On the Relation of Slow Feature Analysis and Laplacian Eigenmaps

Henning Sprekeler
henning.sprekeler@epfl.ch
Laboratory for Computational Neuroscience, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

The past decade has seen a rise of interest in Laplacian eigenmaps (LEMs) for nonlinear dimensionality reduction. LEMs have been used in spectral clustering, in semisupervised learning, and for providing efficient state representations for reinforcement learning. Here, we show that LEMs are closely related to slow feature analysis (SFA), a biologically inspired, unsupervised learning algorithm originally designed for learning invariant visual representations. We show that SFA can be interpreted as a function approximation of LEMs, where the topological neighborhoods required for LEMs are implicitly defined by the temporal structure of the data. Based on this relation, we propose a generalization of SFA to arbitrary neighborhood relations and demonstrate its applicability for spectral clustering. Finally, we review previous work with the goal of providing a unifying view on SFA and LEMs.

1 Introduction

Many algorithms in machine and reinforcement learning suffer from the curse of dimensionality. A common approach to this problem is to apply dimensionality-reduction techniques that reshape the data into a lower-dimensional and more convenient format. Linear dimensionality-reduction techniques are computationally very efficient, but their use is limited in cases where the data reside on curved manifolds embedded in a high-dimensional input space. In this case, nonlinear dimensionality-reduction techniques, although computationally more challenging, can allow a more efficient compression of the data.

An important objective in nonlinear dimensionality reduction, which is trivially fulfilled for linear approaches, is to conserve neighborhood relations in the original data. A technique that allows rich nonlinear mappings while preserving these relations are Laplacian eigenmaps (LEMs; Belkin

The author is now at the Institute for Theoretical Biology, Humboldt-Universität zu Berlin, Germany.

& Niyogi, 2003), also known as diffusion maps. LEMs find an embedding of the data in a low-dimensional space by constructing a graph on the data and finding the eigenvectors of the associated graph Laplacian. They have found applications in several fields, including clustering (Ng, Jordan, & Weiss, 2002; Shi & Malik, 2002; for a review, see von Luxburg, 2007) and semi-supervised learning (Belkin & Niyogi, 2004; Zhou, Bousquet, Lal, Weston, & Schölkopf, 2004), and they have recently been proposed for finding state representations in reinforcement learning (Mahadevan & Maggioni, 2007).

One limitation of LEMs is that the dimensionality of the graph Laplacian scales with the number of data points. Therefore, the associated eigenvalue problem can become intractable for large data sets. This problem can be overcome by a reduction of the effective number of data points (e.g., by constructing a reduced backbone graph or by Nyquist sampling) or by calculating the eigenmodes in marginalized spaces by exploiting the relation of the graph Laplacian to the Laplace operator (Fergus, Weiss, & Torralba, 2009).

Here, we show that for temporally structured data, the optimization problem of the LEM method is formally equivalent to that of slow feature analysis (SFA), a nonlinear signal processing algorithm that aims at minimizing temporal variations in the output signals. The neighborhood function in the associated LEM problem is given by transition probabilities and is therefore implicitly defined by the temporal structure of the data. We then show that the SFA algorithm is a function approximation for the full LEM problem. Based on this equivalence, we suggest a generalization of SFA from temporal to arbitrary neighborhood relations and show in a proof-of-concept simulation that it can provide a good approximation of LEMs while offering a considerable reduction in computational complexity. Finally, we provide a selective review that puts previous work on SFA in relation to the LEM literature.

2 Laplacian Eigenmaps

LEMs are a J-dimensional representation $\mathbf{y}^t \in \mathbb{R}^J$ of a set of T data points $\mathbf{x}^t \in \mathbb{R}^N$. Because we focus on data in the time domain, we use the index t to enumerate the data points. The algorithm used to calculate the LEM representation consists of two steps. First, an undirected graph is constructed with the data points $\mathbf{x}^t$ as nodes and an adjacency matrix $W_{tt'}$ that assigns weights to the edges between the nodes. The adjacency matrix $W_{tt'}$ is usually determined by either a heat kernel $W_{tt'} \sim \exp(-|\mathbf{x}^t - \mathbf{x}^{t'}|^2/(2\sigma^2))$ or a binary representation of the graph structure, that is, $W_{tt'} = 1$, if there is an edge between $\mathbf{x}^t$ and $\mathbf{x}^{t'}$ and $W_{tt'} = 0$ otherwise.

Second, given the adjacency matrix, the representation $\mathbf{y}^t$ is determined using a sequential optimization of the component vectors $\mathbf{y}_j = (y_j^1, \dots, y_j^T)$.

The optimization aims at minimizing the cost function

$$\Psi(\mathbf{y}_j) = \sum_{t,t'} W_{tt'}(y_j^t - y_j^{t'})^2, \tag{2.1}$$

under the constraint that

$$\sum_{t,t'} D_{tt'}\, y_i^t y_j^{t'} = \delta_{ij} \quad \text{for } i \leq j\,. \tag{2.2}$$

Here, $D_{tt'} = \delta_{tt'} \sum_s W_{ts}$ is a diagonal matrix that contains the degree of the data points $\mathbf{x}^t$ on the diagonal, and δ_{ij} denotes the Kronecker symbol. The optimization is sequential in the sense that the first component, $\mathbf{y}_1$, is optimized first (the only constraint being the normalization induced by equation 2.2 for $i = j = 1$); then $\mathbf{y}_2$ is optimized (with the orthogonality constraint 2.2 with respect to the first vector $\mathbf{y}_1$), then $\mathbf{y}_3$, and so on. Note that LEMs assign a representation $\mathbf{y}^t$ to each data point $\mathbf{x}^t$ individually; that is, they do not provide a smooth functional mapping between the input data and the representation.

The optimization problem can be reduced to solving a generalized eigenvalue problem for the components $\mathbf{y}_j$ (Shi & Malik, 2002):

$$L\mathbf{y}_j = \lambda_j D\mathbf{y}_j. \tag{2.3}$$

Here, $L = D - W$ is the graph Laplacian. The eigenvalue problem corresponds to calculating the right eigenvectors of the so-called normalized graph Laplacian $D^{-1}L$. There are alternative approaches, which either do not normalize the Laplacian (this corresponds to a similar constraint as in equation 2.2, but with the unit matrix $D_{tt'} = \delta_{tt'}$) or apply a symmetric normalization ($L \to D^{-1/2}LD^{-1/2}$). Note that the dimension of the graph Laplacian is given by the number of data points. For large data sets, the eigenvalue problem 2.3 can become intractable and requires approximate methods that reduce the dimensionality of the problem.

The cost function Ψ aims at preserving neighborhood relations by punishing large differences $y_j^t - y_j^{t'}$ for neighboring points, that is, for points with large edge weight $W_{tt'}$. For temporally structured data, where subsequent data points are often neighbors, this has the effect that the embedding $\mathbf{y}^t$, when treated as a time series, varies smoothly in time.

3 Slow Feature Analysis

Whereas embeddings $\mathbf{y}^t$ that vary smoothly in time are a natural consequence of the smoothness objective in LEMs, they are the explicit goal of SFA (Wiskott & Sejnowski, 2002). SFA aims at minimizing temporal variations in a set of output signals $y_j^t = g_j(\mathbf{x}^t)$ generated from a given time-dependent, vectorial input signal $\mathbf{x}^t$. The optimization is performed on the functions g_j, which are constrained to lie within in a given function space $\mathcal{F}$.

How quickly a signal $\mathbf{y}_j$ varies in time is quantified by the Δ-value, which is defined as the mean square of the temporal derivative $\Delta(\mathbf{y}_j) := \langle(\dot{y}_j^t)^2\rangle_t$. Because the data are typically sampled in discretized time, the derivative is often replaced by the difference $\dot{y}_j^t \sim y_j^{t+1} - y_j^t$. To avoid the trivial constant solution and degeneracies arising from possible additions of arbitrary constants, the output signals are constrained to have zero mean and unit variance. Just as for LEMs, the optimization is performed sequentially with an asymmetric decorrelation constraint: g_1 is optimized first, yielding the slowest possible signal $\mathbf{y}_1$. Next, g_2 is optimized under the constraint that its output signal $\mathbf{y}_2$ is decorrelated from $\mathbf{y}_1$, $\mathbf{y}_3$ has to be decorrelated from $\mathbf{y}_1$ and $\mathbf{y}_2$, and so on. Iterating this scheme yields a set of functions g_j ordered by slowness, that is, by their Δ-value.

Mathematically, this optimization problem can be formulated as follows:

Sequential optimization problem. *Given a function space $\mathcal{F}$ and an N-dimensional, time-dependent input signal $\mathbf{x}^t$, find a set of J real-valued input-output functions $g_j(\mathbf{x})$ such that the output signals $y_j^t := g_j(\mathbf{x}^t)$ minimize*

$$\Delta(\mathbf{y}_j) = \left\langle \left(\dot{y}_j^t\right)^2 \right\rangle_t \tag{3.1}$$

under the constraints

$$\left\langle y_j^t \right\rangle_t = 0 \quad (\textit{zero mean}), \tag{3.2}$$

$$\left\langle \left(y_j^t\right)^2 \right\rangle_t = 1 \quad (\textit{unit variance}), \tag{3.3}$$

$$\forall i < j : \left\langle y_i^t y_j^t \right\rangle_t = 0 \quad (\textit{decorrelation and order}), \tag{3.4}$$

with $\langle\cdot\rangle_t$ and $\dot{y}$ indicating temporal averaging and the derivative of y, respectively.

The optimal functions can be found in a computationally efficient way. Usually the function space $\mathcal{F}$ is defined by choosing a set of basis functions f_α that span $\mathcal{F}$. All possible output signals can then be generated by a linear superposition of the output signals $z_\alpha^t = f_\alpha(\mathbf{x}^t)$ of the basis functions f_α:

$$y_j^t = \sum_\alpha V_{j\alpha} z_\alpha^t = \mathbf{V}_j \cdot \mathbf{z}^t\,. \tag{3.5}$$

The zero mean constraint is enforced by subtracting suitable constants from the basis functions such that the mean of their output signals vanishes: $\langle \mathbf{z}\rangle_t = 0$. Alternatively, one can drop the zero mean constraint and include the constant function in the function space $\mathcal{F}$. The slowest output signal is then always the constant, and all other functions must have zero mean to meet the decorrelation constraint with the constant. Finding the slowest possible output signals $\mathbf{y}_j$ amounts to finding the optimal coefficient matrix $\mathbf{V}$.

The objective function and the constraints are quadratic in the coefficient matrix $\mathbf{V}$, as they are for LEMs, so that the solution of the optimization problem can be reduced to a generalized eigenvalue problem (Berkes & Wiskott, 2005):

$$\dot{\mathbf{C}}\mathbf{V} = \mathbf{C}\mathbf{V}\mathbf{\Lambda}\,, \tag{3.6}$$

where $\mathbf{C} := \langle \mathbf{z}\mathbf{z}^T \rangle_t$ and $\dot{\mathbf{C}} = \langle \dot{\mathbf{z}}\dot{\mathbf{z}}^T \rangle_t$ are the covariance matrices of signals $\mathbf{z}$ and their temporal derivatives, respectively. $\mathbf{\Lambda}$ is a diagonal matrix that contains the eigenvalues λ_j on the diagonal. The optimal functions are given by $g_j = \sum_\alpha V_{j\alpha} f_\alpha$, and the associated eigenvalue λ_j is the Δ-value of the function g_j. Note that in contrast to LEMs, the dimension of the covariance matrices scales with the dimension of the function space rather than the number of data points.

4 SFA and Laplacian Eigenmaps

4.1 Equivalence of SFA and LEMs. We now show that in the limit of large data sets with temporal structure, the objective functions of SFA and LEMs are equivalent when the neighborhood function for the LEM is determined by the transition probabilities of the data.

Let us assume that the data are elements of a set $\mathcal{S}$, which could be either a finite set or a manifold. We represent the distribution of the data on $\mathcal{S}$ and their temporal dynamics by the joint probability $p(\mathbf{x}, t; \mathbf{x}', t+1)$ of two subsequent data points. To ensure that temporal averages are equivalent to ensemble averages, we assume that the dynamics are stationary, that is, $p(\mathbf{x}, t; \mathbf{x}', t+1) = p(\mathbf{x}, \mathbf{x}')$ is independent of t. If we replace the temporal derivative by the difference between subsequent data points, as usually done in SFA, and the temporal average in the objective function 3.1 of SFA by a weighted average over the points in $\mathcal{S}$, we get the objective function in the limit of infinitely many data points,

$$\Delta(g) = \sum_{\mathbf{x}, \bar{}\mathbf{x} \in \mathcal{S}} p(\mathbf{x}, \mathbf{x}')(g(\mathbf{x}) - g(\mathbf{x}'))^2, \tag{4.1}$$

where $\Delta(g)$ denotes the Δ-value of the signal $g(\mathbf{x}^t)$ that arises from applying the function g to the input signals. If $\mathcal{S}$ is a manifold, the sum is replaced by an integration.

For LEMs, we assume that the data points $\mathbf{x}^t$ are drawn from the marginal distribution $p(\mathbf{x}) = \sum_{\mathbf{x}'} p(\mathbf{x}, \mathbf{x}')$ on $\mathcal{S}$. Moreover, we can interpret the mapping from the data points $\mathbf{x}^t$ to the embedding $\mathbf{y}^t$ as a function $\mathbf{y}^t = g(\mathbf{x}^t)$. Again, taking the limit of many samples, the sample average in the objective function Ψ for the LEMs can also be replaced by a weighted average on $\mathcal{S}$, yielding:

$$\Psi(g) = \sum_{\mathbf{x},\mathbf{x} \in \mathcal{S}} p(\mathbf{x})p(\mathbf{x}')W(\mathbf{x},\mathbf{x}')(g(\mathbf{x}) - g(\mathbf{x}'))^2. \tag{4.2}$$

Here, we assumed that the edge weights $W_{tt'}$ are determined by a neighborhood function $W(\mathbf{x},\mathbf{x}')$. The two objective functions have the same mathematical structure and become identical if the neighborhood function $W(\mathbf{x},\mathbf{x}')$ that measures the topological relations for the LEMs is chosen according to the joint probability distribution for subsequent data points:

$$W(\mathbf{x},\mathbf{x}') = \frac{p(\mathbf{x},\mathbf{x}')}{p(\mathbf{x})p(\mathbf{x}')} = \frac{p(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}')}\,. \tag{4.3}$$

This adjacency function assigns a large weight if the probability of visiting $\mathbf{x}'$ at time $t+1$ given $\mathbf{x}$ at time t is large compared to the marginal probability of visiting $\mathbf{x}'$, independent of the last point. SFA can therefore be thought of as an LEM problem in which neighborhood relationships are implicitly defined by the temporal structure of the data.

Although the objective functions become identical for a specific choice of the adjacency function, the optimization problems are different. In LEMs, the representation $\mathbf{y}^t$ can be chosen individually for each data point $\mathbf{x}^t$. In SFA, in contrast, the functions g_j that map the input data $\mathbf{x}^t$ to the embedding $\mathbf{y}^t$ are chosen from a given function space $\mathcal{F}$. Consequently, the optimization problems for LEMs and for SFA are equivalent only if the function space $\mathcal{F}$ is sufficiently rich to allow arbitrary mappings.

For poorer function spaces, SFA becomes a function approximation to the full LEM problem. The quality of the approximation depends on the character of the function space $\mathcal{F}$.

4.2 Generalized SFA. Due to the close relation between SFA and LEMs, it is straightforward to devise an algorithm that captures the key properties of the two approaches: arbitrary neighborhood relations as in LEMs and the computational tractability of SFA. To this end, let us assume that some neighborhood function has given rise to a graph Laplacian L. To get a function approximation in the spirit of SFA, we approximate the eigenvectors $\mathbf{y}_j$ of the Laplacian by a linear superposition of basis functions f_α:

$$y_j^t = \sum_\alpha V_{j\alpha} f_\alpha(\mathbf{x}^t). \tag{4.4}$$

With this ansatz, the objective function 2.1, and the constraint 2.2, for LEMs can be rewritten in terms of the coefficients $V_{j\alpha}$:

$$\tilde{\Psi}(V_{j\alpha}) = \sum_{\alpha,\beta} V_{j\alpha}\tilde{L}_{\alpha\beta}V_{j\beta}, \tag{4.5}$$

$$\sum_{\alpha,\beta} V_{i\alpha}\tilde{D}_{\alpha\beta}V_{j\beta} = \delta_{ij} \quad \text{for } i < j\,. \tag{4.6}$$

Here $\tilde{L}$ and $\tilde{D}$ denote the reduced Laplacian and degree matrices:

$$\tilde{L}_{\alpha\beta} := \sum_{t,t'} f_\alpha(\mathbf{x}_t)L_{tt'}f_\beta(\mathbf{x}^{t'}) = \sum_{t,t'} W_{tt'}(f_\alpha(\mathbf{x}^t) - f_\beta(\mathbf{x}^{t'}))^2, \tag{4.7}$$

$$\tilde{D}_{\alpha\beta} := \sum_{t,t'} f_\alpha(\mathbf{x}^t)D_{tt'}f_\beta(\mathbf{x}^{t'})\,. \tag{4.8}$$

The eigenvalue equation associated with this optimization problem has the same structure as that for the original SFA problem, equation 2.1, just with the matrices $\tilde{L}$ and $\tilde{D}$ instead of the covariance matrices $\mathbf{C}$ and $\dot{\mathbf{C}}$:

$$\tilde{L}\mathbf{y}_j = \lambda_j\tilde{D}\mathbf{y}_j\,. \tag{4.9}$$

Note that the original optimization problem of SFA can be recovered by using the neighborhood function $W_{tt'} = (\delta_{t,t'+1} + \delta_{t+1,t'})$. In this case, the Laplacian $L_{tt'} = 2\delta_{tt'} - (\delta_{t,t'+1} + \delta_{t+1,t'})$ is simply a discretized version of the second temporal derivative.

This algorithm is a generalization of SFA in the sense that it replaces temporal with arbitrary neighborhoods while maintaining the algorithmic elements of SFA. Being a hybrid algorithm, however, it could with equal right be referred to as a function approximation of LEMs.

One advantage of generalized SFA over LEMs is that the mapping from the input signals to the embedding is explicitly provided as a function. Therefore, the solution can be applied to new data without the need for an interpolation between known data points.

4.3 Hierarchical Function Approximations. In SFA, richer function spaces are often generated by a hierarchical iteration of simpler functions (Wiskott & Sejnowski, 2002; Franzius, Sprekeler, & Wiskott, 2007; Franzius, Wilbert, & Wiskott, 2008). This approach reduces the computational complexity for high-dimensional input signals and tends to avoid overfitting problems (Wiskott & Sejnowski, 2002). A hierarchical iteration for the generalized SFA algorithm introduced in the previous section is straightforward and takes the form shown in algorithm 1.

Algorithm 1: Hierarchical Generalized SFA

```
Step 1. Perform a nonlinear expansion $z^t_\alpha = f_\alpha(\mathbf{x}^t)$ of the input signals in a given
        function space $\mathcal{F}$.
Step 2. Calculate the reduced Laplacian and degree matrices $\tilde{L}$ and $\tilde{D}$ by projecting the
        full Laplacian (which is the same on all levels) onto the expanded signals:
        $\tilde{L}_{\alpha\beta} = \sum_{tt'} z^t_\alpha L_{tt'} z^{t'}_\beta$.
Step 3. Solve the generalized eigenvalue problem 4.9 for the reduced matrices.
Step 4. Calculate the output signals $\mathbf{y}^t = V\mathbf{z}^t$, keeping only the $n$ signals that correspond
        to the $n$ lowest eigenvalues.
Step 5. If the hierarchy has additional levels, use the output signals $\mathbf{y}^t$ as input signals
        for the next level and return to step 1. Otherwise return the output signals as a
        terminal output signal of the hierarchy.
```

Of course, the function space and the number of output signals that are passed on to the next level of the hierarchy can be chosen individually for each layer when suitable.

Because of the similarity of SFA and LEMs, one would expect that such an approach can gradually approach the solution of the full LEM problem. In the following, we provide anecdotal evidence that this is indeed the case by suggesting a novel, hierarchical approach to spectral clustering.

Example: Hierarchical spectral clustering. Spectral clustering approaches rely on the observation that if the adjacency matrix W contains no edges between different clusters, the first eigenvectors of the associated Laplacian are constant within the clusters and maintain their variance by intercluster differences. Thus, the LEM representation tends to separate clusters and thereby simplifies subsequent clustering by standard techniques (von Luxburg, 2007).

We tested the idea of a hierarchical approximation of the full LEM problem by hierarchically applying generalized SFA to a simple clustering task on two intertwined semilunar data clouds (see Figure 1). Each level of the hierarchy performs generalized SFA for the function space of all polynomials of degree 3. The adjacency matrix was calculated using an isotropic gaussian neighborhood function ($\sigma = 0.05$). It is reused on all levels. On each level, only the four output signals corresponding to the smallest eigenvectors are passed on to the next level. Figure 1 shows the dependency of the first output signal on the depth of the hierarchy for a test data set. As the depth increases, the output signal approaches a constant value within each cluster, with different values for the two clusters. Clearly clustering becomes simple on such a representation of the data.

The algorithm requires the solution of several eigenvalue problems with relatively low-dimensional matrices (here: 4 signals + polynomial

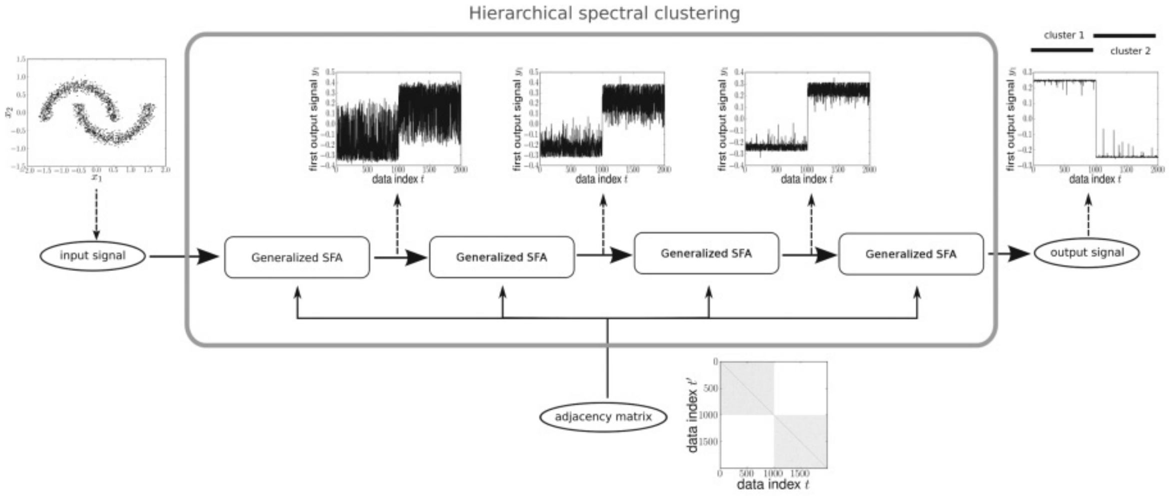

Figure 1: Hierarchical spectral clustering. On each level of the hierarchy, a nonlinear expansion of the output signals from the previous level is used to calculate the reduced Laplacian and degree matrices $\tilde{L}$ and $\tilde{D}$. The same adjacency matrix is used on all layers. Next, the associated generalized eigenvalue problem, equation 4.9, is solved. The eigenvectors with the lowest eigenvalues, scalarly multiplied with the expanded input signals, form the output signals of the current level and provide the input for the next. We tested the performance of the algorithm in a clustering task on two intertwined, semilunar data clouds (left). The half-moon clusters are generated according to $(x, y) = (A - 1/2 + (1 + \xi)\cos(\eta), (-1)^A(1 + \xi)\sin(\eta) - (-1)^A/4)$, where $A \in \{0, 1\}$ enumerates the half-moons, ξ is gaussian noise with standard deviation of 0.1, and η is uniformly distributed on $[0, \pi]$. We used polynomial expansions of degree 3 on all levels, followed by a singular value decomposition to discard singular dimensions (those with a variance that is at least 10^{13} times smaller than the variance of the dominant dimension). The adjacency matrix was given by an isotropic gaussian neighborhood function with a width of $\sigma = 0.05$. Data points were ordered according to cluster affiliation. The output signals of the levels gradually approach a step function—that is, they approach a constant within the clusters. Thus, the separability of the clusters increases, so that subsequent clustering with standard techniques is greatly simplified.

expansion of degree 3 → 34 dimensions) instead of a single eigenvalue problem with the high dimensionality of the data points (in our case, 2000 dimensions) that would have to be solved for the full LEM solution. The number of data points enters only in the calculation of the reduced Laplacians, with a computational complexity that scales quadratically with the number of data points instead of the third power associated with eigenvalue problems. For sparse adjacency matrices and Laplacians, there are, of course, more efficient algorithms that solve the eigenvalue problem, but in this case, the calculation of the reduced Laplacians can also be simplified with sparse matrix multiplication techniques. Therefore, an approximation of LEMs by hierarchical iteration of generalized SFA promises considerable reductions in computational complexity.

Note that the highest layer effectively calculates a polynomial of degree $3^4 = 81$. The space of polynomials of degree 81 in two input dimensions is 3403-dimensional. Therefore, trying to find the solution in this space directly would be computationally more costly than the full LEM problem (matrix dimension 3403 instead of 2000) and highly prone to overfitting because the dimension of the function space is larger than the number of data points. The hierarchical approach evades both of these problems.

5 Earlier Work

In this section, we provide a selective review of previous work on SFA and LEMs, with the goal of a knowledge transfer between researchers working in these two domains.

5.1 Optimal Output Signals. As Wiskott (2003) showed, the optimal output signals for SFA are harmonic oscillations in time. This result was derived for the case where the dependence of the output on the input data is neglected, but the same harmonic oscillations are generated if the function space that SFA can access is sufficiently rich to allow independent values for the output signal for each data point in time. Because this is the case for LEMs, it is not surprising that a similar statement can be made for LEMs if adjacency is determined by temporal neighborhood.

Let us assume that the entries of the adjacency matrix depend on only the time difference between the data points: $W_{tt'} = w(t - t')$, where w is an arbitrary positive function. Clearly the entries of the Laplacian $L = D - W$ then also depend on the time difference only. This class of matrices is known as Toeplitz matrices, and it has been shown that as the size of such a matrix increases, the eigenvectors of the matrix converge to harmonic oscillations (Böttcher, Grudsky, Maksimenko, & Unterberger, 2009), with eigenvalues that are given by the Fourier transform of the temporal neighborhood function w. Therefore, the eigenvectors for LEMs with temporal adjacency are also harmonic oscillations. As a consequence, the idea of temporal neighborhood is not straightforward to incorporate in LEMs because they would overfit to the trajectories.

5.2 Protovalue Functions and SFA for Reinforcement Learning. One of the central problems of reinforcement learning (RL) is that the speed of learning decreases quickly with the complexity of the environment, that is, with the number of states the agent can visit. A popular way out of this dilemma is to approximate the relevant state-dependent quantities (be it a value function or a policy) by a function approximator and learn merely the parameters of the approximator (Sutton & Barto, 1998). An unfortunate aspect of this approach is that the function approximation has to be chosen with care, because its properties critically influence the learning performance of the agent.

Recently both SFA and LEMs were suggested as a data-driven choice of such a function approximation. Mahadevan and Maggioni (2007) used an exploration phase of the agent to learn an LEM embedding—so-called proto-value functions (PVF)—of complicated state spaces, which was then used as a basis for the function approximation. Independently, Legenstein, Wilbert, and Wiskott (2010) trained a hierarchical SFA network on high-dimensional visual input that shows the moving agent. The state representations learned in this way were then used as function approximations for standard RL algorithms and showed substantial performance gains over standard function approximations. The demonstrated relation of SFA and LEMs shows that the approach of Legenstein et al. (2010) is closely related to the PVF approach of Mahadevan and Maggioni (2007). The advantage of SFA in this context is that no explicit knowledge about the state of the agent is required. Instead, the input signals can be arbitrary sensory data because, according to earlier theoretical results (Franzius et al., 2007), the state representation is largely independent of how the states are represented in the input. In constrast, LEMs are not easy to apply to sensory data directly, because the appropriate metrics in the sensor space is not always obvious (e.g., for visual input data).

5.3 Function Approximations. Generalized SFA provides a function approximation approach to LEMs that can lead to a significant reduction in computational complexity. Unfortunately, this comes at a conceptual cost: the freedom of choice for the function approximation. This choice can have a drastic impact on how well the real LEMs are approximated, yet there is no clear rationale which approximation scheme is suitable for the data at hand. For SFA, polynomial expansions are popular, although they often require a failsafe step that clips exceedingly large output signals that arise from outliers in unknown data (Franzius et al., 2007). A possible reason that polynomials perform relatively well is that for gaussian input distributions, the optimal functions are Hermite polynomials of the input (Sprekeler, Zito, & Wiskott, 2010).

Other function approximation schemes with localized basis functions (e.g., radial basis functions) may be more robust to outliers, but for high-dimensional input signals, they are problematic because a large number of basis functions is needed for a decent approximation of the delocalized functions that typically arise in both SFA and LEMs.

The strength of hierarchical function approximations lies in their tendency to overcome the curse of dimensionality and reduce the risk of overfitting. A detailed discussion of the advantages and drawbacks of hierarchical function approximations has been provided elsewhere (Wiskott & Sejnowski, 2002).

5.4 Supervised Learning. Both SFA and LEMs have been used as preprocessing for supervised classification. There are technical differences, though, which are worth highlighting.

5.4.1 SFA. Classification problems usually do not offer any temporal structure, while SFA has no place for a supervision signal. The trick that makes SFA suitable for supervised learning nevertheless is to smuggle the supervision signal into the input statistics by defining temporal sequences that dominantly contain transitions between stimuli of the same class (Berkes, 2005; Klampfl & Maass, 2010). The matrix of transition probabilities between the stimuli then has a block structure, with one block for each class. Consequently, the adjacency matrix and Laplacian of the associated LEM problem also have a block structure, so that the first eigenvectors are constant within the classes. Classification with SFA thus uses the same idea as spectral clustering, but with a hand-engineered adjacency matrix that reflects the supervision signal.

5.4.2 LEMs. The approach used for SFA, to hand-engineer the adjacency matrix to incorporate the supervision signal, is unsuitable for LEMs because it would lead to tremendous overfitting. Instead, LEMs are used to learn the topological structure of the input data in an unsupervised fashion, followed by standard classification algorithms. The advantage of this approach is that the output of the LEMs captures the manifold structure of the data, so that good generalization can be achieved even when the class affiliation is known only for a small fraction of the input data (semisupervised learning; Belkin & Niyogi, 2004; Zhou et al., 2004; Fergus et al., 2009).

5.5 Relation to the Laplace Operator. In the limit of a large data set from a smooth manifold M, both SFA and LEMs were shown to yield eigenfunctions of a Laplace-type differential operator on the input manifold. For LEMs, it was shown that in the large-sample limit, the graph Laplacian (with the normalization we consider here) converges to the Laplace-Beltrami operator on the manifold (Hein, Audibert, & von Luxburg, 2007; Belkin & Niyogi, 2008). For SFA, it was also shown that the optimal functions are the eigenfunctions of a generalized Laplace operator (Franzius et al., 2007) on the input manifold. (The appendix provides a moment expansion-based derivation that links SFA and LEMs to the Laplace operator.)

5.6 Manifolds of Statistically Independent Signals.

Based on the manifold results for SFA, it was recently shown that if the manifold can be parameterized by statistically independent signals, the output signals factorize into functions that depend on one of the independent signals each (Sprekeler et al., 2010). This result forms the basis of a recent extension of SFA for nonlinear blind source separation (Sprekeler et al., 2010). A similar factorization statement has been made for LEMs (Fergus et al., 2009), and an LEM-based approach to nonlinear blind source separation has also been presented (Singer & Coifman, 2008). Interestingly, Singer and Coifman observed that the factorization statement is true only when the neighborhood function is locally adapted to the data, so that the main axes of the gaussian neighborhood function align with the "local directions" of the independent components. SFA does this adaptation automatically: the neighborhood function corresponds to the conditional probability for neighboring data points, which factorizes for statistically independent signals, resulting in main axes that automatically align with the independent components. On the other hand, the approach of Singer and Coifman (2008) has the advantage that it can be applied to blind source separation problems that have no temporal dynamics.

6 Conclusion

We have shown that spectral techniques, that is, LEMs and diffusion maps, are closely related to SFA if the input data have a temporal structure. Based on this relation, we have presented a generalization of SFA that can be hierarchically iterated. In a simple application to spectral clustering, we have illustrated that hierarchical networks of generalized SFA can be seen as a function approximation for LEMs that provides significant reductions in computational complexity. Finally, we selectively reviewed previous work, showing that the two techniques have been applied in similar settings, and highlighting similarities and differences in their use. The relation between the two techniques provides a new perspective on SFA.

Appendix: Derivation of the Associated Generalized Laplace Operator

Both the optimal solutions of SFA and LEMs are eigenfunctions of a Laplace operator. One way of showing this is by a moment expansion of the respective objective functions for the limit case of infinitely many data points,

$$\Psi(g) = \iint_{M\times M} p(\mathbf{x})\pi(\mathbf{x},\mathbf{x}')(g(\mathbf{x}) - g(\mathbf{x}'))^2 \, \mathrm{d}x \, \mathrm{d}x' \,, \tag{A.1}$$

with $\pi(\mathbf{x},\mathbf{x}') = p(\mathbf{x}|\mathbf{x}')$ for SFA and $\pi(\mathbf{x},\mathbf{x}') = p(\mathbf{x}')W(\mathbf{x},\mathbf{x}')$ for LEMs.

We now make the assumption that the function $\pi(\mathbf{x},\mathbf{x}')$ has significant deviations from 0 only for $\mathbf{x} \approx \mathbf{x}'$. For SFA, this requires that subsequent data points are close to each other, so this is basically an assumption of continuity. For LEMs, it means that the neighborhood function that defines the adjacency matrix is sufficiently localized.

Because the integrals in the objective function are dominated by $\mathbf{x} \approx \mathbf{x}'$, we can then replace $g(\mathbf{x}')$ by a Taylor approximation to first order: $g(\mathbf{x}') \approx g(\mathbf{x}) + (\mathbf{x}' - \mathbf{x}) \cdot \nabla g(\mathbf{x})$. Inserting this into the objective function and carrying out the integration over $\mathbf{x}'$ yields

$$\Psi(g) = \int_M p(\mathbf{x})(\nabla g(\mathbf{x}))^T K(\mathbf{x}) \nabla g(\mathbf{x}) \, \mathrm{d}x, \tag{A.2}$$

where $K(\mathbf{x})$ is the matrix of the second moments of $\pi(\mathbf{x},\mathbf{x}')$:

$$K(\mathbf{x}) = \int \pi(\mathbf{x},\mathbf{x}')(\mathbf{x} - \mathbf{x}')(\mathbf{x} - \mathbf{x}')^T \, \mathrm{d}x' \,. \tag{A.3}$$

The constraints (see equations 2.2, 3.3, and 3.4) can also be written in integral form

$$\int_M \zeta(\mathbf{x}) g_i(\mathbf{x}) g_j(\mathbf{x}) \, \mathrm{d}x = \delta_{ij} \,, \tag{A.4}$$

with $\zeta(\mathbf{x}) = p(\mathbf{x})$ for SFA and $\zeta(\mathbf{x}) = p(\mathbf{x}) \int p(\mathbf{x}')W(\mathbf{x},\mathbf{x}') \, \mathrm{d}\mathbf{x}'$ for LEMs.

Since we are dealing with a constrained optimization problem, we can use the method of Lagrange multipliers and transform the problem into that of finding the stationary points of the Lagrange function,

$$\mathcal{L}(g_i) = \Psi(g_i) - \sum_{j\leq i} \eta_{ij} \int_M \zeta(\mathbf{x}) g_i(\mathbf{x}) g_j(\mathbf{x}) \, \mathrm{d}x \,, \tag{A.5}$$

where η_{ij} are Lagrange multipliers.

Using the same variational calculation as in Franzius et al. (2007), it can now be shown that the optimal functions g_j for which $\mathcal{L}$ becomes stationary are the solutions of a partial differential eigenvalue problem with von

Neumann boundary conditions:

$$-\nabla \cdot p(\mathbf{x})K(\mathbf{x})\nabla g_j(\mathbf{x}) = \eta_{jj}\zeta(\mathbf{x})g_j(\mathbf{x}), \tag{A.6}$$

$$p(\mathbf{x})\mathbf{n}(\mathbf{x}) \cdot K(\mathbf{x})\nabla g_j(\mathbf{x}) = 0 \quad \text{on the boundary}, \tag{A.7}$$

where $\mathbf{n}(\mathbf{x})$ is the field of normal vectors on the boundary of the manifold.

The Taylor approximation that was necessary for this derivation is appropriate if the spatial extent of the function π (the square of which is represented in the eigenvalues of K) is smaller than the spatial scale on which the function g varies. Because both LEMs and SFA are aiming primarily at smooth functions, this assumption is most likely fulfilled for the low-eigenvalue solutions of the problem. Significant differences between the eigenfunctions of the full problem and the eigenfunctions of the approximate differential operator will appear only for higher-order functions, which vary on a scale of the same order as or smaller than the characteristic width of π.

References

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*(6), 1373–1396.

Belkin, M., & Niyogi, P. (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning*, *56*(1), 209–239.

Belkin, M., & Niyogi, P. (2008). Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, *74*(8), 1289–1308.

Berkes, P. (2005). Pattern recognition with slow feature analysis. *Cognitive Sciences EPrint Archive (CogPrints)*, 4104.

Berkes, P., & Wiskott, L. (2005). Slow feature analysis yields a rich repertoire of complex cells. *Journal of Vision*, *5*(6), 579–602.

Böttcher, A., Grudsky, S., Maksimenko, E., & Unterberger, J. (2009). The first order asymptotics of the extreme eigenvectors of certain Hermitian Toeplitz matrices. *Integral Equations and Operator Theory*, *63*(2), 165–180.

Fergus, R., Weiss, Y., & Torralba, A. (2009). Semi-supervised learning in gigantic image collections. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, & A. Culote (Eds.), *Advances in neural information processing systems, 22* (pp. 522–530). Cambridge, MA: MIT Press.

Franzius, M., Sprekeler, H., & Wiskott, L. (2007). Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology, 3*(8), e166.

Franzius, M., Wilbert, N., & Wiskott, L. (2008). Invariant object recognition with slow feature analysis. In *Proceedings of the 18th Conf. on Artificial Neural Networks-ICANN 2008* (pp. 961–970). Berlin: Springer.

Hein, M., Audibert, J., & von Luxburg, U. (2007). Convergence of graph Laplacians on random neighborhood graphs. *Journal of Machine Learning Research*, *8*, 1325–1370.

Klampfl, S., & Maass, W. (2010). A theoretical basis for emergent pattern discrimination in neural systems through slow feature extraction. *Neural Computation*, *22*(12), 2979–3035.

Legenstein, R., Wilbert, N., & Wiskott, L. (2010). Reinforcement learning on slow features of high-dimensional input streams. *Plos Computational Biology*, *6*(8).

Mahadevan, S., & Maggioni, M. (2007). Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research*, *8*, 2169–2231.

Ng, A., Jordan, A., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems*, *14.* Cambridge, MA: MIT Press.

Shi, J., & Malik, J. (2002). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Singer, A., & Coifman, R. (2008). Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, *25*(2), 226–239.

Sprekeler, H., Zito, T., & Wiskott, L. (2010). An extension of slow feature analysis for nonlinear blind source separation. *Cognitive Sciences EPrint Archive (CogPrints)*, 7056.

Sutton, R., & Barto, A. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Wiskott, L. (2003). Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, *15*(9), 2147–2177.

Wiskott, L., & Sejnowski, T. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, *14*, 715–770.

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. In S. Thrün, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems*, *16* (pp. 595–602). Cambridge, MA: MIT Press.

Received November 9, 2010; accepted June 13, 2011.