DD2434 - Advanced Machine Learning

Hierarchical Models

Carl Henrik Ek
{chek}@csc.kth.se

Royal Institute of Technology

December 1st, 2014

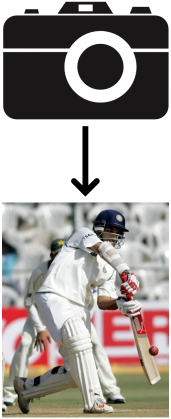
Last Lecture

- Representation Learning
  - Same story as before
  - Priors even more important
  - PPCA
  - GP-LVM
- Quickly: Multidimensional Scaling

Sensory Data

What we are doing

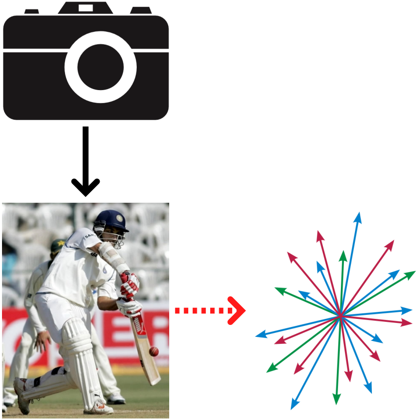
- Sensory representation
  - Capturing process
  - Pixels, Waveforms
- Degrees of freedom and dimensionality



Sensory Data

What we are doing

- Sensory representation
  - Capturing process
  - Pixels, Waveforms
- Degrees of freedom and dimensionality



Sensory Data

What we are doing

- Sensory representation
  - Capturing process
  - Pixels, Waveforms
- Degrees of freedom and dimensionality

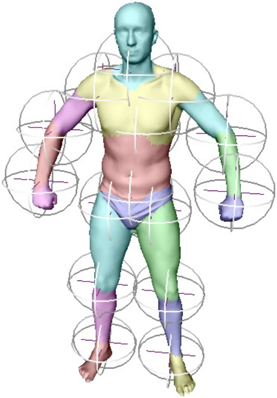


Image data

Image data

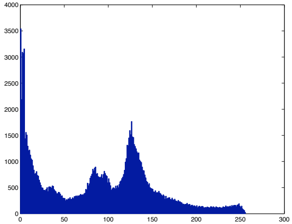

4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300

Image data

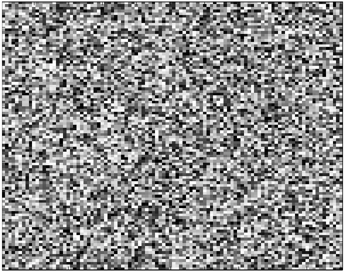


Image data

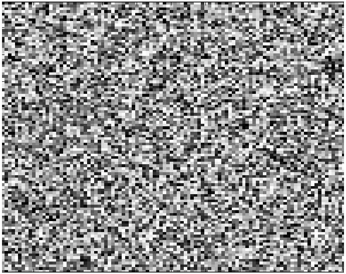


Image data

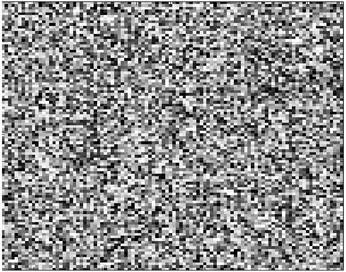
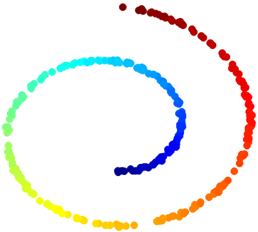


Image data

- Parametrisation
- Degrees of Freedom
- Generating parameters

Motivation

- Want to re-parametrise data
- Computational efficiency
- Discover “data-driven” degrees of freedom
  - Unravel data-manifold
- Interpretability
- Generalisation



Latent Variable Models[1]

$$p(\mathbf{X}) \tag{1}$$

- We have observed some data $\mathbf{X}$
- Lets assume that $\mathbf{X} \in \mathbb{R}^{N \times d}$ have been generated from $\mathbf{Z} \in \mathbb{R}^{N \times q}$
- $\mathbf{Z}$ - latent variable
- f - generative mapping

[1] Murphy 2012, p. 12.

Latent Variable Models[1]

$$p(\mathbf{X}|f, \mathbf{Z}) \tag{2}$$

$$\mathbf{f} : \mathbf{Z} \rightarrow \mathbf{X} \tag{3}$$

- We have observed some data $\mathbf{X}$
- Lets assume that $\mathbf{X} \in \mathbb{R}^{N \times d}$ have been generated from $\mathbf{Z} \in \mathbb{R}^{N \times q}$
- $\mathbf{Z}$ - latent variable
- f - generative mapping

[1] Murphy 2012, p. 12.

Latent Variable Models[1]

$$p(\mathbf{X}|f, \mathbf{Z}) \tag{4}$$

$$\mathbf{f} : \mathbf{Z} \rightarrow \mathbf{X} \tag{5}$$

- We have observed some data $\mathbf{X}$
- Lets assume that $\mathbf{X} \in \mathbb{R}^{N \times d}$ have been generated from $\mathbf{Z} \in \mathbb{R}^{N \times q}$
- $\mathbf{Z}$ - latent variable
- f - generative mapping

[1] Murphy 2012, p. 12.

Latent Variable Models[1]

$$p(\mathbf{X}|f, \mathbf{Z}) \tag{6}$$
$$\mathbf{f} : \mathbf{Z} \rightarrow \mathbf{X} \tag{7}$$

- We have observed some data $\mathbf{X}$
- Lets assume that $\mathbf{X} \in \mathbb{R}^{N \times d}$ have been generated from $\mathbf{Z} \in \mathbb{R}^{N \times q}$
- $\mathbf{Z}$ - latent variable
- f - generative mapping

[1] Murphy 2012, p. 12.

WTF?

The strength of Priors

- Encodes prior belief
- This can also be seen as a preference
  - Given several perfectly valid solutions which one do i prefer
  - Regularises solution space
- Latent variable models what do we prefer?

Factor Analysis[2]

$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i + \epsilon \tag{8}$$

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}) \tag{9}$$

- Assume the generating mapping to be linear
- For regression we assumed that we knew the inputs **Z**
- Now we do not

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i + \epsilon \tag{10}$$

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \Psi) \tag{11}$$

- Assume the generating mapping to be linear
- For regression we assumed that we knew the inputs **Z**
- Now we do not

[2]Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i + \epsilon \tag{12}$$

$$p(\mathbf{X}|\mathbf{Z},\theta) = \mathcal{N}(\mathbf{W}\mathbf{Z},\Psi)) \tag{13}$$

$$p(\mathbf{Z}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{0},\boldsymbol{\Sigma}_\mathbf{0}) \tag{14}$$

- Assume the generating mapping to be linear
- For regression we assumed that we knew the inputs **Z**
- Now we do not $\Rightarrow$ specify a prior

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$p(\mathbf{X}|\theta) = \int p(\mathbf{X}|\mathbf{Z},\theta)p(\mathbf{Z})\mathrm{d}\mathbf{Z} = \tag{15}$$

$$= \mathcal{N}(\mathbf{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\boldsymbol{\Sigma}_0\mathbf{W}^{\mathrm{T}}) \tag{16}$$

- **Z** and **W** are related
- Integrate out **Z**
  - pick $\mu_0 = 0$, $\Sigma_0 = \mathbf{I}$
- Low dimensional density model of **X**
  - $\mathcal{O}(QD)$ compared to $\mathcal{O}(D^2)$

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$p(\mathbf{X}|\theta) = \int p(\mathbf{X}|\mathbf{Z},\theta)p(\mathbf{Z})\mathrm{d}\mathbf{Z} = \tag{17}$$

$$= \mathcal{N}(\mathbf{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\boldsymbol{\Sigma}_0\mathbf{W}^{\mathrm{T}}) \tag{18}$$

$$= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\mathbf{W}^{\mathrm{T}}) \tag{19}$$

- **Z** and **W** are related
- Integrate out **Z**
  - pick $\boldsymbol{\mu}_0 = 0$, $\boldsymbol{\Sigma}_0 = \mathbf{I}$
- Low dimensional density model of **X**
  - $\mathcal{O}(QD)$ compared to $\mathcal{O}(D^2)$

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$p(\mathbf{X}|\theta) = \int p(\mathbf{X}|\mathbf{Z},\theta)p(\mathbf{Z})\mathrm{d}\mathbf{Z} = \quad (20)$$

$$= \mathcal{N}(\mathbf{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\boldsymbol{\Sigma}_0\mathbf{W}^{\mathrm{T}}) \quad (21)$$

$$= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\mathbf{W}^{\mathrm{T}}) \quad (22)$$

- **Z** and **W** are related
- Integrate out **Z**
  - pick $\boldsymbol{\mu}_0 = 0$, $\boldsymbol{\Sigma}_0 = \mathbf{I}$
- Low dimensional density model of **X**
  - $\mathcal{O}(QD)$ compared to $\mathcal{O}(D^2)$

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R} \quad (23)$$

$$p(\mathbf{X}|\theta) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}) \quad (24)$$

$$= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\mathbf{W}^{\mathrm{T}}) \quad (25)$$

$$(26)$$

Identifiability

- The marginal likelihood is invariant to a rotation
  - no unique solution
  - model is the same but interpretation tricky

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$\mathbf{W}_{ML} = \text{argmax}_{\mathbf{W}} p(\mathbf{X}|\theta) \tag{27}$$

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \tag{28}$$

Probabilistic PCA

- Dimensions of **X** independent given **Z**
  - **W** orthogonal matrix
- Closed form solution Murphy 2012, p. 12.2.2

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

$$\mathbf{W}_{ML} = \mathrm{argmax}_{\mathbf{W}} p(\mathbf{X}|\theta) \tag{29}$$

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{30}$$

$$\mathbf{W}_{ML} = \mathbf{U}_q(\Lambda - \sigma^2\mathbf{I})^{\frac{1}{2}} \tag{31}$$

$$\mathbf{S} = \mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}} \tag{32}$$

Probabilistic PCA

- Dimensions of **X** independent given **Z**
  - **W** orthogonal matrix
- Closed form solution Murphy 2012, p. 12.2.2

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

Summary

- Factor Analysis is a linear continous latent variable model
- Solution not unique
- PCA is Factor Analysis with two assumptions
  - factor loadings orthogonal $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$
  - noise free case $\epsilon = \lim_{\sigma^2 \to 0} \sigma^2 \mathbf{I}$
- PCA is incredibly useful but its important to know what you are assuming, the probabilistic formulation allows you to do just that

[2] Murphy 2012, p. 12.1.1.

Factor Analysis[2]

Summary

- Factor Analysis is a linear continous latent variable model
- Solution not unique
- PCA is Factor Analysis with two assumptions
  - factor loadings orthogonal $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$
  - noise free case $\epsilon = \lim_{\sigma^2 \to 0} \sigma^2 \mathbf{I}$
- PCA is incredibly useful but its important to know what you are assuming, the probabilistic formulation allows you to do just that

[2] Murphy 2012, p. 12.1.1.

Gaussian Process Latent Variable Models

History repeats itself

- In PPCA we assumed no uncertainty in the mapping
- We can use $\mathcal{GP}$s over mapping
- Gaussian Process Latent Variable Model [Lawrence 2005]

Gaussian Process Latent Variable Models

History repeats itself

- In PPCA we assumed no uncertainty in the mapping
- We can use $\mathcal{GP}$s over mapping
- Gaussian Process Latent Variable Model [Lawrence 2005]

Gaussian Process Latent Variable Models

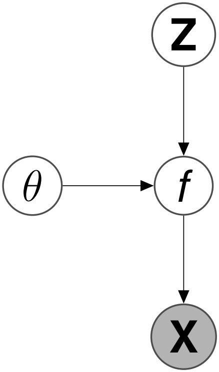
$$p(\mathbf{X}|\mathbf{f}, \mathbf{Z}, \theta) \tag{33}$$

- In PPCA we marginalised out **Z** and optimised for **W**
- Not possible for a general $\mathcal{GP}$

Gaussian Process Latent Variable Models

GP-LVM

- General co-variance function (Ex. SE)
- **Z** appears non-linearly in relation to **X**
- Marginalisation of **Z** intractable



Gaussian Process Latent Variable Models

$$\text{argmax}_{\mathbf{Z},\theta} p(\mathbf{X}|\mathbf{Z},\theta)p(\mathbf{Z}) \tag{34}$$

$$p(\mathbf{X}|\mathbf{Z},\theta) = \int p(\mathbf{X}|\mathbf{f})p(\mathbf{f}|\mathbf{Z},\theta)\mathrm{d}\mathbf{f} \tag{35}$$

$$p(\mathbf{Z}) = \mathcal{N}(\mathbf{0},\mathbf{I}) \tag{36}$$

- **GP**-prior sufficiently regularises objective
- Need to set dimensionality of **Z**

Gaussian Process Latent Variable Models

- You can add different priors on latent representations
  - Topological
  - Dynamic GP and a GP
  - Classification
- Any preference you can formulate as a prior

Gaussian Process Latent Variable Models

$$\mathbf{z}_{t+1} = g(\mathbf{z}_t) + \epsilon_z \tag{37}$$
$$g \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{z}_i, \mathbf{z}_j)) \tag{38}$$

- You can add different priors on latent representations
  - Topological
  - Dynamic GP and a GP
  - Classification
- Any preference you can formulate as a prior

Gaussian Process Latent Variable Models

- You can add different priors on latent representations
  - Topological
  - Dynamic GP and a GP
  - Classification
- Any preference you can formulate as a prior

Gaussian Process Latent Variable Models

- You can add different priors on latent representations
  - Topological
  - Dynamic GP and a GP
  - Classification
- Any preference you can formulate as a prior

Multidimensional Scaling

- N entities with proximity relations δ_{ij}
- Must be metric
- Find embedding $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T$ to minimize

$$E_{MDS} = ||\mathbf{D} - \Delta||_F \quad \begin{cases} \mathbf{D}_{ij} = ||\mathbf{y}_i - \mathbf{y}_j||_{L2} \\ \Delta_{ij} = \delta_{ij} \end{cases}$$

$$||\mathbf{A}||_F = \sqrt{\mathrm{trace}\left(\mathbf{A}\mathbf{A}^T\right)} = \sqrt{\sum_{i=1}^{N} \lambda_i^2}$$

$$||\mathbf{D} - \Delta||_F = \left\{ \Delta = \mathbf{V}\Lambda\mathbf{V}^T \Rightarrow \Delta = \sum_{i=1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^T \right\} =$$

$$= \quad ||\mathbf{D} - \sum_{i=1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^T||_F = ||\sum_{i=1}^{d} q_i \mathbf{v}_i \mathbf{v}_i^T - \sum_{i=1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^T||_F =$$

$$= \quad ||\sum_{i=1}^{d} (q_i - \lambda_i) \mathbf{v}_i \mathbf{v}_i^T - \sum_{i=d+1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^T||_F$$

Choose $\mathbf{D} = \mathbf{A}_{\rightarrow d} \Rightarrow E_{MDS} = \sqrt{\sum_{i=d+1}^{N} \lambda_i^2}$

Multidimensional Scaling

Generate geometrical configuration **Y** that could generate **D**

1. Convert distance matrix D to Gram matrix $\mathbf{G} = \mathbf{Y}\mathbf{Y}^T$

   ▸ Proof

2. Diagonalise Gram matrix G

$$
\begin{aligned}
\mathbf{G} &= \mathbf{Y}\mathbf{Y}^T = \mathbf{V}\Lambda\mathbf{V}^T = \left(\mathbf{V}\Lambda^{\frac{1}{2}}\right)\left(\Lambda^{\frac{1}{2}}\mathbf{V}^T\right) = \\
&= \left(\mathbf{V}\Lambda^{\frac{1}{2}}\right)\left(\mathbf{V}\left(\Lambda^{\frac{1}{2}}\right)^T\right)^T = \left(\mathbf{V}\Lambda^{\frac{1}{2}}\right)\left(\mathbf{V}\Lambda^{\frac{1}{2}}\right)^T
\end{aligned}
$$

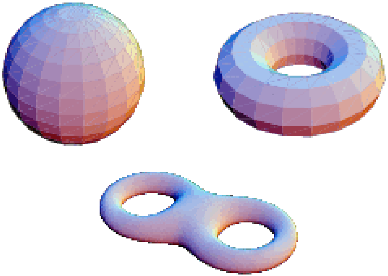
3. Chose $\mathbf{Y} = \mathbf{V}\Lambda^{\frac{1}{2}}$
4. Dimension of **Y**: $\mathrm{rank}(\mathbf{Y}\mathbf{Y}^T) = \mathrm{rank}(\mathbf{G}) = \mathrm{rank}(\mathbf{D}) = d$

▸ PCA Equivalence

Non linearities

Manifold

- Generalisation of low dimensional object embedded in high dimensional space
- Similarity?
- Local similarity
- Extend local similarity to global



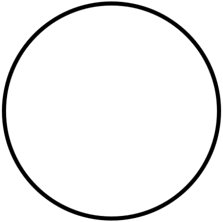
Non linearities

Definition

"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"[a]

[a] http://en.wikipedia.org/wiki/Manifold

Non linearities

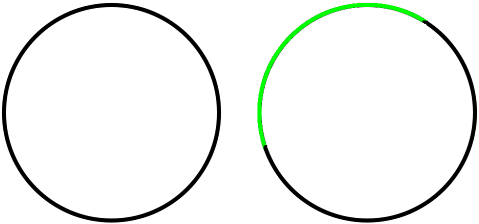


Definition

"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"[a]

[a] http://en.wikipedia.org/wiki/Manifold

Non linearities

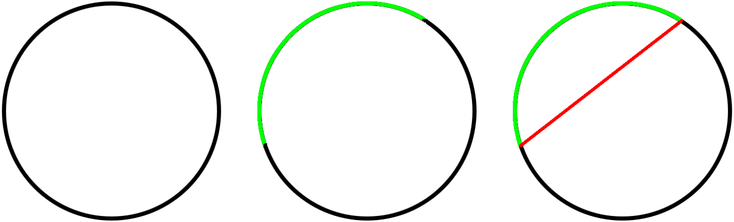


Definition

"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"[a]

[a] http://en.wikipedia.org/wiki/Manifold

Non linearities

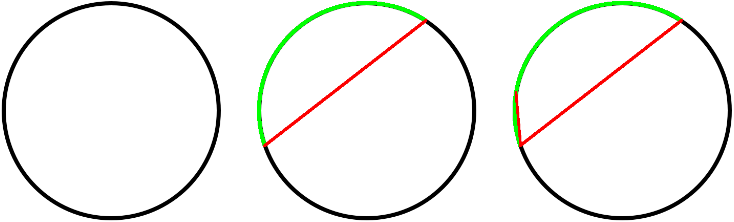


Definition

"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"[a]

[a] http://en.wikipedia.org/wiki/Manifold

Non linearities

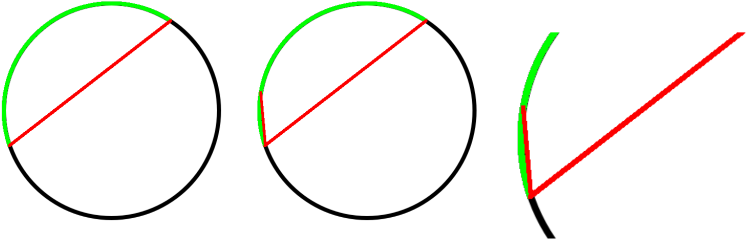


Definition

"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"[a]

[a] http://en.wikipedia.org/wiki/Manifold

Non linearities



Definition

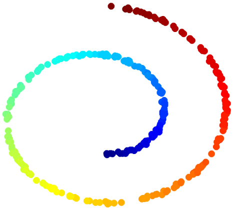
"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"[a]

[a] http://en.wikipedia.org/wiki/Manifold

Non linearities

Manifold

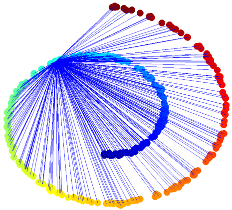
- Generalisation of low dimensional object embedded in high dimensional space
- Similarity?
- Local similarity
- Extend local similarity to global



Non linearities

Manifold

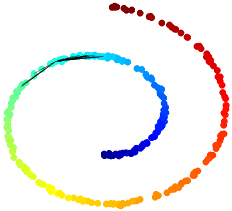
- Generalisation of low dimensional object embedded in high dimensional space
- Similarity?
- Local similarity
- Extend local similarity to global



Non linearities

Manifold

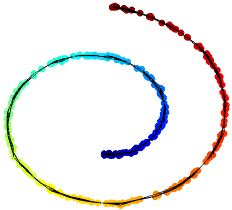
- Generalisation of low dimensional object embedded in high dimensional space
- Similarity?
- Local similarity
- Extend local similarity to global



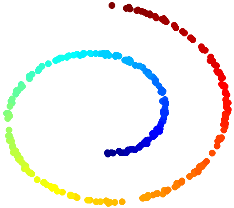
Non linearities

Manifold

- Generalisation of low dimensional object embedded in high dimensional space
- Similarity?
- Local similarity
- Extend local similarity to global



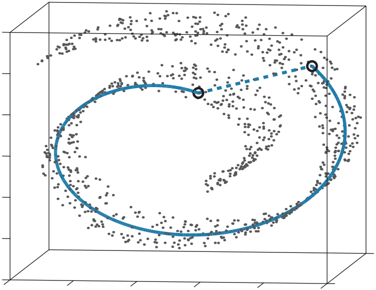
Non linearities



Proximity Graph

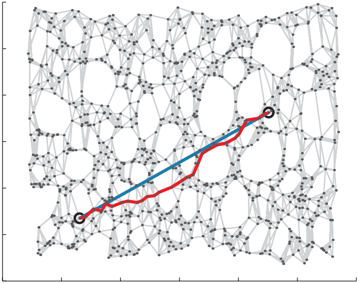
1. Identify neighbors of each data point $\mathbf{x}_i \in N(\mathbf{x_j})$
2. Build graph $\mathbf{P} = \left\{ \underbrace{\mathbf{X}}_{vertexset}, \underbrace{\mathbf{W}}_{edgeset} \right\}$
   - Put edges between vertices's in neighborhood
   - Assume **P** connected (and in most cases symmetric)
3. **Objective:** *Complete* **P** *to make it fully connected*
4. Different algorithms have different strategies
   - What are the edge weights?
   - How to complete **P**

Maximum Variance Unfolding



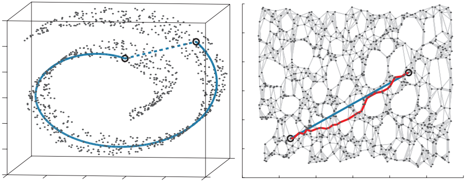
Any “fold” of the manifold between two points will **decrease** the *Euclidean* distance between the points while the *Manifold* distance remains **constant**

Maximum Variance Unfolding



If manifold is **maximally** stretched between two points the *Euclidean* distance will **equal** the *Manifold* distance

Maximum Variance Unfolding

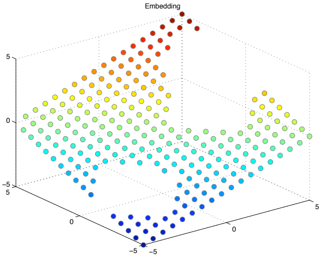


Maximise all pairwise distance outside local neighborhood (upper bound)

$$
\begin{aligned}
&\max \sum_{i=1}^{N} \sum_{j=1}^{N} ||\mathbf{y}_i - \mathbf{y}_j||_{L2}^2 \\
\Rightarrow \quad &\max(\text{trace}(\mathbf{K}))
\end{aligned}
$$

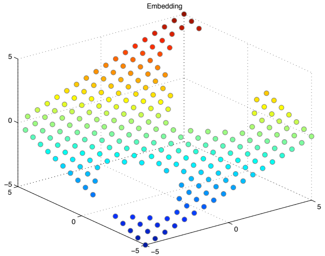
▸ Proof

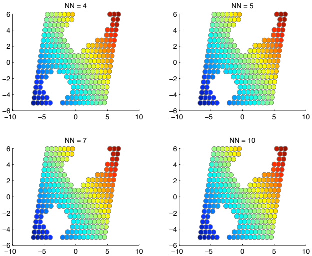
Maximum Variance Unfolding: Example[3]



[3] /algos/mvu_embed.m

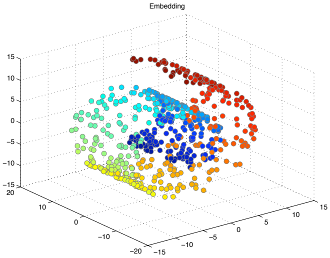
Maximum Variance Unfolding: Example[3]





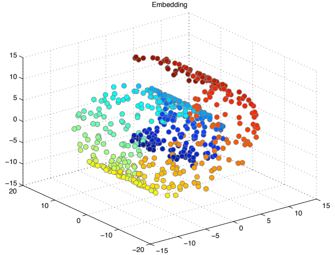
[3] /algos/mvu_embed.m

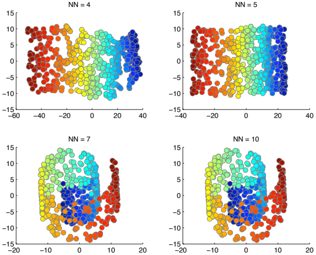
Maximum Variance Unfolding: Example[3]



[3] /algos/mvu_embed.m

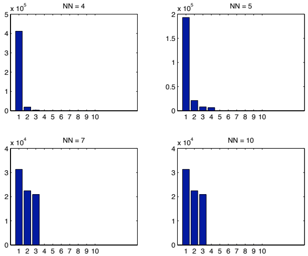
Maximum Variance Unfolding: Example[3]

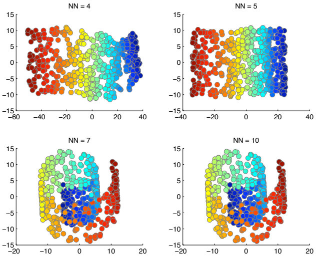




[3] /algos/mvu_embed.m

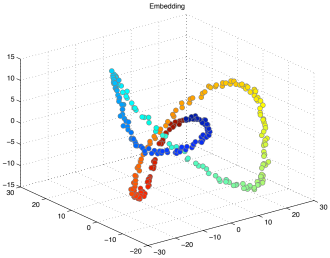
Maximum Variance Unfolding: Example[3]





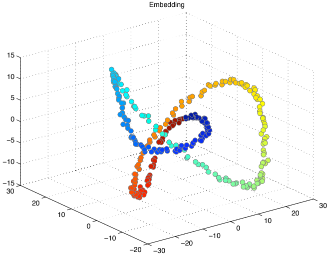
[3] /algos/mvu_embed.m

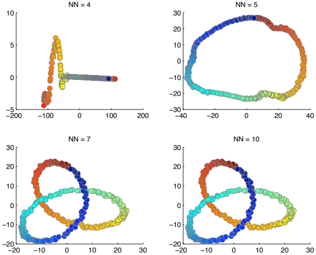
Maximum Variance Unfolding: Example[3]



[3] /algos/mvu_embed.m

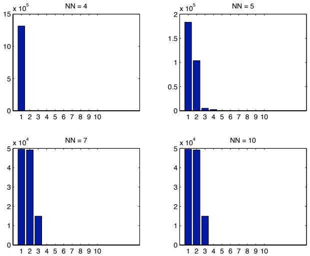
Maximum Variance Unfolding: Example[3]

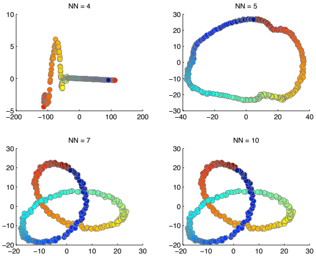




[3] /algos/mvu_embed.m

Maximum Variance Unfolding: Example[3]





[3] /algos/mvu_embed.m

Introduction

Recap

Hierarchical Models

Summary

Outline

- Hierarchical Models
  - motivation
  - history
  - neural networks
  - deep models
  - Why is this exciting?
- Summary of my part

$$f : \mathbf{X} \rightarrow \mathbf{Y} \tag{39}$$

Problem set-up

- Some data **X** (input)
- Some task **Y** (output)
- Estimate mapping from data
- Using a hierarchy

$$f : \mathbf{X} \rightarrow \mathbf{Y} \tag{40}$$

$$\mathbf{X} \rightarrow \mathbf{H}_1 \rightarrow \mathbf{H}_2 \rightarrow \ldots \rightarrow \mathbf{Y} \tag{41}$$

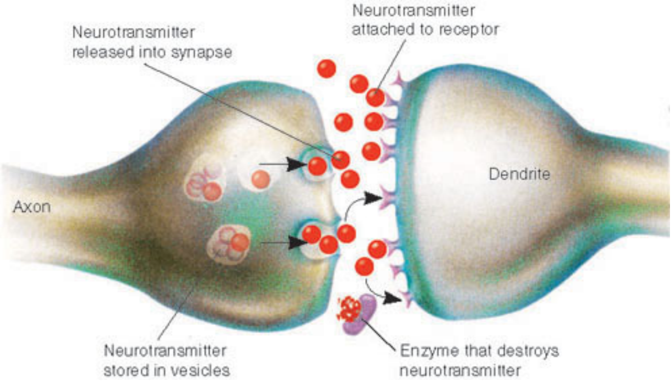
Problem set-up

- Some data $\mathbf{X}$ (input)
- Some task $\mathbf{Y}$ (output)
- Estimate mapping from data
- Using a hierarchy

Standing on the shoulders of giants

`Deep Learning and Neural Networks`

Hierarchical Models



Hierarchical Models

History 1940-1990

- Artificial Neuron McCulloch and Pitts 1943 Rosenblatt 1958
- Only linear functions Minsky and Papert 1969
- Multi-layered Perceptron Rumelhart *et al.* 1986
- Back-propagation

Hierarchical Models

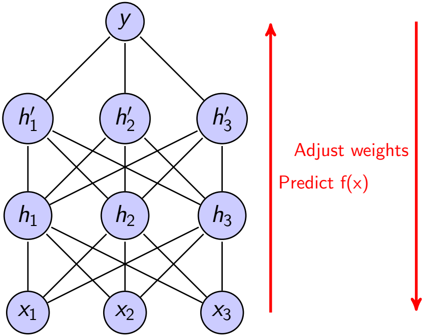
$$y_i = \rho\left(\sum_{j=0}^{N} w_{ij} x_j\right) \tag{42}$$

$$\rho(t) = \frac{1}{1+e^{-t}} \tag{43}$$

Artificial Neuron

- x_j signal j into neuron i
- $w_i j$ weight of signal from j
- ρ activation function

Hierarchical Models



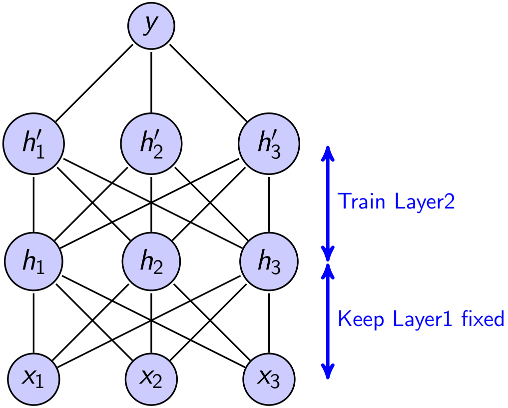
Hierarchical Models

Hierarchical Models

History 2004-2010

- Vanishing Gradients
- Restricted Boltzman Machine
- Layer-wise training Hinton *et al.* 2006
  - *"If you want to do Computer Vision first learn Computer Graphics"*
- Allows for unlabled data

Hierarchical Models



Hierarchical Models

History 2010-

- Heuristic structures
  - Convolutional Neural Networks
- Big-Data
- Infrastructural changes
  - GPUs
  - Distributed computations

Hierarchical Models

Human: "A group of men playing Frisbee in the park."
Computer model: "A group of young people playing a game of Frisbee."

1 of 6

Hierarchical Models

Human: "A young hockey player playing in the ice rink."
Computer model: "Two hockey players are fighting over the puck."

2 of 6

Hierarchical Models

Human: "Three different types of pizza on top of a stove."
Computer model: "A pizza sitting on top of a pan on top of a stove."

5 of 6

Hierarchical Models

26 June 2012 Last updated at 16:03 GMT

Share

Google computer works out how to spot cats

A Google research team has trained a network of 1,000 computers wired up like a brain to recognise cats.

The team built a neural network, which mimics the working of a biological brain, that worked out how to spot pictures of cats in just three days.

The cat-spotting computer was created as part of a larger project to investigate machine learning.

Millions of images were used to train the neural network

How to proceed

- Very active field of research
- Very impressive results
  - on some tasks
- Some science and lots of engineering
- I'll try to give you a flavour of the field
- ... and my opinions

How to proceed

- Very active field of research
- Very impressive results
  - on some tasks
- Some science and lots of engineering
- I'll try to give you a flavour of the field
- ... and my opinions

How to proceed

- Very active field of research
- Very impressive results
  - on some tasks
- Some science and lots of engineering
- I'll try to give you a flavour of the field
- ... and my opinions

How to proceed

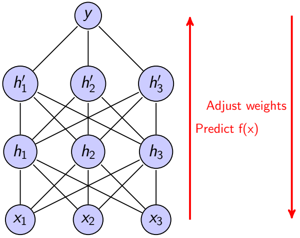
- Very active field of research
- Very impressive results
  - on some tasks
- Some science and lots of engineering
- I'll try to give you a flavour of the field
- ... and my opinions

How to proceed

- Very active field of research
- Very impressive results
  - on some tasks
- Some science and lots of engineering
- I'll try to give you a flavour of the field
- ... and my opinions

Revival of NN

- Back-prop does not handle depth
- Depth requires more data
- Restricted Boltzmann Machine
- Layer-wise training



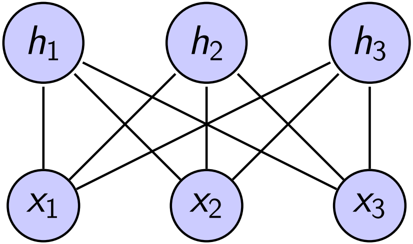
Restricted Boltzmann Machine[4]

$$p(\mathbf{x}, \mathbf{h}|\theta) = \frac{1}{Z(\theta)} \prod_{r}^{R} \prod_{k}^{K} \psi_{rk}(x_r, h_k) \tag{44}$$

- Product of Experts vs. Mixtures of Experts
  - Allows for “sharp” distributions
- $Z(\theta)$ forces normalisation
- Hidden units binary

[4] Murphy 2012, p. 27.7.

Restricted Boltzmann Machine[4]



[4] Murphy 2012, p. 27.7.

Restricted Boltzmann Machine[4]

$$p(\mathbf{h}|\mathbf{x},\theta) = \prod_k p(h_k|\mathbf{x},\theta) \tag{45}$$

$$p(\mathbf{x}|\mathbf{h},\theta) = \prod_r p(x_r|\mathbf{h},\theta) \tag{46}$$

- Variables are conditionally independent
- Learn θ using gradient based means

[4] Murphy 2012, p. 27.7.

Restricted Boltzmann Machine[4]

Binary RBM

$$p(\mathbf{x}, \mathbf{h}|\theta) = \frac{1}{Z(\theta)} e^{-E(\mathbf{x},\mathbf{h};\theta)} \tag{47}$$

$$E(\mathbf{x}, \mathbf{h}; \theta) = -\sum_{r}^{R}\sum_{k}^{K} x_r h_k \tilde{W}_{rk} - \sum_{r}^{R} x_r b_r - \sum_{k}^{K} h_k c_k \tag{48}$$

$$p(\mathbf{h}|\mathbf{x}, \theta) = \prod_{k}^{K} p(h_k|\mathbf{x}, \theta) = \prod_{k}^{K} \mathrm{Ber}(h_k|\mathrm{sigm}(\mathbf{w}_{:,k}\mathbf{x})) \tag{49}$$

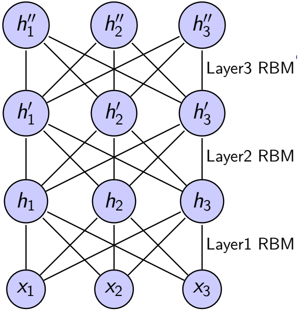
$$\mathbb{E}[\mathbf{h}|\mathbf{x}, \theta] = \mathrm{sigm}(\mathbf{W}^{\mathrm{T}}\mathbf{x}) \tag{50}$$

$$\mathbb{E}[\mathbf{x}|\mathbf{h}, \theta] = \mathrm{sigm}(\mathbf{W}\mathbf{h}) \tag{51}$$

[4] Murphy 2012, p. 27.7.

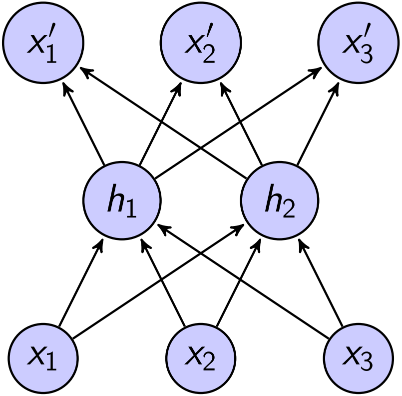
Deep Belief Networks[5]

- Stack several RBMs
- Layer-wise independence
- Each RBM works as a prior for the next level
- *"If you want to do Computer Vision first learn Computer Graphics"*



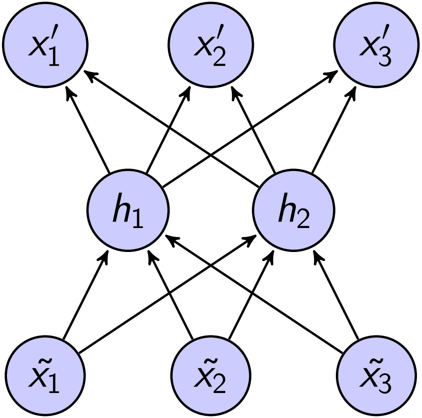
[5] Murphy 2012, p. 28.2.3.

Auto-encoders[6]



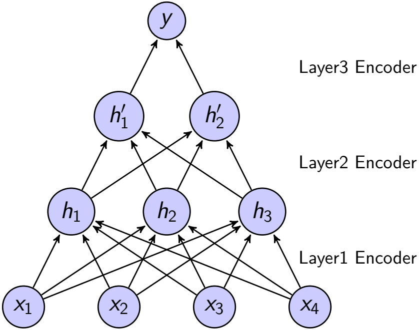
[6] Vincent *et al.* 2010.

Auto-encoders[6]



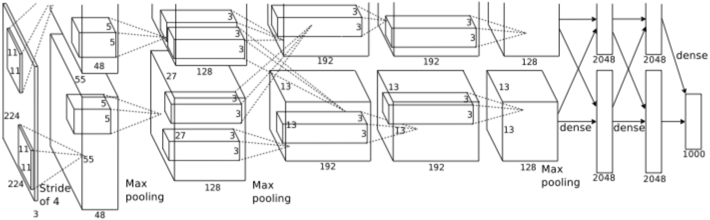
[6]Vincent *et al.* 2010.

Auto-encoders[6]



[6]Vincent *et al.* 2010.

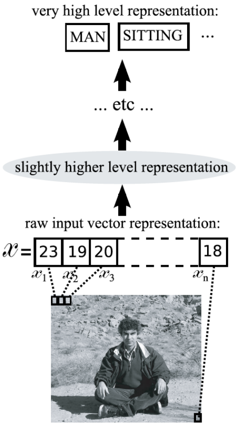
Convolutional Neural Networks[7]



Very structured architecture allows for non-layerwise training

[7] Berkely Caffe

Why[8]



[8]Bengio *et al.* 2013.

Why[8]

[8] Bengio *et al.* 2013.

Why[8]

"It's true there's been a lot of work on trying to apply statistical models to various linguistic problems. I think there have been some successes, but a lot of failures. There is a notion of success which I think is novel in the history of science. It interprets success as approximating unanalyzed data."

[Noam Chomsky]

[8] Bengio *et al.* 2013.

Why[8]

Carls Rant

- These things clearly works
- The science is not to make them work but Why they work
- Quickest short-term progress is often not reached by principles
- We run the risk of disapointing a lot of people by getting lost

[8] Bengio *et al.* 2013.

Deep Gaussian Processes[9]

- Why does a probabilistic model work?
- A good model has sensible priors
- Samples from priors tells us what we prefer to model
- What are hierarchical priors?

[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]

- Why does a probabilistic model work?
- A good model has sensible priors
- Samples from priors tells us what we prefer to model
- What are hierarchical priors?

[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]

- Why does a probabilistic model work?
- A good model has sensible priors
- Samples from priors tells us what we prefer to model
- What are hierarchical priors?

[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]

- Why does a probabilistic model work?
- A good model has sensible priors
- Samples from priors tells us what we prefer to model
- What are hierarchical priors?

[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]

$$f(\mathbf{x}) = \frac{1}{K}\sum_{i}^{K} w_i h_i(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{h}(\mathbf{x}) \tag{52}$$

[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]

$$
\begin{aligned}
f(\mathbf{x}) &= \frac{1}{K}\sum_{i}^{K} w_i h_i(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{h}(\mathbf{x}) && (53)\\
&= \mathbf{w}^{\mathrm{T}}\mathbf{h}^{(2)}(\mathbf{h}^{(1)}(\mathbf{x})) && (54)\\
k_1(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{h}(\mathbf{x}_i)^{\mathrm{T}}\mathbf{h}(\mathbf{x}_j) && (55)\\
k_2(\mathbf{x}_i, \mathbf{x}_j) &= [\mathbf{h}^{(2)}(\mathbf{h}^{(1)}(\mathbf{x}_i))]^{\mathrm{T}}\mathbf{h}^{(2)}(\mathbf{h}^{(1)}(\mathbf{x}_j)) && (56)
\end{aligned}
$$

[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]

$$f(\mathbf{x}) = \frac{1}{K}\sum_{i}^{K} w_i h_i(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{h}(\mathbf{x}) \tag{57}$$

$$= \mathbf{w}^{\mathrm{T}}\mathbf{h}^{(2)}(\mathbf{h}^{(1)}(\mathbf{x})) \tag{58}$$

$$k_1(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{h}(\mathbf{x}_i)^{\mathrm{T}}\mathbf{h}(\mathbf{x}_j) \tag{59}$$

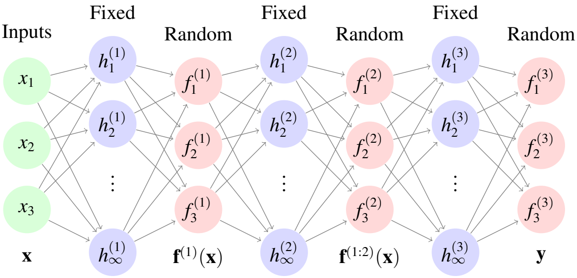
$$k_2(\mathbf{x}_i, \mathbf{x}_j) = [\mathbf{h}^{(2)}(\mathbf{h}^{(1)}(\mathbf{x}_i))]^{\mathrm{T}}\mathbf{h}^{(2)}(\mathbf{h}^{(1)}(\mathbf{x}_j)) \tag{60}$$

$k(\mathbf{x}_i, \mathbf{x}_j)$ has closed form for SE kernel

$$k_{L+1}(\mathbf{x}_i, \mathbf{x}_j) = e^{k_L(\mathbf{x}_i, \mathbf{x}_j) - 1} \tag{61}$$

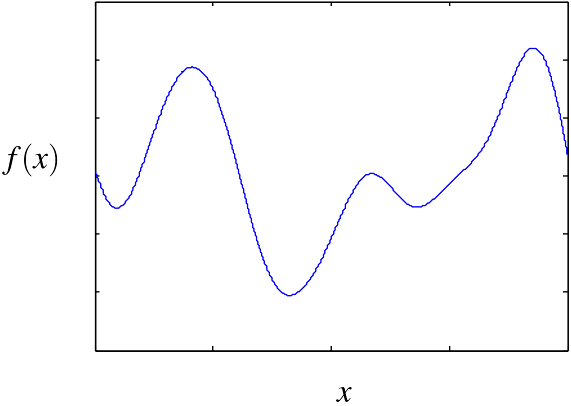
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



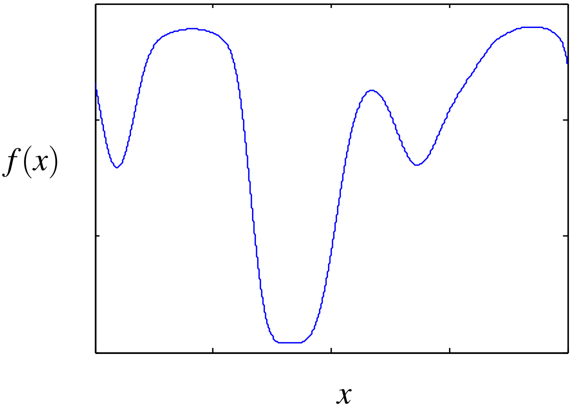
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



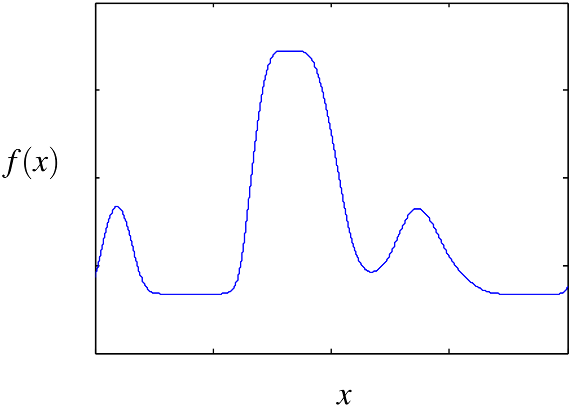
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



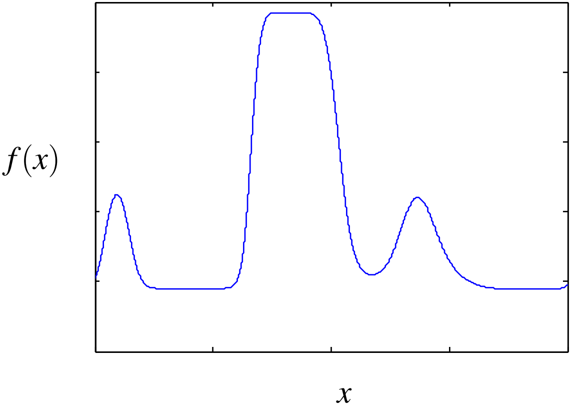
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



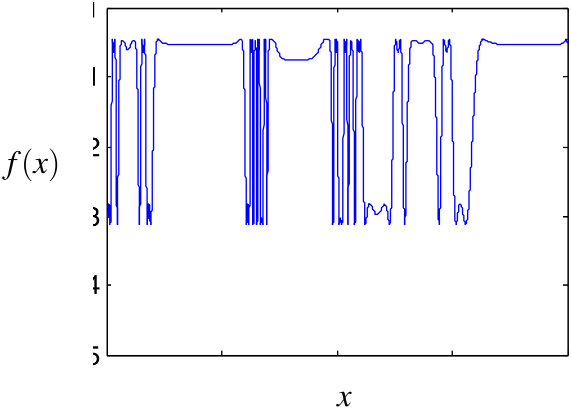
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



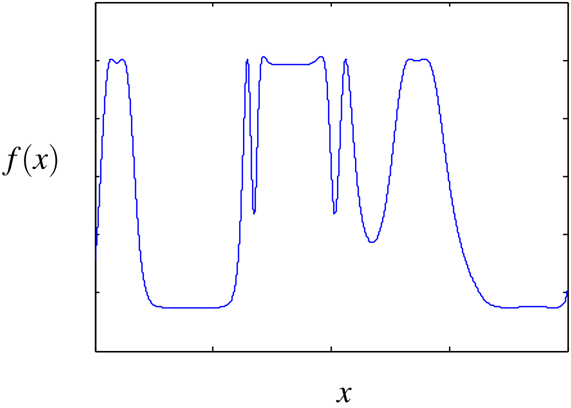
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



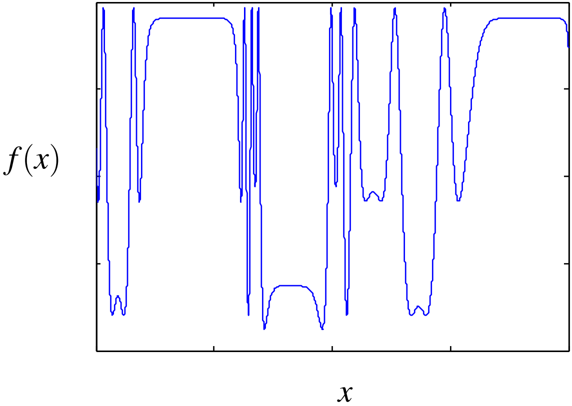
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



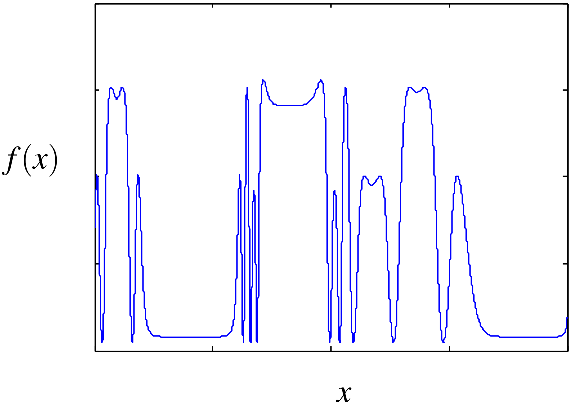
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



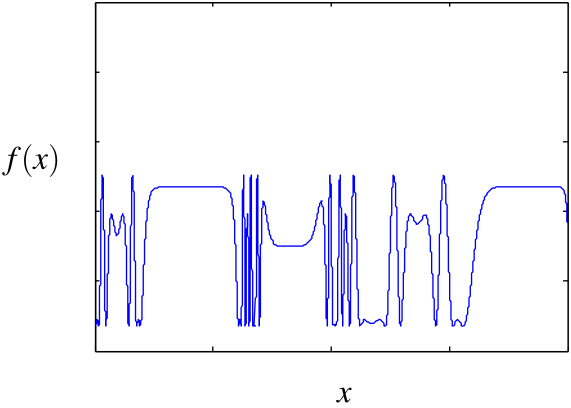
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



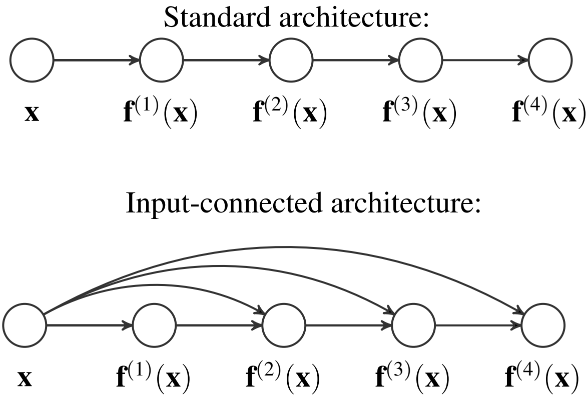
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



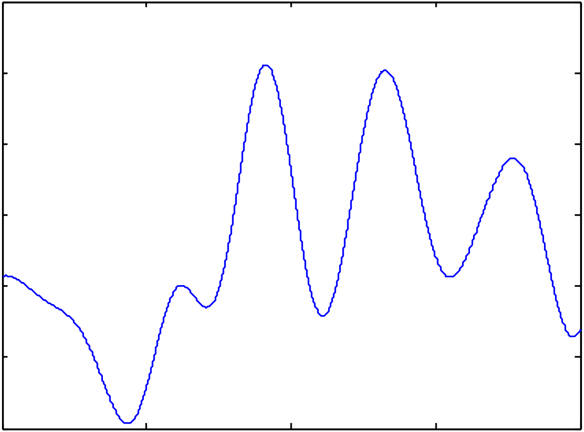
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



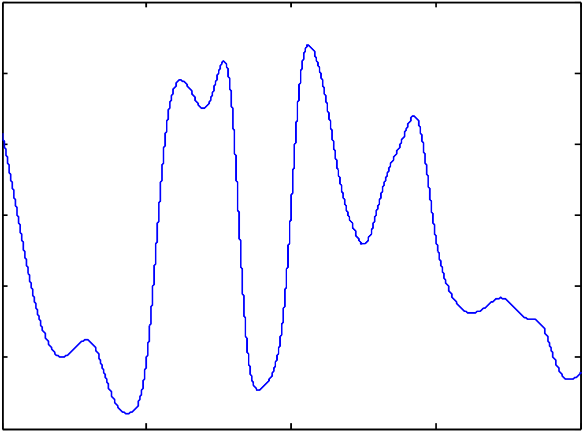
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



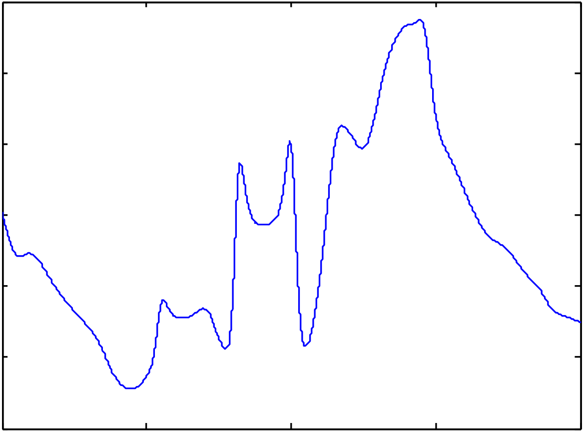
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



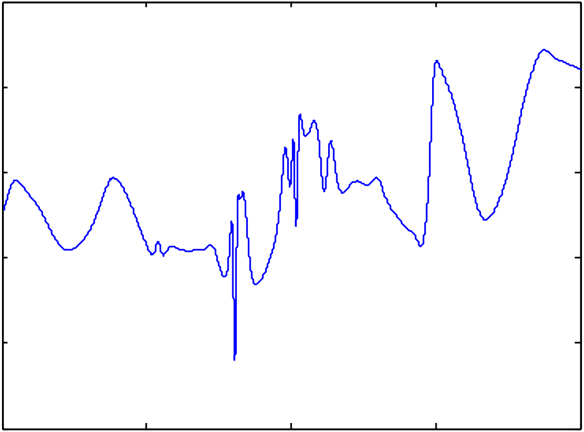
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



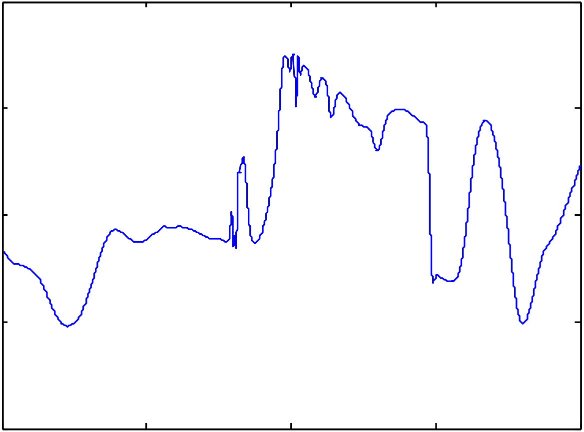
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



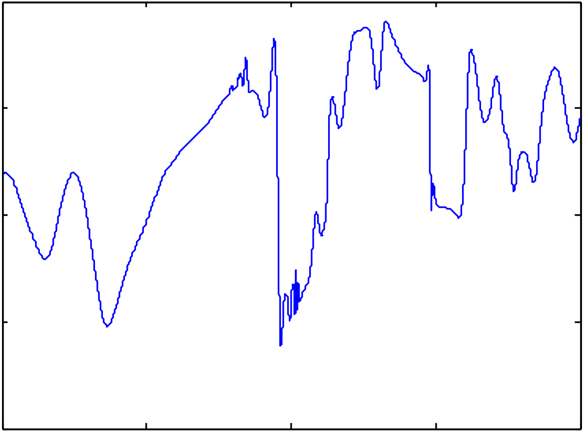
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



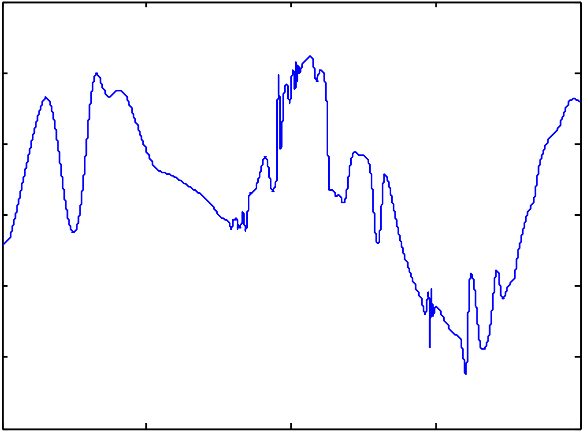
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



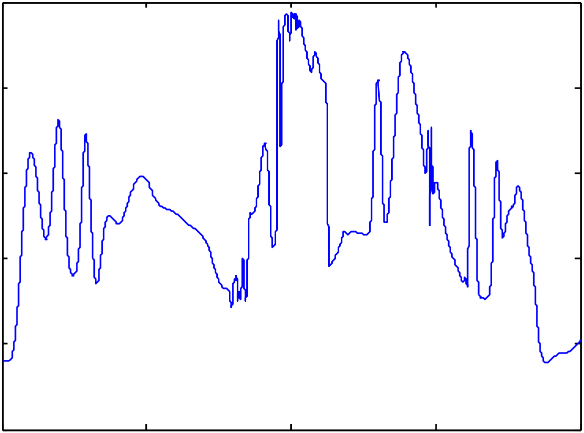
[9]Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



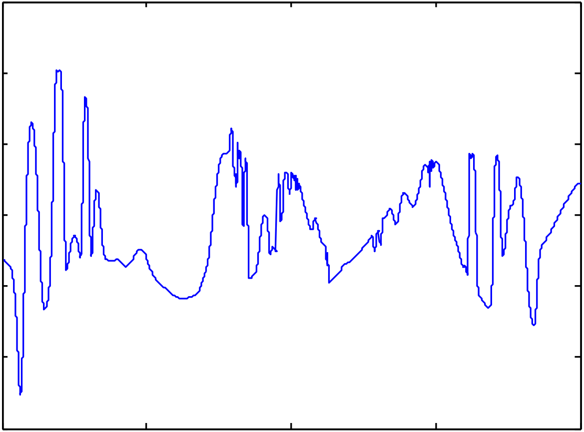
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



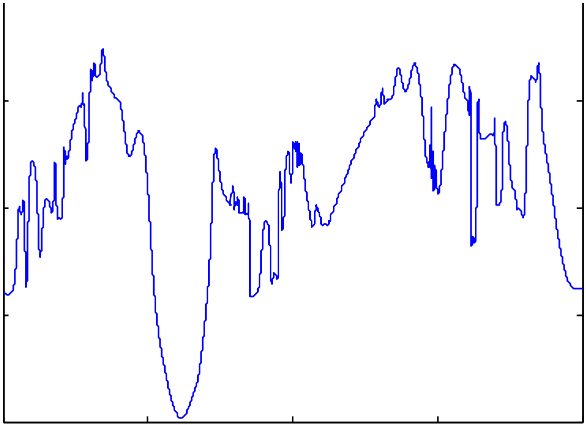
[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]



[9] Duvenaud *et al.* 2014.

Deep Gaussian Processes[9]

- Priors allows us to analyse design before seeing data
- Deep GPs shows what depth provides
  - non-stationary functions
- Allows for deep models on small data-sets
- Shed light on some current design heuristics

[9] Duvenaud *et al.* 2014.

Future

- If we have enough data we do not need priors (Laplace)
- which interesting problems do we have that for?
- no priors (or not formulated priors) makes us headless chickens
- when we need a lot of data to solve a simple problem you should be worried

Future

- If we have enough data we do not need priors (Laplace)
- which interesting problems do we have that for?
- no priors (or not formulated priors) makes us headless chickens
- when we need a lot of data to solve a simple problem you should be worried

Future

- If we have enough data we do not need priors (Laplace)
- which interesting problems do we have that for?
- no priors (or not formulated priors) makes us headless chickens
- when we need a lot of data to solve a simple problem you should be worried

Future

- If we have enough data we do not need priors (Laplace)
- which interesting problems do we have that for?
- no priors (or not formulated priors) makes us headless chickens
- when we need a lot of data to solve a simple problem you should be worried

End of Part 2

- Bayesian modelling
  - specify likelihood and prior
  - inference through posterior
- Strength of priors
- Sensible assumptions and approximations (MAP, ML, Variational)
- We have been very abstract on purpose to focus on understanding learning [Chomsky]

What do you need to do?

- Translate to your own problems/data
- How have you solved problems before, thing of the assumptions you made
- What are sensible priors/likelihoods/structures
- What assumptions do I need to make?
- Don't be afraid of being abstract, when you get too close to the problem you often make assumptions that you are not aware of
- Get your hands dirty, i.e. develop your own priors for developing models

What do you need to do?

- Translate to your own problems/data
- How have you solved problems before, thing of the assumptions you made
- What are sensible priors/likelihoods/structures
- What assumptions do I need to make?
- Don't be afraid of being abstract, when you get too close to the problem you often make assumptions that you are not aware of
- Get your hands dirty, i.e. develop your own priors for developing models

Take home message

- Machine learning is really simple, it should be as even Carl have learnt quite a few things in life
- Formulating learning so that it can be externalised might be very hard and really involved but that is just labour
- Make assumptions, lots of them, that is the basis of learning, but be aware of them

Take home message

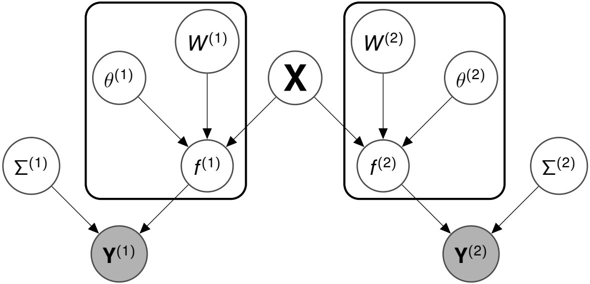
- Machine learning is really simple, it should be as even Carl have learnt quite a few things in life
- Formulating learning so that it can be externalised might be very hard and really involved but that is just labour
- Make assumptions, lots of them, that is the basis of learning, but be aware of them

Take home message

- Machine learning is really simple, it should be as even Carl have learnt quite a few things in life
- Formulating learning so that it can be externalised might be very hard and really involved but that is just labour
- Make assumptions, lots of them, that is the basis of learning, but be aware of them

e.o.f.

My Research


$W^{(1)}$
$\theta^{(1)}$
$\mathbf{X}$
$W^{(2)}$
$\theta^{(2)}$
$\Sigma^{(1)}$
$f^{(1)}$
$f^{(2)}$
$\Sigma^{(2)}$
$\mathbf{Y}^{(1)}$
$\mathbf{Y}^{(2)}$

References I

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020, 9780262018029.

Neil D Lawrence. "Probabilistic non-linear principal component analysis with Gaussian process latent variable models". In: *The Journal of Machine Learning Research* 6 (2005), pp. 1783–1816. URL: http://dl.acm.org/citation.cfm?id=1194904.

References II

- Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". English. In: *The Bulletin of Mathematical Biophysics* 5.4 (Dec. 1943), pp. 115–133. DOI: 10.1007/BF02478259. URL: http://link.springer.com/10.1007/BF02478259.
- F Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". In: *Psychology Review* (Nov. 1958), pp. 386–408. URL: http://www.ncbi.nlm.nih.gov/pubmed/13602029.

References III

- Marvin Minsky and Seymour Papert. "Perceptrons. An Introduction to Computational Geometry. " English. In: *Science* 165.3895 (Aug. 1969), pp. 780–782. DOI: 10.1126/science.165.3895.780. URL: http://www.sciencemag.org/cgi/doi/10.1126/science.165.3895.780.
- D E Rumelhart *et al.* "Learning representations by back-propagating errors". In: *Nature* 323.9 (Oct. 1986), pp. 533–536. URL: http://www.iro.umontreal.ca/~pift6266/A06/refs/backprop_old.pdf.

References IV

Geoffrey E Hinton *et al.* "A Fast Learning Algorithm for Deep Belief Nets". English. In: *Neural Computation* 18.7 (July 2006), pp. 1527–1554. DOI: 10.1162/jmlr.2003.4.7-8.1235. URL: http://www.mitpressjournals.org/doi/abs/10.1162/neco.2006.18.7.1527.

Pascal Vincent *et al.* "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *The Journal of Machine Learning Research* 11 (Mar. 2010), pp. 3371–3408. URL: http://dl.acm.org/citation.cfm?id=1756006.1953039.

References V

Yoshua Bengio *et al.* "Representation learning: A review and new perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (Aug. 2013), pp. 1798–1828. ISSN: 0162-8828. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6472238.

David Duvenaud *et al.* *Avoiding pathologies in very deep networks*. 2014. URL: http://jmlr.org/proceedings/papers/v33/duvenaud14.pdf.

Appendix

Similar Matrices: Self-Similarity

$$\mathbf{A} = \mathbf{I}\mathbf{A}\mathbf{I}^{-1} = \mathbf{I}^{-1}\mathbf{A}\mathbf{I}$$

Return

Similar Matrices: Symmetry

$$\begin{aligned}
\mathbf{A} \quad \sim \quad & \mathbf{B} \Rightarrow \mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} \\
\det\mathbf{B} \quad & = \det\left(\mathbf{P}^{-1}\mathbf{A}\mathbf{P}\right) = \det(\mathbf{P}^{-1})\det(\mathbf{A})\det(\mathbf{P}) = \\
& = \det(\mathbf{A})\det(\mathbf{P}^{-1})\det(\mathbf{P}) = \det(\mathbf{A})\frac{1}{\det(\mathbf{P})}\det(\mathbf{P}) = \\
& \det(\mathbf{B})
\end{aligned}$$

Return

Similar Matrices: Trace

$$\begin{aligned} \mathbf{A} &\sim \mathbf{B} \Rightarrow \mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} \\ \text{trace}(\mathbf{B}) &= \text{trace}(\mathbf{P}^{-1}\mathbf{A}\mathbf{P}) = \{\text{trace}(\mathbf{A}\mathbf{B}) = \text{trace}(\mathbf{A}\mathbf{B})\} = \\ &= \text{trace}\left(\left(\mathbf{P}\mathbf{P}^{-1}\right)\mathbf{A}\right) = \text{trace}(\mathbf{A}) \end{aligned}$$

Return

Similar Matrices: Power

$$
\begin{aligned}
\mathbf{A} \quad &\sim \quad \mathbf{B} \Rightarrow \mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} \\
\mathbf{B}^2 \quad &= \quad \left(\mathbf{P}^{-1}\mathbf{A}\mathbf{P}\right)^2 = \left(\mathbf{P}^{-1}\mathbf{A}\mathbf{P}\right)\left(\mathbf{P}^{-1}\mathbf{A}\mathbf{P}\right) = \\
&= \left(\mathbf{P}^{-1}\mathbf{A}\right)\left(\underbrace{\mathbf{P}\mathbf{P}^{-1}}_{=\mathbf{I}}\right)(\mathbf{A}\mathbf{P}) = \\
&= \mathbf{P}^{-1}\mathbf{A}\mathbf{A}\mathbf{P} = \mathbf{P}^{-1}\mathbf{A}^2\mathbf{P}
\end{aligned}
$$

Prove further powers by induction over exponent

Return

Similar Matrices: Invertability

$$\begin{aligned} \mathbf{A} \quad \sim \quad & \mathbf{B} \Rightarrow \mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} \\ \Rightarrow \quad & \det(\mathbf{A}) = \det(\mathbf{B}) \end{aligned}$$

$\mathbf{A}^{-1}$ Exists if $\det(\mathbf{A}) \neq 0$

$$\det(\mathbf{B}) \neq 0 \iff \det(\mathbf{A}) \neq 0$$

Return

$$\begin{aligned}\mathbf{A}_{ij} &= \sum_{k=1}^{N} \mathbf{V}_{ik}\mathbf{D}_{kk}\left(\mathbf{V}^{T}\right)_{kj} = \sum_{k=1}^{N} (\mathbf{v}_k)_i\,\lambda_k\,(\mathbf{v}_k)_j \\ &= \sum_{k=1}^{N}\left(\lambda_k \mathbf{v}_k \mathbf{v}_k^T\right)_{ij}\end{aligned}$$

Return

Rank Approximation

$$
\begin{aligned}
||\mathbf{A}-\mathbf{B}||_F &= ||\sum_{i=1}^{N}\lambda_i\mathbf{v}_i\mathbf{v}_i^T - \sum_{i=1}^{N} q_i\mathbf{v}_i\mathbf{v}_i^T||_F = \\
&= ||\sum_{i=1}^{N}(\lambda_i - q_i)\mathbf{v}_i\mathbf{v}_i^T|| = \\
&= \left\{ ((\lambda_i - q_i)\mathbf{v}_i\underbrace{\mathbf{v}_i^T)\mathbf{v}_i}_{=1} = (\lambda_i - q_i)\mathbf{v}_i \right\} = \\
&= \sqrt{\sum_{i=1}^{N}(\lambda_i - q_i)^2}
\end{aligned}
$$

◂ Return

Multidimensional Scaling

Define:

$$
\begin{aligned}
d_{ij}^2 &= \sum_{k=1}^{d} (x_{ki} - x_{kj})^2 = \mathbf{x}_i^T \mathbf{x}_i + \mathbf{x}_j^T \mathbf{x}_j - 2\mathbf{x}_i \mathbf{x}_j \\
g_{ij} &= \sum_{k=1}^{d} x_{ki} x_{kj} = \mathbf{x}_i^T \mathbf{x}_j \\
&\Rightarrow \quad d_{ij}^2 = g_{ii} + g_{jj} - 2g_{ij} \\
\text{Centering:} \quad & \sum_{i=1}^{N} g_{ij} = \sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{x}_j = \underbrace{\left(\sum_{i=1}^{N} \mathbf{x}_i^T\right)}_{=\mathbf{0}} \mathbf{x}_j = 0
\end{aligned}
$$

Multidimensional Scaling

Want to Express **G** in terms of **D**

$$g_{ij} = \frac{1}{2}(g_{ii} + g_{jj} - d_{ij}^2)$$

$$\frac{1}{N}\sum_{i=1}^{N} d_{ij}^2 = g_{jj} + \frac{1}{N}\sum_{i=1}^{N} g_{ii}$$

$$\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 = \frac{2}{N}\sum_{i=1}^{N} g_{ii}$$

$$\Rightarrow g_{ij} = \frac{1}{2}\left(\frac{1}{N}\left(\sum_{k=1}^{N} d_{kj}^2 + \sum_{k=1}^{N} d_{ik}^2 - \frac{1}{N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2\right) - d_{ij}^2\right)$$

◂ Return: MDS ◂ Return: MVU

PCA MDS Equivalence

$$
\begin{aligned}
\mathbf{G} &= \mathbf{X}\mathbf{X}^T = \mathbf{V}\Lambda\mathbf{V}^T \\
&\Rightarrow (\mathbf{X}\mathbf{X}^T)\mathbf{v}_i = \lambda_i \mathbf{v}_i \\
&\Rightarrow \frac{1}{N-1}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)\mathbf{v}_i = \lambda_i \frac{1}{N-1}\mathbf{X}^T\mathbf{v}_i \\
&\Rightarrow \underbrace{\frac{1}{N-1}\mathbf{X}^T(\mathbf{X}}_{\mathbf{S}}\mathbf{X}^T)\mathbf{v}_i = \lambda_i \frac{1}{N-1}\mathbf{X}^T\mathbf{v}_i \\
&\Rightarrow \mathbf{S}\underbrace{(\mathbf{X}^T\mathbf{v}_i)}_{\text{eigenvectors?}} = \underbrace{\frac{\lambda_i}{N-1}}_{\text{eigenvalue?}}\underbrace{(\mathbf{X}^T\mathbf{v}_i)}_{\textit{eigenvector}?}
\end{aligned}
$$

PCA MDS Equvalence

Enforce orthogonality

$$
\begin{aligned}
& \left(\mathbf{X}^T\mathbf{v}_i\right)^T\left(\mathbf{X}^T\mathbf{v}_i\right) = \mathbf{v}_i^T\mathbf{X}\mathbf{X}^T\mathbf{v}_i = \lambda_i \\
\Rightarrow \quad & \frac{1}{\sqrt{\lambda_i}}\mathbf{v}_i^T\mathbf{X}\mathbf{X}^T\mathbf{v}_i\frac{1}{\sqrt{\lambda_i}} = \left(\frac{1}{\sqrt{\lambda_i}}\right)^2\lambda_i = 1 \\
& \left(\mathbf{X}^T\mathbf{v}_i\right)\frac{1}{\sqrt{\lambda_i}})^T\left(\mathbf{X}^T\mathbf{v}_i\frac{1}{\sqrt{\lambda_i}}\right) = 1
\end{aligned}
$$

PCA MDS Equivalence

$$
\begin{aligned}
\text{Define: } \mathbf{v}_i^{\text{PCA}} &= \mathbf{X}^T \mathbf{v}_i \frac{1}{\sqrt{\lambda_i}} \\
\mathbf{y}_i^{\text{PCA}} &= \mathbf{X}\mathbf{v}_i^{\text{PCA}} = \mathbf{X}\mathbf{X}^T \mathbf{v}_i \frac{1}{\sqrt{\lambda_i}} = \\
&= \lambda_i \mathbf{v}_i \frac{1}{\sqrt{\lambda_i}} = \sqrt{\lambda_i}\mathbf{v}_i \\
\mathbf{y}_i^{\text{MDS}} &= \mathbf{v}_i \sqrt{\lambda_i} = \sqrt{\lambda_i}\mathbf{v}_i \\
&\Rightarrow \mathbf{y}_i^{\text{PCA}} = \mathbf{y}_i^{\text{MDS}}
\end{aligned}
$$

◂ PCA

Maximum Variance Unfolding: Objective

$$
\begin{aligned}
\sum_{i=1}^{N} g_{ii} &= \sum_{i=1}^{N} \frac{1}{2}\left(\frac{1}{N}\left(\sum_{k=1}^{N} d_{kj}^2 + \sum_{k=1}^{N} d_{ik}^2 - \frac{1}{N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2\right) - d_{ii}^2\right) = \\
&= \underbrace{\frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{N} d_{ki}^2 + \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{N} d_{ik}^2 -}_{\text{symmetry } = \frac{1}{2N} 2 \sum_{i=1}^{N}\sum_{k=1}^{N} d_{ki}^2} \\
&- \frac{1}{2N^2} N \sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2 - \frac{1}{2}\sum_{i}^{N} \underbrace{d_{ii}^2}_{=0} =
\end{aligned}
$$

Maximum Variance Unfolding: Objective

$$\begin{aligned}
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{N} d_{ki}^2 - \frac{1}{2N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2 = \\
&= \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 \\
\mathrm{trace}(\mathbf{G}) &= \sum_{i=1}^{N} g_{ii} = \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 = \\
&= \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} ||\mathbf{y}_i - \mathbf{y}_j||_{L2}^2
\end{aligned}$$

Return

Maximum Variance Unfolding: Centering

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} g_{ii} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{2}\left(\frac{1}{N}\left(\sum_{k=1}^{N} d_{kj}^2 + \sum_{k=1}^{N} d_{ik}^2 - \right.\right. \\
&- \left.\left.\frac{1}{N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2\right) - d_{ij}^2\right) = \\
&= \frac{1}{2N}\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} d_{kj}^2}_{=N\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2} + \frac{1}{2N}\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} d_{ik}^2}_{=N\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2} -
\end{aligned}
$$

Maximum Variance Unfolding: Centering

$$
\begin{aligned}
- \quad & \underbrace{\frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2}_{=N^2\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 = \\
= \quad & \underbrace{\left(\frac{1}{2}+\frac{1}{2}-\frac{1}{2}-\frac{1}{2}\right)}_{=0}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 = 0
\end{aligned}
$$

$$
||\sum_{i=1}^{N}\mathbf{y}_i||_{L2}^2 \quad \Rightarrow \quad \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{ij} = 0
$$

Return

Spectral Theorem

$$\mathbf{x}^T\mathbf{A}\mathbf{x} \qquad \mathbf{A} = \mathbf{V}\Delta\mathbf{V}^T,\ ||\mathbf{x}||_{L2} = 1$$

$$\begin{aligned}
\mathbf{x} &= 1\sum_{i=1}^{N}\alpha_i\mathbf{v}_i \\
&\quad ||\alpha|| = 1 \\
\mathbf{x}^T\mathbf{A}\mathbf{x} &= \left(\sum_{i=1}^{N}\alpha_i\mathbf{v}_i\right)^T \mathbf{A}\left(\sum_{i=1}^{N}\alpha_i\mathbf{v}_i\right) = \\
&= \left(\sum_{i=1}^{N}\alpha_i\mathbf{v}_i\right)^T \left(\sum_{i=1}^{N}\lambda_i\mathbf{v}_i\mathbf{v}_i^T\right)\left(\sum_{i=1}^{N}\alpha_i\mathbf{v}_i\right) =
\end{aligned}$$

Spectral Theorem

$$
\begin{aligned}
&= \left(\sum_{i=1}^{N}\alpha_i\mathbf{v}_i\right)^T\left(\sum_{i=1}^{N}\lambda_i\mathbf{v}_i\mathbf{v}_i^T\right)\left(\sum_{i=1}^{N}\alpha_i\mathbf{v}_i\right) = \\
&= \left\{\mathbf{v}_i^T\mathbf{v}_j = \left\{\begin{array}{cc} 1 & i=j \\ 0 & \text{otherwise} \end{array}\right.\right\} = \\
&= \sum_{i=1}^{N}\alpha_i^2\lambda_i\underbrace{\mathbf{v}_i^T\mathbf{v}_i}_{=1}\underbrace{\mathbf{v}_i^T\mathbf{v}_i}_{=1} = \\
&= \sum_{i=1}^{N}\alpha_i^2\lambda_i\left\{\begin{array}{cccc} \max & : & \mathbf{x}^T\mathbf{A}\mathbf{x} = \lambda_1 & \mathbf{x} = \mathbf{v}_1 \\ \min & : & \mathbf{x}^T\mathbf{A}\mathbf{x} = \lambda_N & \mathbf{x} = \mathbf{v}_N \end{array}\right.
\end{aligned}
$$

◂ Return LLE ◂ Return Laplacian

Maximum Variance Unfolding: Objective

$$\sum_{i=1}^{N} g_{ii} = \sum_{i=1}^{N} \frac{1}{2}\left(\frac{1}{N}\left(\sum_{k=1}^{N} d_{kj}^2 + \sum_{k=1}^{N} d_{ik}^2 - \frac{1}{N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2\right) - d_{ii}^2\right) =$$

$$= \underbrace{\frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{N} d_{ki}^2 + \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{N} d_{ik}^2}_{\text{symmetry } = \frac{1}{2N} 2 \sum_{i=1}^{N} \sum_{k=1}^{N} d_{ki}^2} -$$

$$- \frac{1}{2N^2} N \sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2 - \frac{1}{2}\sum_{i}^{N} \underbrace{d_{ii}^2}_{=0} =$$

Maximum Variance Unfolding: Objective

$$
\begin{aligned}
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{N} d_{ki}^2 - \frac{1}{2N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2 = \\
&= \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 \\
\mathrm{trace}(\mathbf{G}) &= \sum_{i=1}^{N} g_{ii} = \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 = \\
&= \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} ||\mathbf{y}_i - \mathbf{y}_j||_{L2}^2
\end{aligned}
$$

Return

Maximum Variance Unfolding: Centering

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} g_{ii} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{2}\left(\frac{1}{N}\left(\sum_{k=1}^{N} d_{kj}^2 + \sum_{k=1}^{N} d_{ik}^2 - \right.\right. \\
&- \left.\left.\frac{1}{N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2\right) - d_{ij}^2\right) = \\
&= \frac{1}{2N}\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} d_{kj}^2}_{=N\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2} + \frac{1}{2N}\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} d_{ik}^2}_{=N\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2} -
\end{aligned}
$$

Maximum Variance Unfolding: Centering

$$
\begin{aligned}
- & \underbrace{\frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{p=1}^{N} d_{kp}^2}_{=N^2\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 = \\
= & \underbrace{\left(\frac{1}{2}+\frac{1}{2}-\frac{1}{2}-\frac{1}{2}\right)}_{=0}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2 = 0
\end{aligned}
$$

$$
\|\sum_{i=1}^{N}\mathbf{y}_i\|_{L2}^2 \quad \Rightarrow \quad \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{ij} = 0
$$

Return