Locally Linear Landmarks for Large-Scale Manifold Learning

Max Vladymyrov MVLADYMYROV@UCMERCED.EDU
Miguel Á. Carreira-Perpiñán MCARREIRA-PERPINAN@UCMERCED.EDU
Electrical Engineering and Computer Science, School of Engineering, University of California, Merced

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.

1. Introduction

Dimensionality reduction algorithms have long been used either for exploratory analysis of a high-dimensional dataset, to reveal structure such as clustering, or as a preprocessing step to extract some low-dimensional features that are useful for classification or other tasks. Here we concentrate on a well known set of spectral dimensionality reduction algorithms (Saul et al., 2006). The input to these algorithms consists of a symmetric positive semidefinite matrix $\mathbf{A}$ that usually represents the similarity between data points $\mathbf{Y} \in \mathbb{R}^{D\times N}$ and a symmetric positive definite matrix $\mathbf{B}$ that typically represents the scale of the points with respect to each other. Given these two matrices, the generalized spectral problem seeks a solution $\mathbf{X} \in \mathbb{R}^{d\times N}$ to a following optimization problem:

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

Within this framework it is possible to represent such methods as Laplacian Eigenmaps (Belkin & Niyogi, 2003), Kernel PCA (Schölkopf et al., 1998), MDS (Cox & Cox, 1994), ISOMAP (Tenenbaum et al., 2000), MVU (Weinberger & Saul, 2004), LLE (Saul & Roweis, 2003), and spectral clustering (Ng et al., 2002).

The solution of the problem above 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 matrix $\mathbf{C} = \mathbf{B}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}^{-\frac{1}{2}}$. Notice that the solution requires computation of only d trailing eigenvectors. When d is small (for example, for visualization purposes) and the matrix is sparse, one can use techniques such as restarted Arnoldi Iterations (Lehoucq & Sorensen, 1996) to find only a subset of eigenvectors. However, this operation is costly when N and d are large. In this paper, we propose to alleviate this problem by introducing a new method called Locally Linear Landmarks (LLL). There we first select a subset of L landmarks from the data and then find a projection matrix $\mathbf{Z} \in \mathbb{R}^{L\times N}$ that allows linear and local data representation using

$$\mathbf{Y} \approx \widetilde{\mathbf{Y}}\mathbf{Z}. \tag{2}$$

Thus, we can re-express the problem in the same framework (1) as before now using smaller number of variables L, reducing the cost of eigendecomposition dramatically.

The proposed algorithm can be used for multiple purposes. First of all, naturally, it can be used to approximate the results of the generalized spectral problem when N and d are large. Second, the similarity matrices $\mathbf{A}$ and $\mathbf{B}$ are usually constructed using some data-dependent meta-parameters, such as a bandwidth σ and a sparsity level K_W. These parameters depend on the data, rather then on the algorithm, and have to be tuned each time anew for new data. For example, Ng et al. (2002) suggest to choose the bandwidth σ as the one that gives the least distorted clusters

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

in their spectral clustering algorithm. Because LLL uses same similarity matrices as the original algorithm but runs much faster, we can use it to tune the similarity parameters. Third, the asymptotic complexity of LLL depends indirectly on the dimensionally of the solution d, so LLL has much bigger gains when d is large. Thus, the algorithm can be adopted as a preprocessing step for other machine learning tasks (e.g. for classification). Finally, notice that (1) does not define any mapping and there is no explicit way to find a solution for a new point. Bengio et al. (2004a) suggest an out-of-sample extension based on the Nyström approximation and Carreira-Perpiñán & Lu (2007) propose to use KDE on resulting embedding to produce a mapping in low-dimensional space. Comparing to these approaches LLL allows a natural out-of-sample extension that is based on same approximation as the original problem. Because of space limitations, in this paper we will demonstrate results addressing only the first purpose, i.e. approximating the large dataset.

A typical method to find the approximate fast solution of the spectral problem is the Nyström method (Williams & Seeger, 2001; Bengio et al., 2004; Drineas & Mahoney, 2005; Talwalkar et al., 2008) which approximates the eigendecomposition of a large positive semidefinite matrix using the eigendecomposition of a much smaller matrix of landmarks. This can be seen as an out-of-sample extension where first we find the location of landmarks and project the rest of the points afterwards (Bengio et al., 2004). However, during the projection of landmarks, the method does not use the data from non-landmark points readily available from the beginning. This can result in bad quality of the embedding, especially when the number of landmarks is low.

The idea of representing a point by linear coding (2) was used in many different domains of machine learning, such as image classification (Gao et al., 2010; Wang et al., 2010), manifold learning (Roweis & Saul, 2000; Weinberger et al., 2005; Yu et al., 2009), supervised (Ladický & Torr, 2011) and semi-superwised (Liu et al., 2010) learning. In addition to linearity, many of the above algorithms are trying to create local and sparse representation of the data, meaning that the points should be reconstructed using only nearby landmarks.

2. Locally Linear Landmarks

Let us define the landmarks as a set $\widetilde{\mathbf{Y}} = (\widetilde{\mathbf{y}}_1, \dots, \widetilde{\mathbf{y}}_L) \in \mathbb{R}^{D\times L}$ of L points in the same space as the high-dimensional input $\mathbf{Y}$. Now each datapoint $\mathbf{y}_n$ can be expressed as a linear combination of nearby landmark points: $\mathbf{y}_n = \sum_{k=1}^{L} \widetilde{\mathbf{y}}_k z_{nk}$ where $\mathbf{z}_n$ is a local projection vector for the point $\mathbf{y}_n$. There are multiple ways to make this projection local. One can consider choosing K_Z landmarks closest to $\mathbf{y}_n$ or ϵ-balls centered around $\mathbf{y}_n$. Moreover, the

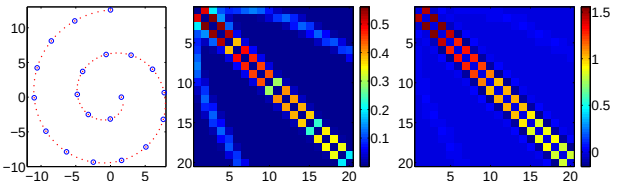

Figure 1. Affinity matrices for landmarks from the spiral dataset. *Left:* 100 points along the spiral (in red) with 20 landmarks selected uniformly (in blue). *Center:* original affinity matrix $\mathbf{W}$ used by LE build using just landmarks (σ=2). *Right:* learned affinity matrix $\mathbf{C}$ of LLL using the whole dataset.

choice of landmarks can be different for every n. In our experiments, we keep only K_Z landmarks that are closest to $\mathbf{y}_n$ and use the same K_Z for all the points. Therefore, the projection matrix $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_N) \in \mathbb{R}^{L\times N}$ has only K_Z nonzero elements for every column. This matrix intuitively corresponds to the proximity of the points in the dataset to the nearby landmarks and it should be invariant to rotation, rescaling and translation. The invariance to rotation and rescaling is given by the linearity of the reconstructing matrix $\widetilde{\mathbf{Y}}\mathbf{Z}$ with respect to $\widetilde{\mathbf{Y}}$, whereas translation invariance must be enforced by constraining columns of $\mathbf{Z}$ to sum to one. This leads to the following optimization problem:

$$\min_{\mathbf{Z}} \|\mathbf{Y} - \widetilde{\mathbf{Y}}\mathbf{Z}\|^2, \text{ s.t. } \mathbf{1}^T\mathbf{Z} = \mathbf{1}^T. \tag{3}$$

Following the approach proposed in (Saul & Roweis, 2003) we introduce point-wise Gram matrix $\mathbf{G} \in \mathbb{R}^{L\times L}$ with elements

$$\mathbf{G}_{ij} = (\mathbf{y}_n - \widetilde{\mathbf{y}}_i)(\mathbf{y}_n - \widetilde{\mathbf{y}}_j) \tag{4}$$

for every $n = 1, \dots, N$. Now, the solution to the problem (3) is found by solving a linear system $\sum_{k=1}^{L} \mathbf{G}_{jk} z_{nk} = \mathbf{1}$ and then rescaling the weights so they sum to one.

We assume that the linearity transformation $\mathbf{Z}$ between landmarks and the rest of the points is preserved in high-dimensional and low-dimensional spaces, i.e.

$$\mathbf{X} = \widetilde{\mathbf{X}}\mathbf{Z}. \tag{5}$$

Substituting this into the spectral problem (1) gives the following reduced problem

$$\min_{\widetilde{\mathbf{X}}} \operatorname{tr}\left(\widetilde{\mathbf{X}}\widetilde{\mathbf{A}}\widetilde{\mathbf{X}}^T\right), \text{s.t. } \widetilde{\mathbf{X}}\widetilde{\mathbf{B}}\widetilde{\mathbf{X}}^T = \mathbf{I}, \tag{6}$$

with

$$\widetilde{\mathbf{A}} = \mathbf{Z}\mathbf{A}\mathbf{Z}^T, \qquad \widetilde{\mathbf{B}} = \mathbf{Z}\mathbf{B}\mathbf{Z}^T. \tag{7}$$

The solution is now 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 $\widetilde{\mathbf{C}} = \widetilde{\mathbf{B}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}^{-\frac{1}{2}}$. After the solution for the landmarks is found, the values of $\mathbf{X}$ can be recovered by applying the formula (5) once again.

We can see the reduced problem (6) as a spectral problem for just landmark points using a similarity matrix $\widetilde{\mathbf{A}}$ that incorporates information from the whole dataset. For example, in Laplacian Eigenmaps spectral problem (see Section 3 below) $\mathbf{A}$ is given by an affinity matrix $\mathbf{W}$ and using LLL method we can dramatically improve the quality of $\mathbf{W}$ by including additional information from the whole dataset. In fig. 1 we show the affinity matrix for 20 landmark points along the spiral constructed in a usual way and using LLL method by including information about the whole dataset into augmented affinities. Notice that LLL affinity (right plot) is more banded and the weights are more uniformly distributed compared to LE affinity (central plot).

For the location of the landmarks, they should be spread as uniformly as possible around the data to provide local reconstruction. For example, we can do this using k-means algorithm by associating landmarks with the points that are closest to the centroids. We can also use heuristics, such as MinMax algorithm used by de Silva & Tenenbaum (2004), that greedily select landmarks one by one such that each new one maximizes the distance between already existing set. However, we cannot spend too many resources on landmarks selection in order to introduce as little computational overhead as possible. Thus, for our experiments, we are just going to select the landmarks randomly from the dataset, requiring almost no extra computations.

The complexity of LLL is $\mathcal{O}\big(N(K_Z c + Ld + DK_Z^2) + L^3\big)$ where c is some constant that depends on the sparsity of matrices $\mathbf{A}$ and $\mathbf{B}$. Thus the cost is linear in the number of points N, which is asymptotically much faster than the cubic cost of eigendecomposition.

This framework also allows for a cheap and natural out-of-sample extension. In particular, given a new point $\mathbf{y}_0 \in \mathbb{R}^D$ that is not a part of the original dataset, we can find its projection on the low-dimensional space by computing a new projection vector $\mathbf{z}_0$ for that point using K_Z landmarks around $\mathbf{y}_0$. The embedding of $\mathbf{y}_0$ is found from a linear combination of the landmark projections $\mathbf{x}_0 = \widetilde{\mathbf{X}}\mathbf{z}_0$. The cost of the out-of-sample is $\mathcal{O}(DK_Z^2 + Ld)$ which is linear for all the parameters except for K_Z, which is usually low.

Choice of Parameters. There are two main parameters of the algorithm: the number of landmarks L and the number of landmarks K_Z reconstructing every point. For L, basically, the more landmarks we can afford, the better the final result. With the increase in the number of landmarks, the results look more and more similar to the results of the original spectral algorithm and with $L = N$, the exact method is recovered.

For the number of landmarks around points, each point should be a local linear reconstruction of the nearby landmarks. Thus, it is important that there are enough landmarks so that for each point its nearest landmarks are chosen along the manifold. On the other hand, with a lot of landmarks the reconstruction may lose locality. Overall, we found that the number of landmarks around each point should be slightly bigger than d.

3. Locally Linear Landmarks for Laplacian Eigenmaps

A particular case of the spectral method for which we can apply LLL is Laplacian Eigenmaps (LE) algorithm (Belkin & Niyogi, 2003). The general embedding formulation is recovered using $\mathbf{A}$ as a graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ defined on a symmetric affinity matrix $\mathbf{W}$ with degree matrix $\mathbf{D} = \operatorname{diag}\left(\sum_{m=1}^{N} w_{nm}\right)$ and using $\mathbf{B}$ as that degree matrix $\mathbf{D}$. The objective function is thus

$$\min_{\mathbf{X}} \operatorname{tr}\left(\mathbf{X}\mathbf{L}\mathbf{X}^T\right), \text{s.t. } \mathbf{X}\mathbf{D}\mathbf{X}^T = \mathbf{I}, \mathbf{X}\mathbf{D}\mathbf{1} = \mathbf{0}. \tag{8}$$

Note that adding additional second constraint does not alter the general formulation of the spectral solution, but just removes the first (trailing) eigenvector, which is constant and equal to $\mathbf{D}^{-\frac{1}{2}}\mathbf{1}$ with eigenvalue 1.

Using (6) the coefficients of the model becomes:

$$\widetilde{\mathbf{A}} = \mathbf{Z}\mathbf{L}\mathbf{Z}^T, \qquad \widetilde{\mathbf{B}} = \mathbf{Z}\mathbf{D}\mathbf{Z}^T. \tag{9}$$

Similarly to the case of the original LE, the second constraint is satisfied by discarding the first eigenvector. We can see it by noticing that $\widetilde{\mathbf{A}}\mathbf{1} = \mathbf{0}$ and looking at the eigendecomposition of $\widetilde{\mathbf{C}}$:

$$\widetilde{\mathbf{B}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}^{-\frac{1}{2}}\widetilde{\mathbf{u}}_1 = \widetilde{\mathbf{B}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{x}}^T = \lambda_1\widetilde{\mathbf{u}}_1.$$

Therefore, the solution corresponding to the eigenvalue $\lambda_1 = 0$ is trivial.

4. Experimental evaluation

We compared LLL for LE to two natural baselines. First, Exact LE runs LE on the full dataset. It gives the best embedding, but the runtime is large. Second, Landmark LE runs LE only on a set of landmark points. It gives faster performance, but the embedding quality can be worse because non-landmark points are completely ignored. For Landmark LE we also need to run out-of-sample algorithm to project non-landmark points. We can either compute the projection matrix $\mathbf{Z}$ and use it as an out-of-sample local linear mapping or use Nyström algorithm. For all our experiments we used MATLAB `eigs` function to compute partial eigendecomposition of a sparse matrix.

First, we evaluated the role of the number of landmarks on the performance of the algorithm. We used 60 000 MNIST digits with sparsity $K_W = 200$ and bandwidth $\sigma = 200$ to build the affinity matrix and reduced the dimensionality to $d = 50$. For LLL, we set $K_Z = 50$ and increased the number of landmarks logarithmically from 50 to 60 000. We

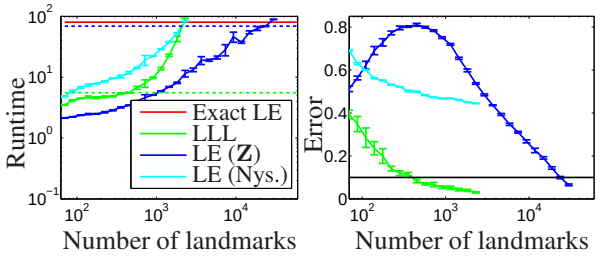

Figure 2. The performance of LLL (in green), Landmark LE with $\mathbf{Z}$ as an out-of-sample (in blue) and Landmark LE with Nyström as an out-of-sample (in cyan). *Left:* runtime as the number of landmarks changes. The green and blue dashed lines correspond to the runtime that gives 10% error with respect to Exact LE for LLL and Landmark LE using $\mathbf{Z}$ respectively. *Right:* The error with respect to Exact LE. The black line corresponds to 10% error. Note the log scale in most of the axes.

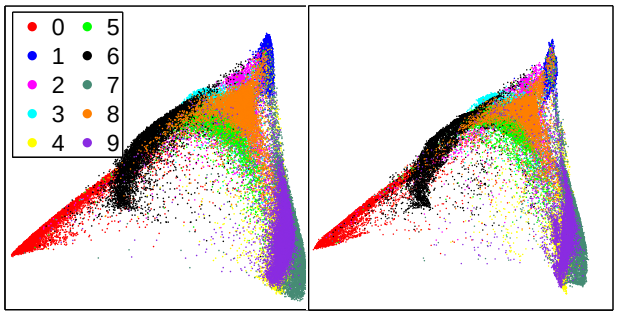

Figure 3. The embedding of 60 000 MNIST digits using the first two dimensions. *Left:* Exact LE ($t = 80$ s.), *right:* LLL ($t = 5.5$ s., 451 landmarks)

chose landmarks at random and repeated the experiment 5 times for different random initialization to see the sensibility of the results to the random choice of the landmarks. To quantitate the error with respect to Exact LE we first used Procrustes alignment (Cox & Cox, 1994, ch. 5) to align the embeddings of the methods and then computed the relative error between aligned embeddings.

In fig. 2 we show the error as well as the overall runtime for different algorithms as the number of landmarks increases. Our first indicator of performance is to see which algorithm can achieve the error of 10% faster. LLL needed 451 landmarks and 5.5 seconds (shown by a dashed green line in the left plot). This is 14 times faster compared to Exact LE which takes 80 seconds. Landmark LE with $\mathbf{Z}$ as an out-of-sample achieves the same error with 23 636 landmarks and the runtime of 69 seconds (1.15 speedup, blue dashed line in the right plot). Landmark LE with Nyström is not able to achieve the error smaller than 50% with any number of landmarks. Notice that the deviation from the mean for 5 runs of randomly chosen landmarks is rather small, suggesting that the algorithm is robust to different locations of landmarks.In fig. 3 we show the embedding of

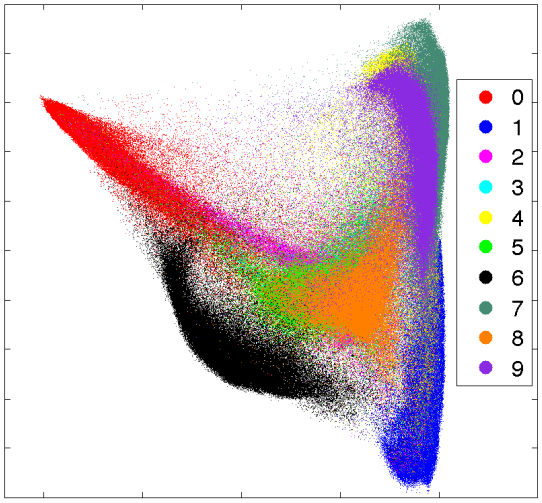

Figure 4. Embedding of 1 020 000 points from infiniteMNIST dataset using LLL with 10 000 landmarks. It took 4.2 minutes to compute the projection matrix $\mathbf{Z}$ and 14 minutes to find the embedding.

Exact LE and the embedding of LLL with 451 randomly selected landmarks. The embedding of LLL is very similar to the one of Exact LE, but the runtime is 15 times faster (5.5 seconds compared to 80 seconds).

Next, for a large-scale experiment we used infinite MNIST dataset from Loosli et al. (2007) where we generated 1 020 000 handwritten digits using elastic deformations to the original MNIST dataset. For LLL we used 10 000 randomly selected landmarks and choose $K_Z = 5$ nearest landmarks. It took the algorithm 4.2 minutes to compute the projection matrix $\mathbf{Z}$ and 14 minutes to compute the embedding. The resulting embedding is available in fig. 4. For a dataset with that many points we cannot run Exact LE with our computer.

5. Conclusion

Typically computing the solution of spectral methods for dimensionality reduction (e.g. Laplacian Eigenmaps (LE), LLE, ISOMAP) or spectral clustering requires computing d trailing eigenvectors of a large $N \times N$ matrix, which is a costly operation when N and d are large. We have proposed an alternative formulation of the problem that optimizes only a small set of landmarks points, while retaining the structure of the whole data. The algorithm is well defined theoretically, has clear computational advantages compared to the original problem and can be applied to speed up the computations and to scale up the applications to much bigger sizes. The method also defines natural out-of-sample extension that is a lot cheaper and better than Nyström method. In case of LE, the method is able to achieve $10-20\times$ speed up with small approximation error.

References

Belkin, Mikhail and Niyogi, Partha. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

Bengio, Yoshua, Delalleau, Olivier, Le Roux, Nicolas, Paiement, Jean-Francois, Vincent, Pascal, and Ouimet, Marie. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10): 2197–2219, 2004a.

Bengio, Yoshua, Paiement, Jean-Francois, Vincent, Pascal, Delalleau, Olivier, , Le Roux, Nicolas, and Ouimet, Marie. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. In *NIPS*, volume 16. MIT Press, Cambridge, MA, 2004.

Carreira-Perpiñán, Miguel Á. and Lu, Zhengdong. The Laplacian Eigenmaps Latent Variable Model. In *AISTATS*, pp. 59–66, San Juan, Puerto Rico, 2007.

Cox, Trevor F. and Cox, M. A. A. *Multidimensional Scaling*. Chapman & Hall, London, New York, 1994.

de Silva, V. and Tenenbaum, Joshua B. Sparse multidimensional scaling using landmark points. 2004.

Drineas, Petros and Mahoney, Michael W. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *JMLR*, 6:2153–2175, 2005.

Gao, Shenghua, Tsang, Ivor Wai-Hung, Chia, Liang-Tien, and Zhao, Peilin. Local features are not lonely — Laplacian sparse coding for image classification. In *CVPR*, pp. 3555–3561, San Francisco, CA, 2010.

Ladický, Ľubor and Torr, Philip H. S. Locally linear support vector machines. In *ICML 2011*, pp. 985–992, Bellevue, WA, 2011.

Lehoucq, R. B. and Sorensen, D. C. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. and Apps.*, 17(4):789–821, 1996.

Liu, Wei, He, Junfeng, and Chang, Shih-Fu. Large graph construction for scalable semi-supervised learning. In *ICML 2010*, Haifa, Israel, 2010.

Loosli, Gaëlle, Canu, Stéphane, and Bottou, Léon. Training invariant support vector machines using selective sampling. In *Large Scale Kernel Machines*, Neural Information Processing Series, pp. 301–320. MIT Press, 2007.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, pp. 849–856. MIT Press, Cambridge, MA, 2002.

Roweis, Sam T. and Saul, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Saul, L. K., Weinberger, K. Q., Ham, J. H., Sha, F., and Lee, D. D. Spectral methods for dimensionality reduction. In *Semi-Supervised Learning*, Adaptive Computation and Machine Learning Series, chapter 16, pp. 293–308. MIT Press, 2006.

Saul, Lawrence K. and Roweis, Sam T. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *JMLR*, 4:June 2003.

Schölkopf, Bernhard, Smola, Alexander, and Müller, Klaus-Robert. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Talwalkar, Ameet, Kumar, Sanjiv, and Rowley, Henry. Large-scale manifold learning. In *CVPR*, Anchorage, AK, 2008.

Tenenbaum, Joshua B., de Silva, Vin, and Langford, John C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Wang, Fa-Yu, Chi, Chong-Yung, Chan, Tsung-Han, and Wang, Yue. Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(5):875–888, 2010.

Weinberger, Kilian, Packer, Benjamin, and Saul, Lawrence. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *AISTATS*, Barbados, 2005.

Weinberger, Kilian Q. and Saul, Lawrence K. Unsupervised learning of image manifolds by semidefinite programming. In *CVPR*, pp. 988–995, Washington, DC, 2004.

Williams, Christopher K. I. and Seeger, Matthias. Using the Nyström method to speed up kernel machines. In *NIPS*, volume 13, pp. 682–688. MIT Press, Cambridge, MA, 2001.

Yu, Kai, Zhang, Tong, and Gong, Yihong. Nonlinear learning using local coordinate coding. In *NIPS*, volume 22. MIT Press, Cambridge, MA, 2009.