Noise-Contrastive Estimation and its Generalizations

Michael Gutmann

https://sites.google.com/site/michaelgutmann

University of Helsinki  Aalto University
Helsinki Institute for Information Technology

21st April 2016
Problem statement

- Task: Estimate the parameters $\theta$ of a parametric model $p(\cdot | \theta)$ of a $d$ dimensional random vector $x$
- Given: Data $X = (x_1, \ldots, x_n)$ (iid)
- Given: Unnormalized model $\phi(\cdot | \theta)$

\[
\int_{\xi} \phi(\xi; \theta) \, d\xi = Z(\theta) \neq 1 \quad p(x; \theta) = \frac{\phi(x; \theta)}{Z(\theta)} \tag{1}
\]

Normalizing partition function $Z(\theta)$ not known / computable.
Why does the partition function matter?

- Consider $p(x; \theta) = \frac{\phi(x; \theta)}{Z(\theta)} = \frac{\exp\left(-\theta \frac{x^2}{2}\right)}{\sqrt{2\pi}/\theta}$

- Log-likelihood function for precision $\theta \geq 0$

$$\ell(\theta) = -n \log \sqrt{\frac{2\pi}{\theta}} - \theta \sum_{i=1}^{n} \frac{x_i^2}{2}$$ (2)

- Data-dependent (blue) and independent part (red) balance each other.

- If $Z(\theta)$ is intractable, $\ell(\theta)$ is intractable.
Why is the partition function hard to compute?

\[ Z(\theta) = \int_{\xi} \phi(\xi; \theta) \, d\xi \]

- Integrals can generally not be solved in closed form.
- In low dimensions, \( Z(\theta) \) can be approximated to high accuracy.
- Curse of dimensionality: Solutions feasible in low dimensions become quickly computationally prohibitive as the dimension \( d \) increases.
Why are unnormalized models important?

- Unnormalized models are widely used.
- Examples:
  - models of images (Markov random fields)
  - models of text (neural probabilistic language models)
  - models in physics (Ising model)
  - ...

- Advantage: Specifying unnormalized models is often easier than specifying normalized models.
- Disadvantage: Likelihood function is generally intractable.
Program

Noise-contrastive estimation
   Properties
   Application

Bregman divergence to estimate unnormalized models
   Framework
   Noise-contrastive estimation as member of the framework
Program

Noise-contrastive estimation
  Properties
  Application

Bregman divergence to estimate unnormalized models
  Framework
  Noise-contrastive estimation as member of the framework
Intuition behind noise-contrastive estimation

- Formulate the estimation problem as a classification problem: observed data vs. auxiliary “noise” (with known properties)
- Successful classification \(\equiv\) learn the differences between the data and the noise
- Differences + known noise properties \(\Rightarrow\) properties of the data

- Unsupervised learning by supervised learning
- We used (nonlinear) logistic regression for classification
Logistic regression (1/2)

- Let $Y = (y_1, \ldots, y_m)$ be a sample from a random variable $y$ with known (auxiliary) distribution $p_y$.
- Introduce labels and form regression function:

$$P(C = 1|\mathbf{u}; \theta) = \frac{1}{1 + G(\mathbf{u}; \theta)} \quad G(\mathbf{u}; \theta) \geq 0 \quad (3)$$

- Determine the parameters $\theta$ such that $P(C = 1|\mathbf{u}; \theta)$ is
  - large for most $x_i$
  - small for most $y_i$. 
Logistic regression (2/2)

Maximize (rescaled) conditional log-likelihood using the labeled data \( \{(x_1, 1), \ldots, (x_n, 1), (y_1, 0), \ldots, (y_m, 0)\} \).

\[
J_n^{\text{NCE}}(\theta) = \frac{1}{n} \left( \sum_{i=1}^{n} \log P(C = 1|x_i; \theta) + \sum_{i=1}^{m} \log [P(C = 0|y_i; \theta)] \right)
\]

For large sample sizes \( n \) and \( m \), \( \hat{\theta} \) satisfying

\[
G(u; \hat{\theta}) = \frac{m p_y(u)}{n p_x(u)}
\]

is maximizing \( J_n^{\text{NCE}}(\theta) \). Without any normalization constraints. (proof in appendix)
Noise-contrastive estimation

(Gutmann and Hyvärinen, 2010; 2012)

- Assume unnormalized model $\phi(\cdot|\theta)$ is parametrized such that its scale can vary freely.

\[
\begin{align*}
\theta \rightarrow (\theta; c) & \quad \phi(u; \theta) \rightarrow \exp(c)\phi(u; \theta) \\
& \quad (5)
\end{align*}
\]

- Noise-contrastive estimation:
  1. Choose $p_y$
  2. Generate auxiliary data $Y$
  3. Estimate $\theta$ via logistic regression with

\[
G(u; \theta) = \frac{m}{n} \frac{p_y(u)}{\phi(u; \theta)}. \quad (6)
\]
Noise-contrastive estimation

Assume unnormalized model $\phi(.|\theta)$ is parametrized such that its scale can vary freely.

\[ \theta \rightarrow (\theta; c) \quad \phi(u; \theta) \rightarrow \exp(c)\phi(u; \theta) \quad (5) \]

Noise-contrastive estimation:

1. Choose $p_y$
2. Generate auxiliary data $Y$
3. Estimate $\theta$ via logistic regression with

\[ G(u; \theta) = \frac{m}{n} \frac{p_y(u)}{\phi(u; \theta)}. \quad (6) \]

\[ G(u; \theta) \rightarrow \frac{m}{n} \frac{p_y(u)}{p_x(u)} \quad \Rightarrow \quad \phi(u; \theta) \rightarrow p_x(u) \]
Example

- Unnormalized Gaussian:

\[ \phi(u; \theta) = \exp(\theta_2) \exp \left(-\theta_1 \frac{u^2}{2}\right), \quad \theta_1 > 0, \ \theta_2 \in \mathbb{R}, \quad (7) \]

- Parameters: \( \theta_1 \) (precision), \( \theta_2 \equiv c \) (scaling parameter)

Contour plot of \( J_{n_{\text{NCE}}}^{\text{NCE}}(\theta) \):

- Gaussian noise with \( \nu = m/n = 10 \)
- True precision \( \theta_1^* = 1 \)
- Black: normalized models
- Green: optimization paths
Statistical properties

Assume $p_x = p(. \mid \theta^*)$

Consistency: As $n$ increases,

$$\hat{\theta}_n = \arg\max_{\theta} J^\text{NCE}_n(\theta),$$

converges in probability to $\theta^*$.

Efficiency: As $\nu = m/n$ increases, for any valid choice of $p_y$, noise-contrastive estimation tends to “perform as well” as MLE (it is asymptotically Fisher efficient).
Let the data follow the ICA model $\mathbf{x} = \mathbf{A}\mathbf{s}$ with 4 sources.

$$
\log p(\mathbf{x}; \theta^*) = -\sum_{i=1}^{4} \sqrt{2}|\mathbf{b}_i^*\mathbf{x}| + c^*
$$

with $c^* = \log |\det \mathbf{B}^*| - \frac{4}{2} \log 2$ and $\mathbf{B}^* = \mathbf{A}^{-1}$.

To validate the method, estimate the unnormalized model

$$
\log \phi(\mathbf{x}; \theta) = -\sum_{i=1}^{4} \sqrt{2}|\mathbf{b}_i\mathbf{x}| + c
$$

with parameters $\theta = (\mathbf{b}_1, \ldots, \mathbf{b}_4, c)$.

Contrastive noise $p_y$: Gaussian with the same covariance as the data.
Validating the statistical properties with toy data

- Results for 500 estimation problems with random $\mathbf{A}$, for $\nu \in \{0.01, 0.1, 1, 10, 100\}$.
- MLE results: with properly normalized model

(a) Mixing matrix

(b) Normalizing constant
Computational aspects

- The estimation accuracy improves as $m$ increases.
- Trade-off between computational and statistical performance.
- Example: ICA model as before but with 10 sources. $n = 8000$, $\nu \in \{1, 2, 5, 10, 20, 50, 100, 200, 400, 1000\}$.

Performance for 100 random estimation problems:
Computational aspects

How good is the trade-off? Compare with

1. MLE where partition function is evaluated with importance sampling. Maximization of

$$J_{IS}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \phi(x_i; \theta) - \log \left( \frac{1}{m} \sum_{i=1}^{m} \frac{\phi(y_i; \theta)}{p_y(y_i)} \right)$$  \hspace{1cm} (11)

2. Score matching: minimization of

$$J_{SM}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{10} \frac{1}{2} \psi_j^2(x_i; \theta) + \psi'_j(x_i; \theta)$$  \hspace{1cm} (12)

with $$\psi_j(x; \theta) = \frac{\partial \log \phi(x; \theta)}{\partial x_j}$$ \hspace{1cm} (here: smoothing needed!)

(see Gutmann and Hyvärinen, 2012, for more comparisons)
Computational aspects

- NCE is less sensitive to the mismatch of data and noise distribution than importance sampling.
- Score matching does not perform well if the data distribution is not sufficiently smooth.
Application to natural image statistics

- Natural images ≡ images which we see in our environment
- Understanding their properties is important
  - for modern image processing
  - for understanding biological visual systems
Human visual object recognition

- Rapid object recognition by feed-forward processing
- Computations in middle layers poorly understood
- Our approach: learn the computations from data
- Idea: the units indicate how probable an input image is. (up to normalization)

(Gutmann and Hyvärinen, 2013)
Unnormalized model of natural images

- Three processing layers (> 2 \cdot 10^5 parameters)
- Fit to natural image data (d = 1024, n = 70 \cdot 10^6)
- Learned computations: detection of curvatures, longer contours, and texture.

(Gutmann and Hyvärinen, 2013)
Noise-contrastive estimation

Properties
Application

Bregman divergence to estimate unnormalized models

Framework
Noise-contrastive estimation as member of the framework
Bregman divergence between two vectors $a$ and $b$

$$d_\Psi(a, b) = \Psi(a) - (\Psi(b) + \Psi'(b)(a - b))$$

$\Psi$ : strictly convex function

$d_\Psi(a, b) = 0 \iff a = b$

$d_\Psi(a, b) > 0$ if $a \neq b$
Bregman divergence between two functions $f$ and $g$

- Compute $d_\psi(f(u), g(u))$ for all $u$ in their domain; take weighted average

$$\tilde{d}_\psi(f, g) = \int d_\psi(f(u), g(u))d\mu(u)$$  \hspace{1cm} (13)

$$= \int \psi(f) - [\psi(g) + \psi'(g)(f - g)]d\mu$$  \hspace{1cm} (14)

- Zero iff $f = g$ (a.e.); no normalization condition on $f$ or $g$

- Fix $f$, omit terms not depending on $g$,

$$J(g) = \int [-\psi(g) + \psi'(g)g - \psi'(g)f]d\mu$$  \hspace{1cm} (15)
Estimation of unnormalized models

\[ J(g) = \int \left[ -\Psi(g) + \Psi'(g)g - \Psi'(g)f \right] d\mu \]

- Idea: Choose \( f, g, \) and \( \mu \) so that we obtain a computable cost function for consistent estimation of unnormalized models.
- Choose \( f = T(p_x) \) and \( g = T(\phi) \) such that

\[ f = g \implies p_x = \phi \quad \text{(16)} \]

Examples:
  - \( f = p_x, \ g = \phi \)
  - \( f = \frac{p_x}{\nu p_y}, \ g = \frac{\phi}{\nu p_y} \)
  - \ldots

- Choose \( \mu \) such that the integral can either be computed in closed form or approximated as sample average.

(Gutmann and Hirayama, 2011)
Several estimation methods for unnormalized models are part of the framework

- Noise-contrastive estimation
- Poisson-transform (Barthelmé and Chopin, 2015)
- Score matching (Hyvärinen, 2005)
- Pseudo-likelihood (Besag, 1975)
- ...

Noise-contrastive estimation:

\[ \Psi(u) = u \log u - (1 + u) \log(1 + u) \tag{17} \]

\[ f(u) = \frac{\nu p_y(u)}{p_x(u)} \quad \text{d}\mu(u) = p_x(u)\text{d}u \tag{18} \]

(proof in appendix)
Conclusions

- Point estimation for parametric models with intractable partition functions (unnormalized models)
- Noise contrastive estimation
  - Estimate the model by learning to classify between data and noise
  - Consistent estimator, has MLE as limit
  - Applicable to large-scale problems
- Bregman divergence as general framework to estimate unnormalized models.
Maximizer of the NCE objective function

Noise-contrastive estimation as member of the Bregman framework
Appendix

Maximizer of the NCE objective function

Noise-contrastive estimation as member of the Bregman framework
Proof of Equation (4)

For large sample sizes $n$ and $m$, $\hat{\theta}$ satisfying

$$G(u; \hat{\theta}) = \frac{m}{n} \frac{p_y(u)}{p_x(u)}$$

is maximizing $J_n^{\text{NCE}}(\theta)$,

$$J_n^{\text{NCE}}(\theta) = \frac{1}{n} \left( \sum_{i=1}^{n} \log P(C = 1|x_i; \theta) + \sum_{i=1}^{m} \log [P(C = 0|y_i; \theta)] \right)$$

without any normalization constraints.
Proof of Equation (4)

\[ J_n^{\text{NCE}}(\theta) = \frac{1}{n} \left( \sum_{i=1}^{n} \log P(C = 1|x_i; \theta) + \sum_{i=1}^{m} \log [P(C = 0|y_i; \theta)] \right) \]

\[ = \frac{1}{n} \sum_{t=1}^{n} \log P(C = 1|x_i; \theta) + \frac{m}{n} \frac{1}{m} \sum_{t=1}^{m} \log [P(C = 0|y_i; \theta)] \]

Fix the ratio \( m/n = \nu \) and let \( n \to \infty \) and \( m \to \infty \). By law of large numbers, \( J_n^{\text{NCE}} \) converges to \( J^{\text{NCE}} \),

\[ J^{\text{NCE}}(\theta) = \mathbb{E}_x (\log P(C = 1|x; \theta)) + \nu \mathbb{E}_y (\log P(C = 0|y; \theta)) \] (19)

With \( P(C = 1|x; \theta) = \frac{1}{1+G(x;\theta)} \) and \( P(C = 0|y; \theta) = \frac{G(y;\theta)}{1+G(y;\theta)} \) ...
... we have

$$J^{\text{NCE}}(\theta) = -\mathbb{E}_x \log(1 + G(x; \theta)) + \nu \mathbb{E}_y \log G(y; \theta) - \nu \mathbb{E}_y \log (1 + G(y; \theta))$$ (20)

Consider the objective $J^{\text{NCE}}(\theta)$ as a function of $G$ rather than $\theta$,

$$J^{\text{NCE}}(G) = -\mathbb{E}_x \log(1 + G(x)) + \nu \mathbb{E}_y \log G(y) - \nu \mathbb{E}_y \log (1 + G(y))$$

$$= -\int p_x(\xi) \log(1 + G(\xi)) d\xi +$$

$$\nu \int p_y(\xi) (\log G(\xi) - \log(1 + G(\xi)))$$

Compute functional derivative $\delta J^{\text{NCE}}/\delta G$,

$$\frac{\delta J^{\text{NCE}}(G)}{\delta G} = -\frac{p_x(\xi)}{1 + G(\xi)} + \nu p_y(\xi) \left( \frac{1}{G(\xi)} - \frac{1}{1 + G(\xi)} \right)$$ (21)
\[
\frac{\delta J^{\text{NCE}}(G)}{\delta G} = - \frac{p_x(\xi)}{1 + G(\xi)} + \nu p_y(\xi) \left( \frac{1}{G(\xi)} - \frac{1}{1 + G(\xi)} \right) \quad (22)
\]

\[
= - \frac{p_x(\xi)}{1 + G(\xi)} + \nu p_y(\xi) \frac{1}{G(\xi)(1 + G(\xi))} \quad (23)
\]

\[
\frac{1}{G(\xi)} = 0 \quad (24)
\]

We obtain

\[
\frac{p_x(\xi)}{1 + G^*(\xi)} = \nu p_y(\xi) \frac{1}{G^*(\xi)(1 + G^*(\xi))} \quad (25)
\]

\[
G^*(\xi)p_x(\xi) = \nu p_y(\xi) \quad (26)
\]

\[
G^*(\xi) = \nu \frac{p_y(\xi)}{p_x(\xi)} \quad (27)
\]

\[
= \frac{m p_y(\xi)}{n p_x(\xi)} \quad (28)
\]

Evaluating \( \partial^2 J^{\text{NCE}}/\partial G^2 \) at \( G^* \) shows that \( G^* \) is a maximizer.
Maximizer of the NCE objective function

Noise-contrastive estimation as member of the Bregman framework
Proof

In noise-contrastive estimation, we maximize

\[ J_{n}^{\text{NCE}}(\theta) = \frac{1}{n} \left( \sum_{i=1}^{n} \log P(C = 1|x_i; \theta) + \sum_{i=1}^{m} \log [P(C = 0|y_i; \theta)] \right) \]

Sample version of

\[ J_{n}^{\text{NCE}}(\theta) = \mathbb{E}_x \left( \log P(C = 1|x; \theta) \right) + \nu \mathbb{E}_y \left( \log P(C = 0|y; \theta) \right) \]

With

\[ P(C = 1|u; \theta) = \frac{1}{1 + G(u; \theta)} \quad P(C = 0|u; \theta) = \frac{1}{1 + 1/G(u; \theta)} \]

\[ J_{n}^{\text{NCE}}(\theta) = -\mathbb{E}_x \log(1 + G(x; \theta)) - \nu \mathbb{E}_y \log(1 + 1/G(y; \theta)) \] (29)

where \( G(u; \theta) = \frac{\nu p_y(u)}{\phi(u; \theta)} \).
The general cost function in the Bregman framework is

\[
J(g) = \int \left[ -\Psi(g) + \Psi'(g)g - \Psi'(g)f \right] d\mu
\]  \hspace{1cm} (30)

With

\[
\Psi(g) = g \log(g) - (1 + g) \log(1 + g)
\]  \hspace{1cm} (31)

\[
\Psi'(g) = \log(g) - \log(1 + g)
\]  \hspace{1cm} (32)

we have

\[
J(g) = \int \left[ -g \log(g) + (1 + g) \log(1 + g) + \log(g)g - \log(1 + g)g \\
+ \log(g)f + \log(1 + g)f \right] d\mu
\]  \hspace{1cm} (33)
\[ J(g) = \int \left[ \log(1 + g) - \log(g)f + \log(1 + g)f \right] d\mu \tag{34} \]

\[ = \int \left[ \log(1 + g) + \log(1 + 1/g)f \right] d\mu \tag{35} \]

With

\[ f(u) = \frac{\nu p_y(u)}{p_x(u)} \quad g(u) = G(u; \theta) \quad d\mu(u) = p_x(u)du \tag{36} \]

we have

\[ J(G(.; \theta)) = \int p_x(u) \log(1 + G(u; \theta)) du \]

\[ + \nu p_y(u) \log(1 + 1/G(u; \theta)) du \tag{37} \]

\[ = - J^{NCE}(\theta) \tag{38} \]