# Noise-Contrastive Estimation of Unnormalized Statistical Models

Kyunseon Lee

Published: February 2012

# Introduction

**This paper is about parameterized density estimation for cases where the partition function is hard to compute to normalize unnormalized models.**

# Introduction

**We want to estimate the joint probability density of variables from data.**

- **Density estimation**: the process of using data to approximate the joint probability distribution.
- We estimate the possible values of variables and how likely they are.
- Examples:
  - **Detecting network hack attempts**: If a new hack is not in training data, a classifier may miss it. By treating data outside the normal density as hack attempts, we can catch new types.
  - **Generating diverse images**: If we estimate an image distribution, we can sample and create images not in the data.

# Introduction

**We define density estimation as an unnormalized function model with a finite number of parameters.**

- **Unnormalized function estimation:**

$$p(x) \propto \exp(-E(x;\theta))$$

  - $x$: observed data
  - $p(x)$: data density function (unnormalized)
  - $E(x;\theta)$: potential function
  - $Z = \int \exp(-E(x))\, dx$: partition function (normalizer)
- **Normalized models:**
  - They must integrate to 1, so methods to compute density are limited.
  - Examples: Gaussian mixtures, kernel density estimation.

# Introduction

**We define density estimation as an unnormalized model with finite parameters.**

- **Problem:** To make an unnormalized function a density, we need a partition constant so the integral is 1. If it has no closed-form, it is very hard.
- **Existing ways to avoid partition constants:**
  - Approximate partition with importance sampling or score matching.
  - There is a trade-off between accuracy and computing cost (MCMC sampling, second derivatives).
- **This paper's solution:** Treat the partition constant as a parameter and learn it from data.
  - First-order gradient on $\theta$ costs $\mathcal{O}(d)$ for input dimension $d$.

## Key techniques: Importance sampling and score matching.

- **Importance sampling:**

$$Z = \int e^{-E(x)} dx = \int \frac{e^{-E(x)}}{q(x)} q(x) dx = \mathbb{E}_{x \sim q}\left[\frac{e^{-E(x)}}{q(x)}\right] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{e^{-E(x_i)}}{q(x_i)}$$

  - $q(x)$: easy-to-sample distribution, e.g. Gaussian or exponential.

- **Score matching:**

$$s_\theta(x) = \nabla_x \ln p(x;\theta) = -\nabla_x E(x;\theta) - \nabla_x \ln Z,$$

$$\mathrm{J}(\theta) = \tfrac{1}{2}\mathbb{E}_{data}[\|s_\theta(x) - s_{data}(x)\|^2] = \mathbb{E}_{data}[\tfrac{1}{2}\|\nabla_x E\|^2 + \Delta_x E]$$

  - $\nabla_x, \Delta_x$: first and second derivatives wrt $x$.
  - $J(\theta)$: objective to minimize.

# Introduction

**This paper's solution: Train the partition constant via logistic regression.**

- Train a logistic model to estimate density ratios of two distributions.
- Use it to estimate the ratio between an easy reference distribution and our unnormalized model plus constant.
- Treat the constant $z$ as a logistic regression parameter.

# Main Framework

**Convert to binary classification of data vs noise and use logistic regression to get density ratio.**

Posterior probability: $P(C=1 \mid u) = \dfrac{p_m(u)\pi_1}{p_m(u)\pi_1 + p_n(u)\pi_0} = \dfrac{1}{1+\nu\frac{p_n(u)}{p_m(u)}}$

$$h(u;\theta) = P(C=1 \mid u;\theta) = \frac{1}{1+\nu e^{-G(u;\theta)}},$$

$$G(u;\theta) = \ln p_m(u;\theta) - \ln p_n(u)$$

- $C = I(u \in X)$: data/noise label.
- $\theta$: logistic parameters, $\nu = T_d/T_n$: ratio of data to noise samples.
- $p_m, p_n$: densities of data and noise, $\pi_0 = P(C=0), \pi_1 = P(C=1)$.

# Main Framework

**Logistic regression and how ratio emerges:**

$$h(u;\theta) = \frac{1}{1+\nu e^{-G(u;\theta)}}, \quad G(u;\theta) = \ln p_m(u;\theta) - \ln p_n(u)$$

$$\ln \frac{h(u;\theta)}{1-h(u;\theta)} = G(u;\theta) \quad \Rightarrow \quad e^{G(u;\hat{\theta})} \approx \frac{p_m(u)}{p_n(u)}$$

- $X$: data, $Y$: noise, $u \in U = X \cup Y$ is a sample.
- $C = I(u \in X)$: data/noise.
- $\theta$: logistic parameters, $\nu = T_d/T_n$.
- $p_m, p_n$: densities of data and noise.

# Main Framework

**Log-likelihood for the binary problem:**

$$\ell(\theta) = \sum_{t=1}^{T_d+T_n} \left[ C_t \ln P(C_t = 1|u_t;\theta) + (1 - C_t) \ln P(C_t = 0|u_t;\theta) \right]$$
$$= \sum_{t=1}^{T_d} \ln[h(x_t;\theta)] + \sum_{t=1}^{T_n} \ln[1 - h(y_t;\theta)].$$

Minimizing negative log-likelihood makes $h(u)$ match $P(C = 1|u)$ and recovers the density ratio.

- $t$: index of data or noise sample.

# Main Framework

**Train the partition constant as part of the model.** Density model form:

$$\ln p_m(u;\theta) = \ln p_m^0(u;\alpha) + c, \quad \theta = (\alpha, c)$$

Logistic objective:

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

After training, we get the density ratio $G(u;\theta)$ without needing $Z$ explicitly.

- $p_m^0(u;\alpha)$: unnormalized model, $c$: normalization parameter.

**Theoretical guarantees.**

- Theorem 1 (Uniqueness): As $n \to \infty$, the maximizer of the NCE objective is unique and matches the true log-density.
- Theorem 2 (Consistency): Under regular conditions, the estimated parameters converge to the true ones.
- Theorem 3 (Asymptotic normality): As $n \to \infty$, the parameter error distribution is normal.
- Corollary 4: MSE converges to $\mathrm{tr}(\Sigma)/T_d$, so more noise samples improve stability.

**Further practical results:**

- Corollary 5: As $\nu \to \infty$, covariance $\Sigma$ is independent of noise choice.
- Corollary 6: As $\nu \to \infty$, NCE reaches the Cramér–Rao lower bound (optimal variance).
- Corollary 7: If noise matches data, $\Sigma$ is minimized (guides noise selection).

# Experiment

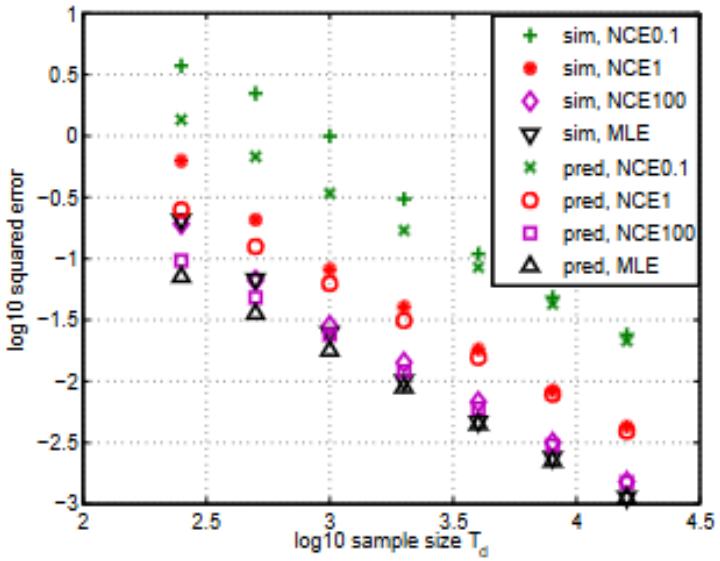
**Experiments:**

- Verify NCE properties (consistency, minimal variance).
- Compare with other unnormalized methods (error vs time).

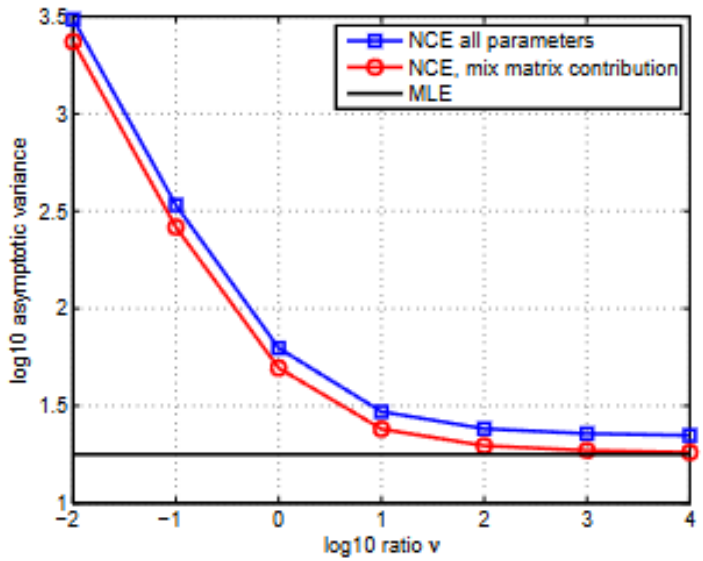
- Model: estimate parameters of unnormalized Gaussian.
- Simulated data: Gaussian and Laplace distributions.

# Experiment

## Verifying theoretical properties:




(a) Prediction of the MSE

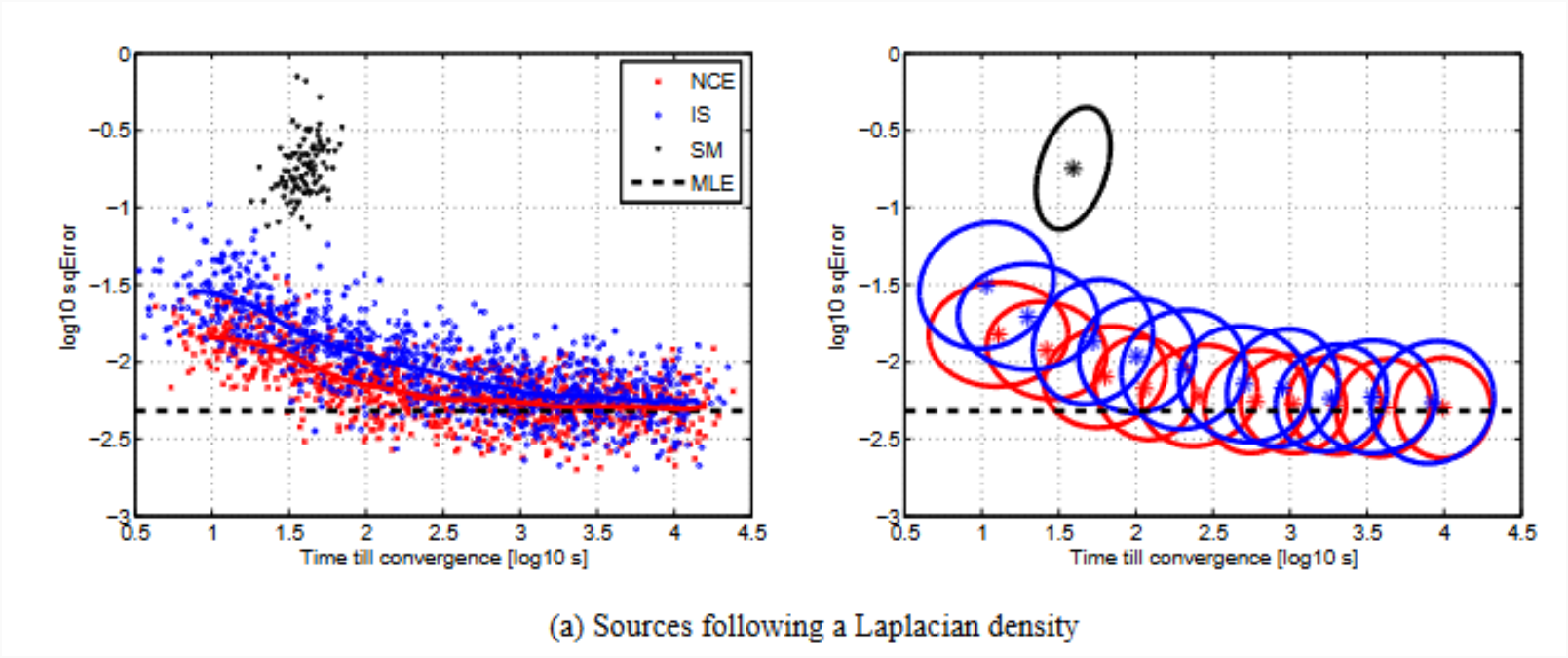



(b) Asymptotic behavior

- (a) Consistency: MSE decreases as sample size grows.
- (b) Minimal variance: noise ratio up gives asymptotic variance drop to MLE.

# Experiment

## Comparison with other methods (error vs time):




(a) Sources following a Laplacian density

- (a) NCE (red) has lowest error for given time.
- (b) Error distribution at fixed noise ratio.

# Conclusion

- NCE offers high flexibility with low compute cost, making it useful for high-dimensional data like images and videos.