The Variational Nyström Method for Large-Scale Spectral Problems

Max Vladymyrov MXV@GOOGLE.COM
Google, Inc.

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

Abstract

Spectral methods for dimensionality reduction and clustering require solving an eigenproblem defined by a sparse affinity matrix. When this matrix is large, one seeks an approximate solution. The standard way to do this is the Nyström method, which first solves a small eigenproblem considering only a subset of landmark points, and then applies an out-of-sample formula to extrapolate the solution to the entire dataset. We show that by constraining the original problem to satisfy the Nyström formula, we obtain an approximation that is computationally simple and efficient, but achieves a lower approximation error using fewer landmarks and less runtime. We also study the role of normalization in the computational cost and quality of the resulting solution.

Spectral problems involve finding eigenvectors of an affinity matrix and have become a standard technique in machine learning problems such as manifold learning (Cox & Cox, 1994; Schölkopf et al., 1998; Tenenbaum et al., 2000; Roweis & Saul, 2000; Belkin & Niyogi, 2003) or spectral clustering (Shi & Malik, 2000; Ng et al., 2002). Their success is due to the power of neighborhood graphs (via an affinity matrix or graph Laplacian) to express similarity between pairs of points, and to the existence of well-developed linear algebra routines to solve the numerical problem. We consider a spectral problem of the type

$$\min_{\mathbf{X}} \operatorname{tr}\left(\mathbf{X}\mathbf{M}\mathbf{X}^T\right) \quad \text{s.t.} \quad \mathbf{X}\mathbf{X}^T = \mathbf{I} \qquad \text{(P)}$$

where $\mathbf{M}$ is an $N \times N$ symmetric matrix (usually, a graph Laplacian) constructed on a high-dimensional dataset $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)$ of $D \times N$, and $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ of $d \times N$ are coordinates in $\mathbb{R}^d$ for the N data points (often called the embedding), where $d < D$. Constraints of the form $\mathbf{X}\mathbf{B}\mathbf{X} = \mathbf{I}$ with positive definite $\mathbf{B}$ can be used with a suitable transformation of $\mathbf{M}$. The solution of (P) is given by the d trailing eigenvectors of $\mathbf{M}$ (note $\mathbf{M}$ need not be positive semidefinite, though it often is). When the number of points N is very large, an exact solution becomes computationally impractical or undesirable, even if $\mathbf{M}$ is sparse. Our goal is to solve problems of the type (P) approximately.

We focus on approximate methods to solve (P) that use sampling, i.e., they solve an eigenproblem on a subset of $L \ll N$ points from $\mathbf{Y}$ ("landmarks") and then use this to extrapolate the solution to all N points. The prototype of these is the *Nyström method* (Williams & Seeger, 2001; Fowlkes et al., 2004), based on an out-of-sample formula that predicts $\mathbf{x} \in \mathbb{R}^d$ for a given point $\mathbf{y} \in \mathbb{R}^D$ as a linear combination of the landmarks' solution using as weights the affinity values between $\mathbf{y}$ and the landmarks. This has the advantage of interpolating the landmarks and being convenient—the weights are simply affinity matrix entries. It gives a good approximation if using sufficiently many landmarks. Its fundamental disadvantage is that the reduced eigenproblem on the landmarks, to which the Nyström formula applies, uses only the landmark-landmark affinity values. If too few landmarks are used, this eigenproblem gives a bad approximation and so does the Nyström extrapolation.

A different approach is that of *Locally Linear Landmarks (LLL)* (Vladymyrov & Carreira-Perpiñán, 2013b), which seeks to define a reduced eigenproblem containing more information than just landmark-landmark affinities. LLL defines a different out-of-sample formula, a linear combination of the projections of the nearest landmarks to $\mathbf{y}$ using weights that reconstruct $\mathbf{y}$ locally linearly in input space. The crucial idea in LLL is that problem (P) is solved constrained to using these weights, resulting in a reduced eigenproblem that does use the entire affinity matrix. Hence, this obtains a better landmark embedding than the Nyström method for the same number of landmarks.

This reasoning naturally leads to our first contribution, the

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

Variational Nyström (VN) method, where we incorporate the Nyström formula as a constraint in (P). As in LLL, we obtain a reduced eigenproblem that uses the entire affinity matrix and thus better represents the manifold structure of the landmarks. This reduced eigenproblem is then "optimal" for the Nyström formula (unlike the one based only on the landmark-landmark affinities). We also save the expensive computation of the LLL weights. We call it "variational" Nyström to refer to its optimality motivation.

Our second contribution addresses an issue that has so far been overlooked in Nyström-type methods: how to use subsampling approximations with data-dependent kernels (e.g. graph Laplacian)? There each kernel element is generated not only by the corresponding points from the original data, but from the other points as well. In this case, applying the approximations directly to the kernel gives bad results. We investigate ways to normalize the data-dependent kernel in order to get best performance with VN and other methods.

Notation. $\widetilde{\mathbf{A}}$ indicates that $\mathbf{A}$ is approximated. $\hat{\mathbf{Y}}$ indicates a landmark subset of $\mathbf{Y}$. A subscript shows to which matrix we apply a certain transformation, e.g. $\mathbf{U_P}$ is a column matrix of eigenvectors of $\mathbf{P}$ and $\mathbf{D_W} = \operatorname{diag}(\mathbf{W1})$ is a degree matrix for $\mathbf{W}$ (and $\mathbf{1}$ is a vector of ones). For degree matrices that are computed for rectangular matrices, an arrow indicates whether the sum is taken row- or column-wise, e.g. for an $N \times L$ matrix $\mathbf{C}$, $\mathbf{D_{C\rightarrow}} = \operatorname{diag}(\mathbf{C1})$ is a $N \times N$ matrix of row-wise sums and $\mathbf{D_{C\downarrow}} = \operatorname{diag}(\mathbf{1C})$ is an $L \times L$ matrix of column-wise sums.

1. Prior Work

The *Nyström method* is the most widely used sampling method in machine learning to approximate the computation of the eigenvectors of a large matrix. It was originally proposed as a quadrature method for numerical integration to approximate eigenfunctions of continuous operators (Atkinson, 1997). Williams & Seeger (2001) introduced it to machine learning to approximate the eigenvectors of a large kernel matrix, as in Gaussian processes. After that it was used for many other applications: kernel methods (Zhang et al., 2008; Zhang & Kwok, 2010; Cortes et al., 2010; Yang et al., 2012), spectral clustering (Fowlkes et al., 2001; Belongie et al., 2002; Fowlkes et al., 2004), manifold learning (Platt, 2004; Talwalkar et al., 2013), etc.

Consider a symmetric $N \times N$ matrix $\mathbf{M} = (m_{ij})$ whose elements come from applying a certain affinity function $K(\cdot,\cdot)$ on pairs of points from $\mathbf{Y}$, i.e., $m_{ij} = K(\mathbf{y}_i, \mathbf{y}_j)$ for all $i, j = 1, \dots, N$. Let its eigendecomposition be $\mathbf{M} = \mathbf{U_M \Lambda_M U_M^T}$. W.l.o.g., let $\hat{\mathbf{Y}}_{D\times L}$ be the first L data points (columns) of $\mathbf{Y}$. Write $\mathbf{M}$ by blocks:

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B}_{21}^T \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B}_{21} \end{pmatrix} \tag{1}$$

so $\mathbf{C}_{N\times L}$ are the columns of $\mathbf{M}$ that correspond to $\hat{\mathbf{Y}}$. The Nyström method uses the eigendecomposition of the small matrix $\mathbf{A} = \mathbf{U_A \Lambda_A U_A^T}$ to approximate the eigenvectors $\mathbf{U_M}$ and eigenvalues $\mathbf{\Lambda_M}$ of the large matrix $\mathbf{M}$ as

$$\widetilde{\mathbf{U}}_{\mathbf{M}}^{\text{Nys}} = \begin{pmatrix} \mathbf{U_A} \\ \mathbf{B}_{21}\mathbf{U_A}\mathbf{\Lambda_A}^{-1} \end{pmatrix}, \quad \widetilde{\mathbf{\Lambda}}_{\mathbf{M}}^{\text{Nys}} = \mathbf{\Lambda_A}. \tag{2}$$

The Nyström extension reconstructs the embedding of the landmarks $\hat{\mathbf{Y}}$ exactly, and approximates the rest of the embedding with $\mathbf{B}_{21}\mathbf{U_A}\mathbf{\Lambda_A}^{-1}$. It approximates $\mathbf{M}$ by

$$\begin{aligned} \widetilde{\mathbf{M}}^{\text{Nys}} &= \widetilde{\mathbf{U}}_{\mathbf{M}}^{\text{Nys}} \widetilde{\mathbf{\Lambda}}_{\mathbf{M}}^{\text{Nys}} (\widetilde{\mathbf{U}}_{\mathbf{M}}^{\text{Nys}})^T = \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T \\ &= \begin{pmatrix} \mathbf{A} & \mathbf{B}_{21}^T \\ \mathbf{B}_{21} & \mathbf{B}_{21}\mathbf{A}^{-1}\mathbf{B}_{21}^T \end{pmatrix}. \end{aligned}$$

The *column sampling (CS)* method (Frieze et al., 1998) approximates the eigenvectors of $\mathbf{M}$ using left singular vectors of $\mathbf{C}$. If $\operatorname{svd}(\mathbf{C}) = \mathbf{U_C \Sigma_C V_C^T}$, then $\widetilde{\mathbf{U}}_{\mathbf{M}}^{\text{CS}} = \mathbf{U_C}$. Alternatively, this approximation can be computed using the eigenvectors of the matrix $\mathbf{C}^T\mathbf{C}$. If $\mathbf{Z} = \mathbf{C}^T\mathbf{C} = \mathbf{V_C \Sigma_C^2 V}_C^T$, then the eigenvectors and eigenvalues of $\mathbf{Z}$ can be expressed using the SVD of $\mathbf{C}$ as $\mathbf{U_Z} = \mathbf{V_C}$ and $\mathbf{\Lambda_Z} = \mathbf{\Sigma_C^2}$. The approximation of $\mathbf{U_M}$ becomes $\widetilde{\mathbf{U}}_{\mathbf{M}}^{\text{CS}} = \mathbf{U_C} = \mathbf{U_C \Sigma_C V_C^T V_C \Sigma_C^{-1}} = \mathbf{C V_C \Sigma_C^{-1}} = \mathbf{C U_Z \Lambda_Z^{-1/2}}$. The approximation of $\mathbf{M}$ is then $\widetilde{\mathbf{M}}^{\text{CS}} = \mathbf{C}^T\mathbf{C}$. This method uses more information from $\mathbf{M}$ than the Nyström method and, intuitively, should perform better. Talwalkar et al. (2013) show empirically that, in most cases, this is true.

The *Locally Linear Landmarks (LLL)* method (Vladymyrov & Carreira-Perpiñán, 2013b) is instead motivated by the minimization problem (P). The Nyström method is, at its core, only an extrapolation formula, whose success relies on having a good solution for the landmarks. However, the latter need not hold if one uses too few landmarks, because the reduced eigenproblem only uses the landmark-landmark affinities in $\mathbf{A}$. In LLL, the reduced eigenproblem does use all the information from the affinity matrix $\mathbf{M}$. LLL constrains the solution of (P) to obey an out-of-sample formula $\mathbf{X} = \hat{\mathbf{X}}\mathbf{Z}^T$, where $\hat{\mathbf{X}}_{d\times L}$ is the solution for landmarks $\hat{\mathbf{Y}}_{D\times L}$. The weights $\mathbf{Z}_{N\times L}$ assume that the projection $\mathbf{x} \in \mathbb{R}^d$ of a given point $\mathbf{y} \in \mathbb{R}^D$ is a locally linear function of its nearest landmarks' projections. The weights are found from the datapoints $\mathbf{Y}$ alone as

$$\min_{\mathbf{Z}} \|\mathbf{Y} - \hat{\mathbf{Y}}\mathbf{Z}^T\|_F^2 \quad \text{s.t.} \quad \mathbf{Z1} = \mathbf{1}, \tag{3}$$

that is, such that $\mathbf{y}$ itself is a locally linear function of its nearest landmarks. This represents the manifold learning assumption that local structure in the high-dimensional space should be preserved in the low-dimensional space, as originally formulated in Locally Linear Embedding (LLE) (Roweis & Saul, 2000). When this assumption holds, LLL produces better approximations than the Nyström method in less runtime. However, computing the reconstruction

weights $\mathbf{Z}$ requires knowing which landmarks are neighbors of each data point and is expensive. It also means that LLL does not apply when only the affinity matrix but not the feature vectors $\mathbf{Y}$ are given.

The *random projection (RP)* algorithm (Halko et al., 2011; Boutsidis et al., 2011) first uses a random matrix $\mathbf{S}_{N\times L}$ to form a low-dimensional sample matrix $\mathbf{M_S} = \mathbf{MS}$, which is an approximation to the range of $\mathbf{M}$. Then, it computes the SVD of $\mathbf{M}$ projected onto the orthogonal basis $\mathbf{Q}$ of $\mathbf{M_S}$ (found with the QR decomposition). The final step is a backward projection of the left singular vectors to the original space. To improve results even further, Halko et al. (2011) propose to construct the sample matrix as $\mathbf{M_S} = \mathbf{M}^q\mathbf{S}$ for $q \geq 1$. This is like performing q power method iterations to make the eigenspace of the projection be similar to the eigenspace of $\mathbf{M}$. The bottleneck of RP is the expensive computation of the random projection matrix $\mathbf{M_S}$ (since $\mathbf{S}$ is dense) and the basis $\mathbf{Q}$.

Li et al. (2010) combine the ideas of both RP and Nyström. On the one hand, Nyström is fast, but it relies on the reduced affinity matrix $\mathbf{A}$ capturing the structure of the data, which requires a large number of landmarks. On the other hand, RP methods give a good approximation of the data with a smaller number of samples, but are expensive to run. Li et al. (2010) propose to approximate the eigenvectors of a sample with random projections and then use the Nyström formula to extrapolate the solution to the whole space.

The *modified Nyström* method (Wang & Zhang, 2013) proposes $\mathbf{C}\left(\mathbf{C}^+\mathbf{A}(\mathbf{C}^+)^T\right)\mathbf{C}^T$ as a low-rank approximation to $\mathbf{M}$. The eigenvectors of this matrix coincide with those of the Variational Nyström method (see suppl.mat.). However, modified Nyström was presented as a low-rank matrix approximation technique and does not approximate the solution of the spectral problem (P), nor is it obvious how it can be used for the task of approximating the coordinates of the low-dimensional points. In addition, while Wang & Zhang (2013) gave bounds for the approximation quality, the formula itself was presented with no derivation. Our paper justifies their choice of low-rank matrix approximation from the spectral learning point of view.

The Nyström formula is not the only possible out-of-sample extension. Another one is the *Laplacian Eigenmaps Latent Variable Model* (Carreira-Perpiñán & Lu, 2007), which has the form of a Nadaraya-Watson estimator, and thus also provides a conditional density of $\mathbf{x}$ given $\mathbf{y}$.

Apart from landmark-based methods, it is also possible to approximate the target eigenvectors using an incomplete iterative eigendecomposition, e.g. *approximate Krylov methods*. However, working with landmarks has some important advantages in machine learning. First, they require a single, intuitive parameter (the number of landmarks L) that is easy to set (as large as computationally feasible, typically), while iterative Krylov methods have multiple non-trivial parameters (e.g. maximum number of iterations, tolerance, number of vectors to retain). Second, landmark-based methods can compute more eigenvectors easily because these appear in the reduced eigenproblem. Third, we observe that the eigendecomposition of the reduced matrix behaves more robustly than that of the original sparse matrix (for which, depending on its sparsity, we sometimes observe convergence problems). Fourth, the cost-dominant operations (constructing the reduced eigenproblem, applying the out-of-sample mapping, computing the weights for LLL) are trivial to parallelize and require each a single pass over the disk if out-of-core. The inherent sequentiality of Krylov methods involves parallelization only within an iteration, and one pass over the disk per iteration. Finally, landmark-based methods can use any eigensolver as a black box for the reduced eigenproblem.

2. Variational Nyström (VN)

We now state formally our proposed Variational Nyström method. This finds an approximate solution of problem (P) by constraining $\mathbf{X}$ to be a l.c. of the landmarks' embedding $\widetilde{\mathbf{X}}$ using as coefficients the point-landmark affinities, i.e.,

$$\min_{\mathbf{X}} \operatorname{tr}\left(\mathbf{X}\mathbf{M}\mathbf{X}^T\right) \quad \text{s.t.} \quad \mathbf{X}\mathbf{X}^T = \mathbf{I},\ \mathbf{X} = \widetilde{\mathbf{X}}\mathbf{C}^T \tag{4}$$

where $\mathbf{M}$ is partitioned as in (1) and $\widetilde{\mathbf{X}}$ is of $L \times N$. This results in a *reduced eigenproblem* of $L \times L$ for $\widetilde{\mathbf{X}}$:

$$\min_{\widetilde{\mathbf{X}}} \operatorname{tr}\left(\widetilde{\mathbf{X}}\mathbf{C}^T\mathbf{M}\mathbf{C}\widetilde{\mathbf{X}}^T\right) \quad \text{s.t.} \quad \widetilde{\mathbf{X}}\mathbf{C}^T\mathbf{C}\widetilde{\mathbf{X}}^T = \mathbf{I} \tag{5}$$

whose exact solution $\widetilde{\mathbf{X}} = \widetilde{\mathbf{U}}$ is given by the d trailing eigenvectors of the generalized eigenproblem

$$(\mathbf{C}^T\mathbf{M}\mathbf{C})\widetilde{\mathbf{U}} = (\mathbf{C}^T\mathbf{C})\widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}}. \tag{6}$$

Hence, the solution for the full embedding is $\mathbf{X} = \widetilde{\mathbf{U}}\mathbf{C}^T$. This can be seen as the answer to the question "what is the best matrix $\mathbf{Q}$ (instead of $\mathbf{Q} = \mathbf{U_A}\mathbf{\Lambda_A}^{-1}$ as in the Nyström method) that can be used if the out-of-sample weights are $\mathbf{C}$?" By construction, VN will find a better approximate embedding in (P) than Nyström's method (and CS). From the LLL perspective, we abandon the local linearity assumption, also saving the computational cost of LLL's weights. The reduced affinity matrix $\mathbf{C}^T\mathbf{M}\mathbf{C}$ (of $L \times L$) uses the information from all the points in $\mathbf{M}$, unlike the Nyström reduced affinity matrix $\mathbf{A}$, and we expect it to represent the manifold structure better.

Runtime It consists of 3 parts. 1) Setting up the reduced eigenproblem. Expanding submatrices using (1) we get $\mathbf{C}^T\mathbf{M}\mathbf{C} = \mathbf{A}^3 + \mathbf{B}_{21}^T\mathbf{B}_{21}\mathbf{A} + (\mathbf{B}_{21}^T\mathbf{B}_{21}\mathbf{A})^T + \mathbf{B}_{21}^T\mathbf{B}_{22}\mathbf{B}_{21}$ and $\mathbf{C}^T\mathbf{C} = \mathbf{A}^2 + \mathbf{B}_{21}^T\mathbf{B}_{21}$. Computing $\mathbf{C}^T\mathbf{C}$ is free as we compute the first two terms of $\mathbf{C}^T\mathbf{M}\mathbf{C}$. The cost is

dominated by $\mathbf{B}_{21}^T\mathbf{B}_{22}\mathbf{B}_{21}$, which is between $\mathcal{O}(N)$ and $\mathcal{O}((N-L)^2L)$ depending on the sparsity of $\mathbf{M}$. 2) Solving the reduced eigenproblem is between $\mathcal{O}(L)$ and $\mathcal{O}(L^3)$ depending on its sparsity. 3) Applying the out-of-sample formula is between $\mathcal{O}(L)$ and $\mathcal{O}(NL)$. A rigorous runtime is difficult to obtain without specifying the sparsity pattern of $\mathbf{M}$, but in the practical case where $L \ll N$ and $\mathbf{M}$ is sufficiently sparse, the cost is dominated by setting up the eigenproblem and is around $\mathcal{O}(NL)$, i.e., linear in N.

Assumptions VN makes the following assumptions, shared with the Nyström method. 1) The data (i.e., affinity matrix or graph Laplacian $\mathbf{M}$) is given as part of the problem definition. In practice, this requires the construction of a nearest-neighbor graph on the dataset $\mathbf{Y}$. For large datasets, this may require using approximate nearest neighbor techniques (in some cases, the graph may be known a priori, as in image segmentation). 2) The affinity matrix fits into main memory. If it does not fit in memory but it fits in local disk, VN can be implemented efficiently and easily. We simply construct the reduced affinity matrix in (6) by reading from disk incrementally. If the data is distributed over machines with local memory/disk, or if the reduced affinity matrix itself does not fit into memory, the problem is more difficult, and a topic of future research.

Unlike LLL, VN does not need the actual feature vectors $\mathbf{Y}$, only their affinity matrix $\mathbf{M}$. This makes VN applicable when the affinity between two points is a sophisticated function of their context, as in image segmentation using intervening contours cues (Cour et al., 2005).

How to select the landmarks from $\mathbf{Y}$? In this paper, we focus on random selection. This works reasonably well most times and has minimal overhead. Many selection mechanisms exist, such as k-means, greedy MaxMin (de Silva & Tenenbaum, 2004) or leverage scores (Mahoney, 2011), but we find random landmarks give a better error-runtime tradeoff unless one uses very few landmarks.

Connection between methods Many of the methods described in this paper can be viewed as approximating the solution with an extrapolation $\widetilde{\mathbf{U}}_\mathbf{M} = \mathbf{ZQ}$, where $\mathbf{Z}_{N\times L}$ is a precomputed matrix of out-of-sample weights, and $\mathbf{Q}_{L\times d}$ is a matrix that depends on the eigendecomposition of a reduced $(L \times L)$ affinity matrix over the landmarks. These methods can then be classified along two axes: how the out-of-sample weight matrix is defined, and how the reduced eigenproblem is set up, as summarized in table 1.

We note the following. Nyström, CS and VN all use the same out-of-sample weight matrix $\mathbf{Z} = \mathbf{C}$, given by actual entries in the affinity matrix, which is simple and efficient. LLL is the only method that depends on the metric structure of the feature vectors $\mathbf{Y}$ (through the reconstruction weights $\mathbf{Z}$) instead of just the elements of $\mathbf{M}$. VN and LLL are the only methods to define a generalized eigenproblem, which is slightly slower to solve than a regular eigenproblem. LLL, VN and RP (see suppl.mat.) can be seen as optimizing (P) over $\mathbf{Q}$ for a certain choice of $\mathbf{Z}$. In contrast, the Nyström method chooses weights $\mathbf{Z}$ such that, when applying the out-of-sample formula to the landmarks themselves, the result equals the reduced eigenproblem solution.

Table 1. Choices of different algorithms that approximate the matrix $\mathbf{M}$'s eigenvectors as $\widetilde{\mathbf{U}}_\mathbf{M} = \mathbf{ZQ}$, where $\mathbf{Z}$ is the out-of-sample weight matrix and $\mathbf{Q}$ the solution for the landmarks.

Algorithm	$\mathbf{Z}_{N\times L}$	$\mathbf{Q}_{L\times d}$	Eigenproblem $\mathcal{A}\mathbf{U} = \mathcal{B}\mathbf{U}\mathbf{\Lambda}$ $\mathcal{A}$, $\mathcal{B}$
Nyström	$\mathbf{C}$	$\mathbf{U}\mathbf{\Lambda}^{-1}$	$\mathbf{A}$, $\mathbf{I}$
CS	$\mathbf{C}$	$\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}$	$\mathbf{Z}^T\mathbf{Z}$, $\mathbf{I}$
RP	$\mathrm{qr}(\mathbf{M}^q\mathbf{S})$	$\mathbf{U}$	$\mathbf{Z}^T\mathbf{M}\mathbf{Z}$, $\mathbf{I}$
LLL	eq. (3)	$\mathbf{U}$	$\mathbf{Z}^T\mathbf{M}\mathbf{Z}$, $\mathbf{Z}^T\mathbf{Z}$
VN	$\mathbf{C}$	$\mathbf{U}$	$\mathbf{Z}^T\mathbf{M}\mathbf{Z}$, $\mathbf{Z}^T\mathbf{Z}$

In terms of increasing amount of affinity information used to construct the reduced eigenproblem, the methods can be ranked as Nyström $<$ CS $<$ {RP, LLL, VN}. Nyström and CS discard most of the affinity matrix $\mathbf{M}$ (they never use $\mathbf{B}_{22}$ in eq. (1), the affinities between non-landmarks), while LLL and VN use all of it. Entirely discarding so much of $\mathbf{M}$ may seem like an advantage, because we save the cost of computing those affinities in the first place, which is large if $\mathbf{M}$ is not sparse. But even in this case it may be better to sparsify $\mathbf{M}$ (e.g. by zeroing small elements) and use the non-landmarks affinities in LLL and VN.

3. Subsampling Graph Laplacians

So far, we have discussed various ways to approximate an eigendecomposition of the matrix $\mathbf{M}$ generated by some kernel K. This kernel usually represents similarity between points in the dataset $\mathbf{Y}$, e.g. for the Gaussian kernel $w_{ij} = K(\mathbf{y}_i, \mathbf{y}_j) = \exp(-\|\mathbf{y}_i^2 - \mathbf{y}_j^2\|/2\sigma^2)$. The problem is that often the kernel is *data dependent* (Bengio et al., 2004), i.e., the element m_{ij} depends not just on a pair $(\mathbf{y}_i, \mathbf{y}_j)$ but on other elements as well. For example, a graph Laplacian $\mathbf{L}_\mathbf{W}$ (unnormalized as $\mathbf{L}_\mathbf{W} = \mathbf{D}_\mathbf{W} - \mathbf{W}$ or normalized as $\mathbf{L}_\mathbf{W} = \mathbf{D}_\mathbf{W}^{-1/2}(\mathbf{D}_\mathbf{W} - \mathbf{W})\mathbf{D}_\mathbf{W}^{-1/2}$) constructed on a subset of L input points is not equal to an $L\times L$ subset of the graph Laplacian constructed on N points (while for the Gaussian affinity matrix $\mathbf{W}$, both cases do give the same result). This happens because the graph Laplacian depends on the degree matrix $\mathbf{D}_\mathbf{W} = \mathrm{diag}\,(\mathbf{W1})$ and this couples all the elements together. This can cause problems for approximation methods, such as Nyström, CS or VN, whose projection $\mathbf{Z}$ depends on the $L \times L$ subsampled matrix, since the solution does vary depending on how we define the subsampled graph Laplacian. LLL or RP do not have this

problem, since their out-of-sample matrix either does not depend on the similarity matrix (for LLL) or involves the entire similarity matrix with no subsampling (for random projections). In this section we investigate which out-of-sample kernel would give better performance for Nyström, CS or VN for approximating the normalized graph Laplacian matrix $\mathbf{L_W} = \mathbf{D_W}^{-1/2}(\mathbf{D_W} - \mathbf{W})\mathbf{D_W}^{-1/2}$. As far as we know, we are the first to propose such an analysis.

When using the Nyström method, two ways have been previously proposed to approximate the graph Laplacian. In the first one, Fowlkes et al. (2004) first apply Nyström to approximate $\mathbf{W}$ to find an estimate of the degree matrix as $\widetilde{\mathbf{D}} = \operatorname{diag}(\widetilde{\mathbf{W}}\mathbf{1}) = \operatorname{diag}\begin{pmatrix} \mathbf{A}\mathbf{1}+\mathbf{B}_{21}^T\mathbf{1} \\ \mathbf{B}_{21}\mathbf{1}+\mathbf{B}_{21}\mathbf{W}_L^{-1}\mathbf{B}_{21}^T\mathbf{1} \end{pmatrix}$ and then use Nyström one more time to find the leading eigenvectors of $\widetilde{\mathbf{D}}^{-1/2}\mathbf{W}\widetilde{\mathbf{D}}^{-1/2}$ that coincide with the eigenvectors of the normalized graph Laplacian. Hence, they apply the Nyström method twice: first to approximate the degree matrix and then to approximate the graph Laplacian. In the second way, Bengio et al. (2004) approximate the normalized graph Laplacian $\mathbf{L_W}$ by defining a subset problem as $\mathbf{L_A} = \mathbf{D_A}^{-1/2}\mathbf{A}\mathbf{D_A}^{-1/2}$ with $\mathbf{D_A} = \operatorname{diag}(\mathbf{A}\mathbf{1})$. After this subproblem is solved, non-landmark points are projected using an out-of-sample kernel $\mathbf{Z} = \mathbf{D}_{\mathbf{C}\rightarrow}^{-1/2}\mathbf{C}\mathbf{D_A}^{-1/2}$, where $\mathbf{D}_{\mathbf{C}\rightarrow} = \operatorname{diag}(\mathbf{C}\mathbf{1})$ is an $N \times N$ row-wise sum of $\mathbf{C}$ which corresponds to a degree matrix of the affinities between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. It is easy to see that $\mathbf{Z}$ contains $\mathbf{L_A}$ as a subset. Bengio et al. (2004) provided no justification of their choice of the out-of-sample kernel.

Here, we propose a more general approach based on two general criteria that define a good normalization of the out-of-sample kernel. First, it should interpolate over the landmarks (i.e., the out-of-sample matrix for the subset should be equal to the subset matrix). Second, the normalized matrix should agree with $\mathbf{M}$ when $L = N$ (i.e., the approximation becomes exact when number of landmarks is equal to the number of points). From the first criterion we see that $\mathbf{Z}$ should be normalized as $\mathbf{Z} = \mathbf{D}_1\mathbf{C}\mathbf{D}_2$ for some diagonal matrices $\mathbf{D}_1 \in \mathbb{R}^{N\times N}$ and $\mathbf{D}_2 \in \mathbb{R}^{L\times L}$. In the following proposition (proven in the suppl.mat.) we show which $\mathbf{D}_1$ and $\mathbf{D}_2$ satisfy our second criterion for the Nyström and CS methods.

Proposition 3.1 (Normalization for Nyström and Column Sampling)**.** *Given a subsample* $\mathbf{L_A^{Nys}} = \mathbf{D_A}^{-1/2}\mathbf{A}\mathbf{D_A}^{-1/2}$ *for the Nyström method, or* $\mathbf{L_A^{CS}} = (\mathbf{D}_1\mathbf{C}\mathbf{D}_2)^T(\mathbf{D}_1\mathbf{C}\mathbf{D}_2)$ *for the Column Sampling, and* $\mathbf{Z} = \mathbf{D}_1\mathbf{C}\mathbf{D}_2$ *as an out-of-sample kernel, the exact eigenvectors of graph Laplacian* $\mathbf{L_W}$ *for* $L = N$ *are recovered when* $\mathbf{D}_1 = \mathbf{D}_2 = \mathbf{D_W}^{-1/2}$.

The equation $\mathbf{D}_1 = \mathbf{D}_2 = \mathbf{D_W}^{-1/2}$ is satisfied only when $L = N$ and for $L < N$ it needs to be approximated. Table 2 shows four different choices that we consider (named according to the affinity matrices used in $\mathbf{D}_1$ and $\mathbf{D}_2$). For $\mathbf{D}_1$ we can either sum the rows of $\mathbf{C}$ as $\mathbf{D}_{\mathbf{C}\rightarrow} = \operatorname{diag}(\mathbf{C}\mathbf{1})$ or the rows of the whole $\mathbf{W}$ as $\mathbf{D_W} = \operatorname{diag}(\mathbf{W}\mathbf{1})$. For $\mathbf{D}_2$ we can either sum the columns of $\mathbf{C}$ as $\mathbf{D}_{\mathbf{C}\downarrow} = \operatorname{diag}(\mathbf{1}\mathbf{C})$ or the columns or $\mathbf{A}$ as $\mathbf{D_A} = \operatorname{diag}(\mathbf{A}\mathbf{1})$. Note that `CA` corresponds to the kernel proposed by Bengio et al. (2004). In section 4 we evaluate these choices empirically.

Table 2. Graph Laplacian normalization for Nyström and CS.

	`WA`	`WC`	`CA`	`CC`
$\mathbf{D}_1$	$\mathbf{D_W}^{-1/2}$	$\mathbf{D_W}^{-1/2}$	$\mathbf{D}_{\mathbf{C}\rightarrow}^{-1/2}$	$\mathbf{D}_{\mathbf{C}\rightarrow}^{-1/2}$
$\mathbf{D}_2$	$\mathbf{D_A}^{-1/2}$	$\mathbf{D}_{\mathbf{C}\downarrow}^{-1/2}$	$\mathbf{D_A}^{-1/2}$	$\mathbf{D}_{\mathbf{C}\downarrow}^{-1/2}$

Table 3. Graph Laplacian normalization for Variational Nyström.

	`None`	`Sqrt`	`Sum`	`Direct`
$\mathbf{D}_2$	$\mathbf{I}$	$\mathbf{D}_{\mathbf{C}\downarrow}^{-1/2}$	$\mathbf{D}_{\mathbf{C}\downarrow}^{-1}$	$\mathbf{D_W}^{-1/2}$

For Variational Nyström the normalization is more general.

Proposition 3.2 (Normalization for Variational Nyström)**.** *Given a subsample* $\mathbf{L_A^{VN}} = \mathbf{D_A}^{-1/2}\mathbf{A}\mathbf{D_A}^{-1/2}$ *and* $\mathbf{Z} = \mathbf{D}_1\mathbf{C}\mathbf{D}_2$ *as an out-of-sample kernel, the exact eigenvectors of graph Laplacian* $\mathbf{L_W}$ *are recovered for any* $L \leq N$ *using any arbitrary symmetrical matrix* $\mathbf{D}_1$. *When* $L = N$ *the eigenvectors are recovered using any symmetrical* $\mathbf{D}_2$.

This means that for $L = N$ any matrices $\mathbf{D}_1, \mathbf{D}_2$ (not only diagonal) result in the exact solution. Moreover, for $\mathbf{D}_1$ this is the case even when $L < N$. Thus, the normalization takes the form $\mathbf{Z} = \mathbf{C}\mathbf{D}_2$. In table 3 we show four different choices for $\mathbf{D}_2$. Note the `Direct` choice corresponds to $(\mathbf{Z}\mathbf{D_W}^{-1/2}\mathbf{W}\mathbf{D_W}^{-1/2}\mathbf{Z}^T)\mathbf{U} = (\mathbf{Z}\mathbf{D_W}^{-1/2}\mathbf{D_W}\mathbf{D_W}^{-1/2}\mathbf{Z}^T)\mathbf{U}\mathbf{\Lambda}$ or $\big(\mathbf{Z}\big(\mathbf{D_W}^{-1/2}\mathbf{W}\mathbf{D_W}^{-1/2}\big)\mathbf{Z}^T\big)\mathbf{U} = \big(\mathbf{Z}\mathbf{Z}^T\big)\mathbf{U}\mathbf{\Lambda}$ and is the same as direct application of VN to $\mathbf{L_W}$.

4. Experiments

To set up the spectral problem (P) we want to approximate, we use Laplacian Eigenmaps (LE) (Belkin & Niyogi, 2003), a spectral manifold learning algorithm. We also use spectral clustering (SC) (Shi & Malik, 2000) in an image segmentation experiment. We compare the approximations to the exact embedding $\mathbf{X}$ or the ground truth, when available. (Note there is no point in comparing VN and Nyström on the basis of (P) because VN is better by construction.) To compute $\mathbf{X}$, we use Matlab's `eigs` routine with default parameters ($\texttt{maxit} = 300, \texttt{tol} = \texttt{eps}, \texttt{p} = 2d$). `eigs` uses an iterative algorithm suitable for large problems, since running `eig` (which uses a direct algorithm) is too expensive for N as small as 5 000. We report least-square relative errors between the embeddings after Procrustes alignment (since problem (P) is invariant to rotation and translation). We do not compare the value of the objective function in (P) because it is not indicative of the solution quality by itself, since the problem is constrained.

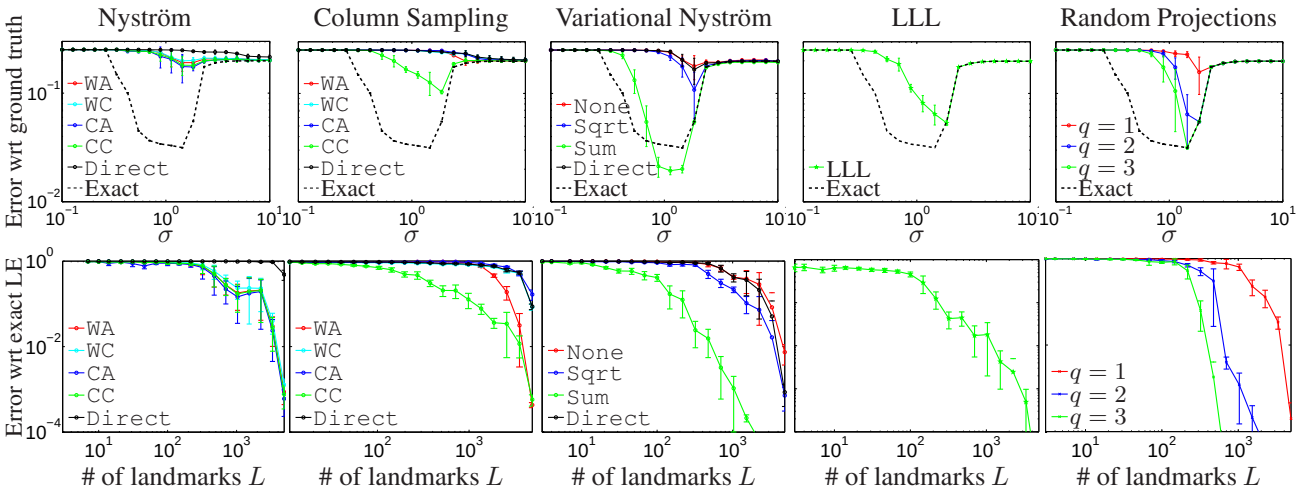

Figure 1. 5 000 points from the Swiss roll dataset. *Top:* error for each of the models wrt the ground truth as a function of the affinity bandwidth σ. The black solid curve specifies the error of exact Laplacian eigenmaps with respect to the ground truth. $L = 500$, $K_W = 500$. *Bottom:* normalization error of different models with respect to exact LE as a function of number of landmarks. $\sigma = 1$, $K_W = 500$.

Out-of-sample normalization First, we evaluate the different choices of out-of-sample normalizations proposed in section 3. We used 5 000 points from the Swiss roll dataset, for which the ground truth is available. For Nyström, CS and VN we used different normalizations and also the `direct` approach, where the approximations are applied directly to the graph Laplacian $\mathbf{L_W}$ (as it is done in e.g. Fowlkes et al., 2004). LLL and RP are independent from the normalization and we run them just for comparison. For RP, we tried different values of q, the number of power iterations used before the projection is applied. Each additional power iteration improves the approximation, but also increases the runtime.

Fig. 1 (top) shows the relative error with respect to the ground truth as we change the Gaussian affinity bandwidth σ. We tried 20 different σ values log-spaced between 10^{-1} and 10. The error bars show the results over 5 runs with different sets of landmarks ($L = 500$, randomly selected). The black dashed line indicates the error of exact LE with respect to the ground truth. There is a distinct region somewhere between 0.26 and 2.33 for which the exact LE is closest to the ground truth. Fig. 1 (bottom) shows all the approximations and normalizations, but now fixing $\sigma = 1$ and varying L (20 log-spaced values in between 3 and 4 900). We compared the embedding with respect to the results of the exact LE. Notice that optimizing $\mathbf{L_W}$ directly, with no normalization, gives the worst results for all the methods. This suggests that the data-dependent kernels, such as graph Laplacian, should be approximated carefully with a custom out-of-sample matrix.

The results of different normalizations are consistent for both experiments. For Nyström all the approximations perform badly for any σ, with the `CA` and `CC` normalizations being a little better. For CS, `CC` gives much better results than any other normalization. Thus, it is much better to sum up all the columns of $\mathbf{C}$, rather than just a sum of the sample from $\mathbf{A}$. For Nyström it is the other way around, however the difference is not that significant.

For VN the `Sum` approximation clearly gives the best results. Interestingly, for some σ it gives an error that is even lower than the exact LE. This is probably a coincidence (after all VN tries to approximate whatever result we should get for the exact LE), although the VN approximation might have a regularizing effect on the embedding. In multiple experiments we have observed that VN *robustifies* the spectral problem. That is, solving the exact problem for LE using Matlab's `eigs` sometimes fails to converge, or does not return the trailing eigenvectors requested. The situations when this happens are somewhat unpredictable, but it seems more likely with small bandwidth values. This leads to wrong LE embeddings or to no solution at all. With VN, the behavior of `eigs` is much more robust, providing correct solutions in all our experiments. Although we do not have a theoretically rigorous explanation for this, empirically it is very noticeable. Vladymyrov & Carreira-Perpiñán (2013b) also observed a similar effect with LLL. We conjecture this is because both VN and LLL construct a reduced eigenproblem that is less sparse, better connecting data points, and easier to solve numerically.

LLL robustly gives a good approximation, better than Nyström and CS, but worse than VN. RP improves dramatically as q increases, however the variability between runs also increases and, as we show later, for a given L its runtime is much larger than for the other methods. Also, RP needs a relatively large number of landmarks ($L \leq 100$) in order for the error to start decreasing.

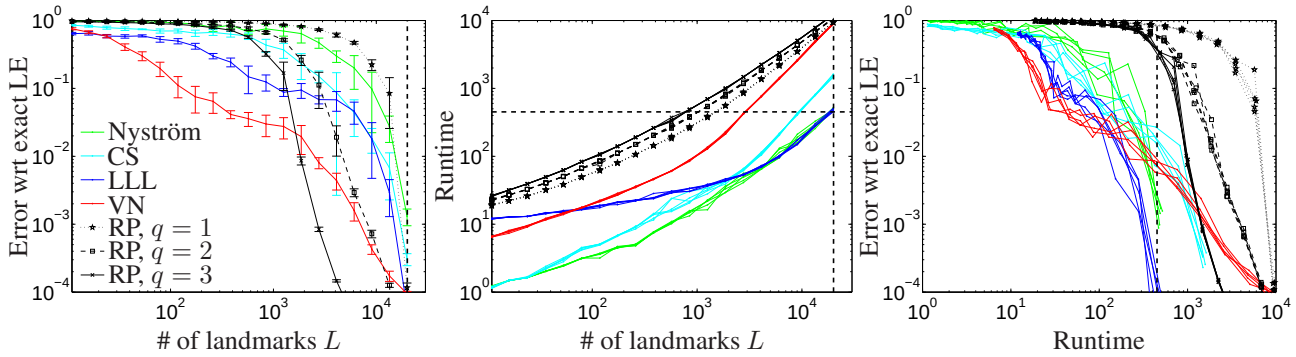

Figure 2. Normalized error for reconstructing the trailing 10 eigenvectors of the normalized graph Laplacian for 20 000 MNIST points. Nyström, CS and VN use the best choice of normalization for $\mathbf{Z}$ as discussed in section 3. *From left to right:* normalized error for a given number of landmarks, runtime for a given number of landmarks and error decrease per second of runtime. The dashed line shows when the number of landmarks is equal to N or when the runtime is equal to the runtime of the exact LE.

Comparison between methods We compare all the methods using their best normalization (Nyström: `CA`, CS: `CC`, VN: `Sum`). We use 20 000 random digits from MNIST and reduce dimensionality to $d = 10$ using exact LE and each of the methods. We construct the affinity matrix using entropic affinities (Hinton & Roweis, 2003; Vladymyrov & Carreira-Perpiñán, 2013a) with perplexity $K = 30$ (i.e., each datapoint $\mathbf{y}_i$ has its own Gaussian bandwidth σ_i, taken such that the effective number of neighbors covered by the kernel at $\mathbf{y}_i$ is K). We also sparsify the affinity matrix by zeroing all but the 200 largest values in each row. For the number of landmarks we use 20 log-spaced values from 11 to 19 900 and run each experiment 5 times, each with a different random choice of landmarks. Fig. 2 shows the error and runtime for different numbers of landmarks L. Each plot graphs two of *(number of landmarks, error, runtime)*.

Fig. 2 (left) shows the error vs L. The methods can be ranked from best to worst as VN $<$ LLL $<$ CS $<$ Nyström. We can see that VN and LLL show fast error decrease right from the beginning. Other methods do not show any good results until $L \gtrsim 400$, which is 2% of all the points. At this level VN already shows quite a low error, around 10^{-2}, and its variability over the random selection of landmarks is small. In pilot runs initialized the landmarks with k-means and observed that while the error did not decrease much the runtime increased a lot. It may be possible to improve the results with a more clever, efficient landmark selection, but we think random selection generally works well. RP has a good error decrease, especially for $q = 3$, but, again, this decrease does not happen until L is large.

Fig. 2 (middle) shows the runtime vs L. Nyström and CS take into account only a subset of the affinity matrix and thus are the fastest. LLL needs some time in the beginning to compute a reconstruction matrix $\mathbf{Z}$. RP methods involve an expensive projection and QR decomposition. VN clearly has two regimes: one for $L \leq 10^3$ (smaller slope) and another for $L > 10^3$ (steeper slope). This is because the affinity matrix becomes denser with more landmarks and thus the runtime changes from being dominated by the matrix product to being dominated by the reduced eigenproblem. This effect also appears in LLL, but for a larger value of L since the sparsity pattern is different (the out-of-sample matrix $\mathbf{Z}$ is much sparser for LLL than for VN).

Fig. 2 (right) combines the left and middle plots into a speed/accuracy tradeoff, or error decrease per runtime, where L grows along each curve. The region of best performance is the bottom-left corner: faster, more accurate results with fewer landmarks. VN has the best results when the error is $\leq 10^{-2}$, i.e., when we want a fast, low-to-medium accuracy solution (which is the most practical regime if approximating a large manifold learning problem). VN stops being competitive if one seeks a high-accuracy solution, because the original affinity matrix is quite dense and the reduced eigenproblem becomes denser and more expensive as L increases. LLL needs more time to build its reconstruction matrix $\mathbf{Z}$, but later catches up since the approximation matrix $\widetilde{\mathbf{M}}$ is more sparse. It is the fastest method if one seeks a high-accuracy solution. Other methods are never in the winning zone. Nyström and CS are fast, but their approximation is quite bad. RP gives a good approximation, but is slow to run.

We also ran an image segmentation experiment using spectral clustering (SC) on an image from the BSDS500 dataset (Arbeláez et al., 2011). Exact SC took 8.5 minutes. Then, we ran Nyström and VN, limiting their runtime to 25 s (a $20\times$ speedup). Fig. 3 clearly shows that VN is much closer to the exact result than the Nyström method.

Large-scale experiment We repeat an experiment from the original LLL paper (Vladymyrov & Carreira-Perpiñán, 2013b). We used 1 020 000 points from the infiniteMNIST dataset (Loosli et al., 2007), where we created 16 distortions of each MNIST digit by an elastic transformation (overall 60 000 original + 960 000 distorted digits). This

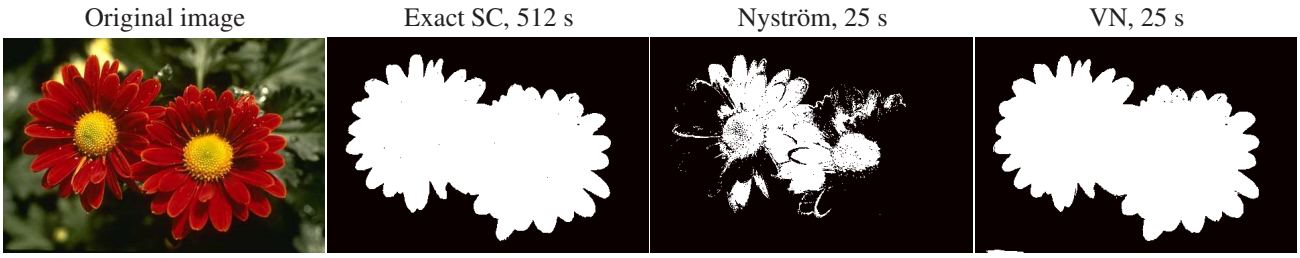

Figure 3. Figure/ground segmentation using exact spectral clustering (SC), and Nyström and VN (limited to 25 seconds' runtime).

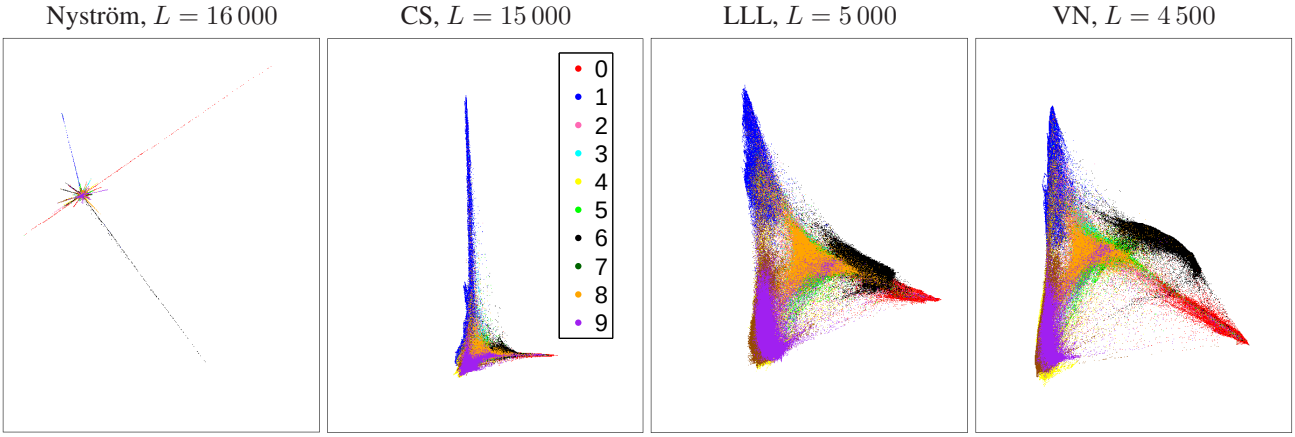

Figure 4. The Laplacian eigenmaps embedding of an infiniteMNIST dataset with $N = 1\,020\,000$ points approximated using Nyström, Column Sampling, LLL and Variational Nyström. The runtime is restricted to 10 minutes.

dataset is convenient for large-scale evaluations because it allows us to increase the sample size as much as desired while still being able to interpret visually the result and compare with the MNIST ground truth as well as with the exact run of LE on the original dataset (see fig. 3 in Vladymyrov & Carreira-Perpiñán 2013b). For each digit we define its nearest neighbors to be a set of 10 neighbors of the original (non-distorted) digit together with their distortions. We then used entropic affinities with perplexity $K = 10$. Our final affinities contain ≈ 150M nonzero elements with 0.01% sparsity level. For such a large affinity matrix, running exact LE is challenging, so we ran only the approximate methods. We set the runtime to 10 minutes and ran each method with the largest number of randomly selected landmarks possible. Fig. 4 shows the results. Nyström clearly gives the worst results even though it uses the most landmarks. CS uses a similar number of landmarks but performs much better. Both LLL and VN give the best embedding while using a much smaller number of landmarks. The VN embedding is the better one: notice the 4s are much more separated from the 9s and there is a more defined gap between the clusters of 0s, 5s and 6s.

5. Conclusion

In hindsight, Variational Nyström seems the right thing to do: if we are going to extrapolate to new data using the Nyström out-of-sample formula, we should incorporate that in the optimization. This results in an eigenproblem over the landmarks that better represents the manifold structure of the data and gives, by construction, a better solution for the same number of landmarks. Variational Nyström remains very easy to implement, with no user parameters other than the number of landmarks, but it does have the extra cost over Nyström's method of setting up and solving a less sparse reduced eigenproblem. Our experiments with manifold learning and spectral clustering show that Variational Nyström does not always beat Nyström or other landmark-based algorithms in an error-runtime tradeoff, but it does for the region of practical interest with large-scale data: achieving a faster solution of low-to-medium accuracy. With 1M points, Variational Nyström provides a good embedding in under 10 min runtime.

We also analyzed the use of subsampling approximations for a graph Laplacian data-dependent kernel. Directly applying those approximations to the graph Laplacian gives poor results. We provided a case-by-case analysis for every approximation method compared in this paper and showed that Variational Nyström has the simplest and most general form of normalization among all them.

Acknowledgements

Work partially supported by NSF award IIS–1423515. Part of this work was performed while the first author was a PhD student at UC Merced.

References

Arbeláez, Pablo, Maire, Michael, and Charless Fowlkes and, Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(5): 898–916, May 2011.

Atkinson, Kendall E. *The Numerical Solution of Integral Equations of the Second Kind*. Number 4 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1997.

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

Belongie, Serge, Fowlkes, Charless, Chung, Fan, and Malik, Jitendra. Spectral partitioning with indefinite kernels using the Nyström extension. In Heyden, A., Sparr, G., Nielsen, M., and Johansen, P. (eds.), *Proc. 7th European Conf. Computer Vision (ECCV'02)*, volume 2, pp. 21–31, Copenhagen, Denmark, May 28–31 2002.

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, October 2004.

Boutsidis, Christos, Drineas, Petros, and Magdon-Ismail, Malik. Near optimal column-based matrix reconstruction. In *Proc. of the 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pp. 305–314, Palm Springs, CA, October 22–25 2011.

Carreira-Perpiñán, Miguel Á. and Lu, Zhengdong. The Laplacian Eigenmaps Latent Variable Model. In Meilă, Marina and Shen, Xiaotong (eds.), *Proc. of the 11th Int. Conf. Artificial Intelligence and Statistics (AISTATS 2007)*, pp. 59–66, San Juan, Puerto Rico, March 21–24 2007.

Cortes, Corinna, Mohri, Mehryar, and Talwalkar, Ameet. On the impact of kernel approximation on learning accuracy. In Teh, Yee Whye and Titterington, Mike (eds.), *Proc. of the 13th Int. Conf. Artificial Intelligence and Statistics (AISTATS 2010)*, pp. 113–120, Chia Laguna, Sardinia, Italy, March 21–24 2010.

Cour, Timothée, Bénézit, Florence, and Shi, Jianbo. Spectral segmentation with multiscale graph decomposition. In *Proc. of the 2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05)*, pp. 1124–1131, San Diego, CA, June 20–25 2005.

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. Unpublished technical report, June 30 2004.

Fowlkes, Charless, Belongie, Serge, and Malik, Jitendra. Efficient spatiotemporal grouping using the Nyström method. In *Proc. of the 2001 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'01)*, pp. 231–238, Kauai, Hawaii, December 9–14 2001.

Fowlkes, Charless, Belongie, Serge, Chung, Fan, and Malik, Jitendra. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):214–225, February 2004.

Frieze, Alan, Kannan, Ravi, and Vempala, Santosh. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proc. of the 39th Annual Symposium on Foundations of Computer Science (FOCS 1998)*, Palo Alto, CA, November 8–11 1998.

Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

Hinton, Geoffrey and Roweis, Sam T. Stochastic neighbor embedding. In Becker, Suzanna, Thrun, Sebastian, and Obermayer, Klaus (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pp. 857–864. MIT Press, Cambridge, MA, 2003.

Li, M., Kwok, J. T., and Lu, B. Making large-scale Nyström approximation possible. In Fürnkranz, Johannes and Joachims, Thorsten (eds.), *Proc. of the 27th Int. Conf. Machine Learning (ICML 2010)*, pp. 631–638, Haifa, Israel, June 21–25 2010.

Loosli, Gaëlle, Canu, Stéphane, and Bottou, Léon. Training invariant support vector machines using selective sampling. In Bottou, Léon, Chapelle, Olivier, DeCoste, Dennis, and Weston, Jason (eds.), *Large Scale Kernel Machines*, Neural Information Processing Series, pp. 301–320. MIT Press, 2007.

Mahoney, M. W. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3 (2), 2011.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In Dietterich, Thomas G., Becker, Suzanna, and Ghahramani, Zoubin (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pp. 849–856. MIT Press, Cambridge, MA, 2002.

Platt, John C. Fast embedding of sparse similarity graphs. In Thrun, Sebastian, Saul, Lawrence K., and Schölkopf, Bernhard (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 16. MIT Press, Cambridge, MA, 2004.

Roweis, Sam T. and Saul, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 22 2000.

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

Shi, Jianbo and Malik, Jitendra. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

Talwalkar, Ameet, Kumar, Sanjiv, Mohri, Mehryar, and Rowley, Henry. Large-scale SVD and manifold learning. *J. Machine Learning Research*, 14(1):3129–3152, 2013.

Tenenbaum, Joshua B., de Silva, Vin, and Langford, John C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 22 2000.

Vladymyrov, Max and Carreira-Perpiñán, Miguel Á. Entropic affinities: Properties and efficient numerical computation. In Dasgupta, Sanjoy and McAllester, David (eds.), *Proc. of the 30th Int. Conf. Machine Learning (ICML 2013)*, pp. 477–485, Atlanta, GA, June 16–21 2013a.

Vladymyrov, Max and Carreira-Perpiñán, Miguel Á. Locally Linear Landmarks for large-scale manifold learning. In Blockeel, Hendrik, Kersting, Kristian, Nijssen, Siegfried, and Zelezný, Filip (eds.), *Proc. of the 24th European Conf. Machine Learning (ECML–13)*, pp. 256–271, Prague, Czech Republic, September 23–27 2013b.

Wang, Shusen and Zhang, Zhihua. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *J. Machine Learning Research*, 14: 2729–2769, September 2013.

Williams, Christopher K. I. and Seeger, Matthias. Using the Nyström method to speed up kernel machines. In Leen, Todd K., Dietterich, Tom G., and Tresp, Volker (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pp. 682–688. MIT Press, Cambridge, MA, 2001.

Yang, Tianbao, Li, Yu-Feng, Mahdavi, Mehrdad, Jin, Rong, and Zhou, Zhi-Hua. Nyström method vs random Fourier features: A theoretical and empirical comparison. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pp. 485–493. MIT Press, Cambridge, MA, 2012.

Zhang, Kai and Kwok, James T. Clustered Nyström method for large scale manifold learning and dimension reduction. *IEEE Trans. Neural Networks*, 21(10):1576–1587, October 2010.

Zhang, Kai, Tsang, Ivor W., and Kwok, James T. Improved Nyström low-rank approximation and error analysis. In McCallum, Andrew and Roweis, Sam (eds.), *Proc. of the 25th Int. Conf. Machine Learning (ICML'08)*, pp. 1232–1239, Helsinki, Finland, July 5–9 2008.