Proximity graphs for clustering and manifold learning

Miguel Á. Carreira-Perpiñán

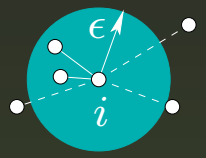
Dept. of Computer Science & Electrical Engineering, OGI/OHSU

http://www.cse.ogi.edu/~miguel

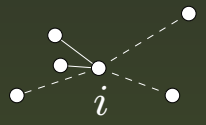
Methods based on pairwise distances

- Consider a cloud of data points $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^D$.
- We want to learn statistical structure based on the pairwise distances $\{d_{ij}\}_{i,j=1}^N$.
  This implies a graph with vertices $\{\mathbf{x}_i\}_{i=1}^N$.
- Examples:
  - Dimensionality reduction: preserve metric information (implied by the graph) in low-dimensional space.
  - Clustering: partition the graph to optimize a cut criterion.
  - Graph priors: learn functions (e.g. for regression) that are smooth on the graph.
- The graph should represent the low-dimensional manifold of the data. Thus, points are locally connected.
- Advantage: flexible representation (model-free).
- Disadvantage: computational complexity is at least $\mathcal{O}(N^2)$.

Some proximity graphs

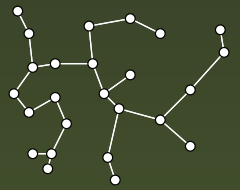


ϵ-ball graph: $i \sim j$ iff $j \in B(i;\epsilon)$

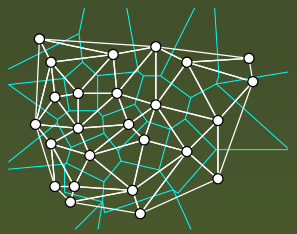


k-nearest-neighbours graph (k-NNG): $i \sim j$ iff j is one of the k nearest neighbours of i.

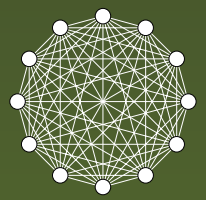
Variations: mutual k-NNG, etc.



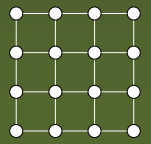
Minimum spanning tree (MST): tree subgraph that contains all the vertices and has a minimum sum of edge weights



Delaunay triangulation (DT): dual of the Voronoi tessellation

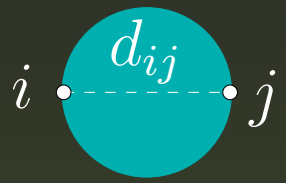


Complete graph

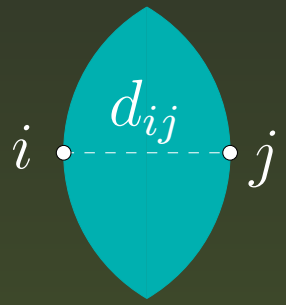


Fixed grid (in image applications)

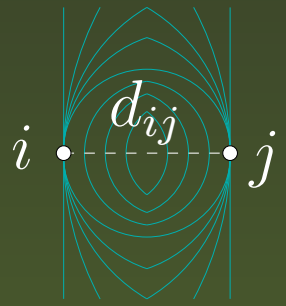
Some proximity graphs (cont.)



Gabriel graph (GG): $i \sim j$ iff no vertex in $B\left(\frac{i+j}{2}; \frac{d_{ij}}{2}\right)$



Relative neighbourhood graph (RNG): $i \sim j$ iff no vertex in $B\left(i; d_{ij}\right) \cap B\left(j; d_{ij}\right)$

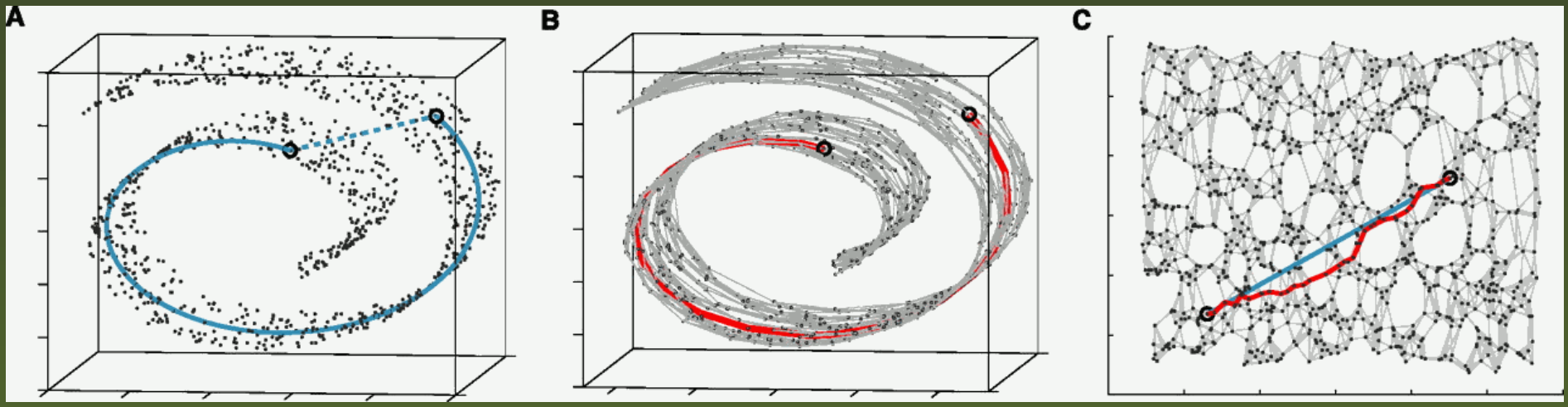


β-skeleton: $i \sim j$ iff no vertex in
$B\left(\left(1-\frac{\beta}{2}\right) i + \frac{\beta}{2} j; \frac{\beta}{2} d_{ij}\right) \cap B\left(\left(1-\frac{\beta}{2}\right) j + \frac{\beta}{2} i; \frac{\beta}{2} d_{ij}\right)$.
$\beta = 1$: GG; $\beta = 2$: RNG

	2D	Higher dimension
Complexity	$\mathcal{O}(N \log N)$	approximately $\mathcal{O}(N^2)$
# edges	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$ (except MST)
Edge set	MST $\subset$ RNG $\subset$ GG $\subset$ DT	

Dimensionality reduction: Isomap

1. Build proximity graph (ϵ-ball or k-NNG)
2. Approximate geodesic distances $\{\hat{g}_{ij}\}_{i,j=1}^{N}$ by shortest-path lengths in the graph: $\mathcal{O}(N^3)$
3. Use multidimensional scaling to obtain low-dimensional points $\{\mathbf{y}_i\}_{i=1}^{N}$, such that the Euclidean distances $\|\mathbf{y}_i - \mathbf{y}_j\|$ optimally preserve the geodesic ones: $\mathcal{O}(N^3)$



Related methods: MDS, LLE, Laplacian eigenmaps, SDE, etc.

Dimensionality reduction (cont.)

In general, specifying the neighborhoods in LLE presents the practitioner with an opportunity to incorporate a priori knowledge about the problem domain.

Saul & Roweis: “Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds”, J. Machine Learning Research (2003)

Clustering: normalized cuts & spectral clustering

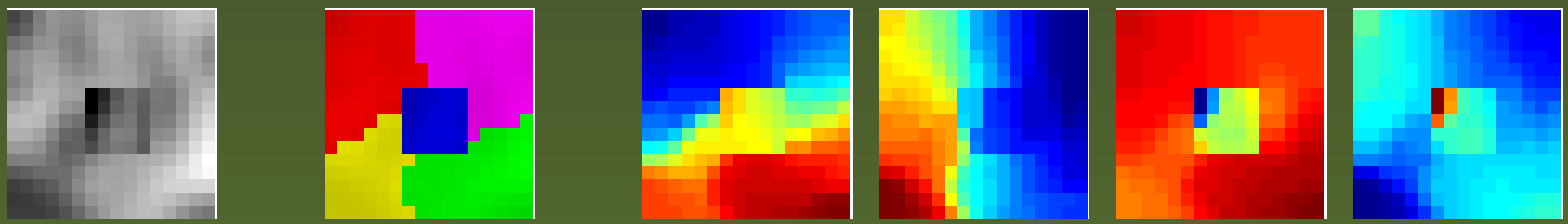
1. Build proximity graph (fully-connected graph or, for image segmentation, fixed-grid)
2. Obtain affinities $w_{ij} = \exp\left(-\frac{1}{2}(d_{ij}/\sigma)^2\right)$ for a "good" scale σ (needs search over σ)
3. Cluster the leading eigenvectors of $\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$: $\mathcal{O}(N^3)$ (where $\mathbf{D} = \mathrm{diag}\left(\sum_i w_{ij}\right)$)

This approximates the normalized cut cost function

$$\mathrm{ncut}(A, B) = \mathrm{cut}(A, B)\left(\frac{1}{\mathrm{vol}\, A} + \frac{1}{\mathrm{vol}\, B}\right)$$

which is NP-complete to optimize.

Original Segmentation First 5 eigenvectors (except the constant one)



Related methods: mincut, typical cut, etc.

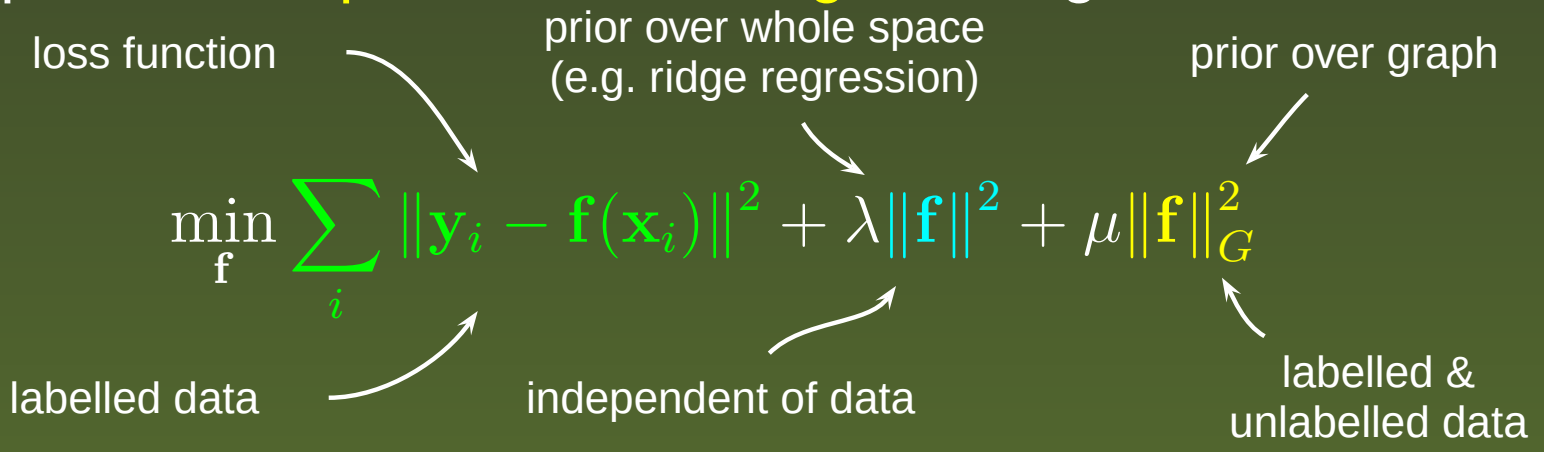
Graph priors

- A function on the graph $\mathbf{f}(\mathbf{x})$ takes values for the vertices $\mathbf{x}_1, \ldots, \mathbf{x}_N$.
- Graph prior:

$$\|\mathbf{f}\|_G^2 = \mathbf{f}^T \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{i \sim j} w_{ij} \, \|\mathbf{f}(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2$$

  where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian. It can be used to penalise functions $\mathbf{f}$ that are not smooth wrt the graph.

- Example: semisupervised learning. For regression:



  Likewise for density estimation, classification. . .

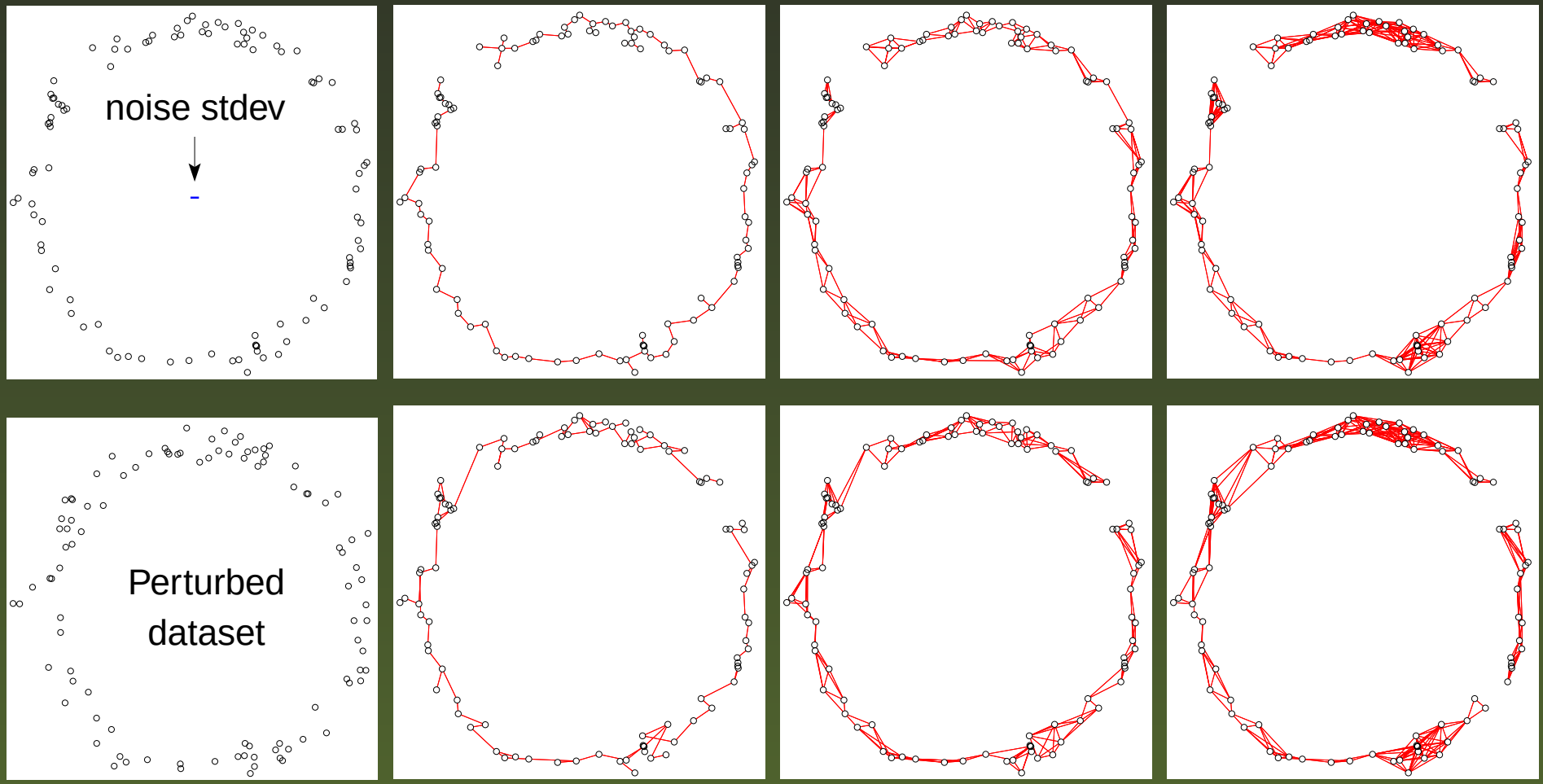
Problems with ϵ-ball, k-NNG and other graphs

- The graph parameter ϵ or k has to be chosen carefully to avoid:
  - connecting the wrong points (shortcuts)
    which underestimates geodesic distances
  - not connecting right points (disconnected graph, gaps)
    which overestimates geodesic distances

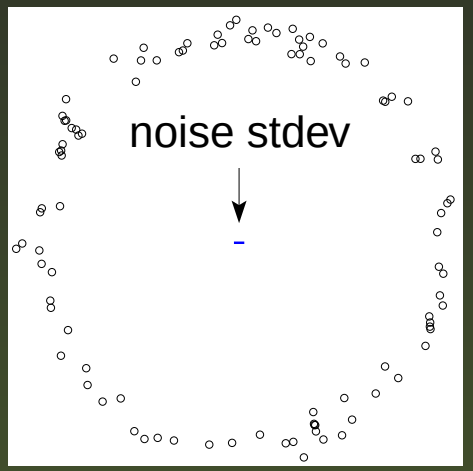
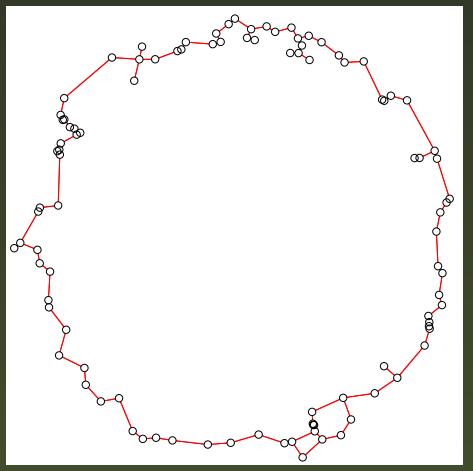
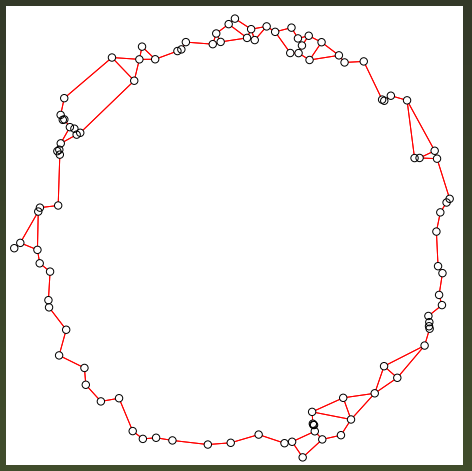
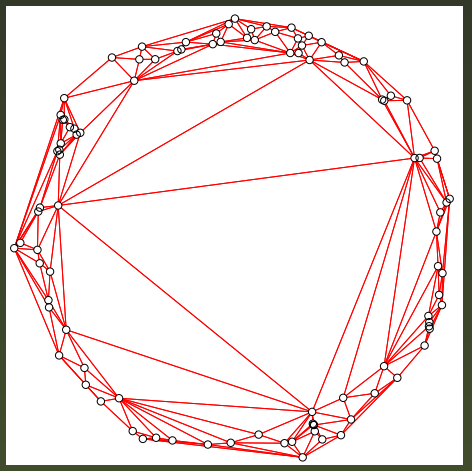
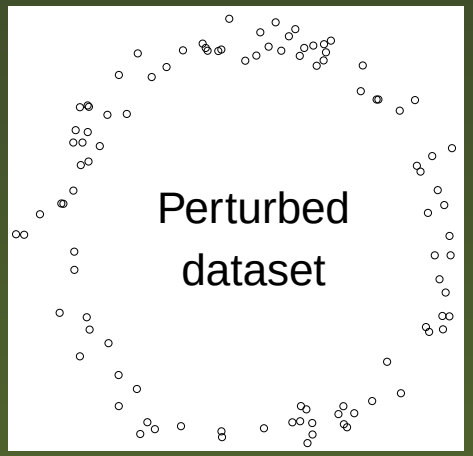
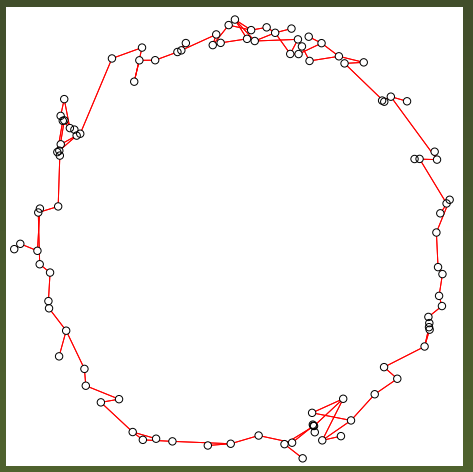
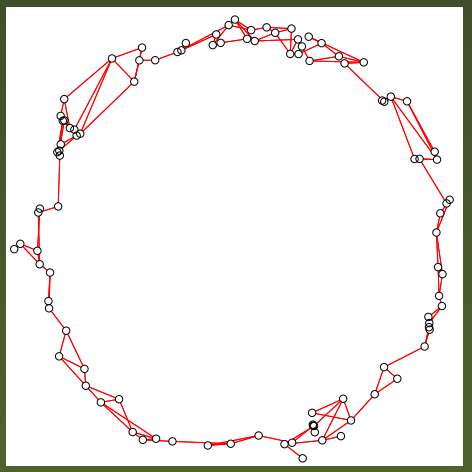
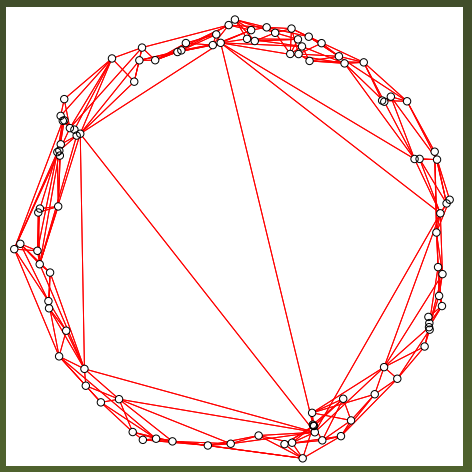
  We also need to search over the scale σ or dimensionality.
- The local neighborhoods are not adaptive
  ϵ, k should depend on $\mathbf{x}_i$
- The graphs are sensitive to small perturbations of the data
  The original data are noisy
- Other types of graphs connect points nonlocally
  e.g. Delaunay triangulation, relative neighborhood graph, Gabriel graph

After building the graph, the subsequent spectral algorithm is $\mathcal{O}(N^3)$, thus trial-and-error of graphs is very expensive.

Sensitivity to noise of proximity graphs



Sensitivity to noise of proximity graphs (cont.)

Dataset	Relative neigh-bourhood graph	Gabriel graph	Delaunay triangulation
			
			

The minimum spanning tree (MST)

An MST is a tree subgraph that contains all the vertices and has a minimum sum of edge weights. It can be computed in $\mathcal{O}(N^2 \log N)$ for N vertices, e.g. using Kruskal's algorithm.

Good properties as skeleton of a data set:

- avoids shortcuts between manifold branches (typically caused by long edges)
- gives connected graph

But:

- too sparse ($N - 1$ edges if N points, and no cycles)
- sensitive to noise

One way to flesh out the MST and attain robustness to noise is to form an MST ensemble that combines multiple MSTs.

Two new types of proximity graphs (1)

Perturbed minimum spanning trees (PMSTs):

1. Estimate local noise model for each data point: uniform zero-mean isotropic with standard deviation rd_i
   $d_i =$ average distance to the k nearest neighbors of $\mathbf{x}_i$ and $r \in [0, 1]$
2. Generate T jittered copies of the entire data set according to this noise
3. For each copy, build its MST
4. Average all MSTs

Result:

- Stochastic graph with edges $e_{ij} \in [0, 1]$
- Number of edges is linear in N
- Insensitive to noise by construction
- Essentially deterministic for large T

Two new types of proximity graphs (2)

Disjoint minimum spanning trees (DMSTs):

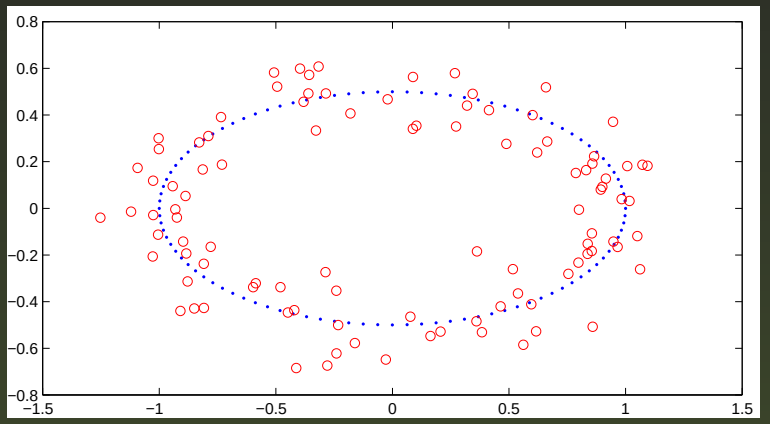
- Deterministic collection of t MSTs such that the nth tree (for $n = 1, \ldots, t$) is the MST of the data subject to not using any edge already in the previous $1, \ldots, t-1$ trees.
- Construction algorithm:
  1. Sort edge list by increasing distance d_{ij}
  2. Run Kruskal's algorithm t times by picking edges without replacement

Result:

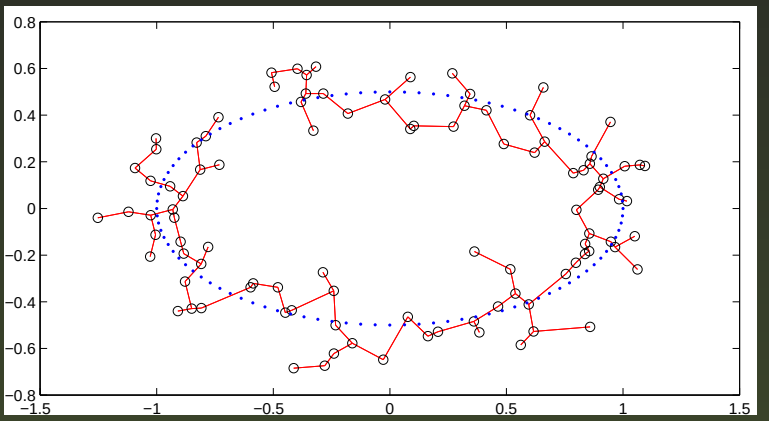
- Binary graph with edges $e_{ij} \in \{0, 1\}$
- Number of edges is linear in N
- Relatively insensitive to noise
- Deterministic

Two new types of proximity graphs (cont.)

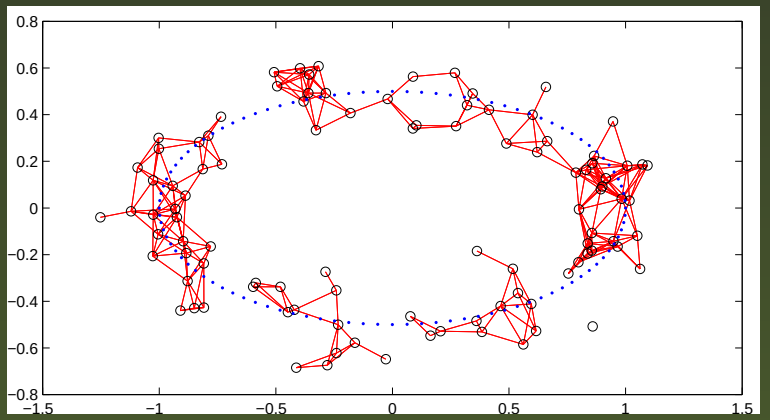
Dataset



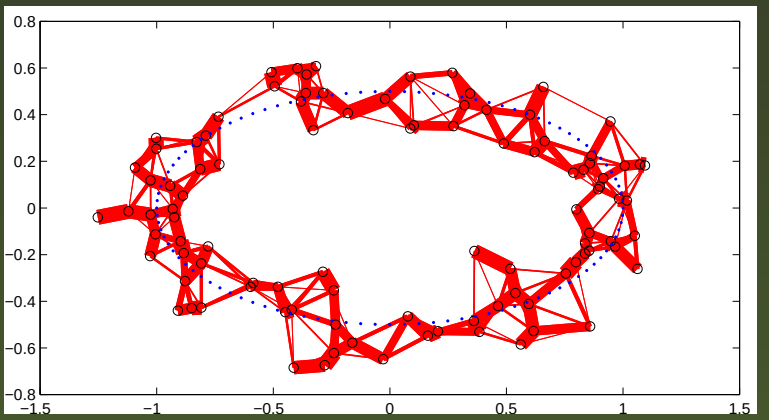
MST



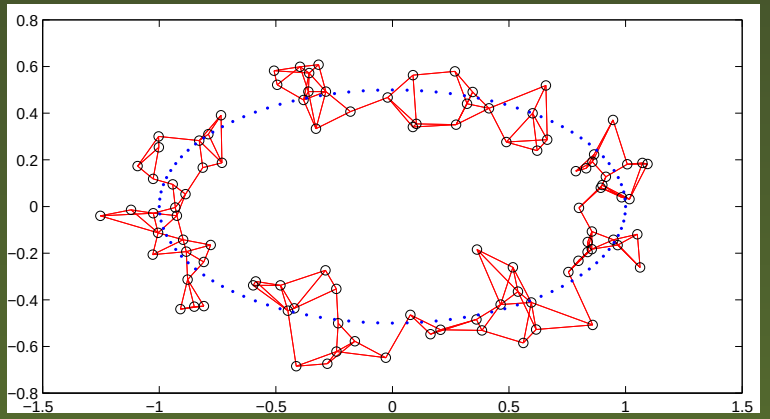
ϵ-ball



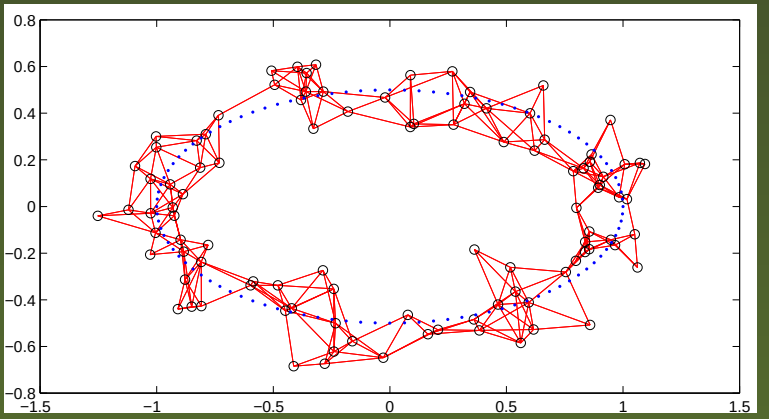
PMSTs



k-NNG



DMSTs



Two new types of proximity graphs (cont.)

An ensemble of MSTs gives a good representation of the manifold:

- Each MST uses short edges, thus avoiding shortcuts, and gives a good skeleton of the data
- Each MST is very sparse, but the combination fleshes out the graph

Computational complexity:

- PMSTs: $\mathcal{O}(TN^2 \log N)$
- DMSTs: $\mathcal{O}(N^2(\log N + t))$

This is:

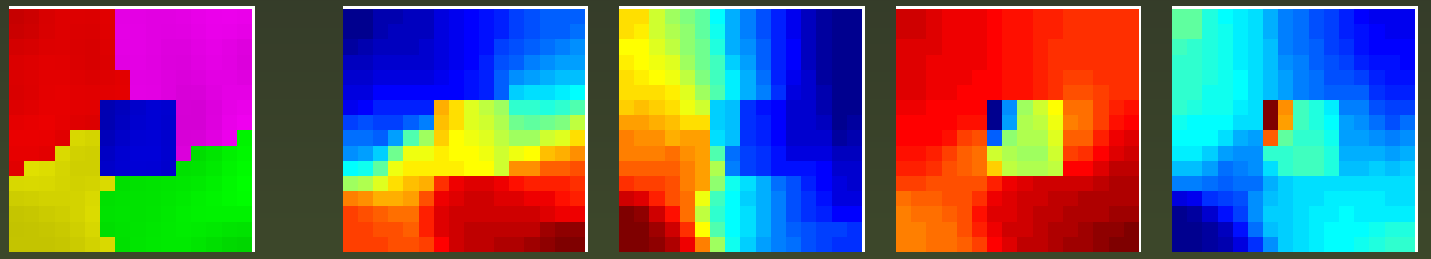
- Just a bit more than searching for nearest neighbors
- Much less than the subsequent $\mathcal{O}(N^3)$ spectral algorithm

Results: spectral clustering (normalized cut)

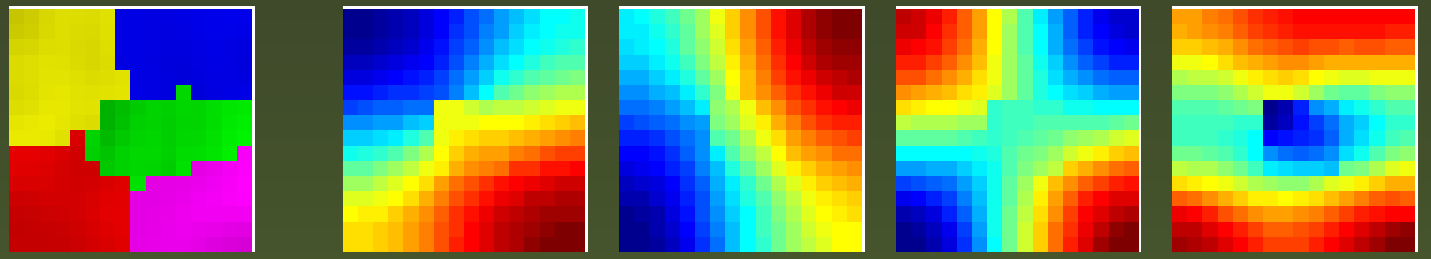
Graph: 8-connected grid; affinities $w_{ij} = \exp\left(-\frac{1}{2}(d_{ij}/\sigma)^2\right) \in [0, 1)$

Segmentation — First 5 eigenvectors (except the constant one)

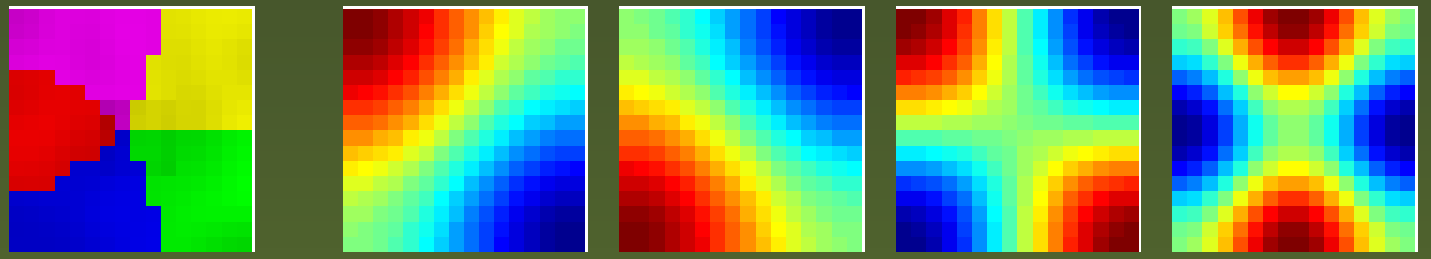
$\sigma_1 = 0.5$



$\sigma_1 = 1.6$



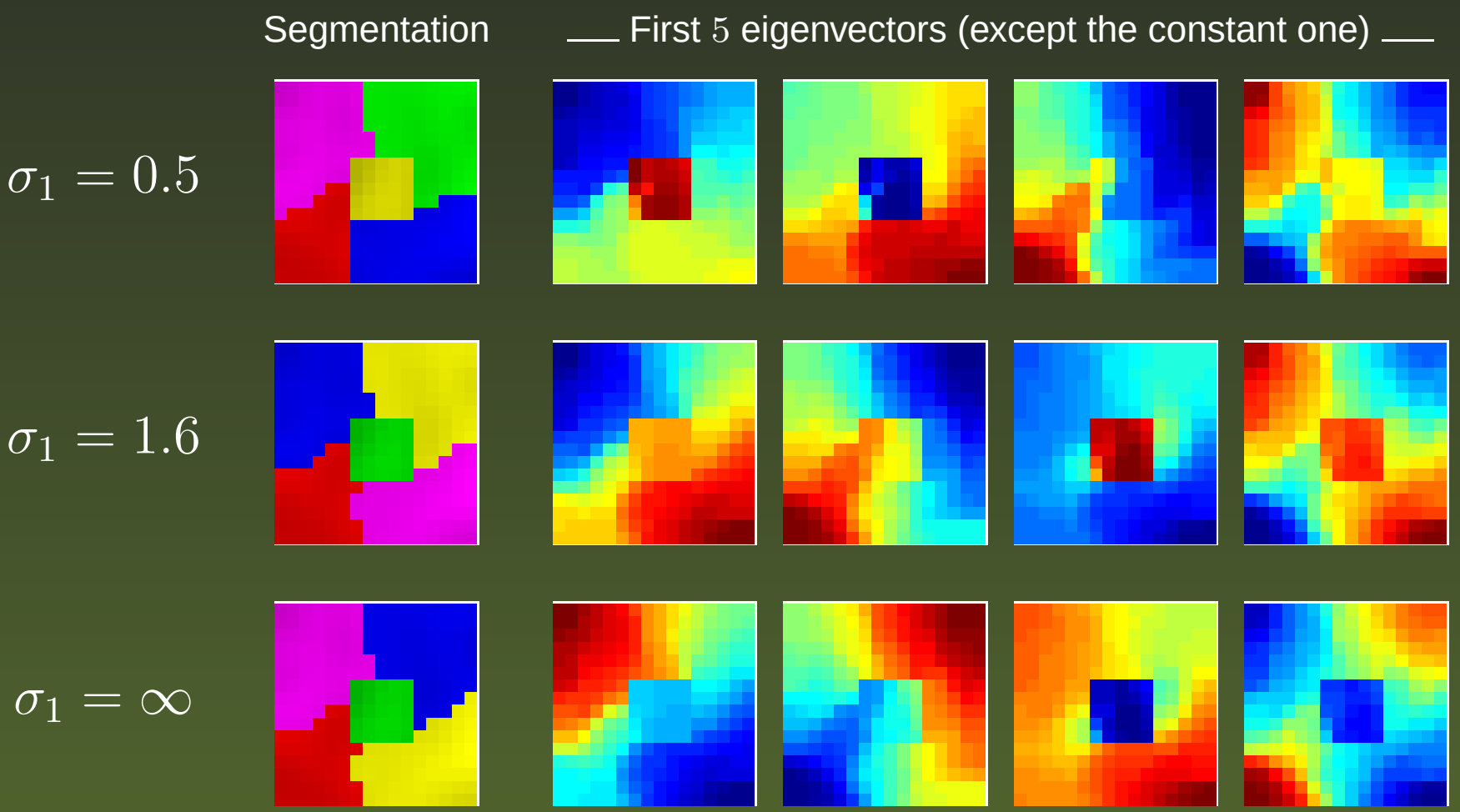
$\sigma_1 = \infty$



Good segmentations only for $\sigma \in [0.2, 1]$ approximately.

Results: spectral clustering (cont.)

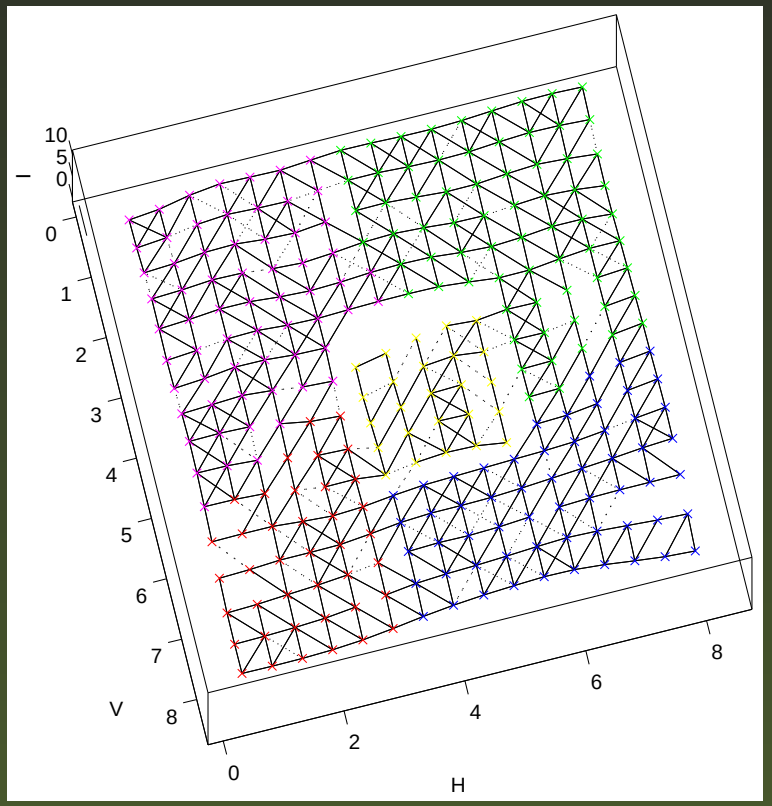
Graph: PMSTs, $r = 0.4$; affinities $e_{ij}w_{ij} \in [0, 1)$



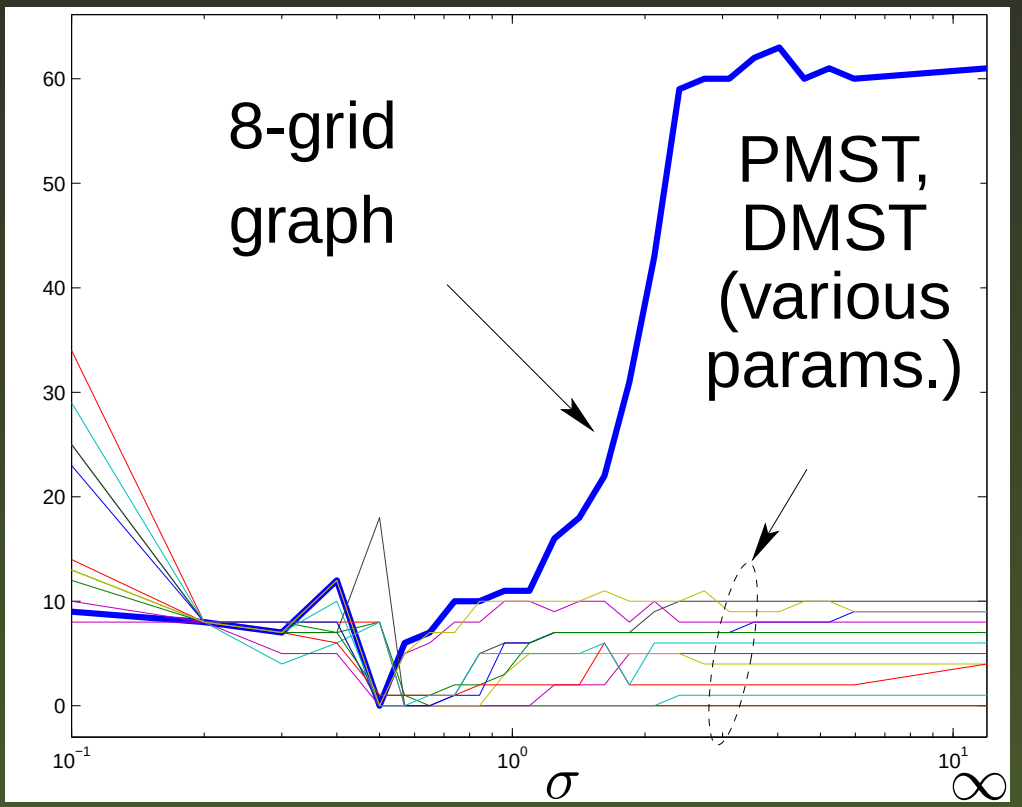
Good segmentations for $\sigma \in [0.2, \infty]$ approximately.

Results: spectral clustering (cont.)

PMSTs ensemble (3D view)



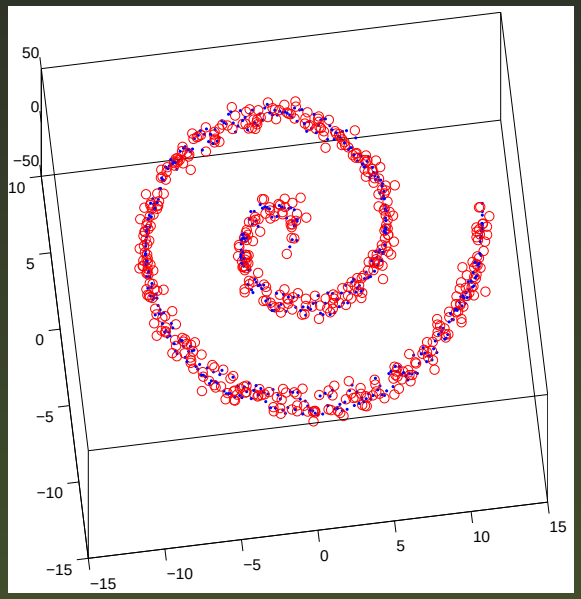
Clustering error over scales σ



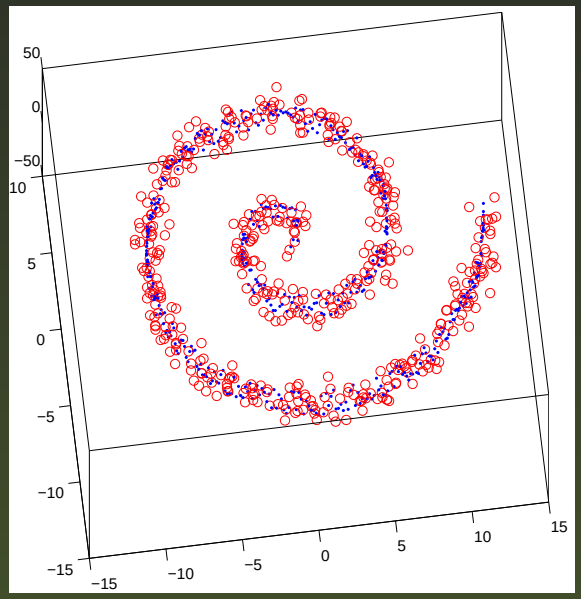
Both PMSTs and DMSTs produce good segmentations for very large σ under a wide range of parameters, because they represent the data manifold better and so facilitate the graph cut. Thus, having a good graph can eliminate an expensive search over scales (each σ value costs $\mathcal{O}(N^3)$).

Results: dimensionality reduction (Isomap)

Swiss roll, low noise



Swiss roll, high noise



We examine the preservation of geodesic distances for several graph types (PMSTs: binarize e_{ij}):

- Average error in the geodesic distances $E = \frac{1}{N^2} \|\hat{\mathbf{G}} - \mathbf{G}\|$
- Isomap's estimated residual variance $\hat{V} = 1 - R^2(\hat{\mathbf{G}}, \mathbf{D}_\mathcal{Y})$
- True residual variance $V = 1 - R^2(\mathbf{G}, \mathbf{D}_\mathcal{Y})$

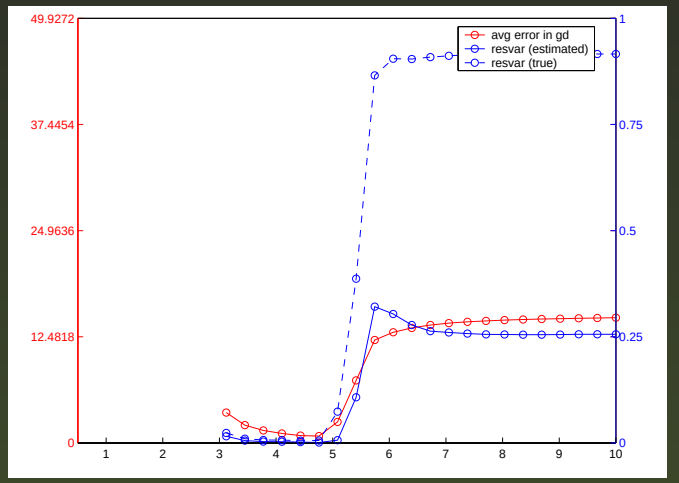
$\mathbf{G}$: true geodesic distances

$\hat{\mathbf{G}}$: estimated (graph shortest-path) geodesic distances

$\mathbf{D}_\mathcal{Y}$: Euclidean distances in the low-dimensional embedding

Results: Isomap, low noise (cont.)

ϵ-ball

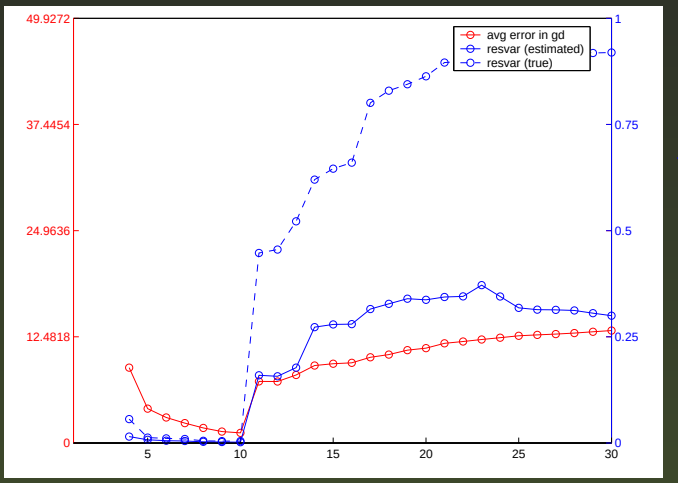


E

V, $\hat{V}$

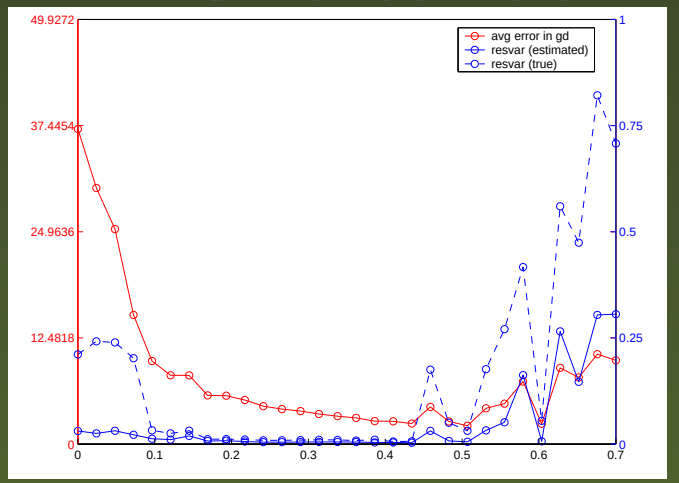
ϵ

k-NNG



k

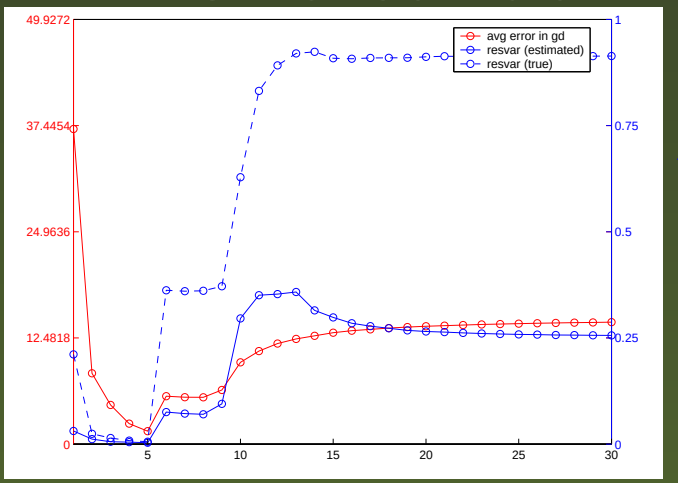
PMST ensemble



E

r

DMST ensemble

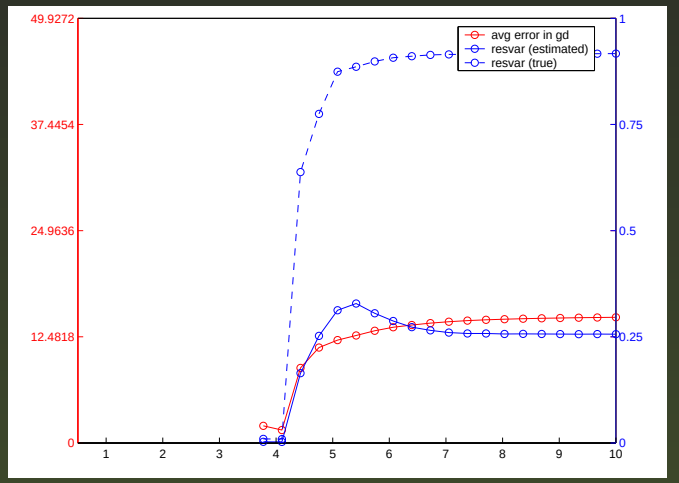


V, $\hat{V}$

t

Results: Isomap, high noise (cont.)

ϵ-ball

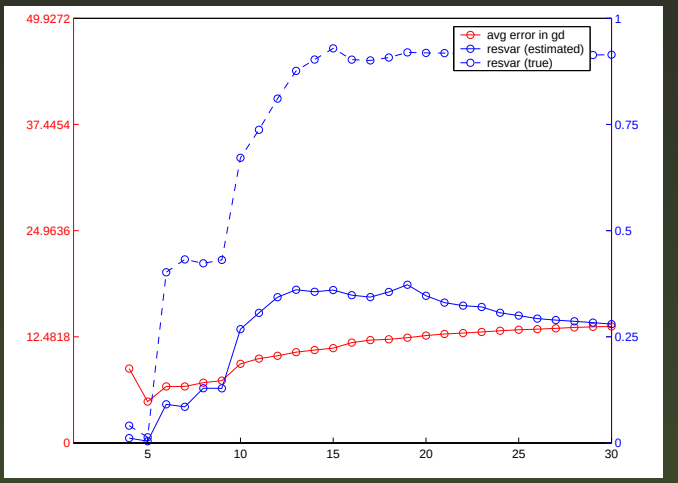


E

V, $\hat{V}$

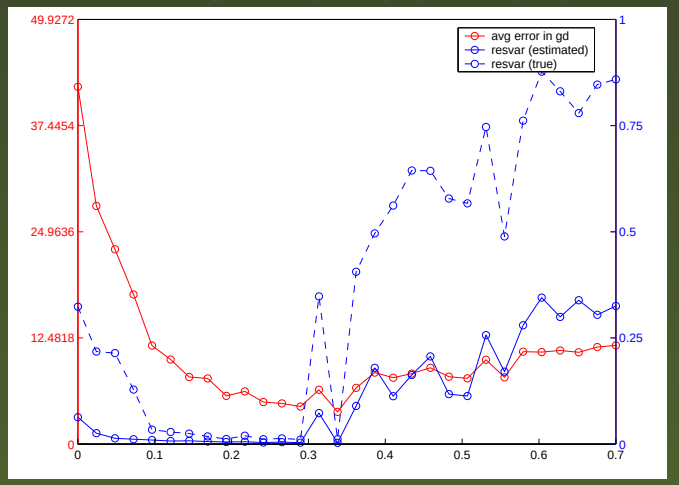
ϵ

k-NNG



k

PMST ensemble

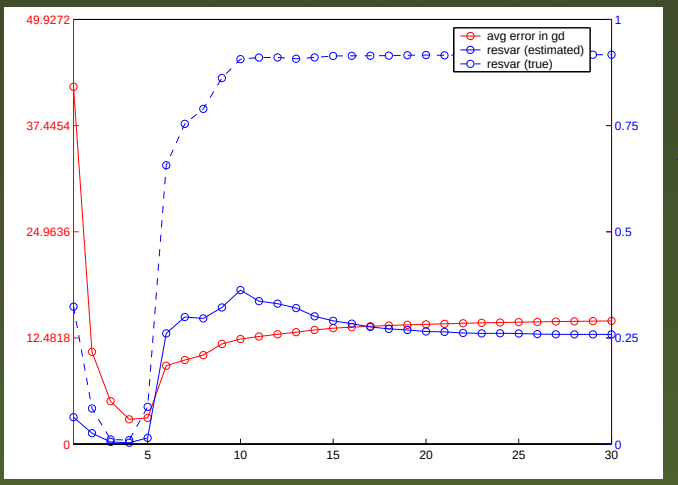


E

V, $\hat{V}$

r

DMST ensemble



t

The PMST and DMST ensembles are more robust than ϵ-ball or k-NNG, particularly for high noise (note the narrow range of good ϵ and k). Again, this eliminates an expensive search (over ϵ or k).

Conclusions

- Pointed out the problem of learning proximity graphs:
  - as scaffolds for clustering, manifold learning & graph priors
  - the graph should represent the structure of the data manifold
- Introduced two new types of proximity graphs based on ensembles of MSTs:
  - not expensive to compute
  - robust across many noise levels and parameter settings
  - limit the required parameter search for clustering & manifold learning

Conclusions (cont.)

- No objective function for unsupervised graph learning
- The MST ensembles tend to reduce both bias and variance of the average error for the geodesic distances (if known a priori)
- Future work:
  - Manifold-aligned noise model
  - Noise model for the similarities (non-Euclidean data)
  - Study stochastic graphs
  - Fast algorithms to find (approximate) nearest neighbours