A Probabilistic Perspective on Spectral Dimensionality Reduction

Neil D. Lawrence

Departments of Neuro- and Computer Science, University of Sheffield, U.K.
Invited Talk at AAAI Symposium 2010

11th November 2010

Outline

Notation

p	data dimensionality	
q	latent dimensionality	
n	number of data points	
$\mathbf{Y}$	*design matrix* containing our data	$n \times p$
$\mathbf{X}$	matrix of latent variables	$n \times q$
$\mathbf{D}$	matrix of interpoint squared distances	$n \times n$
$\mathbf{K}$	similarities/covariance/kernel	$n \times n$
$\mathbf{L}$	Laplacian matrix	$n \times n$

Row vector from matrix $\mathbf{A}$ given by $\mathbf{a}_{i,:}$ column vector $\mathbf{a}_{:,j}$ and element given by $a_{i,j}$.

Outline

Dimensionality Reduction and Distances

- We consider dimensionality reduction algorithms that operate on (squared) distances.
- There is an equivalence between squared distances and similarities/kernels.

$$d_{i,j} = k_{i,i} - 2k_{i,j} + k_{j,j}.$$

- It was originally known as the standard transformation (Mardia et al., 1979).
- If $k_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:})$ it is the "distance in feature space" (Schölkopf and Smola, 2001).
- If $k_{i,j}$ is an element from a covariance matrix $\mathbf{K}$, it is the *expected squared distance* between two samples from the corresponding Gaussian.

Moving from Squared Distance to Similarity

- Matrix form of squared distance,

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

- Centering matrix $\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^\top$: $\mathbf{H}\mathbf{1} = \mathbf{0}$.
- This implies,

$$-\frac{1}{2}\mathbf{H}\mathbf{D}\mathbf{H} = \mathbf{H}\mathbf{K}\mathbf{H}.$$

Spectral Dimensionality Reduction

- Spectral approaches to dimensionality reduction.
  1. Given a matrix of size $n \times n$.
  2. Visualize data with eigenvectors.
- 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).
- Also kernel PCA (Schölkopf et al., 1998; Ham et al., 2004).

Classical Multidimensional Scaling Perspective

- Classical multidimensional scaling (CMDS)
  1. Compute an $n \times n$ squared distance matrix, $\mathbf{D}$.
  2. Form the centered "similarity matrix" $\mathbf{HKH} = -\frac{1}{2}\mathbf{HDH}$.
  3. Visualize through principal eigenvectors.
- This minimizes a particular objective.
- Main innovation in ML work: how we compute the distances.

This Talk

- Probabilistic approach to constructing distance matrices.
- Relate isomap, LLE, LE and MVU to our approach.
- Provide a unifying perspective of *Gaussian random fields* and CMDS.

Outline

Spectral Approaches

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

Kernel PCA

- Kernel PCA define squared distance:

$$d_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{i,:}) - 2k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:}) - k(\mathbf{y}_{j,:}, \mathbf{y}_{j,:}) \tag{1}$$

- $k(\cdot, \cdot)$ is a Mercer kernel (Ham et al., 2004).
- Kernel PCA (KPCA) recover 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

- CMDS procedure performs eigenvalue problem on

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

- 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 (Weinberger et al., 2004).

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

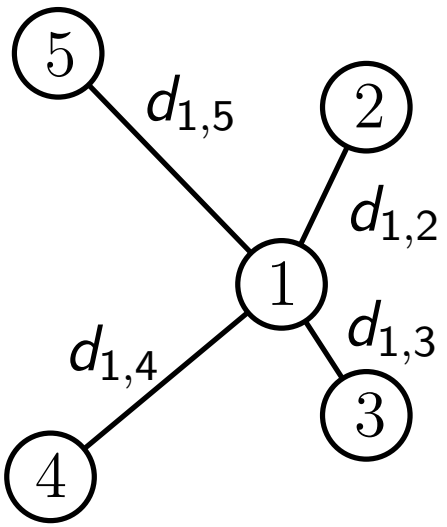
Maximum Variance Unfolding

Learn a "Kernel" for Dimensionality Reduction

- MVU (Weinberger et al., 2004): learn a "kernel matrix" that will allow for dimensionality reduction.
- Consider only *local relationships* in the data.
- Take a set of neighbors.
- Construct a kernel matrix where only distances between neighbors match data distances.

Maximum Variance Unfolding

- Maximize $\mathrm{tr}(\mathbf{K})$.: equivalent to maximizing distances between non-neighbors.[2].



- MVU constrains "feature space" distances to be equal to observed

$$d_{i,j} = k_{i,i} - 2k_{i,j} + k_{j,j}$$

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

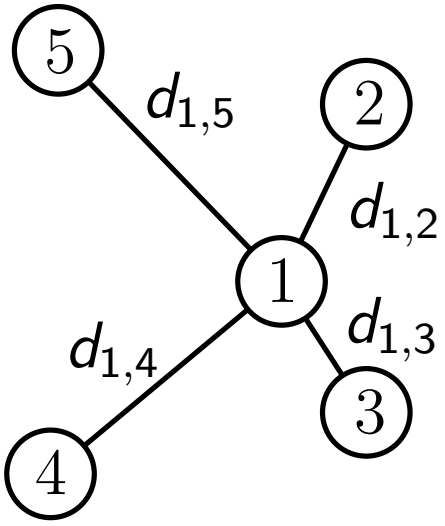
Maximum Entropy Unfolding

Our Contribution

- Maximize *entropy* instead of variance (Jaynes, 1986): MEU.
- Entropy and variance are closely related.
- Maximum entropy leads to a probabilistic model.
- Each spectral approach approximates MEU in some way.

Maximum Entropy Unfolding

- Maximize entropy of distribution subject to constraints on *moments*.

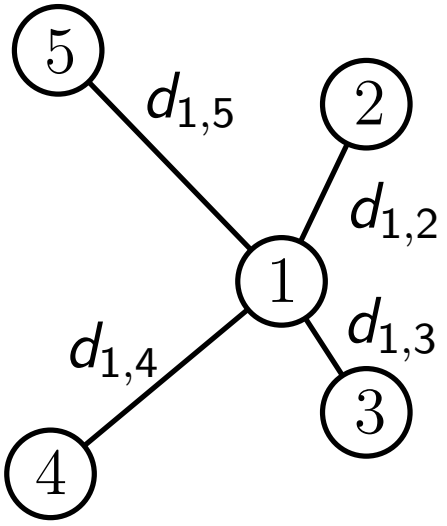


- MEU constraints are on expected distances between neighbors.

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

Maximum Entropy Unfolding

- Maximize entropy of distribution subject to constraints on *moments*.



- MEU constraints are on expected distances between neighbors.

$$d_{i,j} = k_{i,i} - 2k_{i,j} + k_{j,j}$$

Maximum Entropy

- Maximum entropy distribution.

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

$\mathcal{N}(i)$ is neighborhood, $\{\lambda_{i,j}\}$, Lagrange multipliers.

Maximum Entropy

- Maximum entropy distribution.

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

$\mathcal{N}(i)$ is neighborhood, $\{\lambda_{i,j}\}$, Lagrange multipliers. Lagrange multipliers in sparse matrix $\mathbf{\Lambda}$.

Maximum Entropy

- Maximum entropy distribution.

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

$\mathcal{N}(i)$ is neighborhood, $\{\lambda_{i,j}\}$, Lagrange multipliers. Introduce Laplacian: $\ell_{i,j} = -\lambda_{i,j}$, $\ell_{i,i} = \sum_{j \in \mathcal{N}(i)} \lambda_{i,j}$, $\mathbf{L1} = \mathbf{0}$.

Details: Moving to the Laplacian

- $\mathbf{D}$ has a zero diagonal.
- $\mathrm{tr}\left(\mathbf{LD}\right)$ is unaffected by diagonal of $\mathbf{L}$.
- Constrain $\mathbf{L1} = \mathbf{0}$ giving

$$-\mathrm{tr}\left(\mathbf{\Lambda D}\right) = \mathrm{tr}\left(\mathbf{LD}\right)$$

Details: Moving to the Laplacian

- $\mathbf{D}$ has a zero diagonal.
- $\mathrm{tr}\left(\mathbf{LD}\right)$ is unaffected by diagonal of $\mathbf{L}$.
- Constrain $\mathbf{L1} = \mathbf{0}$ giving

$$-\mathrm{tr}\left(\mathbf{\Lambda D}\right) = \mathrm{tr}\left(\mathbf{LD}\right)$$

Details: Moving to the Laplacian

- $\mathbf{D}$ has a zero diagonal.
- $\mathrm{tr}\left(\mathbf{LD}\right)$ is unaffected by diagonal of $\mathbf{L}$.
- Constrain $\mathbf{L1} = \mathbf{0}$ giving

$$-\mathrm{tr}\left(\mathbf{\Lambda D}\right) = \mathrm{tr}\left(\mathbf{L1}\mathrm{diag}\left(\mathbf{YY}^\top\right)^\top - 2\mathbf{LYY}^\top + \mathrm{diag}\left(\mathbf{YY}^\top\right)\mathbf{1}^\top\mathbf{L}\right)$$

Details: Moving to the Laplacian

- $\mathbf{D}$ has a zero diagonal.
- $\mathrm{tr}\left(\mathbf{LD}\right)$ is unaffected by diagonal of $\mathbf{L}$.
- Constrain $\mathbf{L1} = \mathbf{0}$ giving

$$-\mathrm{tr}\left(\mathbf{\Lambda D}\right) = \mathrm{tr}\left(\cancel{\mathbf{L1}\mathrm{diag}\left(\mathbf{YY}^\top\right)^\top} - 2\mathbf{LYY}^\top + \cancel{\mathrm{diag}\left(\mathbf{YY}^\top\right)\mathbf{1}^\top\mathbf{L}}\right)$$

Details: Moving to the Laplacian

- $\mathbf{D}$ has a zero diagonal.
- $\mathrm{tr}\left(\mathbf{LD}\right)$ is unaffected by diagonal of $\mathbf{L}$.
- Constrain $\mathbf{L1} = \mathbf{0}$ giving

$$-\mathrm{tr}\left(\mathbf{\Lambda D}\right) = -2\mathrm{tr}\left(\mathbf{LYY}^\top\right).$$

Gaussian Random Field

- This probability distribution is a *Gaussian random field*

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

Outline

Relationship to Laplacian Eigenmaps

- Laplacian eigenmaps (Belkin and Niyogi, 2003): graph Laplacian is specified across the data points.
- Laplacian has exactly the same form as our matrix $\mathbf{L}$.
- Parameters of the Laplacian are set either as constant or according to the distance between two points.
- Smallest eigenvectors of this Laplacian are then used for visualizing the data.

Smallest Eigenvalues of Laplacian

- Eigendecomposition of the covariance is

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

- Eigendecomposition of the Laplacian is

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

- Principal eigenvalues of $\mathbf{K}$ are smallest eigenvalues of $\mathbf{L}$.
- (smallest eigenvalue of $\mathbf{L}$ is zero)

Laplacian Eigenmaps

- Set parameters of Laplacian.
- Perform CMDS on the implied matrix $\mathbf{K}$.
  1. No constraints are imposed in Laplacian eigenmaps so distances will not be preserved.
  2. LE gains significant computational advantage by not representing the covariance matrix explicitly.
  3. No matrix inverses required, eigenvalue problem sparse.

Locally Linear Embedding

- Factorize the Laplacian as

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

- Now constrain $\mathbf{M}^\top \mathbf{1} = \mathbf{0}$ giving $\mathbf{L}\mathbf{1} = \mathbf{0}$.
  - i.e. $m_{i,i} = -\sum_{j \in \mathcal{N}(i)} m_{j,i}$
  - Set $m_{j,i} = 0$ if $j \notin \mathcal{N}(i)$.

Locally Linear Embedding

- Locally linear embeddings (Roweis and Saul, 2000) are then a specific case of MEU where
  1. The diagonal sums, $m_{i,i}$, are further constrained to unity.
  2. Model parameters found by maximizing *pseudolikelihood* of the data.

Point One

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

Second Point

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

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

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

Relation to LLE

- First note

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

so we have

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

Conditionals

- Factors can be written as conditionals

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

Pseudolikelihood Approximation

- Optimizing the pseudolikelihood is equivalent to optimizing

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

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

LLE Approximates MEU

- LLE is an approximation to maximum likelihood.
- Laplacian has 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) follows the CMDS framework.
- Sparse graph of distances is created.
- Fill in graph for non-neighbors with a shortest path algorithm.
- Element-wise square the matrix.
- Process this in the usual manner.

Compare with MEU

- Both MVU and MEU can be thought of as starting with a sparse graph of (squared) distances.
- Fill in other distances by maximizing the total variance/entropy.
- Interneighbor distances in this graph are preserved just like in isomap.
    1. For isomap the implied covariance can have negative eigenvalues (see (Weinberger et al., 2004)).
    2. Isomap is slower than LLE and LE: requires a dense eigenvalue problem and a shortest path algorithm.

Outline

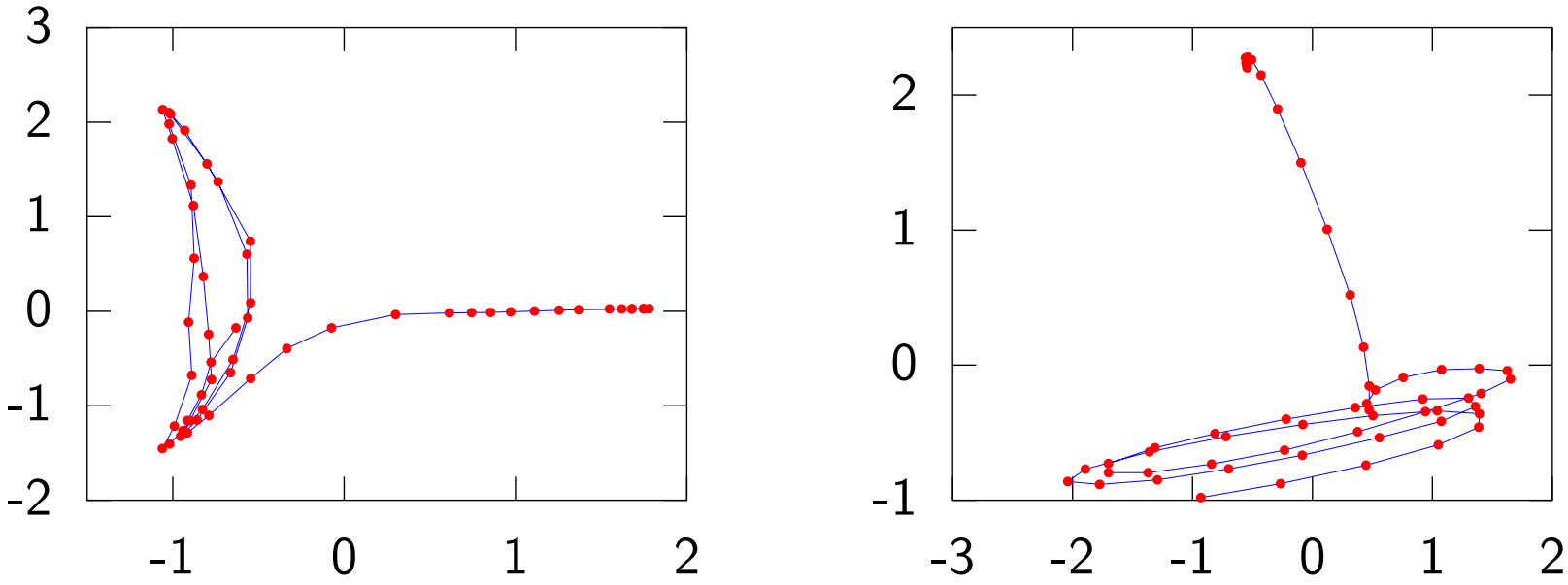
Simple Experiments

- Consider two real data sets.
- We apply each of the spectral methods we have reviewed.
- Apply the MEU framework.
- 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.

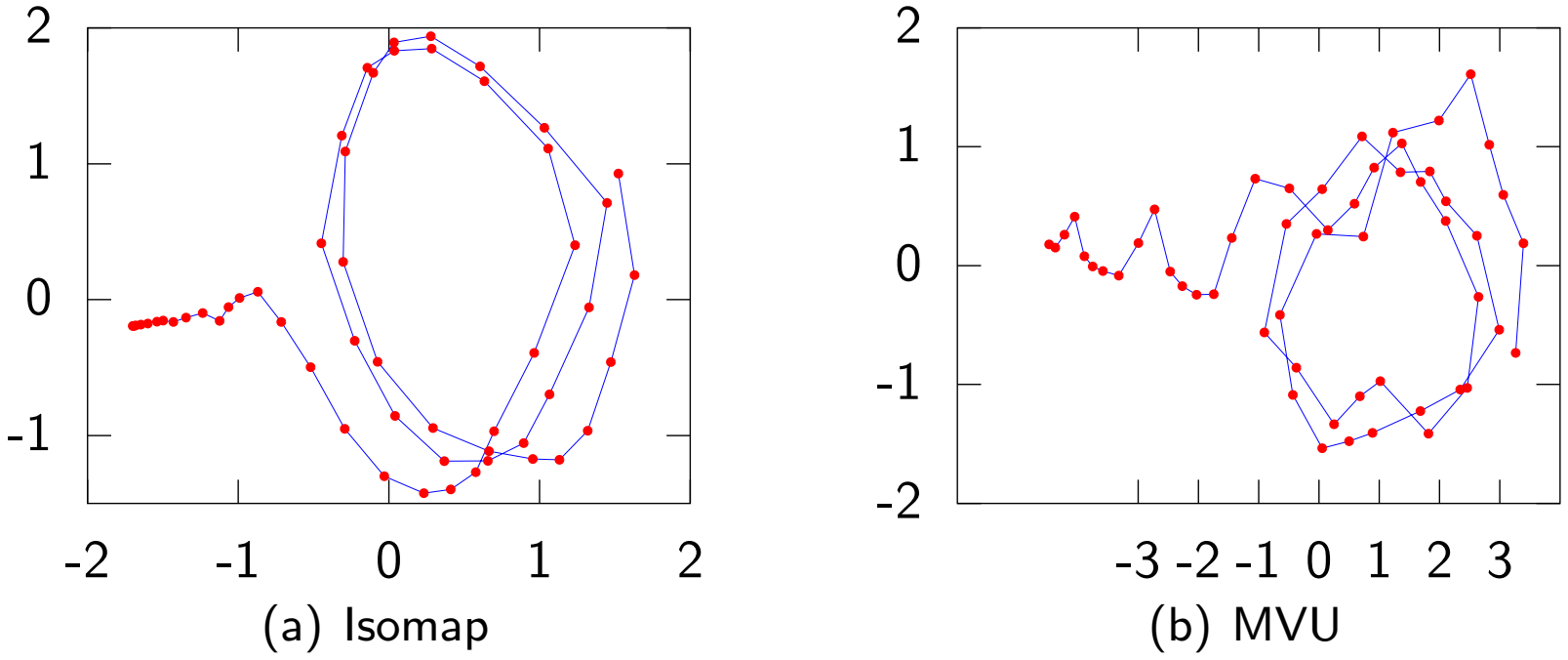
Laplacian Eigenmaps and LLE



(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

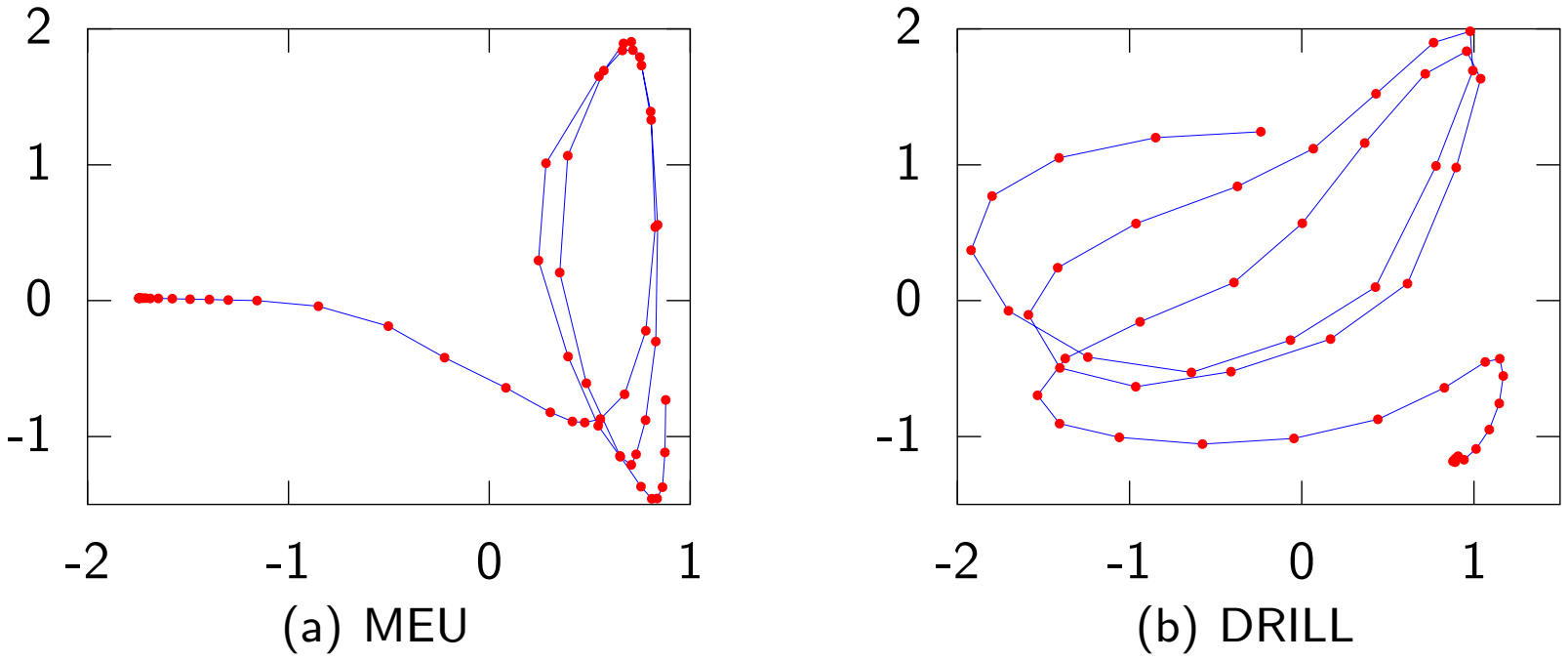


Figure: Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

Motion Capture: Model Scores

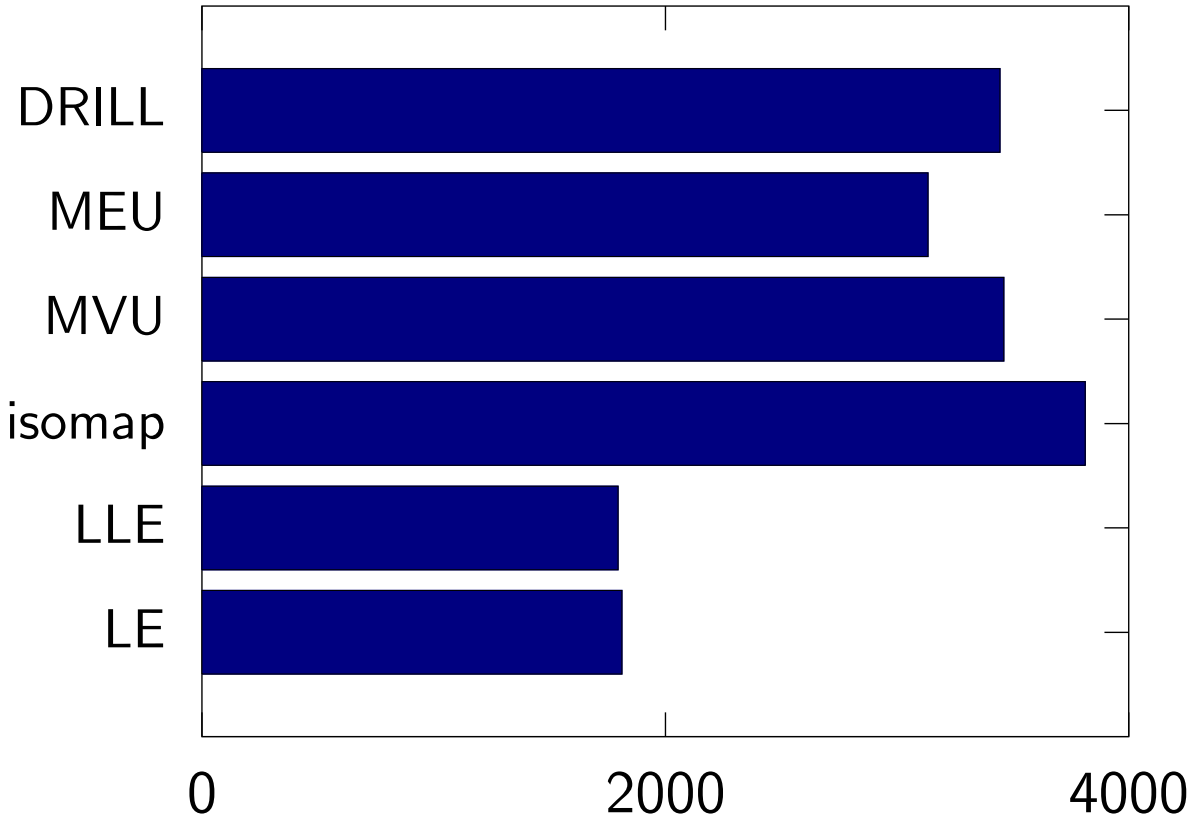
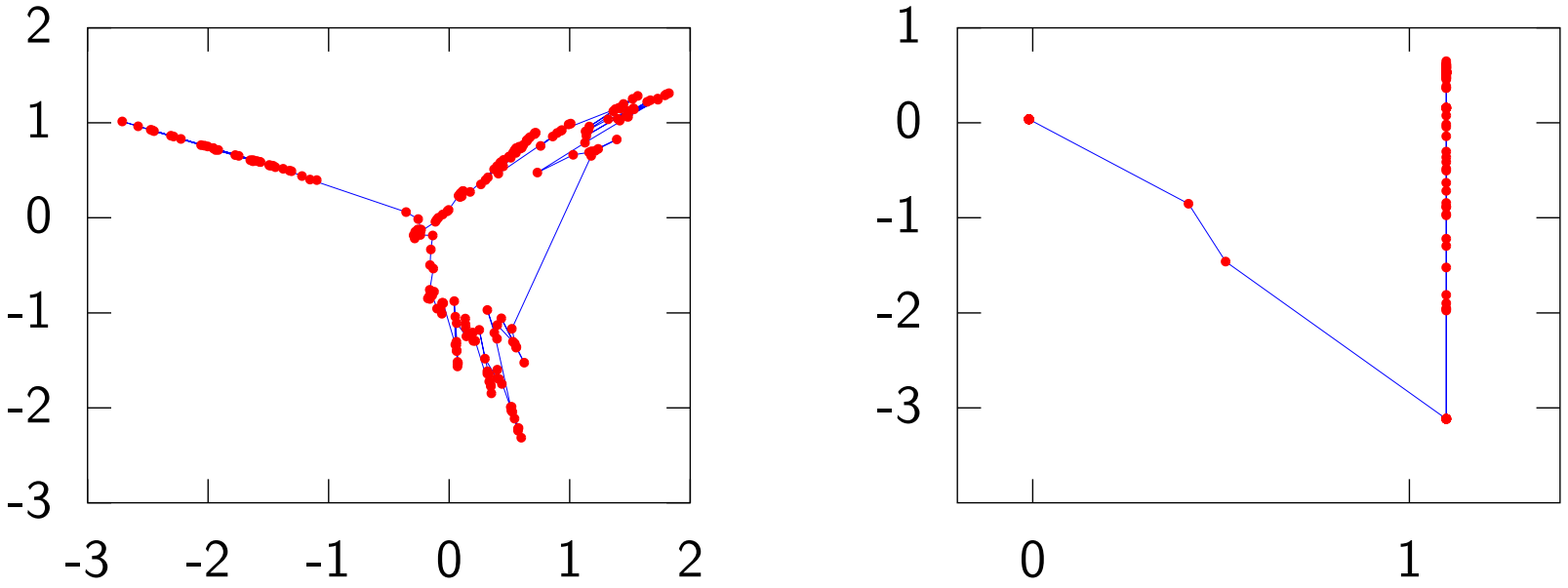


Figure: Model score for the different spectral approaches.

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

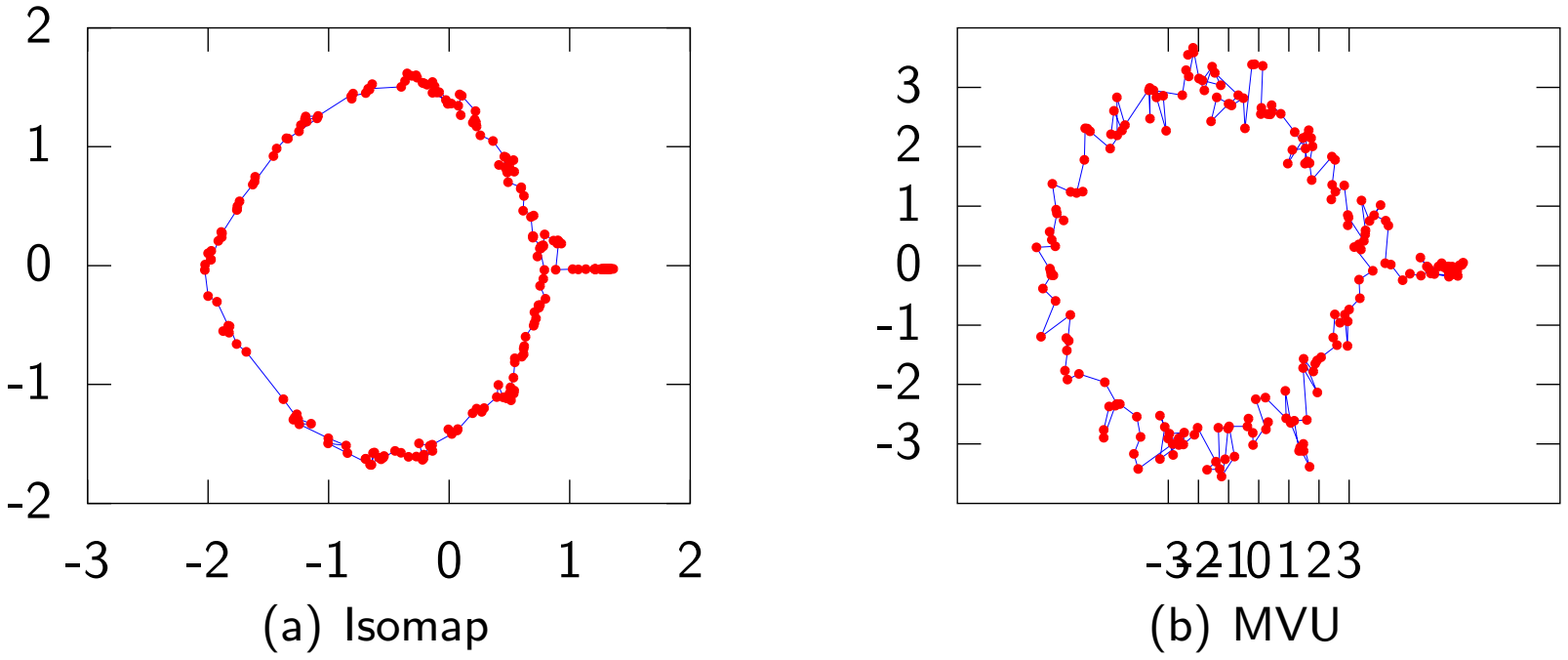


(a) Laplacian Eigenmaps

(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

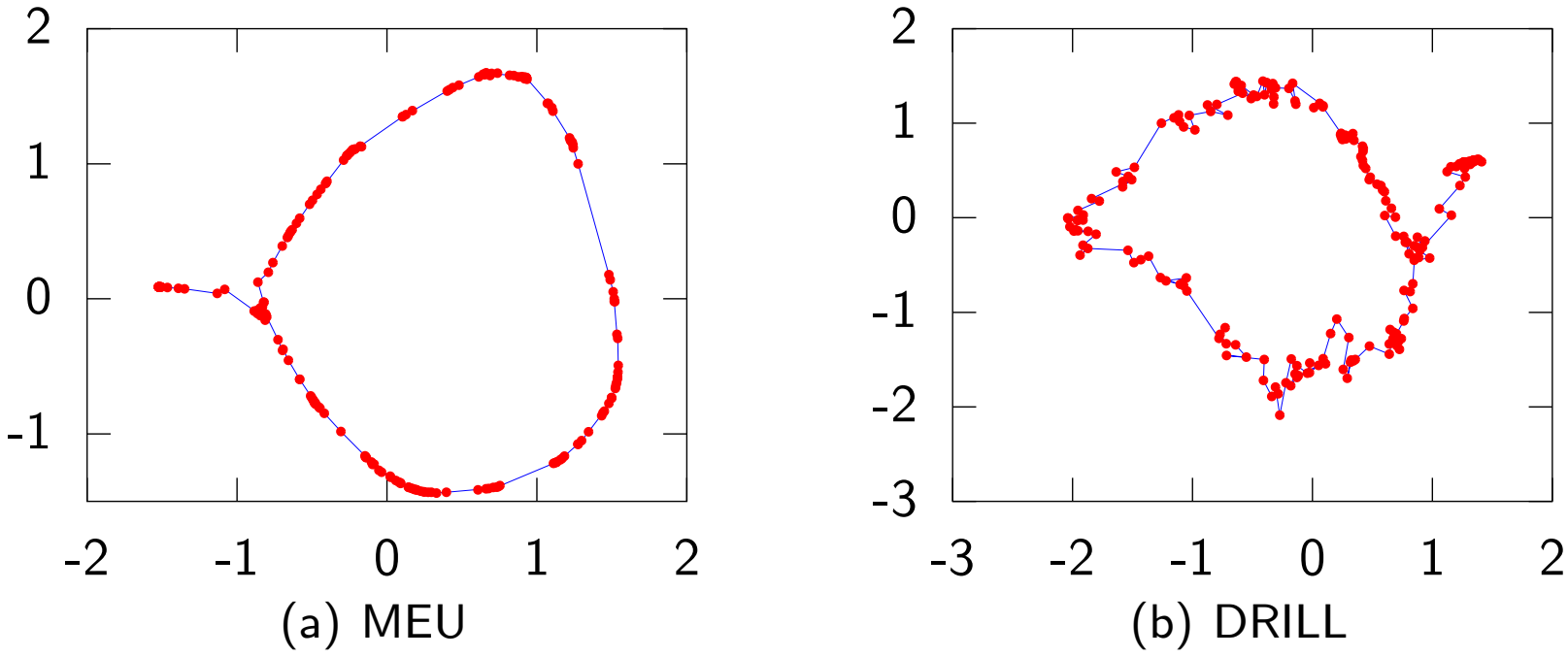


Figure: Models show loop closure but smooth the trace to different degrees.

Robot Navigation: Model Scores

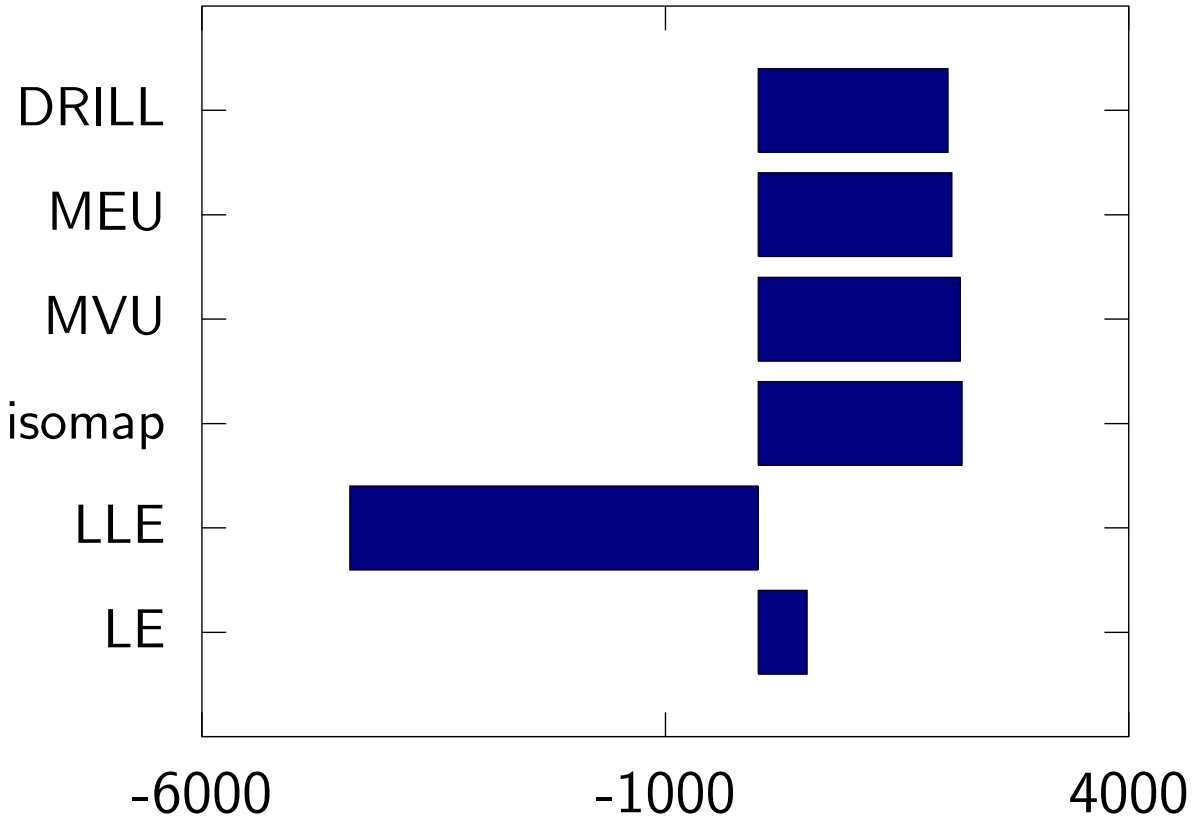


Figure: Model score for the different spectral approaches.

Outline

Discussion

- New perspective on dimensionality reduction algorithms based around maximum entropy.
- Start with MVU and end with GRFs.
- 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.

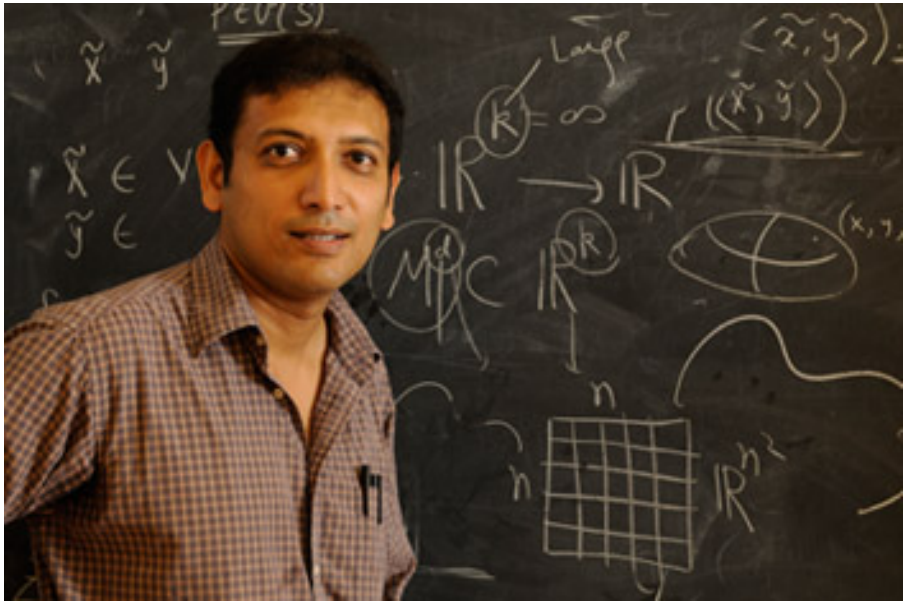
Advantages of Existing Approaches

- 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,

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.

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

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].

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

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2001.

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].

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.