Spectral Dimensionality Reduction via Maximum Entropy

Neil D. Lawrence
Sheffield Institute for Translational Neuroscience and Department of Computer Science,
University of Sheffield

Abstract

We introduce a new perspective on spectral dimensionality reduction which views these methods as Gaussian random fields (GRFs). Our unifying perspective is based on the maximum entropy principle which is in turn inspired by maximum variance unfolding. The resulting probabilistic models are based on GRFs. The resulting model is a nonlinear generalization of principal component analysis. We show that parameter fitting in the locally linear embedding is approximate maximum likelihood in these models. We directly maximize the likelihood and show results that are competitive with the leading spectral approaches on a robot navigation visualization and a human motion capture data set.

1 Introduction

Spectral approaches to dimensionality reduction involve taking a data set containing n points and forming a matrix of size $n \times n$ from which eigenvectors are extracted to give a representation of the data in a low dimensional space. Several spectral methods have become popular in the machine learning community including isomap [Tenenbaum et al., 2000], locally linear embeddings [LLE, Roweis and Saul, 2000], Laplacian eigenmaps [LE, Belkin and Niyogi, 2003] and maximum variance unfolding [MVU, Weinberger et al., 2004]. These approaches [and kernel PCA, Schölkopf et al., 1998] are closely related to classical multidimensional scaling [CMDS, Mardia et al., 1979].

In classical multidimensional scaling an $n \times n$ distance matrix is converted to a similarity matrix and visualized through its principal eigenvectors. Viewed from the perspective of CMDS the main difference between the spectral approaches developed in the machine learning community is in the distance matrices they (perhaps implicitly) proscribe.

In this paper we introduce a probabilistic approach to constructing this distance matrix: maximum entropy unfolding (MEU). We describe how isomap, LLE, LE and MVU are related to MEU using the unifying perspective of Gaussian random fields and CMDS.

The parameters of the model are fitted through maximum likelihood in a Gaussian random field (GRF). The random field specifies dependencies between *data points* rather than the more typical approach which specifies dependencies between *data features*.

Our method is based on maximum likelihood. Normally maximum likelihood algorithms specify a distribution which factorizes over the data points (each data point is independent given the model parameters). Our model's distribution factorizes over the features (each feature from the data set is independent given the model parameters). This means that maximum likelihood in our model is consistent as the number of features increases, $p \to \infty$ rather than the number of data points. This leads to a *blessing* of dimensionality where the parameters are better determined as the number of features increases.

In Section 2 we derive our model through using standard assumptions from the field of dimensionality reduction and the maximum entropy principle [Jaynes, 1986]. We then relate the model to other popular spectral approaches to dimensionality reduction. We then consider how the parameters of the model can be fitted through maximum likelihood. Finally we demonstrate the model (with comparisons) on two real world data sets. First though, we will briefly review classical multidimensional scaling which provides the general framework through which these approaches can be related [see also Ham et al., 2004].

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

1.1 Classical Multidimensional Scaling

Given an $n \times n$ matrix of similarities, $\mathbf{K}$, or dissimilarities, $\mathbf{D}$, between a set of data points, multidimensional scaling considers the problem of how to represent these data in a low dimensional space. One way of doing this is to associate a q dimensional latent vector with each data point, $\mathbf{y}_{i,:}$, and define a set of dissimilarities between each latent point, $\delta_{i,j} = \|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\|_2^2$, to give a matrix $\mathbf{\Delta}$. Here we have specified the squared distance between each point as the dissimilarity.[1]

If the error for the latent representation is then taken to be the sum of absolute values between the dissimilarity matrix entries,

$$E(\mathbf{X}) = \sum_{i=1}^{n}\sum_{j=1}^{i-1} \|d_{i,j} - \delta_{i,j}\|_1 , \qquad (1)$$

and we assume that the data dissimilarities also represent a squared Euclidean distance matrix (perhaps computed in some high, maybe infinite, dimensional space) then the best *linear* dimensionality reduction is given by the following procedure [Mardia et al., 1979, pg 400],

1. Convert the matrix of dissimilarities to a matrix of similarities by taking $\mathbf{B} = -\frac{1}{2}\mathbf{HDH}$ where $\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^\top$ is a centering matrix.
2. Extract the first q principal eigenvectors of $\mathbf{B}$.
3. Setting $\mathbf{X}$ to these principal eigenvectors (appropriately scaled) gives a global minimum for the error function (1).

2 Maximum Entropy Unfolding

Classical multidimensional scaling provides only a linear transformation of space in which the squared distances are expressed. The novelty of modern spectral approaches is distance computation in spaces which are nonlinearly related to the data. This gives a nonlinear algorithm. This can be seen clearest for kernel PCA. In kernel PCA the squared distances are embedded in a Hilbert space and related to the original data through a kernel function,

$$d_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{i,:}) - 2k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:}) + k(\mathbf{y}_{j,:}, \mathbf{y}_{j,:}) \qquad (2)$$

which is recognized as the squared distance in "feature space" [see Ham et al., 2004]. In CMDS this is known as the *standard transformation* between a similarity and distance [Mardia et al., 1979]. Kernel PCA (KPCA) recovers an $\mathbf{x}_{i,:}$ for each data point and a mapping from the data space to the $\mathbf{X}$ space. Under the CMDS procedure the eigenvalue problem is performed on the centered kernel matrix,

$$\mathbf{B} = \mathbf{HKH},$$

where $\mathbf{K} = [k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:})]_{i,j}$. This matches the KPCA algorithm [Schölkopf et al., 1998][2]. However, for the commonly used exponentiated quadratic kernel,

$$k(y_{i,:}, y_{j,:}) = \exp(-\gamma \|\mathbf{y}_{i,:} - \mathbf{y}_{j,:}\|_2^2),$$

KPCA actually *expands* the feature space rather than reducing the dimension [see Weinberger et al., 2004, for some examples of this].

The observation that KPCA expands the feature space motivated the maximum variance unfolding algorithm [MVU, Weinberger et al., 2004]. The idea in MVU is to learn a kernel matrix that will allow for dimensionality reduction. This is achieved by only considering *local relationships* in the data. A set of neighbors is defined (e.g. by k-nearest neighbors) and only distances between neighboring data points are respected. These distances are specified as constraints, and the other elements of the kernel matrix are filled in by maximizing the trace of the kernel matrix, $\mathrm{tr}(\mathbf{K})$, i.e. the *total variance* of the data in feature space, while respecting the distance constraints and keeping the resulting matrix centered. Maximizing $\mathrm{tr}(\mathbf{K})$ in turn maximizes the interpoint squared distances for all points that are unconnected in the neighborhood graph, thereby unravelling the manifold.

In this paper we consider an alternative maximum entropy formalism of this problem. Since entropy is related to variance, we might expect a similar result in the quality of the resulting algorithm, but since maximum entropy also provides a probability distribution we should also obtain a probabilistic model with all the associated advantages (dealing with missing data, extensions to mixture models, fitting parameters by Bayesian methods, combining with other probabilistic models). Importantly, our interpretation will also enable us to relate other well known spectral techniques to our algorithm as they each turn out to approximate maximum entropy unfolding in some way.

In the maximum entropy formalism [see e.g. Jaynes, 1986], we maximise the entropy of a distribution subject to constraints on the moments of that distribution. Here those constraints will be the expectations of the squared distances between two data points sampled from the model. Constraints will only apply to

[1] It is more usual to specify the distance directly as the dissimilarity, however, for our purposes it will be more convenient to work with squared distances

[2] For stationary kernels, kernel PCA also has an interpretation as a particular form of *metric* multidimensional scaling, see Williams [2001] for details.

points that are defined to be "neighbors". For continuous data, the maximum entropy can only be defined relative to a base distribution [Jaynes, 1986]. We follow a common choice and take the base distribution to be a spherical Gaussian with covariance $\gamma^{-1}\mathbf{I}$. The maximum entropy distribution is then given by

$$p(\mathbf{Y}) \propto \exp\left(-\frac{1}{2}\text{tr}\left(\gamma\mathbf{Y}\mathbf{Y}^\top\right)\right) \times \exp\left(-\frac{1}{2}\sum_i \sum_{j\in\mathcal{N}(i)} \lambda_{i,j} d_{i,j}\right),$$

where $\mathcal{N}(i)$ represents the set of neighbors of data point i (a point cannot be its own neighbor), and $\mathbf{Y} = [\mathbf{y}_{1,:}, \ldots, \mathbf{y}_{n,:}]^\top \in \Re^{n\times p}$ is a *design matrix* containing our data. Note that we have introduced a factor of $-1/2$ in front of our Lagrange multipliers, $\{\lambda_{i,j}\}$, for later notational convenience. We now define the symmetric matrix $\mathbf{\Lambda}$ to contain $\lambda_{i,j}$ if i is a neighbor of j and zero otherwise. This allows us to write the distribution as

$$p(\mathbf{Y}) \propto \exp\left(-\frac{1}{2}\text{tr}\left(\gamma\mathbf{Y}\mathbf{Y}^\top\right) - \frac{1}{4}\text{tr}\left(\mathbf{\Lambda}\mathbf{D}\right)\right).$$

We introduce a matrix $\mathbf{L}$ which is symmetric and constrained to have a null space in the constant vector, $\mathbf{L}\mathbf{1} = \mathbf{0}$. Its off diagonal elements are given by $-\mathbf{\Lambda}$ and its diagonal elements are given by

$$\ell_{i,i} = \sum_{j\in\mathcal{N}(i)} \lambda_{i,j}$$

to enforce the null space constraint. This enables us to write

$$p(\mathbf{Y}) = \frac{|\mathbf{L} + \gamma\mathbf{I}|^{\frac{1}{2}}}{(2\pi)^{\frac{np}{2}}} \exp\left(-\frac{1}{2}\text{tr}\left((\mathbf{L} + \gamma\mathbf{I})\mathbf{Y}\mathbf{Y}^\top\right)\right). \quad (3)$$

We arrive here because the distance matrix is zero along the diagonal. This allows us to set the diagonal elements of $\mathbf{L}$ as we please without changing the value of $\text{tr}\left(\mathbf{L}\mathbf{D}\right)$. Our choice to set them as the sum of the off diagonals gives the matrix a null space in the constant vector enabling us to use the fact that

$$\mathbf{D} = \mathbf{1}\text{diag}\left(\mathbf{Y}\mathbf{Y}^\top\right)^\top - 2\mathbf{Y}\mathbf{Y}^\top + \text{diag}\left(\mathbf{Y}\mathbf{Y}^\top\right)\mathbf{1}^\top$$

(where the operator $\text{diag}\left(\mathbf{A}\right)$ forms a vector from the diagonal of $\mathbf{A}$) to write

$$-\text{tr}\left(\mathbf{\Lambda}\mathbf{D}\right) = \text{tr}\left(\mathbf{L}\mathbf{D}\right) = -2\text{tr}\left(\mathbf{L}\mathbf{Y}\mathbf{Y}^\top\right),$$

which in turn allows us to recover (3). This probability distribution is a *Gaussian random field*. It can be written as

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{|\mathbf{L} + \gamma\mathbf{I}|^{\frac{1}{2}}}{(2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^\top(\mathbf{L} + \gamma\mathbf{I})\mathbf{y}_{:,j}\right),$$

which clarifies the fact that the GRF is being expressed independently across data *features* (each vector $\mathbf{y}_{:,j}$ contains the jth feature from all data points). This contrasts with most applications of Gaussian models that are applied independently across data *points*. Notable exceptions include [Zhu et al., 2003, Lawrence, 2004, 2005, Kemp and Tenenbaum, 2008]. As with all maximum entropy methods, maximum likelihood for this model is equivalent to finding the correct setting of the Lagrange multipliers.

2.1 Maximum Likelihood and Blessing of Dimensionality

When we consider maximum likelihood consistency arguments [see e.g. Wasserman, 2003, pg 126] we see that this model isn't consistent as we increase the number of data points, n, for fixed data dimensionality, p, but it *is* consistent as we increase data dimensionality, p, for a fixed number of data points n. The number of parameters will increase as we increase the number of data (each datum requires K parameters to connect with K neighbors). However, as we increase features there is no corresponding increase in parameters. In other words as the number of features increases there is a clear *blessing of dimensionality*.

There is perhaps a deeper lesson here in terms of how we should interpret such consistency results. In the "sampled points" formalism, as we increase the number of data points, the parameters become better determined. In the "sampled features" formalism, as we increase the number of features, the parameters become better determined. However, for consistency results to hold, the class of models we consider must include the actual model that generated the data. If we believe that "Essentially, all models are wrong, but some are useful" [Box and Draper, 1987, pg 424] we may feel that encapsulating the right model within our class is a practical impossibility. Given that, we might pragmatically bias our choice somewhat to ensure utility of the resulting model. From this perspective, in the large p small n domain, the "sampled features" formalism is attractive. A practical issue can arise though. If we wish to compute the likelihood of an out of sample data-point, we must first estimate the parameters associated with that new data point. This can be problematic. Of course, for the sampled-points formalism the same problem exists when you wish to include an out of sample data-feature in your model.

2.1.1 Parameter Gradients

We can find the parameters $\mathbf{\Lambda}$ through maximum likelihood on this distribution. Some algebra shows that

the gradient of each Lagrange multiplier is given by,

$$\frac{\mathrm{d}\log p(\mathbf{Y})}{\mathrm{d}\lambda_{i,j}} = \frac{1}{2}\left\langle d_{i,j}\right\rangle_{p(\mathbf{Y})} - \frac{1}{2}d_{i,j},$$

where $\left\langle\right\rangle_{p(\cdot)}$ represents an expectation under the distribution $p(\cdot)$. This result is expected given our maximum entropy formulation: the Lagrange multipliers have a gradient of zero when the constraints are satisfied. To compute gradients we need the expectation of the squared distance given by

$$\left\langle d_{i,j}\right\rangle = \left\langle y_{i,:}^\top y_{i,:}\right\rangle - 2\left\langle y_{i,:}^\top y_{j,:}\right\rangle + \left\langle y_{j,:}^\top y_{j,:}\right\rangle,$$

which we can compute directly from the covariance matrix of the GRF, $\mathbf{K} = (\mathbf{L} + \gamma\mathbf{I})^{-1}$,

$$\left\langle d_{i,j}\right\rangle = p\left(k_{i,i} - 2k_{i,j} + k_{j,j}\right).$$

This is immediately recognized as a scaled version of the standard transformation between distances and similarities (see (2)). This relationship arises naturally in the probablistic model. Every GRF has an implied associated distance matrix. It is this matrix that is being used in CMDS. The machine learning audience might interpret this as the relationship between distances in "feature space" and the kernel function. Note though that here (and also in MVU) each individual element of the kernel matrix *cannot* be represented only as a function of the corresponding two data points (i.e. we can't represent them as $k_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:})$). Given this we feel it is more correct to think of this matrix as a covariance matrix induced by our specification of the random field rather than a true Mercer kernel.

If K neighbors are used for each data point there are $O(Kn)$ parameters in the model, so the model is non-parametric in the sense that the number of parameters increases with the number of data. For the parameters to be well determined we require a large number of features, p, for each data point, otherwise we would need to look to regularize the model. This implies that the model is well primed for the so-called "large p small n domain".

Once the maximum likelihood solution is recovered the data can be visualized, as for MVU and kernel PCA, by looking at the eigenvectors of the centered covariance matrix $\mathbf{HKH}$. We call this algorithm maximum entropy unfolding (MEU).

Note that the entropy of a Gaussian is related to the determinant of the covariance matrix. The determinant of the covariance can be expressed as the sum of the log of the eigenvalues of $\mathbf{K}$, $\log|\mathbf{K}| = \sum_{i=1}^{n}\log\lambda_i$. In contrast MVU looks to maximize the trace of the covariance matrix $\mathrm{tr}(\mathbf{K}) = \sum_{i=1}^{n}\lambda_i$, subject to distance constraints.

When optimizing in MVU and MEU we need to ensure that the covariance matrix is positive definite. In MVU this is ensured through a semidefinite program. In MEU the objective is not linear in $\mathbf{K}$ so we need to use other approaches. Possibilities include exploiting the fact that if the Lagrange multipliers are constrained to be positive the system is "attractive" and this guarantees a valid covariance [see e.g. Koller and Friedman, 2009, pg 255]. Although now (as in a suggested variant of the MVU) the distance constraints would be inequalities. Another alternative would be to constrain $\mathbf{L}$ to be diagonally dominant through adjusting γ. We will also consider an alternative approach in Section 2.3.

Finally, we note that for MEU and MVU, as we increase the neighborhood size to $K = n - 1$, we recover principal component analysis. In this limit all expected squared distances, implied by the GRF model, are required to match the observed squared distances and $\mathbf{L}$ becomes non-sparse. Classical multidimensional scaling on the resulting squared distance matrix is known as principal coordinate analysis and is equivalent to principal component analysis [see Mardia et al., 1979].

2.2 Relation to Laplacian Eigenmaps

The relationship to Laplacian eigenmaps [Belkin and Niyogi, 2003] is starting to become clear. In Laplacian eigenmaps a graph Laplacian is specified across the data points. This Laplacian has exactly the same form as our matrix $\mathbf{L}$, which we will henceforth refer to as the Laplacian. The parameters of the Laplacian are set either as constant or according to the distance between two points. The smallest eigenvectors of this Laplacian are then used for visualizing the data (disregarding the eigenvector associated with the null space). From the eigendecomposition of $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ it is easy to show that $\mathbf{L} = \mathbf{U}\left(\mathbf{\Lambda}^{-1} - \gamma\mathbf{I}\right)\mathbf{U}^\top$ is the eigendecomposition of $\mathbf{L}$. So in other words, the principal eigenvalues of $\mathbf{K}$ will be the smallest eigenvalues of $\mathbf{L}$. The very smallest eigenvalue of $\mathbf{L}$ is zero and associated with the constant eigenvector. However, in CMDS this would be removed by the centering operation and in LE it is discarded. So we see that once the parameters of the Laplacian have been set CMDS is being performed to recover the latent variables in Laplacian eigenmaps. However, since the moment constraints are not being imposed in Laplacian eigenmaps, the squared distance matrix used for CMDS will not preserve the interneighbor distances as it will for MVU and MEU. In fact since the covariance matrix is never explicitly computed it is not possible to make specific statements about what these distances will be in general. However, LE gains signifi-

cant computational advantage by not representing the covariance matrix explicitly. No matrix inverses are required in the algorithm and the resulting eigenvalue problem is sparse. This means that LE can be applied to much larger data sets than would be possible for MEU or MVU.

2.3 Relation to Locally Linear Embedding

When introducing MEU we discussed how it is necessary to constrain the Laplacian matrix to be positive semidefinite. A further way of doing this is to factorize the Laplacian as

$$\mathbf{L} = \mathbf{M}\mathbf{M}^\top$$

where $\mathbf{M}$ is non-symmetric. If $\mathbf{M}$ is constrained so that $\mathbf{M}^\top\mathbf{1} = \mathbf{0}$ then we will also have $\mathbf{L}\mathbf{1} = \mathbf{0}$. We can achieve this constraint by setting the diagonal elements $m_{i,i} = -\sum_{j\in\mathcal{N}(i)} m_{j,i}$. Then if we force $m_{j,i} = 0$ if $j \notin \mathcal{N}(i)$ we will have a Laplacian matrix which is positive semidefinite without need for any further constraint on $\mathbf{M}$. Note that the sparsity pattern of $\mathbf{L}$ will be different from the pattern of $\mathbf{M}$. The entry for $\ell_{i,j}$ will only be zero if there are no shared neighbors between i and j.

In the locally linear embedding [Roweis and Saul, 2000], an alternative approach to dimensionality reduction is taken. The idea is to first derive a set of weights that allow each data point to be reconstructed from its neighbors. This involves a minimization of the form,

$$E(\mathbf{W}) = \sum_{j=1}^{n} \left\| \mathbf{y}_{i,:} - \sum_{j\in\mathcal{N}(i)} w_{j,i}\mathbf{y}_{j,:} \right\|_2^2. \tag{4}$$

Once the matrix of weights is found, the next step is to find a low dimensional embedding of the data, $\mathbf{X}$, which best respects the linear relationships defined by $\mathbf{W}$. This turns out to be computed through an eigenvalue problem on the matrix $(\mathbf{I}-\mathbf{W})(\mathbf{I}-\mathbf{W})^\top$, where $\mathbf{W}$ is a matrix with elements $w_{i,j}$ if i and j are neighbors and zero otherwise.

Locally linear embeddings turn out to be a specific case of the MEU random field model where

1. The diagonal sums, $m_{i,i}$, are further constrained to unity.
2. The parameters of the model are optimized by maximizing the pseudolikelihood of the resulting GRF.

To see the first point, we note that if the diagonals were constrained to unity then we can write $\mathbf{M} = \mathbf{I} - \mathbf{W}$. Here the sparsity pattern of $\mathbf{W}$ matches $\mathbf{M}$, apart from the diagonal which is set to zero. These constraints mean that $(\mathbf{I}-\mathbf{W})^\top\mathbf{1} = \mathbf{0}$. LLE proscribes that the smallest eigenvectors of $(\mathbf{I}-\mathbf{W})(\mathbf{I}-\mathbf{W})^\top = \mathbf{M}\mathbf{M}^\top = \mathbf{L}$ are used with the constant eigenvector associated with the eigenvalue of 0 being discarded. As for the Laplacian eigenmaps this is equivalent to CMDS on the Gaussian random field described by $\mathbf{L}$.

For the second point the pseudolikelihood approximation [see e.g. Koller and Friedman, 2009, pg 970] to the joint density in a graphical model is the product of the conditional densities:

$$p(\mathbf{Y}) \approx \prod_{i=1}^{n} p(\mathbf{y}_{i,:}|\mathbf{Y}_{\setminus i}),$$

where $\mathbf{Y}_{\setminus i}$ represents all that data other than the ith point. The true joint likelihood is proportional to the product of conditional densities, but it requires renormalization. In pseudolikelihood this normalization is ignored. To see how this decomposition applies we first factorize the model by noting that

$$\operatorname{tr}\left(\mathbf{Y}\mathbf{Y}^\top\mathbf{M}\mathbf{M}^\top\right) = \sum_{i=1}^{n} \mathbf{m}_{:,i}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{m}_{:,i}$$

so for the MEU model we have

$$p(\mathbf{Y}) \propto \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\mathbf{m}_{i,:}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{m}_{i,:}\right).$$

This provides the necessary factorization for the conditionals which can be rewritten as

$$p(\mathbf{y}_{i,:}|\mathbf{Y}_{\setminus i}) \propto \exp\left(-\frac{m_{i,i}^2}{2}\left\|\mathbf{y}_{i,:} - \sum_{j\in\mathcal{N}(i)}\frac{w_{j,i}}{m_{i,i}}\mathbf{y}_{j,:}\right\|_2^2\right).$$

This shows that optimizing the log pseudolikelihood is equivalent to optimizing

$$\log p(\mathbf{Y}) \approx \sum_{i=1}^{n} \log p(\mathbf{y}_{i,:}|\mathbf{Y}_{\setminus i})$$

which is equivalent to solving n independent regression problems with a constraint on the regression weights that they sum to one. This is exactly the optimization suggested in (4). Although for MEU we see that the constraint arises because the regression weights are constrained to be $w_{j,i}/m_{i,i}$ and $m_{i,i} = \sum_{j\in\mathcal{N}(i)} w_{j,i}$. In LLE a further constraint is placed that $m_{i,i} = 1$ which implies none of these regression problems should be solved to a greater precision than another. However, the algorithm also works if this further constraint isn't imposed.

Locally linear embeddings are therefore an approximation to maximum likelihood on the Gaussian random

field. They have a neat way of constraining the Laplacian to be positive semidefinite by assuming a factorized form. The pseudolikelihood also allows for relatively quick parameter estimation by ignoring the partition function from the actual likelihood. This again removes the need to invert to recover the covariance matrix and means that LLE can be applied to larger data sets than MEU or MVU. However, the sparsity pattern in the Laplacian for LLE will not match that used in the Laplacian for the other algorithms due to the factorized representation.

LLE is motivated by considering local linear embeddings of the data, although interestingly, as we increase the neighborhood size to $K = n - 1$ we do not recover PCA, which is known to be the optimal linear embedding of the data under linear Gaussian constraints. The fact that LLE is optimizing the pseudolikelihood makes it clear why this is the case. In contrast the MEU algorithm, which LLE approximates, does recover PCA when $K = n - 1$.

2.4 Relation to Isomap

Isomap more directly follows the CMDS framework. In isomap [Tenenbaum et al., 2000] a sparse graph of distances is created between all points considered to be neighbors. This graph is then filled in for all non-neighboring points by finding the shortest distance between any two neighboring points in the graph (along the edges specified by the neighbors). The resulting matrix is then element-wise squared to give a matrix of square distances which is then processed in the usual manner (centering and multiplying by -0.5) to provide a similarity matrix for multidimensional scaling. Compare this to the situation for MVU and MEU. Both MVU and MEU can be thought of as starting with a sparse graph of (squared) distances. The other distances are then filled in by either maximizing the trace of the associated covariance or maximizing the entropy. Importantly, though, the interneighbor distances in this graph are preserved (through constraints imposed by Lagrange multipliers) just like in isomap. For both MVU and MEU the covariance matrix, $\mathbf{K}$, is guaranteed positive semidefinite because the distances are implied by an underlying covariance matrix that is constrained positive definite. For isomap the shortest path algorithm is effectively approximating the distances between non-neighboring points. This can lead to an implied covariance matrix which has negative eigenvalues [see Weinberger et al., 2004]. The algorithm is still slower than LLE and LE because it requires a dense eigenvalue problem and the application of a shortest path algorithm to the graph provided by the neighbors.

3 Experiments

The advantage of our approach is mainly in the unifying perspective it gives and its potential to exploit the characteristics of the probabilistic formulation to explore extensions based on missing data, Bayesian formulations etc. However, for illustrative purposes we conclude with a short experimental section.

For our experiments we consider two real world data sets. Code to recreate all our experiments is available online. We applied each of the spectral methods we have reviewed along with MEU using positive constraints on the Lagrange multipliers (denoted MEU). The value of γ was always kept fixed to 10^{-4}. To evaluate the quality of our embeddings we follow the suggestion of Harmeling [Harmeling, 2007] and use the GPLVM likelihood [Lawrence, 2004, 2005]. The higher the likelihood the better the embedding. Harmeling conducted exhaustive tests over different manifold types (with known ground truth) and found the GPLVM likelihood was the best indicator of the manifold quality amoungst all the measures he tried. Our first data set consists of human motion capture data.

3.1 Motion Capture Data

The data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run. This leads to a 102 dimensional data set containing 55 frames of motion capture. The subject begins the motion from stationary and takes approximately three strides of run. We hope to see this structure in the visualization: a starting position followed by a series of loops. The data was made available by Ohio State University. The data is characterized by a cyclic pattern during the strides of run. However, the angle of inclination during the run changes so there are slight differences for each cycle. The data is very low noise, as the motion capture rig is designed to extract the point locations of the subject to a high precision.

The two dominant eigenvectors are visualized in Figure 1(a)-(e) and the quality of the visualizations under the GPLVM likelihood is given in Figure 1(f).

There is a clear difference in quality between the methods that constrain local distances (MVU, isomap and MEU) which are much better under the score than those that don't (LE and LLE).

3.2 Robot Navigation Example

The second data set we use is a series of recordings from a robot as it traces a square path in a building. The robot records the strength of WiFi signals in an attempt to localize its position [see Ferris et al.,

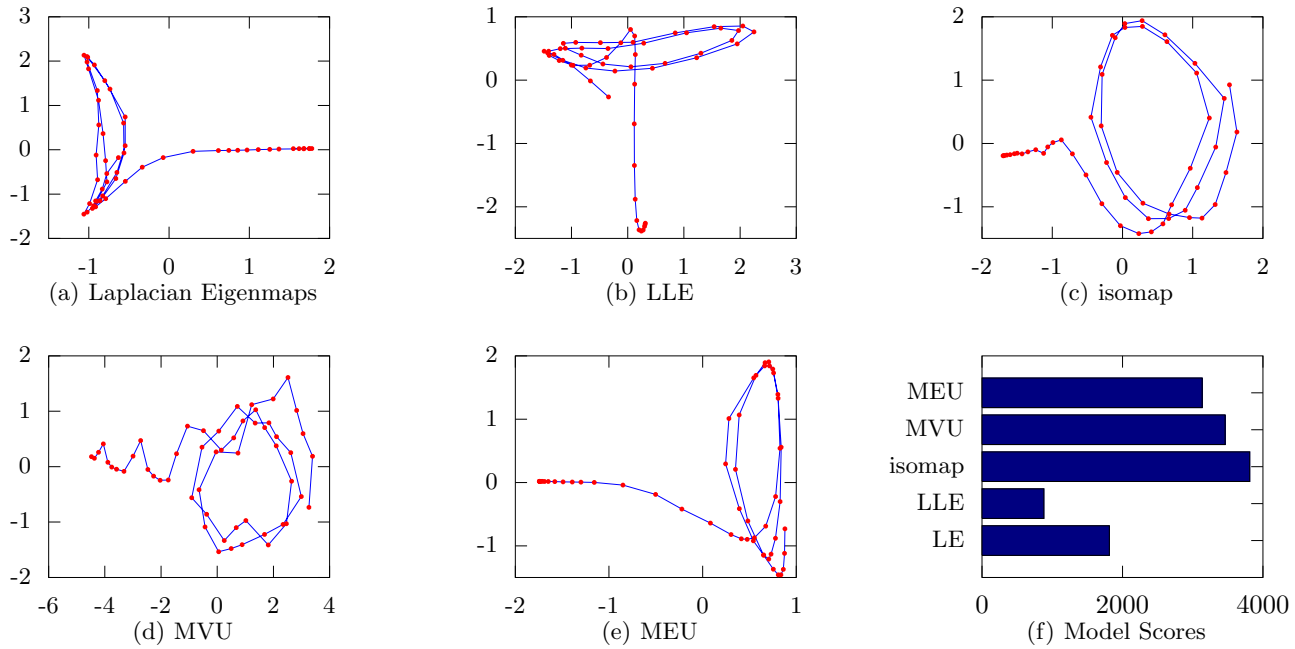

Figure 1: Motion capture data visualized in two dimensions for each algorithm using 6 nearest neighbors. Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

2007, for an application]. Since the robot moves only in two dimensions, the inherent dimensionality of the data should be two: the reduced dimensional space should reflect the robot's movement. The WiFi signals are noisier than the motion capture data, so it makes an interesting contrast. The robot completes a single circuit after entering from a separate corridor, so it is expected to exhibit "loop closure" in the resulting map. The data consists of 215 frames of measurement, each frame consists of the WiFi signal strength of 30 access points.

The results for the range of spectral approaches are shown in Figure 2(a)-(e) with the quality of the methods scored in Figure 2(f). Both in the visualizations and in the GPLVM scores we see a clear difference in quality for the methods that preserve local distances (i.e. again isomap, MVU and MEU are better than LLE and LE).

4 Discussion and Conclusions

We have introduced a new perspective on spectral dimensionality reduction algorithms based around maximum entropy. Our starting point was the maximum variance unfolding and our end point was a novel approach to dimensionality reduction based on Gaussian random fields. We hope that this new perspective on dimensionality reduction will encourage new strands of research at the interface between these areas.

One feature that stands out from our unifying perspective is the three separate stages used in existing spectral dimensionality algorithms.

1. A neighborhood between data points is selected. Normally k-nearest neighbors or similar algorithms are used.

2. Interpoint distances between neighbors are fed to the algorithms which provide a similarity matrix. The way the entries in the similarity matrix are computed is the main difference between the algorithms.

3. The relationship between points in the similarity matrix is visualized using the eigenvectors of the similarity matrix.

Our unifying perspective shows that actually each of these steps is somewhat orthogonal. The neighborhood relations need not come from nearest neighbors, for example we could use sparse graph fitting approaches to derive the neighborhood. The main difference between the different approaches to spectral dimensionality reduction is how the entries of the similarity matrix are determined. Maximum variance unfolding looks to maximize the trace under the distance constraints from the neighbours. Our new algorithms maximize the entropy or, equivalently, the likelihood of the data. Locally linear embedding maximizes an approximation to our likelihood. Laplacian eigenmaps

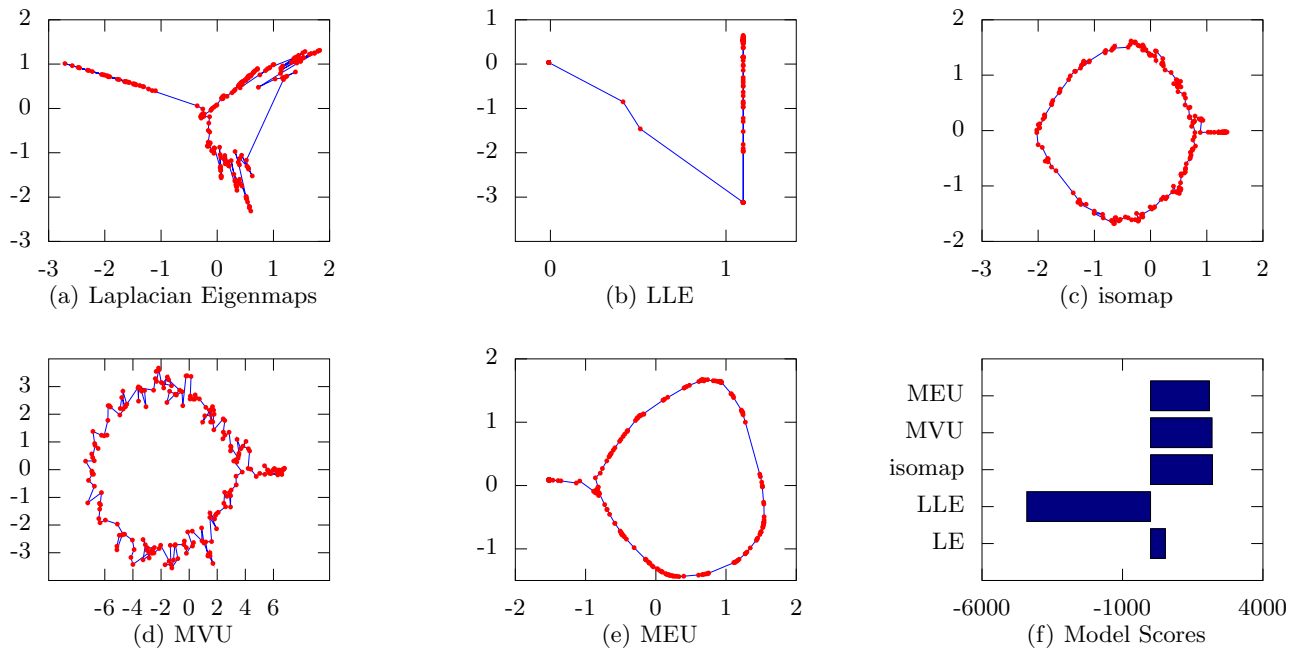

Figure 2: The robot WiFi navigation data for different algorithms with six neighbors used. Some models struggle to captured the loop structure (perhaps because of the higher level of noise). Most models also show the noise present in the data WiFi signals.

parameterize the inverse similarity through appealing to physical analogies. Finally, isomap uses shortest path algorithms to compute interpoint distances and centres the resulting matrix to give the similarities.

The final step of the algorithm attempts to visualize the similarity matrices using their eigenvectors. However, it simply makes use of one possible objective function to perform this visualization. Considering that underlying the similarity matrix, $\mathbf{K}$, is a sparse Laplacian matrix, $\mathbf{L}$, which represents a Gaussian-Markov random field, we can see this final step as visualizing that random field. There are many potential ways to visualize that field and the eigenvectors of the precision is just one of them. In fact, there is an entire field of graph visualization proposing different approaches to visualizing such graphs. However, we could even choose not to visualize the resulting graph. It may be that the structure of the graph is of interest in itself. Work in human cognition by Kemp and Tenenbaum [2008] has sought to fit Gaussian graphical models to data in natural structures such as trees, chains and rings. Visualization of such graphs through reduced dimensional spaces is only likely to be appropriate in some cases, for example planar structures. For this model only the first two steps are necessary.

One advantage to conflating the three steps we've identified is the possibility to speed up the complete algorithm. For example, conflating the second and third step allows us to speed up algorithms through never explicitly computing the similarity matrix. Using the fact that the principal eigenvectors of the similarity are the minor eigenvalues of the Laplacian and exploiting fast eigensolvers that act on sparse matrices very large data sets can be addressed. However, we still can understand the algorithm from the unifying perspective while exploiting the computational advantages offered by this neat shortcut.

There are similarities between maximum entropy unfolding and the GPLVM [Lawrence, 2004, 2005]. Both specify a similar Gaussian density over the training data. A Gauss Markov random field can easily be specified by a Gaussian process through appropriate covariance functions. The O-U covariance function in a one dimensional latent space $k(x, x') = \exp(-\|x - x\prime\|_1)$ gives a sparse inverse with only neighbors connected. In the GPLVM the neigborhood is optimized as part of the training procedure, for MEU it is pre-specified.

Notes

The plots in this document were generated using MATweave. Code was run using Octave version 3.2.4 on the architecture `x86_64-pc-linux-gnu` . They were generated on 05/01/2011 .

References

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

George E. P. Box and Norman R. Draper. *Empirical Model-Building and Response Surfaces*. John Wiley and Sons, 1987. ISBN 0471810339.

Brian D. Ferris, Dieter Fox, and Neil D. Lawrence. WiFi-SLAM using Gaussian process latent variable models. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2480–2485, 2007.

Russell Greiner and Dale Schuurmans, editors. *Proceedings of the International Conference in Machine Learning*, volume 21, 2004. Omnipress.

Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of dimensionality reduction of manifolds. In Greiner and Schuurmans [2004].

Stefan Harmeling. Exploring model selection techniques for nonlinear dimensionality reduction. Technical Report EDI-INF-RR-0960, University of Edinburgh, 2007.

Edwin T. Jaynes. Bayesian methods: General background. In J. H. Justice, editor, *Maximum Entropy and Bayesian Methods in Applied Statistics*, pages 1–25. Cambridge University Press, 1986.

Charles Kemp and Joshua B. Tenenbaum. The discovery of structural form. *Proc. Natl. Acad. Sci. USA*, 105(31), 2008.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. ISBN 978-0-262-01319-2.

Neil D. Lawrence. Gaussian process models for visualisation of high dimensional data. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 329–336, Cambridge, MA, 2004. MIT Press.

Neil D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 11 2005.

Kantilal V. Mardia, John T. Kent, and John M. Bibby. *Multivariate analysis*. Academic Press, London, 1979. ISBN 0-12-471252-5.

Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. doi: 10.1126/science.290.5500.2323.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. doi: 10.1162/089976698300017467.

Joshua B. Tenenbaum, Virginia de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319–2323, 2000. doi: 10.1126/science.290.5500.2319.

Larry A. Wasserman. *All of Statistics*. Springer-Verlag, New York, 2003. ISBN 9780387402727.

Kilian Q. Weinberger, Fei Sha, and Lawrence K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In Greiner and Schuurmans [2004], pages 839–846.

Christopher K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 675–681, Cambridge, MA, 2001. MIT Press.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Semi-supervised learning: From Gaussian fields to Gaussian processes. Technical Report CMU-CS-03-175, Carnegie Mellon University, 2003.