Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics

Michael U. Gutmann and Aapo Hyvarinen

Presented by Yitong Li

November 10, 2016
1 Overview

2 Noise-Contrastive Estimation

3 Simulations

4 Computational Performance

5 Simulations with Natural images
1. Overview

2. Noise-Contrastive Estimation

3. Simulations

4. Computational Performance

5. Simulations with Natural images
Overview

Preliminary

- Input data $X = \{x_1, \cdots, x_T\}$ where $x \in R^n$, have unknown distribution $p_d(.)$.
- $p_d(.)$ is modeled by a parameterized family of functions $\{p_m(\cdot; \theta)\}_\theta$. In other words, $p_d(.) = p_m(\cdot; \theta^*)$.
- Any estimate $\hat{\theta}$ must satisfies $\int p_m(u; \hat{\theta}) du = 1$, and $p_m(\cdot; \hat{\theta}) \geq 0$.
- The unnormalized model is parameterized by $\alpha$, we denote it by $p^0_m(\cdot; \alpha)$.
- The partition function $Z(\alpha) = \int p^0_m(u; \alpha) du$. 
The main idea of this paper is to use noise contrastive estimation to get $c = Z(\alpha)$. Further, $c$ is defined as a parameter, instead of a function of $\alpha$. 

$$\ln p_m(.; \theta) = \ln p_m^0(.; \alpha) + c$$  \hspace{1cm} (1)$$

with parameter vector $\theta = (\alpha, c)$. The estimate $\hat{\theta} = (\hat{\alpha}, \hat{c})$ is then used as parameters for a normalized model.
1. Overview

2. Noise-Contrastive Estimation

3. Simulations

4. Computational Performance

5. Simulations with Natural images
Let’s assume that the reference (noise) data $Y$ is an i.i.d sample $(y_1, \cdots, y_{T_n})$ of a random variable $y \in \mathbb{R}^n$ with pdf $p_n$. Recall our observed data is $X = \{x_1, \cdots, x_{T_d}\}$ with pdf $p_d(.)$. If we know $p_n$ and the ratio $\frac{p_d}{p_n}$, then we can get $p_d$. 

\[ p_n \times \frac{p_d}{p_n} = p_d \]
Classification Problem

Define $U = (u_1, \cdots, u_{T_d+T_n})$ is the union of the two sets $X$ and $Y$. Each sample $u_t$ is attached with a label $C_t$.

$$C_t = \begin{cases} 
0, & u_t \in Y \\
1, & u_t \in X 
\end{cases}$$

The class conditional probability is:

$$p(u|C = 1; \theta) = p_m(u; \theta) \quad p(u|C = 0) = p_n(u)$$
Posterior Probability

The prior probabilities are \( p(C = 1) = \frac{T_d}{T_d + T_n} \) and \( P(C = 0) = \frac{T_n}{T_d + T_n} \). The posterior probabilities for the classes are

\[
\begin{align*}
    p(C = 1 | u; \theta) &= \frac{p_m(u; \theta)}{p_m(u; \theta) + v p_n(u)} \\
    p(C = 0 | u; \theta) &= \frac{v p_n(u)}{p_m(u; \theta) + v p_n(u)}
\end{align*}
\]

where \( v \) is the ratio \( T_n/T_d \). Further, denote \( r_v(u) = \frac{1}{1 + v \exp(-u)} \), we introduce some new notations.

\[
\begin{align*}
    G(u; \theta) &= \ln p_m(u; \theta) - \ln p_n(u) \\
    h(u; \theta) &= P(C = 1 | u; \theta) = r_v(G(u; \theta))
\end{align*}
\]
The class labels $C_t$ are assumed Bernoulli distributed and independent. The conditional log-likelihood is given by

$$L(\theta) = \sum_{t=1}^{T_d + T_n} C_t \ln P(C_t = 1 | u_t; \theta) + (1 - C_t) \ln P(C_t = 0 | u_t; \theta)$$

Optimizing $L(\theta)$ w.r.t $\theta$ leads to an estimate $G(.; \hat{\theta})$, which is the log-ratio $\ln(p_d/p_n)$. Since we know $p_n$, we can estimate $p_d$. 

$$L(\theta) = \sum_{t=1}^{T_d} \ln [h(x_t; \theta)] + \sum_{t=1}^{T_n} \ln [1 - h(y_t; \theta)]$$

(3)
Definition of the Estimator

The estimator is $\hat{\theta}_T$, which maximizes

$$J_T(\theta) = \frac{1}{T_d} \left\{ \sum_{t=1}^{T_d} \ln[h(x_t; \theta)] + \sum_{t=1}^{T_n} \ln[1 - h(y_t; \theta)] \right\}$$

(4)

As the number of observation $X$ increases, $J_T(\theta)$ converges in probability to $J$.

$$J(\theta) = \mathbb{E}\{\ln[h(x; \theta)]\} + v\mathbb{E}\{\ln[1 - h(y; \theta)]\}$$

(5)

Denote the optimized value for $J(\theta)$ is $\theta^*$, the optimized value for $J_T(\theta)$ is $\hat{\theta}_T$, then we have several properties for $\theta^*$ and $\theta_T$. 


Properties of the Estimator

Properties between $\theta^*$ and $\hat{\theta}_T$: (Please refer to the original paper for the detailed theorems)

- If $p_n$ is non-zero when $p_d$ is non-zero, then $J(\theta)$ attains maximum at $p_m(.; \theta) = p_d$.
- If $p_n$ and $p_d$ satisfies some conditions, then $\hat{\theta}_T$ converges in probability to $\theta^*$.
- $\sqrt{T_d}(\hat{\theta}_T - \theta^*)$ is asymptotically normal with mean zero.
1 Overview

2 Noise-Contrastive Estimation

3 Simulations

4 Computational Performance

5 Simulations with Natural images
Gaussian Model

For a Gaussian model, we have

\[
\ln p_d(x) = -\frac{1}{2} x^T \Lambda^* x + c^*
\]

\[
c^* = -\frac{1}{2} \ln |\det \Lambda^*| - \frac{n}{2} \ln(2\pi)
\]  

We want to estimate from the unnormalized statistical model

\[
\ln p_m^0(x; \alpha) = -\frac{1}{2} x^T \Lambda x
\]

The parameter vector \(\alpha\) contains the coefficients for the lower triangular part of \(\Lambda\). The total parameter set is \(\theta = (\alpha, c)\), which is what we want to estimate.

\[
\ln p_m(x; \theta) = \ln p_m^0(x; \alpha) + c
\]
The results are derived by averaging over 500 experiment trials, where the data $\mathbf{x}$ is drawn from a 5 dimensional Gaussian distribution. Thicker lines are mean values, finer curves are 0.9 and 0.1 quantiles.

Note: NCE1 means noise contrastive estimation with $T_n/T_d = 1$, in other words, the ratio between number of reference data and number of true data.
ICA Model

The ICA model is defined as:

\[ x = As \] (7)

With \( B^* = A^{-1} \), the data distribution can be written as:

\[
\ln p_d(x) = \sum_{i=1}^{n} f(b_i^*x) + c^* \quad (8)
\]

where \( b_i^* \) is the \( i \)th row of \( B^* \). Similarly, we want to estimate from the model:

\[
\ln p_m(x; \theta) = \ln p_m^0(x; \alpha) + c \quad (9)
\]

where

\[
\ln p_m^0(x; \alpha) = \sum_{i=1}^{n} f(b_i x) \quad (10)
\]
ICA Model Result

(a) Mixing matrix

(b) Normalizing parameter
1 Overview

2 Noise-Contrastive Estimation

3 Simulations

4 Computational Performance

5 Simulations with Natural images
Methods Compared

In this section, simulations also based on the ICA model. The author also considers several other energy function estimation methods.

- NCE: The proposed Noise Contrastive Estimation
- IS: Monte Carlo Maximum Likelihood
- SM: Score Matching

In each simulation, the dimension of $\mathbf{x}$ is set to 10, which a total of $T_d = 8000$ points.
For the left figure, each point is from one simulation.
For the right figure, each circle corresponds to $v = T_n/T_d \in \{1, 2, 5, 10, 20, 50, 100, 200, 400, 1000\}$. The ellipses are derived by fitting a Gaussian distribution with 90% points for a given $v$ falls into it.
Source Following a Logistic Density

Here, the ICA model is defined as $\ln p_d(x) = \sum_{i=1}^{n} f(b_i^* x) + c^*$,

$f(u) = -2 \ln \cosh \left( \frac{\pi}{2\sqrt{3}} u \right) \text{ and } c^* = \ln |B^*| + n \ln \left( \frac{\pi}{4\sqrt{3}} \right)$.
Analysis

- Noise Contrastive estimation is particularly well suited for the estimation of data distributions with heavy tails.
- In case of thin tails, noise-contrastive estimation performs similarly to Monte Carlo maximum likelihood.
- If the data distribution is particularly smooth, score matching may be the best option.
1 Overview

2 Noise-Contrastive Estimation

3 Simulations

4 Computational Performance

5 Simulations with Natural images
Data Description

Data is sampled from Hateren’s image database, which is a subset showing wildlife scenes only.

- Original image $i_t$ has dimension $d = 25^2 = 625$.
- Reduce image dimension to $n = 160$, which is the observation $x_t$.
- Given $x_t$, we can reconstruct the original image patch via
  \[ i_t = V^{-1}x_t, \quad V^{-1} = ED^{1/2} \]
  where $E$ is the $d \times n$ matrix formed by the leading $n$ eigenvectors of the covariance matrix of the image patches. $D$ contains the corresponding eigenvalues.
Compressed image $x$ is used as real data. Randomly sampled noise is used as reference. This figure shows the result of the decoded $x$ and noise.
The first model that we consider is

\[ \ln p_m(x; \theta) = \sum_{k=1}^{n} f(y_k; a_k, b_k) + c \]

\[ y_k = \sum_{i=1}^{n} Q_{ki}(w_i^T x)^2 \]

\[ f(y; a, b) = f_{th}(\ln(ay + 1) + b), y \geq 0 \]  \hspace{1cm} (11)

Where \( f_{th} \) is like the RELU function.
Learned First Layer features

The feature detectors $w_i$ in the first layer are ‘Gabor-like’.
Learned Second Layer features

The second-layer weights $Q_{ki}$ for five different $k$. The icons were created by representing each first-layer feature by a bar of the same orientation and similar length as the feature and then superimposing them with weights given by $Q_{ki}$. 

1. **Raw result**
2. **Graphical visualization**
Visualization of features
Difference between Noise and Image

(a) Learned nonlinearities

(b) Distribution of second-layer outputs $y_k$
The model that we consider is

$$\ln p_m(x; \theta) = \sum_{k=1}^{n} f(y_k; a_1, a_2, \cdots) + c$$

$$y_k = \sum_{i=1}^{n} Q_{ki}(w_i^T x)^2$$  \hspace{1cm} (12)

Function $f$ is cubic spline.
Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics

Visualization of features

(a) Pooling in the second layer

(b) Representation with icons
Visualization of features

The objective function for different models. Larger score is better.

<table>
<thead>
<tr>
<th></th>
<th>One-layer model</th>
<th></th>
<th>Spline</th>
<th></th>
<th>Two-layer model</th>
<th>Refinement</th>
<th>Spline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Thresholding</td>
<td>Laplacian</td>
<td></td>
<td></td>
<td>Thresholding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$J_T$, av</td>
<td>-1.871</td>
<td>-1.518</td>
<td>-1.062</td>
<td></td>
<td>-0.8739</td>
<td>-0.6248</td>
<td>-0.6139</td>
</tr>
<tr>
<td>$J_T$, std</td>
<td>0.0022</td>
<td>0.0035</td>
<td>0.0030</td>
<td></td>
<td>0.0029</td>
<td>0.0030</td>
<td>0.0037</td>
</tr>
<tr>
<td>$L_T$, av</td>
<td>-223.280</td>
<td>-222.714</td>
<td>-219.786</td>
<td></td>
<td>-220.739</td>
<td>-213.303</td>
<td>-212.598</td>
</tr>
<tr>
<td>$L_T$, std</td>
<td>0.0029</td>
<td>0.0077</td>
<td>0.0137</td>
<td></td>
<td>0.0088</td>
<td>0.0282</td>
<td>0.0273</td>
</tr>
</tbody>
</table>

$L_T$ is the log likelihood of the model. It’s defined as

$$L_T = \frac{1}{T} \sum_{t=1}^{T} \ln p^0_m(x_t; \hat{\alpha}) + \hat{c}$$
The main contribution of this paper is a new estimation method for unnormalized models.

This estimation is performed by discriminating between the observed data and some artificially generated noise.

In the modeling of natural images, the model deals ten-thousands of parameters which demonstrates that this model can handle demanding estimation problems.