Manifold Blurring Mean-Shift Algorithms for Manifold Learning

Weiran Wang

wwang5@ucmerced.edu

EECS Department, UC Merced

Introduction to Manifold Learning

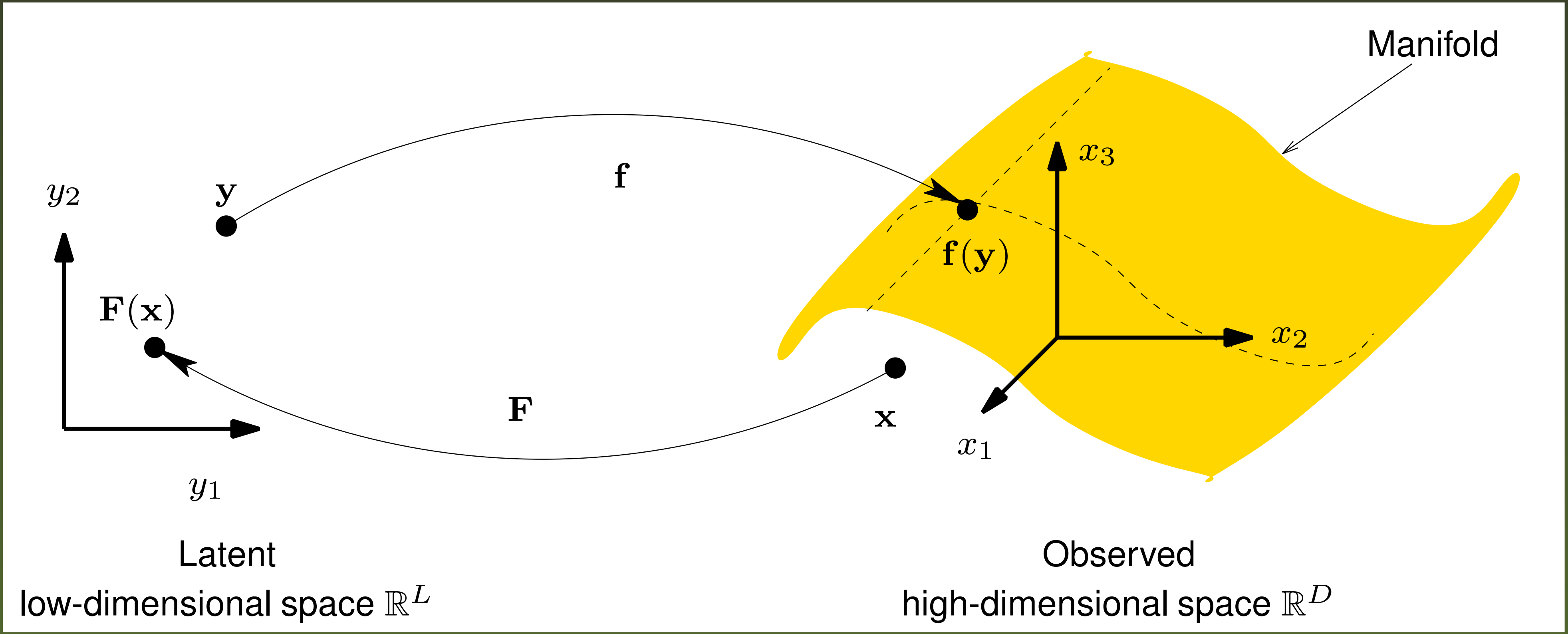
- Machine learning algorithms often take as starting point a high-dimensional dataset, and then learn a model that is useful to infer information from this data, or from unseen data.

- Manifold Learning / Dimensionality Reduction (DR) algorithms deal with data set that has manifold structure.
  - Variations within the data set can be modeled by a few latent variables.
  - Small variation in (low dimensional) latent space leads to small variation in (high dimensional) data space.
  - Mapping from latent space to data space can be highly nonlinear.

Problem setting of DR algorithms

Suppose input data points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are sampled from $\mathbb{R}^D$. A DR algorithm provides the corresponding low dimensional representations $\mathbf{y}_1, \ldots, \mathbf{y}_N \in \mathbb{R}^L (L \ll D)$, and may give

- Dimensionality reduction mapping $\mathbf{y} = \mathbf{F}(\mathbf{x}), \ \mathbf{x} \in \mathbb{R}^D$
- Reconstruction mapping $\mathbf{x} = \mathbf{f}(\mathbf{y}), \ \mathbf{y} \in \mathbb{R}^L$



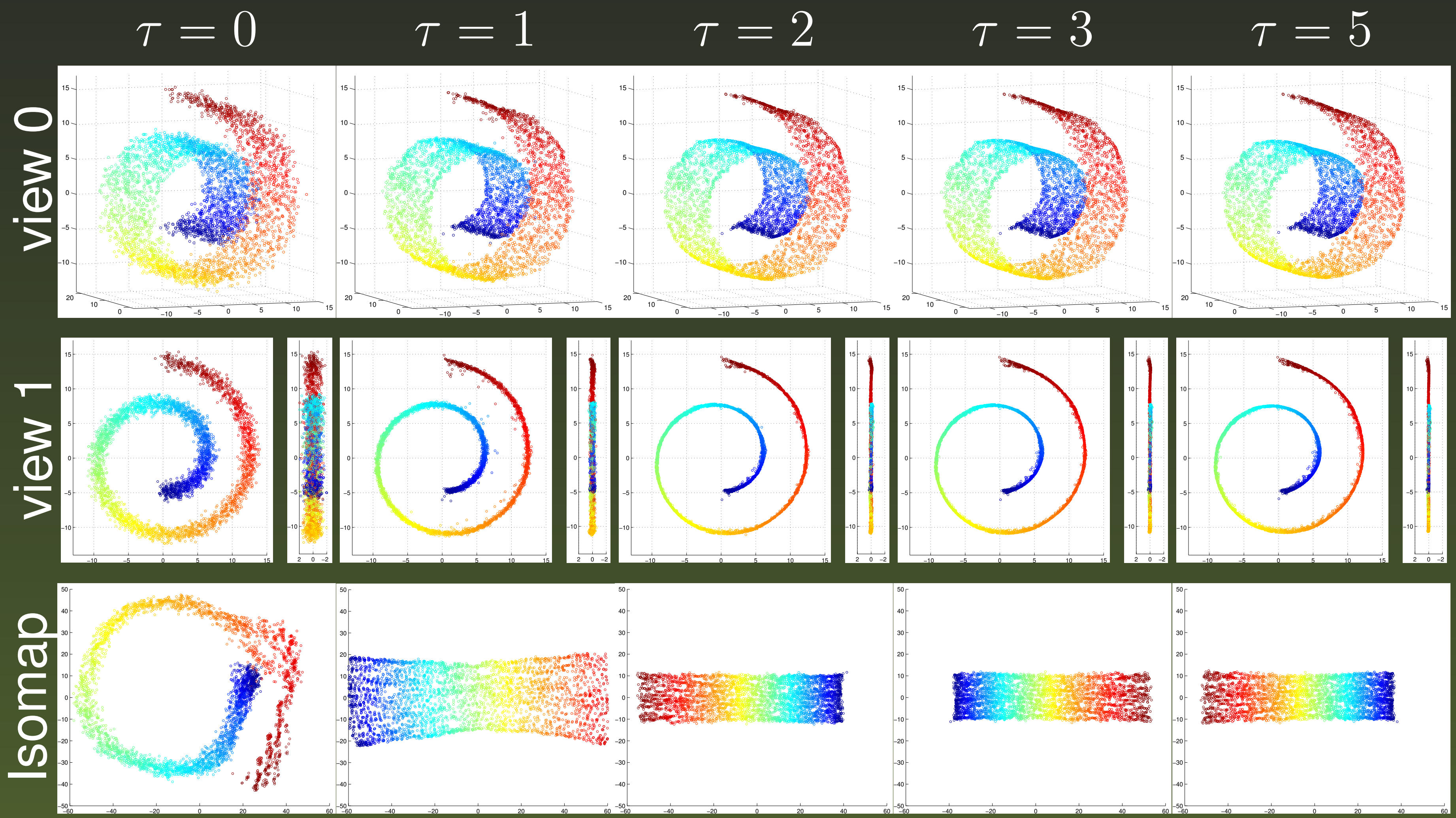
Problem with the existence of noise

DR algorithms do not work well when the input data set contains noise and outliers.

- Spectral methods (Isomap, LLE, etc) are quite sensitive to noise, especially the step of building neighborhood graph.
- Latent Variable Models (e.g. mixtures of probabilistic PCAs) try to learn a parametric model of the manifold and noise by maximum likelihood, but are prone to local optima.

Thus it is desirable to develop an algorithm that denoise the data set, and acts as a preprocessing step for other purposes (unsupervised learning, supervised learning, etc).

Denoising as preprocessing: an example

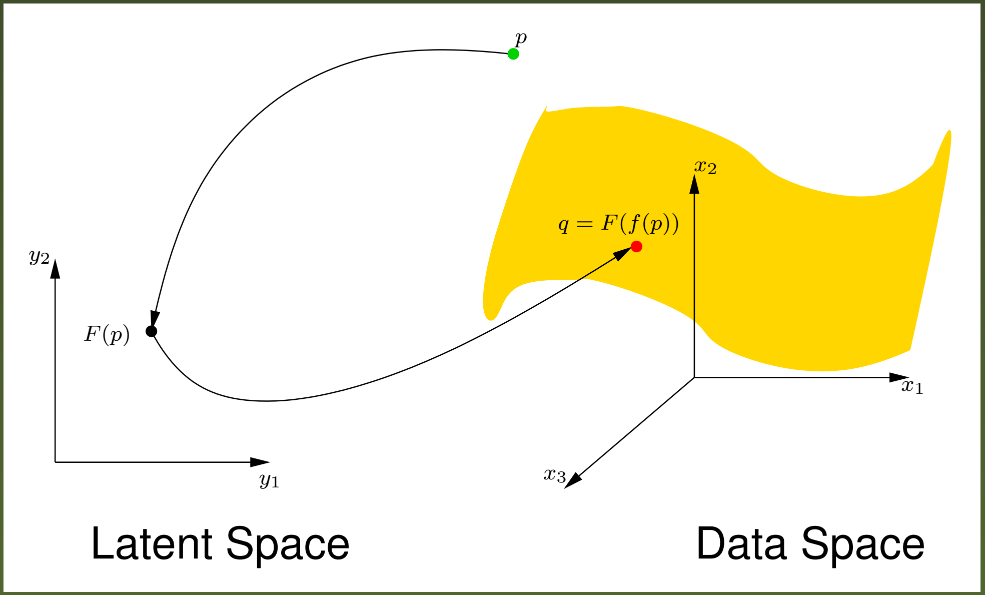


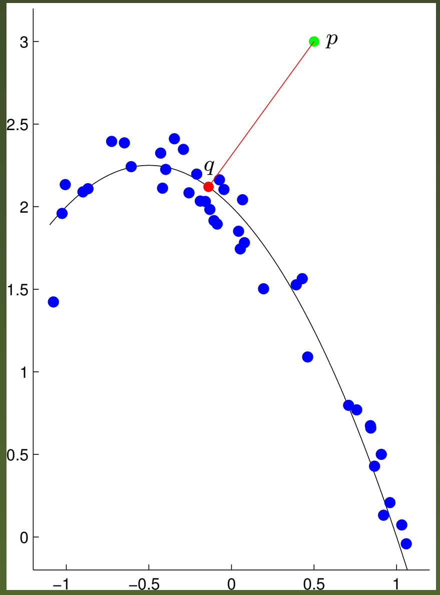
Related Work: Machine Learning Algorithms

Given some point $\mathbf{p} \in \mathbb{R}^D$, after running DR algorithm, a denoised version $\mathbf{q} \in \mathbb{R}^D$ can be obtained in two ways:

1. First project $\mathbf{p}$ to the latent space, and then project its latent representation back to data space, i.e., $\mathbf{q} = \mathbf{f}(\mathbf{F}(\mathbf{p}))$. Autoencoder, Latent Variable Models, Dimensionality Reduction by Unsupervised Regression, . . .
2. Project $\mathbf{p}$ onto the closest point on manifold by minimizing $\min \|\mathbf{p} - \mathbf{f}(\mathbf{y})\|$. Principal Curves, Regularized Principal Manifolds, Gaussian Process Latent Variable Models, . . .

However, it is more desirable to denoise data set before DR.





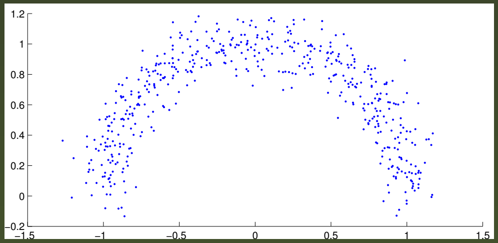
Related Work (cont.)

- Manifold Denoising in Machine Learning/Computer Vision
  - Fukunaga & Hostetler, 1975
  - Park et al, 2004
  - Unnikrishnan & Hebert, 2007
  - Hein & Maier, 2007
- Surface Smoothing in Computer Graphics
  - Taubin et al, 1996
  - Desbrun et al, 1999
  - Levin, 2003
  - Lange et al, 2005
  - Pauly et al, 2006
- Curve and Surface Reconstruction in Computational Geometry
  - Dey, 2007

Overview of Manifold Blurring Mean-Shift Algorithm

Iterative methods combining two basic steps in each iterate:

- predictor averaging step computes one step of GBMS update, responsible for denoising
- corrector projective step computes local PCA and removes the tangential component of the motion, responsible for preserving manifold structure



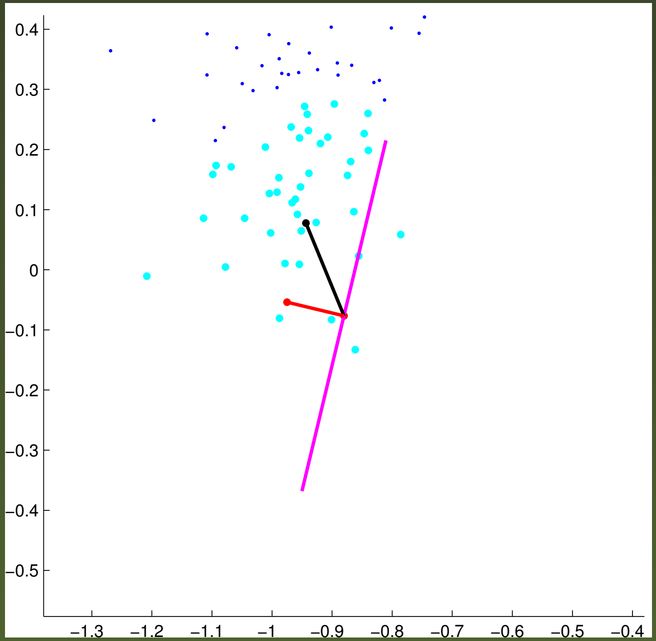
$L = 1, K = 40, \sigma = 0.2$

cyan: neighbors

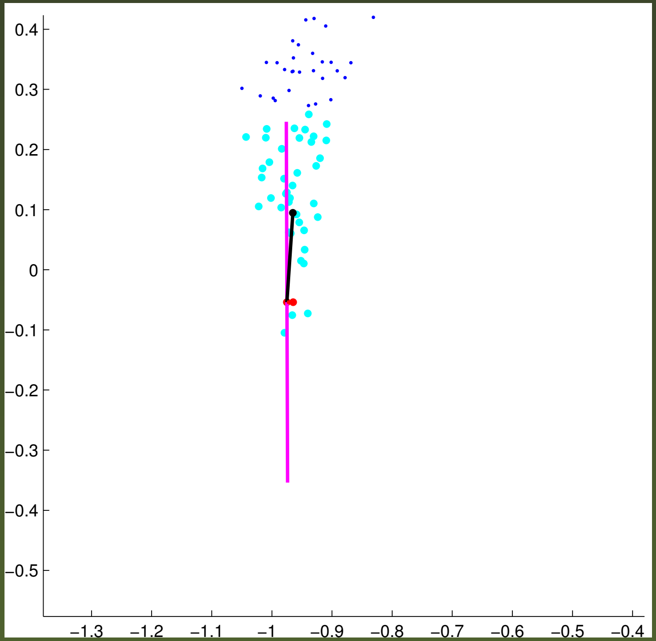
magenta: tangent

black: predictor

red: corrector



iteration 1



iteration 2

Mean-Shift algorithm (Fukunaga et al 1975, Cheng 1995)

- Kernel density estimate (constant weights, isotropic Gaussian Kernel with width σ):

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} G_\sigma(\mathbf{x} - \mathbf{x}_i)$$

- Mode seeking: find stationary points $\frac{\partial p(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{0}$ and solve for $\mathbf{x}$, obtain a fixed point iteration scheme $\mathbf{x}^{(\tau+1)} = \mathbf{f}(\mathbf{x}^{(\tau)})$.
- The motion $\mathbf{f}(\mathbf{x}) - \mathbf{x}$ is called the Mean-Shift vector.

Gaussian Mean-Shift algorithm (Carreira-Perpiñán, 2007)

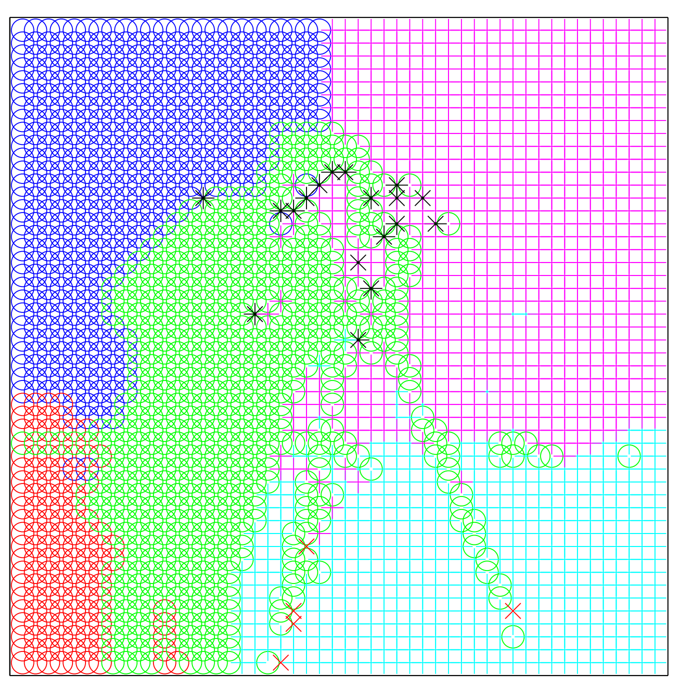
- Gaussian Mean-Shift (GMS):

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \frac{G_\sigma(\mathbf{x} - \mathbf{x}_i)}{\sum_{j=1}^{N} G_\sigma(\mathbf{x} - \mathbf{x}_j)} \mathbf{x}_i$$

- User parameter: σ.
- GMS clustering:
  - each data point is assigned to the mode it converges to
  - nonparametric, deterministic
- GMS is an Expectation Maximization algorithm (linear convergence rate in general).

Blurring Mean-Shift algorithm (Carreira-Perpiñán, 2006)

- In each Mean-Shift iteration, every data point actually moves to a weighted mean of the previous data set, and thus the whole data set gets updated.
- Robust stopping criteria.
- Convergence rate is cubic for Gaussian kernel.
- Shows denoising effect.



Manifold Blurring Mean-Shift

- Local clustering with Gaussian blurring mean shift (GBMS): the blurring mean-shift update with unit step size moves data points to the kernel average of their neighbors:

$$\mathbf{x}_n \leftarrow \sum_{m \in \mathcal{N}_n} \frac{G_\sigma(\mathbf{x}_n, \mathbf{x}_m)}{\sum_{m' \in \mathcal{N}_n} G_\sigma(\mathbf{x}_n, \mathbf{x}_{m'})} \mathbf{x}_m$$

- Local tangent space estimation with PCA: local PCA gives the best linear L-dimensional manifold in terms of reconstruction error (i.e., orthogonal projection on the manifold):

$$\min_{\boldsymbol{\mu}, \mathbf{U}} \sum_{m \in \mathcal{N}'_n} \left\| \mathbf{x}_m - (\mathbf{U}\mathbf{U}^T(\mathbf{x}_m - \boldsymbol{\mu}) + \boldsymbol{\mu}) \right\|^2$$

For simplicity, we use the same neighborhood for the GBMS step and PCA step ($\mathcal{N}_n = \mathcal{N}'_n$) in the experiments.

Variations of MBMS

MBMSf (full graph) and MBMSk (k-nn graph)

```
MBMS $(L, k, \sigma)$ with full or $k$-nn graph: given $\mathbf{X}_{N \times D}$
repeat
    for $n = 1, \dots, N$
        $\mathcal{N}_n \leftarrow \{1, \dots, N\}$ (full graph) or
                $k$ nearest neighbors of $\mathbf{x}_n$ ($k$-nn graph)
        $\partial \mathbf{x}_n \leftarrow -\mathbf{x}_n + \sum_{m \in \mathcal{N}_n} \frac{G_\sigma(\mathbf{x}_n, \mathbf{x}_m)}{\sum_{m' \in \mathcal{N}_n} G_\sigma(\mathbf{x}_n, \mathbf{x}_{m'})} \mathbf{x}_m$        mean-shift step
        $\mathcal{X}_n \leftarrow k$ nearest neighbors of $\mathbf{x}_n$
        $(\boldsymbol{\mu}_n, \mathbf{U}_n) \leftarrow \text{PCA}_L(\mathcal{X}_n)$        estimate $L$-dim tangent space at $\mathbf{x}_n$
        $\partial \mathbf{x}_n \leftarrow (\mathbf{I} - \mathbf{U}_n \mathbf{U}_n^T) \partial \mathbf{x}_n$        subtract parallel motion
    end
    $\mathbf{X} \leftarrow \mathbf{X} + \partial \mathbf{X}$        move points
until stop
return $\mathbf{X}$
```

User parameters: L, K, σ.

Variations of MBMS (cont.)

Local Tangent Projection (LTP): MBMSk with $\sigma = \infty$

```
LTP $(L, k)$ with $k$-nn graph: given $\mathbf{X}_{N \times D}$
repeat
    for $n = 1, \dots, N$
        $\mathcal{X}_n \leftarrow k$ nearest neighbors of $\mathbf{x}_n$
        $(\boldsymbol{\mu}_n, \mathbf{U}_n) \leftarrow \mathrm{PCA}_L(\mathcal{X}_n)$        estimate $L$-dim tangent space at $\mathbf{x}_n$
        $\partial\mathbf{x}_n \leftarrow (\mathbf{I} - \mathbf{U}_n\mathbf{U}_n^T)(\boldsymbol{\mu}_n - \mathbf{x}_n)$        project point onto tangent space
    end
    $\mathbf{X} \leftarrow \mathbf{X} + \partial\mathbf{X}$        move points
until stop
return $\mathbf{X}$
```

User parameters: L, K.

Variations of MBMS (cont.)

GBMS: $L = 0$

```
GBMS $(k, \sigma)$ with full or $k$-nn graph: given $\mathbf{X}_{N \times D}$
repeat
    for $n = 1, \dots, N$
        $\mathcal{N}_n \leftarrow \{1, \dots, N\}$ (full graph) or
                 $k$ nearest neighbors of $\mathbf{x}_n$ ($k$-nn graph)
        $\partial \mathbf{x}_n \leftarrow -\mathbf{x}_n + \sum_{m \in \mathcal{N}_n} \frac{G_\sigma(\mathbf{x}_n, \mathbf{x}_m)}{\sum_{m' \in \mathcal{N}_n} G_\sigma(\mathbf{x}_n, \mathbf{x}_{m'})} \mathbf{x}_m$        mean-shift step
    end
    $\mathbf{X} \leftarrow \mathbf{X} + \partial \mathbf{X}$        move points
until stop
return $\mathbf{X}$
```

User parameters: K, σ.

Complexity & Stopping Criteria

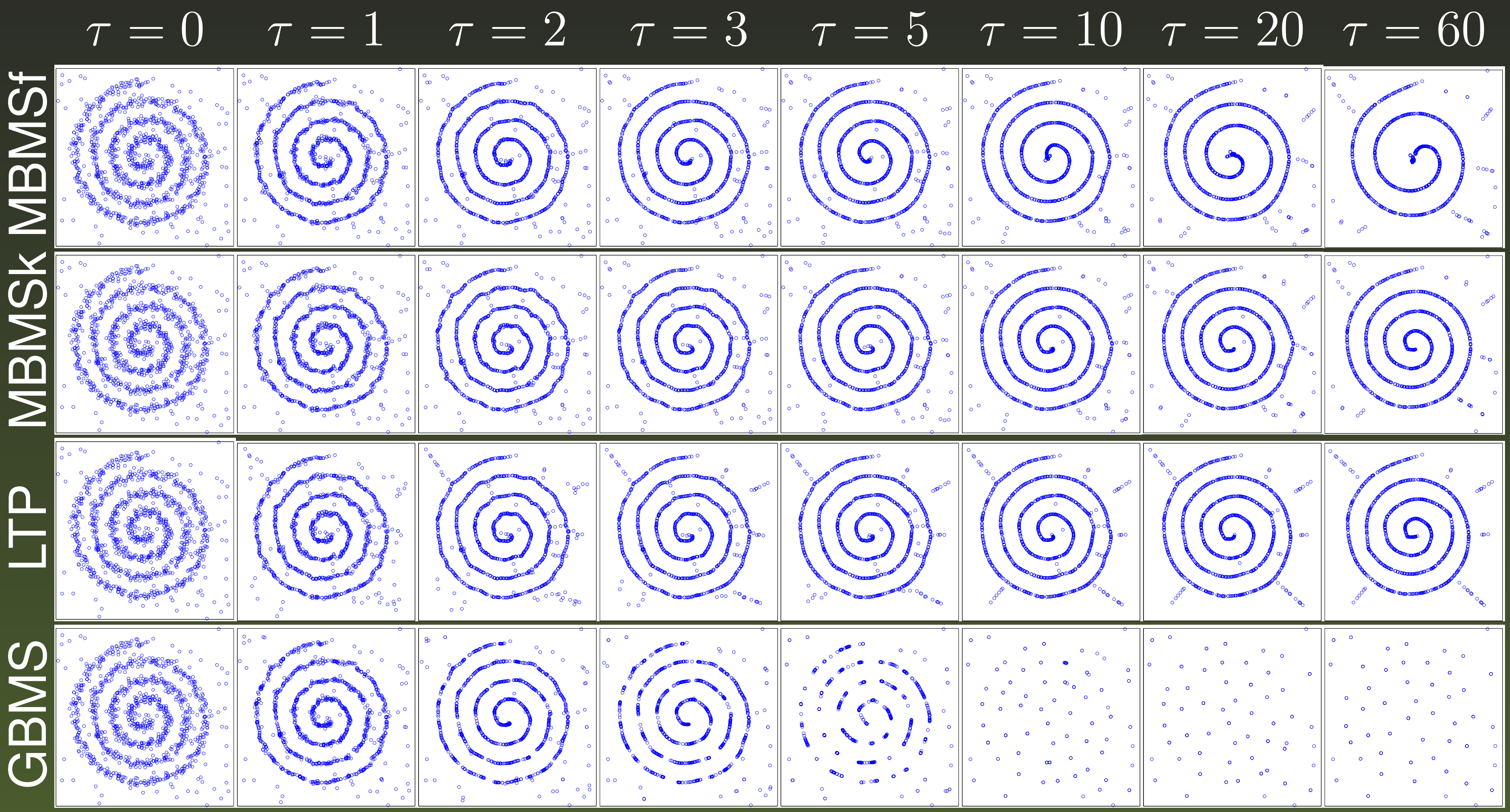
Complexity of MBMS:

- When using full graph, complexity of each iteration is $\mathcal{O}(N^2D + N(D+k)\min(D,k)^2)$. The first term is for finding nearest neighbors and for the mean-shift step, and the second is for local PCAs.
- If one uses the k-nn graph and does not update the neighbors at each iteration, the cost per iteration becomes linear on N.

Stopping criteria:

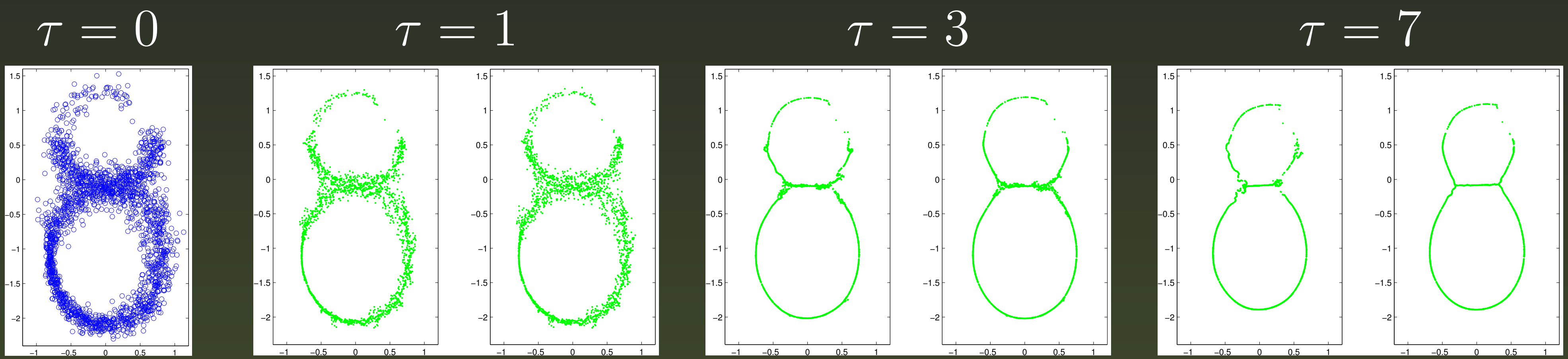
- A practical indicator is the histogram over all data points of the orthogonal variance $\lambda_\perp$ (the sum of the trailing $D-L$ eigenvalues of $\mathbf{x}_n$'s local covariance).

Experiment: Noisy spiral



Denoising a noisy spiral with outliers over iterations ($K = 10$ for local PCA).

Experiment: complex shape

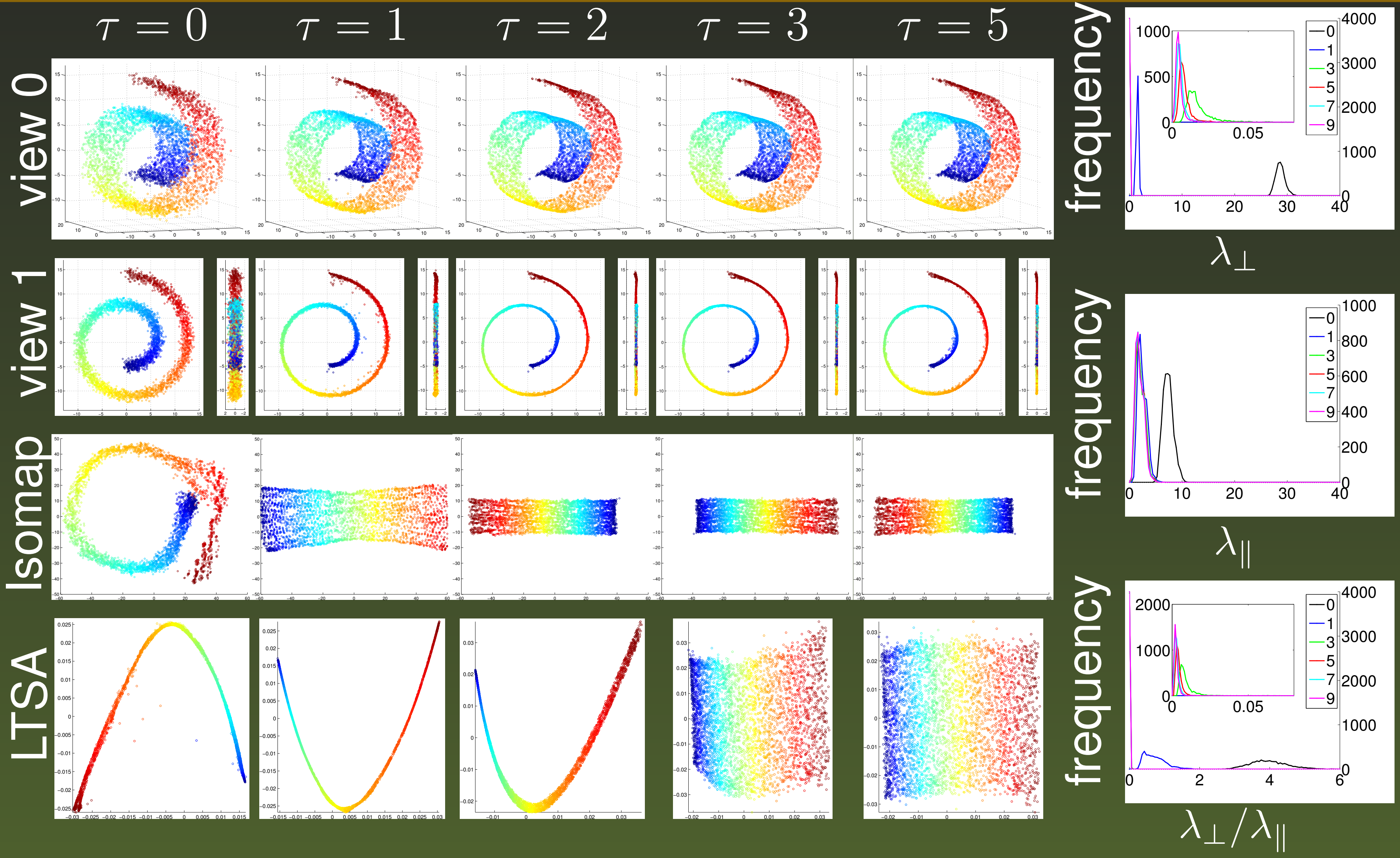


$\tau = 0$ $\tau = 1$ $\tau = 3$ $\tau = 7$

Denoising a complex shape with nonuniform density and noise with MBMSf using the usual affinity (left subplots, $\alpha = 0$) and the diffusion-map affinity normalization (right subplots, $\alpha = 1$):

$$G_\sigma^\alpha(\mathbf{x}_i, \mathbf{x}_j) = \frac{G_\sigma(\mathbf{x}_i, \mathbf{x}_j)}{(\sum_k G_\sigma(\mathbf{x}_i, \mathbf{x}_k))^\alpha (\sum_{k'} G_\sigma(\mathbf{x}_{k'}, \mathbf{x}_j))^\alpha}.$$

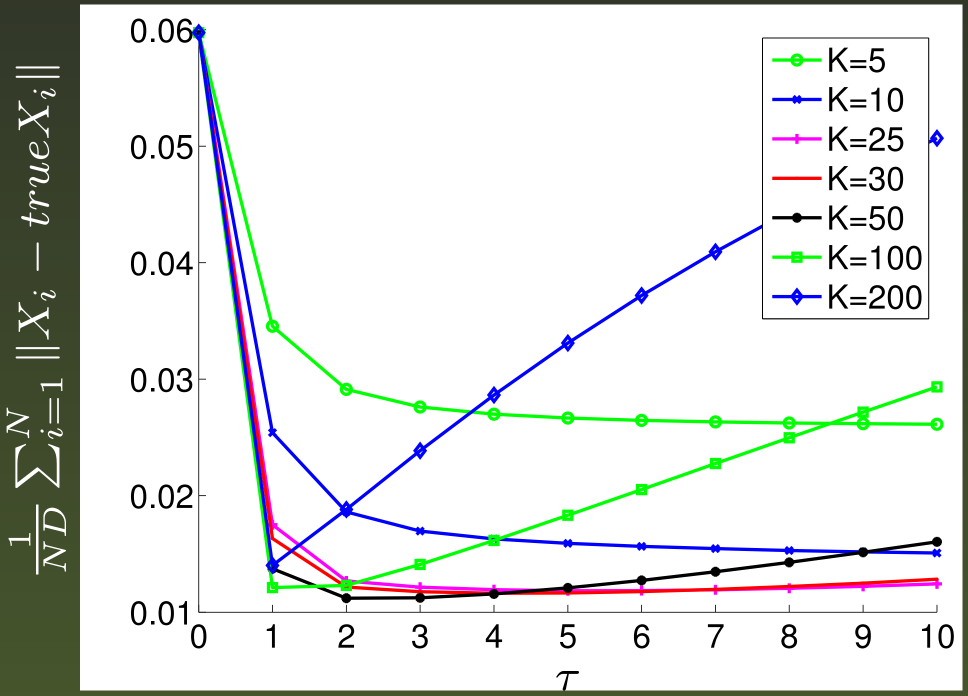
Experiment: preprocessing for spectral methods



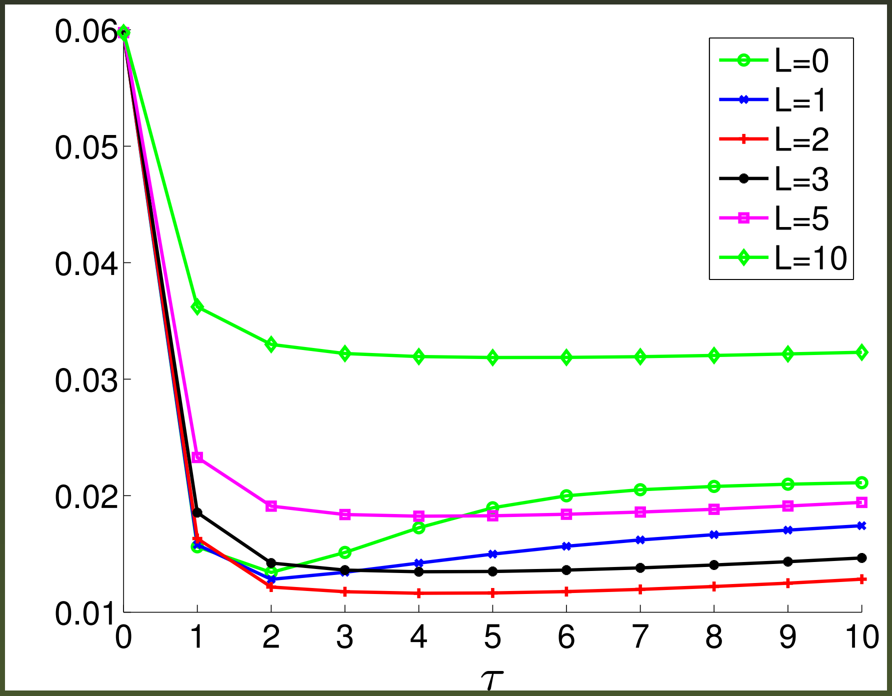
Dimensionality reduction with Isomap and LTSA for different iterations of MBMSk denoising.

Robustness to parameters choice

There are a wide range for each parameter in which MBMS works well.



$$\sigma = \infty, L = 2$$

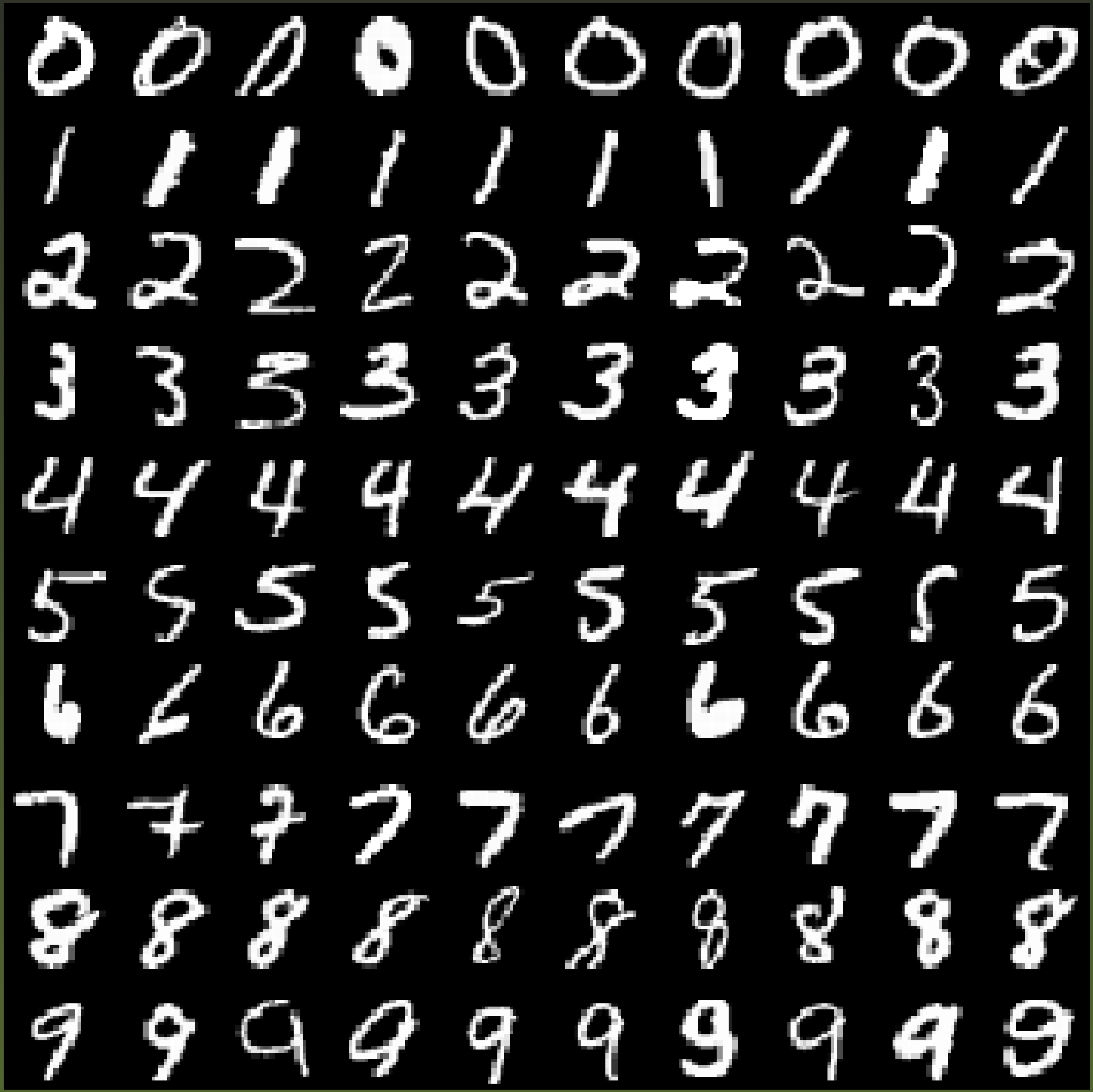


$$\sigma = \infty, K = 30$$

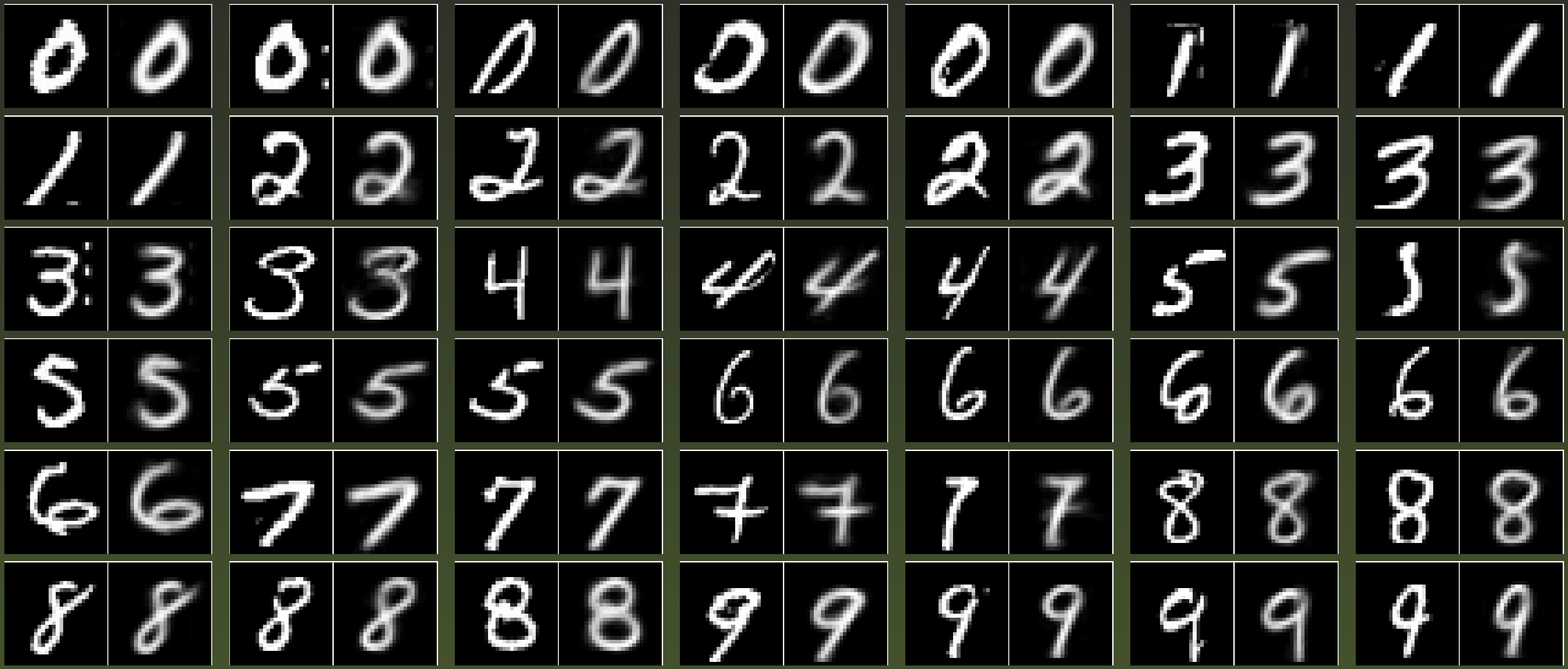
Behaviour of LTP for different parameters K and L. Error decreases for all parameter choices.

Experiment: preprocessing for classifying MNIST

Sample images from MNIST data set

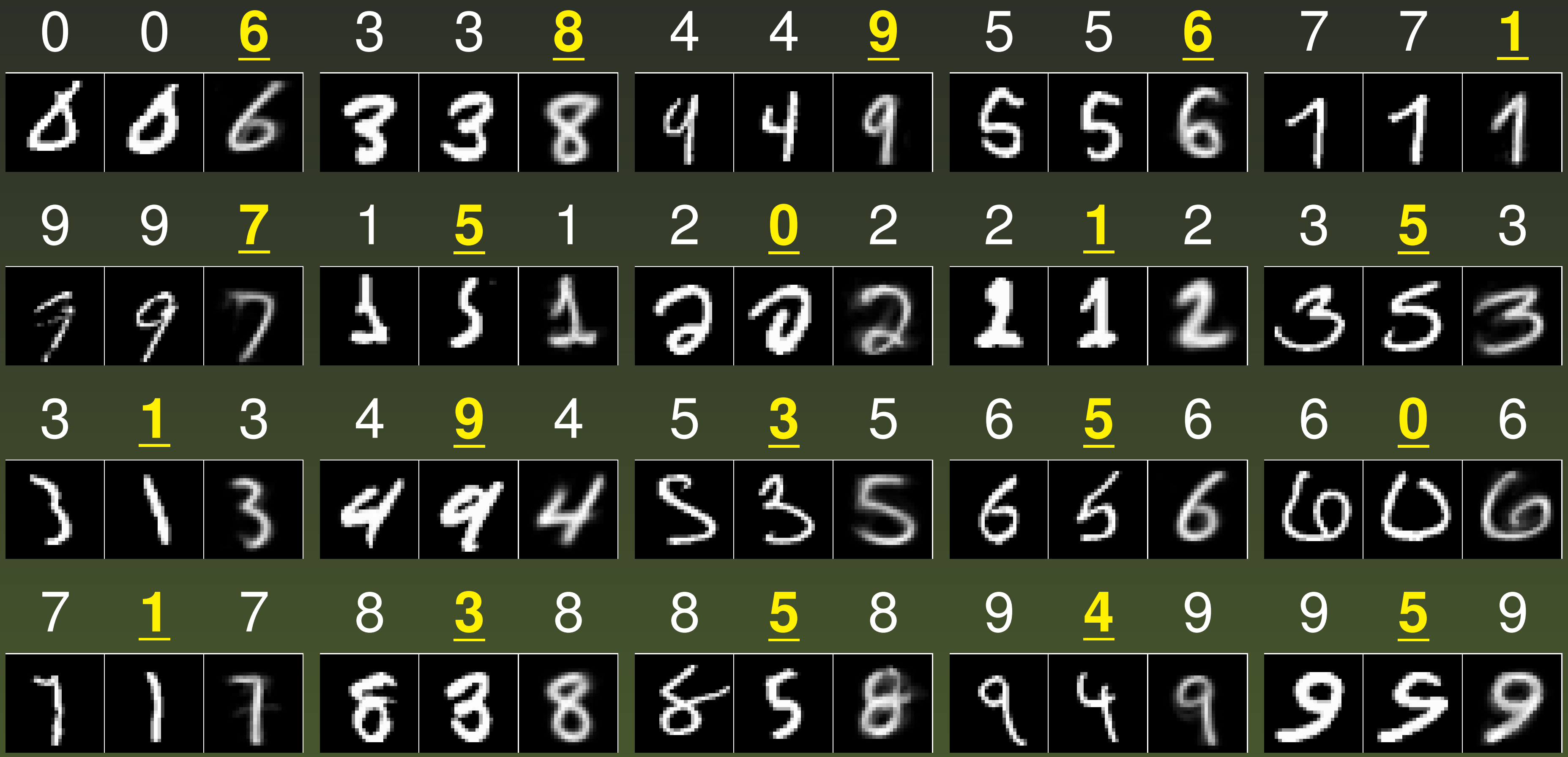


Experiment: preprocessing for classifying MNIST



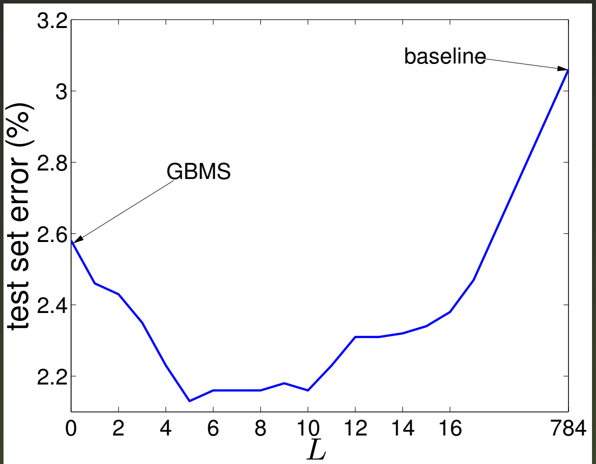
Sample pairs of (original,denoised) images from the training set.

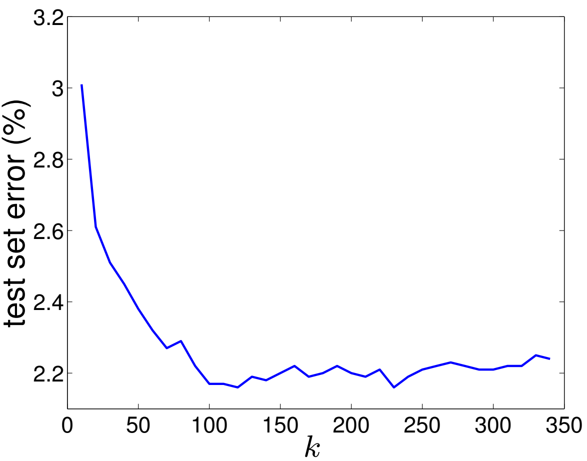
Experiment: preprocessing for classifying MNIST

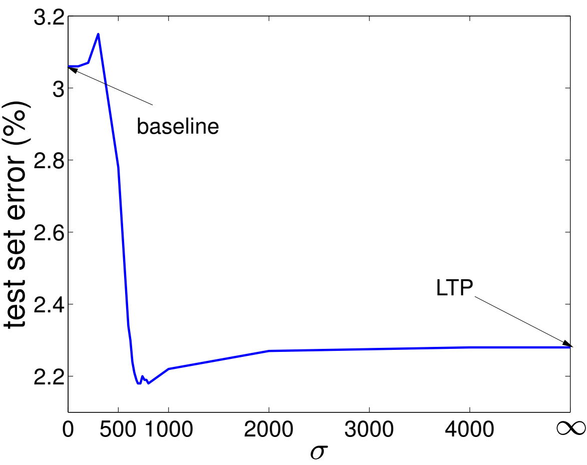


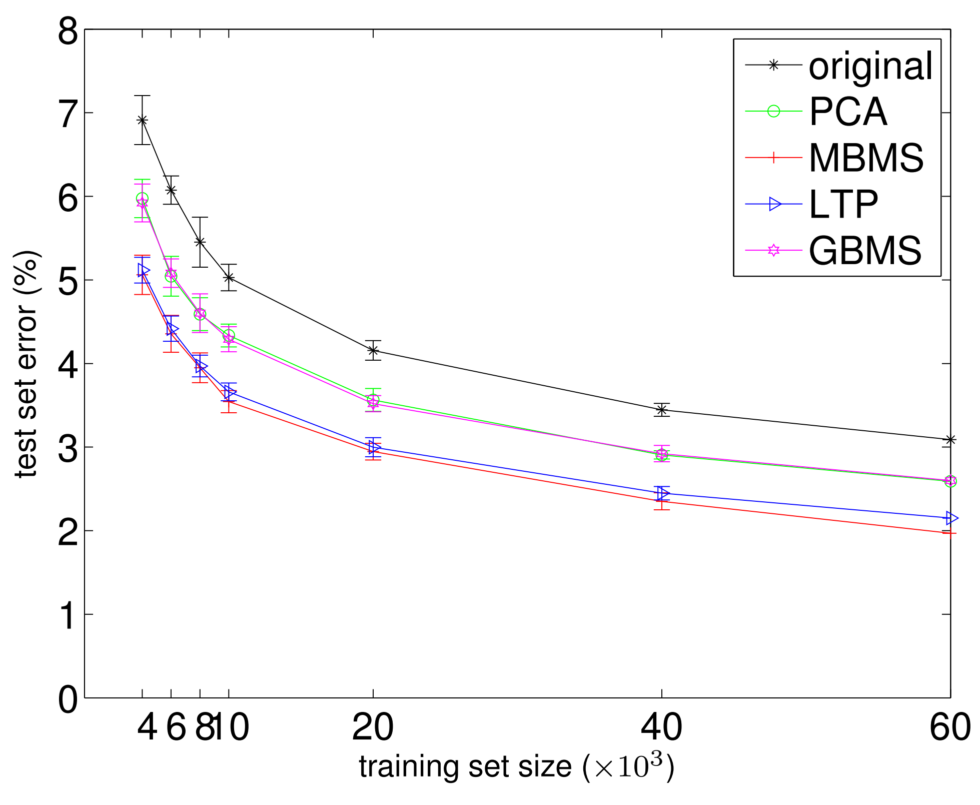
Some misclassified images. Each triplet is (test, original-nearest-neighbor, denoised-nearest-neighbor) and the corresponding label is above each image, with errors underlined.

Experiment: preprocessing for classifying MNIST









Top 3 plots: 5–fold cross-validation error (%) curves with a nearest-neighbor classifier on training set using MBMSk; $L = 9$, $k = 140$, $\sigma = 695$ is selected as final values.

Bottom left plot: denoising and classification of the MNIST test set, by training on the entire training set and also on smaller subsets of it. Algorithms (L, k, σ), MBMSk $(9, 140, 695)$, LTP $(9, 140, \infty)$, GBMS $(0, 140, 600)$, and PCA $(L = 41)$.

Theorems of MBMS

- The MBMS algorithms are covariant under rigid motions.
- If the GBMS step uses a k-nearest-neighbour graph and bandwidth $\sigma > 0$, and the local PCA step uses $k' \geq k$ neighbours and a dimension $L \geq k$, then input data set remains unchanged under MBMS iterations.
- Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$. Then the sequence of random vectors resulting from successively applying the MBMS update with a full graph, bandwidth $\sigma > 0$, $K = \infty$ and assuming a manifold of dimension L converges to $\mathcal{N}(\boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}})$ where $\tilde{\boldsymbol{\Sigma}} = \mathbf{U}\tilde{\boldsymbol{\Lambda}}\mathbf{U}^T$, and $\tilde{\lambda}_d = \lambda_d$ for $d = 1, \dots, L$ and $\tilde{\lambda}_d = 0$ for $d = L + 1, \dots, D$, with cubic order independently for each direction. That is, MBMS removes the variance along the $D - L$ minor axes.

Conclusion

- Very effective at denoising in a handful of iterations
- Able to handle extreme outliers
- Nonparametric and deterministic
- Causing very small shrinkage or distortion
- Possible applications in other area: denoising 3D point sets with 1D or 2D structure, obtained by laser scanning in Robotics and Computer Graphics and Computer Aided Design

Thank You!