A Probabilistic Perspective on Spectral Dimensionality Reduction

Neil D. Lawrence

Departments of Neuro- and Computer Science, University of Sheffield, U.K.

20th October 2010

Outline

Motivation

Maximum Entropy Unfolding

Experiments

Discussion and Conclusions

Outline

Motivation

Maximum Entropy Unfolding

Experiments

Discussion and Conclusions

High Dimensional Data

- Represent objects for processing by a series of features.
- Number of features increases with complexity of representation. E.g.:
  1. the characteristics of a customer in a database;
  2. the pixel intensities in an image;
  3. a time series of angles associated with data captured from human motion for animation;
  4. the energy at different frequencies (or across the cepstrum) as a time series for interpreting speech;
  5. the frequencies of given words as they appear in a set of documents;
  6. the level of expression of thousands of genes, across a time series, or for different diseases.

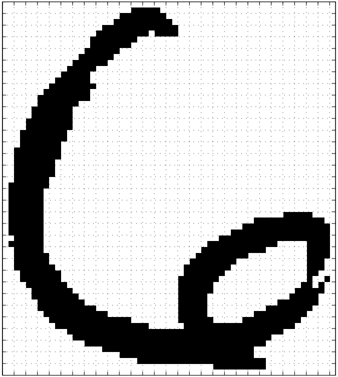
High Dimensional Representation

- As complexity increases so does number of features.
- This is high dimensional data.
- Example: handwritten digit 6.

Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

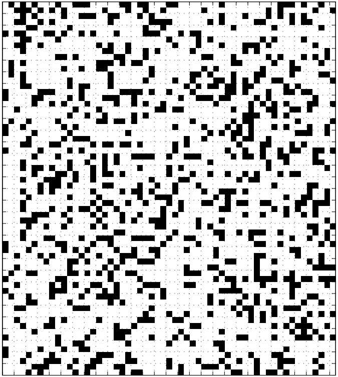
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

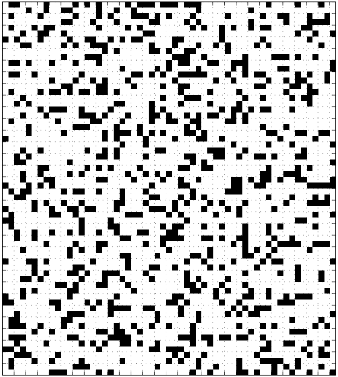
USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

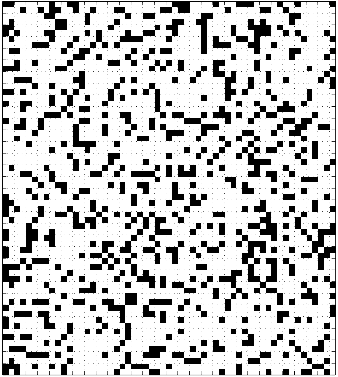


- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!

Motivation for Non-Linear Dimensionality Reduction

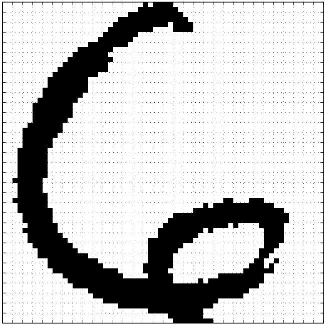
USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



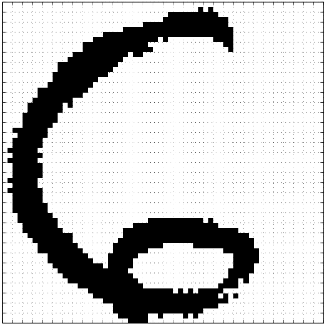
Simple Model of Digit

Rotate a 'Prototype'



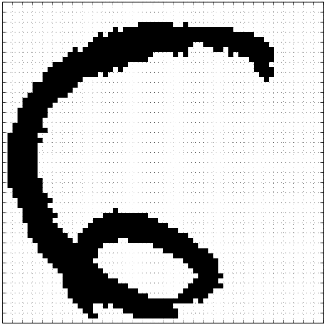
Simple Model of Digit

Rotate a 'Prototype'



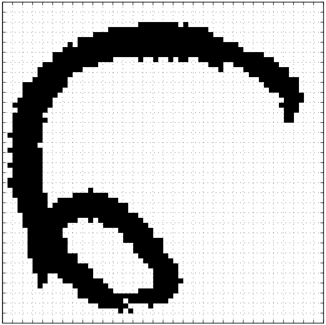
Simple Model of Digit

Rotate a 'Prototype'



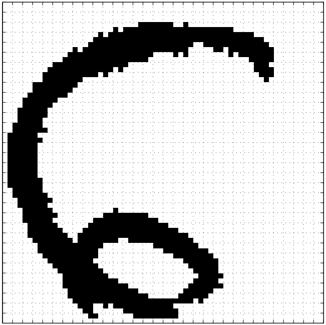
Simple Model of Digit

Rotate a 'Prototype'



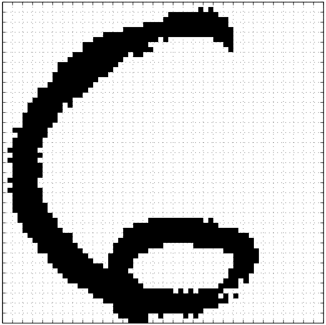
Simple Model of Digit

Rotate a 'Prototype'



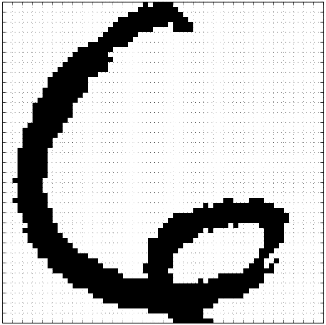
Simple Model of Digit

Rotate a 'Prototype'



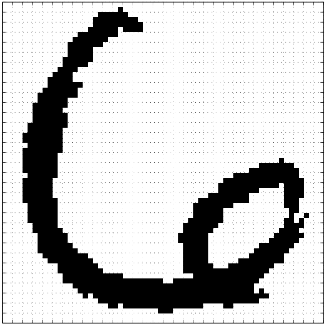
Simple Model of Digit

Rotate a 'Prototype'



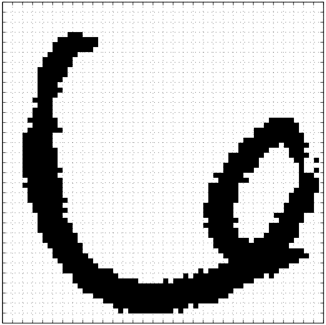
Simple Model of Digit

Rotate a 'Prototype'



Simple Model of Digit

Rotate a 'Prototype'

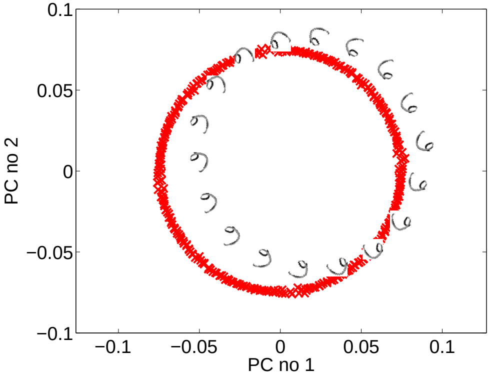


MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```

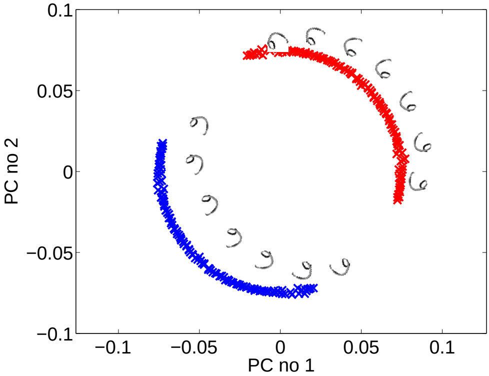
MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```



MATLAB Demo

```
demDigitsManifold([1 2], 'sixnine')
```



Low Dimensional Manifolds

Pure Rotation is too Simple

- In practice the data may undergo several distortions.
  - *e.g.* digits undergo 'thinning', translation and rotation.
- For data with 'structure':
  - we expect fewer distortions than dimensions;
  - we therefore expect the data to live on a lower dimensional manifold.
- Conclusion: deal with high dimensional data by looking for lower dimensional non-linear embedding.

Spectral Dimensionality Reduction

- Spectral approaches to dimensionality reduction.
  1. Take a data set containing n points and form a matrix of size $n \times n$.
  2. Extract eigenvectors and use them to give a representation of the data in a low dimensional space.
- Examples include 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; Ham et al., 2004).) are closely related to classical multidimensional scaling (CMDS, Mardia et al., 1979).

Classical Multidimensional Scaling

- Classical multidimensional scaling (CMDS)
  1. Compute an $n \times n$ distance matrix.
  2. Convert it to a similarity matrix.
  3. Visualize through principal eigenvectors.
- From the CMDS perspective main innovation in ML spectral approaches is how the (implicitly) compute the distances.

Our Contribution

- Introduce a probabilistic approach to constructing distance matrices.
- Relate isomap, LLE, LE and MVU to our approach.
- All these methods sit within a unifying perspective of *Gaussian random fields* and CMDS.

Classical Multidimensional Scaling

- Given an $n \times n$ matrix of similarities, $\mathbf{K}$, or dissimilarities, $\mathbf{D}$
- Multidimensional scaling attempts to represent points, $\mathbf{x}_{i,:}$ in a low q dimensional latent space.
- We define a dissimilarity between these points,

  $$\delta_{i,j} = \left\|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right\|_2^2$$

  giving a matrix $\boldsymbol{\Delta}$.
- **Note**: here we are using *squared* distances)

Classical MDS

- Define an error,

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

  Then the optimal *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).

Outline

Motivation

Maximum Entropy Unfolding

Experiments

Discussion and Conclusions

Spectral Approaches

- CMDS gives a linear transformation between $\mathbf{X}$ and $\mathbf{Y}$.
- The spectral approaches in machine learning give a *nonlinear* relationship between the data and the distances.
- This is very clear for kernel PCA.

Kernel PCA

- In kernel PCA define distance with:

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

- If $k(\cdot, \cdot)$ is a Mercer kernel this is the squared distance in "feature space" (Ham et al., 2004).
- In CMDS this relationship is the *standard transformation* between a similarity and distance (Mardia et al., 1979).
- Kernel PCA (KPCA) recovers an $\mathbf{x}_{i,:}$ and a mapping from $\mathbf{Y}$ to $\mathbf{X}$ space.
- The mapping is induced through the choice of the *Mercer kernel*.

Classical MDS and KPCA

- Under the CMDS procedure the eigenvalue problem is performed on the centered kernel matrix,

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

where $\mathbf{K} = \left[k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:})\right]_{i,j}$.

- This matches the KPCA algorithm (Schölkopf et al., 1998)[1].
- **However**, for the commonly used exponentiated quadratic kernel,

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

KPCA actually *expands* the feature space rather than reducing the dimension (Weinberger et al., 2004).

[1] Kernel PCA also has an interpretation as a particular form of *metric* multidimensional scaling, see Williams (2001) for details.

Maximum Variance Unfolding

Learn a “Kernel” for Dimensionality Reduction

- The observation that KPCA expands the feature space motivated *maximum variance unfolding* (MVU, Weinberger et al., 2004).
- MVU: learn a kernel matrix that will allow for dimensionality reduction.
- Do this by considering only *local relationships* in the data.

Maximum Variance Unfolding

- Define a set of neighbors (e.g. by k-nearest neighbors).
- Construct a kernel matrix where only distances between neighboring data points are respected.
    - Specify the local distances as constraints.
    - Fill in other elements by maximizing the trace of the kernel matrix[2], $\text{tr}\left(\mathbf{K}\right)$.
    - Maximizing $\text{tr}\left(\mathbf{K}\right)$ maximizes the interpoint squared distances for all points that are unconnected in the neighborhood graph.
    - This "unravels" the manifold.

[2] The trace is the *total variance* of the data in feature space

Maximum Entropy Unfolding

Our Contribution

- Instead of maximizing total variance, we maximize entropy (Jaynes, 1986).
- Entropy is related to variance: so maybe resulting algorithm will be similar quality.
- Maximum entropy leads to a probability distribution: so we will also have a probabilistic model.
- The approach is also strongly related to other spectral techniques: they each turn out to approximate maximum entropy unfolding in some way.

Maximum Entropy

- In the maximum entropy formalism (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 points that are defined to be "neighbors".

Maximum Entropy

- For continuous data, the maximum entropy can only be defined relative to a base distribution.
- 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

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

  where $\mathcal{N}\left(i\right)$ represents the set of neighbors of data point i, and $\mathbf{Y} = [\mathbf{y}_{1,:}, \dots, \mathbf{y}_{n,:}]^{\top} \in \Re^{n\times p}$ 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.

Maximum Entropy Unfolding

- We now define the 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.

Gaussian Random Field

- This enables us to write

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

- Recall $\mathbf{D}$ has a zero diagonal. We can constrain $\mathbf{L}\mathbf{1} = \mathbf{0}$ giving

$$\begin{aligned} -\mathrm{tr}\left(\boldsymbol{\Lambda}\mathbf{D}\right) =& \mathrm{tr}\left(\mathbf{L}\mathbf{D}\right) \\ =& \mathrm{tr}\left(\mathbf{L}\mathbf{1}\mathrm{diag}\left(\mathbf{Y}\mathbf{Y}^\top\right)^\top - 2\mathbf{L}\mathbf{Y}\mathbf{Y}^\top + \mathrm{diag}\left(\mathbf{Y}\mathbf{Y}^\top\right)\mathbf{1}^\top\mathbf{L}\right) \\ =& -2\mathrm{tr}\left(\mathbf{L}\mathbf{Y}\mathbf{Y}^\top\right). \end{aligned}$$

- This probability distribution is a *Gaussian random field*

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

Data Feature Independence

- The GRF specifying independence across data *features*.
- Most applications of Gaussian models are applied independently across datar *points*.
  - Notable exceptions include Zhu et al. (2003); Lawrence (2004, 2005); Kemp and Tenebaum (2008).
- Maximum likelihood in this model is equivalent maximizing entropy under distance constraints.

Blessing of Dimensionality

- Maximum likelihood is consistent: (see e.g. Wasserman, 2003, pg 126)
  - As we increase data points parameters become better determined.
  - **Not** in this model.
  - As we increase data features parameters become better determined.
- This turns the large p small n problem on its head.
- There is a "Blessing of Dimensionality" in this model.

Lagrange Multiplier Gradients

- Gradient of each Lagrange multiplier is given by,

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

  $\left\langle\right\rangle_{p(\cdot)}$ is expectation under $p(\cdot)$.
- This result is expected given our maximum entropy formulation.
- Need expectation of squared distance:

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

  which is computed from $\mathbf{K} = (\mathbf{L} + \gamma\mathbf{I})^{-1}$.

Standard Transformation Again

- This is immediately recognized as a scaled version of the *standard transformation* between distances and similarities

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

- This relationship arises naturally in the probabilistic model: each GRF has an associated distance matrix.
- Not strictly speaking in MEU and MVU these are not Mercer kernels because we can't represent them as

$$k_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:})$$

- Really it is a covariance matrix from a Gaussian model.

Nonparametric Model

- If K neighbors are used for each data point there are $O(Kn)$ parameters in the model.
- The model is *nonparametric*.
- For the parameters to be well determined we require a large number of features, p.
- Otherwise we would need to look to regularize the model.
- The model is excellent for the so-called "large p small n domain".

Visualization

- Given the maximum likelihood solution we look for a reduced dimensional representation.
- This is done, 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 this is just one way of visualizing the underlying GRF.
- It happens to be easy to compute!

Relation with MVU

- Determinant and trace of covariance are functions of eigenvalues.
- The entropy of a Gaussian depends on the determinant of the covariance matrix.
- Determinant of **K** is

$$\log|\mathbf{K}| = \sum_{i=1}^{n} \log \lambda_i.$$

- MVU maximizes the total variance (the trace)

$$\mathrm{tr}\left(\mathbf{K}\right) = \sum_{i=1}^{n} \lambda_i.$$

Positive Definitiveness

- Need to ensure that the covariance matrix is positive definite.
- In MVU use a semidefinite program.
- In MEU the objective is not linear in $\mathbf{K}$, need other approaches.
- Possibilities include:
  1. building an "attractive" system (see e.g. Koller and Friedman, 2009, pg 255), although now the distance constraints would be inequalities;
  2. constrain $\mathbf{L}$ to be diagonally dominant through adjusting γ;
  3. factored representation like $\mathbf{L} = \mathbf{B}\mathbf{B}^\top$;
  4. or design an algorithm that maintains positive definitiveness.

Generalization of PCA

- 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).

Relationship to Laplacian Eigenmaps

- In Laplacian eigenmaps (Belkin and Niyogi, 2003) a graph Laplacian is specified across the data points.
- This Laplacian has exactly the same form as our matrix **L**.
- 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

Smallest Eigenvalues of Laplacian

- The eigendecomposition of the covariance is

$$\mathbf{K} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$$

- The eigendecomposition of the Laplacian is

$$\mathbf{L} = \mathbf{U}\left(\boldsymbol{\Lambda}^{-1} - \gamma\mathbf{I}\right)\mathbf{U}^\top$$

- In other words, the principal eigenvalues of **K** will be the smallest eigenvalues of **L**.
- Note the smallest eigenvalue of **L** is zero and associated with the constant eigenvector.
- In CMDS this is removed by the centering operation and in LE it is discarded.

Laplacian Eigenmaps

- Once the parameters of the Laplacian are set CMDS is being performed to recover the latent variables in Laplacian eigenmaps.
- No constraints are imposed in Laplacian eigenmaps so distances will not be preserved.
- LE gains significant computational advantage by not representing the covariance matrix explicitly.
- No matrix inverses are required in the algorithm and the resulting eigenvalue problem is sparse.
- LE can be applied to much larger data sets than would be possible for MEU or MVU.

Factored Algorithm

- Impose positive definite constraint through

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

- Constrain $\mathbf{B}^\top\mathbf{1} = \mathbf{0}$ giving $\mathbf{L}\mathbf{1} = \mathbf{0}$.
- Do this by setting $b_{i,i} = -\sum_{j\in\mathcal{N}(i)} b_{j,i}$
- Force $b_{j,i} = 0$ if $j \notin \mathcal{N}\left(i\right)$.
- Laplacian is now positive definite.

Locally Linear Embeddings

- Locally linear embeddings (Roweis and Saul, 2000) are then a specific case of this random field model where
  1. The diagonal sums, $b_{i,i}$, are further constrained to unity.
  2. The parameters of the model are optimized by maximizing the pseudolikelihood of the resulting GRF.

Point One

- For unit diagonals we have $\mathbf{B} = \mathbf{I} - \mathbf{W}$.
- Here the off diagonal sparsity pattern of $\mathbf{W}$ matches $\mathbf{B}$.
- Thus
$$(\mathbf{I} - \mathbf{W})^\top \mathbf{1} = \mathbf{0}.$$
- LLE proscribes that the smallest eigenvectors of
$$(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})^\top = \mathbf{B}\mathbf{B}^\top = \mathbf{L}$$
(like Laplacian Eigenmaps).
- This is equivalent to CMDS on the Gaussian random field described by $\mathbf{L}$.

Second Point

- Pseudolikelihood approximation (see e.g. Koller and Friedman, 2009, pg 970) is the product of the conditional densities:

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

  $\mathbf{Y}_{\backslash i}$ represents all that data other than the ith point.
- True joint likelihood is proportional to this but requires renormalization.
- In pseudolikelihood this normalization is ignored.

Relation to LLE

- To see how it relates to LLE note

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

  so we have

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

- The conditionals can be rewritten as

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

Pseudolikelihood Approximation

- Optimizing the pseudolikelihood is equivalent to optimizing

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

  which is equivalent to solving n independent regression problems with a constraint on the regression weights that they sum to one.
- This is how parameters in LLE (Roweis and Saul, 2000) are optimized.
- The constraint arises because the regression weights are constrained to be $w_{j,i}/b_{i,i}$ and $b_{i,i} = \sum_{j \in \mathcal{N}(i)} w_{j,i}$.
- In LLE a further constraint is imposed $b_{i,i} = 1$.

LLE Approximates MEU

- Locally linear embeddings are an approximation to maximum likelihood on the Gaussian random field.
- Constrain Laplacian to be positive semidefinite by a factorized form.
- Pseudolikelihood also allows for relatively quick parameter estimation.
  - ignoring the partition function removes the need to invert to recover the covariance matrix.
  - LLE can be applied to larger data sets than MEU or MVU.

Note: 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 and PCA

- LLE is motivated by considering local linear embeddings of the data.
- Interestingly, as we increase the neighborhood size to $K = n - 1$ we do not recover PCA.
- Strange because PCA is the optimal linear embedding of the data under linear Gaussian constraints.
- But LLE is optimizing a pseudolikelihood: in contrast the MEU algorithm, which LLE approximates, does recover PCA when $K = n - 1$.

Isomap

- Isomap (Tenenbaum et al., 2000) directly follows the CMDS framework.
- A sparse graph of distances is created between all points considered to be neighbors.
- This graph is then filled in for all non-neighbors with a shortest path algorithm.
- This matrix is element-wise squared to give a matrix of *square distances*.
- This is then processed in the usual manner (centering and multiplying by -0.5) to provide a similarity matrix for multidimensional scaling.

Compare with MEU

- Both MVU and MEU can be thought of as starting with a sparse graph of (squared) distances.
- Other distances are then filled in by maximizing the total variance or entropy.
- The interneighbor distances in this graph are preserved just like in isomap.
- For MVU and MEU **K**, is constrained positive definite.
- For isomap the implied covariance can have negative eigenvalues (see (Weinberger et al., 2004)).
- Isomap is slower than LLE and LE: requires a dense eigenvalue problem and a shortest path algorithm.

Sparse Inverse Covariances

- Spectral algorithms are related to Gaussian Random Fields.
- Suggests fitting a GRF with a sparse inverse covariance, $\mathbf{L}$.
- Regularize the elements of the inverse covariance with e.g. L1.
- This can be done quickly through an iterative regression problems (see Hastie et al., 2009, Chapter 17).
- The method retains a positive definite $\mathbf{L}$.
- We call this algorithm Dimensionality Reduction through Iterative Log Likelihood maximization (DRILL).
- This involves a slightly different interpretation of the inverse covariance.
- It is no longer distances that are constrained (directly) but covariances.
- Marginal variances are also constrained, ensuring distances also match.

Outline

Motivation

Maximum Entropy Unfolding

Experiments

Discussion and Conclusions

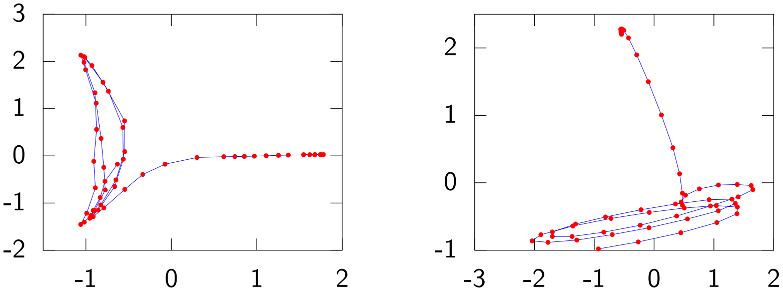
Simple Experiments

- Consider two real data sets.
- We apply each of the spectral methods we have reviewed.
- Also apply MEU using positive constraints on the Lagrange multipliers (denoted MEU) and the DRILL.
- Follow the suggestion of Harmeling (Harmeling, 2007) and use the GPLVM likelihood (Lawrence, 2005) for embedding quality.
- The higher the likelihood the better the embedding.

Motion Capture Data

- Data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run.
- 102 dimensional data set containing 55 frames of motion capture.
- Subject begins the motion from stationary and takes approximately three strides of run.
- Should see this structure in the visualization: a starting position followed by a series of loops.
- Data was made available by Ohio State University.
- The two dominant eigenvectors are visualized in following figures.

Laplacian Eigenmaps and LLE

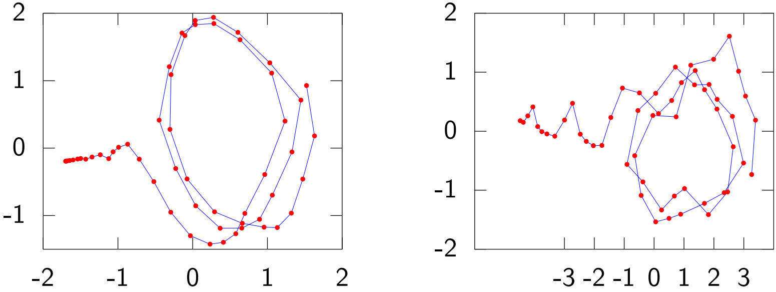


(a) Laplacian Eigenmaps

(b) Locally Linear Embedding

Figure: Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

Isomap and MVU

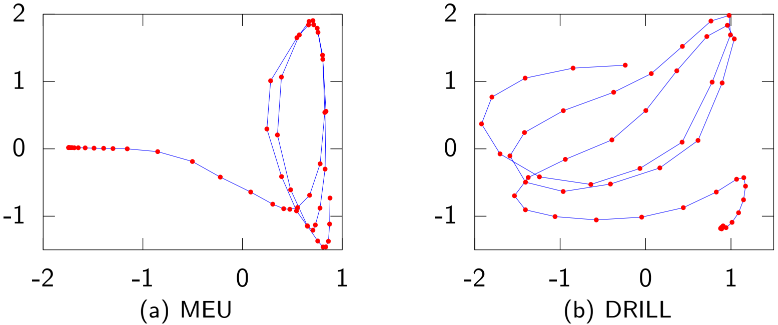


(a) Isomap

(b) MVU

Figure: Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

MEU and DRILL



(a) MEU

(b) DRILL

Figure: Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

Motion Capture: Model Scores

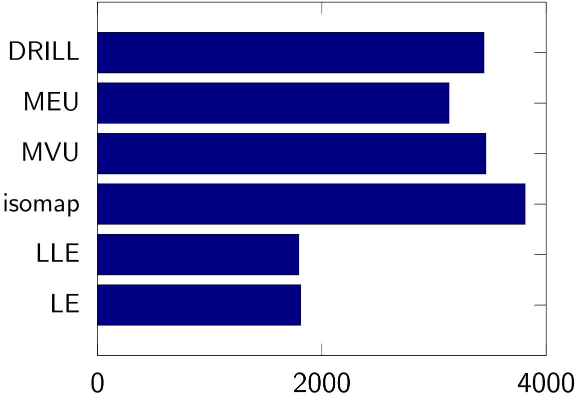
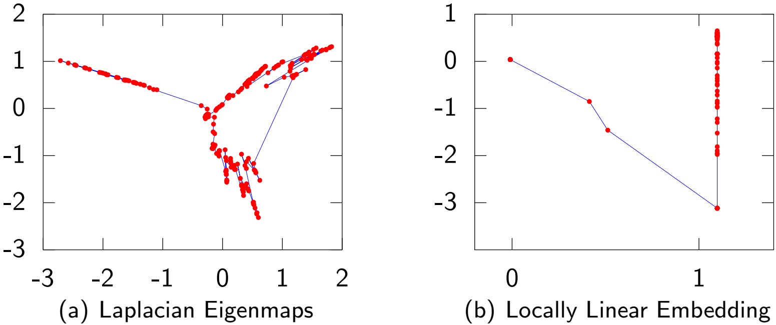


Figure: Model score for the different spectral approaches.

Robot Navigation Example

- Second data set: series of recordings from a robot as it traces a square path in a building.
- It records the strength of WiFi signals (see Ferris et al., 2007, for an application).
- Robot only in two dimensions, the inherent dimensionality of the data should be two.
- Robot completes a single circuit after entry: it is expected to exhibit "loop closure".
- Data consists of 215 frames of measurement of WiFi signal strength of 30 access points.

Laplacian Eigenmaps and LLE

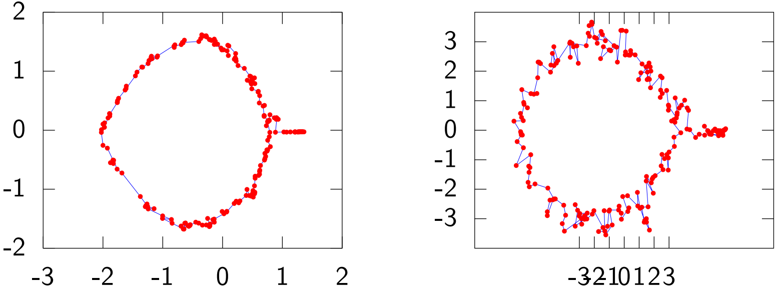


(a) Laplacian Eigenmaps

(b) Locally Linear Embedding

Figure: Models show loop closure but smooth the trace to different degrees.

Isomap and MVU

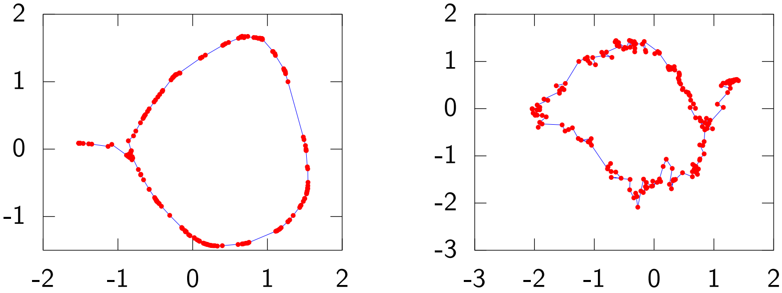


(a) Isomap

(b) MVU

Figure: Models show loop closure but smooth the trace to different degrees.

MEU and DRILL



(a) MEU

(b) DRILL

Figure: Models show loop closure but smooth the trace to different degrees.

Robot Navigation: Model Scores

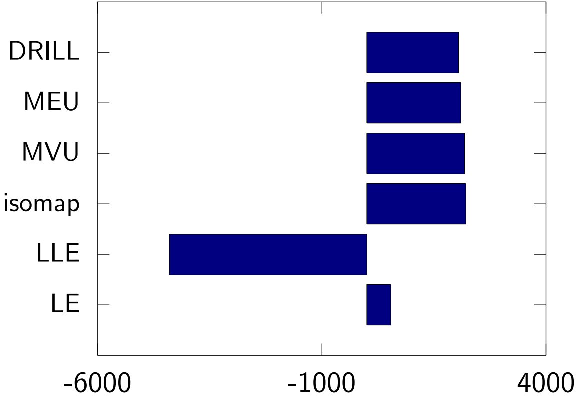


Figure: Model score for the different spectral approaches.

Final Experiment: Structure Learning

- Test the ability of L1 regularization of the random field to learn the neighborhood.
- Considered the motion capture data and used the DRILL with a neighborhood size of 20 and full connectivity.
- L1 regularization on the parameters: vary regularization size and seek a maximum under the GPLVM.

Structure Learning from Neighborhood of 20

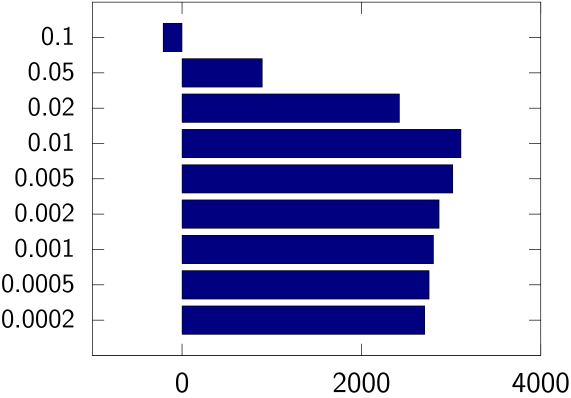


Figure: Model scores for different regularization coefficients.

Structure Learning from Neighborhood of 20

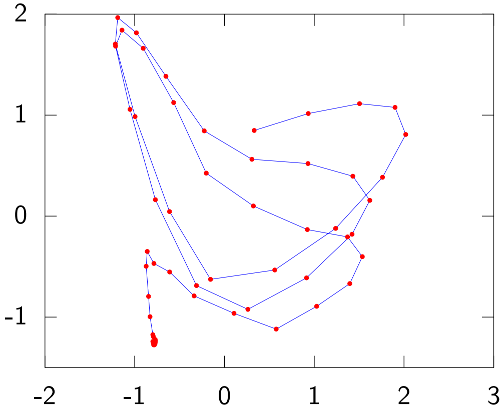


Figure: Visualization associated with highest model score.

Full Structure Learning

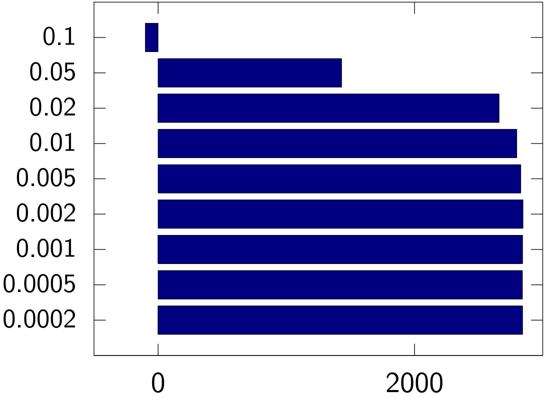


Figure: Model scores for different regularization coefficients.

Full Structure Learning

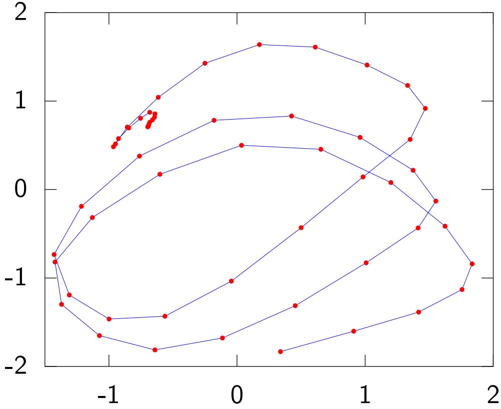


Figure: Visualization associated with highest model score.

Outline

Motivation

Maximum Entropy Unfolding

Experiments

Discussion and Conclusions

Discussion

- New perspective on dimensionality reduction algorithms based around maximum entropy.
- Start with MVU and end with GRFs and L1 based structure learning.
- We hope that this perspective on dimensionality reduction will encourage new strands of research at the interface of these areas.

Stages of Spectral Dimensionality Reduction

- Our perspective shows there are 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 different algorithms.
    3. The relationship between points in the similarity matrix is visualized using the eigenvectors of the similarity matrix.

Our Perspective

- Each step is somewhat orthogonal.
- Neighborhood relations need not come from nearest neighbors: can use structure learning.
- Main difference between approaches is how similarity matrix entries are determiend.
- Final step attempts to visualize the similarity using eigenvectors. This is just one possible approach.
- There is an entire field of graph visualization proposing different approaches to visualizing such graphs.

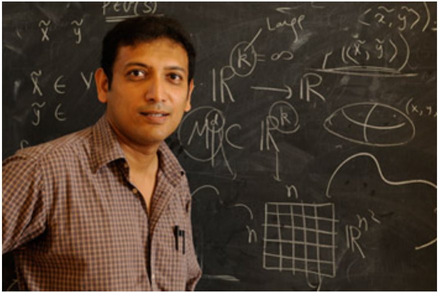
Advantages of Existing Approaches

- Conflating the three steps allows faster complete algorithms.
- E.g. mixing 2nd & 3rd allows speed ups by never computing the similarity matrix.
- We still can understand the algorithm from the unifying perspective while exploiting the computational advantages offered by this neat shortcut.

Acknowledgements

Conversations with John Kent, Chris Williams, Brenden Lake, Joshua Tenenbaum and John Lafferty have influenced the thinking in this work.

Partha and Sam



References I

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

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

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

J. Ham, D. D. Lee, S. Mika, and B. Schölkopf. A kernel view of dimensionality reduction of manifolds. In Greiner and Schuurmans (2004). [PDF].

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

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2nd edition, 2009.

E. 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.

C. Kemp and J. B. Tenebaum. The discovery of structural form. *Proc. Natl. Acad. Sci. USA*, 105(31), 2008.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. [Google Books] .

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

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

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press, London, 1979. [Google Books] .

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

References II

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. [DOI].

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

L. A. Wasserman. *All of Statistics*. Springer-Verlag, New York, 2003. [Google Books] .

K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In Greiner and Schuurmans (2004), pages 839–846.

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

X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From Gaussian fields to Gaussian processes. Technical Report CMU-CS-03-175, Carnegie Mellon University, [PDF].