Latent-variable representations for speech processing and research

Miguel Á. Carreira-Perpiñán
Dept. of Neuroscience, Georgetown University Medical Center
miguel@giccs.georgetown.edu
http://www.giccs.georgetown.edu/~miguel
Contents

- Motivation
- Continuous latent variable models
- Dimensionality reduction of EPG data
- Reconstruction of sequential data
  - Acoustic-to-articulatory mapping problem
Continuous latent variable models

Aim: to infer a low-dimensional representation of an observed, high-dimensional process.

Prior $p(x)$

Induced $p(t)$

Manifold $\mathcal{M} = f(\mathcal{X})$

Latent space $\mathcal{X}$ of dimension $L$

Observed space $\mathcal{T}$ of dimension $D$

The data distribution in observed space $\mathcal{T}$ is modelled using a low-dimensional representation in latent space $\mathcal{X}$. In the figure the observed space consists of the positions of vocal articulators and the acoustic waveform features.
Continuous latent variable models (cont.)

- The prior in latent space $p(x)$, the mapping $f(x)$ and the noise model in observed space $p(t|x)$ are equipped with parameters $\Theta$.

- Marginalisation in latent space (often difficult): $p(t) = \int p(t|x)p(x)\,dx$
  - $f$ linear, $p(t|x)$ normal centred in $t = f(x)$, $p(x)$ normal: analytically solvable
  - $f$ nonlinear: analytically insolvable; approximate integral by a Monte Carlo method.

- Maximum likelihood parameter estimation from sample $\{t_n\}_{n=1}^N$, usually via an EM algorithm:
  $$\arg\max_\Theta \mathcal{L}(\Theta) = \sum_{n=1}^N \log p(t_n|\Theta).$$

- Dimensionality reduction mapping $F$ via posterior distribution in latent space:
  $$p(x|t) = \frac{p(t|x)p(x)}{p(t)} \sim x = F(t).$$
## Continuous latent variable models (cont.)

| Model                       | Prior in latent space $p(x)$ | Mapping $f: x \rightarrow t$ | Noise model $p(t|x)$ | Density in observed space $p(t)$ |
|-----------------------------|------------------------------|-------------------------------|----------------------|----------------------------------|
| Factor analysis (FA)        | $\mathcal{N}(\mathbf{0}, \mathbf{I})$ | linear                        | diagonal normal      | constrained Gaussian             |
| Principal component analysis (PCA) | $\mathcal{N}(\mathbf{0}, \mathbf{I})$ | linear                        | spherical normal     | constrained Gaussian             |
| Independent component analysis (ICA) | unknown but factorised | linear                        | Dirac delta          | depends                          |
| Independent factor analysis (IFA) | product of 1D Gaussian mixtures | linear                        | normal               | constrained Gaussian mixture     |
| Generative topographic mapping (GTM) | discrete uniform         | generalised linear model      | spherical normal     | constrained Gaussian mixture     |

Problems of latent variable models:

- The latent dimension $L$ must be fixed in advance—could use model selection.
- The complexity of GTM is $\mathcal{O}(e^L)$. 
Electropalatography

A plastic pseudopalate fitted to a person’s mouth detects the presence or absence of contact between the tongue and the palate in 62 different locations during an utterance (sampled at 200 Hz).

Representative EPG frames for the typical stable phase of different phonemes, pictured rowwise from alveoli (top) to velum (bottom).

62-dimensional signal sampled at 200 Hz: data reduction is necessary (traditionally performed with fixed linear indices).
Factors / prototypes of the models

Factor analysis

PCA

Mixture of factor analysers

Bernoulli mixture

Speaker RK, ACCOR database.
Log-likelihood and reconstruction error of the models

Speaker RK, ACCOR database.

Miguel Á. Carreira-Perpiñán

Latent-variable representations for speech processing and research
Two-dimensional representation of EPG data

Trajectory in latent space of the highlighted utterance fragment “I prefer Kant to Hobbes for a good bedtime book” (/əɪ prɪˈfɛr ˈkænt tu ˈhɒbz fɔr ə ˈɡʊd ˈbedtɪm ˈbʊk/). Speaker RK, ACCOR database.

Suggests a low intrinsic dimensionality of the EPG data!
The abrupt transition from /æ/ to /ɪt/ in “Kant to” (frames 335–336) produces a discontinuity in latent space.


Matlab code at http://www.giccs.georgetown.edu/~miguel.
The acoustic-to-articulatory mapping problem

The positions, speeds, etc. of supraglottal mechanical elements of the vocal tract (tongue, jaw, etc.) are called articulatory variables and are continuous functions of the time.

A vocal tract configuration produces a unique acoustic signal. Thus the mapping articulatory $\rightarrow$ acoustic is univalued (forward problem).

An acoustic signal can be produced by different vocal tract configurations. The mapping acoustic $\rightarrow$ articulatory is multivalued (inverse problem).

Main approaches:

- Dynamic programming search in a large articulatory codebook: best performance, but very slow.
- Neural network (trained with the codebook): faster and more compact, but worse performance.
- Carefully prepared assembly of neural networks: approaches codebook performance and is fast.
The acoustic-to-articulatory mapping problem (cont.)

- Internal inverse models for motor control:
  - Motor theory of speech perception (Liberman et al. 1967, 1985): postulated the existence in the brain of a motor code of phonetic gestures and a translation module between that code and the acoustics.
  - Arm control, locomotion, etc.

- Biologically, not clear what kind of internal representations support motor control.

- But models based on latent variables are computationally attractive:
  - Gestures in articulatory phonology (Browman and Goldstein, 1992).
  - More parsimonious (low transmission rate), less variable (coarticulation) than the acoustics.

- Addition of production information improves automatic speech recognition over acoustic HMMs:
  - Augment acoustic vector with articulatory vector (requires articulatory inversion!).
  - Analysis-by-synthesis models (Bakis, hidden dynamic models).

The acoustic-to-articulatory mapping is a particular case of reconstruction of missing data.
Reconstruction of missing data: applications

Acoustic-to-articulatory mapping with the Wisconsin X-ray microbeam database.

Audiovisual mappings for multimodal speech processing: face $\leftrightarrow$ acoustics.

Reconstruction of occluded speech (e.g. in noisy environments) from acoustics alone.

Decoding neural population activity: reconstruction from hippocampal place cells.

Inverse kinematics and dynamics of a redundant manipulator.

Wind velocity vector retrieval from satellite scatterometer data.
Definition of the problem of data reconstruction

Given a sequence \( \{ t^{(1)}, \ldots, t^{(N)} \} \subset \mathbb{R}^D \) where part of the data are missing, reconstruct the whole sequence to minimise an error criterion.

Assume \( \{ t^{(n)} \}_{n=1}^{N} \) are samples from a continuous function \( \tilde{f} \) of an independent variable \( z \) (the experimental conditions) at points \( \{ z^{(n)} \}_{n=1}^{N} \).
### Definition of the problem of data reconstruction (cont.)

<table>
<thead>
<tr>
<th>Examples of problem</th>
<th>( z )</th>
<th>( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory of a mobile point</td>
<td>time, 1D</td>
<td>spatial coordinates, 3D (Cartesian, spherical)</td>
</tr>
<tr>
<td>Spoken utterance</td>
<td>time, 1D</td>
<td>speech feature vector, 13D (e.g. LSPs, MFCCs, PLPs)</td>
</tr>
<tr>
<td>Wind field on the ocean surface</td>
<td>spatial coordinates, 2D</td>
<td>wind velocity vector, 2D</td>
</tr>
</tbody>
</table>

Reconstruction is possible due to two kinds of redundancy in the data:

- **Pointwise**: low intrinsic dimensionality of observed variables \( t_1, \ldots, t_D \) due to correlations.
- **Across a whole sequence**: due to the continuity of \( t = \mathcal{F}(z) \).

Reconstruction can be a previous step to some further processing, e.g. classification.

Types of reconstruction:

- **Pointwise (or local)**: reconstruction of a vector \( t^{(n)} \) given information present in \( t^{(n)} \) only.
- **Multiple pointwise**: like pointwise, but several candidate reconstructions for \( t^{(n)} \) are provided.
- **Global**: reconstruction of the whole sequence \( \{t^{(n)}\}_{n=1}^N \) given information present in it.
Related problems and approaches

Statistical mapping approximation and inversion:

- Construct function $x \xrightarrow{\mathcal{E}} y$ given input-output pairs $\{(x_n, y_n)\}_{n=1}^N$.
- Efficiently solved by universal function approximators, e.g. MLPs.
- $L_2$-optimal mapping: conditional mean or regression $E \{y|x\}$.
- Can’t deal with multivalued mappings (e.g. inverses)
- generic patterns of missing data.
- Complex, unreliable extensions: ensembles, recurrent nets.

Imputation modelling (statistical analysis of missing data):

- Aims not at predicting the missing values with the greatest accuracy, but at creating plausible imputations of them to properly reflect uncertainty and preserve important aspects of the data distribution.
- Then, perform complete-data analysis to produce inferential statements (e.g. interval estimates or $p$-values).
- Based on a predictive distribution of the missing data given the present data (obtained from a model for the complete data).
- Examples: mean imputation, multiple imputation.

Vector quantisation and dynamic programming: particular case of our method.

Sequential modelling: model temporal evolution rather than reconstruction (HMMs, Kalman filters, time series, etc.).
My approach in a nutshell

- Joint density model $p(t)$ of the observed variables $t_1, \ldots, t_D$ estimated offline from a (complete) training set $\{t_n\}_{n=1}^N$.

- At reconstruction time, given an incomplete sequence $\{t^{(n)}\}_{n=1}^N$:
  - Multiple pointwise reconstruction at point $n$ by the modes of the conditional distribution
    $$p(\text{missing variables}^{(n)} | \text{present variables}^{(n)}).$$
  - The reconstructed sequence is obtained by minimizing a global continuity constraint using dynamic programming.
Offline estimation of joint density model

- We want to estimate a parametric model for the pdf of the observed variables $t_1, \ldots, t_D$ from a training set consisting of (complete) data $\{t_n\}_{n=1}^N$ not necessarily sequential.

- Several techniques exist with universal approximation properties: mixtures of Gaussians, kernel density estimation, etc.

- It is also possible to estimate the model from incomplete data using an EM algorithm.

- It ought to allow efficient computation of conditional probabilities of the form $p(t_3 t_5 | t_1 t_2 t_4)$ and be able to represent multimodal conditional distributions.

- The low intrinsic dimensionality of the data suggests using latent variable models (but other models could be used too).
Deriving a functional relation from a conditional distribution

Consider a conditional distribution \( f(y) \overset{\text{def}}{=} p(y|x = x_0) \). How much does \( x = x_0 \) determine \( y \)?

\( f \) is sparse if it basically assigns probability mass to a finite set of \( y \)s (to a low-dimensional manifold of \( y \)-space, in general). Sparse distributions are very informative: \( y \) is strongly determined by \( x \).

Quantitative measure (not ideal . . . ) of sparseness: negative entropy \( -h(f) \overset{\text{def}}{=} \int f(y) \log f(y) \, dy \).

If \( p(y|x) \) is sparse, we can pick representative points of it:

- Single choice (univalued mapping):
  - Mean: may not be good if the solution domain is not convex.
  - Global mode: not much better than the other modes.
- Multiple choice (multivalued mapping): select all the modes; one of them should be the correct one—knowing which one requires additional information.

For unimodal, symmetric distributions (e.g. factor analysis) nothing of this matters!
Mixtures of Gaussians: conditional distributions and entropy

Consider a \( D \)-dimensional mixture of Gaussians with \( M \) components:

\[
p(t) = \sum_{m=1}^{M} p(m)p(t|m) \quad p(t|m) \sim \mathcal{N}_{D}(\mu_m, \Sigma_m) \quad t \in \mathbb{R}^D.
\]

Any conditional distribution of a mixture of Gaussians is a mixture of Gaussians itself (where many mixture components can be discarded), trivially computable if the components are diagonal:

\[
p(t_M|t_P) = \sum_{m=1}^{M} p(m|t_P) p(t_M|m) \quad p(t_M|m) \sim \mathcal{N}_{J}(\mu_{m,M}, \Sigma_{m,M,M}) \quad t_M \in \mathbb{R}^J.
\]

Entropy: no exact formula, but bounds: \( \max(LB_1, LB_2) \leq h(p(t)) \leq UB_1 \), where:

\[
LB_1 \overset{\text{def}}{=} \frac{1}{2} \log \left\{ (2\pi e)^{D} \prod_{m=1}^{M} |\Sigma_m|^{\pi_m} \right\} \quad UB_1 \overset{\text{def}}{=} \frac{1}{2} \log \left( (2\pi e)^{D} |\Sigma| \right)
\]

\[
LB_2 \overset{\text{def}}{=} -\log \left\{ \sum_{m,n=1}^{M} p(m)p(n) |2\pi(\Sigma_m + \Sigma_n)|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mu_m - \mu_n)^T(\Sigma_m + \Sigma_n)^{-1}(\mu_m - \mu_n)} \right\}
\]

and \( \Sigma = \sum_{m=1}^{M} \pi_m \Sigma_m + \sum_{m=1}^{M} \pi_m \mu_m \mu_m^T - \mu \mu^T \) is the mixture covariance.
Gaussian mixtures: exhaustive mode finding

Aim: find all the modes of the Gaussian mixture \( p(t) \), \( t \in \mathbb{R}^D \) (no direct methods exist). \( p(t) \) has at most \( M \) modes and they all lie in the convex hull of the mean vectors \( \{ \mu_m \}_{m=1}^M \). Algorithm: use an iterative maximisation method starting from each mean vector:

- **Gradient descent**: \( t^{(\tau+1)} = t^{(\tau)} + s g(t^{(\tau)}) \) with the gradient
  \[
g = \nabla p(t) = \sum_{m=1}^{M} p(t, m) \Sigma_m^{-1} (\mu_m - t).
\]

- **Quadratic maximisation**: \( t^{(\tau+1)} = t^{(\tau)} - \mathbf{H}^{-1}(t^{(\tau)})g(t^{(\tau)}) \) with the Hessian
  \[
  \mathbf{H} \overset{\text{def}}{=} (\nabla \nabla^T) p(t) = \sum_{m=1}^{M} p(t, m) \Sigma_m^{-1} \left( (\mu_m - t)(\mu_m - t)^T - \Sigma_m \right) \Sigma_m^{-1}.
  \]

- **Fixed-point iteration**: \( t^{(\tau+1)} = f(t^{(\tau)}) \) with
  \[
  f(t) \overset{\text{def}}{=} \left( \sum_{m=1}^{M} p(m|t) \Sigma_m^{-1} \right)^{-1} \sum_{m=1}^{M} p(m|t) \Sigma_m^{-1} \mu_m.
  \]

From the stationary points found, remove minima, saddle points and coincident maxima. This requires some care because of the finite numerical precision.
Gaussian mixtures: exhaustive mode finding (cont.)

- The formulas simplify considerably for $\Sigma_m = \sigma^2 I$ (GTM's case). In particular, the fixed-point iteration becomes $t^{(\tau+1)} = \sum_{m=1}^{M} p(m|t^{(\tau)}) \mu_m$.
- Confidence sets: approximate the pdf at each mode by a Gaussian (using the Hessian).
- Other applications:
  - Clustering, e.g. galaxy substructure.
  - Posterior modes in Bayesian analysis.

Matlab code at [http://www.giccs.georgetown.edu/~miguel](http://www.giccs.georgetown.edu/~miguel).
Global reconstruction: constraining the solutions

- Applying multiple pointwise reconstruction to each point in the sequence produces a layered graph which must be traversed from left to right (or vice versa):

\[
\prod_{n=1}^{N} \nu_n \sim \mathcal{O}(\nu^N)
\]

possible paths

- To select a unique candidate for each point we need external info. For continuous sequential data, we apply a local continuity constraint: consecutive points must be close to each other.

- Define a distance \(d(x, y)\), e.g. the weighted Euclidean distance \(\sum_{d=1}^{D} w_d (x_d - y_d)^2\).

- Extend to a global continuity constraint: \(\mathcal{C} \left( \{t^{(n)}\}_{n=1}^{N} \right) = \sum_{n=1}^{N-1} d(t^{(n)}, t^{(n+1)})\). If \(d\) is the Euclidean distance then \(\mathcal{C}\) is the curve length: shortest path algorithm.

- In the mapping inversion case, add a fitness term based in the forward mapping \(t_M \xrightarrow{g} t_P\):

\[
\mathcal{C} \left( \{t^{(n)}\}_{n=1}^{N} \right) = \lambda \sum_{n=1}^{N-1} d(t^{(n)}, t^{(n+1)}) + \sum_{n=1}^{N} d \left( \left( \begin{array}{c} t_P^{(n)} \\ t_M^{(n)} \end{array} \right), \left( \begin{array}{c} g(t_M^{(n)}) \\ t_M^{(n)} \end{array} \right) \right).
\]

continuity fitness

Candidates to reconstruct \(t^{(n)}\)
Global reconstruction: constraining the solutions (cont.)

- The search problem
  \[
  \min_{\text{all combinations}} \mathcal{C} \left( \{ t^{(n)} \}_{n=1}^{N} \right)
  \]
can be efficiently solved by dynamic programming in \( O(N \nu^2) \) time. A suboptimal solution can be found even faster by a greedy algorithm.

- Additional information may be easily included, e.g. bounds on the missing variables given by the present ones.

- The dynamic programming search can break down if there are isolated discontinuities:
  - due to artifacts, such as undersampling
  - due to the intrinsic nature of the data (e.g. fronts in wind fields).

- The arc-length constraint is independent of the speed at which the sequence is travelled (i.e., to reparametrisations of it) and so is robust to time warping (e.g. fast/slow speech).
Experiments with a toy problem

Consider one-dimensional data \((L = 1)\) observed in \(D = 2\) dimensions, i.e., \(t = (t_1, t_2)\):

Curve \(t = (x, x + 3\sin x)\) for \(x \in [-2\pi, 2\pi]\).

Point cloud \(\{t_n\}_{n=1}^N\), sampled from the curve with additive \(\mathcal{N}(0, \sigma^2 I)\) noise.

Univalued forward mapping \(t_2 = g(t_1) = t_1 + 3\sin t_1\) but multivalued inverse \(t_1 = g^{-1}(t_2)\).
Demonstration of the method

$$t_2^{(n)}$$

$$t_1^{(n+2)}$$

all modes & constraints

mean

global mode
Density models used

Factor analysis: $L = 1$, 6 parameters, log-likelihood $= -4807$

GTM: $L = 1$, $K = 200$, 21 parameters, log-likelihood $= -3109$
# Masks

### P0
Complete sequence (0% missing)

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-6.1</td>
<td>-5.6</td>
<td>-4.5</td>
<td>-4.4</td>
<td>-3.7</td>
<td>-3.1</td>
<td>-2.2</td>
<td>-1.8</td>
<td>-0.7</td>
<td>-0.8</td>
<td></td>
</tr>
<tr>
<td>-6.5</td>
<td>-3.8</td>
<td>-1.9</td>
<td>-1.4</td>
<td>-2.0</td>
<td>-3.4</td>
<td>-4.3</td>
<td>-4.8</td>
<td>-3.7</td>
<td>-1.1</td>
<td></td>
</tr>
</tbody>
</table>

### P1
$t_2$ always missing: regression $t_1 \to t_2$ (50% missing)

### P2
$t_1$ always missing: regression $t_2 \to t_1$ (50% missing)

### P3
$t_1$ or $t_2$ missing at random (75% missing)

### P4
$t_1$ or $t_2$ missing at random (50% missing)

### P5
$t_1$ or $t_2$ missing at random (25% missing)

### P6
One of \{$t_1$, $t_2$\} missing at random (50% missing)

### P7
Random blocks of missing data (10% missing)
Reconstruction results: forward problem (mask P1)

Factor analysis and multilayer perceptron

GTM with $K = 200$, various methods

Miguel Á. Carreira-Perpiñán

Latent-variable representations for speech processing and research
Reconstruction results: inverse problem (mask P2)

Factor analysis and multilayer perceptron

GTM with $K = 200$, various methods
Rec. results: noisy trajectory, 50% missing data (mask P4)

Factor analysis and $\text{GTM}_K=200$ (mean)

NB: the multilayer perceptron can’t deal with varying patterns of missing data.
Reconstruction results: average squared error

Average squared error \( \frac{1}{N} \sum_{n=1}^{N} \| t_n - \hat{t}_n \|^2 \) for a sequence with \( N = 100 \) points.

<table>
<thead>
<tr>
<th>mask (% missing)</th>
<th>Factor analysis</th>
<th>MLP</th>
<th>GTM with ( \tilde{K} = 200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>mean</td>
</tr>
<tr>
<td>P1 (50%)</td>
<td>3.8011</td>
<td>0.0129</td>
<td>0.0196</td>
</tr>
<tr>
<td>P2 (50%)</td>
<td>4.2702</td>
<td>2.1475</td>
<td>2.1184</td>
</tr>
<tr>
<td>P3 (76%)</td>
<td>15.5385</td>
<td></td>
<td>14.6874</td>
</tr>
<tr>
<td>P4 (56%)</td>
<td>11.4116</td>
<td></td>
<td>9.7848</td>
</tr>
<tr>
<td>P5 (25%)</td>
<td>2.9049</td>
<td></td>
<td>1.7891</td>
</tr>
<tr>
<td>P6 (50%)</td>
<td>4.1377</td>
<td></td>
<td>0.9555</td>
</tr>
<tr>
<td>P7 ( 8%)</td>
<td>1.8476</td>
<td></td>
<td>1.5252</td>
</tr>
</tbody>
</table>

Problem type (mask) | Low error | High error |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward (P1)</td>
<td>cmode = dpmode = mean = MLP = gmode = rmode &lt; FA</td>
<td></td>
</tr>
<tr>
<td>Inverse (P2)</td>
<td>cmode ( \lesssim ) dpmode ( \ll ) mean = MLP &lt; gmode &lt; FA &lt; rmode</td>
<td></td>
</tr>
<tr>
<td>General (P3–P7)</td>
<td>cmode &lt; dpmode ( \ll ) mean ( \ll ) gmode = rmode = FA</td>
<td></td>
</tr>
</tbody>
</table>
Reconstruction results: average squared error (cont.)

- \( \text{gmode} \) can become \( \text{rmode} \) depending on the implementation of the maximisation algorithm.

- Other mode-based strategies:
  - Greedy algorithm implementation of constraint: large error.
  - Picking \( S \) samples from the conditional distribution: large error.
  - Combination of mean (if the conditional distribution is unimodal) and \( \text{dpmode} \) (if it is multi-modal): nearly no change over \( \text{dpmode} \).

- When does \( \text{dpmode} \) break down?
  - For non-sequential data (e.g. a shuffled sequence): the continuity condition is not satisfied.
  - When the number of points in the sequence \( N \) is very large compared with the data noise: oversampling (for regression-type problems only).
  - When the joint density model is inaccurate, notably when it lacks smoothness (for regression-type problems only).
Smoothness of the density model

GTM: $L = 1$, $K = 20$, 21 parameters, log-likelihood = $-3468$

Conditional distribution $p(t_2|t_1 = 5.39)$ for each model
Smoothness of the density model (cont.)

Mixture of $M = 20$ full-covariance Gaussians: 119 parameters, log-likelihood = $-3053$

Conditional distribution $p(t_2|t_1 = 5.39)$ for various mixtures
Reconstruction results: forward problem (mask P1)

Cond. distrib. $t_2 | t_1$ for several values of $t_1$

Model: GTM with $K = 20$ Gaussian components.
Rec. error: forward problem (mask P1)

GTM models: $K = 20$ (cyan), $K = 60$ (green) and $K = 200$ (red).
Smoothness of the density model: discussion

- The modes are very sensitive to the joint density model, while the mean is more robust.
- If the density model is not smooth, the conditional distribution presents spurious modes which may give rise to wrong solutions of the dynamic programming search.
- The increase of reconstruction error is very small for general patterns of missing data (P3–P7), but can be large for regression problems (P1, P2).
- Removing “low-probability” modes does not always help.
- For GTM, increasing the number of mixture components $K$ smoothens the density model at a higher computational cost. For mixtures of full-covariance Gaussians, overfitting limits the number of components.
Inverse kinematics of a robot arm

Two-link, planar robot arm of joint angles $(\theta_1, \theta_2)$ and end-effector position $(x_1, x_2)$

Forward mapping $(\theta_1, \theta_2) \xrightarrow{g} (x_1, x_2)$ with multivalued inverse (elbow-up, elbow-down):

\[
\begin{align*}
    x_1 &= l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\
    x_2 &= l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)
\end{align*}
\]
Reconstruction results for the robot arm problem

Average squared error \( \frac{1}{N} \sum_{n=1}^{N} \| t_n - \hat{t}_n \|^2 \) for a sequence with \( N = 34 \) points.

<table>
<thead>
<tr>
<th>mask (% missing)</th>
<th>Factor analysis</th>
<th>MLP</th>
<th>GTM with ( K = 225 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>mean</td>
</tr>
<tr>
<td>P1 (50%)</td>
<td>0.0130</td>
<td>0.0007</td>
<td>0.0014</td>
</tr>
<tr>
<td>P2 (50%)</td>
<td>0.7690</td>
<td>0.6371</td>
<td>0.6767</td>
</tr>
<tr>
<td>P3 (74%)</td>
<td>0.7369</td>
<td></td>
<td>0.7084</td>
</tr>
<tr>
<td>P4 (50%)</td>
<td>0.4136</td>
<td></td>
<td>0.3655</td>
</tr>
<tr>
<td>P5 (25%)</td>
<td>0.2297</td>
<td></td>
<td>0.1486</td>
</tr>
<tr>
<td>P6 (50%)</td>
<td>0.4033</td>
<td></td>
<td>0.3511</td>
</tr>
<tr>
<td>P7 (5%)</td>
<td>0.1964</td>
<td></td>
<td>0.1940</td>
</tr>
</tbody>
</table>

These results were obtained with only the continuity constraint, without using knowledge of the forward mapping.
Reconstruction method: summary

General approach to reconstruction of sequential data based in:

- **Multiple pointwise reconstruction** via a joint density model of the observed variables, using exhaustive mode finding in the conditional distributions (here implemented with latent variable models or Gaussian mixtures): “modal regression.”

- **Sequence reconstruction** via a continuity constraint on the set of pointwise candidates (here implemented with a weighted Euclidean distance and dynamic programming).

The modes of the conditional distribution contain the potentially correct values. The continuity constraint—in those cases where it is applicable—may recover the actually correct ones.

Reconstruction method: summary (cont.)

- **Advantages:**
  - Applicable to varying patterns of missing data: the variables are treated symmetrically.
  - Deals by design with multivalued mappings rather than converting them into univalued ones.
  - Abstract: no physical knowledge of the problem is required, e.g. no need for a forward mapping—as long as powerful models are used.
  - Insensitive to time warping.
  - Can give confidence regions for each reconstructed value.
  - Does not suffer from numerical instability or out-of-domain problems.

- **Disadvantages:**
  - Sensitivity to the smoothness of the density model for regression problems.
  - With many missing data, the mode-finding step is computationally costly.
  - Difficulty of density estimation in high dimensions (the curse of the dimensionality).

- **Dynamical modelling:** unlike hidden Markov models or Kalman filters, this approach does not model the temporal evolution of the data.
Reconstruction method: extensions and further work

- Other types of constraints:
  - **Continuity**: first-order finite differences.
  - **Smoothness**: second-order finite differences (stronger condition).
  - **Quadratic**: minimum effort (for linear forces).

- Multidimensional constraints, e.g. reflecting spatial structure (1- to 3D) as well as temporal (1D).

- Unbounded horizon reconstruction: $N \to \infty$.

- Bump finding rather than mode finding.

- Efficient estimation of smooth density models.
### Notation

Vectors are boldface lowercase, scalars are italics lowercase, sets are calligraphic uppercase.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{t} = (t_1, \ldots, t_D) \in \mathcal{T} \subset \mathbb{R}^D$</td>
<td>$D$-dimensional vector in observed space $\mathcal{T}$</td>
</tr>
<tr>
<td>$\mathbf{x} = (x_1, \ldots, x_L) \in \mathcal{X} \subset \mathbb{R}^L$</td>
<td>$L$-dimensional vector in latent space $\mathcal{X}$</td>
</tr>
<tr>
<td>$\mathcal{P}, \mathcal{M} \in {1, \ldots, D}$</td>
<td>Sets of indices for variable selection. Example: if $\mathcal{P} = {1, 7, 3}$ and $\mathcal{M} = {2, 5}$ then:</td>
</tr>
<tr>
<td>$\mathbf{t}_\mathcal{P}$</td>
<td>is $\ (t_1, t_7, t_3)$</td>
</tr>
<tr>
<td>$\mathbf{t}_\mathcal{M}$</td>
<td>is $\ (t_2, t_5)$</td>
</tr>
<tr>
<td>$p(\mathbf{t}_\mathcal{P})$</td>
<td>is $\ p(t_1, t_3, t_7)$</td>
</tr>
<tr>
<td>$p(\mathbf{t}_\mathcal{M}</td>
<td>\mathbf{t}_\mathcal{P})$</td>
</tr>
<tr>
<td>$p(\mathbf{t}<em>\mathcal{P}, \mathbf{t}</em>\mathcal{M})$</td>
<td>is $\ p(t_1, t_2, t_3, t_5, t_7)$</td>
</tr>
<tr>
<td>$\mathbf{t}<em>\mathcal{P} \rightarrow \mathbf{t}</em>\mathcal{M}$</td>
<td>$\mathbf{t}<em>\mathcal{M}$ as a function of $\mathbf{t}</em>\mathcal{P}$</td>
</tr>
<tr>
<td>${\mathbf{t}<em>n}</em>{n=1}^N$</td>
<td>Set of $N$ vectors not necessarily in sequence</td>
</tr>
<tr>
<td>${\mathbf{t}^{(n)}}_{n=1}^N$</td>
<td>Set of $N$ vectors in (temporal) sequence</td>
</tr>
</tbody>
</table>