The neural autoregressive distribution estimator

Hugo Larochelle and Iain Murray

CS 598 presentation by Varun Kelkar
Outline

● Motivation
  ○ What is the paper trying to do?
  ○ Why is the problem important?
  ○ Why is the problem difficult?

● Prior approaches
  ○ Mixture of Bernoullis
  ○ Restricted Boltzmann machines
  ○ Bayesian Networks

● Approach

● Numerical studies

● Summary
Motivation

- *What the paper wishes to achieve:*
  
  Distribution estimation of high dimensional discrete/binary vectors.
Motivation

- *What the paper wishes to achieve:*
  Distribution estimation of high dimensional discrete/binary vectors.

- *Why is this problem important:*
  If one knows the joint distribution of objects, one can potentially begin to answer any question about the dependencies between them, including all of supervised learning.
Motivation

- **What the paper wishes to achieve:**
  Distribution estimation of high dimensional discrete/binary vectors.

- **Why is this problem important:**
  If one knows the joint distribution of objects, one can potentially begin to answer any question about the dependencies between them, including all of supervised learning.

- **Why is this problem difficult:**
  Curse of dimensionality, the PMF is a vector in a $d^n$-dimensional space, where $d$ is the number of discrete levels, $n$ is the dimensionality of the vector.
Previous approaches

- Mixture of Bernoullis (MoB)
- Restricted Boltzmann Machines (RBMs)
- Bayesian Networks
- Fully visible sigmoid belief network (FVSBN)
Restricted Boltzmann Machines (RBMs)

\[ h = Wv + b \]

Probabilities evaluated using the energy function:

\[ E(v, h) = -h^T Wv - b^T v - c^T h \quad (1) \]

Probabilities are assigned to any observation \( v \) as follows:

\[ p(v) = \sum_h \exp(-E(v, h))/Z, \quad (2) \]

Problems:

- Computing partition function \( Z \) is intractable for all except the small networks. Approximations needed.
- Hence, RBMs cannot be used to model parts of a probabilistic system.
- Difficulty in evaluating the learned distribution
Bayesian networks

Strategy: Decompose the distribution using its conditionals

\[ p(\mathbf{v}) = \prod_{i=1}^{D} p(v_i|\mathbf{v}_{\text{parents}(i)}) , \quad (3) \]

Example: Fully visible sigmoid belief networks

\[ p(v_i|\mathbf{v}_{\text{parents}(i)}) = \text{sigm}(b_i + \sum_{j<i} W_{ij}v_j) , \quad (4) \]
Converting RBMs into Bayesian networks

Rewrite the RBM PDF in terms of conditionals

\[
p(v) = \prod_{i=1}^{D} p(v_i | v_{<i})
= \prod_{i=1}^{D} \frac{p(v_i, v_{<i})}{p(v_{<i})}
= \prod_{i=1}^{D} \frac{\sum_{v_j \geq i} \sum_h \exp(-E(v, h))}{\sum_{v_j \geq i} \sum_h \exp(-E(v, h))},
\]

(5)

Use a simplified model for the still intractable conditionals

\[
q(v_i, v_{<i}, h | v_{<i}) = \mu_i(i)^{v_i} (1 - \mu_i(i))^{1 - v_i} \\
\prod_{j > i} \mu_j(i)^{v_j} (1 - \mu_j(i))^{1 - v_j} \\
\prod_k \tau_k(i)^{h_k} (1 - \tau_k(i))^{1 - h_k},
\]
Converting RBMs into Bayesian networks

Minimize the KL divergence by setting its gradients to 0, which gives the following. Use fixed-point iterations to find the parameters of the distribution:

\[
\tau_k(i) = \text{sigm} \left( c_k + \sum_{j \geq i} W_{kj} \mu_j(i) + \sum_{j < i} W_{kj} v_j \right) \quad (7)
\]

\[
\mu_j(i) = \text{sigm} \left( b_j + \sum_k W_{kj} \tau_k(i) \right) \quad \forall j \geq i. \quad (8)
\]

Problems:

- Can be slow to converge.
- Needs to be repeated for each component \( v_i \).
Neural autoregressive distribution estimators (NADE)

Taking inspiration from the first fixed-point iteration above, formulate the network architecture.

\[
p(v_i = 1 | v_{<i}) = \text{sigm} \left( b_i + (W^\top_i) h_i \right)
\]

\[
h_i = \text{sigm} (c + W_{:,<i} v_{<i})
\]

Training: Minimize the log-likelihood averaged over a training dataset

\[
\frac{1}{T} \sum_{t=1}^{T} - \log p(v_t) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{D} - \log p(v_i | v_{<i})
\]
## Results of experiments

Experiments comparing various baselines using the average test log-likelihood (ALL) relative to the MoB baseline

<table>
<thead>
<tr>
<th>Model</th>
<th>ADULT</th>
<th>CONNECT-4</th>
<th>DNA</th>
<th>MUSHROOMS</th>
<th>NIPS-0-12</th>
<th>OCR-LETTERS</th>
<th>RCV1</th>
<th>WEB</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoB</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>± 0.10</td>
<td>± 0.04</td>
<td>± 0.53</td>
<td>± 0.10</td>
<td>± 1.12</td>
<td>± 0.32</td>
<td>± 0.11</td>
<td>± 0.23</td>
</tr>
<tr>
<td>RBM</td>
<td>4.18</td>
<td>0.75</td>
<td>1.29</td>
<td>-0.69</td>
<td>12.65</td>
<td>-2.49</td>
<td>-1.29</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>± 0.06</td>
<td>± 0.02</td>
<td>± 0.48</td>
<td>± 0.09</td>
<td>± 1.07</td>
<td>± 0.30</td>
<td>± 0.11</td>
<td>± 0.20</td>
</tr>
<tr>
<td>RBM</td>
<td>4.15</td>
<td>-1.72</td>
<td>1.45</td>
<td>-0.69</td>
<td>11.25</td>
<td>0.99</td>
<td>-0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>mult.</td>
<td>± 0.06</td>
<td>± 0.03</td>
<td>± 0.40</td>
<td>± 0.05</td>
<td>± 1.06</td>
<td>± 0.29</td>
<td>± 0.11</td>
<td>± 0.21</td>
</tr>
<tr>
<td>RBForest</td>
<td>4.12</td>
<td>0.59</td>
<td>1.39</td>
<td>0.04</td>
<td>12.61</td>
<td>3.78</td>
<td>0.56</td>
<td>-0.15</td>
</tr>
<tr>
<td></td>
<td>± 0.06</td>
<td>± 0.02</td>
<td>± 0.49</td>
<td>± 0.07</td>
<td>± 1.07</td>
<td>± 0.28</td>
<td>± 0.11</td>
<td>± 0.21</td>
</tr>
<tr>
<td>FVSBN</td>
<td>7.27</td>
<td>11.02</td>
<td>14.55</td>
<td>4.19</td>
<td>13.14</td>
<td>1.26</td>
<td>-2.24</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>± 0.04</td>
<td>± 0.01</td>
<td>± 0.50</td>
<td>± 0.05</td>
<td>± 0.98</td>
<td>± 0.23</td>
<td>± 0.11</td>
<td>± 0.20</td>
</tr>
<tr>
<td>NADE</td>
<td>7.25</td>
<td>11.42</td>
<td>13.38</td>
<td>4.65</td>
<td>16.94</td>
<td>13.34</td>
<td>0.93</td>
<td>1.77</td>
</tr>
<tr>
<td></td>
<td>± 0.05</td>
<td>± 0.01</td>
<td>± 0.57</td>
<td>± 0.04</td>
<td>± 1.11</td>
<td>± 0.21</td>
<td>± 0.11</td>
<td>± 0.20</td>
</tr>
<tr>
<td>Normalization</td>
<td>-20.44</td>
<td>-23.41</td>
<td>-98.19</td>
<td>-14.46</td>
<td>-290.02</td>
<td>-40.56</td>
<td>-47.59</td>
<td>-30.16</td>
</tr>
</tbody>
</table>
Generative performance for binarized images

Figure 2: (Left): samples from NADE trained on a binary version of MNIST. (Middle): probabilities from which each pixel was sampled. (Right): visualization of some of the rows of $W$. This figure is better seen on a computer screen.
Summary

- MoBs not sufficiently expressive to model the complex dependencies in high dimensional distributions
- RBMs can have an intractable computation of the partition function, rendering it difficult to use for downstream applications
- Bayesian networks, such as fully visible sigmoid belief networks may still be less expressive than desired.
- Approaches that convert RBMs to Bayesian networks are slow to converge.
- The proposed approach outperforms other approaches because it is able to utilize the recursiveness in Bayesian network architectures to decompose a complex probability distribution to a tractable form, while still having sufficient generality in the form of the network architecture chosen.
References
