# LOCALLY LINEAR LANDMARKS FOR LARGE-SCALE MANIFOLD LEARNING

**Max Vladymyrov** and **Miguel Á. Carreira-Perpiñán.** EECS, UC Merced, USA

## 1 Abstract

Spectral methods for manifold learning and clustering typically construct a graph weighted with affinities (e.g. Gaussian or shortest-path distances) from a dataset and compute eigenvectors of a graph Laplacian. With large datasets, the eigendecomposition is too expensive, and is usually approximated by solving for a smaller graph defined on a subset of the points (landmarks) and then applying the Nyström formula to estimate the eigenvectors over all points. This has the problem that the affinities between landmarks do not benefit from the remaining points and may poorly represent the data if using few landmarks. We introduce a modified spectral problem that uses all data points by constraining the latent projection of each point to be a local linear function of the landmarks' latent projections. This constructs a new affinity matrix between landmarks that preserves manifold structure even with few landmarks and allows one to reduce the eigenproblem size and works specially well when the desired number of eigenvectors is not trivially small. The solution also provides a nonlinear out-of-sample projection mapping that is faster and more accurate than the Nyström formula.

## 2 Spectral methods

Given the input data points $\mathbf{Y} \in \mathbb{R}^{D\times N}$, the generalized spectral problem seeks a solution $\mathbf{X} \in \mathbb{R}^{d\times N}$ to a following optimization problem:

$$\min_{\mathbf{X}} \operatorname{tr}\left(\mathbf{X}\mathbf{A}\mathbf{X}^T\right), \text{s.t. } \mathbf{X}\mathbf{B}\mathbf{X}^T = \mathbf{I}. \qquad (1)$$

- $\mathbf{A}$ - symmetric positive semidefinite matrix, usually represents the similarity between data points,
- $\mathbf{B}$ - symmetric positive definite matrix, typically represents the scale of the points with respect to each other.

The solution is given by $\mathbf{X} = \mathbf{U}_d^T\mathbf{B}^{-\frac{1}{2}}$, where $\mathbf{U}_d = (\mathbf{u}_1, \dots, \mathbf{u}_d)$ are $d$ trailing eigenvectors of a $N \times N$ matrix $\mathbf{C} = \mathbf{B}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}^{-\frac{1}{2}}$. It is too costly to find the solution when $N$ and $d$ are large.

## 3 Locally Linear Landmarks (LLL)

Define:

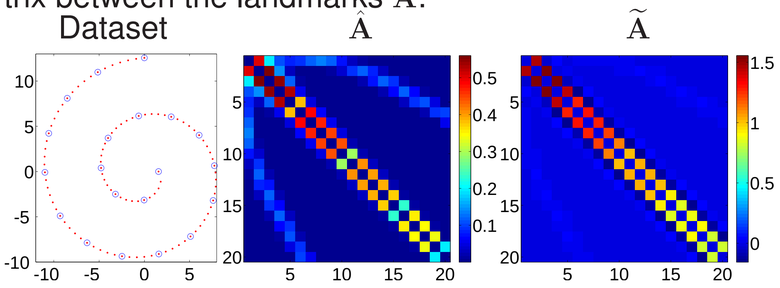
- $\widetilde{\mathbf{Y}} = (\widetilde{\mathbf{y}}_1, \dots, \widetilde{\mathbf{y}}_L) \in \mathbb{R}^{D\times L}$ a set of L landmarks chosen from the data set $\mathbf{Y}$.
- $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_N) \in \mathbb{R}^{L\times N}$ local projection matrix, which corresponds to the proximity of the points in the dataset to nearby landmarks.

Now, we can express each point as a linear combination of $K_Z$ nearby landmarks: $\mathbf{y}_n = \sum_{k=1}^{K_Z} \widetilde{\mathbf{y}}_k z_{nk}$. We assume that the transformation between landmarks and the rest of the points is preserved in both high- and low-dimensional spaces, i.e. $\mathbf{X} = \widetilde{\mathbf{X}}\mathbf{Z}$. Substituting this into the spectral problem (1) gives reduced spectral problem:

$$\min_{\widetilde{\mathbf{X}}} \operatorname{tr}\left(\widetilde{\mathbf{X}}\widetilde{\mathbf{A}}\widetilde{\mathbf{X}}^T\right), \text{s.t. } \widetilde{\mathbf{X}}\widetilde{\mathbf{B}}\widetilde{\mathbf{X}}^T = \mathbf{I}, \qquad (2)$$

with $\widetilde{\mathbf{A}} = \mathbf{Z}\mathbf{A}\mathbf{Z}^T$, $\widetilde{\mathbf{B}} = \mathbf{Z}\mathbf{B}\mathbf{Z}^T$. The solution is given by $\widetilde{\mathbf{X}} = \widetilde{\mathbf{U}}_d^T\widetilde{\mathbf{B}}^{-\frac{1}{2}}$, where $\widetilde{\mathbf{U}}_d$ are $d$ trailing eigenvectors of the matrix $\widetilde{\mathbf{C}} = \widetilde{\mathbf{B}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}^{-\frac{1}{2}}$.

1. After $\widetilde{\mathbf{X}}$ is found, the values of $\mathbf{X}$ can be recovered using $\mathbf{X} = \widetilde{\mathbf{X}}\mathbf{Z}$.
2. Dramatic cost reduction: the total cost is $\mathcal{O}\big(N(K_Z c + Ld + DK_Z^2) + L^3)\big)$ where $c$ is a constant that depends on the sparsity of $\mathbf{A}$ and $\mathbf{B}$.
3. New similarity matrix $\widetilde{\mathbf{A}}$ takes into account the whole dataset and can dramatically improve the quality of similarity matrix between the landmarks $\hat{\mathbf{A}}$:




4. Matrix $\mathbf{Z}$ can be used as a cheep out-of-sample extension with cost $\mathcal{O}(DK_Z^2 + Ld)$ per point:
   - for a new point $\mathbf{y}_0$ find a projection vector $\mathbf{z}_0$ using $K_Z$ landmarks around $\mathbf{y}_0$.
   - the embedding $\mathbf{x}_0$ is found using landmark projection of the training set: $\mathbf{x}_0 = \widetilde{\mathbf{X}}\mathbf{z}_0$

## 3 Locally Linear Landmarks for Laplacian Eigenmaps

We can apply LLL to Laplacian Eigenmaps (LE) algorithm (Belkin & Niyogi, 2003). In this case:

- $\mathbf{A}$ - graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ for a symmetric affinity matrix $\mathbf{W}$ with degree matrix $\mathbf{D} = \operatorname{diag}\left(\sum_{m=1}^{N} w_{nm}\right)$.
- $\mathbf{B}$ - degree matrix $\mathbf{D}$.

$$\min_{\mathbf{X}} \operatorname{tr}\left(\mathbf{X}\mathbf{L}\mathbf{X}^T\right), \text{s.t. } \mathbf{X}\mathbf{D}\mathbf{X}^T = \mathbf{I},\ \mathbf{X}\mathbf{D}\mathbf{1} = \mathbf{0}.$$

Using (2) the coefficients of the model becomes:

$$\widetilde{\mathbf{A}} = \mathbf{Z}\mathbf{L}\mathbf{Z}^T, \qquad \widetilde{\mathbf{B}} = \mathbf{Z}\mathbf{D}\mathbf{Z}^T.$$

## 4 Properties of LLL

1. **Projection matrix $\mathbf{Z}$.** We need to keep $K_Z$ landmarks closest to $\mathbf{y}_n$. Solve the optimization problem:

$$\min_{\mathbf{Z}} \|\mathbf{Y} - \widetilde{\mathbf{Y}}\mathbf{Z}\|^2, \text{ s.t. } \mathbf{1}^T\mathbf{Z} = \mathbf{1}^T.$$

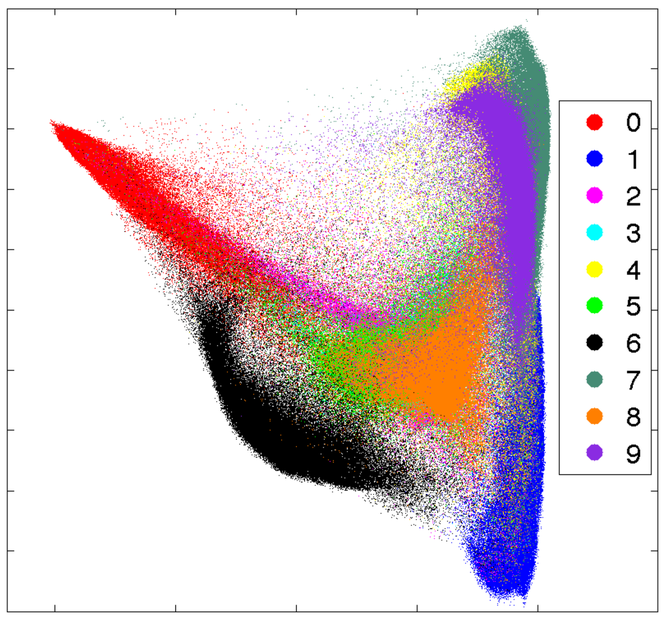
   For the solution (a) compute a local Gram matrix $\mathbf{G}_{ij} = (\mathbf{y}_n - \widetilde{\mathbf{y}}_i)(\mathbf{y}_n - \widetilde{\mathbf{y}}_j)$, (b) solve a linear system $\sum_{k=1}^{L} \mathbf{G}_{jk} z_{nk} = 1$ and (c) rescale the weights so they sum to one.
2. **Location of landmarks.** The landmarks should be spread as uniformly as possible along the manifold to provide local reconstruction. It can be done using:
   - centroids of clustering algorithm (e.g. k-means);
   - greedy algorithm (e.g. MinMax algorithm; de Silva & Tenenbaum, 2004);
   - random subset of the data.
3. **Total number of landmarks L.** The more landmarks we can afford, the better is the final result. $L \ll N$ (approx.) $\xrightarrow[\text{slower}]{\text{better approximation}}$ $L = N$ (original),
4. **Number of landmarks $K_Z$ for the projection matrix $\mathbf{Z}$.** Each point should be a locally linear reconstruction of the nearby landmarks:
   - Few landmarks $\Rightarrow$ inexact reconstruction.
   - Too many landmarks $\Rightarrow$ lose locality.

   Practically, choosing $K_Z \approx d$ works well.

## 5 Experimental Evaluation

We compare LLL for LE to three baselines:

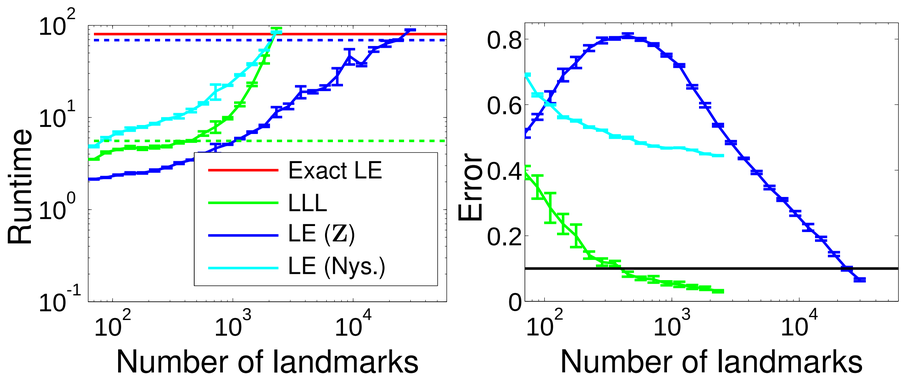
1. Exact LE runs LE on the full dataset. Best embedding, but the runtime is large.
2. Landmark LE runs LE only on a set of landmark points. Once their projection is found, the rest of the points are embedded using:
   - LE(Z): $\mathbf{Z}$ as an out-of-sample.
   - LE(Nys.): Nyström method as an out-of-sample.



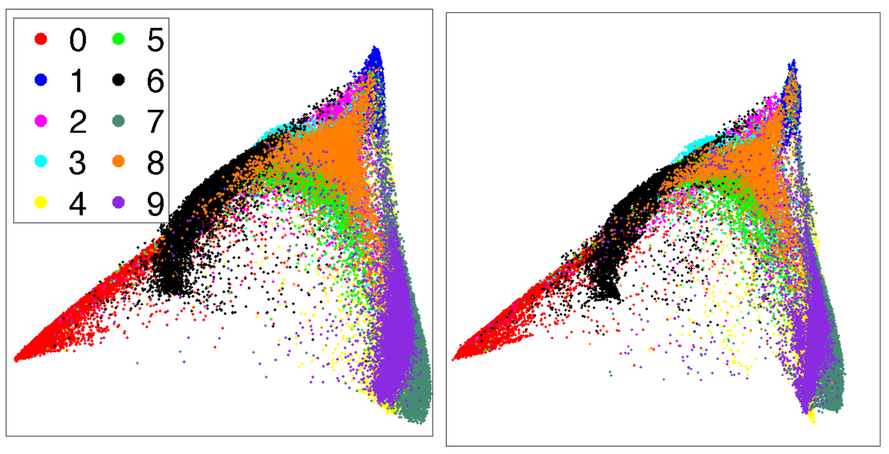

## 1. Number of landmarks

The role of number of landmarks on the performance of LLL:

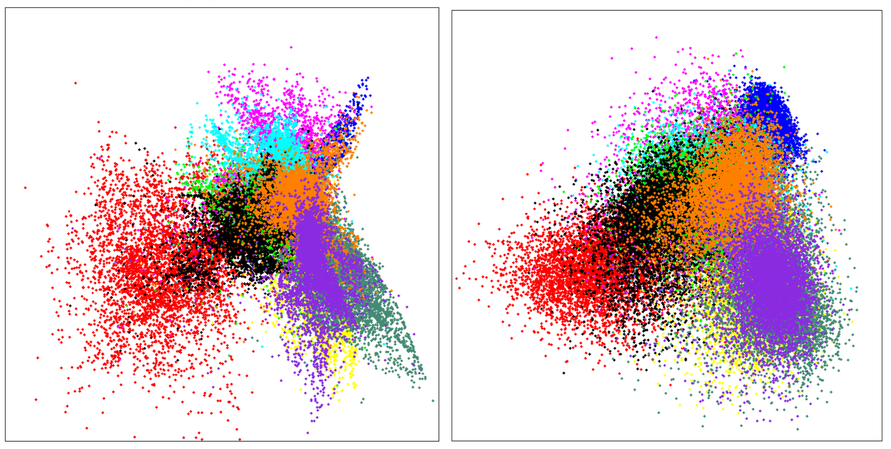
- Use $60\,000$ MNIST digits.
- Reduce the dimensionality to $d = 50$.
- Set $K_Z = 50$, chose landmarks randomly and increase their number logarithmically from $L = 50$ to $L = 60\,000$.
- Compute the error between the embeddings with Exact LE.




Exact LE, 80 s. LLL, 5.5 s.



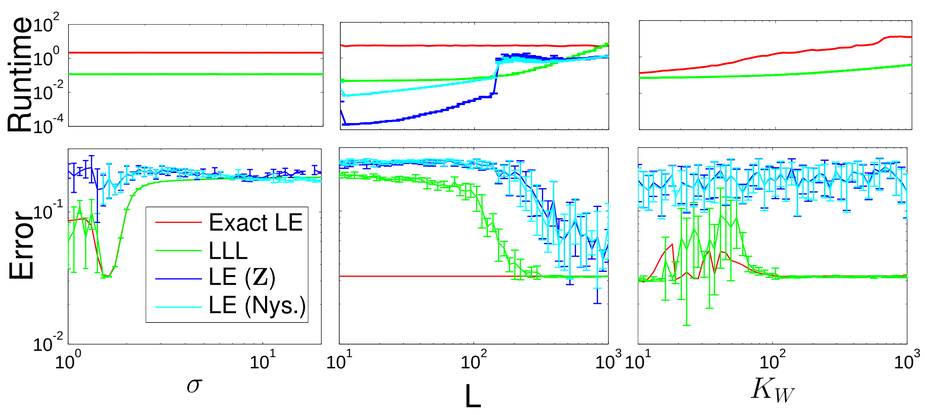

LE (Z), 5.5 s. LE (Nys.), 5.5 s.



## 2. Model selection

Use LLL to predict the parameters of the affinity matrix:

- Use $4\,000$ points from swiss roll dataset.
- Vary parameters of the algorithm (bandwidth $\sigma$, number of landmarks $L$ and sparsity $K_W$ of the affinity matrix $A$) and compute the relative error of the embedding with respect to the ground truth.




## 3. Large-scale: $10^6$ points from infinite MNIST

- Generate $1\,020\,000$ handwritten digits using elastic transformation to the MNIST digits (see Loosli et al., 2007).
- Use $K_Z = 5$ and $L = 10\,000$ randomly selected landmarks.
- It took $4.2$ minutes to compute $\mathbf{Z}$ and $14$ minutes to compute the embedding.

## 6 Conclusions

The bottleneck of spectral methods is expensive eigenvalue decomposition. We propose to optimize only for a small set of landmark points, while retaining the structure of the whole data. The algorithm can be used (1) to find a fast approximate embedding of large dataset, (2) as a model parameters selection method, (3) as an out-of-sample extension to spectral methods. For the Laplacian Eigenmaps the algorithm is able to achieve $10 - 20\times$ speed-up with small approximation error.

ICML | Atlanta
International Conference on Machine Learning

Partially supported by NSF CAREER award IIS–075408