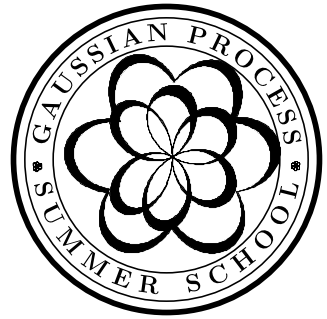
Deep Gaussian Processes

Neil D. Lawrence

11th July 2015
Deep Learning@ICML



Outline

Background

Massively Missing Data

Deep Gaussian Processes

Samples and Results

Ouroboros


Gaussian processes
2000
convolutional neural ne
1990
2010
Boltzman machines

Ouroboros

Gaussian processes
2000
1970
convolutional neural ne
1990
1960
Boltzman machines
2010
1980

Separation of Model and Algorithm

- Machine Learning:

$$\text{data} + \text{model} = \text{prediction}$$

- Model encodes our beliefs about the regularities of the universe.
- I'm using *simple* and *complex* in the sense of how easy to understand.
  - Neural network: complex model, simple algorithm
  - Gaussian process: simple model, complex algorithm

Is the Model Complex?

- More details in this blog post: `http://inverseprobability.com/2015/02/28/questions-on-deep-gaussian-processes/`
- This talk: It's about the model.
- Talk later today at Large Scale Kernel Machines is about *part* of the algorithm.

GaussianFace

(Lu and Tang, 2014)

- First system to surpass human performance on cropped Learning Faces in Wild Data.
  `http://tinyurl.com/nkt9a38`
- Lots of feature engineering, followed by a Discriminative GP-LVM.

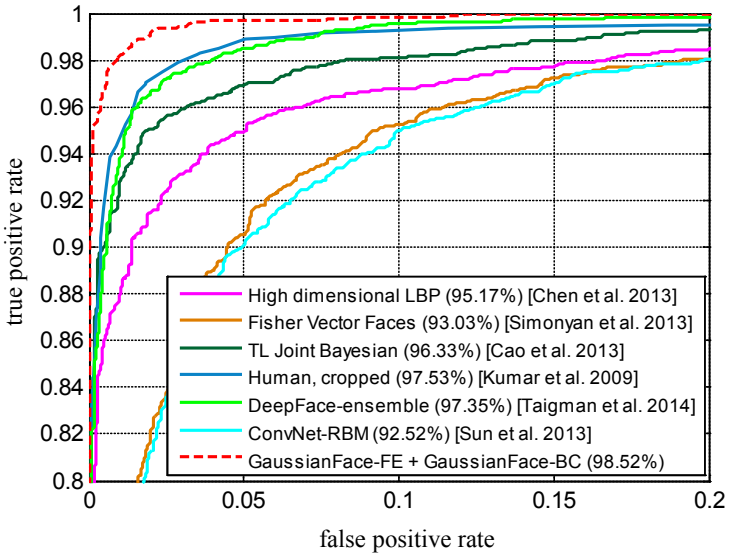


Figure 4: The ROC curve on LFW. Our method achieves the best performance, beating human-level performance.

Figure 5: The two rows present examples of matched and mismatched pairs respectively from LFW that were incorrectly classified by the GaussianFace model.

Conclusion and Future Work

Latent Variable

- The core component in GaussianFace is dimensionality reduction.
- Model data, $\mathbf{y}$, with a vector value function that maps from $\mathbf{x}$ to $\mathbf{y}$.

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i) + \epsilon$$

- Treat the unknown latent dimension $\mathbf{x}$ as a nuiscance parameter and place prior, $p(\mathbf{x})$ over it to integrate it out.
- In GaussianFace the latent variable model uses a Gaussian process to handle $\mathbf{f}$.

Difficulty for Probabilistic Approaches

- Propagate a probability distribution through a non-linear mapping.
- Normalisation of distribution becomes intractable.

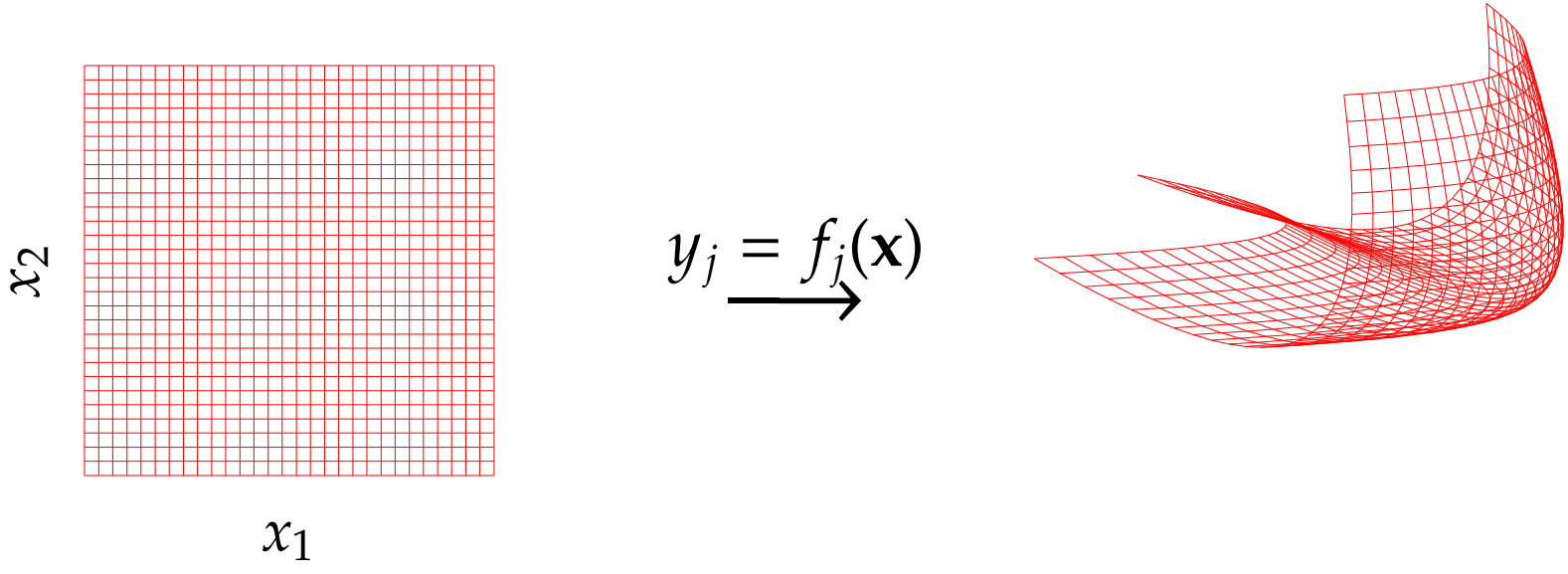


Figure : A three dimensional manifold formed by mapping from a two dimensional space to a three dimensional space.

Difficulty for Probabilistic Approaches

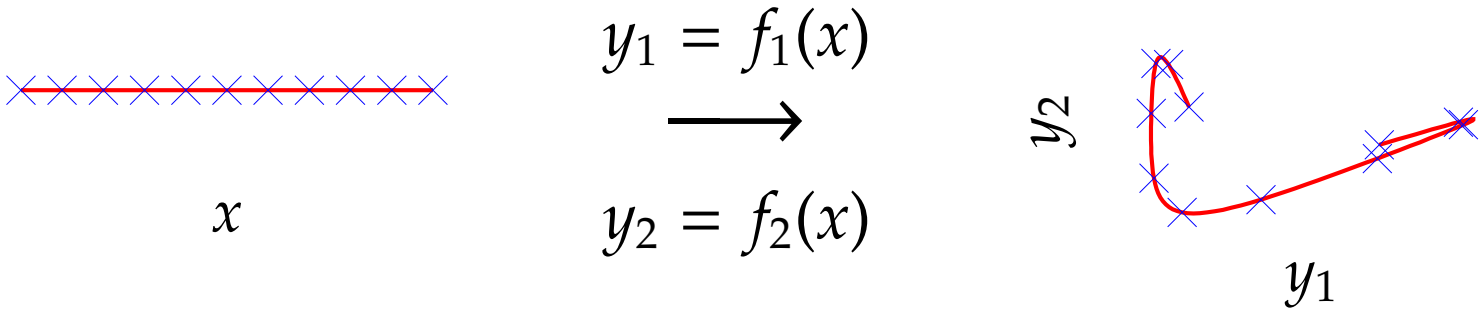


Figure : A string in two dimensions, formed by mapping from one dimension, x, line to a two dimensional space, $[y_1, y_2]$ using nonlinear functions $f_1(\cdot)$ and $f_2(\cdot)$.

Difficulty for Probabilistic Approaches

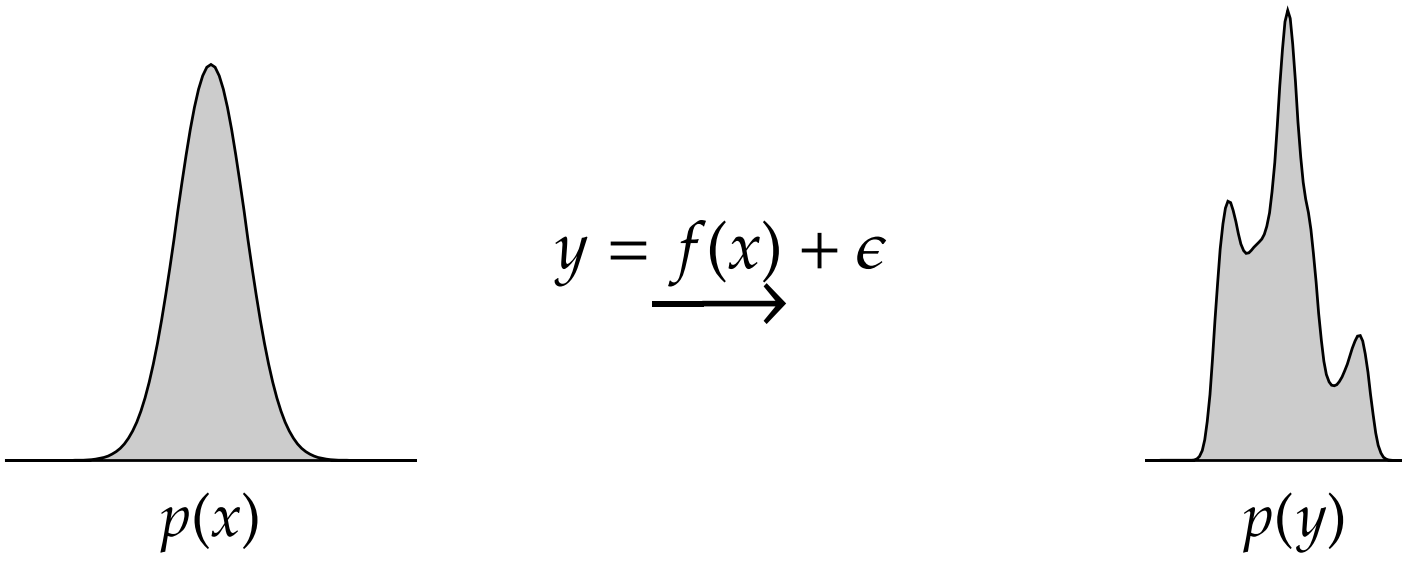
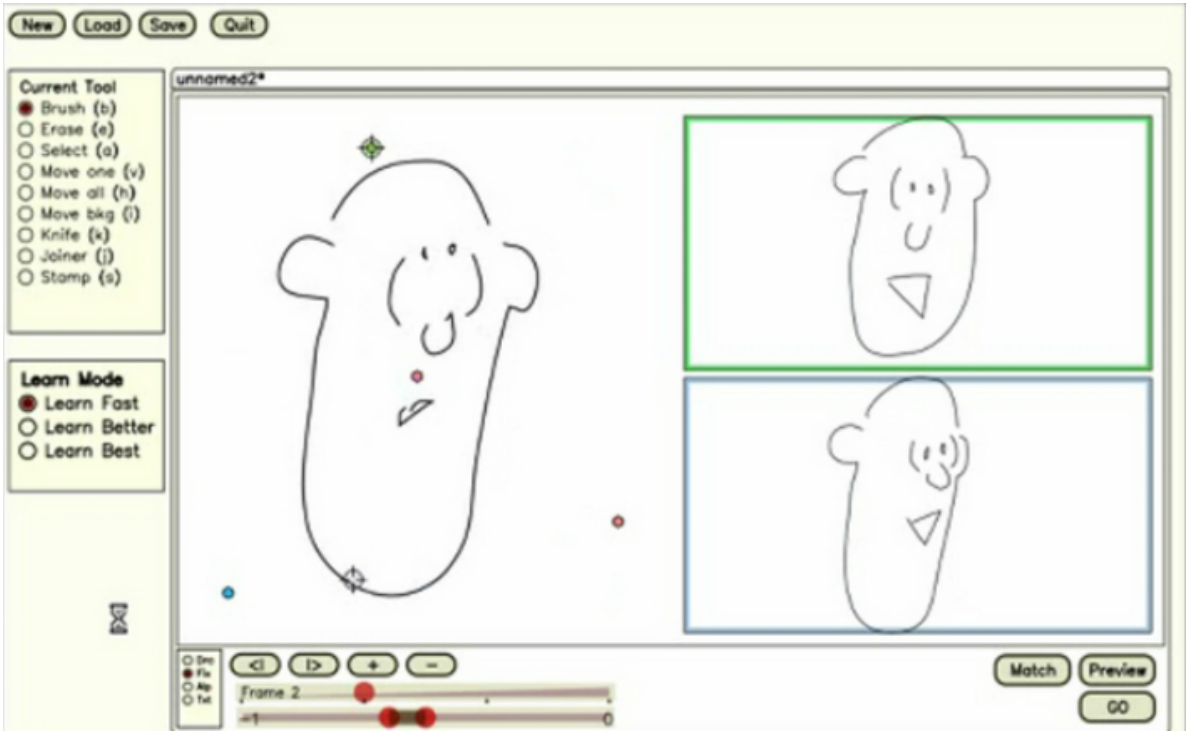


Figure : A Gaussian distribution propagated through a non-linear mapping. $y_i = f(x_i) + \epsilon_i$. $\epsilon \sim \mathcal{N}\left(0, 0.2^2\right)$ and $f(\cdot)$ uses RBF basis, 100 centres between -4 and 4 and $\ell = 0.1$. New distribution over y (right) is multimodal and difficult to normalize.

Example: Latent Doodle Space

(Baxter and Anjyo, 2006)



http://vimeo.com/3235882

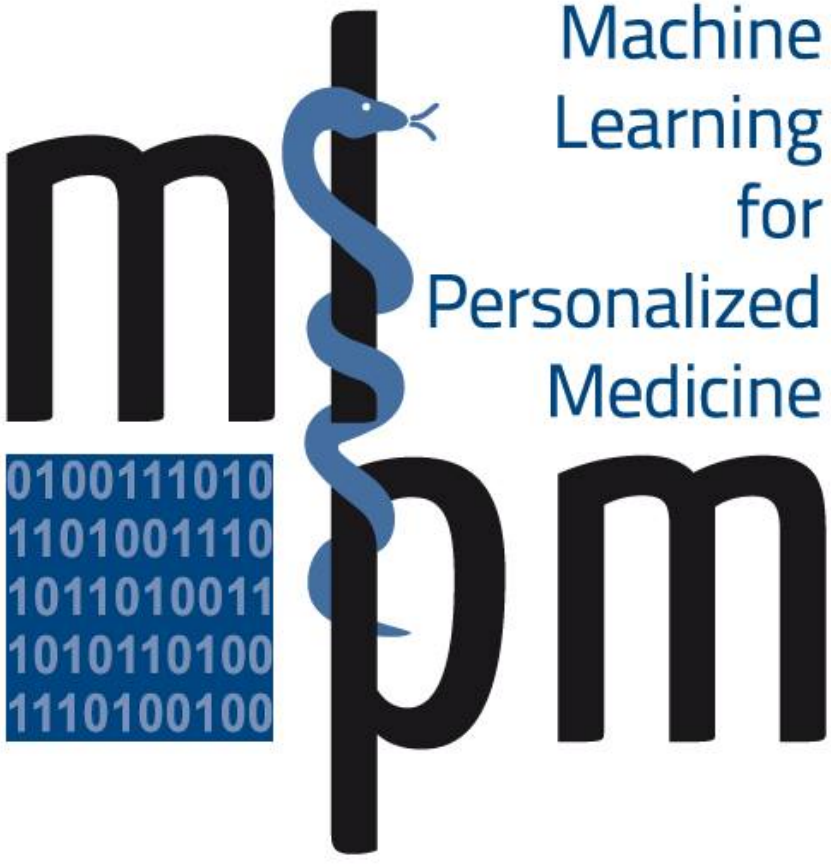
Example: Latent Doodle Space

(Baxter and Anjyo, 2006)

Generalization with much less Data than Dimensions

- Powerful uncertainly handling of GPs leads to surprising properties.
- Non-linear models can be used where there are fewer data points than dimensions *without overfitting*.

Spiral of Progression



Massive Missing Data

- If missing at random it can be marginalized.
- As data sets become very large (39 million in EMIS) data becomes extremely sparse.
- Imputation becomes impractical.

Imputation

- Expectation Maximization (EM) is gold standard imputation algorithm.
- Exact EM optimizes the log likelihood.
- Approximate EM optimizes a lower bound on log likelihood.
  - e.g. variational approximations (VIBES, Infer.net).
- Convergence is *guaranteed* to a local maxima in log likelihood.

Expectation Maximization

Require: An initial guess for missing data

Expectation Maximization

Require: An initial guess for missing data
repeat

Expectation Maximization

Require: An initial guess for missing data
  repeat
    Update model parameters (M-step)

Expectation Maximization

Require: An initial guess for missing data
  repeat
    Update model parameters (M-step)
    Update guess of missing data (E-step)

Expectation Maximization

Require: An initial guess for missing data
  repeat
    Update model parameters (M-step)
    Update guess of missing data (E-step)
  until convergence

Imputation is Impractical

- In very sparse data imputation is impractical.
- EMIS: 39 million patients, thousands of tests.
- For most people, most tests are missing.
- M-step becomes confused by poor imputation.

Direct Marginalization is the Answer

- Perhaps we need joint distribution of two test outcomes,

$$p(y_1, y_2)$$

- Obtained through marginalizing over all missing data,

$$p(y_1, y_2) = \int p(y_1, y_2, y_3, \ldots, y_p) \mathrm{d}y_3, \ldots \mathrm{d}y_p$$

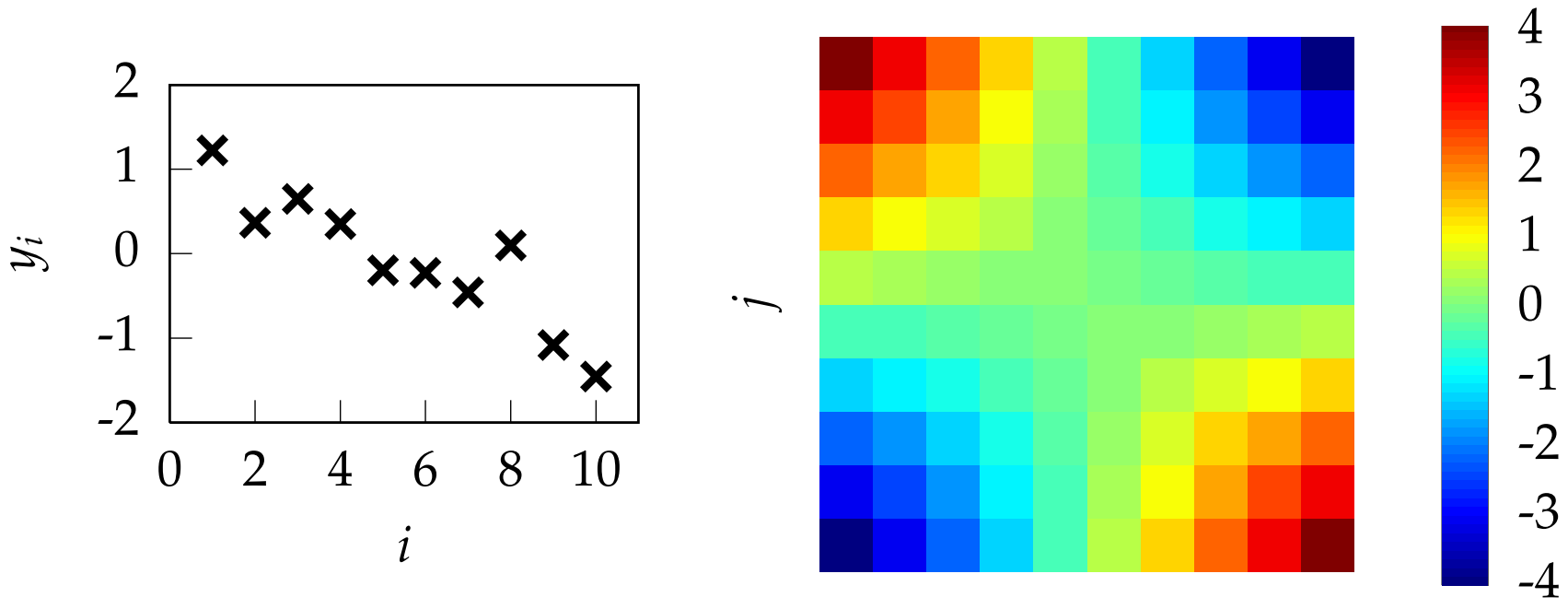
- Where $y_3, \ldots, y_p$ contains:
  1. all tests not applied to this patient
  2. all tests not yet invented!!

Magical Marginalization in Gaussians

Multi-variate Gaussians

- Given 10 dimensional multivariate Gaussian, $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$.
- Generate a single correlated sample $\mathbf{y} = [y_1, y_2 \dots y_{10}]$.
- How do we find the marginal distribution of y_1, y_2?

Gaussian Marginalization Property

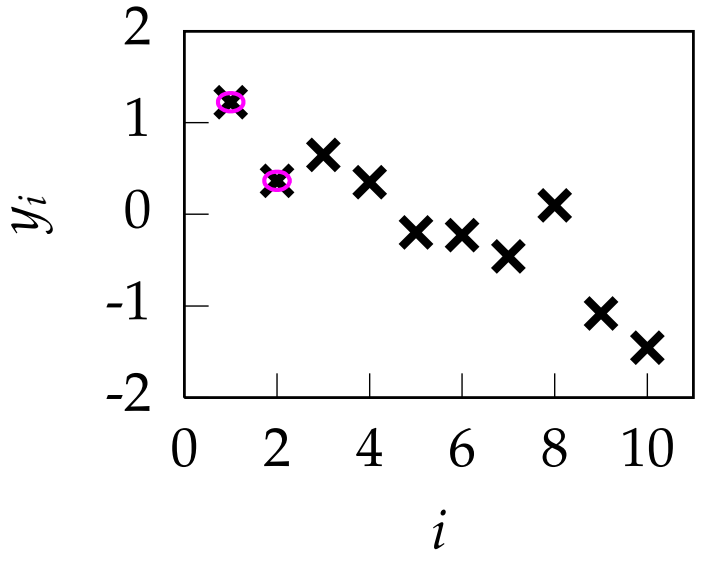


(a) A 10 dimensional sample

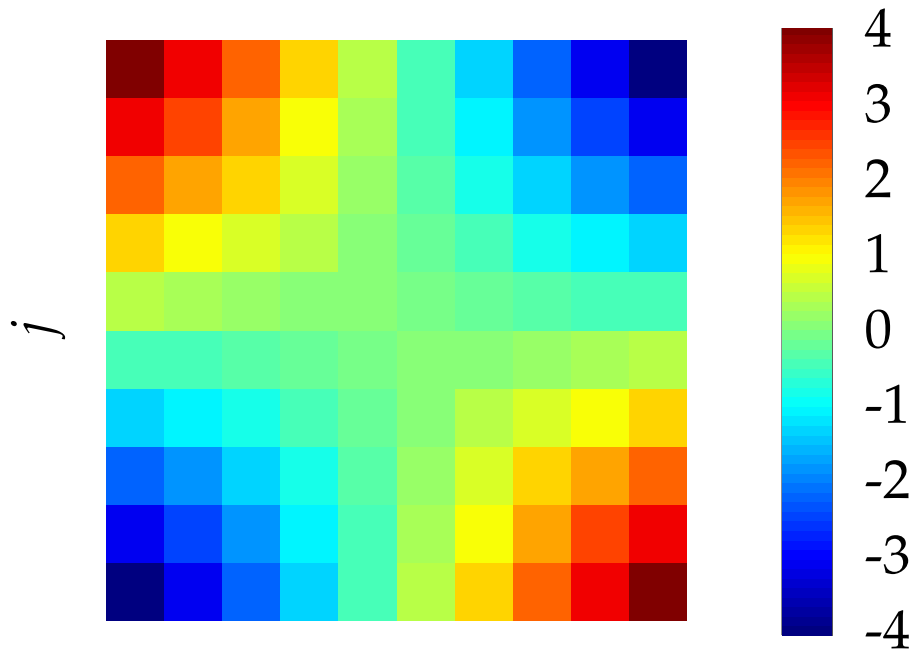
(b) colormap showing covariance between dimensions.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



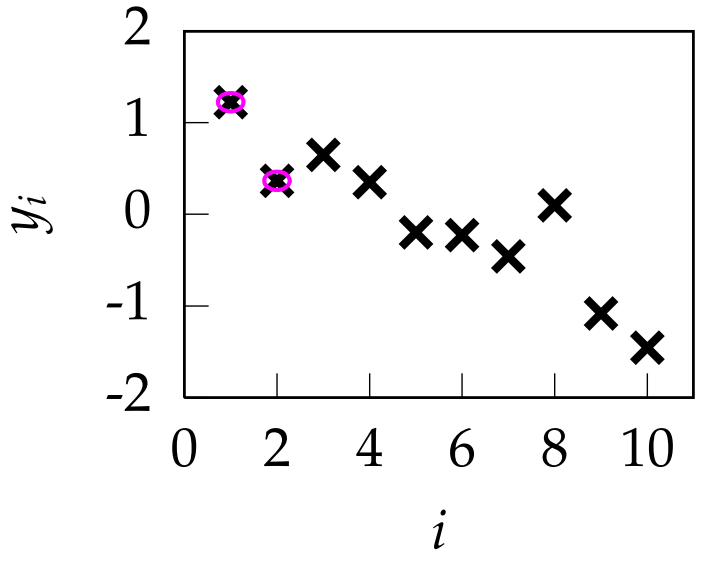
(a) A 10 dimensional sample



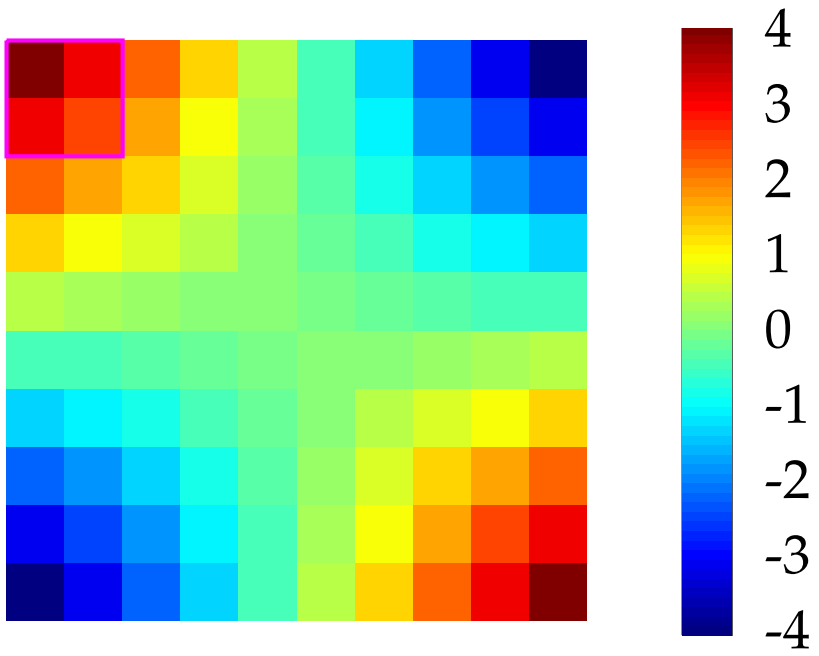
(b) colormap showing covariance between dimensions.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



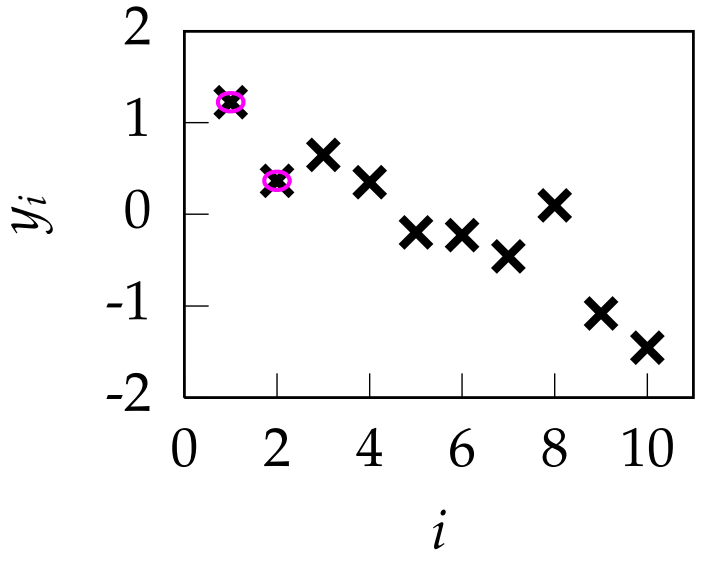
(a) A 10 dimensional sample



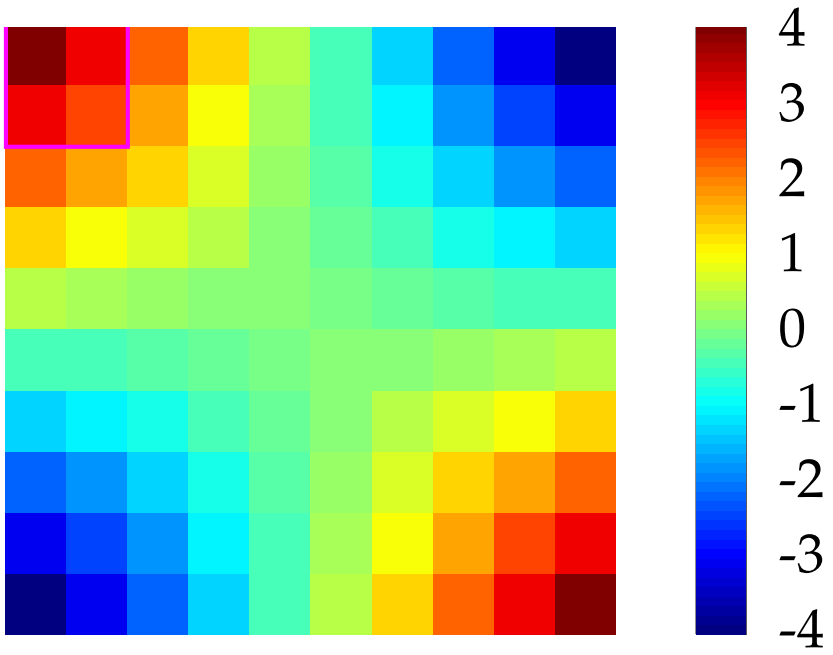
(b) colormap showing covariance between dimensions.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



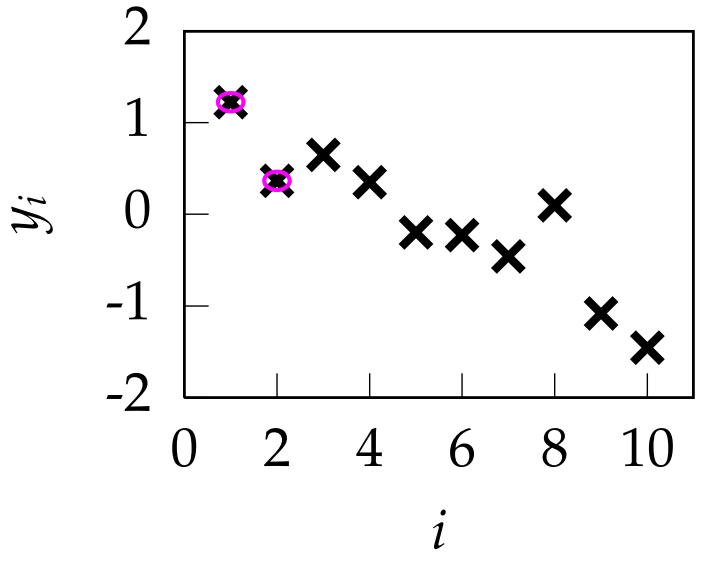
(a) A 10 dimensional sample



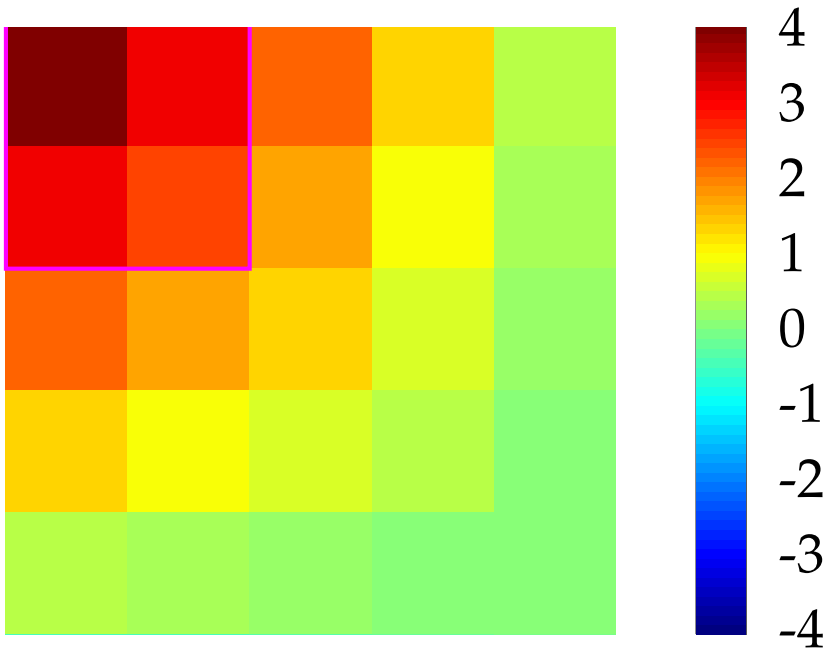
(b) colormap showing covariance between dimensions.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



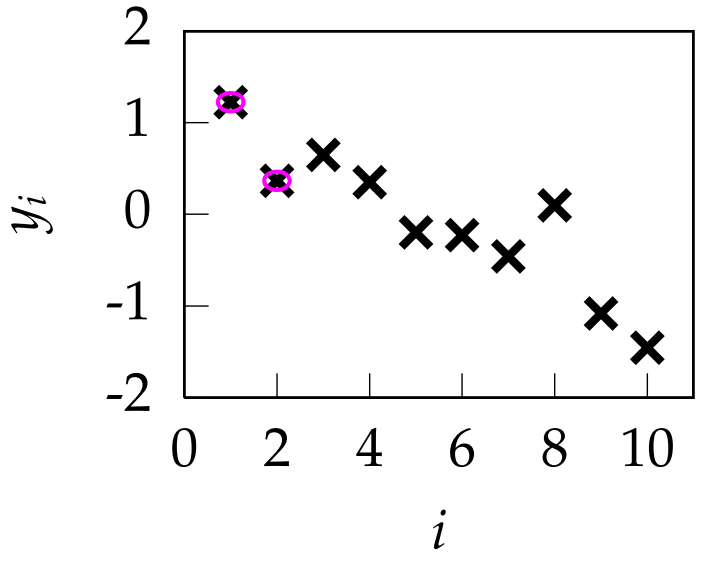
(a) A 10 dimensional sample



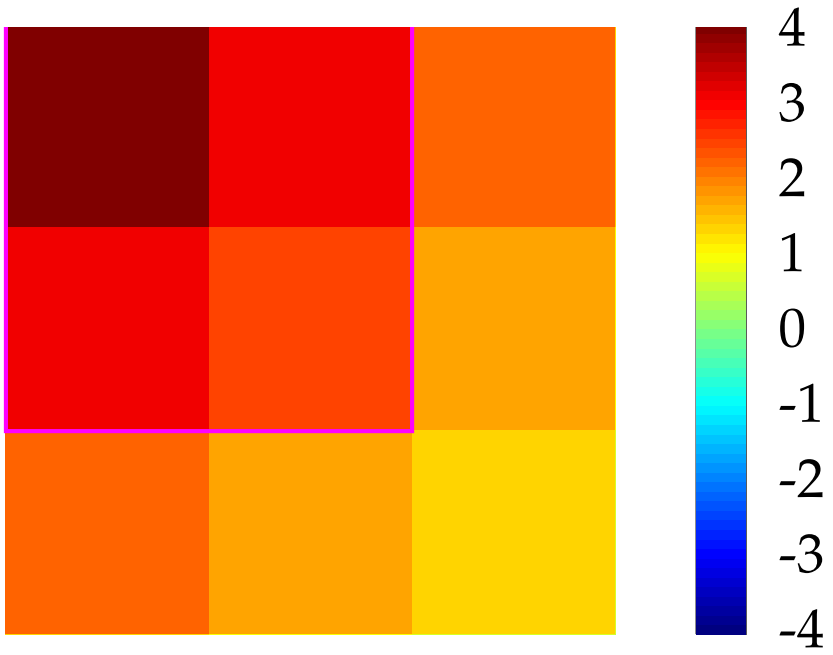
(b) colormap showing covariance between dimensions.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



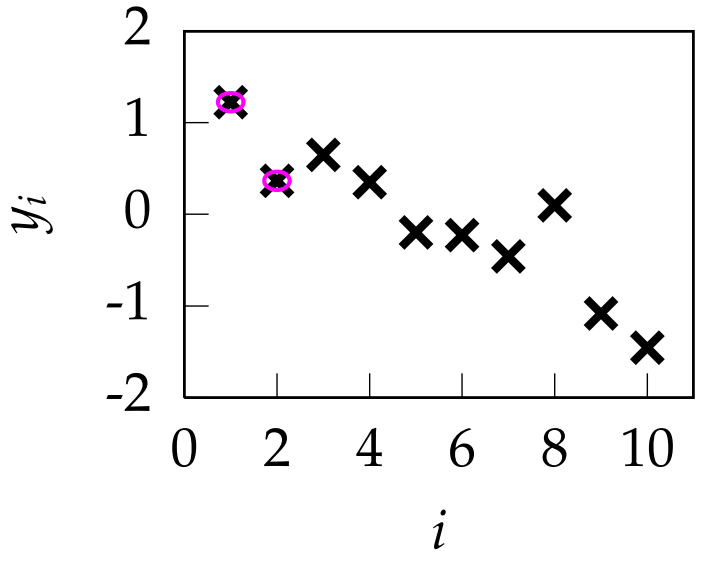
(a) A 10 dimensional sample



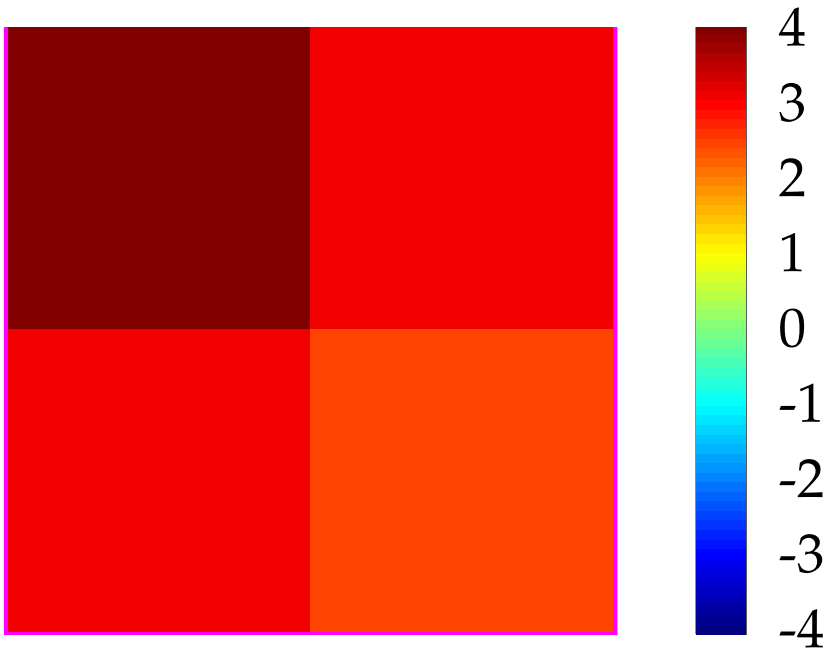
(b) colormap showing covariance between dimensions.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



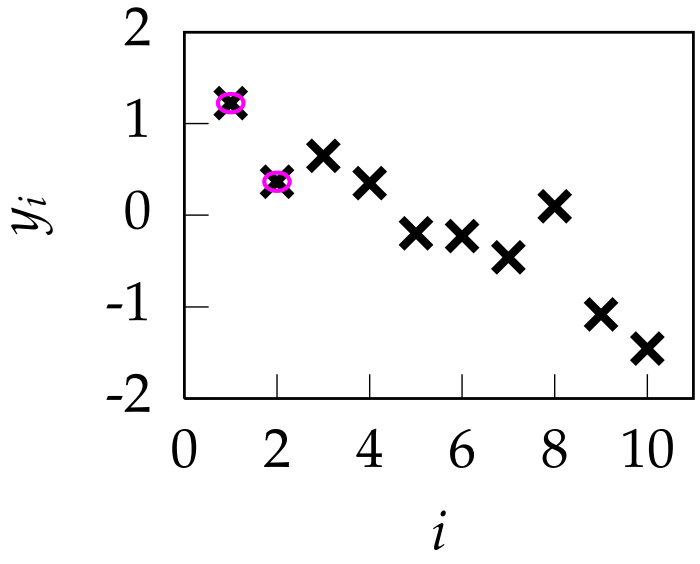
(a) A 10 dimensional sample



(b) colormap showing covariance between dimensions.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



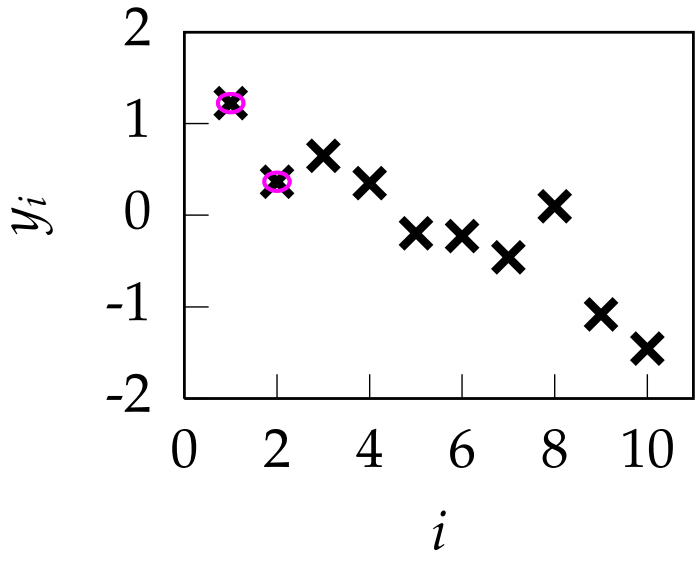
(a) A 10 dimensional sample

$$\begin{bmatrix} 4.1 & 3.1111 \\ 3.1111 & 2.5198 \end{bmatrix}$$

(b) covariance between y_1 and y_2.

Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Gaussian Marginalization Property



(a) A 10 dimensional sample

$$\begin{bmatrix} 1 & 0.96793 \\ 0.96793 & 1 \end{bmatrix}$$

(b) correlation between y_1 and y_2.

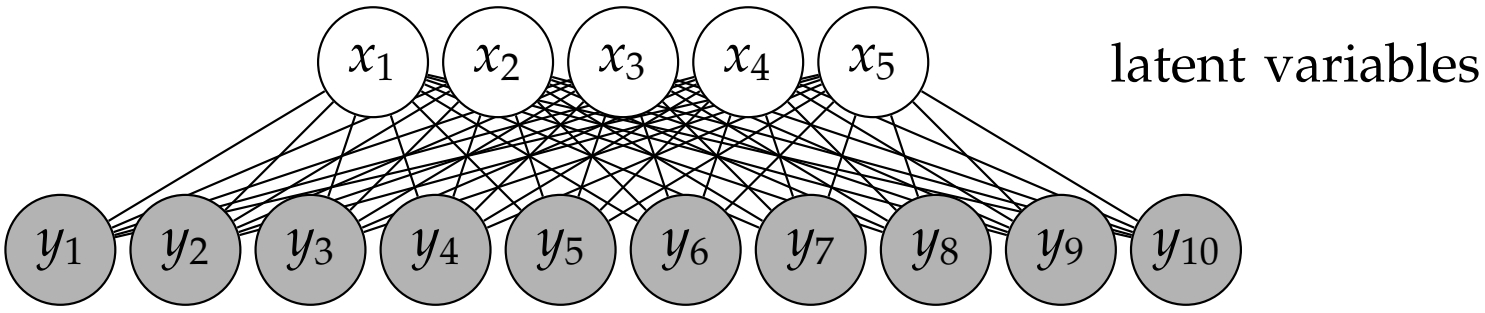
Figure : A sample from a 10 dimensional correlated Gaussian distribution.

Avoid Imputation: Marginalize Directly

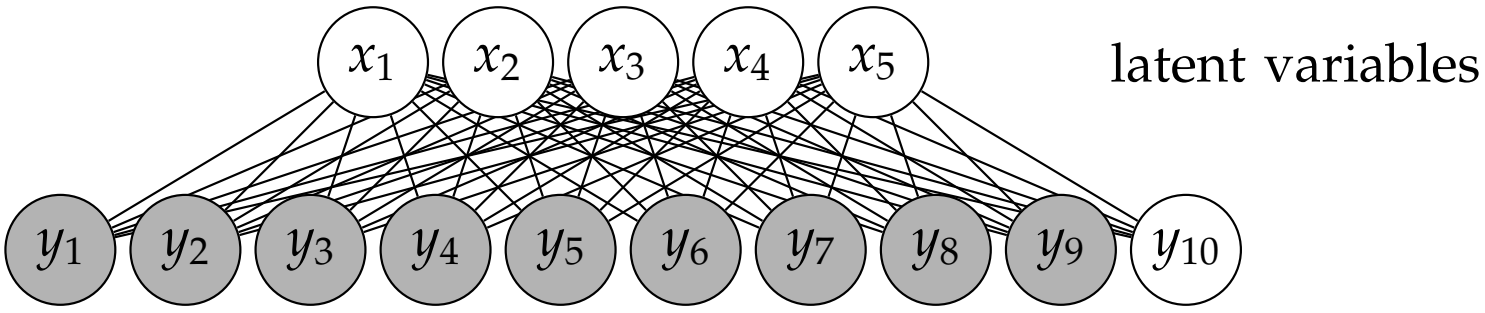
- Our approach: Avoid Imputation, Marginalize Directly.
- Explored in context of Collaborative Filtering.
- Similar challenges:
  - many users (patients),
  - many items (tests),
  - sparse data
- Implicitly marginalizes over all future tests too.

Work with Raquel Urtasun (Lawrence and Urtasun, 2009) and ongoing work with Max Zwießele and Nicoló Fusi.

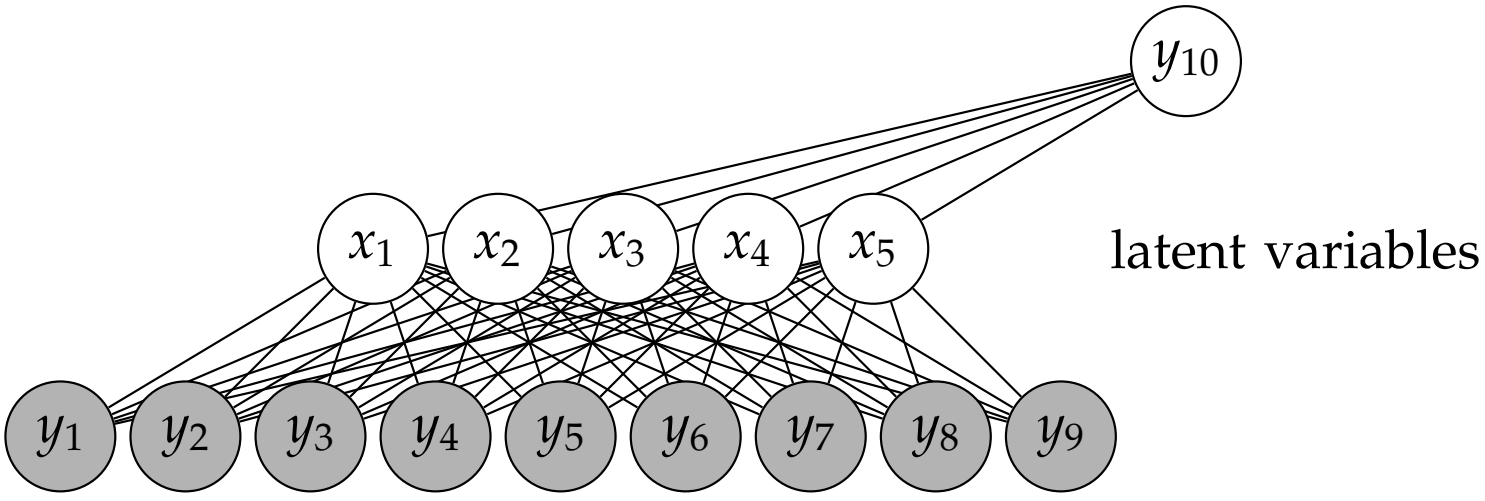
Marginalization in Bipartite Undirected Graph



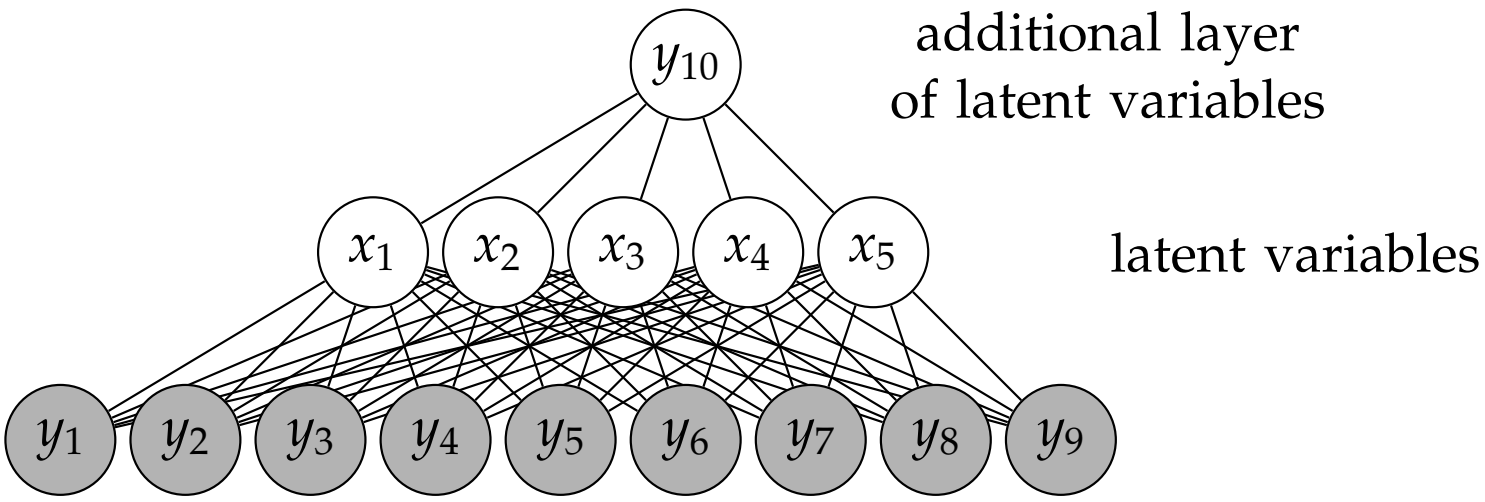
Marginalization in Bipartite Undirected Graph



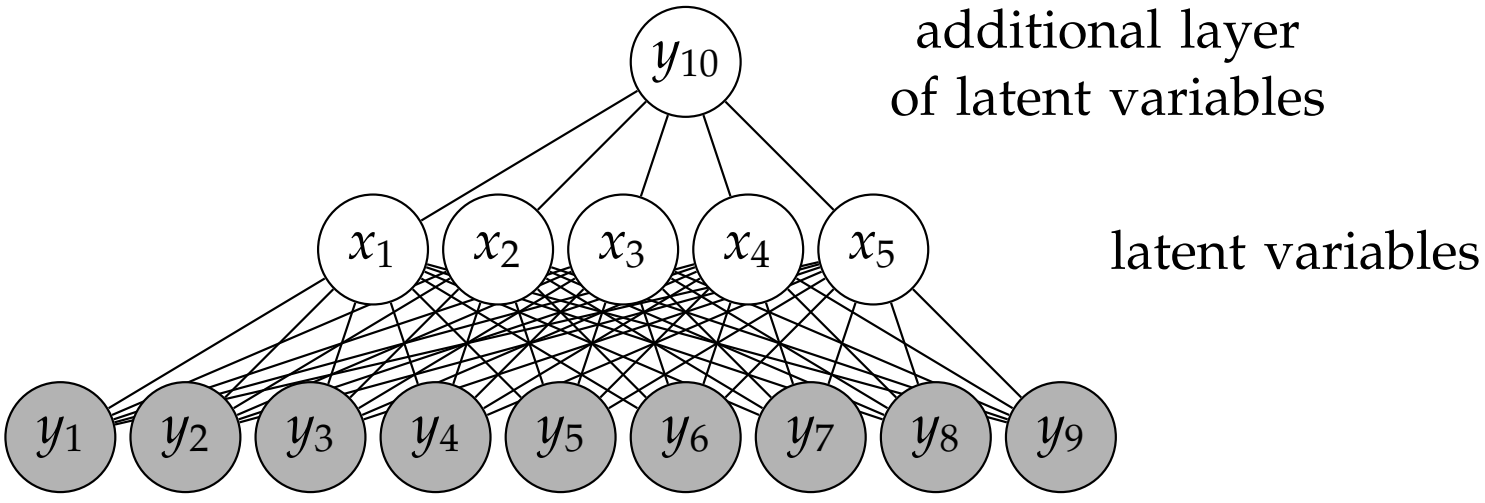
Marginalization in Bipartite Undirected Graph



Marginalization in Bipartite Undirected Graph



Marginalization in Bipartite Undirected Graph

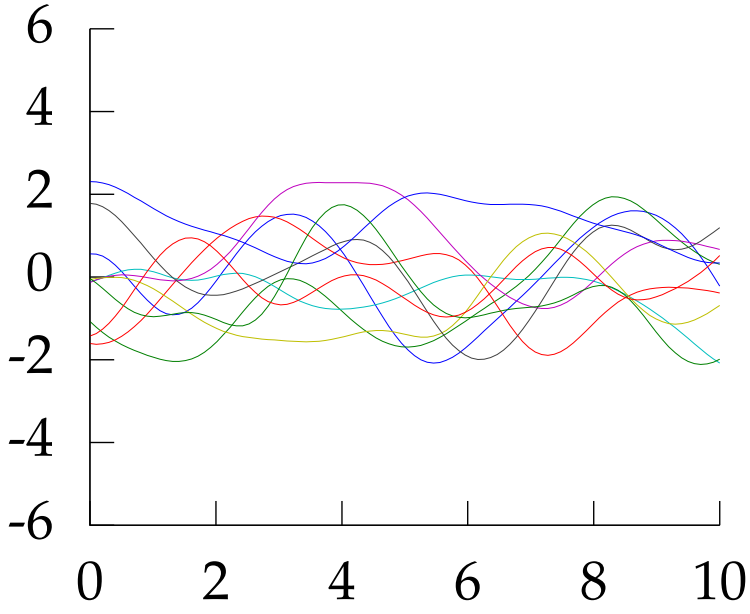


For *massive missing data,* how many additional latent variables?

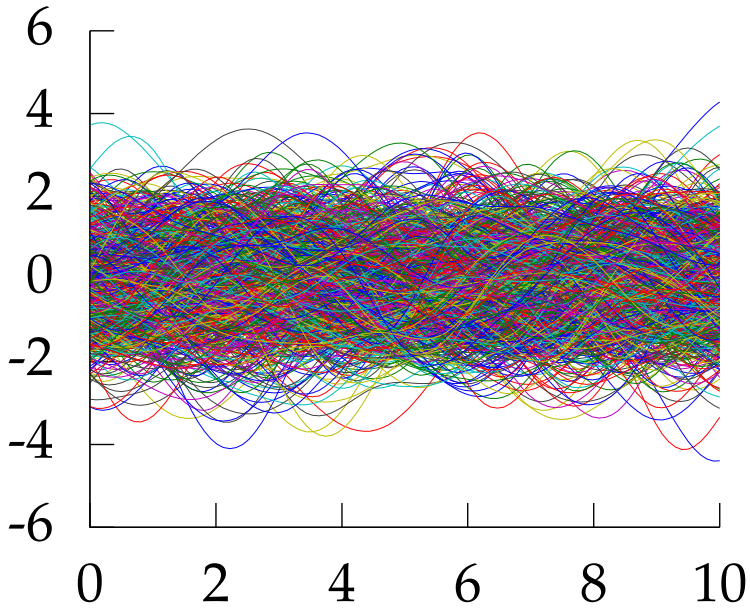
Methods that Interrelate Covariates

- Need Class of models that interrelates data, but allows for variable p.
- Common assumption: high dimensional data lies on low dimensional manifold.
- Want to retain the marginalization property of Gaussians but deal with non-Gaussian data!

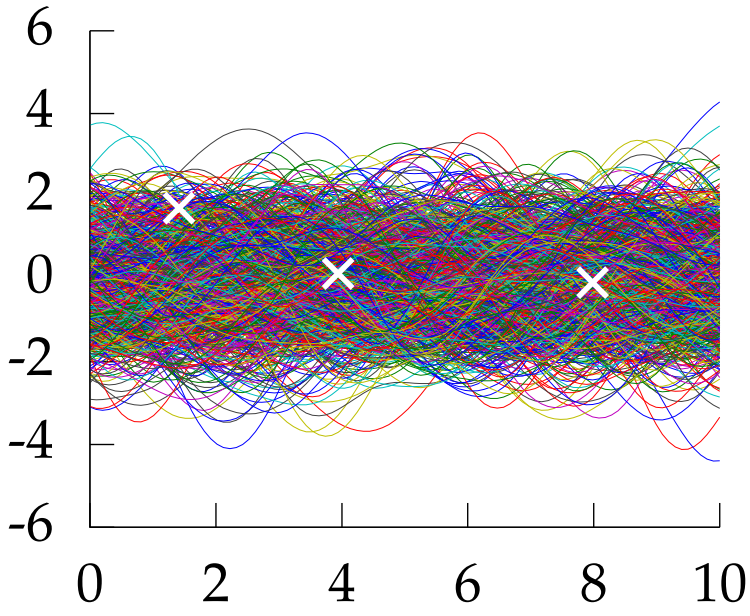
Gaussian Processes: Extremely Short Overview



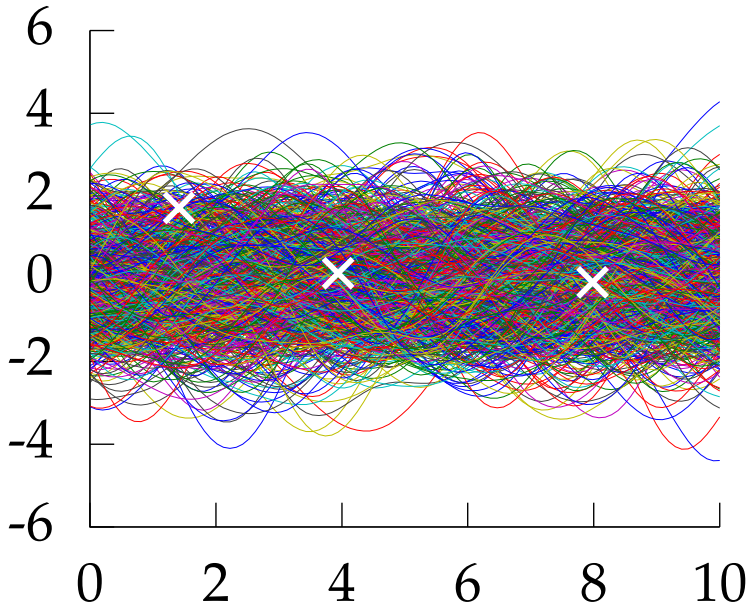
Gaussian Processes: Extremely Short Overview

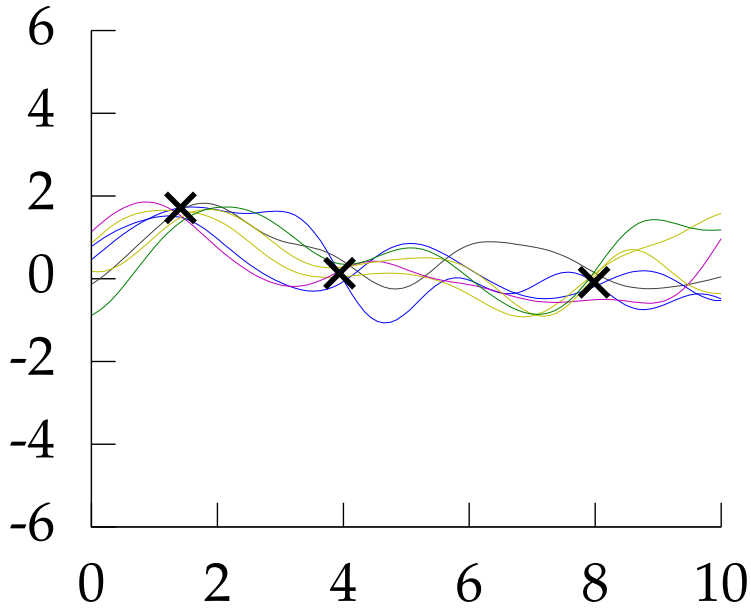


Gaussian Processes: Extremely Short Overview



Gaussian Processes: Extremely Short Overview





Gaussian Processes and Kernels

- The kernel, $\mathbf{K}$ acts as the *covariance* of the Gaussian.
- Predictions in Gaussian processes consist of the posterior *mean function* and *covariance*.
- For non degenerate covariance matrix, $\mathbf{K}$, the model is non parametric.

Gaussian Process Summer School

http://gpss.cc

14th-17th September 2015

Sheffield, UK

What's the Algorithm?

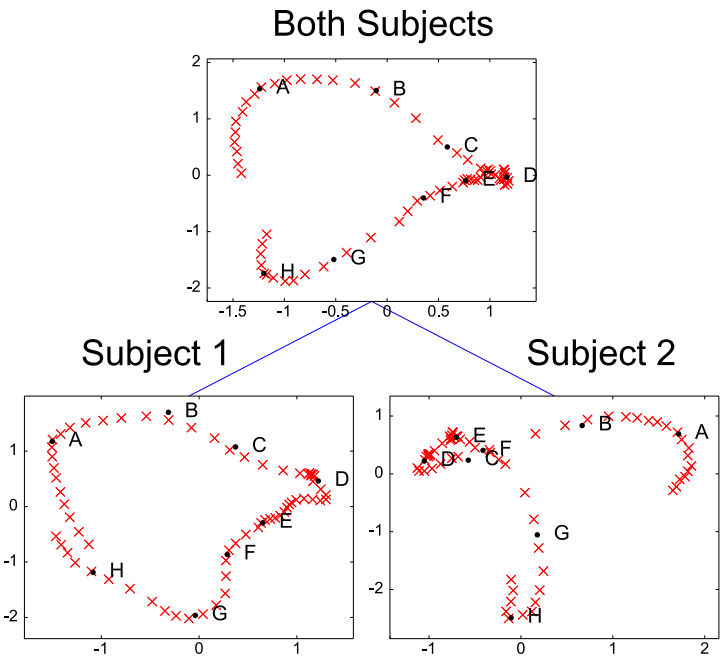
- For some algorithmic details:
  1. Attend a Gaussian process summer school (`http://gpss.cc`)
  2. Come to my talk at the Large Scale Kernel Machines Workshop (11:15 am, Pasteur Room)
- We make our software available on GitHub: `https://github.com/SheffieldML/GPy`

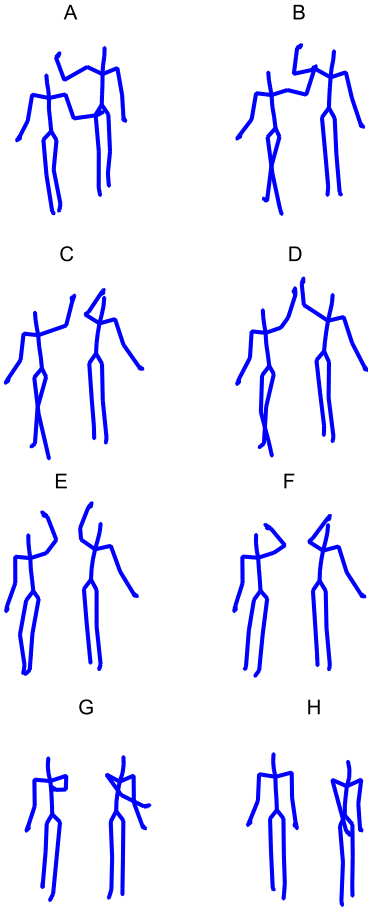
Original Motivation for Deep

- Assuming low dimensional embedding is one way of handling high dimensional data.
- Another is to assume conditional independencies (sparse graph structure).
- This inspires a layered hierarchy.

Hierarchical GP-LVM

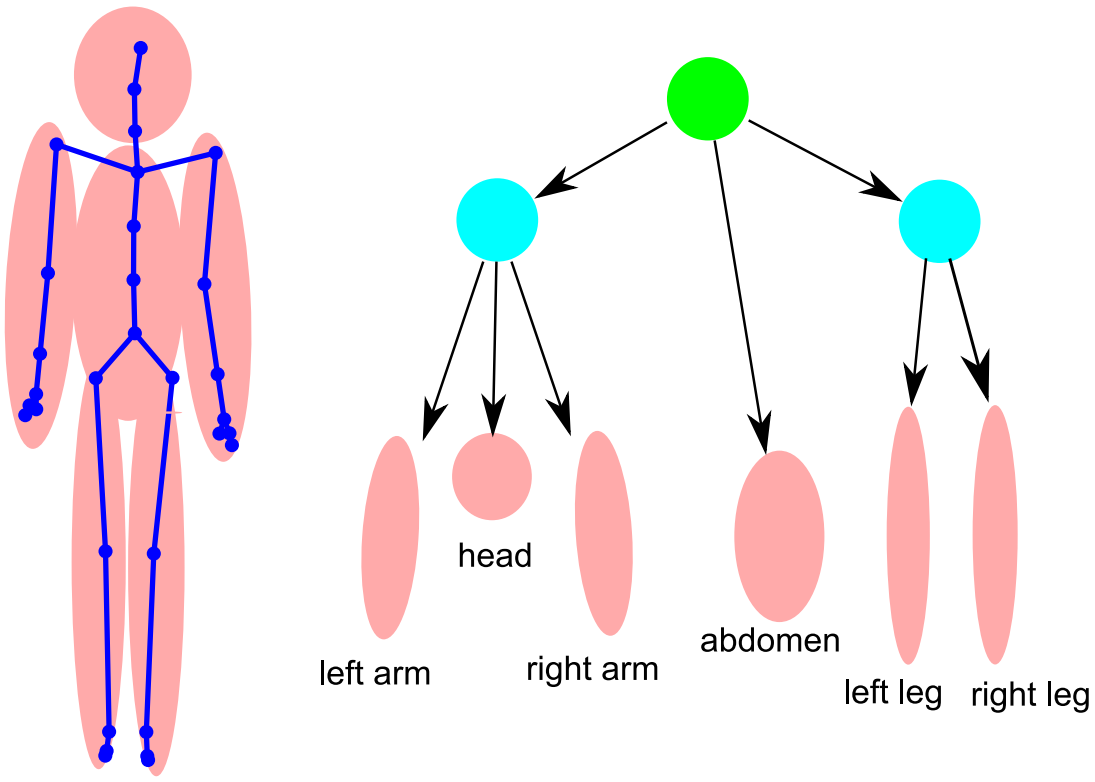
(Lawrence and Moore, 2007)





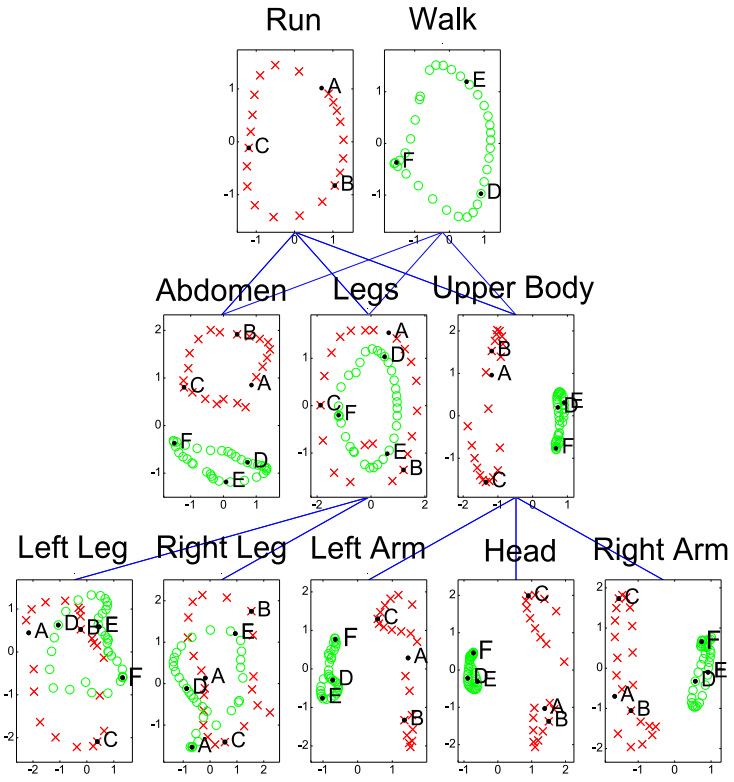
Hierarchical GP-LVM

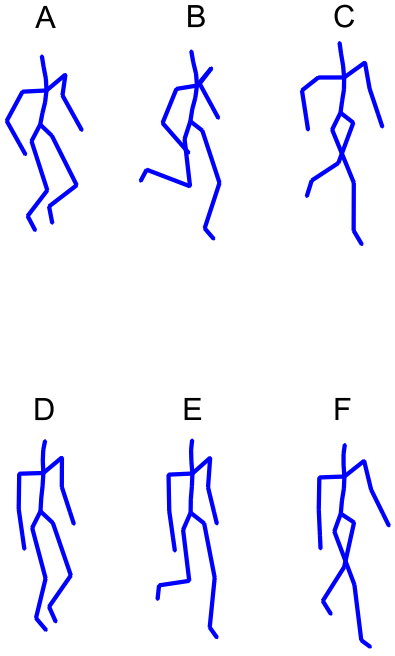
(Lawrence and Moore, 2007)



Hierarchical GP-LVM

(Lawrence and Moore, 2007)





Structure Learning

- Then we used MAP inference for learning.
- Prevents learning of structures.
- Our ‘modern’ approaches use variational approximations.
- Allows for learning of structure of model (number of latent variables in each layer).
- Allows for learning the right depth of model (number of layers).

Mathematically

- Composite *multivariate* function

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}_5(\mathbf{f}_4(\mathbf{f}_3(\mathbf{f}_2(\mathbf{f}_1(\mathbf{x})))))$$

Why Deep?

- Gaussian processes give priors over functions.
- Elegant properties:
  - e.g. *Derivatives* of process are also Gaussian distributed (if they exist).
- For particular covariance functions they are ‘universal approximators’, i.e. all functions can have support under the prior.
- Gaussian derivatives might ring alarm bells.
- E.g. a priori they don’t believe in function ‘jumps’.

Process Composition

- From a process perspective: *process composition*.
- A (new?) way of constructing more complex *processes* based on simpler components.

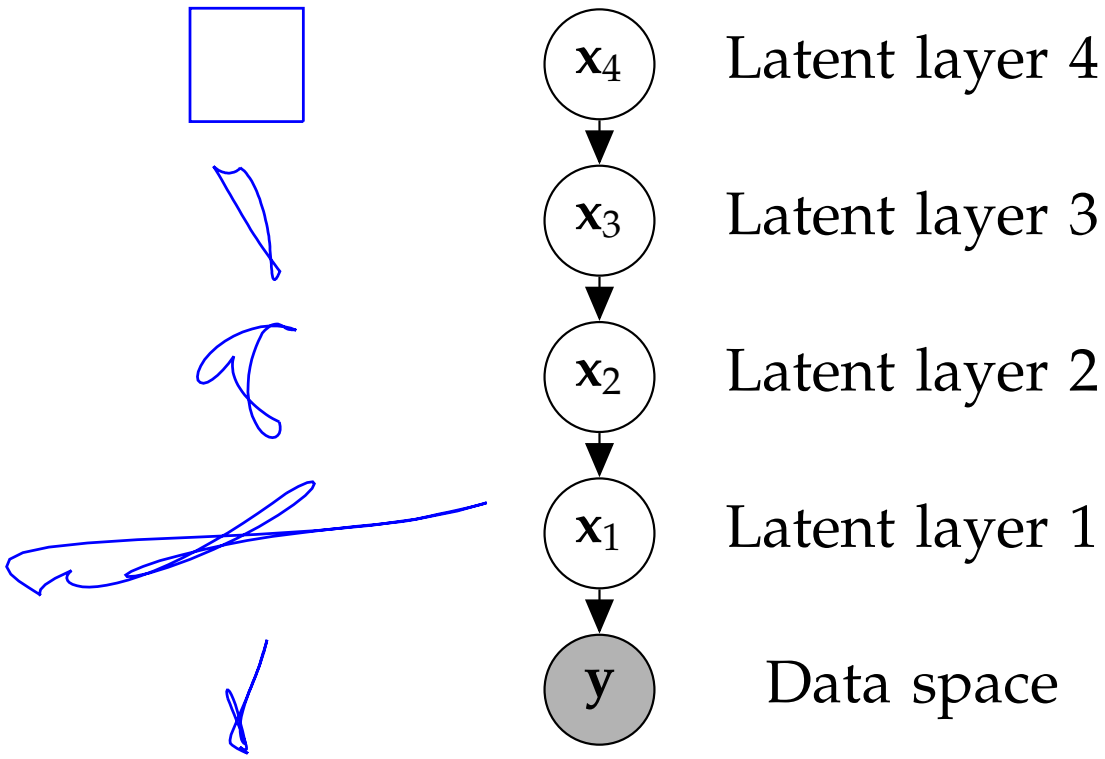
Note: To retain *Kolmogorov consistency* introduce IBP priors over latent variables in each layer (Zhenwen Dai).

Analysis of Deep GPs

- Duvenaud et al. (2014) Duvenaud et al show that the derivative distribution of the process becomes more *heavy tailed* as number of layers increase.
- Gal and Ghahramani (2015) Gal and Ghahramani show that Drop Out is a variational approximation to a deep Gaussian process.

Outline

Structures for Extracting Information from Data



Damianou and Lawrence (2013)

Deep Gaussian Processes

Andreas C. Damianou **Neil D. Lawrence**
Dept. of Computer Science & Sheffield Institute for Translational Neuroscience,
University of Sheffield, UK

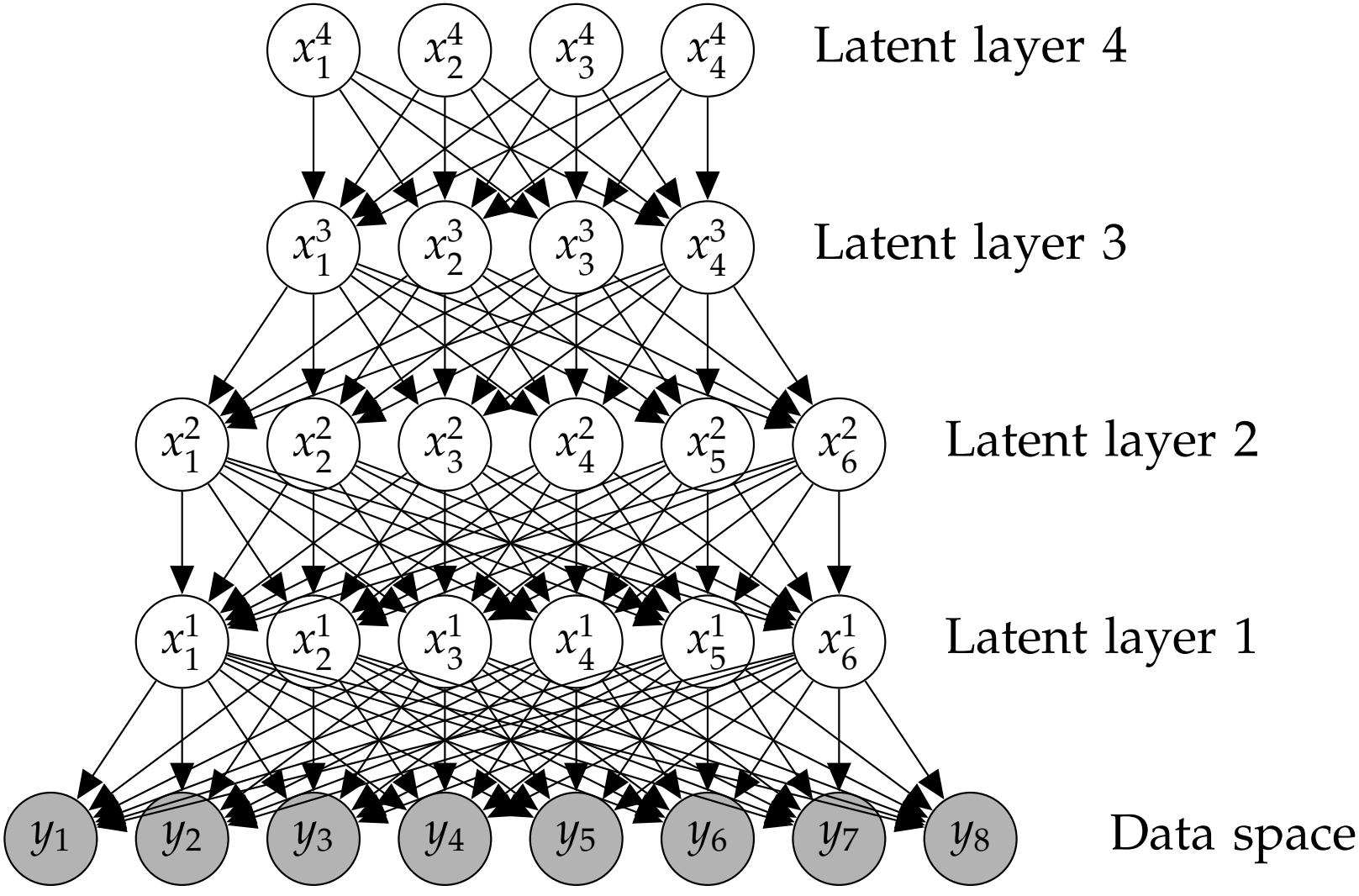
Abstract

In this paper we introduce deep Gaussian process (GP) models. Deep GPs are a deep belief network based on Gaussian process mappings. The data is modeled as the output of a multivariate GP. The inputs to that Gaussian process are then governed by another GP. A single layer model is equivalent to a standard GP or the GP latent variable model (GP-LVM). We perform inference in

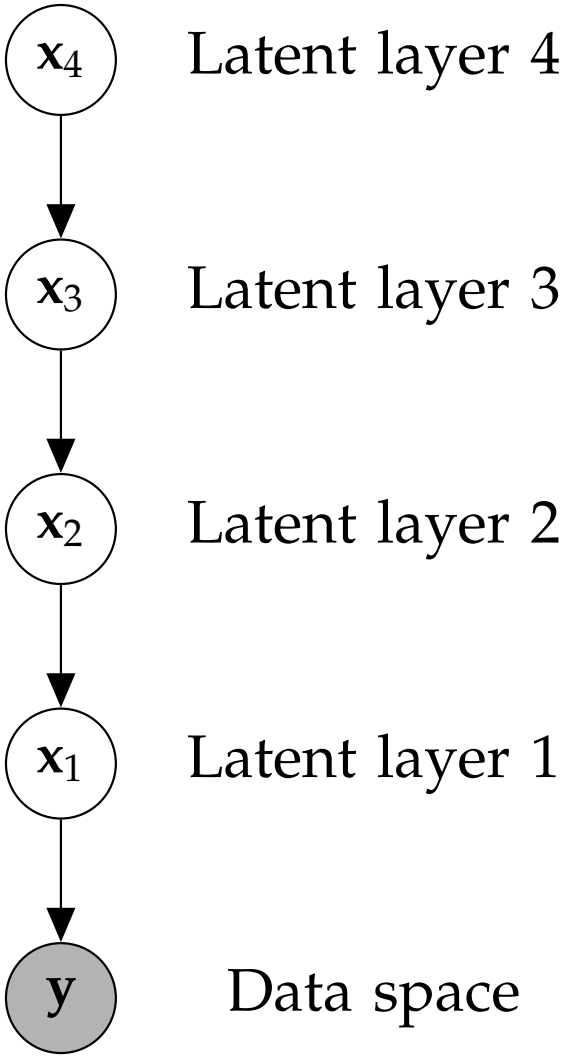
the question as to whether deep structures and the learning of abstract structure can be undertaken in *smaller* data sets. For smaller data sets, questions of generalization arise: to demonstrate such structures are justified it is useful to have an objective measure of the model's applicability.

The traditional approach to deep learning is based around binary latent variables and the restricted Boltzmann machine (RBM) [Hinton, 2010]. Deep hierarchies are constructed by stacking these models and various approximate inference techniques (such as contrastive divergence)

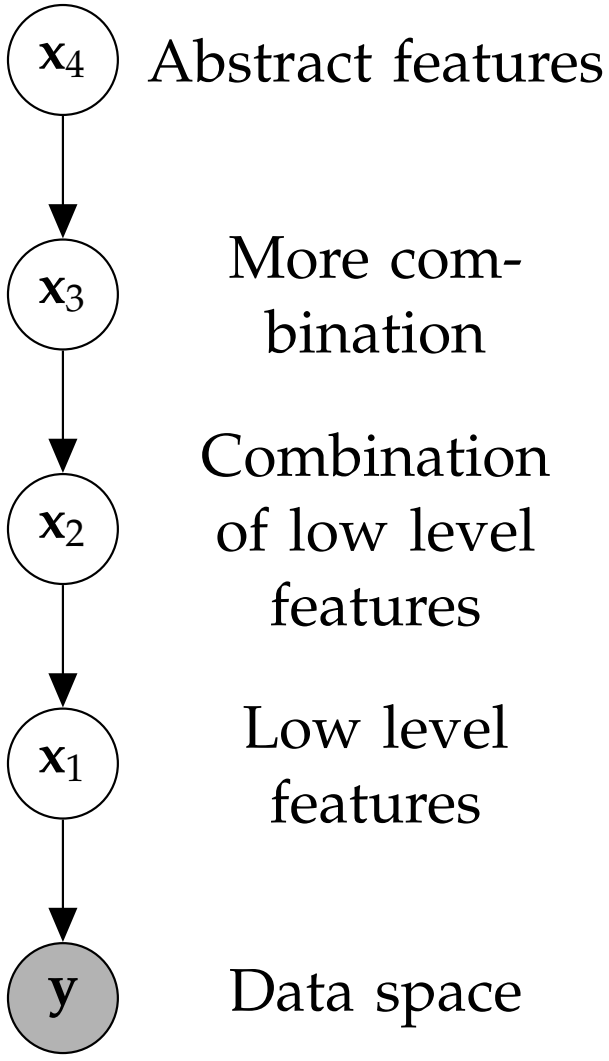
Deep Models


x_1^4 x_2^4 x_3^4 x_4^4
Latent layer 4
x_1^3 x_2^3 x_3^3 x_4^3
Latent layer 3
x_1^2 x_2^2 x_3^2 x_4^2 x_5^2 x_6^2
Latent layer 2
x_1^1 x_2^1 x_3^1 x_4^1 x_5^1 x_6^1
Latent layer 1
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8
Data space

Deep Models


$\mathbf{x}_4$
Latent layer 4
$\mathbf{x}_3$
Latent layer 3
$\mathbf{x}_2$
Latent layer 2
$\mathbf{x}_1$
Latent layer 1
$\mathbf{y}$
Data space

Deep Models


$\mathbf{x}_4$
Abstract features
$\mathbf{x}_3$
More combination
$\mathbf{x}_2$
Combination of low level features
$\mathbf{x}_1$
Low level features
$\mathbf{y}$
Data space

Deep Gaussian Processes

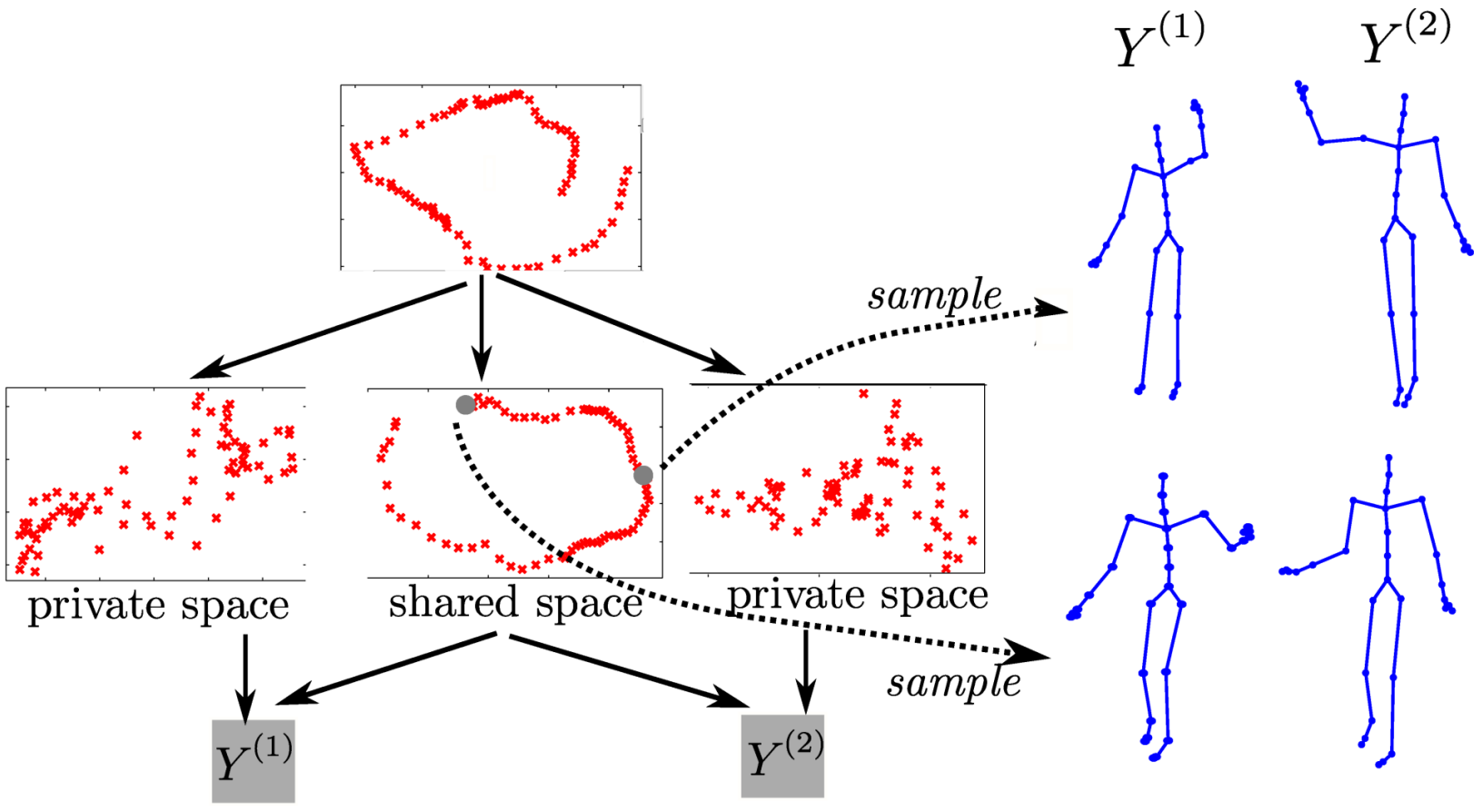
Damianou and Lawrence (2013)

- Deep architectures allow abstraction of features (Bengio, 2009; Hinton and Osindero, 2006; Salakhutdinov and Murray, 2008).
- We use variational approach to stack GP models.

Motion Capture

- 'High five' data.
- Model learns structure between two interacting subjects.

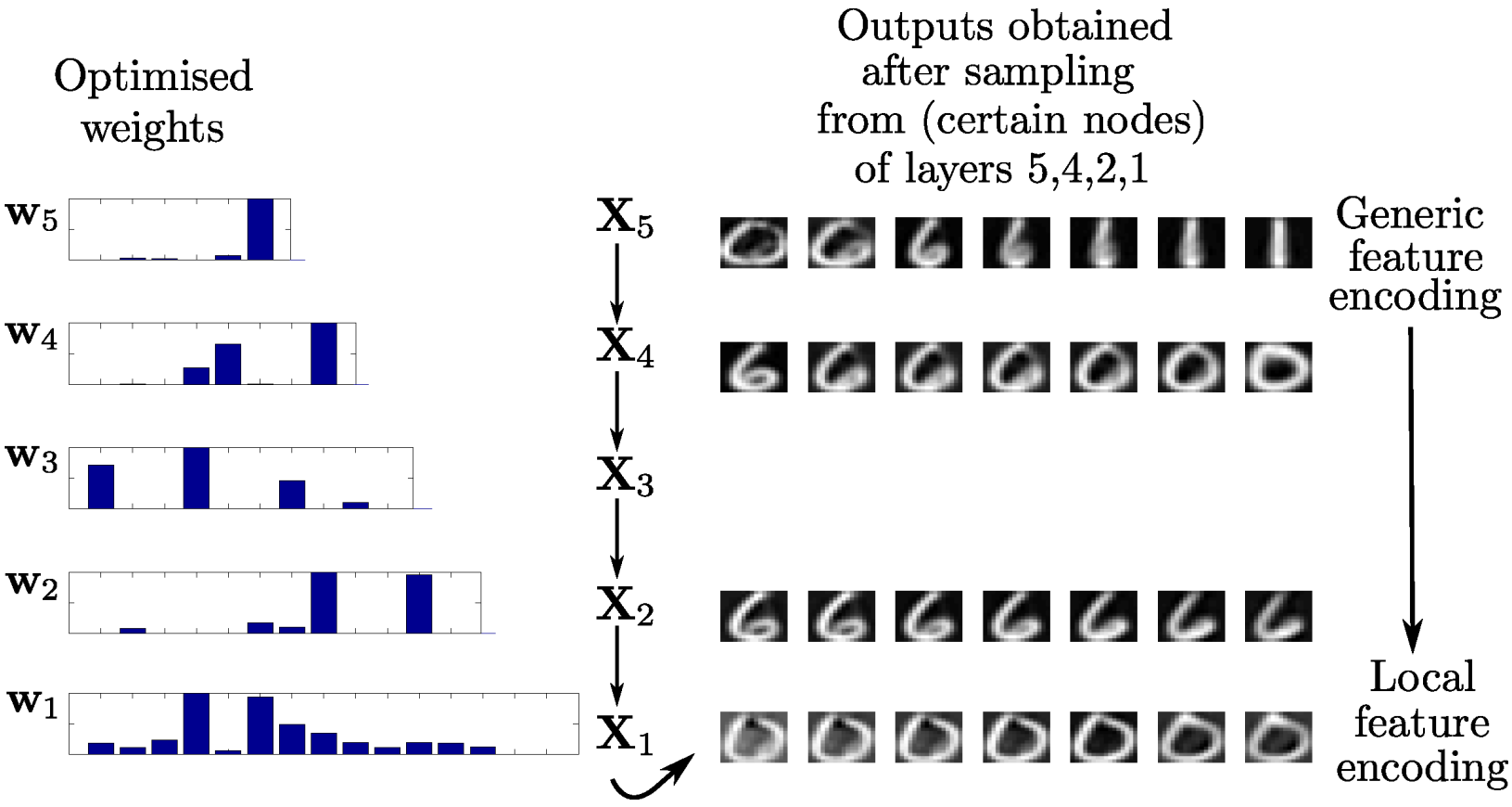
Deep hierarchies – motion capture



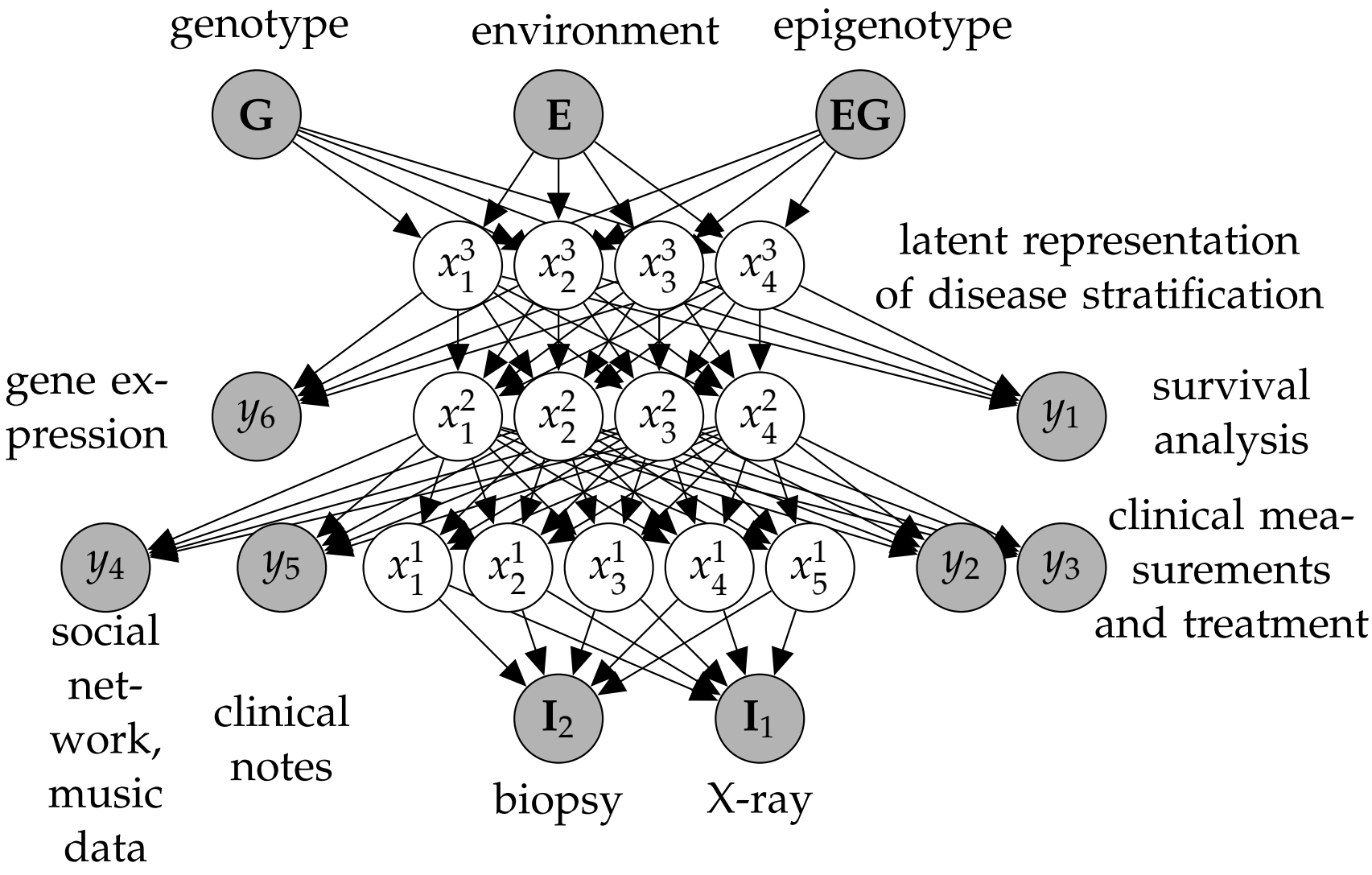
Digits Data Set

- Are deep hierarchies justified for small data sets?
- We can lower bound the evidence for different depths.
- For 150 6s, 0s and 1s from MNIST we found at least 5 layers are required.

Deep hierarchies – MNIST



Deep Health


genotype
environment
epigenotype
G
E
EG
x_1^3 x_2^3 x_3^3 x_4^3
latent representation of disease stratification
gene expression
y_6
x_1^2 x_2^2 x_3^2 x_4^2
y_1
survival analysis
y_4
y_5
x_1^1 x_2^1 x_3^1 x_4^1 x_5^1
y_2
y_3
clinical measurements and treatment
social network, music data
clinical notes
I_2
I_1
biopsy
X-ray

References I

W. V. Baxter and K.-I. Anjyo. Latent doodle space. In *EUROGRAPHICS*, volume 25, pages 477–485, Vienna, Austria, September 4-8 2006.

Y. Bengio. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.*, 2 (1):1–127, Jan. 2009. ISSN 1935-8237. [DOI].

A. Damianou and N. D. Lawrence. Deep Gaussian processes. In C. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics*, volume 31, AZ, USA, 2013. JMLR W&CP 31. [PDF].

D. Duvenaud, O. Rippel, R. Adams, and Z. Ghahramani. Avoiding pathologies in very deep networks. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Workshop on Artificial Intelligence and Statistics*, volume 33, Iceland, 2014. JMLR W&CP 33.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *arXiv:1506.02142*, 2015.

G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

References II

N. D. Lawrence and A. J. Moore. Hierarchical Gaussian process latent variable models. In Z. Ghahramani, editor, *Proceedings of the International Conference in Machine Learning*, volume 24, pages 481–488. Omnipress, 2007. [Google Books] . [PDF].

N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In L. Bottou and M. Littman, editors, *Proceedings of the International Conference in Machine Learning*, volume 26, San Francisco, CA, 2009. Morgan Kauffman. [PDF].

C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with GaussianFace. Technical report,

R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In S. Roweis and A. McCallum, editors, *Proceedings of the International Conference in Machine Learning*, volume 25, pages 872–879. Omnipress, 2008.