Unsupervised Learning

for Features and Representations

using Probabilistic, Bayesian and Deep Architectures

Yee Whye Teh & Nicolas Heess

Acknowledgement:
Andrew Ng, Kai Yu, Honglak Lee, Marc’Aurelio Ranzato, Yoshua Bengio
Unsupervised Learning
Major Types of Learning

• Supervised learning

• Reinforcement learning

• Unsupervised learning
Architecture for Machine Perception

Input

Learning algorithm

Andrew Ng
Architecture for Machine Perception

Input

pixel 1

pixel 2

Learning algorithm

Andrew Ng
Architecture for Machine Perception

Input

Input space

Learning algorithm

Motorbikes
“Non”-Motorbikes

Andrew Ng
Architecture for Machine Perception

Input

Feature representation

Learning algorithm

handle
wheel

Andrew Ng
Architecture for Machine Perception

Input
- handle
- wheel

Input space

Feature representation

Motorbikes
“Non”-Motorbikes

Learning algorithm

Andrew Ng
Architecture for Machine Perception

Input

Input space

Feature space

Feature representation

Learning algorithm

- Motorbikes
- “Non”-Motorbikes

Andrew Ng
Architecture for Machine Perception

Input

- deep architectures

Feature representation

- Bayesian nonparametrics

Learning algorithm
Probabilistic Models in Machine Learning
Probabilistic Machine Learning

- Machine Learning is all about data.
  - Stochastic, chaotic and/or complex process
  - Noisily observed
  - Partially observed
- **Probability theory** is a rich language to express these uncertainties.
  - **Probabilistic models**
- Graphical tool to visualize complex models for complex problems.
- Complex models can be built from simpler parts (task decomposition!).
- Computational tools to derive algorithmic solutions.
- Separation of modelling questions from algorithmic questions.
Probabilistic Modelling

• Data: $x_1, x_2, \ldots, x_n$.

• Latent variables: $y_1, y_2, \ldots, y_n$.

• Parameter: $\theta$.

• A probabilistic model is a parametrized joint distribution over variables.

$$P(x_1, \ldots, x_n, y_1, \ldots, y_n | \theta)$$

• Typically interpreted as a generative model of data.

• Inference, of latent variables given observed data:

$$P(y_1, \ldots, y_n | x_1, \ldots, x_n, \theta) = \frac{P(x_1, \ldots, x_n, y_1, \ldots, y_n | \theta)}{P(x_1, \ldots, x_n | \theta)}$$
Probabilistic Modelling

• Learning, typically by maximum likelihood:
\[ \theta_{\text{ML}} = \arg\max_{\theta} P(x_1, \ldots, x_n | \theta) \]

• Prediction:
\[ P(x_{n+1}, y_{n+1} | x_1, \ldots, x_n, \theta) \]

• Classification:
\[ \arg\max_{c} P(x_{n+1} | \theta^c) \]

• Visualization, interpretation, summarization.

• Standard algorithms: EM, junction tree, variational inference, MCMC...
Graphical Models

- **Nodes** = variables
- **Edges** = dependencies
- Lack of edges = conditional independencies

### Example

- **Earthquake**
- **Burglar**
- **Alarm**
Model-based Clustering

- Model for data from heterogeneous unknown sources.
- Each cluster (source) modelled using a parametric model (e.g. Gaussian).
- Data item $i$:
  \[ z_i | \pi \sim \text{Discrete}(\pi) \]
  \[ x_i | z_i, \theta^*_k \sim F(\theta^*_{z_i}) \]
- Mixing proportions:
  \[ \pi = (\pi_1, \ldots, \pi_K) | \alpha \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) \]
- Cluster $k$:
  \[ \theta^*_k | H \sim H \]
Hidden Markov Models

- Popular model for time series data.
- Unobserved dynamics modelled using a Markov model
- Observations modelled as independent conditioned on current state.
Collaborative Filtering

- Data: for each user $i$ ratings $R_{ij}$ for a subset of products $j$.
- Problem: predict how much users would like products that they haven’t seen.

$$R_{ij} | \xi_i, \eta_j \sim \mathcal{N}(\xi_i^T \eta_j, \sigma^2)$$
Restricted Boltzmann Machines

- Undirected graphical model used as a layer of deep belief networks.
- Data: binary vectors (pixels in images, spectrogram).
- Model joint distribution over binary vector using vector of latent variables.

\[
p(x, y) = \frac{1}{Z} e^{\sum_{i,j} w_{ij} x_i y_j + \sum_i b_i x_i + \sum_j c_j y_j}
\]

- Conditional distribution of latent variables:

\[
p(y|x) = \prod_j \frac{1}{Z_j} e^{\sum_i w_{ij} x_i y_j + c_j y_j}
\]

- Learning via maximum likelihood:

\[
\frac{\partial \log p(x)}{\partial w_{ij}} = E_{p(y|x)}[x_i y_j] - E_{p(x,y)}[x_i y_j]
\]
Architecture for Machine Perception

Input

- deep architectures

Feature representation

- Bayesian nonparametrics

Learning algorithm
Feature Representations in Computer Perception
How is computer perception done?

Object detection

Image → Low-level vision features → Recognition
How is computer perception done?

Object detection

Image → Low-level vision features → Recognition

Audio classification

Audio → Low-level audio features → Speaker identification
How is computer perception done?

Object detection

Image → Low-level vision features → Recognition

Audio classification

Audio → Low-level audio features → Speaker identification

Helicopter control

Helicopter → Low-level state features → Action
Learning representations

Sensor → Feature Representation → Learning algorithm
Learning representations

Sensor -> Feature Representation -> Learning algorithm
Computer vision features

SIFT

Spin image

HoG

RIFT

Textons

GLOH
Audio features

Spectrogram

MFCC

Flux

ZCR

Rolloff
Problems of hand-tuned features:
1. Needs expert knowledge
2. Time-consuming and expensive
3. Does not generalize to other domains
Problems of hand-tuned features:
1. Needs expert knowledge
2. Time-consuming and expensive
3. Does not generalize to other domains

Key question: Can we automatically learn a good feature representation?
The goal of Unsupervised Feature Learning

- SIFT
- Spin image
- HoG
- RIFT
- Textons
- GLOH
- Spectrogram
- MFCC
- Flux
- ZCR
- Rolloff
The goal of Unsupervised Feature Learning
The goal of Unsupervised Feature Learning
The goal of Unsupervised Feature Learning

Unlabeled images

Learning algorithm

Feature representation
Bayesian Nonparametrics
Bayesian Modelling

- Prior distribution:
  \[ P(\theta) \]

- Posterior distribution (both inference and learning):
  \[ P(y_1, \ldots, y_n, \theta|x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n, y_1, \ldots, y_n|\theta)P(\theta)}{P(x_1, \ldots, x_n)} \]

- Prediction:
  \[ P(x_{n+1}|x_1, \ldots, x_n) = \int P(x_{n+1}|\theta)P(\theta|x_1, \ldots, x_n)d\theta \]

- Classification:
  \[ P(x_{n+1}|x_1^c, \ldots, x_n^c) = \int P(x_{n+1}|\theta^c)P(\theta^c|x_1^c, \ldots, x_n^c)d\theta^c \]
Bayesian Nonparametrics

• What is a nonparametric model?
  • A really large parametric model;
  • A parametric model where the number of parameters increases with data;
  • A family of distributions that is dense in some large space relevant to the problem at hand.
Large Function Spaces

- Large function spaces.
- More straightforward to infer the infinite-dimensional objects themselves.
Bayesian Nonparametrics
Bayesian Nonparametrics
Bayesian Nonparametrics

\[ f^* \]
Bayesian Nonparametrics
Bayesian Nonparametrics

\[ f^* \]
Bayesian Nonparametrics

- Models with large coverage in hypothesis space, and with support for rich structural prior knowledge.
Model Selection and Averaging

• Model selection/averaging typically very expensive computationally.
• Used to prevent overfitting and underfitting.
• But a well-specified Bayesian model should not overfit anyway.
• By using a very large Bayesian model or one that grows with amount of data, we will not underfit either.
  • Bayesian nonparametric models.
Structural Learning

- Learning structures.
- Bayesian prior over combinatorial structures.
- Nonparametric priors sometimes end up simpler than parametric priors.

[Adams et al 2010, Blundell et al 2010]
Novel and Useful Properties

• Many interesting Bayesian nonparametric models with interesting and useful properties:
  • Projectivity, exchangeability.
  • Zipf, Heap and other power laws (Pitman-Yor, 3-parameter IBP).
  • Flexible ways of building complex models (Hierarchical nonparametric models, dependent Dirichlet processes).
Are Nonparametric Models Nonparametric?

- Nonparametric just means *not parametric*: *cannot be described by a fixed set of parameters*.
  - Nonparametric models still have parameters, they just have an infinite number of them.

- No free lunch: *cannot learn from data unless you make assumptions*.
  - Nonparametric models still make modelling assumptions, they are just less constrained than the typical parametric models.

- Models can be nonparametric in one sense and parametric in another: *semiparametric* models.
## Classes of Bayesian Nonparametric Models

<table>
<thead>
<tr>
<th>Problem</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering</td>
<td>Dirichlet process</td>
</tr>
<tr>
<td>Density estimation</td>
<td>Dirichlet process mixture model</td>
</tr>
<tr>
<td>Hierarchical clustering, trees</td>
<td>coagulation and fragmentation processes</td>
</tr>
<tr>
<td>Regression, classification</td>
<td>Gaussian process</td>
</tr>
<tr>
<td>Survival analysis</td>
<td>Beta process</td>
</tr>
<tr>
<td>Distributed representations</td>
<td>Indian buffet process</td>
</tr>
<tr>
<td>Power laws</td>
<td>Pitman-Yor process</td>
</tr>
<tr>
<td>Time series</td>
<td>Infinite hidden Markov model</td>
</tr>
<tr>
<td>Sequence prediction, compression</td>
<td>Sequence memoizer</td>
</tr>
</tbody>
</table>
Gaussian Process

- Prior over space of functions $f : X \rightarrow R$.
- Joint distribution over an uncountable set of random variables $\{f(x)\}_{x \in X}$.
- Defined as the (unique) distribution such that for any $x_1, \ldots, x_n$,
  \[
  \begin{bmatrix}
  f(x_1) \\
  \vdots \\
  f(x_n)
  \end{bmatrix}
  \sim \mathcal{N}
  \left(
  \begin{bmatrix}
  m(x_1) \\
  \vdots \\
  m(x_n)
  \end{bmatrix},
  \begin{bmatrix}
  c(x_1, x_1) & \cdots & c(x_1, x_n) \\
  \vdots & \ddots & \vdots \\
  c(x_n, x_1) & \cdots & c(x_n, x_n)
  \end{bmatrix}
  \right)
  \]
- Structural prior: smoothness of function as described by covariance kernel.
- Asymptotic properties depend on smoothness of $c$ and $f^*$. 
Dirichlet Process

• Prior over space of measures $\mu$ over probability space $X$.
• Joint distribution over an uncountable set of random variables $\{\mu(A)\}_{A \in \Sigma}$.
• Defined as the (unique) distribution such that for any partition $A_1, \ldots, A_n$,
  $$(\mu(A_1), \ldots, \mu(A_n)) \sim \text{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_n))$$
Dirichlet Process

\[(\mu(A_1), \ldots, \mu(A_n)) \sim \text{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_n))\]

- Large support over space of measures, but probability one on discrete \(\mu\).
- For density estimation, clustering, can convolve with smooth kernel:
  \[\nu(A) = \int c(x, A) \mu(dx)\]
- Structural prior: smoothness of convolution kernel and tail behaviour.
- Asymptotic properties depend on prior and true density.
Sequence Models

• Probabilistic models for discrete sequence data.
• Particularly, of words and characters, e.g.
  south, parks, road
  s, o, u, t, h, _, p, a, r, k, s, _, r, o, a, d

• High order Markov models of such discrete sequence:
  \[ P(\text{sentence}) = \prod_i P(\text{word}_i | \text{word}_{i-N+1} \ldots \text{word}_{i-1}) \]

• Non-Markov models:
  \[ P(\text{sentence}) = \prod_i P(\text{word}_i | \text{word}_1 \ldots \text{word}_{i-1}) \]
Modelling Transition Probabilities

- Parametrize the transition probabilities of model:
  \[ P(\text{word}_i = w | \text{word}_{i-N+1}^{i-1} = u) = G_u(w) \]
  \[ G_u = [G_u(w)]_{w \in \text{vocabulary}} \]

  - \( G_u \) is transition probability vector in context \( u \).

  - Structural prior: similar contexts have similar predictions.

  - Context tree.
Power-law Prior using Pitman-Yor Processes

- Chinese restaurant process description of Pitman-Yor processes:

\[
P(\text{sit at table } c) = \frac{n_c - d}{\alpha + \sum_{c \in q} n_c}
\]

\[
P(\text{sit at new table}) = \frac{\alpha + d|q|}{\alpha + \sum_{c \in q} n_c}
\]

- Rich-gets-richer: larger tables grow at faster pace.
- With more occupied tables, chance of even more tables becomes higher.
- Tables with small occupancy numbers tend to have lower chance of getting new customers.
Power-law of English Word Frequencies

![Graph showing the power-law distribution of word frequencies in English text compared to Pitman-Yor and Dirichlet models.](image)
Tutorials and Reviews

- Mike Jordan’s tutorial at NIPS 2005.
- Zoubin Ghahramani’s tutorial at UAI 2005.
- Peter Orbanz’ tutorial at MLSS 2009 (videolectures).
- My own tutorials at MLSS 2007, 2009 (videolectures), 2011 (Singapore, France) and elsewhere.
- Introduction to Dirichlet process [Teh 2010], nonparametric Bayes [Orbanz & Teh 2010, Gershman & Blei 2011], hierarchical Bayesian nonparametric models [Teh & Jordan 2010].
- Bayesian nonparametrics book [Hjort et al 2010].
WP1 Aims

- Learn features and representations from sensory and experiential data.

- Closing the “perception-action” cycle.

Questions to you:
- What would you like to see coming out from WP1 to benefit your WPs?
- What control/planning/reinforcement learning domains will benefit from perceptually learned representations?