Manifold Blurring Mean Shift Algorithms for Manifold Denoising

Weiran Wang and Miguel Á. Carreira-Perpiñán

Electrical Engineering and Computer Science

University of California, Merced

http://eecs.ucmerced.edu
Manifold denoising

**Input:** $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^D$

noisy dataset

with low-dimensional structure

**Output:** $\tilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_N\} \subset \mathbb{R}^D$

denoised dataset

preserving the manifold structure

Applications:

- **Computer graphics, CAD, robotics:** curve or surface smoothing given a 3D point cloud obtained by laser scanning an object.
- **Computer vision:** stereo motion segmentation, subspace estimation.
- **Machine learning:** preprocessing step to improve the robustness of dimensionality reduction or other tasks in high-dimensional spaces.
Denoising given the manifold

One approach is to learn the data manifold and then project the data on it:

- Learn mappings \( \begin{align*} f &: \text{low to high} \\
F &: \text{high to low} \end{align*} \)
- Project data onto the manifold: \( \tilde{x}_n = f(F(x_n)) \).

... But dimensionality reduction algorithms are sensitive to noise; we want to denoise before learning the manifold.

In order to make the user’s life easy and perturb the data as little as possible, the denoising algorithm should:

- be nonparametric: little model assumptions
- be deterministic: no local optima, repeatable
- require few user parameters.

Such an algorithm can be constructed using the ideas of local averaging and local low dimensionality.
The GBMS update moves each data point to the local average of its neighbours:

\[
x^{(\tau+1)}_n = \sum_{m \in \mathcal{N}_n} \frac{G_\sigma(x^{(\tau)}_m - x^{(\tau)}_n)}{\sum_{m' \in \mathcal{N}_n} G_\sigma(x^{(\tau)}_{m'} - x^{(\tau)}_n)} x^{(\tau)}_m
\]

\[
\mathcal{N}_n = k \text{ nearest neighbours of } x_n
\]

User parameters: scale \( \sigma \), number of neighbours \( k \).

The GBMS update can be seen as:

- mode seeking in the kernel density estimate \( p^{(\tau)} \) of \( X^{(\tau)} \)
- power iteration \( X^{(\tau+1)} = X^{(\tau)} P^{(\tau)} \) with the low-pass filter matrix \( P_{N \times N} \) of normalised Gaussian affinities of \( X^{(\tau)} \)

but updating \( p \) or \( P \) after each iteration, and stopping early.

Fukunaga & Hostetler 1975; Cheng 1995; Comaniciu & Meer 2002; Carreira-Perpiñán 2006.
Gaussian blurring mean shift (GBMS): denoising

Taubin 1995; Desbrun et al 1999; Park et al 2004; Hein & Maier 2007; Unnikrishnan & Hebert 2007; etc.
Gaussian blurring mean shift (GBMS): denoising
Gaussian blurring mean shift (GBMS): denoising

\[ \tau = 2 \]
Gaussian blurring mean shift (GBMS): denoising

\( \tau = 3 \)
Gaussian blurring mean shift (GBMS): denoising
Gaussian blurring mean shift (GBMS): denoising

\[ \tau = 5 \]
Gaussian blurring mean shift (GBMS): denoising
GBMS ignores the manifold structure, moving points within the manifold, particularly at boundaries or low-density regions, so the dataset shrinks nonuniformly. In-manifold motion changes properties of the data, e.g. handwriting style in MNIST.

- Smaller $\sigma$ or anisotropic kde does not solve the problem.
- Even worse in high dimensions with nonuniform, sparse data.

Idea: correct the GBMS update to prevent tangential motion.
Manifold blurring mean shift (MBMS)

**Predictor step**  Local clustering with Gaussian blurring mean shift:
move data points to the kernel average of their neighbours:

\[ x_n \leftarrow \sum_{m \in \mathcal{N}_n} \frac{G_\sigma(x_n, x_m)}{\sum_{m' \in \mathcal{N}_n} G_\sigma(x_n, x_{m'})} x_m \]

**Corrector step**  Tangent motion removal along local PCA space:
best linear \( L \)-dim. manifold in terms of reconstruction error:

\[
\min_{\mu, U} \sum_{m \in \mathcal{N}_n'} \left\| x_m - (UU^T(x_m - \mu) + \mu) \right\|^2 \\
\text{s.t.} \quad U^T U = I
\]

**Iterate**

For simplicity, we use the same neighbourhood for both the GBMS and PCA steps \( (\mathcal{N}_n = \mathcal{N}_n') \).
Manifold blurring mean shift (MBMS): denoising
Manifold blurring mean shift (MBMS): denoising

\[ \tau = 1 \]

GBMS update \( \bullet \), MBMS update \( \circ \)
Manifold blurring mean shift (MBMS): denoising

\[ \tau = 2 \]

GBMS update \( \bullet \), MBMS update \( \circ \)
Manifold blurring mean shift (MBMS): denoising

\[ \tau = 3 \]

GBMS update \( \textcolor{red}{\bullet} \), MBMS update \( \textcolor{blue}{\bullet} \)
Pseudocode: MBMS

MBMS \((L, k, \sigma)\) with full or \(k\)-nn graph: given \(X_{D \times N}\)

repeat
  for \(n = 1, \ldots, N\)
    \(\mathcal{N}_n \leftarrow \{1, \ldots, N\}\) (full graph) \hspace{1cm} MBMSf
    or \(k\) nearest neighbors of \(x_n\) \((k\)-nn graph\) \hspace{1cm} MBMSk
    \[
    \partial x_n \leftarrow -x_n + \sum_{m \in \mathcal{N}_n} \sum_{m' \in \mathcal{N}_n} \frac{G_{\sigma}(x_n, x_m)}{G_{\sigma}(x_n, x_{m'})} x_m
    \]
    mean-shift step
    \(\mathcal{X}_n \leftarrow k\) nearest neighbors of \(x_n\)
    \((\mu_n, U_n) \leftarrow \text{PCA}_L(\mathcal{X}_n)\)
    estimate \(L\)-dim tangent space at \(x_n\)
    \(\partial x_n \leftarrow (I - U_n U_n^T) \partial x_n\)
    subtract parallel motion
  end
end

\(X \leftarrow X + \partial X\)

until stop

return \(X\)

User parameters: \(L, k, \sigma\).
Particular cases of MBMS

- $\sigma = 0$ or $L = D$: no denoising (the dataset does not change)
- $\sigma = \infty$ (with full graph), $k = N$: PCA
- $L = 0$: GBMS
- $\sigma = \infty$ (with $k$–nn graph): local tangent projection (LTP).
Pseudocode: LTP

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

LTP $(L, k)$ with $k$-nn graph: given $X_{D \times N}$

repeat
  for $n = 1, \ldots, N$
    $X_n \leftarrow k$ nearest neighbors of $x_n$
    $(\mu_n, U_n) \leftarrow \text{PCA}_L(X_n)$
    $\partial x_n \leftarrow (I - U_n U_n^T)(\mu_n - x_n)$  \hspace{1cm} \text{estimate $L$-dim tangent space at $x_n$}
    $x_n \leftarrow x_n + \partial x_n$  \hspace{1cm} \text{project point onto tangent space}
  end

$X \leftarrow X + \partial X$

until stop

return $X$

User parameters: $L, k$.

Simpler to use, and achieves near-optimal results.
Complexity, convergence and stopping criterion

Computational cost per iteration:

- Full graph: $O(N^2 D + N(D + k) \min(D, k)^2)$.
  - neighbours + GBMS
  - local PCAs
- Given $k$-nn graph and not updating the neighbours at each iteration: cost linear on $N$. Other speedups possible.

Convergence:

- GBMS converges to a dataset where all points coincide.
- In general, we don’t know whether MBMS converges. It does in particular cases, e.g. it flattens a Gaussian dataset onto its principal component space with cubic convergence.
- In both GBMS and MBMS we stop early anyway, since the desired results appear after 1–5 iterations.

Stopping criterion: 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 $x_n$’s local covariance).
Experiment: noisy spiral with outliers

- Mainstream points are unaffected by the presence of outliers.
- Likewise, MBMS denoises connected components of a manifold separately.
Preprocessing for spectral methods: 100D swissroll

\( \tau = 0 \), \( \tau = 1 \), \( \tau = 2 \), \( \tau = 3 \), \( \tau = 5 \)

view 0

view 1

Isomap

LTSA

MBMSk

LTP

GBMSk

\( \lambda_{\perp} \)

\( \lambda_{\parallel} \)

\( \lambda_{\perp}/\lambda_{\parallel} \)
100D swissroll: robustness to parameter choice

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

$L = \cdot, k = 30, \sigma = \infty$

LTP over iterations $\tau = 0, 1, 2, \ldots$ for different values of $k$ and $L$. The ground-truth error $\frac{1}{ND} \sum_{n=1}^{N} \| x_n - x_{n}^{\text{true}} \|$ decreases for a wide range of parameter values.
Other datasets

- High-dimensional swissroll with a hole.
- High-dimensional swissroll with outliers.
- Shape with varying density, varying noise, self-intersection, gaps (left: Gaussian affinities; right: diffusion-map affinities):

<table>
<thead>
<tr>
<th>( \tau = 0 )</th>
<th>( \tau = 1 )</th>
<th>( \tau = 3 )</th>
<th>( \tau = 7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Preprocessing for MNIST digit classification

- Greyscale images of $28 \times 28$ pixels with varying thickness, slant, strokes, writing style, considered as vectors of $D = 729$.
- $N = 60\,000$ images for training; we denoise them with 1 iteration and feed to a nearest-neighbour classifier.
- No other preprocessing of any kind.
- $10\,000$ noisy images for testing the classifier.
Sample pairs of (original, denoised) images from the training set.

Note “intelligent corrections”: filling in gaps, erasing unusual strokes, removing speckle noise, etc., while respecting the digit’s slant, thickness and style.
5–fold cross-validation error (%) on the training set with MBMSk: optimum around \((L, k, \sigma) = (9, 140, 695)\). Decreases the baseline error by 36\% (3.06\% → 1.97\%).

MBMSk \((9, 140, 695)\), LTP \((9, 140, \infty)\), GBMSk \((0, 140, 600)\), PCA \((L = 41)\).
Preprocessing for MNIST: confusion matrix

Without denoising

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Each row: % probability of recognizing a ground-truth digit.

With denoising

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Each row: % probability of recognizing a ground-truth digit.

Improvement (without – with)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Each row: % probability of recognizing a ground-truth digit.

p. 28
Preprocessing for MNIST digit classification

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>6</th>
<th>3</th>
<th>3</th>
<th>8</th>
<th>4</th>
<th>4</th>
<th>9</th>
<th>5</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>7</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>

Some misclassified images:

- Triplet (test, original-nearest-neighbour, denoised-nearest-neighbour).
- Corresponding label above each image, with errors underlined.
Conclusions

- **MBMS**: generic preprocessing algorithm for data with low intrinsic dimensionality, prior to (un)supervised learning.
- **Local Tangent Projection (LTP)**: simple, effective special case.
- Denoising by local averaging (predictor step), manifold structure preserved along local tangent space (corrector step).
- Nonparametric, deterministic, few user parameters.
- Effective denoising in a few iterations with little distortion of the manifold, robust to outliers.

- Extensions:
  - Predictor step: use step size; use implicit GBMS.
  - Corrector step: estimate local dimensionality from local covariance (useful with multiple manifolds).
  - Speeding up both steps.

Matlab code: [http://eecs.ucmerced.edu](http://eecs.ucmerced.edu)

Work supported by NSF CAREER award IIS–0754089.