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 $\boldsymbol{\theta}$ of a parametric model $p(.|\boldsymbol{\theta})$ of a d dimensional random vector $\mathbf{x}$
- Given: Data $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ (iid)
- Given: Unnormalized model $\phi(.|\boldsymbol{\theta})$

$$\int_{\boldsymbol{\xi}} \phi(\boldsymbol{\xi}; \boldsymbol{\theta})\, d\boldsymbol{\xi} = Z(\boldsymbol{\theta}) \neq 1 \qquad p(\mathbf{x}; \boldsymbol{\theta}) = \frac{\phi(\mathbf{x}; \boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \tag{1}$$

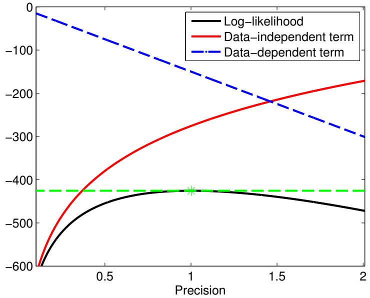
Normalizing partition function $Z(\boldsymbol{\theta})$ not known / computable.

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 $\theta \geq 0$

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

- Data-dependent (blue) and independent part (red) balance each other.
- If $Z(\boldsymbol{\theta})$ is intractable, $\ell(\boldsymbol{\theta})$ is intractable.



Why is the partition function hard to compute?

$$Z(\boldsymbol{\theta}) = \int_{\boldsymbol{\xi}} \phi(\boldsymbol{\xi}; \boldsymbol{\theta}) \, d\boldsymbol{\xi}$$

- Integrals can generally not be solved in closed form.
- In low dimensions, $Z(\boldsymbol{\theta})$ 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

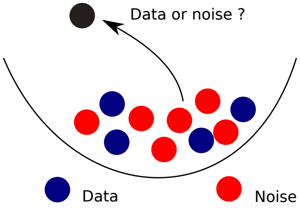
Bregman divergence to estimate unnormalized models

- 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

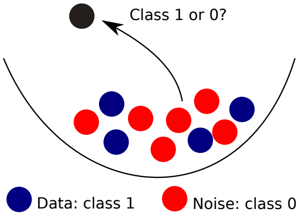


Logistic regression (1/2)

- Let $\mathbf{Y} = (\mathbf{y}_1, \ldots \mathbf{y}_m)$ be a sample from a random variable $\mathbf{y}$ with known (auxiliary) distribution $p_\mathbf{y}$.
- Introduce labels and form regression function:

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

- Determine the parameters $\boldsymbol{\theta}$ such that $P(C = 1|\mathbf{u};\boldsymbol{\theta})$ is
  - large for most $\mathbf{x}_i$
  - small for most $\mathbf{y}_i$.



Logistic regression (2/2)

- Maximize (rescaled) conditional log-likelihood using the labeled data $\{(\mathbf{x}_1, 1), \ldots, (\mathbf{x}_n, 1), (\mathbf{y}_1, 0), \ldots, (\mathbf{y}_m, 0)\}$,

$$J_n^{\mathrm{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\left(\sum_{i=1}^{n} \log P(C = 1|\mathbf{x}_i; \boldsymbol{\theta}) + \sum_{i=1}^{m} \log\left[P(C = 0|\mathbf{y}_i; \boldsymbol{\theta})\right]\right)$$

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

$$G(\mathbf{u}; \hat{\boldsymbol{\theta}}) = \frac{m}{n}\frac{p_{\mathbf{y}}(\mathbf{u})}{p_{\mathbf{x}}(\mathbf{u})} \tag{4}$$

is maximizing $J_n^{\mathrm{NCE}}(\boldsymbol{\theta})$. Without any normalization constraints. (proof in appendix)

Noise-contrastive estimation

(Gutmann and Hyvärinen, 2010; 2012)

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

$$\boldsymbol{\theta} \rightarrow (\boldsymbol{\theta}; c) \qquad\qquad \phi(\mathbf{u}; \boldsymbol{\theta}) \rightarrow \exp(c)\phi(\mathbf{u}; \boldsymbol{\theta}) \tag{5}$$

- Noise-contrastive estimation:
  1. Choose $p_{\mathbf{y}}$
  2. Generate auxiliary data $\mathbf{Y}$
  3. Estimate $\boldsymbol{\theta}$ via logistic regression with

$$G(\mathbf{u}; \boldsymbol{\theta}) = \frac{m}{n} \frac{p_{\mathbf{y}}(\mathbf{u})}{\phi(\mathbf{u}; \boldsymbol{\theta})}. \tag{6}$$

Noise-contrastive estimation

(Gutmann and Hyvärinen, 2010; 2012)

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

$$\boldsymbol{\theta} \rightarrow (\boldsymbol{\theta}; c) \qquad \phi(\mathbf{u}; \boldsymbol{\theta}) \rightarrow \exp(c)\phi(\mathbf{u}; \boldsymbol{\theta}) \tag{5}$$

- Noise-contrastive estimation:
  1. Choose $p_{\mathbf{y}}$
  2. Generate auxiliary data $\mathbf{Y}$
  3. Estimate $\boldsymbol{\theta}$ via logistic regression with

$$G(\mathbf{u}; \boldsymbol{\theta}) = \frac{m}{n} \frac{p_{\mathbf{y}}(\mathbf{u})}{\phi(\mathbf{u}; \boldsymbol{\theta})}. \tag{6}$$

- $G(\mathbf{u}; \boldsymbol{\theta}) \rightarrow \frac{m}{n} \frac{p_{\mathbf{y}}(\mathbf{u})}{p_{\mathbf{x}}(\mathbf{u})} \quad \Rightarrow \quad \phi(\mathbf{u}; \boldsymbol{\theta}) \rightarrow p_{\mathbf{x}}(\mathbf{u})$

Example

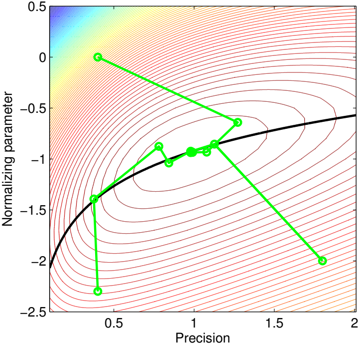
- Unnormalized Gaussian:

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

- Parameters: θ_1 (precision), $\theta_2 \equiv c$ (scaling parameter)

Contour plot of $J_n^{\mathrm{NCE}}(\boldsymbol{\theta})$:

- Gaussian noise with $\nu = m/n = 10$
- True precision $\theta_1^\star = 1$
- Black: normalized models
  Green: optimization paths



Statistical properties

(Gutmann and Hyvärinen, 2012)

- Assume $p_{\mathbf{x}} = p(.|\boldsymbol{\theta}^\star)$
- Consistency: As n increases,

$$\hat{\boldsymbol{\theta}}_n = \text{argmax}_{\boldsymbol{\theta}} J_n^{\text{NCE}}(\boldsymbol{\theta}), \tag{8}$$

  converges in probability to $\boldsymbol{\theta}^\star$.
- Efficiency: As $\nu = m/n$ increases, for any valid choice of $p_{\mathbf{y}}$, noise-contrastive estimation tends to "perform as well" as MLE (it is asymptotically Fisher efficient).

Validating the statistical properties with toy data

- Let the data follow the ICA model $\mathbf{x} = \mathbf{A}\mathbf{s}$ with 4 sources.

$$\log p(\mathbf{x}; \boldsymbol{\theta}^\star) = -\sum_{i=1}^{4} \sqrt{2}|\mathbf{b}_i^\star \mathbf{x}| + c^\star \tag{9}$$

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

- To validate the method, estimate the unnormalized model

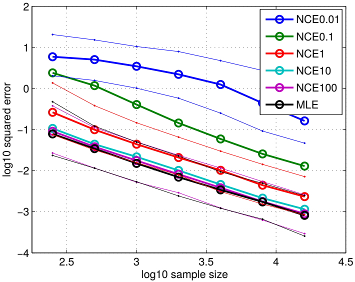
$$\log \phi(\mathbf{x}; \boldsymbol{\theta}) = -\sum_{i=1}^{4} \sqrt{2}|\mathbf{b}_i \mathbf{x}| + c \tag{10}$$

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

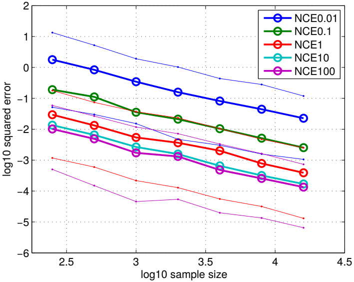
- Contrastive noise $p_\mathbf{y}$: Gaussian with the same covariance as the data.

Validating the statistical properties with toy data

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



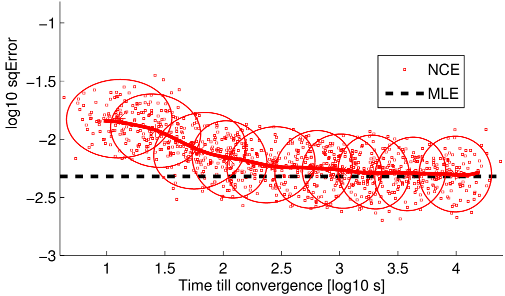
(a) Mixing matrix



(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

$$J_{\mathrm{IS}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \log \phi(\mathbf{x}_i; \boldsymbol{\theta}) - \log\left(\frac{1}{m}\sum_{i=1}^{m} \frac{\phi(\mathbf{y}_i; \boldsymbol{\theta})}{p_{\mathbf{y}}(\mathbf{y}_i)}\right) \quad (11)$$

2. Score matching: minimization of

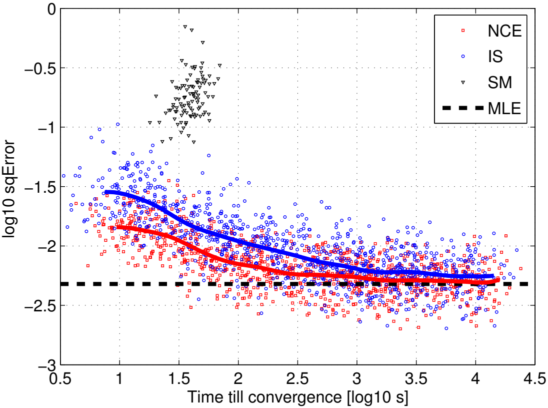
$$J_{\mathrm{SM}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{10} \frac{1}{2}\Psi_j^2(\mathbf{x}_i; \boldsymbol{\theta}) + \Psi_j'(\mathbf{x}_i; \boldsymbol{\theta}) \quad (12)$$

with $\Psi_j(\mathbf{x}; \boldsymbol{\theta}) = \frac{\partial \log \phi(\mathbf{x}; \boldsymbol{\theta})}{\partial x_j}$ (here: smoothing needed!)

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

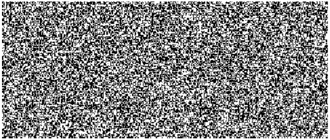
Computational aspects

- 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.



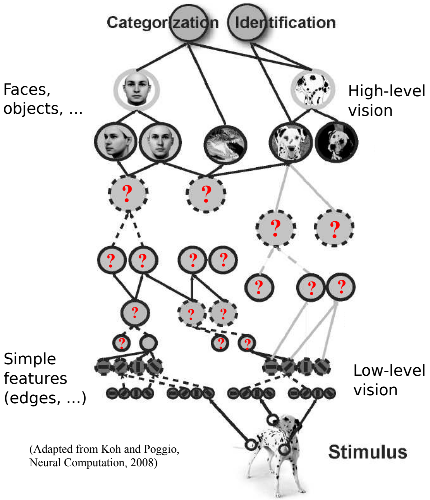
Application to natural image statistics

- Natural images $\equiv$ 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)

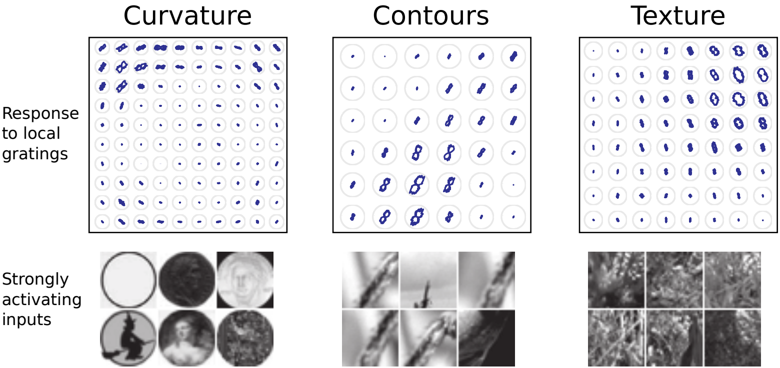


(Adapted from Koh and Poggio, Neural Computation, 2008)

(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)

Program

Noise-contrastive estimation

Properties

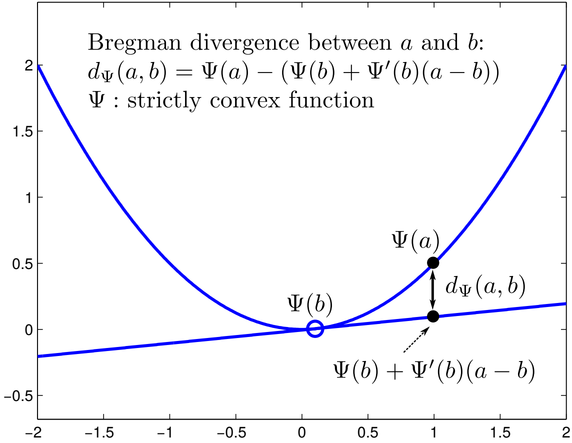
Application

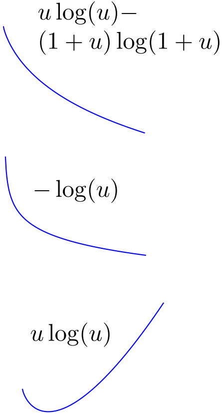
Bregman divergence to estimate unnormalized models

Framework

Noise-contrastive estimation as member of the framework

Bregman divergence between two vectors a and b





$d_\Psi(a,b) = 0 \Leftrightarrow a = b$ $\quad$ $d_\Psi(a,b) > 0$ if $a \neq b$

Bregman divergence between two functions f and g

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

$$\tilde{d}_\Psi(f, g) = \int d_\Psi(f(\mathbf{u}), g(\mathbf{u})) \mathrm{d}\mu(\mathbf{u}) \tag{13}$$

$$= \int \Psi(f) - \left[\Psi(g) + \Psi'(g)(f - g)\right] \mathrm{d}\mu \tag{14}$$

- Zero iff $f = g$ (a.e.); no normalization condition on f or g
- Fix f, omit terms not depending on g,

$$J(g) = \int \left[-\Psi(g) + \Psi'(g)g - \Psi'(g)f \right] \mathrm{d}\mu \tag{15}$$

Estimation of unnormalized models

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

- Idea: Choose f, g, and μ so that we obtain a computable cost function for consistent estimation of unnormalized models.
- Choose $f = T(p_{\mathbf{x}})$ and $g = T(\phi)$ such that

$$f = g \Rightarrow p_{\mathbf{x}} = \phi \tag{16}$$

  Examples:
  - $f = p_{\mathbf{x}}$, $g = \phi$
  - $f = \frac{p_{\mathbf{x}}}{\nu p_{\mathbf{y}}}$, $g = \frac{\phi}{\nu p_{\mathbf{y}}}$
  - ...
- Choose μ such that the integral can either be computed in closed form or approximated as sample average.

(Gutmann and Hirayama, 2011)

Estimation of unnormalized models

(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:

$$\Psi(u) = u \log u - (1+u)\log(1+u) \quad (17)$$

$$f(\mathbf{u}) = \frac{\nu p_{\mathbf{y}}(\mathbf{u})}{p_{\mathbf{x}}(\mathbf{u})} \qquad \mathrm{d}\mu(\mathbf{u}) = p_{\mathbf{x}}(\mathbf{u})\mathrm{d}\mathbf{u} \quad (18)$$

(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.

Appendix

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, $\hat{\boldsymbol{\theta}}$ satisfying

$$G(\mathbf{u}; \hat{\boldsymbol{\theta}}) = \frac{m}{n} \frac{p_{\mathbf{y}}(\mathbf{u})}{p_{\mathbf{x}}(\mathbf{u})}$$

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

$$J_n^{\mathrm{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\left(\sum_{i=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \sum_{i=1}^{m} \log\left[P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\right]\right)$$

without any normalization constraints.

Proof of Equation (4)

$$J_n^{\mathrm{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\left(\sum_{i=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \sum_{i=1}^{m} \log\left[P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\right]\right)$$

$$= \frac{1}{n}\sum_{t=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \frac{m}{n}\frac{1}{m}\sum_{t=1}^{m} \log\left[P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\right]$$

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

$$J^{\mathrm{NCE}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}}\left(\log P(C=1|\mathbf{x};\boldsymbol{\theta})\right) + \nu\mathbb{E}_{\mathbf{y}}\left(\log P(C=0|\mathbf{y};\boldsymbol{\theta})\right) \quad (19)$$

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

... we have

$$J^{\mathrm{NCE}}(\boldsymbol{\theta}) = -\,\mathbb{E}_{\mathbf{x}} \log(1 + G(\mathbf{x};\boldsymbol{\theta})) + \nu\mathbb{E}_{\mathbf{y}} \log G(\mathbf{y};\boldsymbol{\theta}) - \nu\mathbb{E}_{\mathbf{y}} \log\left(1 + G(\mathbf{y};\boldsymbol{\theta})\right) \quad (20)$$

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

$$\begin{aligned}\mathcal{J}^{\mathrm{NCE}}(G) &= -\,\mathbb{E}_{\mathbf{x}} \log(1 + G(\mathbf{x})) + \nu\mathbb{E}_{\mathbf{y}} \log G(\mathbf{y}) - \nu\mathbb{E}_{\mathbf{y}} \log\left(1 + G(\mathbf{y})\right) \\ &= -\int p_{\mathbf{x}}(\boldsymbol{\xi}) \log(1 + G(\boldsymbol{\xi})) d\boldsymbol{\xi} + \\ &\quad \nu \int p_{\mathbf{y}}(\boldsymbol{\xi}) \left(\log G(\boldsymbol{\xi}) - \log(1 + G(\boldsymbol{\xi}))\right)\end{aligned}$$

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

$$\frac{\delta\mathcal{J}^{\mathrm{NCE}}(G)}{\delta G} = -\frac{p_{\mathbf{x}}(\boldsymbol{\xi})}{1 + G(\boldsymbol{\xi})} + \nu p_{\mathbf{y}}(\boldsymbol{\xi}) \left(\frac{1}{G(\boldsymbol{\xi})} - \frac{1}{1 + G(\boldsymbol{\xi})}\right) \quad (21)$$

$$\frac{\delta \mathcal{J}^{\text{NCE}}(G)}{\delta G} = -\frac{p_{\mathbf{x}}(\boldsymbol{\xi})}{1 + G(\boldsymbol{\xi})} + \nu p_{\mathbf{y}}(\boldsymbol{\xi}) \left(\frac{1}{G(\boldsymbol{\xi})} - \frac{1}{1 + G(\boldsymbol{\xi})} \right) \tag{22}$$

$$= -\frac{p_{\mathbf{x}}(\boldsymbol{\xi})}{1 + G(\boldsymbol{\xi})} + \nu p_{\mathbf{y}}(\boldsymbol{\xi}) \frac{1}{G(\boldsymbol{\xi})(1 + G(\boldsymbol{\xi}))} \tag{23}$$

$$\stackrel{!}{=} 0 \tag{24}$$

We obtain

$$\frac{p_{\mathbf{x}}(\boldsymbol{\xi})}{1 + G^*(\boldsymbol{\xi})} = \nu p_{\mathbf{y}}(\boldsymbol{\xi}) \frac{1}{G^*(\boldsymbol{\xi})(1 + G^*(\boldsymbol{\xi}))} \tag{25}$$

$$G^*(\boldsymbol{\xi}) p_{\mathbf{x}}(\boldsymbol{\xi}) = \nu p_{\mathbf{y}}(\boldsymbol{\xi}) \tag{26}$$

$$G^*(\boldsymbol{\xi}) = \nu \frac{p_{\mathbf{y}}(\boldsymbol{\xi})}{p_{\mathbf{x}}(\boldsymbol{\xi})} \tag{27}$$

$$= \frac{m}{n} \frac{p_{\mathbf{y}}(\boldsymbol{\xi})}{p_{\mathbf{x}}(\boldsymbol{\xi})} \tag{28}$$

Evaluating $\partial^2 \mathcal{J}^{\text{NCE}} / \partial G^2$ at G^* shows that G^* is a maximizer.

Appendix

Maximizer of the NCE objective function

Noise-contrastive estimation as member of the Bregman framework

Proof

In noise-contrastive estimation, we maximize

$$J_n^{\mathrm{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\left(\sum_{i=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \sum_{i=1}^{m} \log\left[P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\right]\right)$$

Sample version of

$$J^{\mathrm{NCE}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}}\left(\log P(C=1|\mathbf{x};\boldsymbol{\theta})\right) + \nu\mathbb{E}_{\mathbf{y}}\left(\log P(C=0|\mathbf{y};\boldsymbol{\theta})\right)$$

With

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

$$J^{\mathrm{NCE}}(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x}}\log(1+G(\mathbf{x};\boldsymbol{\theta})) - \nu\mathbb{E}_{\mathbf{y}}\log(1+1/G(\mathbf{y};\boldsymbol{\theta})) \quad (29)$$

where $G(\mathbf{u};\boldsymbol{\theta}) = \frac{\nu p_{\mathbf{y}}(\mathbf{u})}{\phi(\mathbf{u};\boldsymbol{\theta})}$.

The general cost function in the Bregman framework is

$$J(g) = \int \left[-\Psi(g) + \Psi'(g)g - \Psi'(g)f \right] \mathrm{d}\mu \tag{30}$$

With

$$\Psi(g) = g\log(g) - (1+g)\log(1+g) \tag{31}$$

$$\Psi'(g) = \log(g) - \log(1+g) \tag{32}$$

we have

$$\begin{aligned} J(g) = \int \Big[& -g\log(g) + (1+g)\log(1+g) \\ & + \log(g)g - \log(1+g)g \\ & - \log(g)f + \log(1+g)f \Big] \mathrm{d}\mu \end{aligned} \tag{33}$$

$$J(g) = \int \Big[\log(1+g) - \log(g) f + \log(1+g) f \Big] \mathrm{d}\mu \tag{34}$$
$$= \int \Big[\log(1+g) + \log(1+1/g) f \Big] \mathrm{d}\mu \tag{35}$$

With

$$f(\mathbf{u}) = \frac{\nu p_{\mathbf{y}}(\mathbf{u})}{p_{\mathbf{x}}(\mathbf{u})} \qquad g(\mathbf{u}) = G(\mathbf{u}; \boldsymbol{\theta}) \qquad \mathrm{d}\mu(\mathbf{u}) = p_{\mathbf{x}}(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{36}$$

we have

$$J(G(.;\boldsymbol{\theta})) = \int p_{\mathbf{x}}(\mathbf{u}) \log(1 + G(\mathbf{u};\boldsymbol{\theta})) \mathrm{d}\mathbf{u}$$
$$+ \nu p_{\mathbf{y}}(\mathbf{u}) \log(1 + 1/G(\mathbf{u};\boldsymbol{\theta})) \mathrm{d}\mathbf{u} \tag{37}$$
$$= - J^{\mathrm{NCE}}(\boldsymbol{\theta}) \tag{38}$$