered explicitly.

The details

- The dimension-inflating mapping is $\phi : \mathbb{R}^D \to \mathbb{R}^Q$ and the Gram matrix used by MDS is $\mathbf{\Phi} = (\phi(\mathbf{y}_i) \cdot \phi(\mathbf{y}_j))_{ij}$.
- The Q-dimensional data $\phi(\mathbf{Y})$ must be centered before EVD, this can circumvented by double-centering the Gram matrix $\mathbf{\Phi}$ similarly to earlier double-centerings.
- Problem not yet discussed: What *is* ϕ? How to define it?
- Simple (?) answer: nobody really knows.
- As one can note, ϕ was used only to calculate the Gram matrix $\mathbf{\Phi}$.
- We can forget about ϕ and go straight to $\mathbf{\Phi}$ by using a kernel function. Then the mapping ϕ or even Q is never actually considered explicitly.

Kernel function

- We will use a kernel function

$$\kappa : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}, \kappa(y_i, y_j) = \phi(y_i) \cdot \phi(y_j).$$

- Of course κ should be selected so that there exists a ϕ that induces the scalar products $\kappa(\mathbb{R}^D \times \mathbb{R}^D)$.
- The kernel ideology has similarities to Support Vector Machines (SVM).

Mercer's theorem

Theorem (Mercer's theorem)

Suppose κ is a continuous kernel of a positive and positive-definite integral operator

$$\mathcal{K} : L_2 \to L_2, (\mathcal{K}f)(v) = \int \kappa(u,v)f(v)dv.$$

Then κ can be decomposed into a series

$$\kappa(u,v) = \sum_{q=1}^{\infty} \lambda_q \phi_q(u)\phi_q(v),$$

where λ_q are the eigenvalues and ϕ_q the orthonormal eigenfunctions of κ.

Shortcut to the scalar products

- This implies that the function defined as

$$\phi(y) = \sum_{q=1}^{\infty} \sqrt{\lambda_q}\phi_q(y)$$

  induces the scalar products given by κ.
- Of course once again it is overly difficult to check these conditions for some desired κ. Therefore *de facto* solutions are usually used:
  - Polynomial kernel: $\kappa(u,v) = (u \cdot v + 1)^p$, $p \in \mathbb{Z}$.
  - Radial basis functions, e.g. Gaussian kernels $\kappa(u,v) = \exp(-\frac{||u-v||^2}{2\sigma^2})$.
  - Sigmoidal functions such as $\kappa(u,v) = \tanh(u \cdot v + b)$.

KPCA properties

- In the bone KPCA is nothing but a fancy way of using basic MDS.
- This time the Gram matrix gets calculated by the kernel function.
- Nothing really justifies the use of the mentioned kernel functions, we're just optimistic that the selected kernel AND its parameters would be "compatible" with the data.
- Luckily KPCA is fast if the kernel is easy enough to compute. The EVD in MDS is the bottleneck then.
- Using sloppier kernel parameters makes KPCA more like the linear PCA.

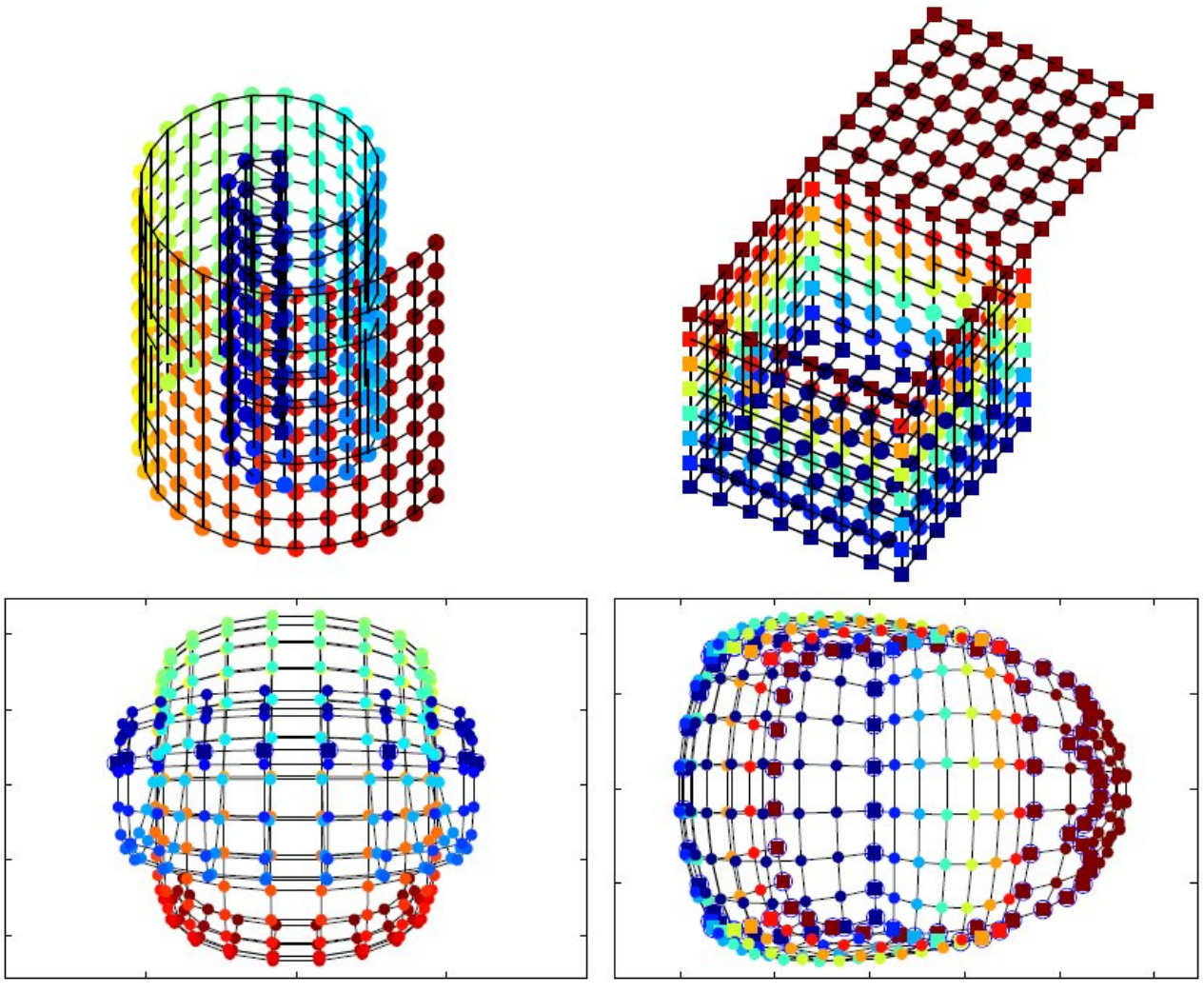
Drawbacks

- As already mentioned, KPCA gives no guarantees of the kernel's abilities.
- In fact, KPCA might as well increase the data's dimension even from D to N.
- This happens in the following examples.
- The first six eigenvalues combined don't contribute even half of the variance in either of the examples!

Drawbacks

- As already mentioned, KPCA gives no guarantees of the kernel's abilities.
- In fact, KPCA might as well increase the data's dimension even from D to N.
- This happens in the following examples.
- The first six eigenvalues combined don't contribute even half of the variance in either of the examples!

KPCA example



Outline

Semidefinite embedding
Evolving KPCA

- KPCA's problem was that its impossible to know the correct kernel and its parameters.
- SDE's idea is to learn a suitable kernel function i.e. the Gram matrix from the data itself.
- SDE applies realism in forgetting about preserving large distances and concentrates on trying to achieve local isometry.
- This requires the manifold to be smooth enough for isometries to be possible.

Constructing a local isometry

- In practice the local isometry means creating cliques to the adjacency graph.
- Construct a clique of size $K+1$ for each point and its K neighbours.
- Let $\mathbf{A}$ be the adjacency matrix of the graph. Then we require that $\mathbf{A}_{ij} = 1 \implies ||x_i - x_j||_2 = ||y_i - y_j||_2$.
- Our objective becomes maximizing

$$\phi = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} d_x^2(i,j)$$

while taking care of the requirement above.

Simplifying the problem

- This is still too difficult, but it can be simplified by making use of the scalar products.
- Let $\mathbf{S}$ be the Gram matrix constructed from the δ_y matrix, and $\mathbf{K}$ the Gram matrix in the latent space.
- Now we need to have $\mathbf{A}_{ij} = 1 \implies \mathbf{S}_{ij} = \mathbf{K}_{ij}$
- Solution is unique up to translations. Requiring $\sum_i x_i = 0$ can also be stated as $\sum_{i,j} \mathbf{K}_{ij} = 0$.
- When remembering the null sum of $\mathbf{K}$, objective ϕ actually reduces to $\mathrm{tr}(\mathbf{K})$ after some manipulation.

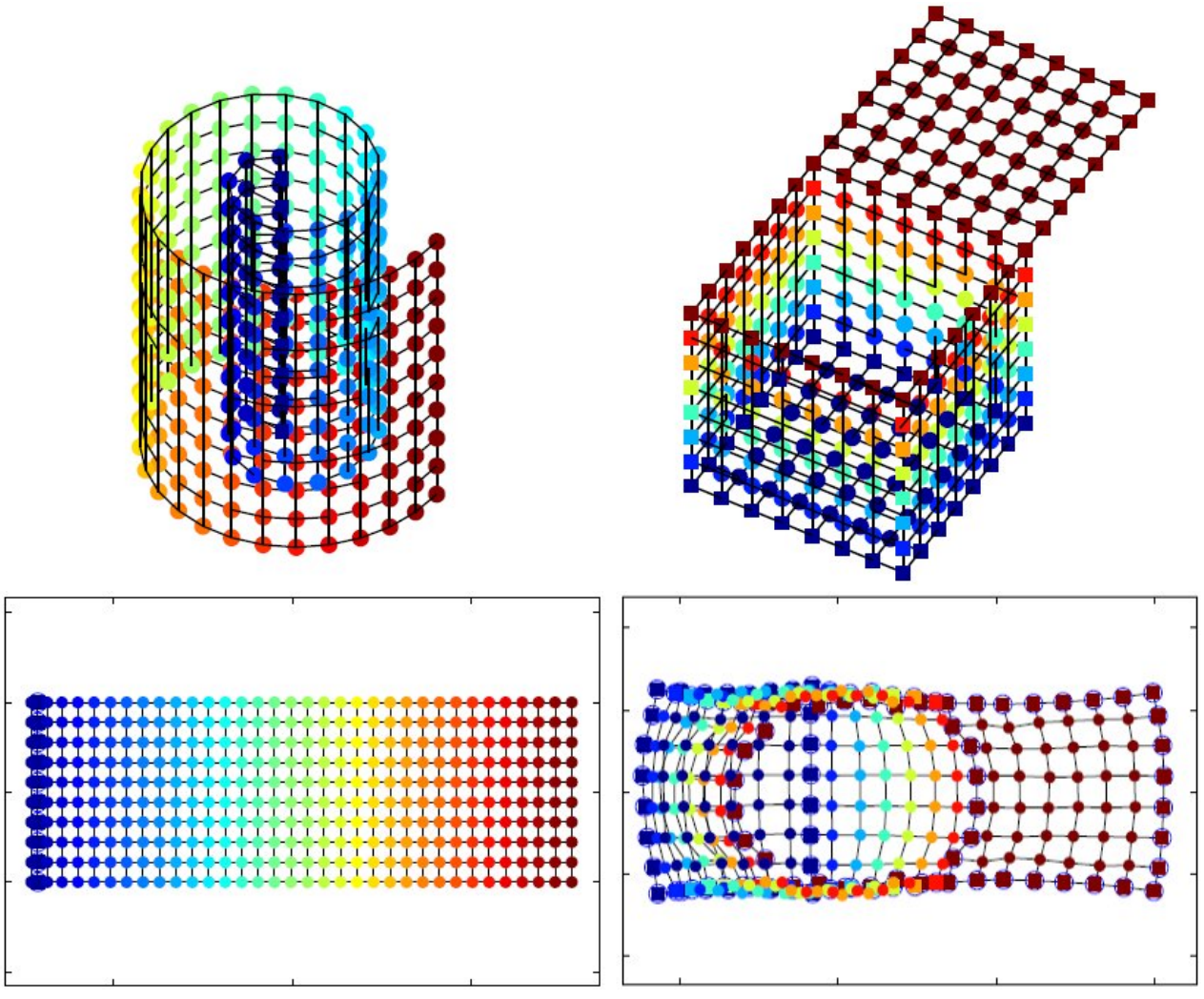
Final solution

- So our problem is:
- Find $\arg\max_{\mathbf{K}} \operatorname{tr}(\mathbf{K})$, where
  1. $\mathbf{K}$ is symmetric and positive-definite (it is a Gram matrix),
  2. $\sum_{i,j} \mathbf{K}_{ij} = 0$ and
  3. $\mathbf{A}_{ij} = 1 \implies \mathbf{S}_{ij} = \mathbf{K}_{ij}$.
- This looks really bad, but actually can be solved with semidefinite programming (SDP) (although computations are big).
- Actually the target function is even convex and bounded.
- Finally we do EVD: $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and output $\hat{\mathbf{X}} = \mathbf{I}_{P\times N}\mathbf{\Lambda}^{1/2}\mathbf{U}^T$.

SDE properties

- As SDE is a close cousin to KPCA, it is just basic MDS in steroids.
- SDE is slow, the trace optimization takes lots of time and space. Vector quantization is one solution.
- The kernel function is never revealed.
- It is possible to incrementally select the embedding dimension D.
- SDE and local isometry vs. Isomap and full isometry. Macrostructure is better preserved in SDE.

SDE example



Summary

- The distances can be calculated as if taken from a perfect embedding to a lower dimension.
- Geodesic and graph distances try to achieve this.
- Still, the methods have few new ideas. All of them depend on MDS or NLM.
- Methods are successful in unwrapping manifolds, other structures and noise cause problems.