Learning with Hidden Variables and RBMs

Ankur Gupta

University of British Columbia

August, 2015
On Monday we looked at learning parameters of a UGM,

- Vancouver rain example, $x$: rain/no-rain for each month
- Modeled the energy using a log-linear model: $E(x) = w^T F(x)$
- NLL: $f(w) = -\frac{1}{N} \sum_t \log(p(x^{(t)}|w)) = -w^T F(D) + \log(Z(w))$
- The objective function is convex
On Monday we looked at learning parameters of a UGM,

- Vancouver rain example, $x$: rain/no-rain for each month
- Modeled the energy using a log-linear model: $E(x) = w^T F(x)$
- NLL: $f(w) = -\frac{1}{N} \sum_t \log(p(x^{(t)}|w)) = -w^T F(D) + \log(Z(w))$
- The objective function is convex

Today we focus on learning parameters to model $p(x, h)$

- where only $x$ is observed and
- $h$ is not observed (or hidden) in the training examples
- e.g., if the rain entry for a few days each month is missing, how to still use the data for learning
Learning with Hidden Variables

We can obtain $p(x)$ by summing over all values of $h$: 

$$p(x) = \sum_h p(x, h) = \sum_h \exp(-E(x, h)) / Z_h(x)$$
Learning with Hidden Variables

We can obtain $p(x)$ by summing over all values of $h$

$$p(x) = \sum_h p(x, h) = \sum_h \frac{\exp(-E(x, h))}{Z} = \frac{\sum_h \exp(-E(x, h))}{\sum_{x, h} \exp(-E(x, h))} = \frac{Z_h(x)}{Z}$$
We can obtain \( p(x) \) by summing over all values of \( h \)

\[
p(x) = \sum_h p(x, h)
\]

\[
= \sum_h \frac{\exp(-E(x, h))}{Z}
\]

\[
= \frac{\sum_h \exp(-E(x, h))}{\sum_{x,h} \exp(-E(x, h))} = \frac{Z_h(x)}{Z}
\]
We can obtain $p(x)$ by summing over all values of $h$

$$p(x) = \sum_h p(x, h)$$

$$= \sum_h \frac{\exp(-E(x, h))}{Z}$$

$$= \frac{\sum_h \exp(-E(x, h))}{\sum_{x, h} \exp(-E(x, h))} = \frac{Z_h(x)}{Z}$$

NLL:

$$f(w) = -\frac{1}{N} \sum_t \log(p(x^{(t)}|w)) = \frac{1}{N} \sum_t (-\log(Z_h(x^{(t)}))) + \log(Z)$$
Learning with Hidden Variables

We can obtain $p(x)$ by summing over all values of $h$

$$p(x) = \sum_h p(x, h)$$

$$= \sum_h \frac{\exp(-E(x, h))}{Z}$$

$$= \frac{\sum_h \exp(-E(x, h))}{\sum_{x,h} \exp(-E(x, h))} = \frac{Z_h(x)}{Z}$$

NLL:

$$f(w) = -\frac{1}{N} \sum_t \log(p(x^{(t)}|w)) = \frac{1}{N} \sum_t (-\log(Z_h(x^{(t)}))) + \log(Z)$$

- Note that the second term is the same as fully observed case
- Now, even for a log-linear model the NLL is no longer convex
- We can use exact or approximate inference (as applicable) to evaluate both the terms
Restricted Boltzmann Machines

UGM with the following structure:

- No lateral connections
- $x$ and $h$ are both binary

The figures/slides are from videos by Hugo Larochelle, available at

https://www.youtube.com/watch?v=p4Vh_zMw-HQ
Restricted Boltzmann Machines

UGM with the following structure:

- No lateral connections
- $x$ and $h$ are both binary

The figures/slides are from videos by Hugo Larochelle, available at https://www.youtube.com/watch?v=p4Vh_zMw-HQ
A compact description:

Energy:

\[ E(x, h) = -x^T W h - b^T h - c^T x \]

\[ = - \sum_{jk} w_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k \]

Distribution:

\[ p(x, h) = \frac{\exp(-E(x, h))}{Z} \]
RBM: Inference

\[ p(x|h) = \prod_k p(x_k|h) \]
RBM: Inference

\[ p(x|h) = \prod_k p(x_k|h) \]

\[ p(x_k = 1|h) = \frac{1}{1 + \exp(-(c_k + h^T w_{.k})))} \]

\[ = \text{sigm}(c_k + h^T w_{.k}) \]
RBM: Inference

\[ p(x|h) = \prod_k p(x_k|h) \]

\[ p(x_k = 1|h) = \frac{1}{1 + \exp(-(c_k + h^T W_k))} \]
\[ = \text{sigm}(c_k + h^T W_k) \]

\[ p(h_j = 1|x) = \text{sigm}(b_j + W_j \cdot x) \]
Due to conditional independence:
- conditional distribution $p(x|h)$ factorizes
- we can calculate it in closed form
- decoding, inference and sampling is easy if $x$ or $h$ is given
RBM: Learning

- Given a set of examples \( \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \)
- Learn the parameters \( W, b, \) and \( c \)
- Example: a set of binary images from MNIST dataset
Given a set of examples \( \{ x^{(1)}, x^{(2)}, ..., x^{(N)} \} \)

Learn the parameters \( W, b, \) and \( c \)

Example: a set of binary images from MNIST dataset

Motivation

- Unsupervised feature discovery
- Compression/non-linear dimensionality reduction
- A generative model of the image
To minimize the NLL:

$$\arg\min_{W,b,c} \frac{1}{N} \sum_t - \log(p(x^{(t)}))$$

$$- \log(p(x^{(t)})) = - \log \left( \frac{\sum_h \exp(-E(x^{(t)}, h))}{\sum_{h,x} \exp(-E(x,h))} \right)$$

$$= - \log \left( \sum_h \exp(-E(x^{(t)}, h)) \right) + \log \left( \sum_{h,x} \exp(-E(x,h)) \right)$$
To minimize the NLL:

$$\text{arg min}_{W,b,c} \frac{1}{N} \sum_t - \log(p(x^{(t)}))$$

$$- \log(p(x^{(t)})) = - \log \left( \frac{\sum_h \exp(-E(x^{(t)}, h))}{\sum_{h,x} \exp(-E(x, h))} \right)$$

$$= - \log \left( \sum_h \exp(-E(x^{(t)}, h)) \right) + \log \left( \sum_{h,x} \exp(-E(x, h)) \right)$$

Let’s consider,

$$\frac{\partial \left( - \log p(x^{(t)}) \right)}{\partial W_{jk}} = \sum_h p(h|x^{(t)}) \frac{\partial E(x^{(t)}, h)}{\partial W_{jk}} - \sum_{h,x} p(x, h) \frac{\partial E(x, h)}{\partial W_{jk}}$$

$$= \mathbb{E}_{h|x} \left[ \frac{\partial E(x^{(t)}, h)}{\partial W_{jk}} \right] - \mathbb{E}_{x,h} \left[ \frac{\partial E(x, h)}{\partial W_{jk}} \right]$$
To minimize the NLL:

$$\arg \min_{W,b,c} \frac{1}{N} \sum_t - \log(p(x^{(t)}))$$

$$- \log(p(x^{(t)})) = - \log \left( \frac{\sum_h \exp(-E(x^{(t)}, h))}{\sum_{h,x} \exp(-E(x, h))} \right)$$

$$= - \log \left( \sum_h \exp(-E(x^{(t)}, h)) \right) + \log \left( \sum_{h,x} \exp(-E(x, h)) \right)$$

Let’s consider,

$$\frac{\partial (- \log p(x^{(t)}))}{\partial W_{jk}} = \sum_h p(h|x^{(t)}) \frac{\partial E(x^{(t)}, h)}{\partial W_{jk}} - \sum_{h,x} p(x, h) \frac{\partial E(x, h)}{\partial W_{jk}}$$

$$= \mathbb{E}_{h|x} \left[ \frac{\partial E(x^{(t)}, h)}{\partial W_{jk}} \right] - \mathbb{E}_{x,h} \left[ \frac{\partial E(x, h)}{\partial W_{jk}} \right]$$

So far we have not assumed an RBM. This can work for any hidden variable model.
Contrastive Divergence

Recall from previous slides:

\[ E(x, h) = - \sum_{jk} W_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k \]
Recall from previous slides:

\[ E(x, h) = -\sum_{jk} W_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k \]

Derivative:

\[ \frac{\partial E(x, h)}{\partial W_{jk}} = -h_j x_k \]
Recall from previous slides:

\[ E(x, h) = -\sum_{jk} W_{jk} h_j x_k - \sum_j b_j h_j - \sum_k c_k x_k \]

Derivative:

\[ \frac{\partial E(x, h)}{\partial W_{jk}} = -h_j x_k \]

Plugging in values:

\[ \frac{\partial \left( -\log p(x^{(t)}) \right)}{\partial W_{jk}} = \mathbb{E}_{h|x} \left[ \frac{\partial E(x^{(t)}, h)}{\partial W_{jk}} \right] - \mathbb{E}_{x,h} \left[ \frac{\partial E(x, h)}{\partial W_{jk}} \right] \]

\[ = -\mathbb{E}_{h|x} [h_j x_k^{(t)}] + \mathbb{E}_{x,h} [h_j x_k] \]

\[ \approx -\mathbb{E}_{h|x} [h_j x_k^{(t)}] + \mathbb{E}_{h|x} [h_j \tilde{x}_k] \]

\[ = -p(h_j = 1|x^{(t)}) x_k^{(t)} + p(h_j = 1|\tilde{x}) \tilde{x}_k \]
Update rule:

\[ W_{jk} \leftarrow W_{jk} + \alpha \left( p(h_j = 1|x^{(t)}_k)x^{(t)}_k - p(h_j = 1|\tilde{x})\tilde{x}_k \right) \]
Update rule:

\[ W_{jk} \leftarrow W_{jk} + \alpha \left( p(h_j = 1 | x^{(t)}) x_k^{(t)} - p(h_j = 1 | \tilde{x}) \tilde{x}_k \right) \]

- We can obtain similar expressions for \( b_j \) and \( c_k \)
Contrastive Divergence

Update rule:

\[ W_{jk} \leftarrow W_{jk} + \alpha \left( p(h_j = 1 | x^{(t)}_k) x^{(t)}_k - p(h_j = 1 | \tilde{x}) \tilde{x}_k \right) \]

- We can obtain similar expressions for \( b_j \) and \( c_k \)

Sampling \( \tilde{x} \): use block Gibb’s sampling
Contrastive Divergence

Putting everything together: CD-k algorithm

- For each training example $x^{(t)}$
  - Initialize a Gibb’s chain with $x^{(t)}$
  - Run $k$ rounds to obtain $\tilde{x}$
  - Update $W$, $b$, and $c$

- Go back to the first step until a stopping criteria
Contrastive Divergence

\[ W_{jk} \leftarrow W_{jk} + \alpha \left( p(h_j = 1|x^{(t)}_k x^{(t)}_k) - p(h_j = 1|\tilde{x})\tilde{x}_k \right) \]

CD intuition:
Persistent CD or Younes’ Algorithm

Pseudo code

- For each training example $x^{(t)}$
  - Initialize a Gibb’s chain with $\tilde{x}^{(t-1)}$
  - Run $k$ rounds to obtain $\tilde{x}$
  - Update $W$, $b$, and $c$
- Go back to the first step until a stopping criteria

Works better in theory as well as in practice.
Learned Features

Weights $W$ in an image form:
Sample the generative model

Samples obtained from an RBM trained on MNIST data
Gaussian-Bernoulli RBM

- Input $x$ can be real-valued
- Modified energy function

$$E(x, h) = -x^T W h - b^T h - c^T x + \frac{1}{2} x^T x$$

- $p(x|h)$ turns out to be a Gaussian distribution
Extensions of RBM

Gaussian-Bernoulli RBM

- Input $x$ can be real-valued
- Modified energy function
  \[
  E(x, h) = -x^T W h - b^T h - c^T x + \frac{1}{2} x^T x
  \]
- $p(x|h)$ turns out to be a Gaussian distribution

Deep Belief Networks

- Can be trained greedily one layer at a time (same as RBM training)
All the variables may not be observed in the training data. We can still learn the parameters of a UGM.

Restricted Boltzmann Machines (RBM) are binary UGMs with hidden variables (no lateral connection)

RBMs are useful for unsupervised feature discovery, non-linear dimensionality reduction etc.

RBMs can be trained efficiently using Persistent-CD