Unsupervised learning by discriminating data from artificial noise

Michael Gutmann
Dept of Computer Science
University of Helsinki
michael.gutmann@helsinki.fi

Aapo Hyvärinen
Dept of Computer Science
Dept of Math and Statistics
University of Helsinki
aapo.hyvarinen@helsinki.fi

Abstract

Noise-contrastive estimation is a new estimation principle that we have developed for parameterized statistical models. The idea is to train a classifier to discriminate between the observed data and some artificially generated noise, using the model log-density function in a logistic regression function. It can be proven that this leads to a consistent (convergent) estimator of the parameters. The method is shown to directly work for models where the density function does not integrate to unity (unnormalized models). The normalization constant (partition function) can be estimated like any other parameter. We compare the method with other methods that can be used to estimate unnormalized models, including score matching, contrastive divergence, and maximum-likelihood where the correct normalization is estimated with importance sampling. Simulations show that noise-contrastive estimation offers the best trade-off between computational and statistical efficiency. The method is then applied to the modeling of natural images.

1 Introduction

Estimation of unnormalized parameterized statistical models is a computationally difficult problem. Here, we propose a new principle for estimating such models. Our method provides, at the same time, an interesting theoretical connection between unsupervised learning and supervised learning.

The basic estimation problem is formulated as follows. Assume we observe a sample of a random vector $\mathbf{x} \in \mathbb{R}^n$ which follows an unknown probability density function (pdf) $p_d(.)$. The data pdf $p_d(.)$ is modeled by a parameterized family of functions. We assume that $p_d(.)$ belongs to this family. The problem we consider here is how to estimate the parameters from the observed sample by maximizing some objective function.

Any solution to this estimation problem must yield a properly normalized density, that is a density which integrates to unity. This defines essentially a constraint in the optimization problem. The constraint is hard to fulfill because even numerical integration can easily become problematic when the data is high-dimensional. Examples of statistical models where the normalization constraint poses a problem can be found in Markov random fields (see e.g. [Koster2009]), products of experts and energy-based models [Hinton2002, Teh2004], and multilayer networks [Osindero2006, Koster2007].

Methods have thus been proposed which estimate models without explicitly computing integrals; the most recent ones are contrastive divergence [Hinton2002] and score matching [Hyvarinen2005c]. Here, we present a new estimation principle for unnormalized models which shows advantages over contrastive divergence or score matching. The basic idea is to estimate the model by learning to discriminate between the data and some artificially generated noise. The estimation principle thus relies on noise with which the data is contrasted, so that we will refer to the new method as "noise-contrastive estimation".

2 Noise-contrastive estimation

2.1 Definition of the estimator

Denote by $X = (\mathbf{x}(1), \ldots, \mathbf{x}(T))$ the observed data set, consisting of T observations of the data $\mathbf{x}$, and by $Y = (\mathbf{y}(1), \ldots, \mathbf{y}(T))$ an artificially generated data set of noise $\mathbf{y}$ with known distribution $p_n(.)$. Define a parametrized function $p_m(.;\theta)$ which models the data pdf $p_d(.)$. The estimator is then defined to be the argument which maximizes the objective function

$$J_T(\theta) = \frac{1}{2T} \sum_t \ln\left[h(\mathbf{x}(t);\theta)\right] + \ln\left[1 - h(\mathbf{y}(t);\theta)\right], \quad \text{where} \tag{1}$$

$$h(\mathbf{u};\theta) = \frac{1}{1+\exp[-G(\mathbf{u})]}, \qquad G(\mathbf{u};\theta) = \ln p_m(\mathbf{u};\theta) - \ln p_n(\mathbf{u}). \tag{2}$$

2.2 Important properties of the estimator

Nonparametric setting Assume the model pdf $p_m(\mathbf{u};\theta)$ can approximate any function and that $p_n(.)$ is nonzero whenever $p_d(.)$ is nonzero. Then, for large sample sizes T, the maximum of the objective function J_T is attained when the model pdf equals the data pdf.

Parametric setting For data generated according to the model, i.e. $\log p_d(\mathbf{u}) = \log p_m(\mathbf{u};\theta^*)$ for a certain θ^*, we can show that the estimator is statistically consistent.

The essential point in the above properties is that the maximization is performed without any normalization constraint for $p_m(.;\theta)$. This is in stark contrast to Maximum Likelihood Estimation (MLE), where $p_m(.;\theta)$ must integrate to unity. With our objective function, no such constraints are necessary. The maximizing pdf is found to have unit integral automatically.

2.3 Connection to supervised learning

The objective function in Eq. (1) occurs also in supervised learning. It is the log-likelihood in logistic regression with the nonlinearity $G(\mathbf{u};\theta)$. Our results show thus that unsupervised learning can be performed by supervised learning, in particular by logistic regression and classification. This connection provides us also with intuition of how the proposed estimator works: By discriminating, or comparing, between data X and noise Y, we are able to learn properties of the data, that is the statistical model. In less mathematical terms, the idea behind noise-contrastive estimation can thus be described by "learning by comparison".

2.4 Choice of the contrastive noise distribution

The noise distribution $p_n(.)$, which is used for contrast, is a design parameter. In practice, we would like to have a noise distribution which fulfills the following:

1. It is easy to sample from.
2. It has an analytical expression.
3. It leads to a small mean-squared error of the estimator. (This can be analyzed, but finding the optimum is difficult.)

Some examples which proved to be useful are the Gaussian or uniform distribution, a Gaussian mixture distribution, and an ICA distribution. Intuitively, the noise distribution should be as close to the data distribution as possible, because otherwise, the classification problem might be too easy and would not require the system to learn much about the structure of the data.

3 Simulations

3.1 Estimation of an ICA model

We illustrate noise-contrastive estimation with the learning of an ICA model [Hyvarinen2001], and compare its performance with other estimation methods, namely MLE, MLE with importance sam-

pling, contrastive divergence [Hinton2002], and score matching [Hyvarinen2005]. MLE gives the performance baseline. It can, however, only be used if an analytical expression for the partition function is available. The other methods, like noise-contrastive estimation, can be used to learn unnormalized models.

Data $\mathbf{x} \in \mathbb{R}^4$ is generated via the ICA model $\mathbf{x} = A\mathbf{s}$, where $A = (\mathbf{a}_1, \ldots, \mathbf{a}_4)$ is a 4×4 mixing matrix. All four independent sources in $\mathbf{s}$ follow a Laplacian density of unit variance and zero mean. The data log-pdf $\ln p_d(.)$ and the model log-pdf $\ln p_m(.;\theta)$ are

$$\ln p_d(\mathbf{x}) = -\sum_{i=1}^{4} \sqrt{2}\,|\mathbf{b}_i^\star \mathbf{x}| + c^* \qquad \text{and} \qquad \ln p_m(\mathbf{x};\theta) = -\sum_{i=1}^{4} \sqrt{2}|\mathbf{b}_i \mathbf{x}| + c, \tag{3}$$

respectively, where $\mathbf{b}_i^\star$ is the i-th row of the matrix $B^\star = A^{-1}$ and c^* is the negative logarithm of the normalization constant (partition function). It equals here $c^* = \ln|\det B^\star| - \ln 4$. The parameters $\theta \in \mathbb{R}^{17}$ are the row vectors $\mathbf{b}_i$ and c. In MLE, the correct value of c is known for every choice of $\mathbf{b}_i$.

Figure 1 shows the simulation results:

- The error in the parameters for the demixing matrix B decreases with increasing sample size T (red circles). The same holds for the error in the log-normalization constant c (red squares). This illustrates the consistency of noise-contrastive estimation as convergence in quadratic mean implies convergence in probability.
- Noise-contrastive estimation performs better than MLE where the normalization constant is calculated with importance sampling (markers in magenta).
- Contrastive divergence (green triangles) yields, for fixed sample sizes, more accurate results than noise-contrastive estimation with Gaussian contrastive-noise. The performance of noise-contrastive estimation improves, however, for Laplacian noise (results not shown).
- Noise-contrastive estimation requires about three times less computation time than the other methods to reach a required level of precision in the estimates. Among the methods for unnormalized models, noise-contrastive estimation offers thus the best trade-off between computational and statistical efficiency.
- Score matching (blue diamonds) is outperformed by the other methods. The reason is that we had to resort to an approximation of the Laplacian density for the estimation with score matching.

3.2 Estimating models of natural images

We use here noise-contrastive estimation to learn the statistical structure of natural images. Current multi-layer models for natural image (patches) are [Karklin2005, Osindero2006, Koster2007, Lee2008, Osindero2008]. A three-layer model is presented in [Osindero2008], while the other citations are two-layer models.

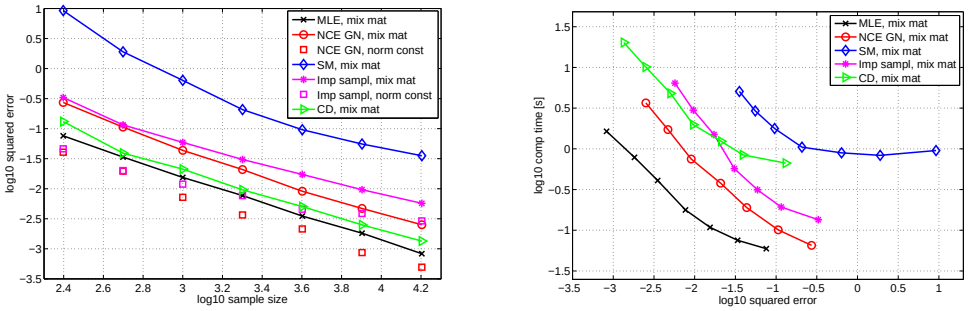
The two-layer model is

$$\log p_m(\mathbf{x};\theta) = \sum_n f_{\text{th}}\left(\ln\left[\mathbf{v}_n (W\mathbf{x})^2 + 1\right] + b_n\right) + c, \tag{4}$$

where f_{th} is a smooth thresholding function. The parameters θ of the model are the matrix W, the row vectors $\mathbf{v}_n$ and the bias terms b_n, which define the thresholds. The only constraint we are imposing is that the vectors $\mathbf{v}_n$ are limited to have only positive entries.

Figure 2 shows shows the estimation results. The first layer features $\mathbf{w}_i$ (rows of W) are Gabor-like. The second layer weights $\mathbf{v}_i$ pool together features of similar orientation and frequency, which are not necessarily centered at the same location. The results correspond to those reported in [Koster2007], as well as [Osindero2006].

Preliminary simulations where we learned a model which pools in a third layer together the outputs of the second layer, i.e. $\mathbf{z}^{(2)} = \ln\left[V(W\mathbf{x})^2 + 1\right]$, led to the emergence of cross-frequency and cross-orientation inhibition of the complex-cells.



(a) Estimation accuracy

(b) Estimation accuracy versus computation time

Figure 1: Performance comparison for the estimation of an ICA model. Black crosses show maximum-likelihood estimation (MLE), red circles and squares show noise-contrastive estimation with Gaussian noise (NCE GN), blue diamonds show score matching (SM), pink stars and squares show MLE with importance sampling, and green triangles show contrastive divergence (CD) where we used one cycle of Hamiltonian Monte Carlo with three leapfrog steps. For each sample size T, we created 500 random mixing matrices A, the figures show the median of the results. The computation time indicates the total time needed by each method to estimate the parameters. It included thus sampling of noise when needed. It was measured by matlab's built-in commands tic and toc.

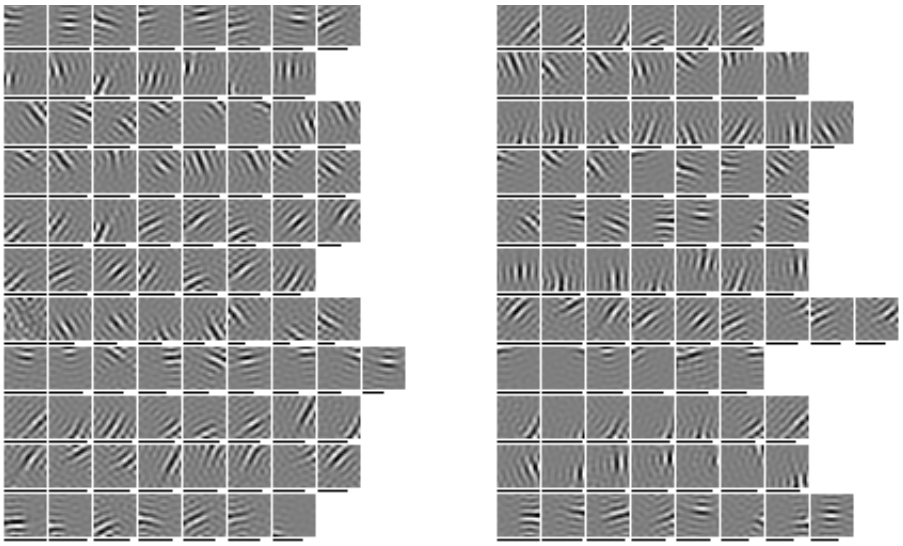


Figure 2: Complex cell-like pooling of similarly oriented Gabor filters. Each row shows one pooling pattern of simple cells, giving a complex cell. The black bar under each feature $\mathbf{w}_i$ (row of the matrix W) indicates the value of an element in the vector $\mathbf{v}_n$. Gaussian noise with the same covariance as natural images was used for the contrastive noise distribution $p_n(.)$. The patch size of the natural images was 20 pixels.