A Brief Introduction to
Generative Models

Jörg Lücke

FIAS, Goethe-Universität Frankfurt, Germany

FIAS Summer School
Frankfurt, August 2008

Contents

Please note that this talk was supported by derivations of formulas on the blackboard (e.g., EM-related) and by numerical demonstrations of different algorithms.

Introduction

What are generative models?

What is modelled? Data.

What is generated? Data.

A generative model is a model of data – nothing more.

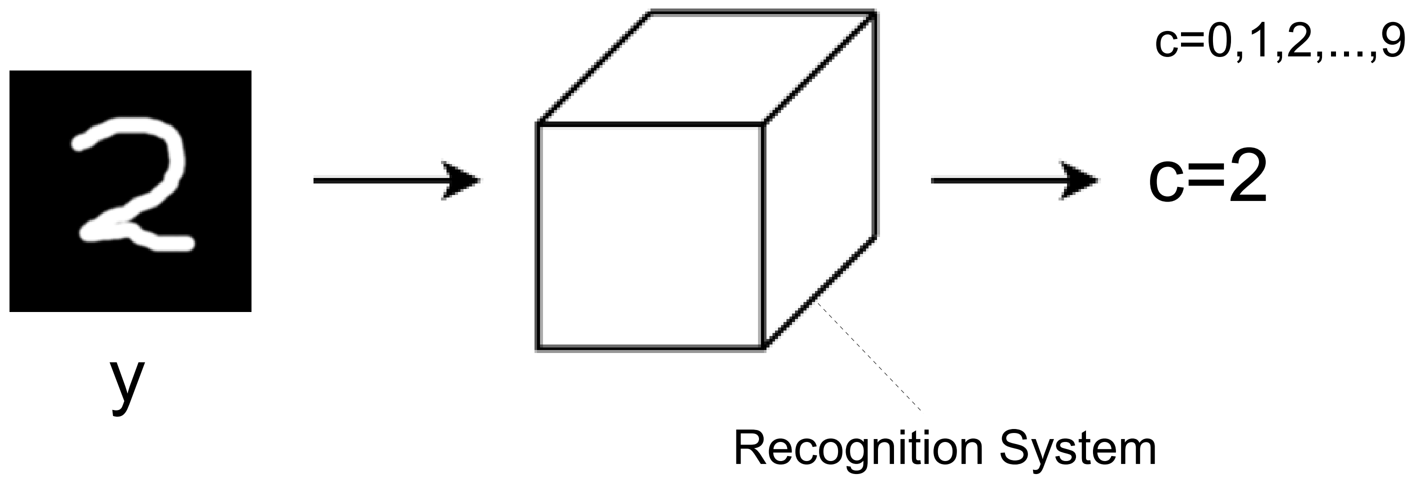
So we could actually stop at this point, or couldn't we?

Introduction

What are generative models used for?

Inference – given an input a generative model allows to extract \`higher-level’ knowledge

Example 1

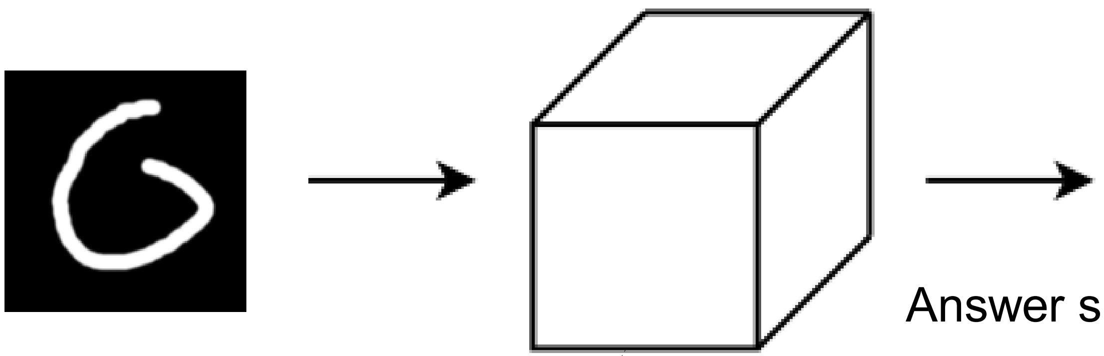


Introduction

What are generative models used for?

Inference – given an input a generative model allows to extract `higher-level’ knowledge

Example 1



y

Recognition System

c=0,1,2,...,9

c=0 or c=6?

Answer should be probabilistic:

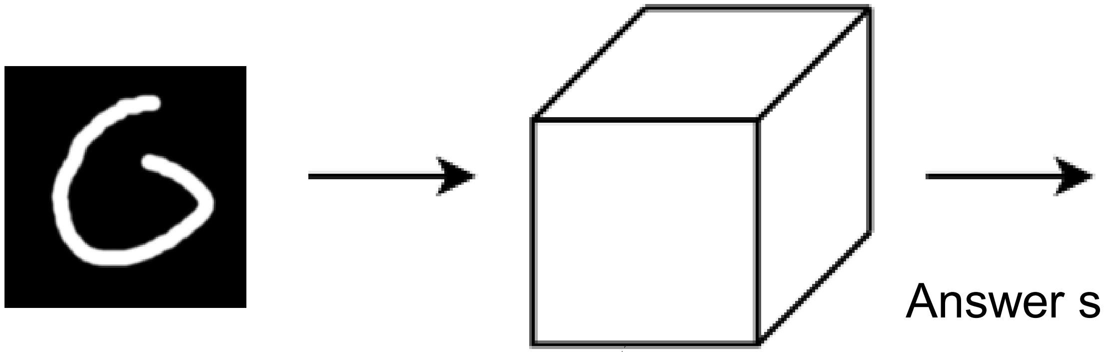
c=0	p=0.61...
c=1	p=0.0...01
c=2	p=0.0...01
...	
c=6	p=0.38...
c=7	p=0.0...01
...	

Introduction

What are generative models used for?

Inference – given an input a generative model allows to extract \`higher-level' knowledge

Example 1



c=0,1,2,...,9

c=0 or c=6?

y

Recognition System

Answer should be probabilistic.

Posterior probability:

$$p(c|y)$$

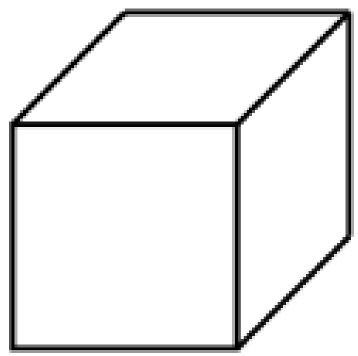
Introduction

What are generative models used for?

Inference – given an input a generative model allows to extract \`higher-level’ knowledge

Example 1

y



c=0,1,2,...,9

c=0 or c=6?

$$p(c|y) = \frac{p(y|c)\, p(c)}{\sum_{c'} p(y|c')\, p(c')}$$

Generative model + Bayes’ rule

→ posterior probability p(c|y)

Introduction

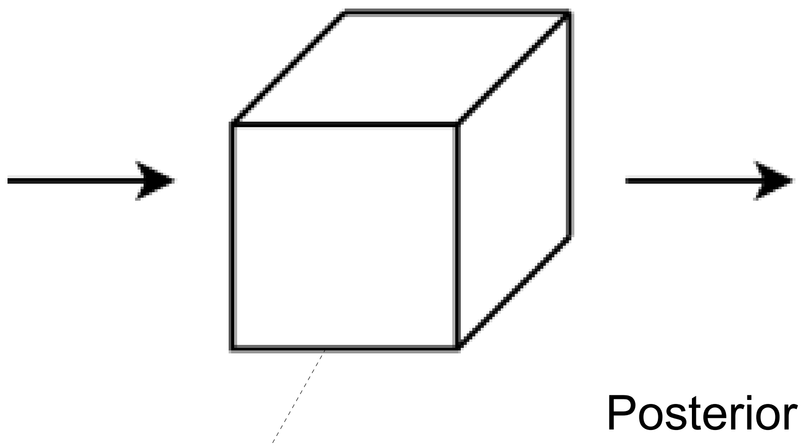
What are generative models used for?

Inference – given an input a generative model allows to extract \`higher-level' knowledge

Example 2

y

Image taken from
Bishop, ECCV '04



c

c=car, aeroplane, tree, ...

Recognition System

Posterior probability:

$$p(c|y)$$

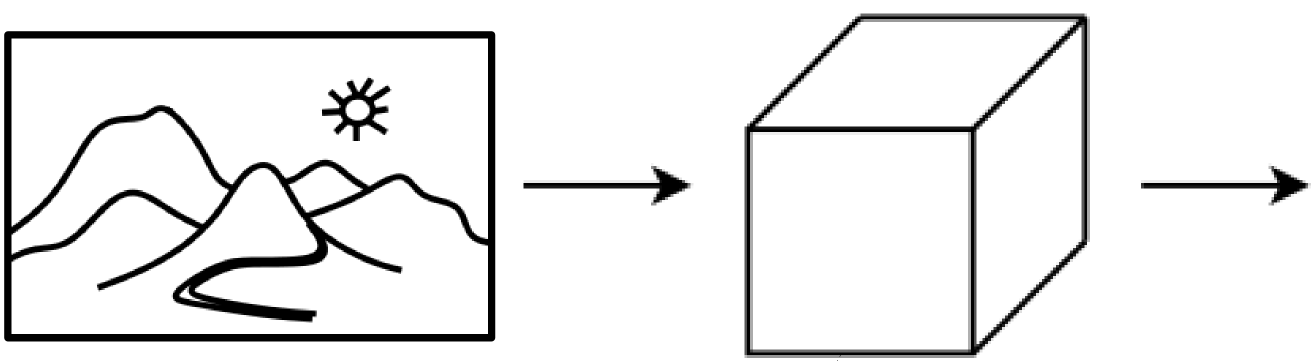
Introduction

What are generative models used for?

Inference – given an input a generative model allows to extract \`higher-level' knowledge

c = hills with street and sun,
sandcastle with hedgehog,
snake with ...,
synapse and transmitters ...

Example 2



c

y

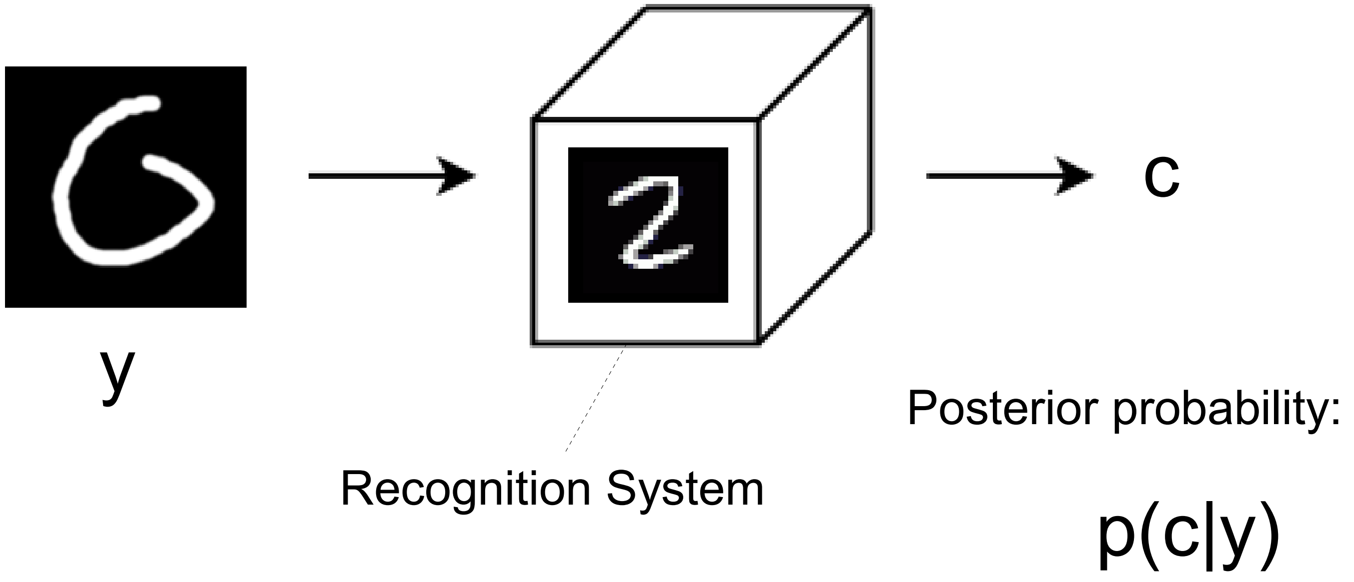
Recognition System

Posterior probability:

$$p(c|y)$$

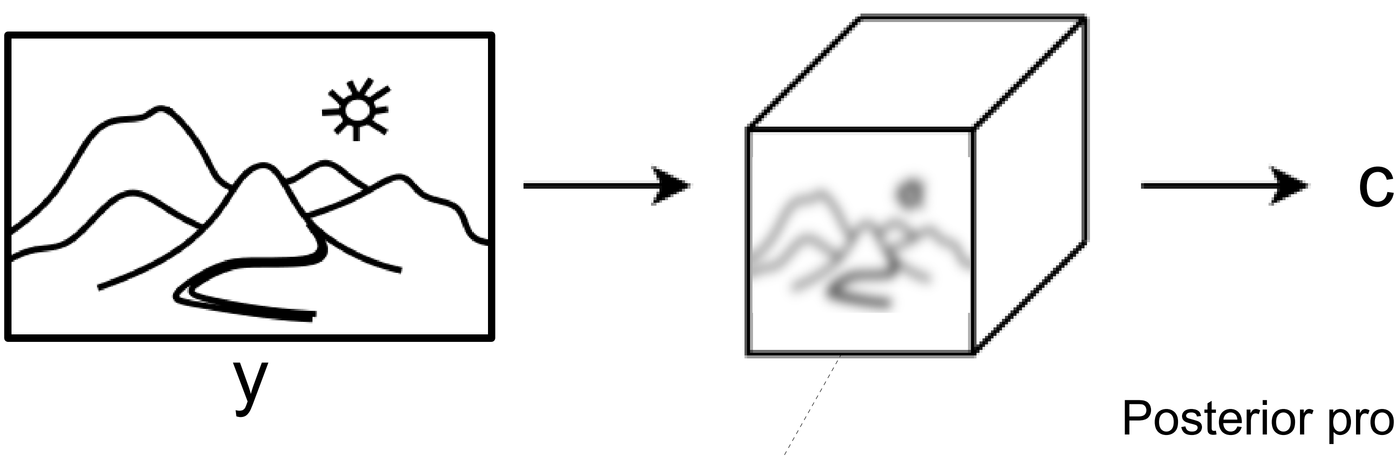
Introduction

Generative models try to infer knowledge from input using an explicit representation of the input.



Introduction

Generative models try to infer knowledge from input using an explicit representation of the input.



Recognition System

Posterior probability:

$$p(c|y)$$

Introduction

Generative models try to infer knowledge from input using an explicit representation of the input.

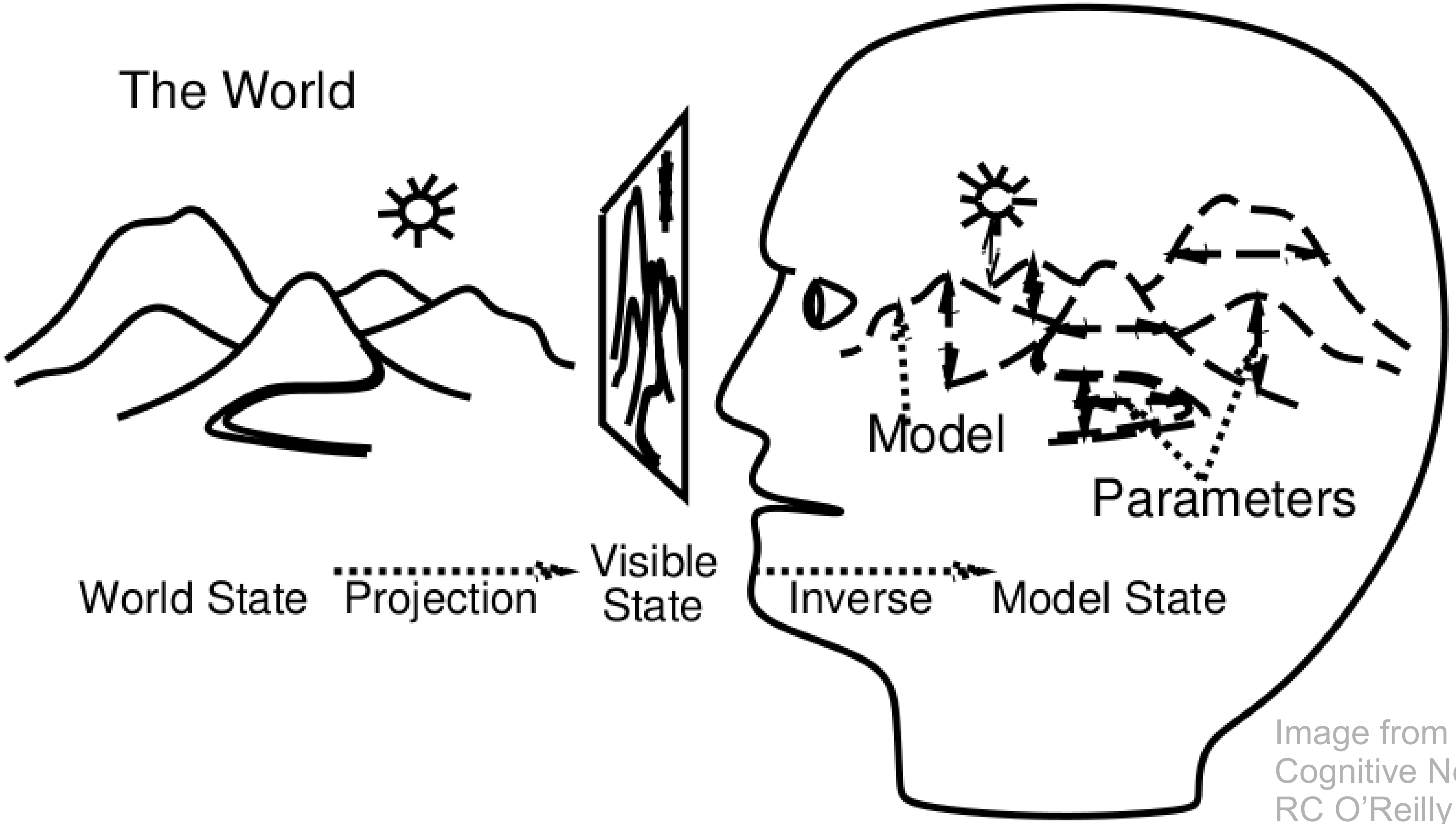


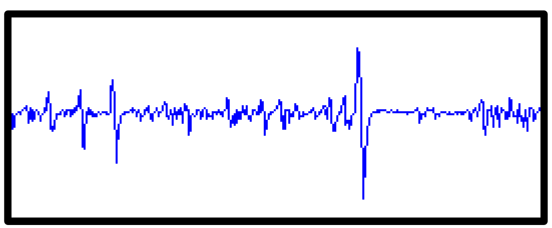
Image from "Computational Cognitive Neuroscience", RC O'Reilly and Y Munakata, MIT Press, 2000.

Introduction

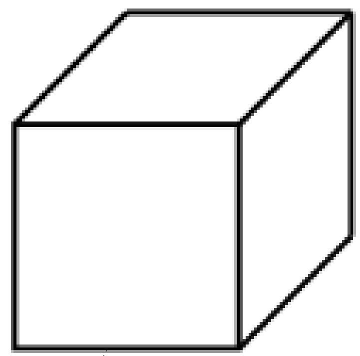
What are generative models used for?

Data analysis

Example 3



y



Recognition System

e.g., c=0,1,2,3

c

Answer should be probabilistic.

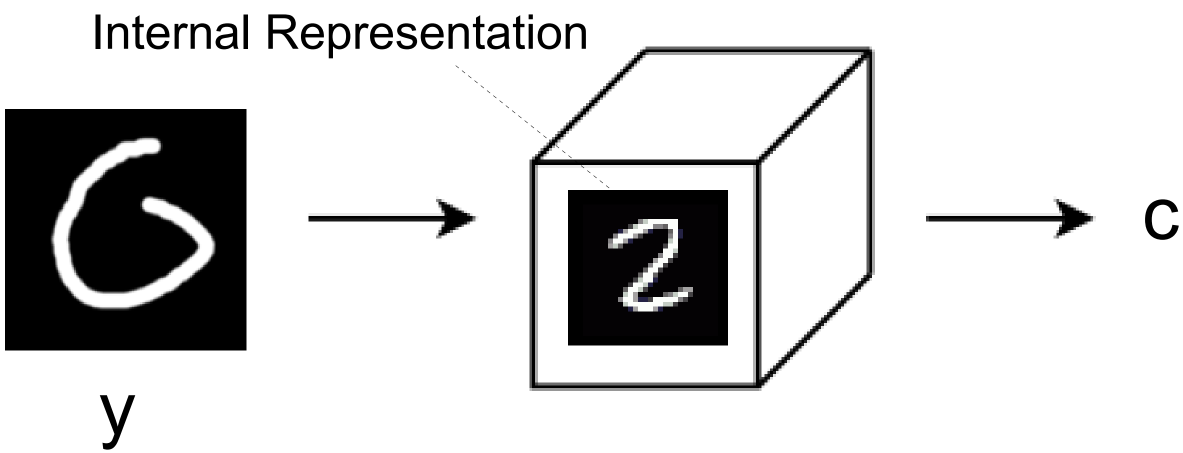
Posterior probability:

$$p(c|y)$$

Introduction - Learning

But how does our black-box generative model acquire the knowledge for internal representations?

It can learn it.



Generative models can learn from examples.

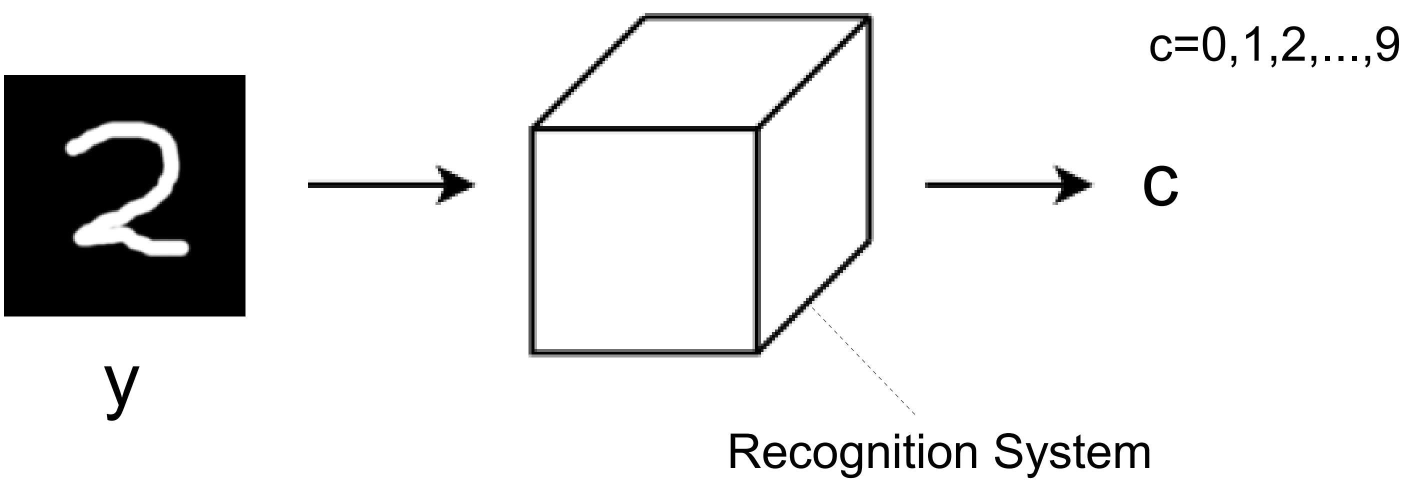
usually unsupervised

Introduction

What are generative models used for?

Inference – given an input a generative model allows to extract \`higher-level' knowledge

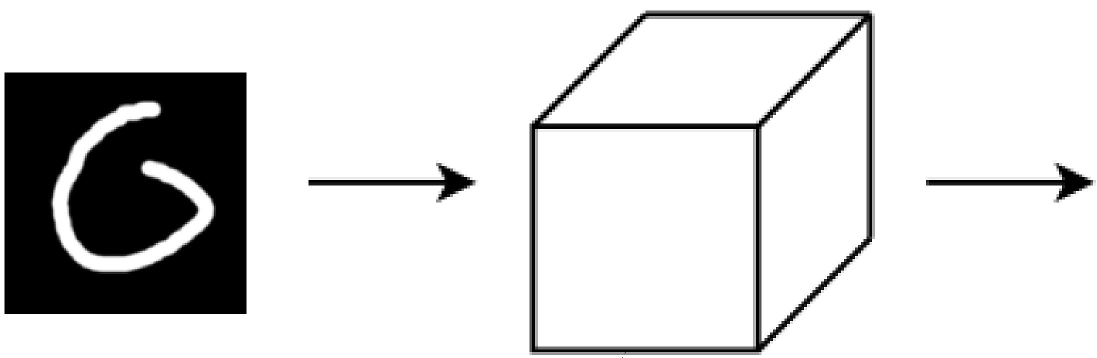
Learning – given a set of data points, a generative model can learn a data representation



Optimal Coding

There is an appealing theoretical result for generative models:

If the right model is used, knowledge extraction is <u>optimal</u>.



y

Recognition System

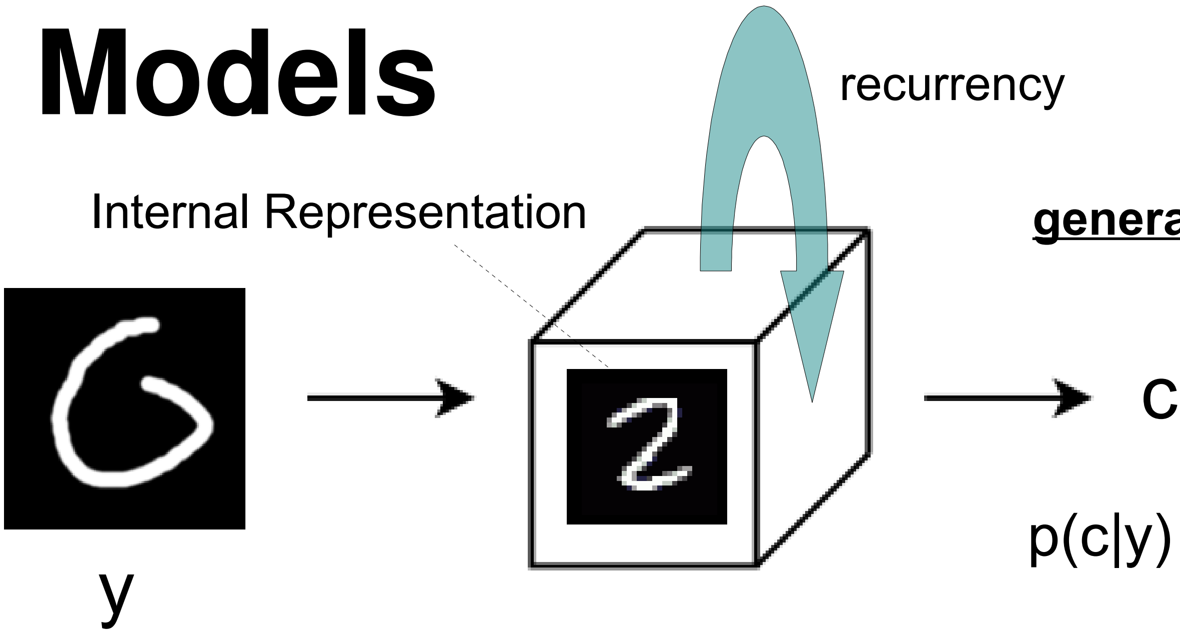
c=0,1,2,...,9

c=0 or c=6?

Probabilistic answer:

c=0	p=0.61...
c=1	p=0.0...01
c=2	p=0.0...01
...	
c=6	p=0.38...
c=7	p=0.0...01
...	

Generative vs. Discriminative Models

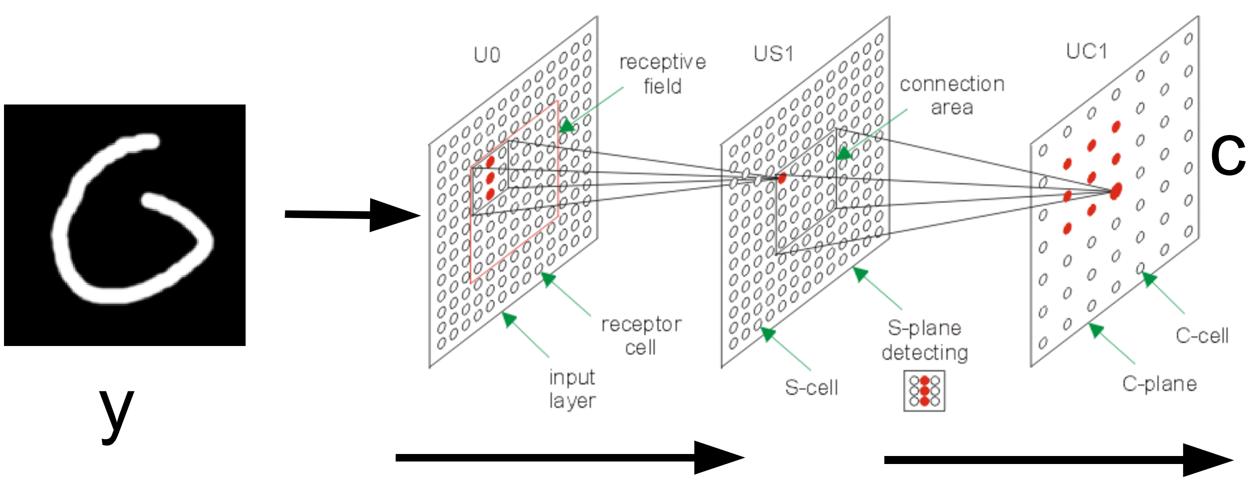


generative

Usual Features:

- internal representation (for inference and learning)
- recurrent processing
- probabilistic
- slow

Recognition.



- no or limited internal representation
- feed-forward
- often deterministic
- fast

Classification.

feed-forward processing

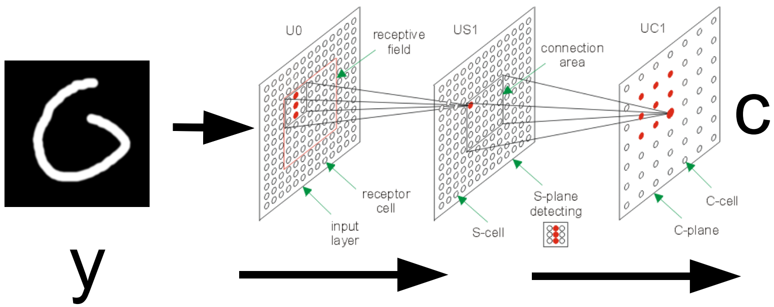
discriminative

Generative vs. Discriminative Models

There is currently a debate. The brain seems to provide evidence for both.

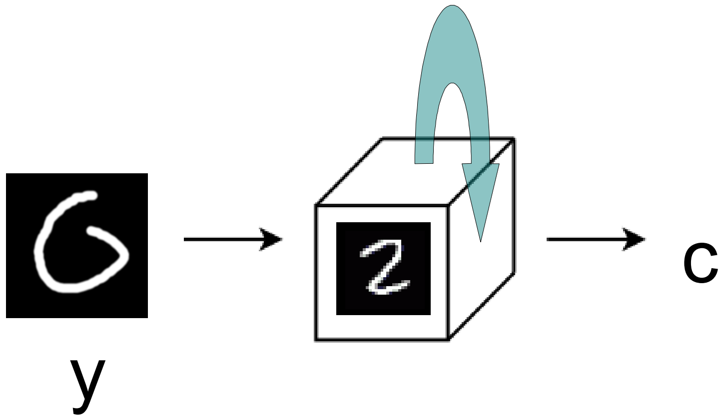
`Ultra Rapid' feed-forward sweep (e.g. S. Thorpe).

=> Early classification.



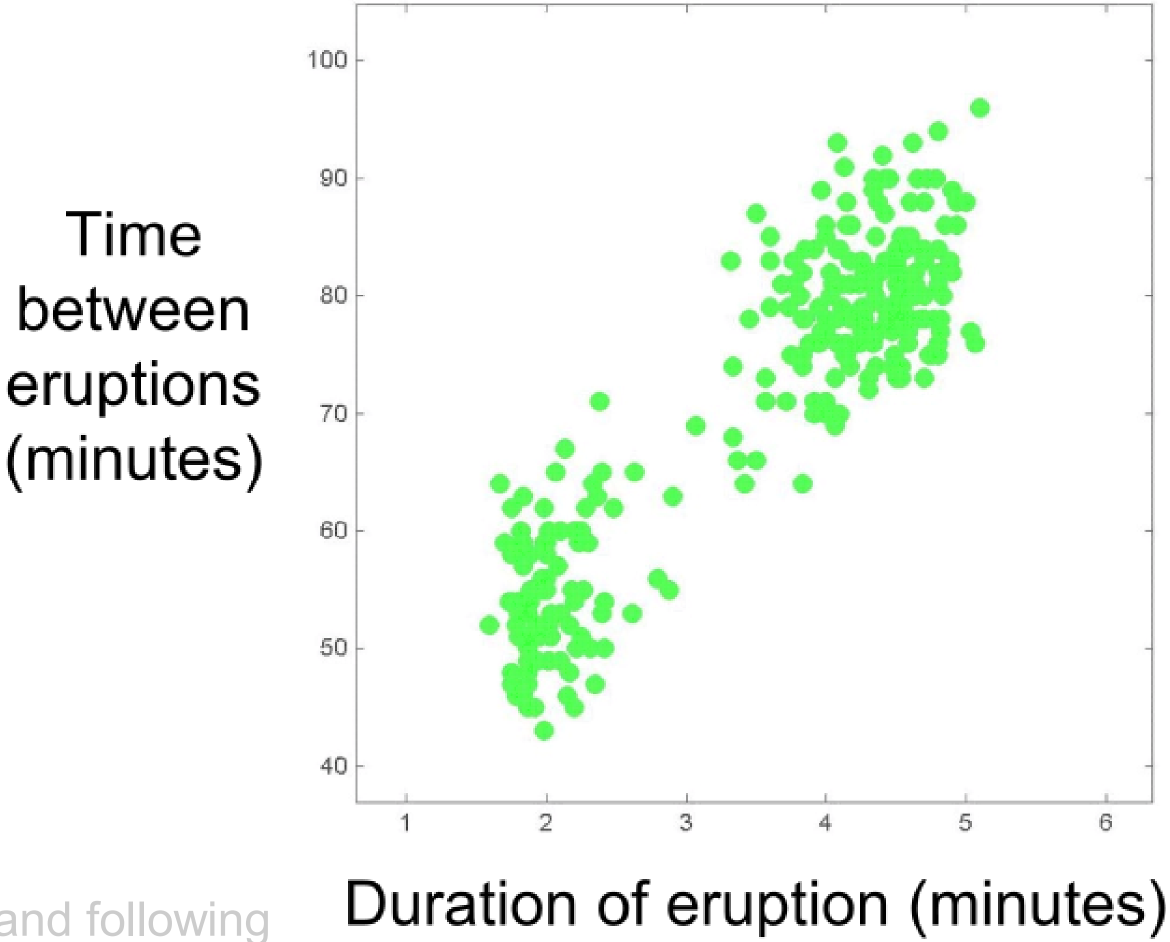
`Rapid' but slower recurrent processing.

=> Elaborate Recognition.



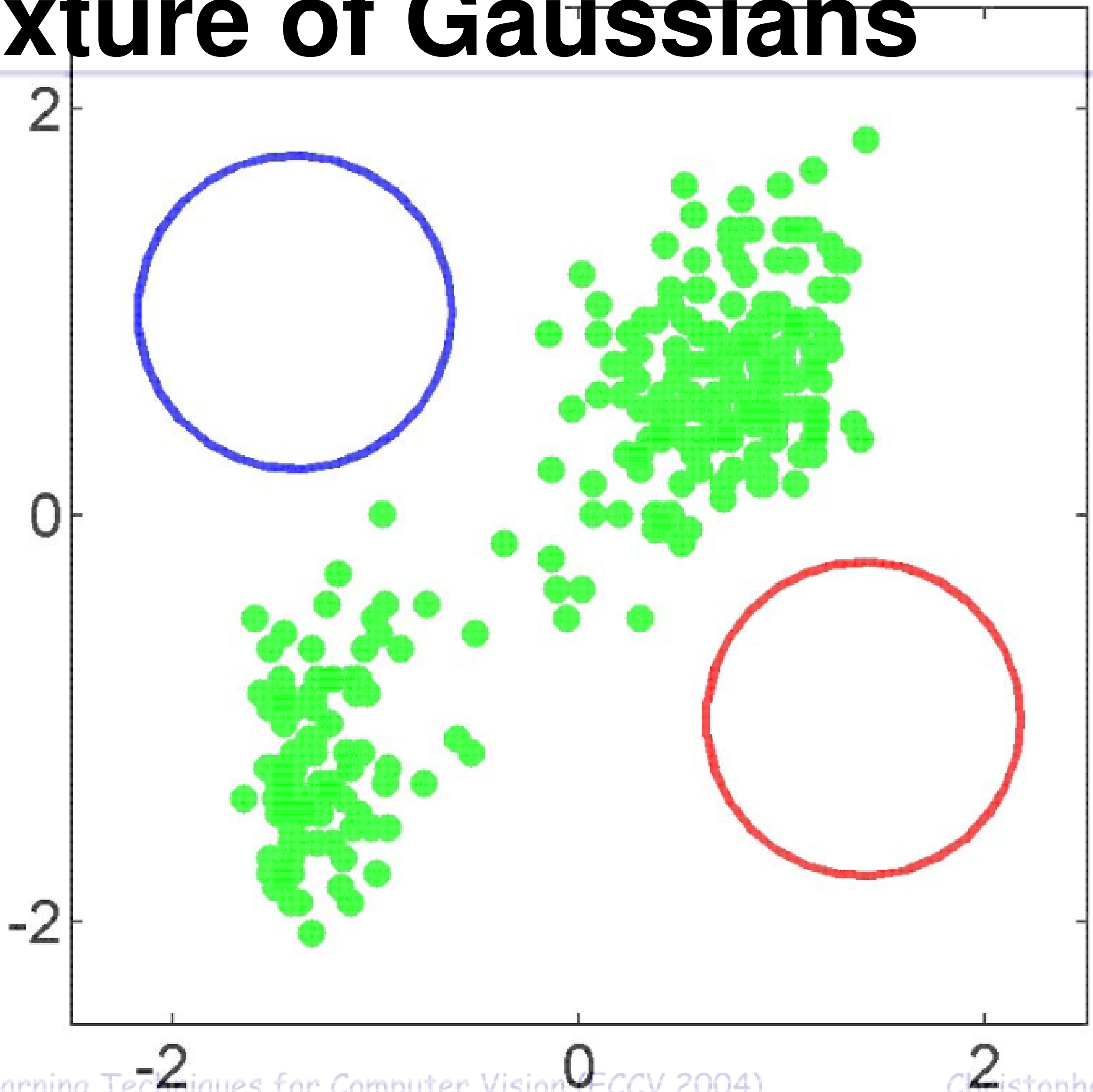
Classical Examples of Generative Models

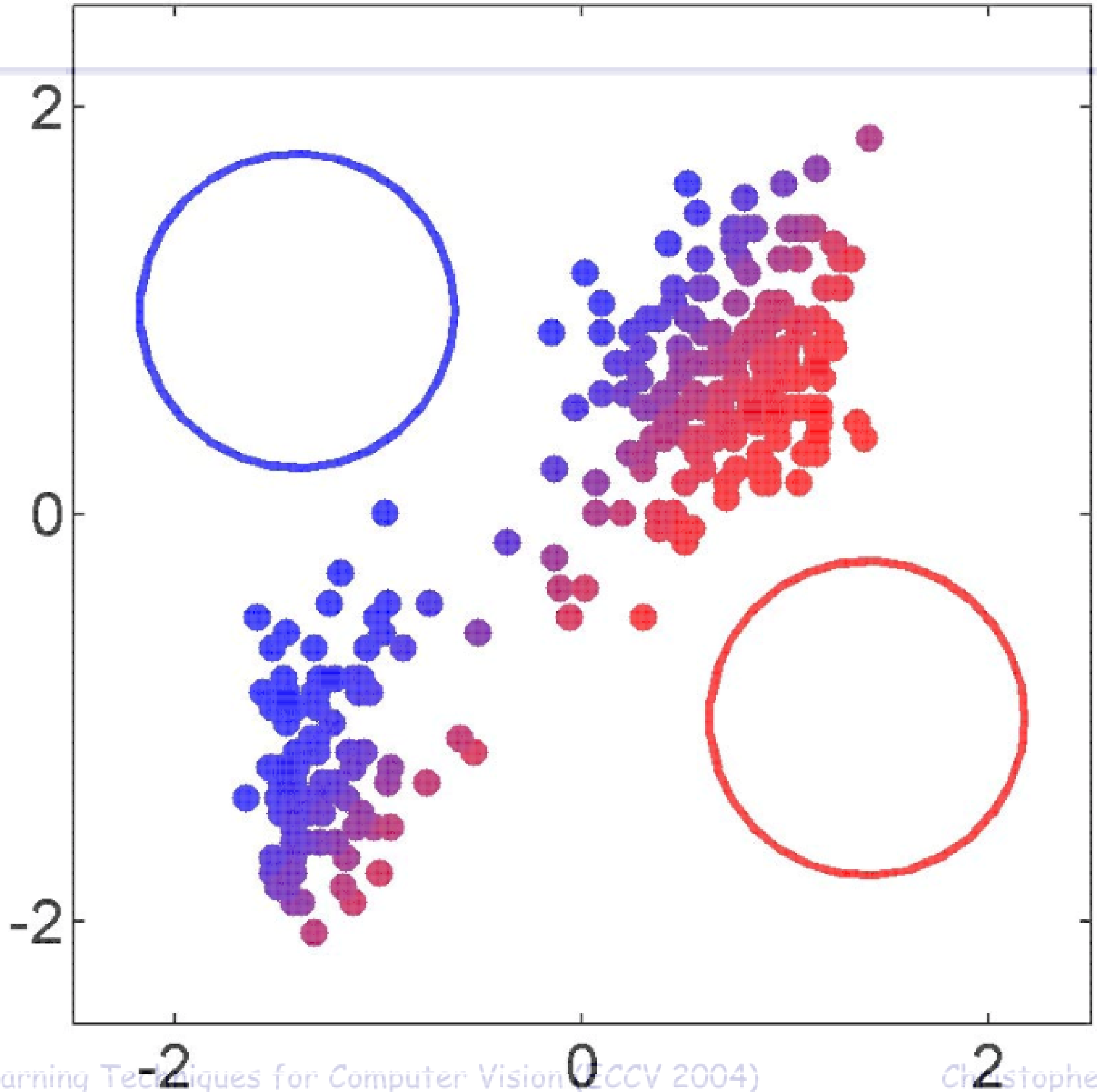
Old Faithful Data Set

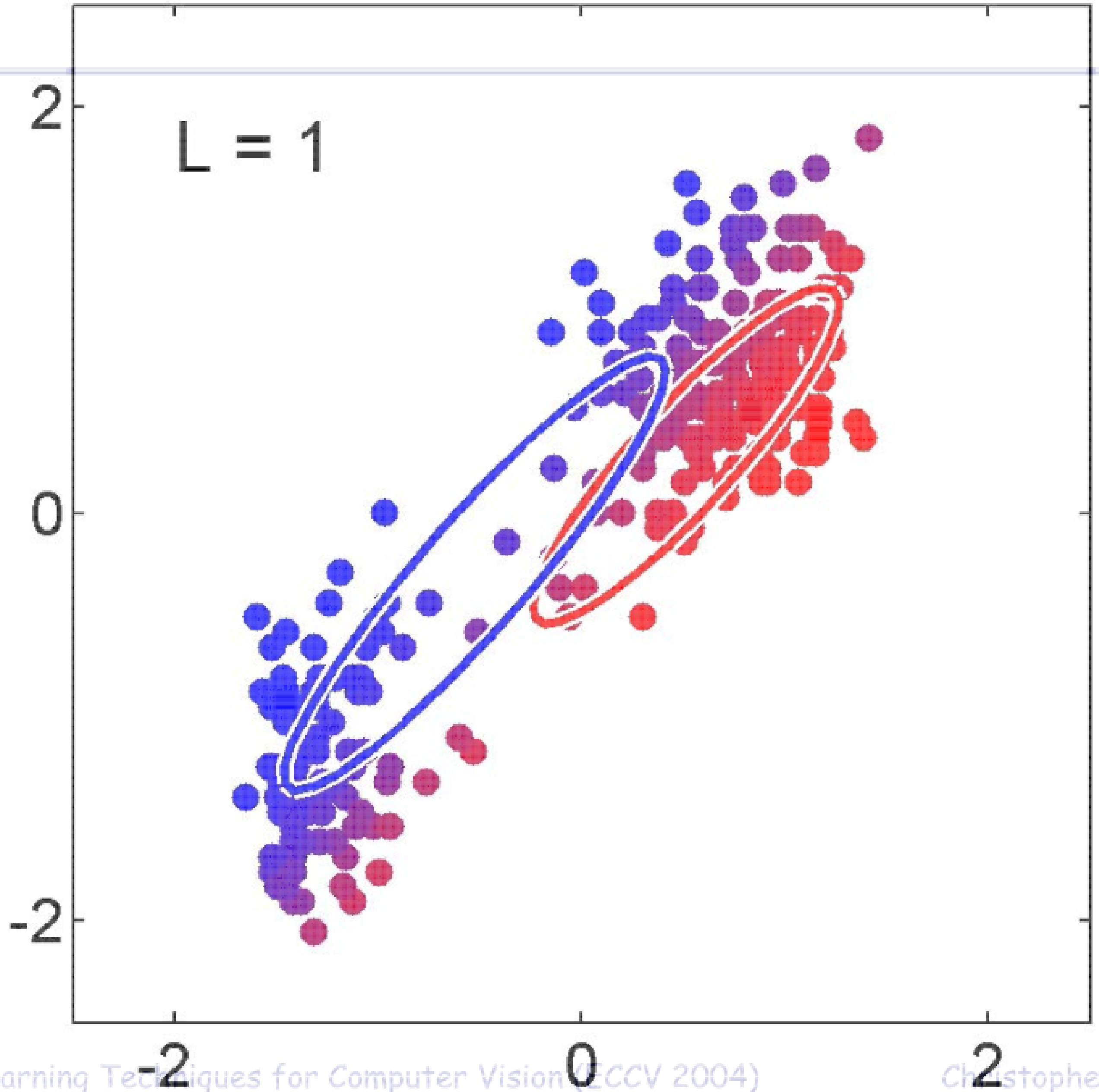


This and following slides are taken from: Machine Learning Techniques for Computer Vision (ECCV 2004) Christopher M. Bishop

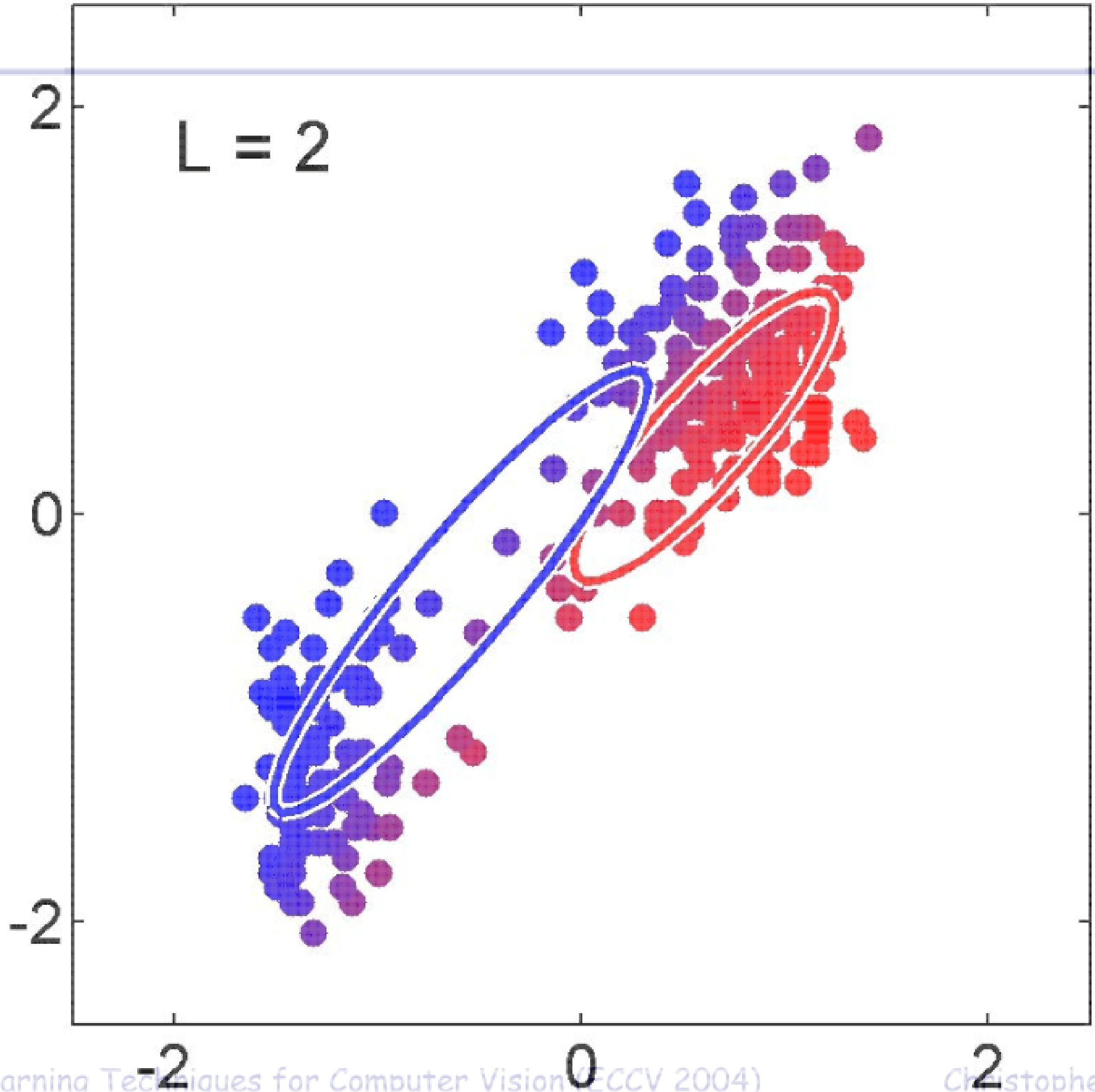
A) Mixture of Gaussians



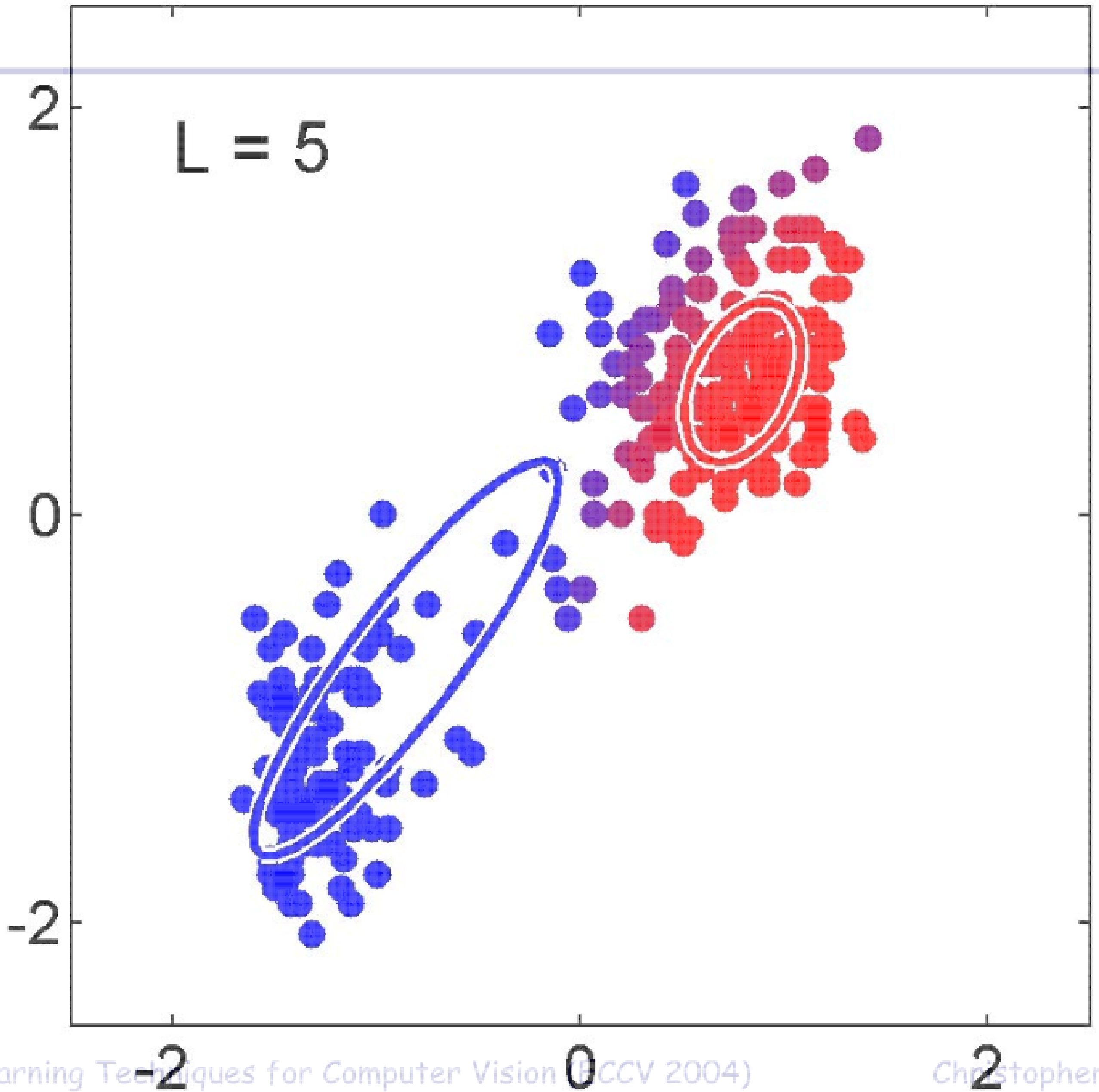

2
0
-2
-2
0
2
Machine Learning Techniques for Computer Vision (ECCV 2004)
Christopher M. Bishop



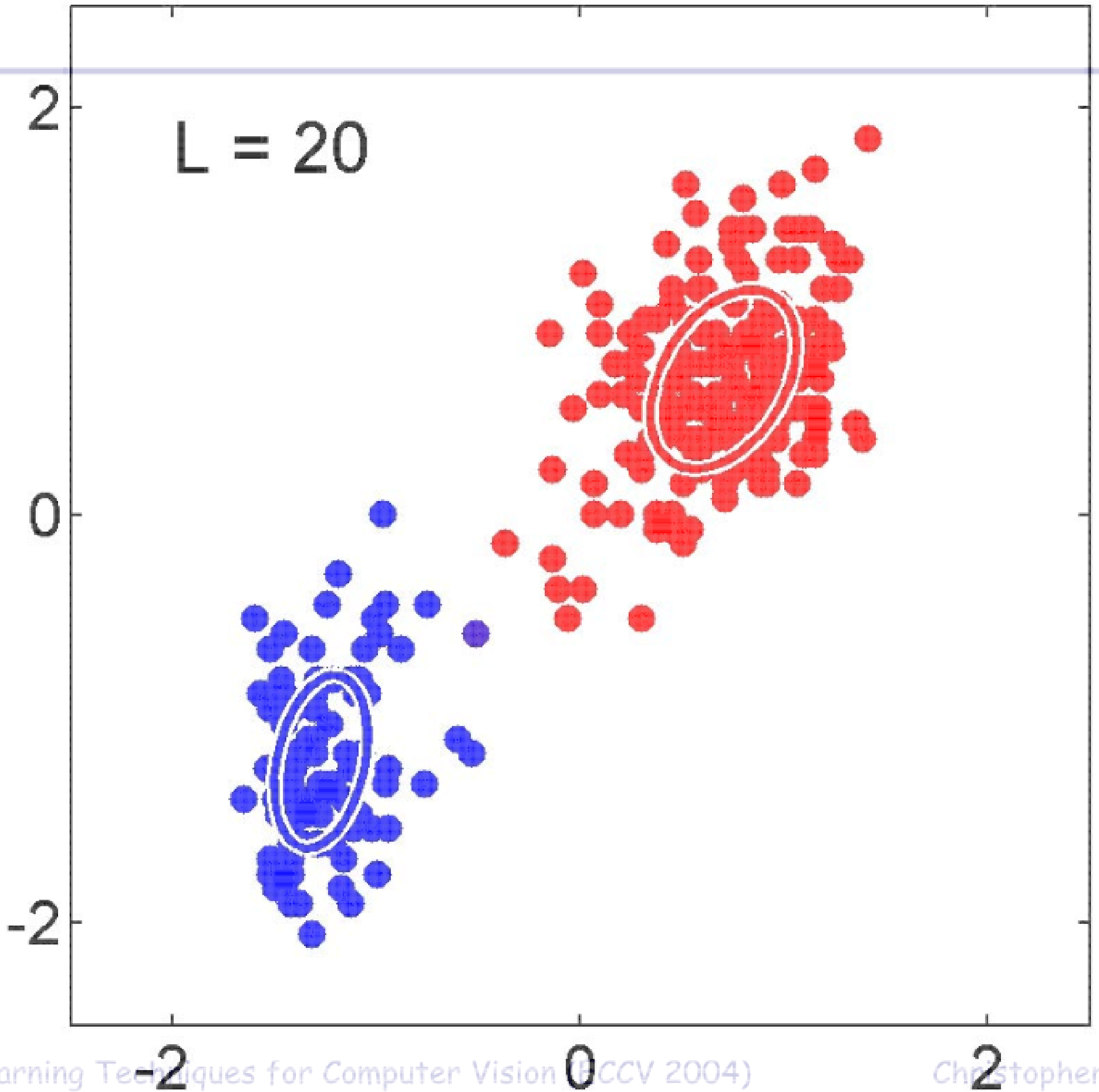
L = 1
2
0
-2
-2
0
2



L = 2
2
0
-2
-2
0
2



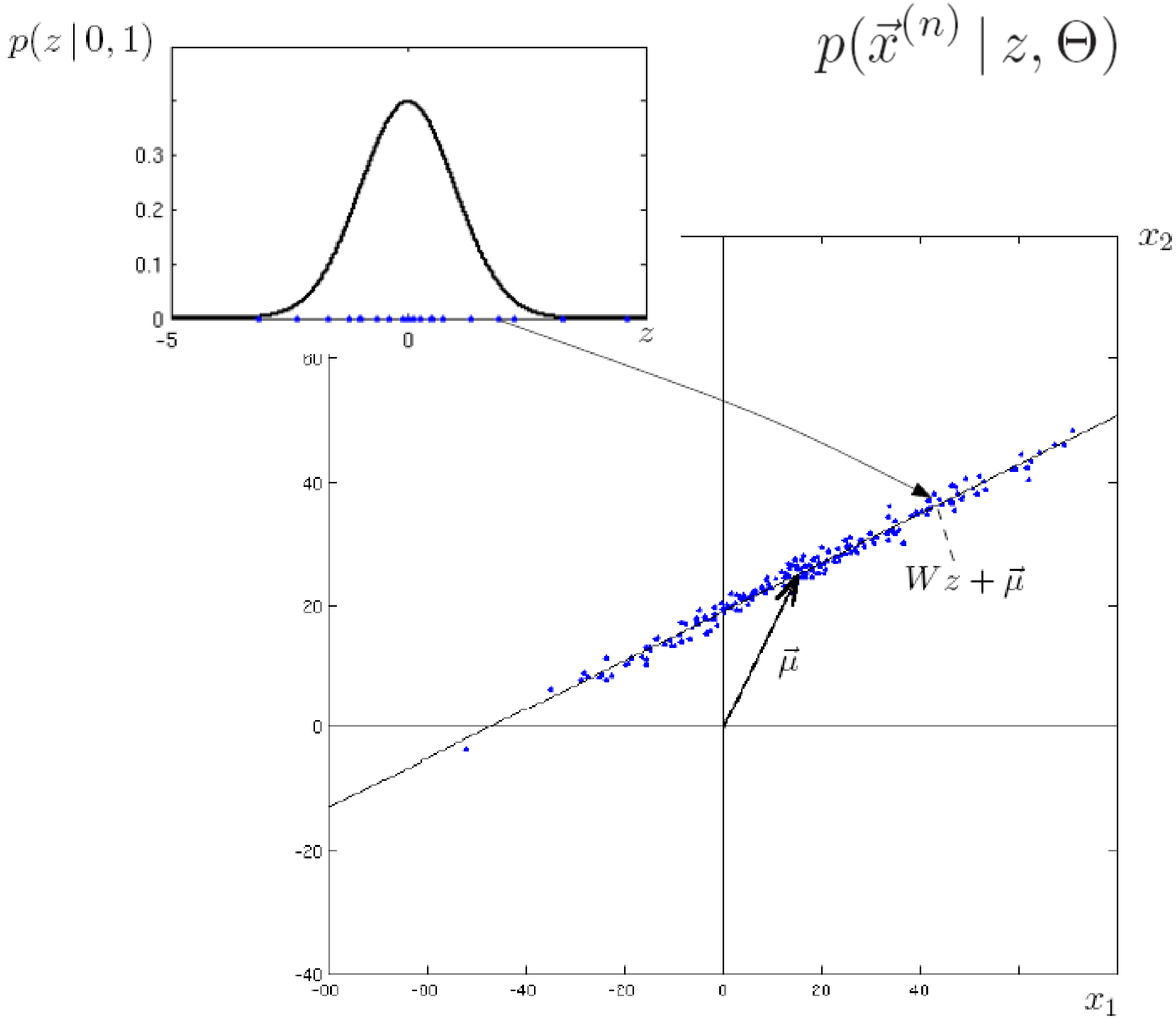
L = 5
2
0
-2
-2
0
2



-> also see matlab program for 1-dim, and blackboard

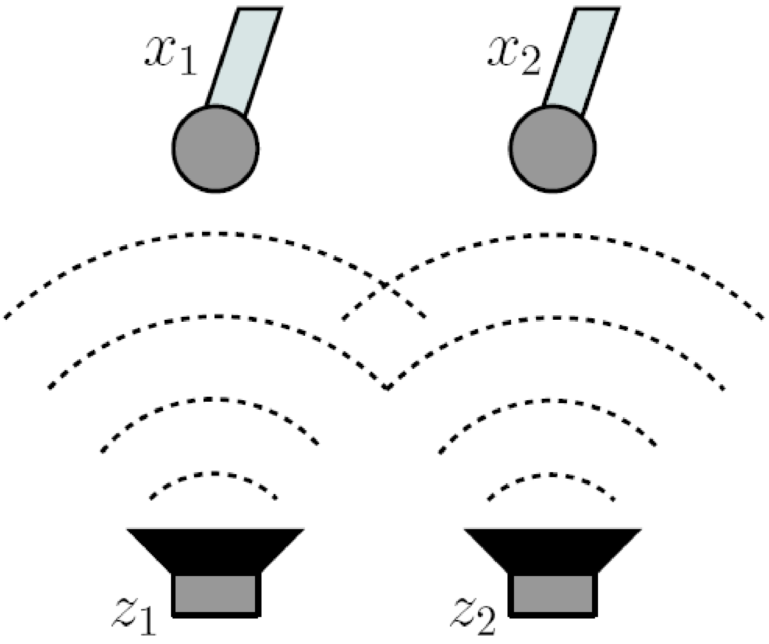
B) Principle Component Analysis

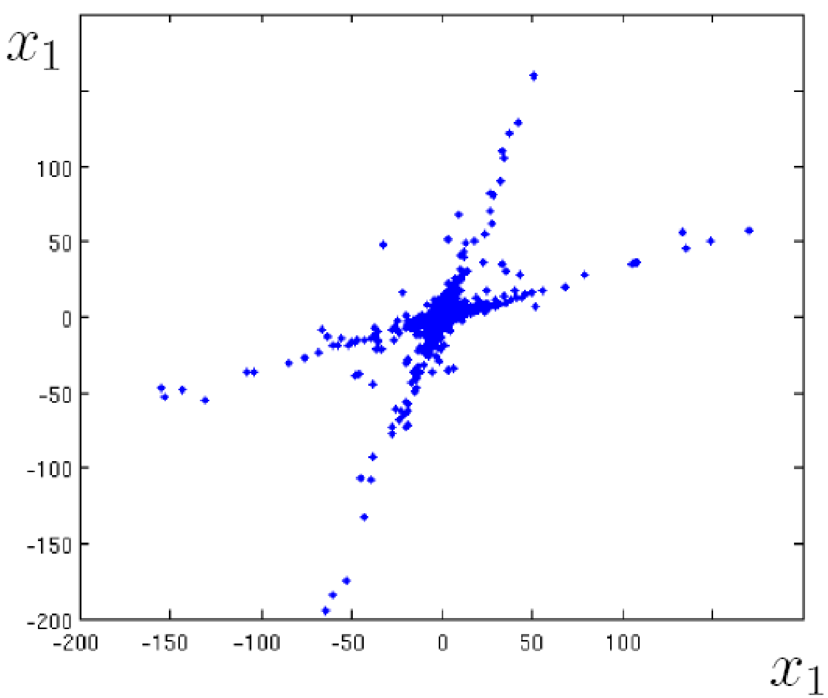
$$
\begin{aligned}
p(z \,|\, \Theta) &= \mathcal{N}(z; 0, 1) \\
p(\vec{x}^{(n)} \,|\, z, \Theta) &= \mathcal{N}(\vec{x}^{(n)}; W z + \vec{\mu}, \sigma^2 \mathbf{I})
\end{aligned}
$$



-> matlab program,
and blackboard

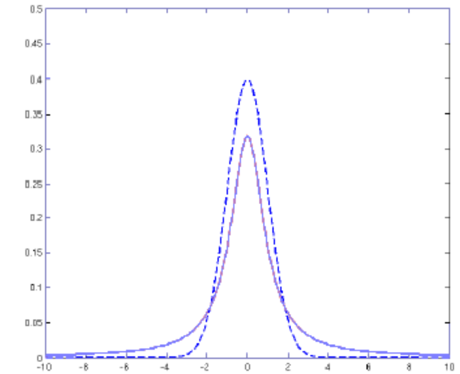
C) Sparse Coding / Independent Component Analysis





C) Sparse Coding / Independent Component Analysis

dotted = Gaussian
solid = Cauchy

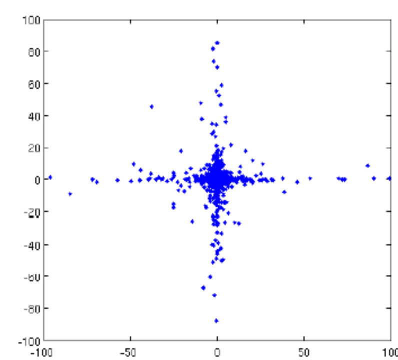


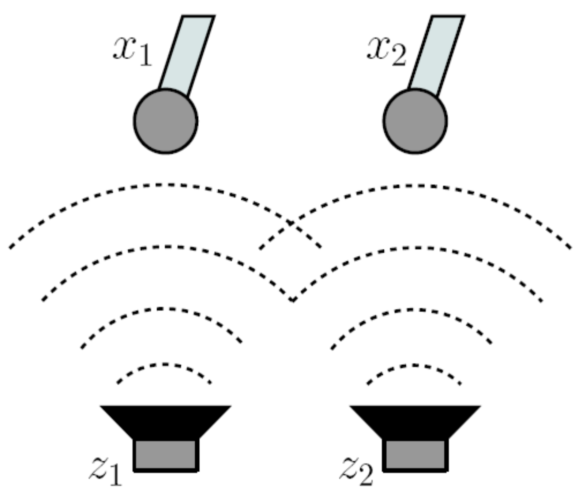
$$p(\vec{z}\,|\,\Theta) = \prod_{i=1}^{m} \mathcal{C}(z_i)\,, \quad \text{where} \quad \mathcal{C}(z_i) = \frac{1}{\pi\,(1+z_i^2)}$$

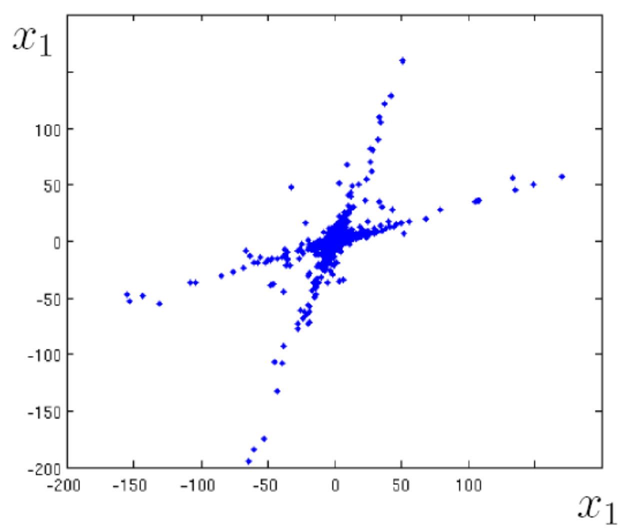
$$p(\vec{x}^{(n)}\,|\,\vec{z},\Theta) = \mathcal{N}(\vec{x}^{(n)}; W\vec{z}+\vec{\mu}, \sigma^2\,\mathbf{I})$$

sampling from prior

-> matlab program







linear projection + noise

C) Sparse Coding / Independent Component Analysis

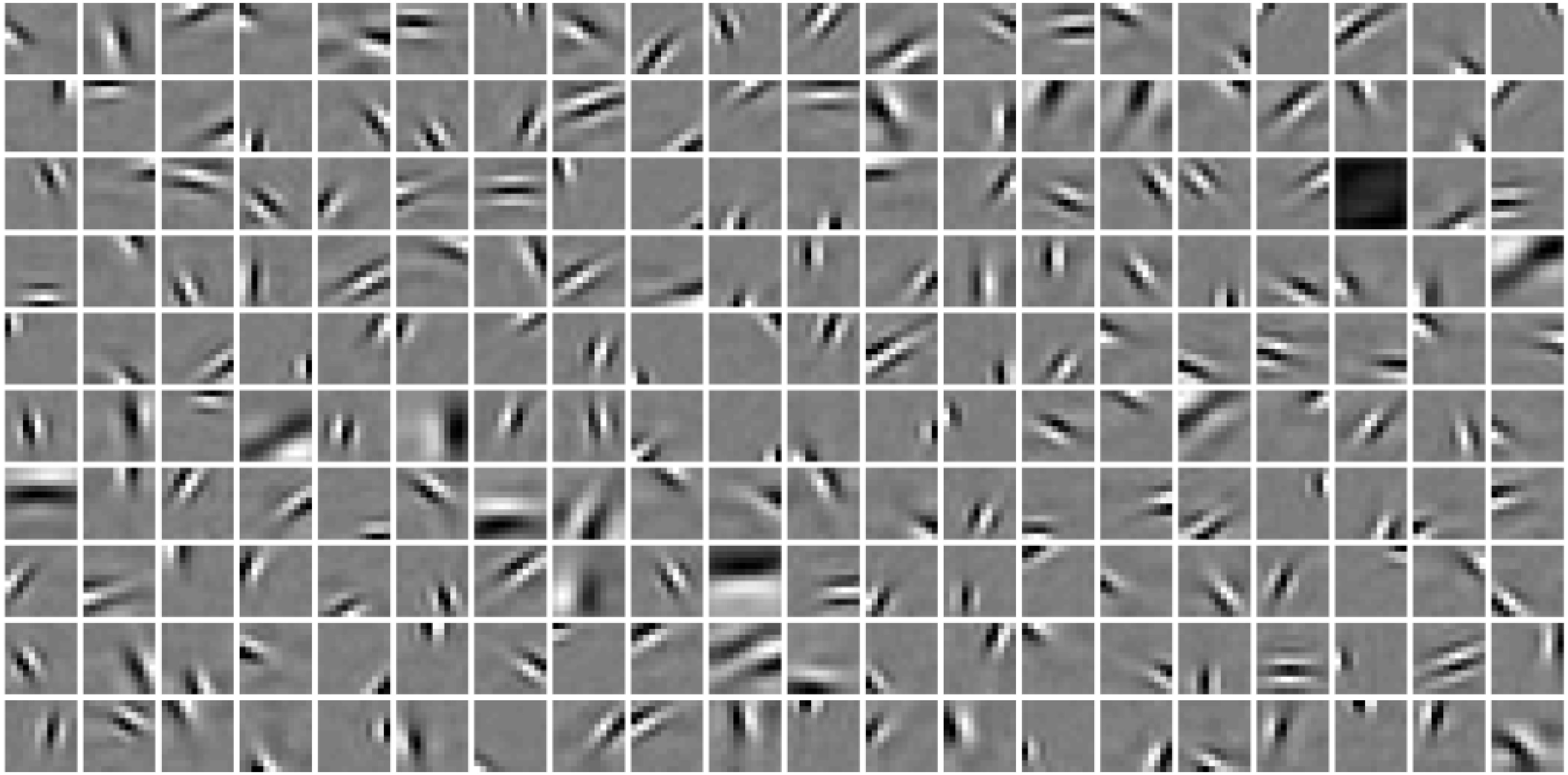


Figure 13.3. Basis functions learned from static natural images. Shown is a set of 200 basis functions which were adapted to 12 × 12 pixel image patches, according to equations (13.14) and (13.15). Initial conditions were completely random. The basis set is approximately 2×'s overcomplete, since the images occupy only about 3/4 of the dimensionality of the input space. (See Olshausen & Field, 1997, for simulation details.)

Discussion

- Generative models provide a common principled framework
- k-Means is a special form of a Mixture of Gaussians model
- ICA is a special form of Sparse Coding
- Generative models enable optimal coding
  But: learning often takes too long => approximations
- Generative models allow for the incorporation of ones beliefs
- The brain (or part of it) might be interpretable as a generative model
- Simple-cell receptive fields might be evidence for optimal coding
  But: Sparse Coding / ICA might be too simple

How people see the relation between generative models and neuroscience:

- generative models are elaborate functional models,
  they are the best way to approach many problems,
  but leave me alone with neuroscience

- generative models are a very good way to described the
  function of the brain or the function of a brain area,
  neuroscience is to study how they are implemented

- generative models are a great tool that allows to study
  how information can be processed,
  good inspiration for neuroscience

- generative models are a statistical / computer science tool,
  neuroscience is something different,
  the brain is best understood using other approaches

Further Reading

Pattern Recognition and Machine Learning
C. M. Bishop, ISBN: 978-0-387-31073-2, Springer, 2006.

Theoretical Neuroscience – Computational and Mathematical Modeling of Neural Systems
P. Dayan and L. F. Abbott, ISBN: 0-262-04199-5, MIT Press, 2001.

Information Theory, Inference, and Learning Algorithms
D. MacKay, ISBN-10: 0521642981, Cambridge University Press, 2003.

Computational Cognitive Neuroscience
RC O'Reilly and Y Munakata, ISBN-10: 0262650541, MIT Press, 2000.

... and many more

Thanks.