The Laplacian Eigenmaps Latent Variable Model

Miguel Á. Carreira-Perpiñán **Zhengdong Lu**
Dept. of Computer Science & Electrical Engineering, OGI, Oregon Health & Science University
20000 NW Walker Road, Beaverton, OR 97006, USA
Email: {miguel,zhengdon}@csee.ogi.edu

Abstract

We introduce the Laplacian Eigenmaps Latent Variable Model (LELVM), a probabilistic method for nonlinear dimensionality reduction that combines the advantages of spectral methods—global optimisation and ability to learn convoluted manifolds of high intrinsic dimensionality—with those of latent variable models—dimensionality reduction and reconstruction mappings and a density model. We derive LELVM by defining a natural out-of-sample mapping for Laplacian eigenmaps using a semi-supervised learning argument. LELVM is simple, nonparametric and computationally not very costly, and is shown to perform well with motion-capture data.

Consider the problem of dimensionality reduction, where we observe data $\mathbf{y} \in \mathbb{R}^D$ assumed to lie on a manifold of dimension $L < D$, and want to model this data using latent variables $\mathbf{x} \in \mathbb{R}^L$. One class of dimensionality reduction methods are continuous latent variable models (LVMs; see Carreira-Perpiñán, 2001 for review); examples include factor analysis (linear) and the generative topographic mapping (GTM, nonlinear; Bishop et al., 1998). LVMs have two important properties: (1) they define a joint probability density $p(\mathbf{x}, \mathbf{y})$, thus densities $p(\mathbf{x})$ and $p(\mathbf{y})$ for the latent and observed variables, respectively. This means they can deal with missing data naturally, they can be used to model prior distributions (e.g. in tracking), and they can also be combined into mixtures of LVMs. (2) They define mappings for dimensionality reduction $\mathbf{x} = \mathbf{F}(\mathbf{y})$ and reconstruction $\mathbf{y} = \mathbf{f}(\mathbf{x})$ which can be applied to unseen data. However, they also have two important practical problems: (1) it is difficult to use more than 2 latent variables if using nonlinear models because the computational cost grows exponentially with L (or, if using mixtures of local linear models, aligning the component spaces is difficult). (2) Parameter estimation (by maximum likelihood) can easily result in bad local optima which yield entangled mappings. For some important applications, for example tracking 3D articulated pose from monocular video (Urtasun et al., 2005), it is crucial to have nonlinear, global, differentiable mappings and densities that use a latent (state) space of dimension possibly quite higher than 2.

Another class of dimensionality reduction methods of interest in recent years are spectral methods, such as Isomap (Tenenbaum et al., 2000), LLE (Roweis and Saul, 2000) or Laplacian eigenmaps (Belkin and Niyogi, 2003). These methods provide a set of correspondences $(\mathbf{x}_n, \mathbf{y}_n)$ of latent and observed points for points in the training set, and this correspondence is often able to learn complex manifolds that are very challenging for LVMs. Their estimation (by solving an eigenproblem) has no local optima and can easily yield latent spaces of dimension more than 2. However, they define neither mappings for out-of-sample points nor a probability density.

Thus, LVMs and spectral methods have somewhat complementary advantages and disadvantages, so it would be desirable to combine them to get the best of both worlds. One way of partially doing this is to use the correspondences produced by a spectral method to initialise a LVM, or to fit a dimensionality reduction or reconstruction mapping (sec. 4). However, up to now there has not been a formal relation between both types of methods. We address this question in this paper by proposing a natural extension of Laplacian eigenmaps to out-of-sample points that yields both dimensionality reduction and reconstruction mappings, and a joint density $p(\mathbf{x}, \mathbf{y})$, thus defining a LVM that we call Laplacian eigenmaps latent variable model (LELVM). Before describing this extension in section 3, we first review LVMs (in particular GTM) and spectral methods (in particular Laplacian eigenmaps) in sections 1–2. We demonstrate the

method in toy problems and real-world problems (in particular motion-capture data) in section 4, and discuss related work in section 5.

1 Latent variable models: GTM

A LVM is defined by a prior distribution $p(\mathbf{x})$ in latent space ($\mathbf{x} \in \mathbb{R}^L$), a (reconstruction) mapping $\mathbf{f}: \mathbf{x} \rightarrow \mathbf{y}$, and a noise model $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{f}(\mathbf{x}))$ in observed space ($\mathbf{y} \in \mathbb{R}^D$, $D > L$). The density in observed space is then obtained by marginalising the joint density $p(\mathbf{x}, \mathbf{y})$ in latent space: $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\, d\mathbf{x}$. A dimensionality reduction mapping can be defined from the posterior distribution in latent space $p(\mathbf{x}|\mathbf{y})$ as either the mean or, for multimodal distributions, the modes. Estimating the parameters in $p(\mathbf{x})$, $\mathbf{f}$ and $p(\mathbf{y}|\mathbf{x})$ is achieved by maximum likelihood given a sample $\{\mathbf{y}_n\}_{n=1}^N$ in observed space, often by an EM algorithm (where the latent variables are the missing data). Mixtures of LVMs $p(\mathbf{y}|m)$ for $m = 1, \dots, M$ are readily defined as $\sum_{m=1}^M p(m)p(\mathbf{y}|m)$. A linear example of LVM is factor analysis, where $p(\mathbf{x})$ is Gaussian, $\mathbf{f}$ is linear and $p(\mathbf{y}|\mathbf{x})$ is diagonal Gaussian; other linear LVMs include probabilistic PCA and independent component analysis. A nonlinear example is GTM (Bishop et al., 1998), where $p(\mathbf{x}) = \sum_{k=1}^K \delta(\mathbf{x} - \mathbf{x}_k)$ (discretised uniform with points $\mathbf{x}_k$ arranged on a square grid for visualisation), $\mathbf{f} = \mathbf{W}\mathbf{\Phi}(\mathbf{x})$ is a radial basis function mapping (with fixed basis functions $\mathbf{\Phi} = (\phi_1, \dots, \phi_F)$ and tunable weights $\mathbf{W}$), and $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{f}(\mathbf{x}), \sigma^2\mathbf{I})$ is isotropic Gaussian. The observed density $p(\mathbf{y})$ is then a Gaussian mixture with centres constrained by $\mathbf{f}$. The discretisation of $\mathbf{x}$ is a computational device to obtain a closed-form marginal $p(\mathbf{y})$ in observed space, but the number of grid points K grows exponentially with the latent dimension L, limiting it in practice to $L \lesssim 3$. When modelling complex, convoluted manifolds such as a spiral, initialising GTM's training from either random parameter values or the PCA solution almost always results in a tangled manifold (see fig. 1). The problem is not one of modelling ability but of search: the log-likelihood function is full of bad optima.

The recently proposed Gaussian Process Latent Variable Model (GPLVM) (Lawrence, 2005) marginalises over functions $\mathbf{f}$ rather than over latent variables $\mathbf{x}$. Essentially, it yields a Gaussian process reconstruction mapping $\mathbf{f}$ and a conditional probability $p(\mathbf{y}|\mathbf{x})$, but neither a joint density $p(\mathbf{x}, \mathbf{y})$ nor a posterior distribution $p(\mathbf{x}|\mathbf{y})$ in latent space. Unlike in GTM, in GPLVM the latent points $\{\mathbf{x}_n\}_{n=1}^N$ associated with the data points are tunable parameters, which makes initialisation from a LE solution direct (and so can succeed with convoluted mappings), and also allows practical use of $L > 2$. However, its log-likelihood function also has many bad local optima and training is computationally very costly: each gradient iteration is $\mathcal{O}(N^3)$ (since the covariance matrix involved is not sparse), though approximations based on using few points and so small N exist to accelerate it.

2 Spectral methods: Laplacian eigenmaps (LE)

Define a neighbourhood graph on the sample data $\{\mathbf{y}_n\}_{n=1}^N$, such as a k-nearest-neighbour or ϵ-ball graph, or a full mesh, and weigh each edge $\mathbf{y}_n \sim \mathbf{y}_m$ by a symmetric affinity function $K(\mathbf{y}_n, \mathbf{y}_m) = w_{nm}$, typically Gaussian: $w_{nm} = \exp\left(-\frac{1}{2}\left\|(\mathbf{y}_n - \mathbf{y}_m)/\sigma\right\|^2\right)$. Given this weighted graph, in Laplacian eigenmaps (LE) (Belkin and Niyogi, 2003) we seek latent points $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^L$ that are the solution of the optimisation problem:

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

where we define the matrix $\mathbf{X}_{L\times N} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$, the symmetric affinity matrix $\mathbf{W}_{N\times N}$, the degree matrix $\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^N w_{nm}\right)$, the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, and $\mathbf{1} = (1, \dots, 1)^T$. The objective function can be rewritten as $\sum_{n\sim m} w_{nm}\left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2$, which discourages placing far apart latent points that correspond to similar observed points. The constraints eliminate the two trivial solutions $\mathbf{X} = \mathbf{0}$ (by setting an arbitrary scale) and $\mathbf{x}_1 = \cdots = \mathbf{x}_N$ (by removing $\mathbf{1}$, which is an eigenvector of $\mathbf{L}$ associated with a zero eigenvalue). The solution is given by the leading $\mathbf{u}_2, \dots, \mathbf{u}_{L+1}$ eigenvectors of the normalised affinity matrix $\mathbf{N} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, namely $\mathbf{X} = \mathbf{V}^T\mathbf{D}^{-\frac{1}{2}}$ with $\mathbf{V}_{N\times L} = (\mathbf{v}_2, \dots, \mathbf{v}_{L+1})$ (an a posteriori translated, rotated or uniformly scaled $\mathbf{X}$ is equally valid).

Other spectral methods result from optimisation problems like (1) but defining matrices different from $\mathbf{L}$ in the objective, and possibly different constraints. For example, metric multidimensional scaling uses the matrix of Euclidean squared distances an a full mesh (all point pairs), Isomap uses instead geodesic distances (approximated as shortest-path distances in a neighbourhood graph) and LLE uses reconstruction weights $\mathbf{W}$ and an objective matrix $(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$. The result is a set of latent points $\{\mathbf{x}_n\}_{n=1}^N$ in correspondence with the observed points $\{\mathbf{y}_n\}_{n=1}^N$ which often succeeds in learning complex manifolds. However, they do not provide a mapping for out-of-sample points (see section 5), let alone a density in the latent or observed space.

3 The Laplacian Eigenmaps Latent Variable Model (LELVM)

Assume we have obtained a LE embedding $\mathbf{X}_\mathrm{s} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ of seen points $\mathbf{Y}_\mathrm{s} = (\mathbf{y}_1, \dots, \mathbf{y}_N)$, and consider a set of unseen (out-of-sample) points in observed space $\mathbf{Y}_\mathrm{u} = (\mathbf{y}_{N+1}, \dots, \mathbf{y}_{N+M})$. The natural way to embed the new points would be to recompute the whole embedding $(\mathbf{X}_\mathrm{s}\ \mathbf{X}_\mathrm{u})$ for $(\mathbf{Y}_\mathrm{s}\ \mathbf{Y}_\mathrm{u})$ from eq. (1). This is computationally costly and does not lead to defining a mapping for the new points; we seek a way of keeping the old embedding fixed and embed new points based on that. Then, the next most natural way is to recompute the embedding but keeping the old points fixed:

$$\min_{\mathbf{X}_\mathrm{u} \in \mathbb{R}^{L \times M}} \operatorname{tr}\left(\begin{pmatrix} \mathbf{X}_\mathrm{s} & \mathbf{X}_\mathrm{u} \end{pmatrix} \begin{pmatrix} \mathbf{L}_\mathrm{ss} & \mathbf{L}_\mathrm{su} \\ \mathbf{L}_\mathrm{us} & \mathbf{L}_\mathrm{uu} \end{pmatrix} \begin{pmatrix} \mathbf{X}_\mathrm{s}^T \\ \mathbf{X}_\mathrm{u}^T \end{pmatrix} \right). \tag{2}$$

We need not use the constraints from (1) because the trivial solutions $\mathbf{X} = \mathbf{0}$ and $\mathbf{X} =$ constant were already removed in the old embedding[1]. The solution is $\mathbf{X}_\mathrm{u} = -\mathbf{X}_\mathrm{s}\mathbf{L}_\mathrm{su}\mathbf{L}_\mathrm{uu}^{-1}$. This point of view can also be considered as semi-supervised learning, where we consider the embedding $\mathbf{X}_\mathrm{s}$ as (real-valued) labels for $\mathbf{Y}_\mathrm{s}$ and want to label $\mathbf{Y}_\mathrm{u}$ by using a graph prior (Zhu et al., 2003; Ham et al., 2005; Yang et al., 2006). If we now consider a *single* out-of-sample point (i.e., $M = 1$) and write $\mathbf{y} = \mathbf{Y}_\mathrm{u} \in \mathbb{R}^D$ and $\mathbf{x} = \mathbf{X}_\mathrm{u} \in \mathbb{R}^L$, and recalling that $\mathbf{L} = \mathbf{D} - \mathbf{W}$ so that $\mathbf{L}_\mathrm{su} = -\mathbf{W}_\mathrm{su} = -\mathbf{K}(\mathbf{y}) \in \mathbb{R}^N$ and $l_\mathrm{uu} = d_\mathrm{u} - w_\mathrm{uu} = \mathbf{1}^T\mathbf{K}(\mathbf{y})$, the previous argument allows us to derive an out-of-sample dimensionality reduction mapping $\mathbf{x} = \mathbf{F}(\mathbf{y})$ applicable to any point $\mathbf{y}$ (new or old), namely:

$$\begin{aligned} \mathbf{x} = \mathbf{F}(\mathbf{y}) &= -\frac{1}{l_\mathrm{uu}}\mathbf{X}_\mathrm{s}\mathbf{L}_\mathrm{su} = \frac{\mathbf{X}_\mathrm{s}\mathbf{K}(\mathbf{y})}{\mathbf{1}^T\mathbf{K}(\mathbf{y})} \\ &= \sum_{n=1}^N \frac{K(\mathbf{y}, \mathbf{y}_n)}{\sum_{n'=1}^N K(\mathbf{y}, \mathbf{y}_{n'})}\mathbf{x}_n. \end{aligned} \tag{3}$$

This mapping is formally identical to a Nadaraya-Watson estimator (kernel regression; Wand and Jones, 1994) using data $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ and kernel K. We can take this a step further by defining a LVM that has as joint distribution a kernel density estimate (KDE):

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{N}\sum_{n=1}^N K_\mathbf{y}(\mathbf{y}, \mathbf{y}_n) K_\mathbf{x}(\mathbf{x}, \mathbf{x}_n)$$

where $K_\mathbf{y}$ is proportional to K so it integrates to 1, and $K_\mathbf{x}$ is a pdf kernel in $\mathbf{x}$–space. Consequently, the marginals in observed and latent space are also KDEs:

$$p(\mathbf{y}) = \frac{1}{N}\sum_{n=1}^N K_\mathbf{y}(\mathbf{y}, \mathbf{y}_n) \qquad p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^N K_\mathbf{x}(\mathbf{x}, \mathbf{x}_n)$$

[1] For the case $M = 1$ considered later, the constraints would trivially determine $\mathbf{X}_\mathrm{u}$ and the formulation would be nonsensical anyway.

and the dimensionality reduction and reconstruction mappings are given by kernel regression:

$$\begin{aligned} \mathbf{F}(\mathbf{y}) &= \sum_{n=1}^N \frac{K_\mathbf{y}(\mathbf{y}, \mathbf{y}_n)}{\sum_{n'=1}^N K_\mathbf{y}(\mathbf{y}, \mathbf{y}_{n'})}\mathbf{x}_n = \sum_{n=1}^N p(n|\mathbf{y})\mathbf{x}_n \\ \mathbf{f}(\mathbf{x}) &= \sum_{n=1}^N \frac{K_\mathbf{x}(\mathbf{x}, \mathbf{x}_n)}{\sum_{n'=1}^N K_\mathbf{x}(\mathbf{x}, \mathbf{x}_{n'})}\mathbf{y}_n = \sum_{n=1}^N p(n|\mathbf{x})\mathbf{y}_n \end{aligned} \tag{4}$$

which are the conditional means $\mathrm{E}\{\mathbf{x}|\mathbf{y}\}$ and $\mathrm{E}\{\mathbf{y}|\mathbf{x}\}$. In other words, we have augmented the mapping $\mathbf{F}$ of (3) with a probability model $p(\mathbf{x}, \mathbf{y})$ that is consistent with $\mathbf{F}$ (since $\mathbf{F}(\mathbf{y}) = \mathrm{E}\{\mathbf{x}|\mathbf{y}\}$) and symmetric wrt $\mathbf{x}$ and $\mathbf{y}$. The equations are valid if we assume that $K(\mathbf{y}, \mathbf{y}_n)$ is proportional to a pdf with mean $\mathbf{y}_n$, which holds for the Gaussian affinity; they may be valid under more general assumptions on K. The kernel $K_\mathbf{x}$ need not be the same as K, and in particular will usually have a different bandwidth. We call this model the *Laplacian Eigenmaps Latent Variable Model (LELVM)*, and for simplicity consider in the rest of the paper that both $K_\mathbf{y}$ and $K_\mathbf{x}$ are isotropic Gaussian (consistent with Gaussian affinities), i.e., $K_\mathbf{x}(\mathbf{x}, \mathbf{x}_n) \propto \exp\left(-\frac{1}{2}\left\|(\mathbf{x} - \mathbf{x}_n)/\sigma_\mathbf{x}\right\|^2\right)$ and $K_\mathbf{y}(\mathbf{y}, \mathbf{y}_n) \propto \exp\left(-\frac{1}{2}\left\|(\mathbf{y} - \mathbf{y}_n)/\sigma_\mathbf{y}\right\|^2\right)$.

Note that, under the mappings defined in (4), the correspondences from LE are not respected anymore unless $\sigma_\mathbf{x},\ \sigma_\mathbf{y} \to 0$: $\mathbf{x}_n \neq \mathbf{F}(\mathbf{y}_n)$ and $\mathbf{y}_n \neq \mathbf{f}(\mathbf{x}_n)$ (though the error is small; see also sec. 5). This is not inconsistent: the role of the model should be to smooth the training data rather than interpolate it, so as to generalise well. As seen in sec. 4, lower error for small bandwidths is achieved at the cost of smoothness.

The LELVM has several attractive properties. It defines both dimensionality reduction and reconstruction mappings applicable to any new point; both mappings are continuous (even infinitely differentiable if $K_\mathbf{x}$, $K_\mathbf{y}$ are), nonlinear and based on a global coordinate system (unlike in mixtures of local models). It defines a probability model that can represent multimodal distributions and deal with missing data by marginalisation. It can use a continuous latent space of arbitrary dimension L (unlike GTM) by simply choosing L eigenvectors in the LE embedding. It has no local optima since it is based on the LE embedding. Besides being useful as a general-purpose dimensionality reduction method, these properties make it attractive for representing priors in tracking in a Bayesian framework—in particular, in articulated pose tracking from monocular video, where a high-dimensional state space of joint angles can be represented by a nonlinear manifold whose intrinsic dimension may exceed 2.

The densities are defined over the whole Euclidean space, however since the mappings in (4) are convex

sums, the ranges of $\mathbf{f}$ and $\mathbf{F}$ are the convex hulls of the training data $\mathbf{Y}$ and the latent points $\mathbf{X}$, respectively. Thus, for visualisation purposes it makes sense to focus attention on a bounding box of the convex hull. The prior density in this convex hull need not be uniform though, because the centres are not uniformly distributed (compare with GTM's latent space, given by a square, discretised uniform grid). If the posterior distributions $\mathbf{x}|\mathbf{y}$ or $\mathbf{y}|\mathbf{x}$ are multimodal, it may make more sense to define multivalued mappings given by their modes (Carreira-Perpiñán, 2000b). Iterative algorithms exist for locating all modes of a Gaussian mixture (Carreira-Perpiñán, 2000, 2007), in particular the mean-shift algorithm. For isotropic Gaussian mixtures, the modes (in fact all stationary points) also lie within the convex hull of the centres. Interestingly, note how the noise model $\mathbf{y}|\mathbf{x}$ is a Gaussian mixture and can thus represent skewed or even multimodal distributions, unlike most usual LVMs where it is taken as a Gaussian.

So far we have implicitly assumed that the graph used to obtain the LE embedding is a full mesh. For a neighbourhood graph such as a k-nearest-neighbour or ϵ-ball graph, the graph Laplacian $\mathbf{L}$ in problem (1) is sparse (with nonzero weights corresponding to the graph edges). The out-of-sample dimensionality reduction mapping is still given by $\mathbf{x} = -\mathbf{X}_\mathrm{s}\mathbf{L}_\mathrm{su}/l_\mathrm{uu}$ but now it involves only the points adjacent to $\mathbf{y}$:

$$\mathbf{x} = \mathbf{F}(\mathbf{y}) = \sum_{\mathbf{y}_n \sim \mathbf{y}} \frac{K_\mathbf{y}(\mathbf{y}, \mathbf{y}_n)}{\sum_{\mathbf{y}_{n'} \sim \mathbf{y}} K_\mathbf{y}(\mathbf{y}, \mathbf{y}_{n'})} \mathbf{x}_n. \qquad (5)$$

The points in $\mathbf{Y}_\mathrm{s}$ adjacent to $\mathbf{y}$ are determined using the same graph construction, e.g. the k nearest neighbours of $\mathbf{y}$, or the points at distance ϵ or less from $\mathbf{y}$. Note that only the edges adjacent to $\mathbf{y}$ matter—edges between points in $\mathbf{Y}_\mathrm{s}$ (even if they were to be updated after adding $\mathbf{y}$ to the graph) contribute to the matrix $\mathbf{L}_\mathrm{ss}$ which affects only the constant term $\mathrm{tr}\left(\mathbf{X}_\mathrm{s}\mathbf{L}_\mathrm{ss}\mathbf{X}_\mathrm{s}^T\right)$ in the optimisation problem. In practice, neighbourhood graphs are crucial for the success of LE and also reduce the computational complexity of the eigenproblem from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$. However, computing the graph costs $\mathcal{O}(DN^2)$.

Just as a KDE is a nonparametric density estimate, a LELVM is a nonparametric LVM. The only parameters to be fit or set by the user are the graph parameters for the LE embedding (affinity width σ, and k or ϵ), and the bandwidths for the KDE ($\sigma_\mathbf{x}$, $\sigma_\mathbf{y}$). And as with a KDE, multiple criteria seem possible to select $\sigma_\mathbf{x}$ and $\sigma_\mathbf{y}$ (Wand and Jones, 1994), e.g.:

- equal to σ in the affinity function ($\sigma_\mathbf{y}$ only)
- maximum likelihood estimator (the usual objective function for LVMs)
- average of distances to k nearest neighbours (also useful to define adaptive KDEs, i.e., a different bandwidth for each centre)
- cross-validation of the log-likelihood $\log p(\mathbf{y})$ or the reconstruction error $\|\mathbf{y} - \mathbf{f}(\mathbf{F}(\mathbf{y}))\|^2$
- minimising the embedding error on a test set.

We explore some of these in section 4. It is also possible to use a different bandwidth per dimension (diagonal covariance), or a different bandwidth per point (adaptive KDE). Note that, given a fixed LE embedding, trying different bandwidths takes $\mathcal{O}(N)$ with sparse graphs or $\mathcal{O}(N^2)$ with full mesh, which is N times faster than redoing the embedding.

4 Experiments

In our experiments, we selected LE parameters (k, σ) that produced a good embedding $\mathbf{X}$. Figure 1 shows results using the spiral data set (with added Gaussian noise), whose convoluted manifold is difficult to disentangle for GTM (we obtained similar results with the Swiss roll). To obtain the LELVM, we ran LE with $k = 10$ and Gaussian affinity width $\sigma = 0.4$. We tested different values for $\sigma_\mathbf{x}$ and $\sigma_\mathbf{y}$, which affect the smoothness of $p(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$, and of $p(\mathbf{y})$ and $\mathbf{F}(\mathbf{y})$, respectively (panels $\mathbf{A}$, $\mathbf{B}$). We find that the cross-validation rules tend to give small bandwidths ($\sigma_\mathbf{x} \approx 1.3 \cdot 10^{-4}$) that result in relatively jagged mappings; better mappings result for larger bandwidths (e.g. $\sigma_\mathbf{x} \approx 2.5 \cdot 10^{-4}$ with the average neighbour distances), while very large bandwidths bias the mapping excessively. An interesting smoothing effect of the bandwidth is that, if large enough, it can unfold loops (partially visible in the plots for small $\sigma_\mathbf{x}$) mistakenly produced by the LE embedding (i.e., points $\mathbf{x}_n$ locally disordered wrt $\mathbf{y}_n$). A noticeable effect is the shortening of the manifold at the boundaries; this is caused by the fact that the KDE is a convex sum of the centres. Good mappings may be obtained for small datasets (panel $\mathbf{C}$), at a lower computational cost. Predictably, GTM fails to recover the manifold if initialised from the PCA (or random) solution, but succeeds if initialised from the LE embedding. The latter is tricky, though: by design, the GTM mapping $\mathbf{f}$ has its RBF centres distributed on a square grid, which may not match well (even after a rigid transformation) the convex hull of the $\mathbf{X}$ points from LE. Another possibility (not explored here) could be to give up GTM's square grid and use directly the LE embedding $\mathbf{X}$ to define GTM's prior $p(\mathbf{x}) = \sum_{n=1}^N \delta(\mathbf{x} - \mathbf{x}_n)$ (and suitably choosing the RBF centres); this would allow use of latent dimension > 2. The GTM mapping $\mathbf{f}$ does not shorten at the boundaries because it is not a convex sum but a

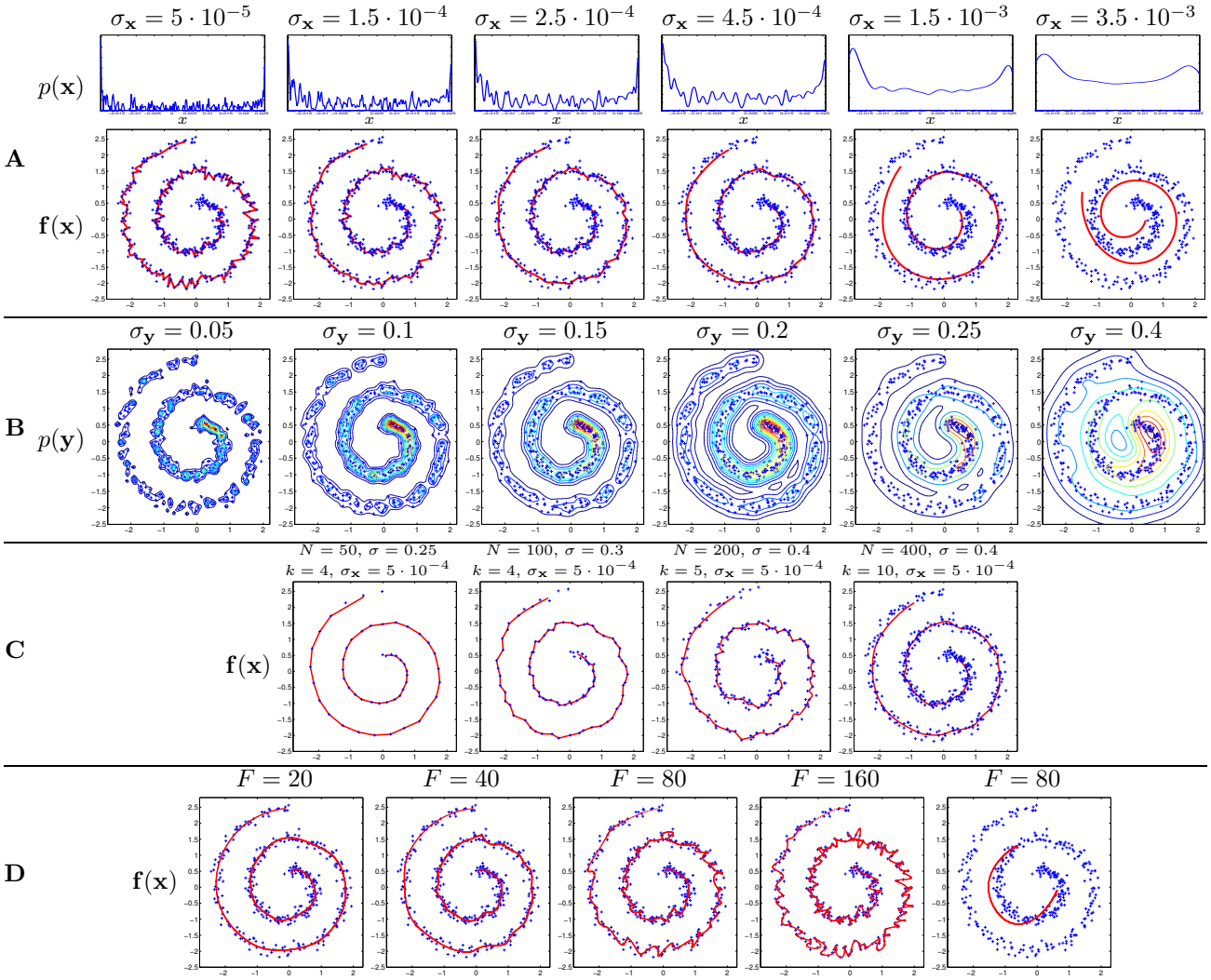

Figure 1: Results for the spiral dataset ($N = 400$ points in 2D). **A**–**B** show LELVM results obtained from the LE embedding using a k-nearest-neighbour graph ($k = 10$) and Gaussian affinities ($\sigma = 0.4$). **A** shows the latent density $p(\mathbf{x})$ and dimensionality reduction mapping $\mathbf{f}(\mathbf{x})$ for different values of $\sigma_{\mathbf{x}}$; **B** shows the observed density $p(\mathbf{y})$ for different values of $\sigma_{\mathbf{y}}$. **C**: LELVM mapping $\mathbf{f}(\mathbf{x})$ obtained for different numbers of training samples N (appropriate k, σ, $\sigma_{\mathbf{x}}$ were used in each case). **D**: GTM results ($N = 400$ points) for different numbers F of RBFs (the width of each RBF equals the separation between grid points in latent space). The right plot was initialised from the PCA solution, while the others were initialised from the LE embedding.

RBF; its smoothness depends on the number of RBFs F and their width (note the loops for high F).

Figure 2 shows results using motion-capture data recorded from several cycles of a running sequence. This data is important for articulated pose tracking, and difficult because of the nonlinear constraints of body motion and its high dimensionality (Urtasun et al., 2005). We used the same data and preprocessing as Lawrence and Quiñonero-Candela (2006), resulting in a sequence of 217 points in 102D (corresponding to 34 3D markers), normalised for translation (i.e., zero-mean) but not for rotation. A 2D latent space is enough for this problem and the LELVM manifold reveals a cyclic pattern where the initial points (at the top; start of motion) quickly converge on a closed loop that is repeatedly travelled by subsequent cycles. This loop can be used to characterise this particular running pattern. We also tried a 3D space, where the loop merely bends; higher L (e.g. for more complex motion data) would be simple to achieve by simply adding more eigenvectors to the embedding. The prior density $p(\mathbf{x})$ is higher where the latent points pile up, i.e., where the runner changes pose more slowly. The loop is not circular because the pose changes at different velocities depending on its position in the cycle.

Panels **A**(right, bottom) show how the continuous reconstruction mapping $\mathbf{F}(\mathbf{x})$ can smoothly interpolate between poses by travelling around the latent space. None of the reconstructed poses are in the training set (except the first and last ones) but they do interpolate the motion in a realistic way. Since spectral methods lack mappings, interpolating in them requires selecting exemplars from the training set (and possibly averaging them) in some ad-hoc way. Experiments (not shown) with faces and digits data (Tenenbaum et al., 2000) also confirm the ability to interpolate smoothly in image manifolds.

Panel **B** shows the ability of LELVM to reduce dimension and reconstruct when part of the observed variables are missing. Here, we computed the posterior latent distribution

$$p(\mathbf{x}|\mathbf{y}_{\text{obs}}) = \frac{\int p(\mathbf{x},\mathbf{y})\, d\mathbf{y}_{\text{mis}}}{\int p(\mathbf{y})\, d\mathbf{y}_{\text{mis}}} = \sum_n \alpha_n e^{-\frac{1}{2}\left\|\frac{\mathbf{x}-\mathbf{x}_n}{\sigma_{\mathbf{x}}}\right\|^2}$$

with $\alpha_n \propto \exp\left(-\frac{1}{2}\left\|(\mathbf{y}_{\text{obs}}-\mathbf{y}_{n,\text{obs}})/\sigma_{\mathbf{y}}\right\|^2\right)$ (shown as the contour), and used its mean if unimodal or its modes otherwise (found with the algorithms in Carreira-Perpiñán, 2000, 2007). Then we reconstructed with $\mathbf{f}$. We could have operated directly in the 102D observed space by using $p(\mathbf{y}_{\text{mis}}|\mathbf{y}_{\text{obs}}) = p(\mathbf{y})/p(\mathbf{y}_{\text{obs}})$ (using $\mathbf{f}$ effectively turns $p(\mathbf{x}|\mathbf{y}_{\text{obs}})$ into a delta function). However, mode finding is more efficient and reliable in low dimensions, and reducing dimensionality also has a beneficial denoising effect.

GTM (panel **C** left) fails even when started from the LE embedding. The GPLVM latent space (panel **C** middle, right) partly captures the periodic nature of the motion but does not quite collect the loops, even when back constraints on the distances are added.

5 Discussion and related work

We do not claim that our out-of-sample extension (though natural) is the only possible one. Another out-of-sample extension has been proposed for spectral methods by Bengio et al. (2004), based on the Nyström formula (originally applied by Williams and Seeger (2001) to approximate Gaussian process computations in terms of a small subset of representative vectors). The formula involves the L leading (except the top one) eigenvalues $\mathbf{\Lambda} = \operatorname{diag}(\lambda_2,\dots,\lambda_{L+1})$ and eigenvectors $\mathbf{V}_{N\times L}$ of the normalised affinity matrix $\mathbf{N} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ (thus $\mathbf{NV} = \mathbf{\Lambda V}$), and requires defining a kernel function that "generates" the observed normalised affinities. For Laplacian eigenmaps, the kernel choice of Bengio et al. (2004) yields the out-of-sample mapping (in matrix notation):

$$\tilde{\mathbf{F}}(\mathbf{y}) = \sqrt{N}\mathbf{\Lambda}^{-1}\mathbf{X}\frac{\mathbf{K}(\mathbf{y})}{\sqrt{\mathbf{1}^T\mathbf{K}(\mathbf{y})}} \tag{6}$$

where $\mathbf{X} = \mathbf{V}^T\mathbf{D}^{-\frac{1}{2}}$ are the latent points defined by LE (sec. 2). Bengio et al. (2004) use a different embedding given by $\tilde{\mathbf{X}} = \sqrt{N}\mathbf{V}^T$, for which one can show that $\tilde{\mathbf{F}}(\mathbf{Y}) = \tilde{\mathbf{X}}$ so the mapping interpolates the pairs $(\tilde{\mathbf{x}}_n, \mathbf{y}_n)$. In the same matrix notation, the LELVM dimensionality reduction mapping of (3) is:

$$\mathbf{F}(\mathbf{y}) = \mathbf{X}\frac{\mathbf{K}(\mathbf{y})}{\mathbf{1}^T\mathbf{K}(\mathbf{y})} = \mathbf{V}^T\mathbf{D}^{-\frac{1}{2}}\frac{\mathbf{K}(\mathbf{y})}{\mathbf{1}^T\mathbf{K}(\mathbf{y})} \tag{7}$$

which shows two differences with (6): (a) formula (6) scales the points by the inverse eigenvalues $\mathbf{\Lambda}^{-1}$ while (7) does not; this probably makes a small difference in practice, since the leading eigenvalues are very close to 1 ($= \lambda_1$). (b) More importantly, formula (7) is a convex sum (which allowed us to define the LELVM) while (6) is not. For $\sigma_{\mathbf{y}} = \sigma$ (the value used in the affinities) we obtain for (7) that $\mathbf{F}(\mathbf{Y}) = \mathbf{\Lambda X}$ which is different from $\mathbf{X}$ in general though again very close in practice. However, LELVM allows to choose $\sigma_{\mathbf{y}}$ in order to smooth more or less the mapping. As in Bengio et al. (2004), experiments with the spiral show the error to be comparable to the effect of small perturbations in the training set. Incidentally, from a model selection point of view this suggests that the latent dimension L should be small enough not to reach an eigenvalue λ_{L+1} significantly smaller than 1. Finally, Bengio et al. (2004) did not consider the case of a neighbourhood graph, and did not give a mapping $\mathbf{f}$ or a density model.

Meinicke et al. (2005) have proposed using the Nadaraya-Watson estimator to represent the reconstruction mapping $\mathbf{f}$, trained to minimise the reconstruction error over the latent points $\mathbf{X} = (\mathbf{x}_1,\dots,\mathbf{x}_N)$ and the bandwidth σ, essentially turning the unsupervised problem of dimensionality reduction into a supervised (regression) problem. However, the large number of parameters (for $\mathbf{X}$) and the nonlinear nature of the reconstruction error makes optimisation very unreliable. They propose several heuristics such as using different initialisations (including the $\mathbf{X}$ from LLE or LE), and using deterministic annealing. Memisevic (2006) also considers a supervised view of dimensionality reduction where a mutual information objective (equivalent to conditional log-likelihood) is optimised over $\mathbf{X}$ and σ using a KDE as plug-in estimator of the entropy, again using deterministic annealing. This defines mappings $\mathbf{f}$ and $\mathbf{F}$ but not in explicit form (rather, as the solution of an optimisation problem).

Model selection for the latent dimension L is computationally simple in LELVM because the LE embedding for L dimensions (which is uniquely defined) yields also the embeddings for $1,\dots,L-1$ dimensions at a cost only slightly larger than for $L = 1$ (as in PCA). This allows to apply criteria (e.g. for model complexity) to

select the best L, or even the best subset of L eigenvectors (which may differ from the leading L). In contrast, GTM or GPLVM need to run a costly nonlinear optimisation from scratch for each L, and the solution found is one among the many existing local optima.

6 Conclusion and future work

We have proposed a natural way (derived from semi-supervised learning arguments) to define out-of-sample mappings for Laplacian eigenmaps that suggests an extension to latent variable models, the LELVM. It is a very simple model: a kernel density estimate obtained from one-shot spectral training followed by bandwidth selection—thus a nonparametric LVM. Yet it is very powerful, combining the advantages of LVMs—continuous, global, nonlinear dimensionality reduction mappings and joint density—and spectral methods—training without local optima, and ability to use latent spaces of dimension as high as desired, and to deal with convoluted manifolds. This makes it suitable for tasks that require learning complex priors from sparse high-dimensional data, such as people tracking in video. The only free parameters are the bandwidths in latent and observed space (which control the mapping and density smoothness), whose tuning is problem dependent as is the case for kernel density estimation.

We are working on applying this approach to other spectral methods such as LLE. A question of theoretical interest is the behaviour of LELVM for large samples; consistency results for kernel density estimators and (though incomplete at present) for Laplacian eigenmaps suggest we might expect good approximation if the number of neighbours increases and the bandwidths decrease as $N \to \infty$.

Acknowledgements

MACP thanks Chris Williams for valuable discussions. Work funded by NSF CAREER award IIS–0546857.

References

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

Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10):2197–2219, Oct. 2004.

C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, Jan. 1998.

M. Á. Carreira-Perpiñán. Mode-finding for mixtures of Gaussian distributions. *IEEE Trans. PAMI*, 22 (11):1318–1323, Nov. 2000.

M. Á. Carreira-Perpiñán. Reconstruction of sequential data with probabilistic models and continuity constraints. In *NIPS*, volume 12, pages 414–420, 2000b.

M. Á. Carreira-Perpiñán. *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction*. PhD thesis, Dept. of Computer Science, University of Sheffield, UK, 2001. Available online at `http://www.csee.ogi.edu/~miguel/papers/phd-thesis.html`.

M. Á. Carreira-Perpiñán. Gaussian mean shift is an EM algorithm. *IEEE Trans. PAMI*, 2007.

J. Ham, D. Lee, and L. Saul. Semisupervised alignment of manifolds. In *AISTATS*, pages 120–127, 2005.

N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, Nov. 2005.

N. Lawrence and J. Quiñonero-Candela. Local distance preservation in the GP-LVM through back constraints. In *ICML*, pages 513–520, 2006.

P. Meinicke, S. Klanke, R. Memisevic, and H. Ritter. Principal surfaces from unsupervised kernel regression. *IEEE Trans. PAMI*, 27(9):1379–1391, Sept. 2005.

R. Memisevic. Kernel information embeddings. In *ICML*, pages 633–640, 2006.

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

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

R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *ICCV*, pages 403–410, 2005.

M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall, 1994.

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *NIPS*, volume 13, pages 682–688, 2001.

X. Yang, H. Fu, H. Zha, and J. Barlow. Semi-supervised nonlinear dimensionality reduction. In *ICML*, pages 1065–1072, 2006.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.

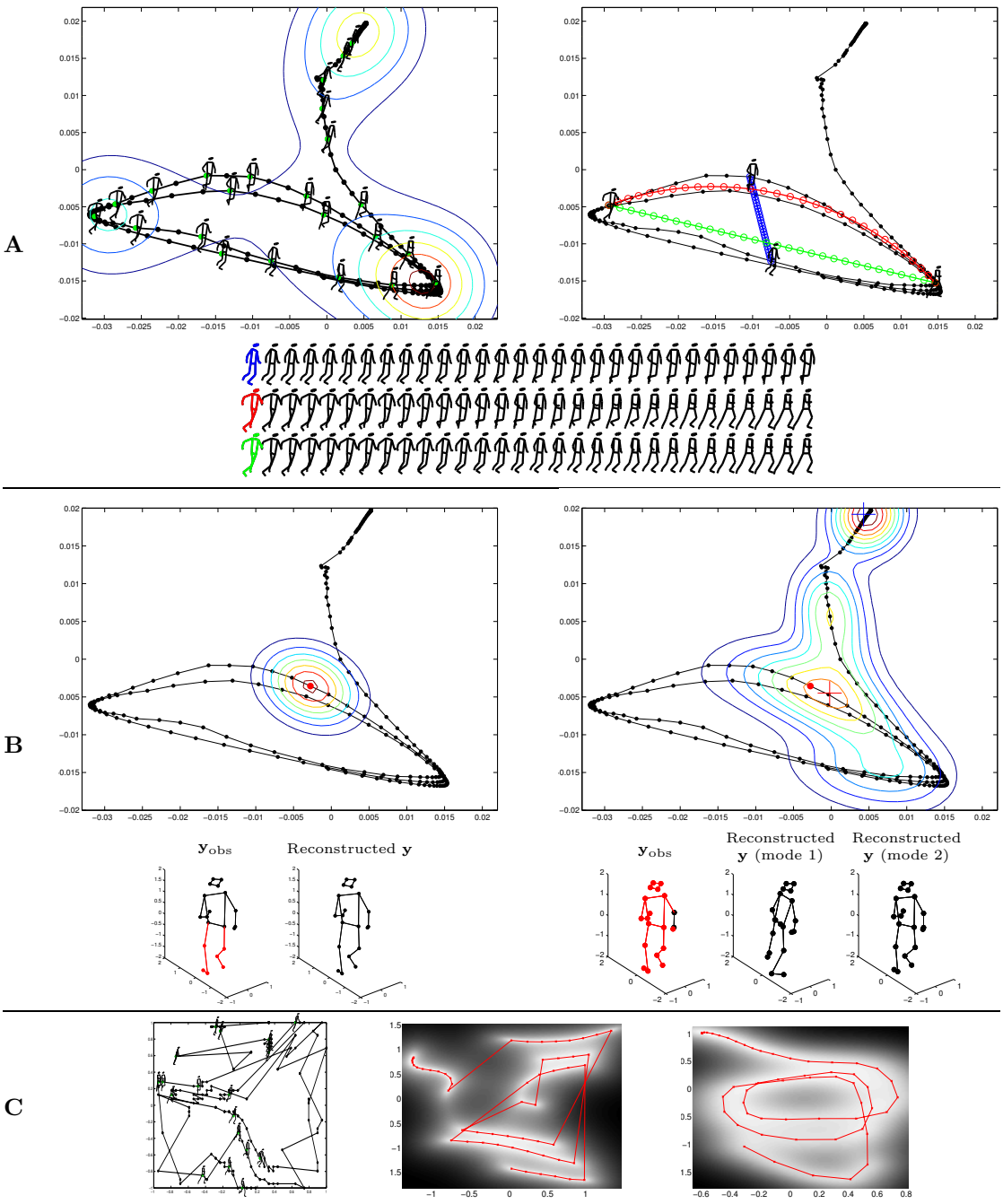

Figure 2: Results for the motion capture dataset ($N = 217$ points in 102D). **A**–**B** show LELVM results obtained from the LE embedding using a k-nearest-neighbour graph ($k = 40$) and Gaussian affinities ($\sigma = 1.5$), and with $\sigma_{\mathbf{x}} = 0.005$ and $\sigma_{\mathbf{y}} = 0.3$. **A**(left) shows the latent space; we connect data points $\mathbf{x}_n$ in the sequential order of their corresponding data points $\mathbf{y}_n$, and for some of those we plot $\mathbf{y}_n$ as a stick man; the loop is travelled clockwise. The contours indicate $p(\mathbf{x})$. **A**(right) shows 3 trajectories in latent space (containing 30 equispaced samples) and uses the mapping $\mathbf{f}(\mathbf{x})$ to reconstruct the corresponding trajectory in observed space (lower plot). For each trajectory, only the initial and final points were in the dataset; the rest are smoothly produced by the mapping. **B**: reconstruction of missing data with LELVM. Given a partially observed stick man (black: observed, red: missing) we show the contours of $p(\mathbf{x}|\mathbf{y}_{\text{obs}})$ and the reconstructed stick men. When the legs are missing, $p(\mathbf{x}|\mathbf{y}_{\text{obs}})$ is unimodal, but when only the forearm is observed, it is multimodal. **C**(left): latent space using GTM initialised from the LE embedding ($K = 40 \times 40$ grid, $F = 100$ RBFs of unit width wrt the grid constant). **C**(middle)/**C**(right): latent space using GPLVM without/with back constraints (from Lawrence and Quiñonero-Candela, 2006); note that, unlike the contours in panels **A**–**B**, the greyscales do not represent $p(\mathbf{x}|\mathbf{y})$.