Bayesian Manifold Learning: The Locally Linear Latent Variable Model (LL-LVM)

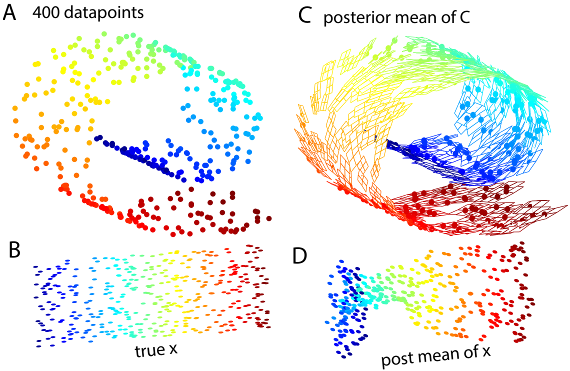
Mijung Park[1], **Wittawat Jitkrittum**[1], Ahmad Qamar[2],
Zoltán Szabó[1], Lars Buesing[3], Maneesh Sahani[1]

[1] Gatsby Unit, UCL
[2] Thread Genius
[3] Google DeepMind

CSML lunch talk. NIPS preview.
4 Dec 2015

Overview

- Observe $\mathbf{y} = (\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top)^\top$ in $\mathbb{R}^{d_y}$. Large d_y.
- Manifold learning = discover low-d structure in high-d data space.
- Propose a model $p(\mathbf{y}, \mathbf{C}, \mathbf{x})$, over observations $\mathbf{y}$, locally linear maps $\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_n)$, and manifold coordinates $\mathbf{x} = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top)^\top$.



- **Goal**: $p(\mathbf{C}, \mathbf{x} \mid \mathbf{y})$. Observe $\mathbf{y}$.
- Produce Fig. C, D from Fig. A.

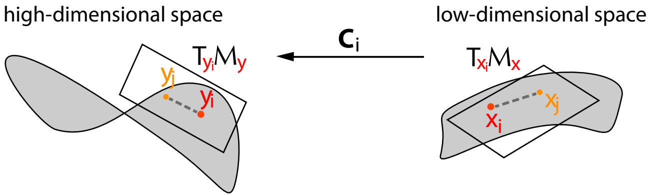
Existing Works on Manifold Learning

- **Non-probabilistic**: PCA, multidimensional scaling (MDS), ISOMAP, Locally Linear Embedding (LLE), etc.
  - Easy optimization.
  - Preserve local neighbourhood geometries.
  - No uncertainty estimates.
  - No principled way to choose neighbourhood graph.
- **Probabilistic**: GP-LVM [Lawrence, 2003].
  - Uncertainty estimates available.
  - Out-of-sample extension.
  - Inference requires auxiliary variables.
  - Manifold structure defined by a covariance function (can be unintuitive).

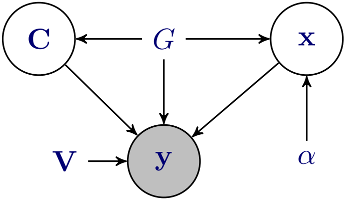
Proposed LL-LVM:

- All advantages above.
- Probabilistic. Graph-based.
- Can choose the right neighbourhood graph.
- Optimization = standard variational Bayes.

Locally Linear Latent Variable Model (LL-LVM)



- Locally linear assumption: $\mathbf{y}_j - \mathbf{y}_i \approx \mathbf{C}_i(\mathbf{x}_j - \mathbf{x}_i)$ where $\mathbf{C}_i \in \mathbb{R}^{d_y \times d_x}$.
- Model:



$$p(\mathbf{y}, \mathbf{C}, \mathbf{x}|G) = \underbrace{p(\mathbf{y}|\mathbf{C}, \mathbf{x}, G)}_{\text{likelihood}} \underbrace{p(\mathbf{C}|G)}_{\text{prior on } \mathbf{C}} \underbrace{p(\mathbf{x}|G)}_{\text{prior on } \mathbf{x}}$$

where G = neighbourhood graph.

- $\mathbf{V}, \alpha$: model parameters

Likelihood: $p(\mathbf{y}|\mathbf{C}, \mathbf{x}, G)$

Penalize the approximation error under the locally linear assumption.

$$\log p(\mathbf{y}|\mathbf{C}, \mathbf{x}, G, \mathbf{V})$$
$$= -\frac{\epsilon}{2}\left\|\sum_{i=1}^{n}\mathbf{y}_i\right\|^2 - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\eta_{ij}\left(\Delta_{\mathbf{y}_{j,i}} - \mathbf{C}_i\Delta_{\mathbf{x}_{j,i}}\right)^\top \mathbf{V}^{-1}\left(\Delta_{\mathbf{y}_{j,i}} - \mathbf{C}_i\Delta_{\mathbf{x}_{j,i}}\right) - \log Z_{\mathbf{y}}$$

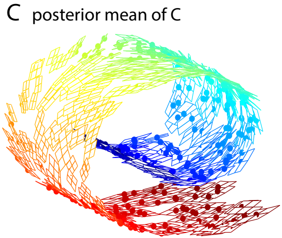
- $\Delta_{\mathbf{y}_{j,i}} := \mathbf{y}_j - \mathbf{y}_i$ and $\Delta_{\mathbf{x}_{j,i}} := \mathbf{x}_j - \mathbf{x}_i$
- $\eta_{ij} = (G)_{ij} \in \{0, 1\}$. If points i, j are neighbours, $\eta_{ij} = 1$.
- $Z_{\mathbf{y}}$ = normalizer
- $\mathbf{V}^{-1}$ = parameter to learn
- $p(\mathbf{y}|\mathbf{C}, \mathbf{x}, G)$ = normal distribution in $\mathbf{y} = (\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top)^\top$.

Prior on $\mathbf{x}$ (latent) and $\mathbf{C}$ (linear maps)

$$\log p(\{\mathbf{x}_1,\dots,\mathbf{x}_n\}|G,\alpha) = \overbrace{-\frac{\alpha}{2}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2}^{\{\mathbf{x}_i\}_i \text{ not too large}} \overbrace{-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\eta_{ij}\|\mathbf{x}_i-\mathbf{x}_j\|^2}^{\text{neighbours} \implies \text{similar latent}} - \log Z_{\mathbf{x}}$$

$$\log p(\{\mathbf{C}_1,\dots,\mathbf{C}_n\}|G) = -\frac{\epsilon}{2}\left\|\sum_{i=1}^{n}\mathbf{C}_i\right\|_F^2 - \frac{1}{2}\underbrace{\overbrace{\sum_{i=1}^{n}\sum_{j=1}^{n}\eta_{ij}\|\mathbf{C}_i-\mathbf{C}_j\|_F^2}^{\text{neighbours} \implies \text{similar maps}}}_{\Downarrow} - \log Z_{\mathbf{C}}$$

- α = parameter to learn.
- $Z_{\mathbf{x}}, Z_{\mathbf{C}}$ = normalizers
- $p(\mathbf{x}|G,\alpha)$ = normal distribution in $\mathbf{x} = (\mathbf{x}_1^\top,\dots,\mathbf{x}_n^\top)^\top$.
- $p(\mathbf{C}|G)$ = matrix normal distribution in $\mathbf{C} = (\mathbf{C}_1,\dots,\mathbf{C}_n)$.



Variational Inference

- Infer $q(\mathbf{C}, \mathbf{x}) \approx p(\mathbf{C}, \mathbf{x}|\mathbf{y})$ and learn $\theta = \{\alpha, \mathbf{V}^{-1}\}$.
- Maximize evidence lowerbound (ELBO) $\mathcal{L}(q, \theta)$:

$$\log p(\mathbf{y}|G, \theta) \geq \iint q(\mathbf{C}, \mathbf{x}) \log \frac{p(\mathbf{y}, \mathbf{C}, \mathbf{x}|G, \theta)}{q(\mathbf{C}, \mathbf{x})} \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{C} := \mathcal{L}(q(\mathbf{C}, \mathbf{x}), \theta).$$

- Assume $q(\mathbf{C}, \mathbf{x}) = q(\mathbf{C})q(\mathbf{x})$. Use variational Bayes.

1. **Variational E:**

$$q(\mathbf{x}) \propto \exp\left[\int q(\mathbf{C}) \log p(\mathbf{y}, \mathbf{C}, \mathbf{x}|G, \theta)\, \mathrm{d}\mathbf{C}\right] \quad \text{(normal distribution)}$$

$$q(\mathbf{C}) \propto \exp\left[\int q(\mathbf{x}) \log p(\mathbf{y}, \mathbf{C}, \mathbf{x}|G, \theta)\, \mathrm{d}\mathbf{x}\right] \quad \text{(matrix normal distribution)}$$

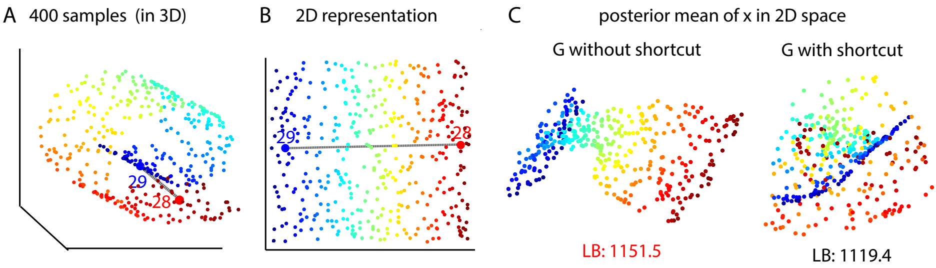
2. **Variational M:**

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(q(\mathbf{C}, \mathbf{x}), \theta)$$

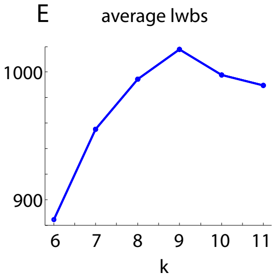
3. Repeat 1, 2

Experiment 1: Detecting a Graph Shortcut

- LL-LVM requires as input a neighbourhood graph G.
- ELBO can be used to evaluate a hypothetical G.

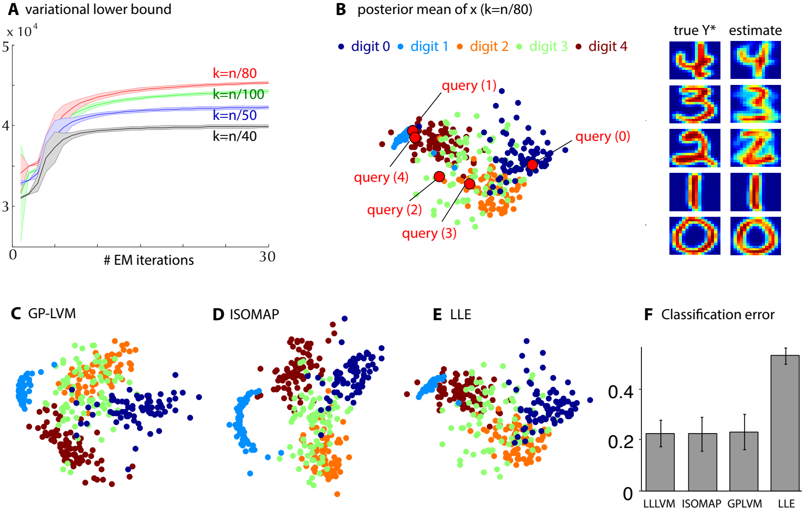


- LB (lower bound) = ELBO value.
- Fig. C: G with a shortcut $\implies$ lower ELBO.
- Fig. E: Choose the right k in k-NN graph construction.



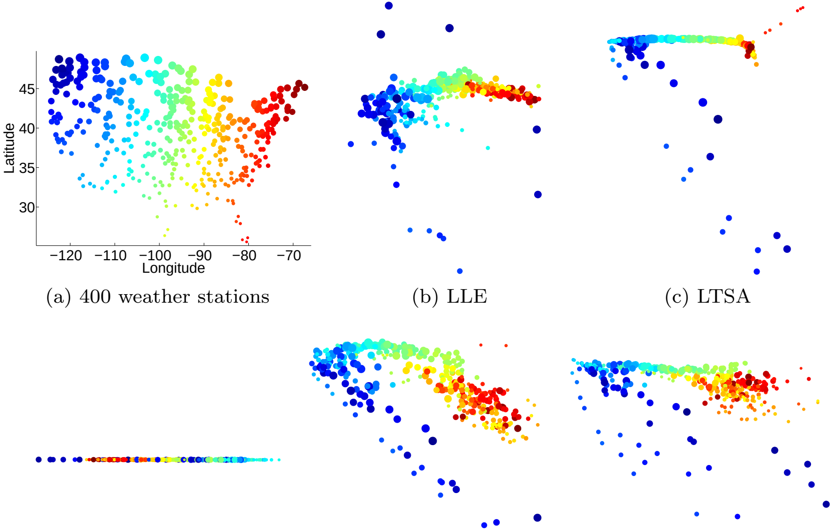
Experiment 2: Modelling USPS Handwritten Digits

- $n = 400, d_y = 256$. Reduce to $d_x = 2$.



- **Top-right**: Draw from $p(\mathbf{y}_i | \mathbf{C}, \mathbf{x}, \text{other } \mathbf{y}\text{'s})$
- Classify with 1-NN.

Experiment 3: Modelling Climate Data



(a) 400 weather stations (b) LLE (c) LTSA

(d) ISOMAP (e) GP-LVM (f) LL-LVM

- $\mathbf{y}_i$ = 12-d vector of monthly precipitation measurements at i^{th} location.
- $n = 400$. Use 12-NN to construct G.

Conclusion

- New probabilistic approach to manifold learning.
- Assumption: locally linear manifold
- LL-LVM:
  - preserve local geometries
  - uncertainty estimates
  - principled way to evaluate a neighbourhood structure (with ELBO)
  - easy inference
- Matlab code available: https://github.com/mijungi/lllvm.
- **Future work:** Learn neighbourhood graph G.

References I

Lawrence, N. (2003).
Gaussian process latent variable models for visualisation of high dimensional data.
In NIPS, pages 329–336.