Density estimation

Computing, and avoiding, partition functions

Roadmap:
— Motivation: density estimation
— Understanding annealing/tempering
— NADE

Iain Murray
School of Informatics, University of Edinburgh
Includes work with Ruslan Salakhutdinov and Hugo Larochelle
Probabilistic model $\mathcal{H}$

Predict new images: $P(x \mid \mathcal{H})$
High density can be boring

\[ P(x \mid \mathcal{H}) \]
Image reconstruction

Observation model: $P(y \mid x)$

Underlying image:

$$P(x \mid y) \propto P(y \mid x) \cdot P(x)$$

(e.g., Zoran and Weiss, 2011; Lucas Theis’s work)
Roadmap

— Unsupervised learning and $P(x \mid \mathcal{H})$

— Evaluating $P(x \mid \mathcal{H})$
  Salakhutdinov and Murray (2008)
  Murray and Salakhutdinov (2009)
  Wallach, Murray, Salakhutdinov & Mimno (2009)

— NADE: “density estimation put first”
  Larochelle and Murray (2011)
Restricted Boltzmann Machines

\[ P(x, h \mid \theta) = \frac{1}{Z(\theta)} \exp \left[ \sum_{ij} W_{ij} x_i h_j + \sum_i b^x_i x_i + \sum_j b^h_j h_j \right] \]

\[ P(x \mid \theta) = \sum_h P(x, h \mid \theta) = \frac{1}{Z(\theta)} \sum_h \exp \left[ \cdots \right] \]

\[ f(x; \theta), \text{ tractable} \]
\[ p(x) = \frac{f(x)}{\mathcal{Z}} \]

\[ x \sim \text{Uniform} \]

\[ x \sim \text{Model} \]
Annealing / Tempering

\[ P(x; \beta) \propto P^*(x)^\beta \pi(x)^{(1-\beta)} \]

\[
\begin{array}{cccccc}
\beta = 0 & \beta = 0.01 & \beta = 0.1 & \beta = 0.25 & \beta = 0.5 & \beta = 1
\end{array}
\]

\[
1/\beta = \text{“temperature”}
\]
Annealed Importance Sampling

\[ x_0 \sim p_0(x) \]

\[ P(X) : \quad x_0 \xrightarrow{\tilde{T}_1} x_1 \xrightarrow{\tilde{T}_2} x_2 \xrightarrow{\ldots} x_{K-1} \xrightarrow{\tilde{T}_K} x_K \]

\[ Q(X) : \quad x_0 \xleftarrow{T_1} x_1 \xleftarrow{T_2} x_2 \xleftarrow{\ldots} x_{K-1} \xrightarrow{T_K} x_K \]

\[ x_K \sim p_{K+1}(x) \]

\[ P(X) = \frac{P^*(x_K)}{Z} \prod_{k=1}^{K} \tilde{T}_k(x_{k-1}; x_k), \quad Q(X) = \pi(x_0) \prod_{k=1}^{K} T_k(x_k; x_{k-1}) \]

Standard importance sampling of \( P(X) = \frac{P^*(X)}{Z} \) with \( Q(X) \)
Annealed Importance Sampling

\[ Z \approx \frac{1}{S} \sum_{s=1}^{S} \frac{P^*(X)}{Q(X)} \]

Number of AIS runs

log \( Z \)

Large Variance

20 sec
3.3 min
17 min
33 min
5.5 hrs

Estimated logZ

True logZ

Number of AIS runs
Parallel tempering

**Standard MCMC:** Transitions + swap proposals on joint:

$$P(X) = \prod_{\beta} P(X; \beta)$$

- larger system
- information from low $\beta$ diffuses up by slow random walk
Tempered transitions

**Drive temperature up...**

\[ P(X) : \]

\[ \hat{x}_0 \sim P(x) \]

... and back down

**Proposal:** swap order, final point \( \tilde{x}_0 \) putatively \( \sim P(x) \)

**Acceptance probability:**

\[
\min \left[ 1, \frac{P_{\beta_1}(\hat{x}_0)}{P(\hat{x}_0)} \ldots \frac{P_{\beta_K}(\hat{x}_{K-1}) P_{\beta_{K-1}}(\tilde{x}_{K-1})}{P_{\beta_{K-1}}(\hat{x}_0) P_{\beta_K}(\tilde{x}_{K-1})} \ldots \frac{P(\tilde{x}_0)}{P_{\beta_1}(\tilde{x}_0)} \right]
\]
Whirlwind tour of annealing / tempering
Must be able to get anywhere in distribution
Methods to use generally for hardest problems.
An experiment

Take 60,000 binarized MNIST digits, like these:

— Train an RBM using CD (and then find $\mathcal{Z}$)
— Train a mixture of multivariate Bernoullis with EM

Compare samples and test-set log-probs
A comparison

Samples from:

• mixture of Bernoullis, $-143$ nats/test digit

• RBM, $-106$ nats/test digit

Which is which?
A better fitted RBM

RBM samples

Training set examples

Test log-prob now 20 nats better (−86 nats/digit)
Dependent latent variables

“Deep Belief Net”

Lateral connections

Directed model

\[ P(x) = \frac{1}{\mathcal{Z}} \sum_h P^*(x, h), \text{ not available} \]
Chib-style estimates

Bayes Rule:

\[ P(h \mid x) = \frac{P(h, x)}{P(x)} \]

For any special state \( h^* \):

\[ P(x) = \frac{P(h^*, x)}{P(h^* \mid x)} \leftarrow \text{Estimate} \]

Murray and Salakhutdinov (2009)
\[
\log P(x) = \log \sum_h \frac{1}{Z} P^*(x, h)
\]

\[
\geq \sum_h Q(h) \log P^*(x, h) - \log Z + \mathcal{H}[Q(h)]
\]
Results MNIST

Estimated Test Log-probability vs. Number of Markov chain steps for AIS Estimator and Our Proposed Estimator.
Results Natural Scenes

![Graph showing the comparison between AIS Estimator, Our Proposed Estimator, and Estimate of Variational Lower Bound. The x-axis represents the number of Markov chain steps, ranging from 5 to 40, and the y-axis represents the estimated test log-probability, ranging from -585 to -565. The graph illustrates the performance of the estimators across different steps.](image-url)
$P(x \mid \mathcal{H})$ taught me

— RBM: state-of-the-art for binary dists

— Deep nets only very slightly better on MNIST

— Some Gaussian RBMs are really bad... ...and going deep won’t help

— Most topic model $P(x \mid \mathcal{H})$ ests wrong
Roadmap

— Unsupervised learning and $P(x \mid \mathcal{H})$

— Evaluating $P(x \mid \mathcal{H})$
  Salakhutdinov and Murray (2008)
  Murray and Salakhutdinov (2009)
  Wallach, Murray, Salakhutdinov & Mimno (2009)

— NADE: “density estimation put first”
  Larochelle and Murray (2011)
Decompose into scalars

\[ P(x) = P(x_1) P(x_2 | x_1) P(x_3 | x_1, x_2) \ldots \]

\[ = \prod_{k} P(x_k | x_{<k}) \]

Fully Visible Bayesian Networks

• Good at estimating (tractable)
• Not as good a model as RBMs
FVSBN: Fully Visible Sigmoid Belief Net

Logistic regression for conditionals

FVSBNs beat mixtures, but not RBMs
Approximate RBM

\[ P(x_k \mid x_{<k}) \] from MCMC

or mean field

(Requires fitted RBM. Creates new model.)
One Mean Field step

We turn this into an efficient autoencoder by:
1. by using only 1 up-down iteration
2. untying the up and down weights
3. by fitting

\[
\hat{x}_k = \sigma \left( b^x_k + W_{k,.} h^{(k)} \right)
\]

\[
h^{(k)} = \sigma \left( b^h + W_{.,<k}^\top x_{<k} \right)
\]
Neural Autoregressive Distribution Estimator (NADE)

We turn this into an efficient autoencoder by:
1. by using only 1 up-down iteration
2. untying the up and down weights
3. by fitting $h(k) = \text{sigmoid}(b + W \cdot x^{<k})$ to the data

\[
x_k = \text{sigmoid}(c_k + V_k \cdot h(k))
\]

Fit as new model

\[
P(x | \mathcal{H}) = \prod_k \hat{p}(x_k)
\]

Tractable, $\mathcal{O}(DH)$
While NADE was inspired by the RBM, does its performance come close to that of the RBM in its most typical regime, i.e. with hundreds of hidden units? In other words, was tractability gained with a loss in performance?

It then appears that tractability was gained at almost the confidence intervals of Table 1. Hence, it does not seem necessary to optimize the ordering of the 12 datasets. We then computed the standard deviation of the twelve associated test log-likelihood averages, for each of the datasets. Standard deviations of 0.045, 0.050 and 0.150 were observed on DNA = +/- 0.05, MUSHROOMS = +/- 0.045 and NIPS-0-12 = +/- 0.15 respectively, which is quite reasonable when compared to the intrinsic uncertainty associated with using a finite test set (see Table 1). Hence, little variation when changing input ordering:

DNA = +/- 0.05
MUSHROOMS = +/- 0.045
NIPS-0-12 = +/- 0.15

Table 1: Distribution estimation results. To normalize the results, the average test log-likelihood of each model on a given dataset was subtracted by the all of MoB (which is given in the last row under "Normalization"), so that even better performance could have been achieved with a better optimization method than stochastic gradient ascent.

Little variation when changing input ordering:

DNA = +/- 0.05
MUSHROOMS = +/- 0.045
NIPS-0-12 = +/- 0.15

NADE results

<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></td>
<td></td>
<td></td>
<td></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>
## NADE results

<table>
<thead>
<tr>
<th>Model</th>
<th>Log. Like.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoB*</td>
<td>-137.64</td>
</tr>
<tr>
<td>RBM (CD1)*</td>
<td>≈ -125.53</td>
</tr>
<tr>
<td>RBM (CD3)*</td>
<td>≈ -105.50</td>
</tr>
<tr>
<td>RBM (CD25)*</td>
<td>≈ -86.34</td>
</tr>
<tr>
<td>FVSBN</td>
<td>-97.45</td>
</tr>
<tr>
<td>NADE</td>
<td>-88.86</td>
</tr>
</tbody>
</table>

### Intractable

Model results for intractable models on a binarized version of MNIST:

- **MoB**: -137.64
- **RBM (CD1)**: ≈ -125.53
- **RBM (CD3)**: ≈ -105.50
- **RBM (CD25)**: ≈ -86.34
- **FVSBN**: -97.45
- **NADE**: -88.86

*References:*

Talking points

When should we learn $P(x \mid \mathcal{H})$?

Monte Carlo methods

Autoregressive models (see also Lucas Theis)

A longer talk on NADE:
http://videolectures.net/aistats2011_larochelle_neural/
Appendix slides
Markov chain estimation

Stationary condition for Markov chain:

$$P(h^*|x) = \sum_{h} T(h^* \leftarrow h) P(h|x)$$

$$\sim \left[ \frac{1}{S} \sum_{s=1}^{S} T(h^* \leftarrow h^{(s)}), \quad h^{(s)} \sim \mathcal{P}(H) \right] = \hat{p}$$

$\mathcal{P}(H)$ draws a sequence from an equilibrium Markov chain:

$h^{(1)} \sim P(h|v)$
Bias in answer

\[ P(x) = \frac{P(h^*, x)}{P(h^* | x)} = \frac{P(h^*, x)}{\mathbb{E}[\hat{p}]} \leq \mathbb{E} \left[ \frac{P(h^*, x)}{\hat{p}} \right] \]

Idea: bias Markov chain by starting at \( h^* \)

\[ \frac{1}{S} \sum_{s=1}^{S} T(h^* \leftarrow h^{(s)}) \] will often overestimate \( P(h^* | x) \)

\[ Q(H) = \frac{T(h^{(1)} \leftarrow h^*)}{P(h^{(1)} | x)} \mathcal{P}(H) \]
New estimator

We actually need a slightly more complicated $Q$:

$$Q(H) = \frac{1}{S} \sum_{s=1}^{S} \tilde{T}(h^{(s)} \leftarrow h^*) \frac{P(h^{(s)}|x)}{P(h^*) P(H)}$$

$$\mathbb{E}_{Q(H)} \left[ 1 / \frac{1}{S} \sum_{s=1}^{S} T(h^* \leftarrow h^{(s)}) \right] = \frac{1}{P(h^*|x)}$$

$$\hat{P}(x) = \frac{P(x, h^*)}{\hat{P}(h^*|x)} \text{ unbiased } \Rightarrow \text{ stochastic lower bound on } \log P(x)$$