A Brief Introduction to

Generative Models

Jörg Lücke
FIAS, Goethe-Universität Frankfurt, Germany

FIAS Summer School
Frankfurt, August 2008
Contents

Introduction

Introductory Examples

Optimal Coding Hypothesis

Generative vs. Discriminative Models

Classical Examples:  - Mixture of Gaussians
                    - Probabilistic PCA
                    - Sparse Coding / ICA

Simple-Cell Receptive Fields

Discussion

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

```
\[ \begin{array}{c}
\text{Recognition System} \\
\end{array} \]
```

\[ y \rightarrow \text{cube} \rightarrow c=2 \]

\[ c=0,1,2,...,9 \]
Introduction

What are generative models used for?

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

Example 1

<table>
<thead>
<tr>
<th>c</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.61</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.38</td>
</tr>
<tr>
<td>7</td>
<td>0.01</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

[Image of a recognition system with inputs and outputs showing probabilities for different categories.]
Introduction

What are generative models used for?

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

Example 1

<table>
<thead>
<tr>
<th>y</th>
<th>Recognition System</th>
<th>c=0 or c=6?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>c=0,1,2,...,9</td>
</tr>
</tbody>
</table>

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

\[
p(c|y) = \frac{p(y|c) p(c)}{\sum_{c'} p(y|c') p(c')}
\]

Generative model + Bayes’ rule

\[ c=0,1,2,...,9 \]

\[ c=0 \text{ or } c=6? \]

\[ \text{posterior probability } p(c|y) \]
Introduction

What are generative models used for?

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

Example 2

Image taken from Bishop, ECCV ‘04

\[ p(c|y) \]
Introduction

What are generative models used for?

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

Example 2

\[ y \rightarrow \text{Recognition System} \rightarrow p(c|y) \rightarrow c \]

\[ c = \text{hills with street and sun, sandcastle with hedgehog, snake with …, synapse and transmitters …} \]
Introduction

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

\[ p(c|y) \]
Introduction

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

\[ p(c|y) \]
Introduction

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

Introduction

What are generative models used for?

Data analysis

Example 3

Recognition System

Answer should be probabilistic.

Posterior probability:

\[ p(c|y) \]

e.g., \( c=0,1,2,3 \)
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 optimal.

Probabilistic answer:

- $c=0$, $p=0.61\ldots$
- $c=1$, $p=0.0\ldots01$
- $c=2$, $p=0.0\ldots01$
- $\ldots$
- $c=6$, $p=0.38\ldots$
- $c=7$, $p=0.0\ldots01$
- $\ldots$
Generative vs. Discriminative Models

**Internal Representation**
- **Generative**
  - internal representation (for inference and learning)
  - recurrent processing
  - probabilistic
  - slow

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

**Recognition.**

**Classification.**

\[
p(c|y)
\]
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

Time between eruptions (minutes)

Duration of eruption (minutes)

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

$L = 2$
A) Mixture of Gaussians

$L = 5$
A) Mixture of Gaussians
-> also see matlab program for 1-dim, and blackboard
B) Principle Component Analysis

\[ p(z | \Theta) = \mathcal{N}(z; 0, 1) \]
\[ p(\vec{x}^{(n)} | z, \Theta) = \mathcal{N}(\vec{x}^{(n)}; Wz + \bar{\mu}, \sigma^2 I) \]

-> matlab program, and blackboard
C) Sparse Coding / Independent Component Analysis
C) Sparse Coding / Independent Component Analysis

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

\[ p(\tilde{x}^{(n)} | \tilde{z}, \Theta) = \mathcal{N}(\tilde{x}^{(n)}; W\tilde{z} + \bar{\mu}, \sigma^2 I) \]

-> matlab program

dotted = Gaussian
solid = Cauchy

sampling from prior

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 describe 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*

*Theoretical Neuroscience – Computational and Mathematical Modeling of Neural Systems*

*Information Theory, Inference, and Learning Algorithms*

*Computational Cognitive Neuroscience*

... and many more
Thanks.