Noise-contrastive estimation of unnormalized statistical models, and its application to natural image statistics

Michael U. Gutmann MICHAEL.GUTMANN@HELSINKI.FI
Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
Department of Computer Science
Department of Mathematics and Statistics
Helsinki Institute for Information Technology HIIT
P.O. Box 68, FIN-00014 University of Helsinki
Finland

Abstract

Statistical models that do not integrate to one are called unnormalized models. In principle, any model can be normalized by computing the partition function. However, it is often impossible to obtain the partition function in closed form. Gibbs distributions, Markov and multi-layer networks are examples where analytical normalization is often impossible. Maximum likelihood estimation can then not be used without resorting to numerical approximations which are often computationally expensive. We propose here a new objective function for the estimation of both normalized and unnormalized models. The basic idea is to perform nonlinear logistic regression to discriminate between the observed data and some artificially generated noise. With this approach, the partition function can be estimated as any other parameter. We prove that the new estimation method leads to a consistent (convergent) estimator of the parameters. For large noise sample sizes, the new estimator is furthermore shown to behave like the maximum likelihood estimator. In the estimation of unnormalized models, there is a trade-off between statistical and computational performance. We show that the new method compares favorably to other estimation methods for unnormalized models. As an application to real data, we estimate large-scale parametric and nonparametric two-layer models of natural image statistics.

Keywords: Unnormalized models, partition function, computation, estimation, natural image statistics

1. Introduction

This paper is about parametric density estimation, where the general setup is as follows. A sample $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{T_d})$ of a random vector $\mathbf{x} \in \mathbb{R}^n$ is observed which follows an unknown probability density function (pdf) p_d. The data pdf p_d is modeled by a parameterized family of functions $\{p_m(.;\boldsymbol{\theta})\}_{\boldsymbol{\theta}}$ where $\boldsymbol{\theta}$ is a vector of parameters. It is commonly assumed that p_d belongs to this family. In other words, $p_d(.) = p_m(.;\boldsymbol{\theta}^\star)$ for some parameter $\boldsymbol{\theta}^\star$. The parametric density estimation problem is then about finding $\boldsymbol{\theta}^\star$ from the observed sample X. Any estimate $\hat{\boldsymbol{\theta}}$ must yield a properly normalized pdf $p_m(.;\hat{\boldsymbol{\theta}})$ which satisfies

$$\int p_m(\mathbf{u};\hat{\boldsymbol{\theta}})\mathrm{d}\mathbf{u} = 1, \qquad p_m(.;\hat{\boldsymbol{\theta}}) \geq 0. \tag{1}$$

These are two constraints in the estimation.

One approach to deal with the constraints is to specify the model such that Eq. (1) is fulfilled for all values of the parameters. The maximum likelihood principle can then be used for estimation. A model which is specified such that it satisfies the positivity constraint for all values of the parameters but does not integrate to one is called an unnormalized model. We denote here an unnormalized model, parameterized by some $\boldsymbol{\alpha}$, with $p_m^0(.;\boldsymbol{\alpha})$. Unnormalized models are easy to specify by taking, for example, the exponential transform of a suitable function. The partition function $Z(\boldsymbol{\alpha})$,

$$Z(\boldsymbol{\alpha}) = \int p_m^0(\mathbf{u};\boldsymbol{\alpha})\mathrm{d}\mathbf{u}, \tag{2}$$

allows to convert the unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ into a normalized one: $p_m^0(.;\boldsymbol{\alpha})/Z(\boldsymbol{\alpha})$ integrates to one for every value of $\boldsymbol{\alpha}$. Examples of distributions which are specified in that way are Gibbs distributions, Markov networks or multilayer networks. The function $\boldsymbol{\alpha} \mapsto Z(\boldsymbol{\alpha})$ is, however, defined via an integral. Unless $p_m^0(.;\boldsymbol{\alpha})$ has some particularly convenient form, the integral cannot be computed analytically so that the function $Z(\boldsymbol{\alpha})$ is not available in closed form. For low-dimensional problems, numerical integration can be used to approximate $Z(\boldsymbol{\alpha})$ to a very high accuracy but for high-dimensional problems this is computationally expensive. Our paper deals with density estimation in this case, that is, with density estimation when the computation of the partition function is problematic.

Several solutions for the estimation of unnormalized models which cannot be normalized in closed form have been suggested so far. Geyer (1994) proposed to approximate the calculation of the partition function by means of importance sampling and then to maximize the approximate log-likelihood (Monte-Carlo maximum likelihood). Approximation of the gradient of the log-likelihood led to another estimation method (contrastive divergence by Hinton, 2002). Estimation of the parameter $\boldsymbol{\alpha}$ directly from an unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ has been proposed by Hyvärinen (2005). This approach, called score matching, avoids the problematic integration to obtain the partition function altogether. All these methods need to balance the accuracy of the estimate and the time to compute the estimate.

In this paper[1], we propose a new estimation method for unnormalized models. The idea is to consider Z, or $c = \ln 1/Z$, not any more as a function of $\boldsymbol{\alpha}$ but as an additional parameter of the model. That is, we extend the unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ to include a normalizing parameter c and estimate

$$\ln p_m(.;\boldsymbol{\theta}) = \ln p_m^0(.;\boldsymbol{\alpha}) + c, \tag{3}$$

with parameter vector $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. The estimates $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\alpha}}, \hat{c})$ are then such that the unnormalized model $p_m^0(.;\hat{\boldsymbol{\alpha}})$ matches the shape of p_d, while $\hat{c}$ provides the proper scaling so that Eq. (1) holds. Unlike in the approach based on the partition function, we aim not at normalizing $p_m^0(.;\boldsymbol{\alpha})$ for all $\boldsymbol{\alpha}$ but only for $\hat{\boldsymbol{\alpha}}$. This avoids the problematic integration in the definition of the partition function $\boldsymbol{\alpha} \mapsto Z(\boldsymbol{\alpha})$. Such a separate estimation of shape and scale is, however, not possible for maximum likelihood estimation (MLE). The reason is that the likelihood can be made arbitrarily large by setting the normalizing parameter c to larger and larger numbers. The new estimation method is based on the maximization of a well

1. Preliminary versions were presented at AISTATS (Gutmann and Hyvärinen, 2010) and ICANN (Gutmann and Hyvärinen, 2009).

defined objective function. There are no constraints in the optimization so that powerful optimization techniques can be employed. The intuition behind the new objective function is to learn to classify between the observed data and some artificially generated noise. We approach thus the density estimation problem, which is an unsupervised learning problem, via supervised learning. The new method relies on noise which the data is contrasted to, so that we will refer to it as "noise-contrastive estimation".

The paper is organized in four main sections. In Section 2, we present noise-contrastive estimation and prove fundamental statistical properties such as consistency. This section is more of theoretical nature. In Section 3, we validate and illustrate the derived properties on artificial data. We use artificial data also to compare in Section 4 the new method to the aforementioned estimation methods with respect to their statistical and computational efficiency. In Section 5, we apply noise-contrastive estimation to real data. We estimate large-scale parametric and nonparametric two-layer models of natural images. This section is fairly independent from the other ones. The reader who wants to focus on natural image statistics may not need to go first through the previous sections. On the other hand, the reader whose interest is in estimation theory only can skip this section without missing pieces of the theory. Conclusions are drawn in Section 6.

2. Noise-contrastive estimation

This section presents the theory of noise-contrastive estimation. In Subsection 2.1, we motivate noise-contrastive estimation and relate it to supervised learning. The definition of noise-contrastive estimation is given in Subsection 2.2. In Subsection 2.3, we prove that the estimator is consistent for both normalized and unnormalized models, and derive its asymptotic distribution. In Subsection 2.4, we discuss practical aspects of the estimator and show that, in some limiting case, the estimator performs as well as MLE.

2.1 Density estimation by comparison

Density estimation is much about characterizing properties of the observed data X. A convenient way to describe properties is to describe them relative to the properties of some reference data Y. Let us assume that the reference (noise) data Y is a iid sample $(\mathbf{y}_1, \ldots \mathbf{y}_{T_n})$ of a random variable $\mathbf{y} \in \mathbb{R}^n$ with pdf p_n. A relative description of the data X is then given by the ratio p_d/p_n of the two density functions. If the reference distribution p_n is known, one can, of course, obtain p_d from the ratio p_d/p_n. In other words, if one knows the differences between X and Y, and also the properties of Y, one can deduce from the differences the properties of X.

Comparison between two data sets can be performed via classification: In order to discriminate between two data sets, the classifier needs to compare their properties. In the following, we show that training a classifier based on logistic regression provides a relative description of X in the form of an estimate of the ratio p_d/p_n.

Denote by $U = (\mathbf{u}_1, \ldots, \mathbf{u}_{T_d+T_n})$ the union of the two sets X and Y, and assign to each data point $\mathbf{u}_t$ a binary class label C_t: $C_t = 1$ if $\mathbf{u}_t \in X$ and $C_t = 0$ if $\mathbf{u}_t \in Y$. In logistic regression, the posterior probabilities of the classes given the data are estimated. As the pdf p_d of the data $\mathbf{x}$ is unknown, we model the class-conditional probability $p(.|C = 1)$ with

$p_m(.;\boldsymbol{\theta})$.[2] The class-conditional probability densities are thus

$$p(\mathbf{u}|C=1;\boldsymbol{\theta}) = p_m(\mathbf{u};\boldsymbol{\theta}), \qquad p(\mathbf{u}|C=0) = p_n(\mathbf{u}). \tag{4}$$

The prior probabilities are $P(C=1) = T_d/(T_d+T_n)$ and $P(C=0) = T_n/(T_d+T_n)$. The posterior probabilities for the classes are therefore

$$P(C=1|\mathbf{u};\boldsymbol{\theta}) = \frac{p_m(\mathbf{u};\boldsymbol{\theta})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \tag{5}$$

$$P(C=0|\mathbf{u};\boldsymbol{\theta}) = \frac{\nu p_n(\mathbf{u})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \tag{6}$$

where ν is the ratio $P(C=0)/P(C=1) = T_n/T_d$. In the following, we denote $P(C=1|\mathbf{u};\boldsymbol{\theta})$ by $h(\mathbf{u};\boldsymbol{\theta})$. Introducing the log-ratio $G(.;\boldsymbol{\theta})$ between $p_m(.;\boldsymbol{\theta})$ and p_n,

$$G(\mathbf{u};\boldsymbol{\theta}) = \ln p_m(\mathbf{u};\boldsymbol{\theta}) - \ln p_n(\mathbf{u}), \tag{7}$$

$h(\mathbf{u};\boldsymbol{\theta})$ can be written as

$$h(\mathbf{u};\boldsymbol{\theta}) = r_\nu\left(G(\mathbf{u};\boldsymbol{\theta})\right), \tag{8}$$

where

$$r_\nu(u) = \frac{1}{1+\nu\exp(-u)} \tag{9}$$

is the logistic function parameterized by ν.

The class labels C_t are assumed Bernoulli-distributed and independent. The conditional log-likelihood is given by

$$\ell(\boldsymbol{\theta}) = \sum_{t=1}^{T_d} C_t \ln P(C_t=1|\mathbf{u}_t;\boldsymbol{\theta}) + \sum_{t=1}^{T_n}(1-C_t)\ln P(C_t=0|\mathbf{u}_t;\boldsymbol{\theta}) \tag{10}$$

$$= \sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + \sum_{t=1}^{T_n} \ln\left[1-h(\mathbf{y}_t;\boldsymbol{\theta})\right]. \tag{11}$$

Optimizing $\ell(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ leads to an estimate $G(.;\hat{\boldsymbol{\theta}})$ of the log-ratio $\ln(p_d/p_n)$. That is, an approximate description of X relative to Y can be obtained by optimization of Eq. (11). The sign-flipped objective function, $-\ell(\boldsymbol{\theta})$, is also known as the cross-entropy error function (Bishop, 1995).

Thus, density estimation, which is an unsupervised learning problem, can be performed by logistic regression, that is, supervised learning. While this connection has been discussed earlier by Hastie et al. (2009), in the next sections, we will prove that even unnormalized models can be estimated with the same principle.

2. Classically, $p_m(.;\boldsymbol{\theta})$ would, in the context of this section, be a normalized pdf. In our paper, however, $\boldsymbol{\theta}$ may include a parameter for the normalization of the model.

2.2 Definition of the estimator

Given an unnormalized statistical model $p_m^0(.;\boldsymbol{\alpha})$, we include for normalization an additional parameter c into the model. That is, we define the model as

$$\ln p_m(.;\boldsymbol{\theta}) = \ln p_m^0(.;\boldsymbol{\alpha}) + c, \tag{12}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. Parameter c scales the unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ so that Eq. (1) can be fulfilled. After learning, $\hat{c}$ provides an estimate for $\ln 1/Z(\hat{\boldsymbol{\alpha}})$. If the initial model is normalized in the first place, no such inclusion of a normalizing parameter c is needed.

In line with the notation so far, we denote by $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{T_d})$ the observed data set that consists of T_d independent observations of $\mathbf{x} \in \mathbb{R}^n$. We denote by $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_{T_n})$ an artificially generated data set that consists of T_n independent observations of noise $\mathbf{y} \in \mathbb{R}^n$ with known distribution p_n. The estimator is defined to be the argument $\hat{\boldsymbol{\theta}}_T$ which maximizes

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\left\{\sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + \sum_{t=1}^{T_n} \ln\left[1 - h(\mathbf{y}_t;\boldsymbol{\theta})\right]\right\}, \tag{13}$$

where the nonlinearity $h(.;\boldsymbol{\theta})$ was defined in Eq. (8). The objective function J_T is, up to the division by T_d, the log-likelihood of Eq. (11). It can also be written as

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + \nu\frac{1}{T_n}\sum_{t=1}^{T_n} \ln\left[1 - h(\mathbf{y}_t;\boldsymbol{\theta})\right]. \tag{14}$$

Note that $h(.;\boldsymbol{\theta}) \in (0\ 1)$, where zero is obtained in the limit of $G(.;\boldsymbol{\theta}) \to -\infty$ and one in the limit of $G(.;\boldsymbol{\theta}) \to \infty$. Zero is an upper bound for J_T, which is reached if, for all t, $h(\mathbf{x}_t;\boldsymbol{\theta})$ and $h(\mathbf{y}_t;\boldsymbol{\theta})$ tend to one and zero, respectively. Therefore, the optimal parameter $\hat{\boldsymbol{\theta}}_T$ is such that $G(\mathbf{u}_t;\hat{\boldsymbol{\theta}}_T)$ is as large as possible for $\mathbf{u}_t \in X$ and as small as possible for $\mathbf{u}_t \in Y$. Intuitively, this means that logistic regression has learned to discriminate between th two sets as well as possible.

2.3 Properties of the estimator

We characterize here the behavior of the estimator $\hat{\boldsymbol{\theta}}_T$ for large sample sizes T_d. The weak law of large numbers shows that in that case, the objective function $J_T(\boldsymbol{\theta})$ converges in probability to J,

$$J(\boldsymbol{\theta}) = \mathrm{E}\left\{\ln\left[h(\mathbf{x};\boldsymbol{\theta})\right] + \nu \ln\left[1 - h(\mathbf{y};\boldsymbol{\theta})\right]\right\}. \tag{15}$$

Let us denote by $\tilde{J}$ the objective J seen as a function of $f_m(.) = \ln p_m(.;\boldsymbol{\theta})$,

$$\tilde{J}(f_m) \;=\; \mathrm{E}\left\{\ln\left[r_\nu\left(f_m(\mathbf{x}) - \ln p_n(\mathbf{x})\right)\right] + \nu \ln\left[1 - r_\nu\left(f_m(\mathbf{y}) - \ln p_n(\mathbf{y})\right)\right]\right\}. \tag{16}$$

We start the characterization of the estimator $\hat{\boldsymbol{\theta}}_T$ with a description of the optimization landscape for f_m. The following theorem shows that the data pdf p_d can be found by maximization of $\tilde{J}$, that is by learning a nonparametric classifier under the ideal situation of an infinite amount of data.

Theorem 1 (Nonparametric estimation) *$\tilde{J}$ attains a maximum at $f_m = \ln p_d$. There are no other extrema if the noise density p_n is chosen such it is nonzero whenever p_d is nonzero.*

The proof is given in Section A.2 of the appendix. A fundamental point in the theorem is that the maximization is performed without any normalization constraint for f_m. This is in stark contrast to MLE, where $\exp(f_m)$ must integrate to one. With our objective function, no such constraints are necessary. The maximizing pdf is found to have unit integral automatically.

The positivity condition for p_n in the theorem tells us that the data pdf p_d cannot be inferred where there are no contrastive noise samples for some relevant regions in the data space. Estimation of a pdf p_d which, for example, is nonzero only on the positive real line by means of a noise distribution p_n that has its support on the negative real line is thus impossible. The positivity condition can be easily fulfilled by taking, for example, a Gaussian as contrastive noise distribution.

In practice, the amount of data is limited and a finite number of parameters $\boldsymbol{\theta} \in \mathbb{R}^m$ specify $p_m(.;\boldsymbol{\theta})$. This has two consequences for any estimation method that is based on optimization: First, it restricts the space where the data pdf p_d is searched for. Second, it may introduce local maxima into the optimization landscape. For the characterization of the estimator in this situation, it is normally assumed that p_d follows the model, so that there is a $\boldsymbol{\theta}^\star$ with $p_d(.) = p_m(.;\boldsymbol{\theta}^\star)$. In the following, we make this assumption.

Our second theorem shows that $\hat{\boldsymbol{\theta}}_T$, the value of $\boldsymbol{\theta}$ which (globally) maximizes J_T, converges to $\boldsymbol{\theta}^\star$. The correct estimate of p_d is thus obtained as the sample size T_d increases. For unnormalized models, the conclusion of the theorem is that maximization of J_T leads to the correct estimates for both the parameters $\boldsymbol{\alpha}$ in the unnormalized pdf $p^0_m(.;\boldsymbol{\alpha})$ and the normalizing parameter c.

Theorem 2 (Consistency) *If conditions (a) to (c) are fulfilled then $\hat{\boldsymbol{\theta}}_T$ converges in probability to $\boldsymbol{\theta}^\star$, $\hat{\boldsymbol{\theta}}_T \xrightarrow{P} \boldsymbol{\theta}^\star$.*

(a) p_n is nonzero whenever p_d is nonzero

(b) $\sup_\theta |J_T(\boldsymbol{\theta}) - J(\boldsymbol{\theta})| \xrightarrow{P} 0$

(c) The matrix $\boldsymbol{\mathcal{I}}_\nu = \int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$ *has full rank, where*

$$\mathbf{g}(\mathbf{u}) = \nabla_{\boldsymbol{\theta}} \ln p_m(\mathbf{u};\boldsymbol{\theta})|_{\boldsymbol{\theta}^\star}, \qquad P_\nu(\mathbf{u}) = \frac{\nu p_n(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}.$$

The proof is given in Section A.3 of the appendix. Condition (a) is inherited from Theorem 1. Conditions (b) and (c) have their counterparts in MLE (see for example Wasserman, 2004, Theorem 9.13): We need in (b) uniform convergence in probability of J_T to J; in MLE, uniform convergence of the log-likelihood to the Kullback-Leibler distance is required likewise. Condition (c) assures that for large sample sizes, the objective function J_T becomes peaked enough around the true value $\boldsymbol{\theta}^\star$. This imposes a constraint on the model $p_m(.;\boldsymbol{\theta})$ via the vector $\mathbf{g}$. A similar constraint is required in MLE.

The next theorem describes the distribution of the estimation error $(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$ for large sample sizes. The proof is given in Section A.4 of the appendix.

Theorem 3 (Asymptotic normality) *$\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$ is asymptotically normal with mean zero and covariance matrix $\boldsymbol{\Sigma}$,*

$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{I}}_\nu^{-1} - \left(1 + \frac{1}{\nu}\right) \boldsymbol{\mathcal{I}}_\nu^{-1} \operatorname{E}(P_\nu \mathbf{g}) \operatorname{E}(P_\nu \mathbf{g})^T \boldsymbol{\mathcal{I}}_\nu^{-1},$$

where $\operatorname{E}(P_\nu \mathbf{g}) = \int P_\nu(\mathbf{u}) \mathbf{g}(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$.

From the distribution of $\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$, we can easily evaluate the asymptotic mean squared error (MSE) of the estimator.

Corollary 4 *For large sample sizes T_d, the mean squared error $\operatorname{E}\left(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2\right)$ equals $\operatorname{tr}(\boldsymbol{\Sigma})/T_d$.*

Proof Using that for any vector $\mathbf{v}$, $||\mathbf{v}||^2 = \operatorname{tr}(\mathbf{v}\mathbf{v}^T)$, the corollary follows directly from the definition of the MSE and Theorem 3. ■

2.4 Choosing the noise

Theorem 3 and Corollary 4 show that the noise distribution p_n and the ratio $\nu = T_n/T_d$ have an influence on the accuracy of the estimate $\hat{\boldsymbol{\theta}}_T$. A natural question to ask is what, from a statistical standpoint, the best choice of p_n and ν is. Our result on consistency (Theorem 2) also includes a technical constraint for p_n but this one is so mild that many distributions will satisfy it.

Theorem 2 shows that P_ν tends to one as the size T_n of the contrastive noise sample is made larger and larger. This implies that for large ν, the covariance matrix $\boldsymbol{\Sigma}$ does not depend on the choice of the noise distribution p_n. We have thus the following corollary.

Corollary 5 *For $\nu \to \infty$, $\boldsymbol{\Sigma}$ is independent of the choice of p_n and equals*

$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{I}}^{-1} - \boldsymbol{\mathcal{I}}^{-1} \operatorname{E}(\mathbf{g}) \operatorname{E}(\mathbf{g})^T \boldsymbol{\mathcal{I}}^{-1},$$

where $\operatorname{E}(\mathbf{g}) = \int \mathbf{g}(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$ *and* $\boldsymbol{\mathcal{I}} = \int \mathbf{g}(\mathbf{u}) \mathbf{g}(\mathbf{u})^T p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$.

The asymptotic distribution of the estimation error becomes then also independent from p_n. Hence, as the size of the contrastive-noise sample Y increases, the choice of the contrastive-noise distribution becomes less and less important. Moreover, for normalized models, we have the result that the estimation error has the same distribution as the estimation error in MLE.

Corollary 6 *For normalized models, noise-contrastive estimation is, in the limit of $\nu \to \infty$, asymptotically Fisher-efficient for all choices of p_n.*

Proof For normalized models, no normalizing parameter c is needed. In Corollary 5, the function $\mathbf{g}$ is then the score function as in MLE, and the matrix $\boldsymbol{\mathcal{I}}$ is the Fisher information matrix. Since the expectation $\operatorname{E}(\mathbf{g})$ is zero, the covariance matrix $\boldsymbol{\Sigma}$ is the inverse of the Fisher information matrix. ■

The corollaries above give one answer to the question on how to choose the noise distribution p_n and the ratio ν: If ν is made large enough, the actual choice of p_n is not of great importance. Note that this answer considers only estimation accuracy and ignores the computational load associated with the processing of noise. In Section 4, we will analyze the trade-off between estimation accuracy and computation time.

For any given ν, one could try to find the noise distribution with minimizes the MSE $\mathrm{E}\,||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2$. This minimization turns, however, out to be quite difficult. Intuitively, one could think that a good candidate for the noise distribution p_n is a distribution which is close to the data distribution p_d. If p_n is too different from p_d, the classification problem might be too easy and would not require the system to learn much about the structure of the data. This intuition is partly justified by the following theoretical result:

Corollary 7 *If* $p_n = p_d$ *then* $\boldsymbol{\Sigma} = \left(1 + \frac{1}{\nu}\right)\left(\boldsymbol{\mathcal{I}}^{-1} - \boldsymbol{\mathcal{I}}^{-1}\,\mathrm{E}(\mathbf{g})\,\mathrm{E}(\mathbf{g})^T\boldsymbol{\mathcal{I}}^{-1}\right)$.

Proof The corollary follows from Corollary 5 and the fact that P_ν equals $\nu/(1+\nu)$ for $p_n = p_d$. ■

For normalized models, we see that for $\nu = 1$, $\boldsymbol{\Sigma}$ is two times the inverse of the Fisher information matrix, and that for $\nu = 10$, the ratio is already down to 1.1. For a noise distribution that is close to the data distribution, we have thus even for moderate values of ν some guarantee that the MSE is reasonably close to the theoretical optimum.

To get estimates with a small estimation error, the foregoing discussion suggests the following

1. Choose noise for which an analytical expression for $\ln p_n$ is available.

2. Choose noise that can be sampled easily.

3. Choose noise that is in some aspect, for example with respect to its covariance structure, similar to the data.

4. Make the noise sample size as large as computationally possible.

Some examples for suitable noise distributions are Gaussian distributions, Gaussian mixture distributions, or ICA distributions. Uniform distributions are also suitable as long their support includes the support of the data distribution so that condition (a) in Theorem 2 holds.

3. Basic simulations to validate the theory

In this section[3], we validate and illustrate the theoretical properties of noise-contrastive estimation. In Subsection 3.1, we illustrate consistency and the idea of estimating the normalizing constant as an additional parameter. In Subsection 3.2, we go more into details in the illustration of noise-contrastive estimation.

3. Matlab code for this and the other sections can be downloaded from www.cs.helsinki.fi/michael.gutmann.

3.1 Gaussian distribution

We estimate here the parameters of a zero mean multivariate Gaussian. Its log-pdf is

$$\ln p_d(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\mathbf{\Lambda}^\star\mathbf{x} + c^\star, \qquad c^\star = \left(-\frac{1}{2}\ln|\det\mathbf{\Lambda}^\star| - \frac{n}{2}\ln(2\pi)\right), \tag{17}$$

where $c^\star$ does not depend on $\mathbf{x}$ and normalizes p_d to integrate to one. The information matrix $\mathbf{\Lambda}^\star$ is the inverse of the covariance matrix. It is thus a symmetric matrix. The dimension of $\mathbf{x}$ is here $n = 5$.

As we are mostly interested in the estimation of unnormalized models, we consider here the hypothetical situation where we want to estimate the model

$$\ln p^0_m(\mathbf{x};\boldsymbol{\alpha}) = -\frac{1}{2}\mathbf{x}^T\mathbf{\Lambda}\mathbf{x} \tag{18}$$

without knowing how to normalize it in closed form. This unnormalized model is a pairwise Markov network with quadratic node and edge potentials (Koller and Friedman, 2009). The parameters $\boldsymbol{\alpha} \in \mathbb{R}^{15}$ are the coefficients in the lower-triangular part of $\mathbf{\Lambda}$ as the matrix is symmetric. For noise-contrastive estimation, we add an additional normalizing parameter c to the model. The model that we estimate is thus

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \ln p^0_m(\mathbf{x};\boldsymbol{\alpha}) + c. \tag{19}$$

The model has 16 parameters given by $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. They are estimated by maximization of the objective function $J_T(\boldsymbol{\theta})$ in Eq. (13). We used a standard normal distribution for p_n. The optimization was performed with the nonlinear conjugate gradient algorithm of Rasmussen (2006).

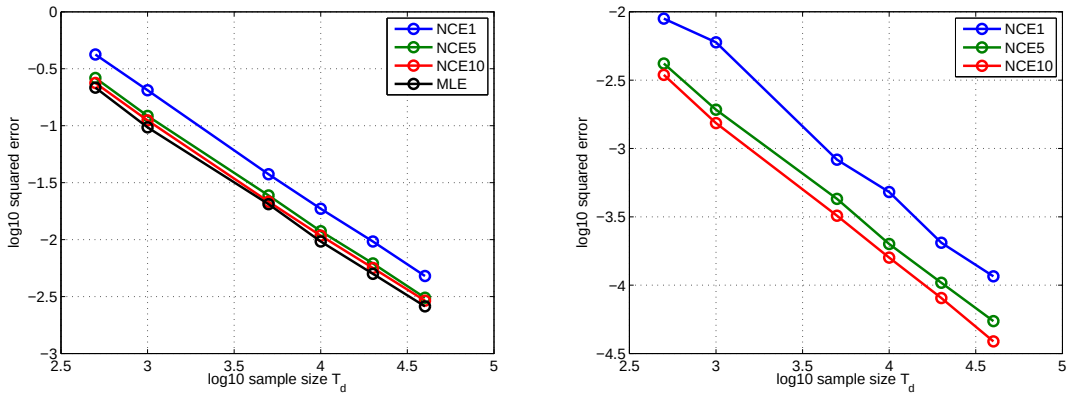
3.1.1 Results

Figure 1(a) and (b) show the mean squared error (MSE) for the parameters $\boldsymbol{\alpha}$, corresponding to the information matrix $\mathbf{\Lambda}$, and the normalizing parameter c, respectively. The results are an average over 500 estimation problems where the true information matrix $\mathbf{\Lambda}^\star$ was drawn at random. The MSE decays linearly on a log-log scale in function of the data sample size T_d. This illustrates our result of consistency of the estimator, stated as Theorem 2, as convergence in quadratic mean implies convergence in probability. The plots also show that taking more noise samples T_n than data samples T_d leads to more and more accurate estimates. The performance for noise-contrastive estimation with $\nu = T_n/T_d$ equal to one is shown in blue. For that value of ν, there is a clear difference compared to MLE (shown in black in Figure 1(a)). However, the accuracy of the estimate improves strongly for $\nu = 5$ (green) or $\nu = 10$ (red) where the performance is close to the performance of MLE.

3.2 ICA model

We estimate here the ICA model (see for example Hyvärinen et al., 2001b),

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{20}$$



(a) Information matrix (b) Normalizing parameter

Figure 1: Validation of the theory of noise-contrastive estimation: Estimation errors for a 5 dimensional Gaussian distribution. Figure (a) and (b) show the mean squared error for the information matrix $\mathbf{\Lambda}$ and the normalizing parameter c, respectively. The performance of noise-contrastive estimation approaches the performance of maximum likelihood estimation (MLE, black curve) as the ratio $\nu = T_n/T_d$ increases: $\nu = 1$ is shown in blue, $\nu = 5$ in green, and $\nu = 10$ in red. The curves are the median of the performance for 500 random information matrices with condition number less than 10.

where $\mathbf{x} \in \mathbb{R}^4$ and $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_4)$ is a 4×4 mixing matrix. The sources in the vector $\mathbf{s} \in \mathbb{R}^4$ are identically distributed and independent from each other so that the data log-pdf $\ln p_d$ is

$$\ln p_d(\mathbf{x}) = \sum_{i=1}^{n} f(\mathbf{b}_i^\star \mathbf{x}) + c^\star, \tag{21}$$

with $n = 4$. The i-th row of the matrix $\mathbf{B}^\star = \mathbf{A}^{-1}$ is denoted by $\mathbf{b}_i^\star$. We consider here Laplacian sources of unit variance and zero mean. The nonlinearity f and the constant $c^\star$, which normalizes p_d to integrate to one, are then given by

$$f(u) = -\sqrt{2}|u|, \qquad c^\star = \ln|\det \mathbf{B}^\star| - \frac{n}{2}\ln 2. \tag{22}$$

As in Subsection 3.1, we apply noise-contrastive estimation to the hypothetical situation where we want to estimate the unnormalized model

$$\ln p_m^0(\mathbf{x}; \boldsymbol{\alpha}) = \sum_{i=1}^{n} f(\mathbf{b}_i \mathbf{x}) \tag{23}$$

without knowing how to normalize it in closed form. The parameters $\boldsymbol{\alpha} \in \mathbb{R}^{16}$ are the row vectors $\mathbf{b}_i$. For noise-contrastive estimation, we add an additional normalizing parameter c and estimate the model

$$\ln p_m(\mathbf{x}; \boldsymbol{\theta}) = \ln p_m^0(\mathbf{x}; \boldsymbol{\alpha}) + c, \tag{24}$$

with parameters $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. As for the Gaussian case, we estimate $\boldsymbol{\theta}$ by maximization of $J_T(\boldsymbol{\theta})$ in Eq. (13) with the nonlinear conjugate gradient algorithm of Rasmussen (2006).

For the noise distribution p_n, we used a Gaussian distribution with covariance matrix given by the sample covariance of the data.

3.2.1 Results

In Figures 2 and 3, we illustrate the theorem on consistency (Theorem 2) and the theorem on the asymptotic distribution of the estimator (Theorem 3), as well as its corollaries. The results are averages over 500 random estimation problems.

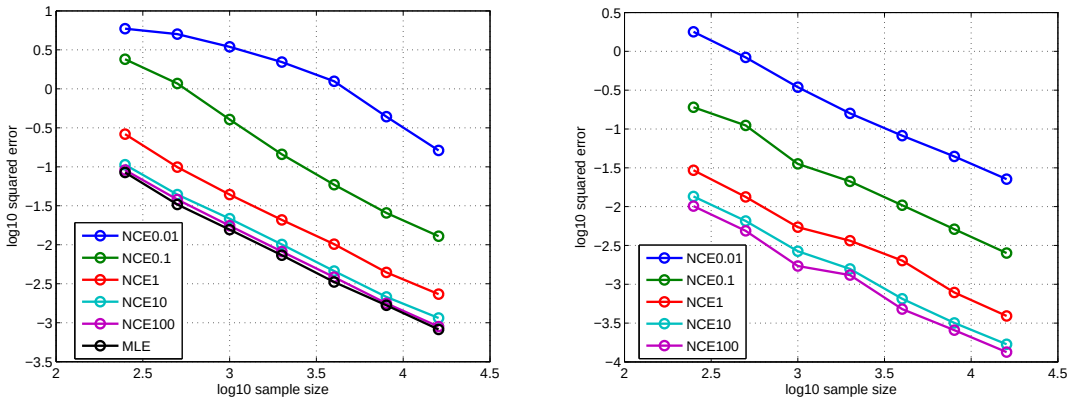
Figure 2(a) and (b) show the mean squared error (MSE) for the parameters $\boldsymbol{\alpha}$, corresponding to the mixing matrix, and the normalizing parameter c, respectively. As illustrated for the Gaussian case in Figure 1, this figure visualizes the consistency of noise-contrastive estimation. Furthermore, we see again that making $\nu = T_n/T_d$ larger leads to a reduction of the error. The reduction gets, however, smaller as ν increases. Changing ν from one (curve in red) to ten (shown in light blue) leads to a notable reduction of the MSE but increasing ν from ten to hundred (shown in magenta) leads only to a relatively small improvement.

In Figure 3(a), we test the theoretical prediction of Corollary 4 that, for large samples sizes T_d, the MSE decays like $\operatorname{tr} \boldsymbol{\Sigma}/T_d$. The covariance matrix $\boldsymbol{\Sigma}$ can be numerically evaluated according to its definition in Theorem 3. This allows for a prediction of the MSE that can be compared to the MSE obtained in the simulations. The figure shows that the MSE from the simulations (shown with circles as the markers) matches the prediction for large T_d (shown with squares). Furthermore, we see again that for large ν, the performance of noise-contrastive estimation is close to the performance of MLE. In other words, the trace of $\boldsymbol{\Sigma}$ is close to the trace of the Fisher information matrix. Note that for clarity, we only show the curves for $\nu \in \{0.1, 1, 100\}$. The curve for $\nu = 10$ was, as in Figure 2(a) and (b), very close to the curve for $\nu = 100$.

In Figure 3(b), we investigate how the value of $\operatorname{tr} \boldsymbol{\Sigma}$ (the asymptotic variance) depends on the ratio ν. Note that the covariance matrix $\boldsymbol{\Sigma}$ includes terms related to the parameter c. The Fisher information matrix includes, in contrast to $\boldsymbol{\Sigma}$, only terms related to the mixing matrix. For better comparison with MLE, we show thus in the figure the trace of $\boldsymbol{\Sigma}$ both with the contribution of the normalizing parameter c (blue curve) and without (red curve). For the latter case, the reduced trace of $\boldsymbol{\Sigma}$, which we will denote by $\operatorname{tr} \boldsymbol{\Sigma}_B$, approaches the trace of the Fisher information matrix. Corollary 6 stated that noise-contrastive estimation is asymptotically Fisher-efficient for large values of ν if the normalizing constant is not estimated. Here, we see that this result also approximately holds for our unnormalized model where the normalizing constant needs to be estimated.

Figure 3(c) gives further details to which extent the estimation becomes more difficult if the model is unnormalized. We computed numerically the asymptotic variance $\operatorname{tr} \tilde{\boldsymbol{\Sigma}}$ if the model is correctly normalized, and compared it to the asymptotic variance $\operatorname{tr} \boldsymbol{\Sigma}_B$ for the unnormalized model. The figure shows the distribution of the ratio $\operatorname{tr} \boldsymbol{\Sigma}_B / \operatorname{tr} \tilde{\boldsymbol{\Sigma}}$ for different values of ν. Interestingly, the ratio is almost equal to one for all tested values of ν. Hence, additional estimation of the normalizing constant does not really seem to have had a negative effect on the accuracy of the estimates for the mixing matrix.

In Corollary 7, we have considered the hypothetical case where the noise distribution p_n is the same as the data distribution p_d. In Figure 3(d), we plot the asymptotic variance in function of ν for that situation (green curve). For reference, we plot again the curve for



(a) Mixing matrix

(b) Normalizing parameter

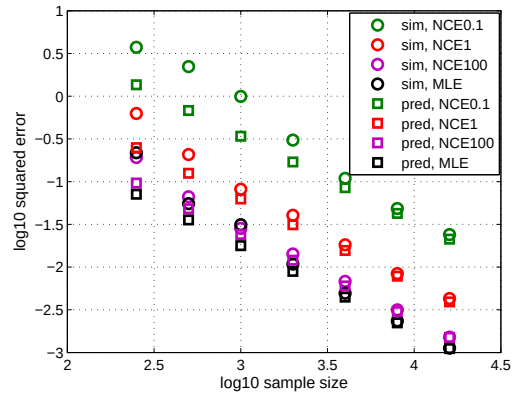
Figure 2: Validation of the theory of noise-contrastive estimation: Estimation errors for a 4×4 ICA model. Figure (a) and (b) show the mean squared error for the mixing matrix $\mathbf{B}$ and the normalizing parameter c, respectively. The performance of noise-contrastive estimation approaches the performance of maximum likelihood estimation (MLE, black curve) as the ratio $\nu = T_n/T_d$ increases: $\nu = 0.01$ is shown in blue, $\nu = 0.1$ in green, $\nu = 1$ in red, $\nu = 10$ in light blue, and $\nu = 100$ in magenta. The curves are the median of the performance for 500 random mixing matrices with condition number less than 10.

Gaussian contrastive noise (red, same curve as in Figure 3(b)). In both cases, we only show the asymptotic variance $\operatorname{tr} \mathbf{\Sigma}_B$ for the parameters that correspond to the mixing matrix. The asymptotic variance for $p_n = p_d$ is, for a given value of ν, always smaller than the asymptotic variance for the case where the noise is Gaussian. However, by choosing ν large enough for the case of Gaussian noise, it is possible to get estimates which are as accurate as those obtained in the hypothetical situation where p_n equals p_d. Moreover, for larger ν, the performance is the same for both cases: both converge to the performance of MLE.

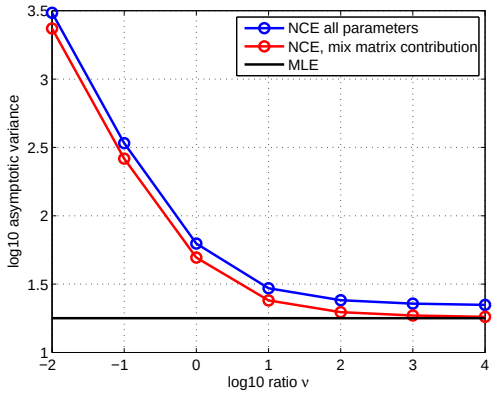
4. Investigating the trade-off between statistical and computational performance

We have seen that for large ratios $\nu = T_n/T_d$ of noise sample size to data sample size, the estimation error for noise-contrastive estimation behaves like the error in MLE. For large ν, however, the computational load becomes also heavier because more noise samples need to be processed. There is thus a trade-off between statistical and computational performance. Such a trade-off exists also in other estimation methods for unnormalized models. In this section, we investigate the trade-off in noise-contrastive estimation, and compare it to the trade-off in Monte-Carlo maximum likelihood estimation (Geyer, 1994), contrastive divergence (Hinton, 2002), and score matching (Hyvärinen, 2005).

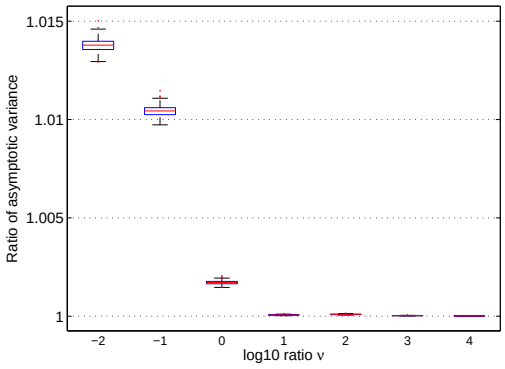
In Subsection 4.1, we comment on the data which we use in the comparison. In Subsection 4.2, we review the different estimation methods with focus on the trade-off between



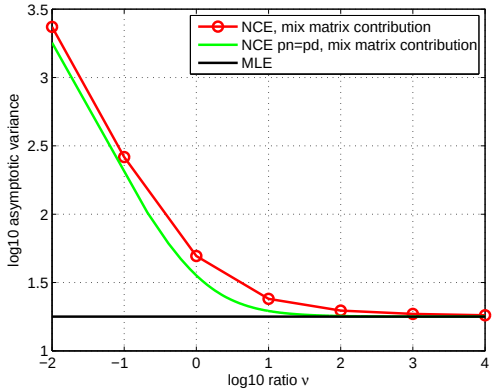
(a) Prediction of the MSE



(b) Asymptotic behavior



(c) Normalized vs unnormalized model



(d) Effect of the noise distribution

Figure 3: Validation of the theory of noise-contrastive estimation: Estimation error for large sample sizes. Figure (a) shows that Corollary 4 correctly predicts the MSE for large samples sizes T_d. Figure (b) shows the asymptotic variance $\operatorname{tr} \mathbf{\Sigma}$ in function of ν. Figure (c) shows a box plot of the ratio between the asymptotic variance when the model is unnormalized and the asymptotic variance when the model is normalized. Figure (d) compares noise-contrastive estimation with Gaussian noise to the hypothetical case where p_n equals the data distribution p_d. As in Figure 2, the curves in all figures but in Figure (c) are the median of the results for 500 random mixing matrices. The box plot in (c) shows the distribution for all the 500 matrices.

statistical and computational performance. Simulation results are presented in Subsection 4.3.

4.1 Data used in the comparison

For the comparison, we use artificial data which follows the ICA model in Eq. (20) with data log-pdf $\ln p_d$ given by Eq. (21). We set the dimension n to ten and use $T_d = 8000$ observations to estimate the parameters. In a first comparison, we assume Laplacian sources in the ICA model. The log-pdf $\ln p_d$ is then specified with Eq. (22). Note that the log-pdf has sharp peak around zero where it is not continuously differentiable. In a second comparison, we use sources that follow the smoother logistic density. The nonlinearity f and the log normalizing constant c^* in Eq. (21) are in that case

$$f(u) = -2 \ln \cosh \left(\frac{\pi}{2\sqrt{3}} u \right), \qquad c^* = \ln |\det \mathbf{B}^\star| + n \ln \left(\frac{\pi}{4\sqrt{3}} \right), \tag{25}$$

respectively. We are thus making the comparison for a relatively nonsmooth and smooth density. Both comparisons are based on 100 randomly chosen mixing matrices with condition number less than 10.

4.2 Estimation methods used in the comparison

We give here a short overview of the methods and comment on our implementation.

4.2.1 Noise-contrastive estimation

To estimate the parameters, we maximize J_T in Eq. (13). We use here a Gaussian noise density p_n with a covariance matrix equal to the sample covariance of the data. As before, J_T is maximized using the nonlinear conjugate gradient method of Rasmussen (2006). To map out the trade-off between statistical and computational performance, we measured the estimation error and the time needed to optimize J_T for $\nu \in \{1, 2, 5, 10, 20, 40, 80, 100\}$.

4.2.2 Monte-Carlo maximum likelihood estimation

For normalized models, an estimate for the parameters $\boldsymbol{\alpha}$ can be obtained by choosing them such that the probability of the observed data is maximized. This is done by maximization of

$$J_{\mathrm{MLE}}(\boldsymbol{\alpha}) = \frac{1}{T_d} \sum_{t=1}^{T_d} \ln p_m^0(\mathbf{x}_t; \boldsymbol{\alpha}) - \ln Z(\boldsymbol{\alpha}). \tag{26}$$

If no analytical expression for $Z(\boldsymbol{\alpha})$ is available, importance sampling can be used to numerically approximate $Z(\boldsymbol{\alpha})$ via its definition in Eq. (2)

$$Z(\boldsymbol{\alpha}) \approx \frac{1}{T_n} \sum_{t=1}^{T_n} \frac{p_m^0(\mathbf{n}_t; \boldsymbol{\alpha})}{p_{\mathrm{IS}}(\mathbf{n}_t)}. \tag{27}$$

The $\mathbf{n}_t$ are independent observations of "noise" with distribution p_{IS}. This procedure for calculating $Z(\boldsymbol{\alpha})$ in J_{MLE} gives rise to a new objective function J_{IS} for the parameters $\boldsymbol{\alpha}$,

$$J_{\mathrm{IS}}(\boldsymbol{\alpha}) = \frac{1}{T_d} \sum_{t=1}^{T_d} \ln p_m^0(\mathbf{x}_t; \boldsymbol{\alpha}) - \ln \left(\frac{1}{T_n} \sum_{t=1}^{T_n} \frac{p_m^0(\mathbf{n}_t; \boldsymbol{\alpha})}{p_{\mathrm{IS}}(\mathbf{n}_t)} \right). \tag{28}$$

This objective function, which uses importance sampling to approximate the log-likelihood, is known as Monte-Carlo maximum likelihood (Geyer, 1994). For the comparison, we maximized J_{IS} with the nonlinear conjugate gradient algorithm of Rasmussen (2006).

Like in noise-contrastive estimation, there is a trade-off between statistical performance and running time: The larger T_n gets, the better the approximation of the log-likelihood, and hence the more accurate the estimates, but the optimization of J_{IS} takes also more time. To map out the trade-off curve, we used the same values of $T_n = \nu T_d$ as in noise-contrastive estimation. Furthermore, we chose the same noise as in noise-contrastive estimation: $p_{\mathrm{IS}} = p_n$ and the $\mathbf{n}_t$ were the same as the noise samples $\mathbf{y}_t$ in noise-contrastive estimation. This choice was made to ease the comparison with noise-contrastive estimation. Monte-Carlo maximum likelihood is, of course, not limited to that particular choice of p_{IS}.

4.2.3 Contrastive divergence

If J_{MLE} is maximized with a steepest ascent algorithm, the update for the parameter $\boldsymbol{\alpha}$ is

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \mu_k \nabla_{\boldsymbol{\alpha}} J_{\mathrm{MLE}}(\boldsymbol{\alpha})|_{\boldsymbol{\alpha}_k}. \tag{29}$$

For the calculation of $\nabla_{\boldsymbol{\alpha}} J_{\mathrm{MLE}}$, the gradient of the log partition function $\ln Z(\boldsymbol{\alpha})$ is needed. Above, importance sampling was used to evaluate $\ln Z(\boldsymbol{\alpha})$ and its gradient $\nabla_{\boldsymbol{\alpha}} \ln Z(\boldsymbol{\alpha})$. The gradient of the log-partition function can, however, also be expressed as

$$\nabla_{\boldsymbol{\alpha}} \ln Z(\boldsymbol{\alpha}) = \frac{\nabla_{\boldsymbol{\alpha}} Z(\boldsymbol{\alpha})}{Z(\boldsymbol{\alpha})} = \int \frac{p_m^0(\mathbf{n}; \boldsymbol{\alpha})}{Z(\boldsymbol{\alpha})} \nabla_{\boldsymbol{\alpha}} \ln p_m^0(\mathbf{n}; \boldsymbol{\alpha}) \mathrm{d}\mathbf{n}. \tag{30}$$

If we had data $\mathbf{n}_t$ at hand which follows the normalized model density $p_m^0(.; \boldsymbol{\alpha})/Z(\boldsymbol{\alpha})$, the last equation could be evaluated by taking the sample average. In contrastive divergence (Hinton, 2002), data $\mathbf{n}_t$ is created which approximately follows $p_m^0(.; \boldsymbol{\alpha})/Z(\boldsymbol{\alpha})$ by means of Markov Chain Monte Carlo sampling. The gradient $\nabla_{\boldsymbol{\alpha}} J_{\mathrm{MLE}}$ can then, approximately, be evaluated, so that the parameters $\boldsymbol{\alpha}$ can be learned based on Eq. (29). Note that this update rule for $\boldsymbol{\alpha}$ is not directly optimizing a known objective function.

In our implementation, we used Hamiltonian Monte Carlo (MacKay, 2002) with a rejection ratio of 10% for the sampling (like in Teh et al., 2004; Ranzato and Hinton, 2010). There are then three tuning parameters for contrastive divergence: The number of Monte Carlo steps, the number of "leapfrog" steps in Hamiltonian Monte Carlo, and the choice of μ_k. The choice of the tuning parameters will affect the estimation error and the computation time. For the comparison, we used contrastive divergence with one and three Monte Carlo steps (denoted by CD1 and CD3 in the figures below), together with either three or twenty leapfrog steps. Ranzato and Hinton (2010) used CD1 with twenty leapfrog steps (below denoted by CD1 20), while Teh et al. (2004) used CD1 30 to estimate unnormalized models from natural image data. For the μ_k, we considered constant step sizes, as well as linearly and exponentially decaying step sizes.

4.2.4 Score matching

In score matching (Hyvärinen, 2005), the parameters $\boldsymbol{\alpha}$ are estimated by minimization of the cost function J_{SM},

$$J_{\mathrm{SM}}(\boldsymbol{\alpha}) = \frac{1}{T_d} \sum_{t=1}^{T_d} \sum_{i=1}^{n} \frac{1}{2} \Psi_i^2(\mathbf{x}_t; \boldsymbol{\alpha}) + \Psi_i'(\mathbf{x}_t; \boldsymbol{\alpha}). \tag{31}$$

The term $\Psi_i(\mathbf{x}; \boldsymbol{\alpha})$ is the derivative of the unnormalized model with respect to $\mathbf{x}(i)$, the i-th element of the vector $\mathbf{x}$,

$$\Psi_i(\mathbf{x}; \boldsymbol{\alpha}) = \frac{\partial \ln p_m^0(\mathbf{x}; \boldsymbol{\alpha})}{\partial \mathbf{x}(i)}. \tag{32}$$

The term $\Psi_i'(\mathbf{x}; \boldsymbol{\alpha})$ denotes the derivative of $\Psi_i(\mathbf{x}; \boldsymbol{\alpha})$ with respect to $\mathbf{x}(i)$. The presence of this derivative may make the objective function and its gradient algebraically rather complicated if a sophisticated model is estimated. For the ICA model with Laplacian sources, $\Psi_i(\mathbf{x}; \boldsymbol{\alpha})$ equals

$$\Psi_i(\mathbf{x}; \boldsymbol{\alpha}) = \sum_{j=1}^{n} -\sqrt{2}\mathrm{sign}(\mathbf{b}_j \mathbf{x}) B_{ji}. \tag{33}$$

The sign-function is not smooth enough to be used in score matching. We use therefore the approximation $\mathrm{sign}(u) \approx \tanh(10u)$ for its use in $\Psi_i(\mathbf{x}; \boldsymbol{\alpha})$ and its derivative. The optimization of J_{SM} is done by the nonlinear conjugate gradient algorithm of Rasmussen (2006). Note that, unlike the estimation methods considered above, score matching does not have a tuning parameter which controls the trade-off between statistical and computational performance. Moreover, score matching does not rely on sampling.

4.3 Results

For noise-contrastive estimation, Monte-Carlo maximum likelihood estimation and score matching, we are using the same optimization algorithm. We first compare these methods in terms of statistical and computational performance. In a second step, we compare the different variants of contrastive divergence. As third step, we include contrastive divergence into the comparison of the estimation methods.

4.3.1 Noise-contrastive estimation, Monte-Carlo MLE, and score matching

Figure 4 shows the comparison of noise-contrastive estimation (NCE in red), Monte-Carlo maximum likelihood (IS in blue) and score matching (SM in black). The left panels show the simulation results in form of "result points" where the x-coordinate represents the time till the algorithm converged and the y-coordinate the estimation error at the end of the optimization. For score matching, 100 result points, which correspond to 100 different random mixing matrices, are shown in each figure. For noise-contrastive estimation and Monte-Carlo maximum likelihood, we used eight different values of ν so that for these methods, each figure shows 800 result points. The panels on the right presents the simulation result in a more schematic way. For noise-contrastive estimation and Monte-Carlo maximum likelihood, the different ellipses represent the outcomes for different values of ν. Each ellipse

contains 90% of the result points. We can see that increasing ν reduces the estimation error but it also increases the running time. For score matching, there is no such trade-off.

Figure 4(a) shows that for Laplacian sources, noise-contrastive estimation outperforms the other methods in terms of the trade-off between statistical and computational performance. Score matching has a large estimation error because of the approximation of the involved nonlinearity. Monte-Carlo maximum likelihood seems to suffer from the problems of importance sampling when the data distribution has heavier tails than the noise (proposal) distribution (Wasserman, 2004, chap. 24). This is not the case for noise-contrastive estimation. The reason is that the nonlinearity $h(\mathbf{u};\boldsymbol{\theta})$ in the objective function in Eq. (13) is bounded even if data and noise distribution do not match well (see also Pihlaja et al., 2010).

For logistic sources, shown in Figure 4(b), noise-contrastive estimation and Monte-Carlo maximum likelihood perform equally. Score matching reaches its level of accuracy about 20 times faster than the other methods. Noise-contrastive estimation and Monte-Carlo maximum likelihood can, however, have a higher estimation accuracy than score matching if ν is large enough. Score matching can thus be considered to have a built-in trade-off between estimation performance and computation time: Computations are fast but the speed comes at the cost of not being able to reach an estimation accuracy as high as, for instance, noise-contrastive estimation.

4.3.2 Contrastive divergence

For contrastive divergence, we first assessed the influence of the stepsize μ_k. In these preliminary simulations, we used the first 10 of the 100 estimation problems and limited ourself to contrastive divergence with one Monte Carlo step and three leapfrog steps (CD1 3). Linear decay of the stepsize according to $\mu_k = 0.05(1-10^{-4}k)$ where $k = 1 \ldots 10^4$ led in 9/10 cases to the best performance both in time till convergence and resulting estimation error (results not shown[4]). Note that such an analysis to choose the best μ_k cannot be done in real applications since the true parameter values are not known. The choice of the step size must solely be based on experience, as well as trial and error. In Figure 5, we show, for that choice of μ_k, the estimation error in function of the running time of the algorithms. Note that we did not impose any stopping criterion. The algorithm had always converged before the maximal number of iterations was reached in the sense that there was no visible reduction of the estimation error any more. In real applications, where the true parameters are not known, assessing convergence in contrastive divergence is more difficult since the update rule of the parameters is not associated with a proper objective function. Figure 5 shows that for logistic sources CD3 3 (in dark green) leads to the best performance. In the case of Laplacian sources, there is a trade-off between estimation accuracy and computation time: CD1 3 (in cyan) converges fastest but has also the highest estimation error. CD1 20 (in light green), on the other hand, requires more time to converge but leads to a lower estimation error.

4. Results can be downloaded from www.cs.helsinki.fi/michael.gutmann.

4.3.3 All methods

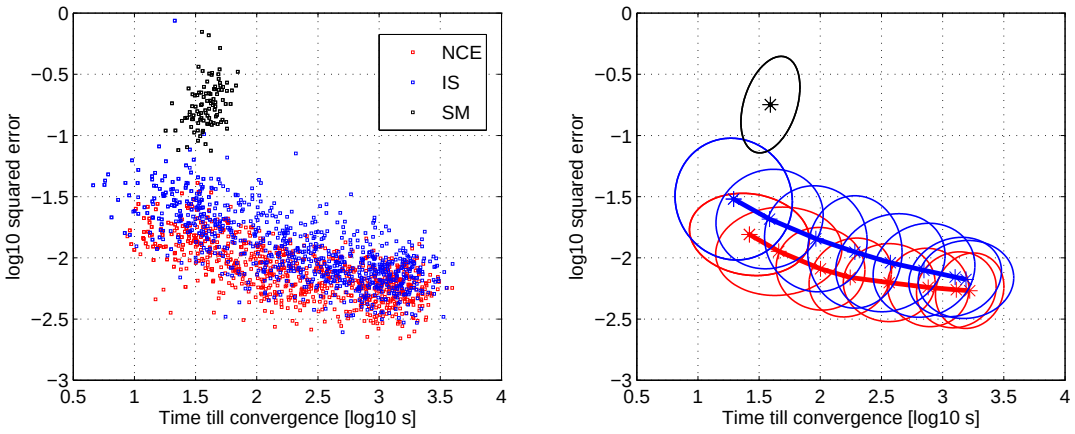
In Figure 6, we combine the results from Figures 4 and 5 to include contrastive divergence into the comparison. Figure 6 shows that, on average, the accuracy for the best-performing variants of contrastive divergence equals, after convergence, the accuracy for noise-contrastive estimation in the case of large ν: they are both close to the performance of MLE. We tested whether the collected data gives evidence that the distribution of the estimation errors for noise-contrastive estimation and contrastive divergence is different. In case of Laplacian sources, no such evidence was found for CD1 20 (in light green) and noise-contrastive estimation with $\nu = 100, 80$ (p-values in a two-sample Kolmogorov test are larger than 0.19). For logistic sources, the data gives no evidence for CD3 3 (in dark green) and noise-contrastive estimation for $\nu = 100, 80, 40$ (p-values are larger than 0.55, 0.3, and 0.095, respectively). Hence, for large running times, noise-contrastive estimation and contrastive divergence perform similarly. For smaller running times, however, Figure 6 shows that contrastive divergence is outperformed by noise-contrastive estimation; noise-contrastive estimation offers thus a better trade-off between statistical and computational performance.

The foregoing simulation results and discussion suggest that all estimation methods trade, in one form or the other, estimation accuracy against computation speed. In terms of this trade-off, 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 and the model algebraically not too complicated, score matching may, depending on the required estimation accuracy, be a better option.

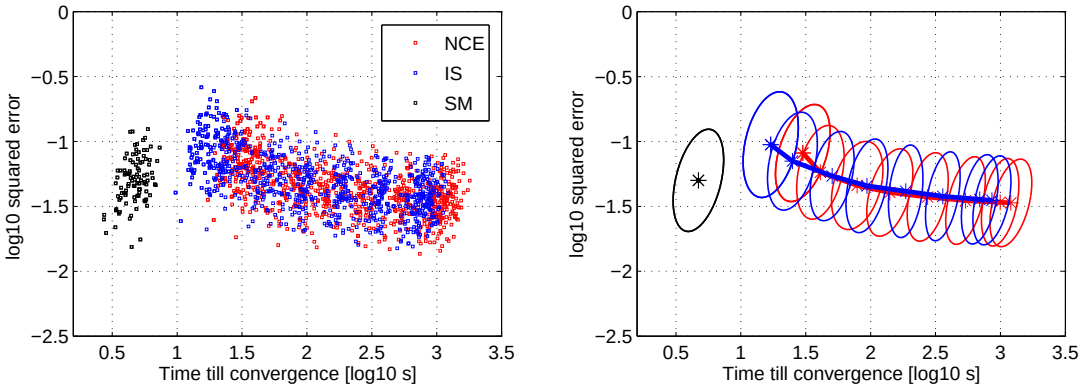
5. Simulations with natural images

In this section, we estimate models of natural images with our new estimation method. In the theory of noise-contrastive estimation, we have assumed that all variables can be observed. Noise-contrastive estimation can thus not be used for models that have latent variables which cannot be analytically integrated out. For examples of such models, see Olshausen and Field (1996); Hyvärinen et al. (2001a); Karklin and Lewicki (2005); Lücke and Sahani (2008); Osindero and Hinton (2008). Models which avoid latent variables, or where the latent variables can be integrated out, are often called energy-based models. Recent energy-based models are given by Osindero et al. (2006); Köster and Hyvärinen (2010); Ranzato and Hinton (2010). In these models, the value of the pdf is computed by passing the image through two processing layers ("two-layer models").

We start with giving some preliminaries in Subsection 5.1. In Subsection 5.2, we then comment on the settings of noise-contrastive estimation. Subsection 5.3 validates our new estimation method on a large-scale two-layer model with more than 50000 parameters. In Subsection 5.4, we present some nonparametric extensions. The different models are compared in Subsection 5.5.

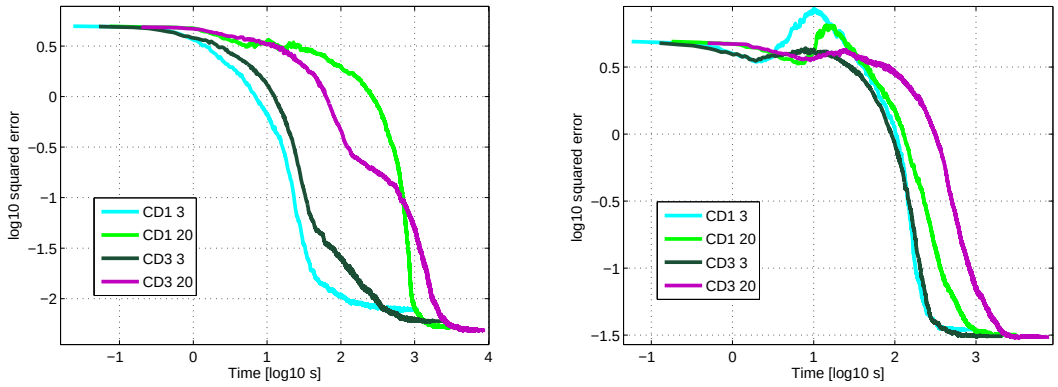


(a) Performance for sources following a Laplacian density



(b) Performance for sources following a logistic density

Figure 4: Trade-off between statistical and computational performance for noise-contrastive estimation (NCE, red), Monte-Carlo maximum likelihood (IS, blue) and score matching (SM, black). Each ellipse contains 90% of the result points. It was obtained by fitting a Gaussian to the distribution of the result points. The asterisks mark the center of each ellipse. For noise-contrastive estimation and Monte-Carlo maximum likelihood, the eight ellipses represent the outcomes for the eight different values of $\nu \in \{1, 2, 5, 10, 20, 40, 80, 100\}$. For an ICA model with Laplacian sources, NCE has the best trade-off between statistical and computational performance. For logistic sources, NCE and IS perform equally well. For moderate estimation accuracy, score matching outperforms the other estimation methods.



(a) Laplacian sources

(b) Logistic sources

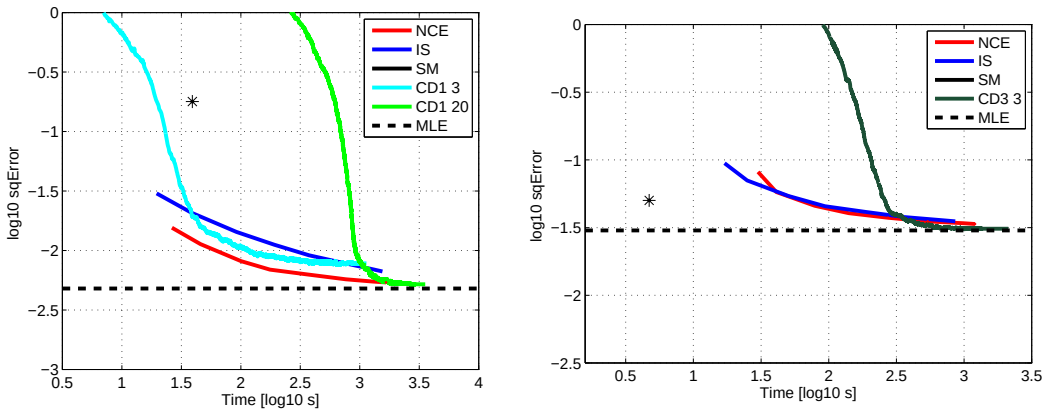
Figure 5: Trade-off between statistical and computational performance for contrastive divergence. While the algorithms are running, measurements of the estimation error at a given time are made. The time variable indicates thus the time since the algorithm was started. Note the difference to Figure 4 where the time indicates the time-till-convergence. The plots show the median performance over the 100 estimation problems. CDx y refers to contrastive divergence with x Monte Carlo steps, each using y leapfrog steps.

5.1 Data, preprocessing and modeling goal

Our basic data are a random sample of 25px $\times$ 25px image patches that we extracted from a subset of van Hateren's image database (van Hateren and van der Schaaf, 1998). The images in the subset showed wildlife scenes only.

Our goal is to model structure in the image patches. In line with this modeling goal, we ignore the scale and average value of each patch. The motivation is as follows: For the optimal visualization of image patches and their structure, each patch is usually rescaled such that all gray levels in the colormap are used. This implies that each image patch is essentially only defined up to a scaling constant and an offset. In other words, the local standard deviation (scale) of each image patch and the average pixel value (DC value) is arbitrary. We show next that, as consequence, the data can be constrained to lie on a sphere.

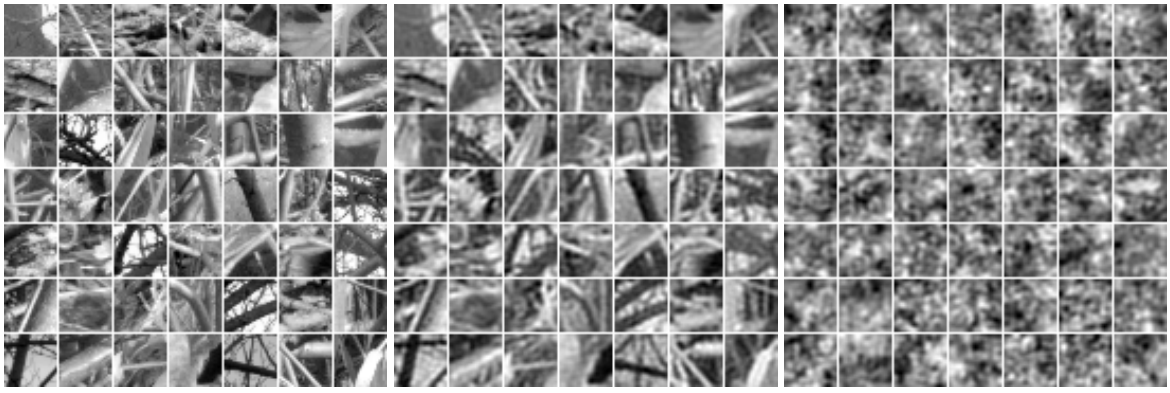
We organize the image patches as $D = 625$ dimensional vectors. The $n \leq D$ largest eigenvalues d_k and the corresponding eigenvectors $\mathbf{e}_k$ of the sample covariance matrix of the image patches yield an orthogonal basis $\sqrt{d_k}\mathbf{e}_k$ for a n dimensional subspace of $\mathbb{R}^D$. The matrix with column vectors $\mathbf{e}_k$ will be denoted by $\mathbf{E}$, and the diagonal matrix with the corresponding eigenvalues by $\mathbf{D}$. Denote by $\mathbf{x} = (x_1, \ldots x_n)^T$ the coordinates with respect to the above basis. Representing an image patch by its coordinate vector $\mathbf{x}$ corresponds to dimension reduction and whitening. In the following, we use $n = 160$ dimensions so that the coordinate vectors account for 93% of the variance of the image patches. In order to fix the scale and the DC value of each image patch, we rescale each patch such that the



(a) Laplacian sources

(b) Logistic sources

Figure 6: Combining the results from Figures 4 and 5. Note that the time variable in the two figures had slightly different definitions which we are confounding in this figure. For noise-contrastive estimation (NCE, red) and Monte-Carlo maximum likelihood (IS, blue), the bold lines show the average trade-off curves of Figure 4. The black asterix shows the average performance of score matching (SM). For contrastive divergence (CD1 3, cyan; CD1 20, light green; CD3 3, dark green), the bold curves show, as in Figure 5, the median performance. The median of the estimation error obtained in maximum likelihood is shown as a dashed line. For Laplacian sources, noise-contrastive estimation gives the best trade-off. For logistic sources, the best trade-off curves are given by noise-contrastive estimation and Monte-Carlo maximum likelihood. Score matching is, however, the best option if a particularly high estimation accuracy is not required.



(a) Image patches (b) After dimension reduction (c) Noise

Figure 7: (a) Image patches of size 25px × 25px. (b) The image patches after dimension reduction from $D = 625$ to $n = 160$ dimensions. These are examples of the image patches denoted by $\mathbf{I}$ in Eq. (35). Their coordinate vectors $\mathbf{x}$ lie on a $n = 160$ dimensional sphere $\mathbb{S}^n$. (c) Noise images which are obtained via Eq. (35) if the coordinates are uniformly distributed on $\mathbb{S}^n$. Comparison with Figure (b) shows that the coordinate vectors $\mathbf{x}$ for natural images are clearly not uniformly distributed on the sphere. In the next sections, we model their distribution.

coordinates fulfill

$$\sum_{k=1}^{n} x_k = 0, \qquad \frac{1}{n-1}\sum_{k=}^{n} x_k^2 = 1. \tag{34}$$

The vector $\mathbf{x}$ lies on a n dimensional sphere $\mathbb{S}^n$ centered at zero and with radius $\sqrt{n-1}$. In the following sections, we model the distribution of $\mathbf{x}$. We will denote a rescaled and dimension reduced image patch by $\mathbf{I}$,

$$\mathbf{I} = \mathbf{V}^{-}\mathbf{x}, \qquad \mathbf{V}^{-} = \mathbf{E}\mathbf{D}^{1/2}, \tag{35}$$

where the vector $\mathbf{x} \in \mathbb{S}^n$ fulfills the condition in Eq. (34). Notice that the matrices $\mathbf{E}$ and $\mathbf{D}$ are defined based on the images patches before the rescaling. The matrix $\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^T$ is thus only a whitening matrix for the data before rescaling, and not for $\mathbf{I}$. Examples of image patches before and after dimension reduction are shown in Figure 7. In line with the above discussion, all image patches in this figure were rescaled to use the full colormap.

5.2 Settings for noise-contrastive estimation

Matlab code for the simulations is available from the authors' homepage so that our description here will not be exhaustive. All the models are estimated with noise-contrastive estimation. We learn the parameters by optimization of the objective J_T in Eq. (13). Two-layer models are estimated by first estimating one-layer models. The learned parameters are used for initialization in the estimation of the complete two-layer model.

For the contrastive noise distribution p_n, we take a uniform distribution on the n dimensional sphere $\mathbb{S}^n$ on which $\mathbf{x}$ is defined. Examples of image patches with coordinates

following p_n are shown in Figure 7 (c). Samples from p_n can easily be created by sampling from a standard normal distribution and projecting then each sample onto the sphere. The pdf p_n is the inverse of the surface area of the n dimensional sphere with radius $\sqrt{n-1}$.[5] Since p_n is a constant, the log-ratio $G(.;\boldsymbol{\theta})$ in Eq. (7) is up to an additive constant equal to $\ln p_m(.;\boldsymbol{\theta})$,

$$G(.;\boldsymbol{\theta}) = \ln p_m(.;\boldsymbol{\theta}) + \text{constant}. \tag{36}$$

As pointed out in Subsection 2.2, $\boldsymbol{\theta}$ evolves in the maximization of J_T such that $G(\mathbf{u};\hat{\boldsymbol{\theta}}_T)$ is as large as possible for $\mathbf{u} \in X$ (natural images) but as small as possible for $\mathbf{u} \in Y$ (noise). For uniform noise, the same must thus also hold for $\ln p_m(\mathbf{u};\hat{\boldsymbol{\theta}}_T)$. This observation will be a useful guiding tool for the interpretation of the models below.

The factor $\nu = T_n/T_d$ was set to 10 and T_d was 160000. We found that an iterative optimization procedure where we separate the data into "minibatches" and optimize J_T for increasingly larger values of ν reduced computation time. The size of the minibatch is still kept rather larger, for example 80000 in the simulation of the next subsection. An analysis of such an optimization procedure is made in appendix B. The optimization is done with the nonlinear conjugate gradient method of Rasmussen (2006).

5.3 Two-layer model with thresholding nonlinearities

The first model that we consider is

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{n} f(y_k; a_k, b_k) + c, \qquad y_k = \sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2, \tag{37}$$

where f is a smooth, compressive thresholding function that is parameterized by a_k and b_k, see Figure 8. The parameters $\boldsymbol{\theta}$ are the second-layer weights $Q_{ki} \geq 0$, the first-layer weights $\mathbf{w}_i \in \mathbb{R}^n$, the normalizing parameter $c \in \mathbb{R}$, as well as $a_k > 0$ and $b_k \in \mathbb{R}$ for the nonlinearity f. The definition of y_k shows that multiplying Q_{ki} by a factor γ_i^2 and $\mathbf{w}_i$ at the same time by the factor $1/\gamma_i$ does not change the value of y_k. There is thus some ambiguity in the parameterization which could be resolved by imposing a norm constraint either on the $\mathbf{w}_i$ or on the columns of the matrix $\mathbf{Q}$ formed by the weights Q_{ki}. It turned out that for the estimation of the model such constraints were not necessary. For the visualization and interpretation of the results, we chose γ_i such that all the $\mathbf{w}_i$ had norm one.

We comment here on the motivation for the family of nonlinearities f shown in Figure 8. The motivation for the thresholding property is that, in line with Subsection 5.2, $\ln p_m(.;\boldsymbol{\theta})$ can easily be made large for natural images and small for noise. The y_k must just be above the thresholds for natural image input and below for noise. This occurs when the vectors $\mathbf{w}_i$ detect features (regularities) in the input which are peculiar to natural images, and when, in turn, the second-layer weights Q_{ki} detect characteristic regularities in the squared first-layer feature outputs $\mathbf{w}_i^T\mathbf{x}$. The squaring implements the assumption that the regularities in $\mathbf{x}$ and $(-\mathbf{x})$ are the same so that the pdf of $\mathbf{x}$ should be an even function of the $\mathbf{w}_i^T\mathbf{x}$. Another property of the nonlinearity is its compressive log-like behavior for inputs above the threshold. The motivation for this is robustness against outliers.

A model like in Eq. (37) has been studied before by Osindero et al. (2006); Köster and Hyvärinen (2010). There are, however, a number of important differences. The main

5. $\ln p_n = -\ln(2) - \frac{n}{2}\ln(\pi) - \frac{n-1}{2}\ln(n-1) + \ln\Gamma\left(\frac{n}{2}\right)$, where Γ is the gamma function.

difference is that in our case $\mathbf{x}$ lies on a sphere while in the cited work, $\mathbf{x}$ was defined in whole space $\mathbb{R}^n$. This difference allows us to use nonlinearities that do not decay asymptotically to $-\infty$ which is necessary if $\mathbf{x}$ is defined in $\mathbb{R}^n$. A smaller difference is that we do not need to impose norm constraints to facilitate the learning of the parameters.

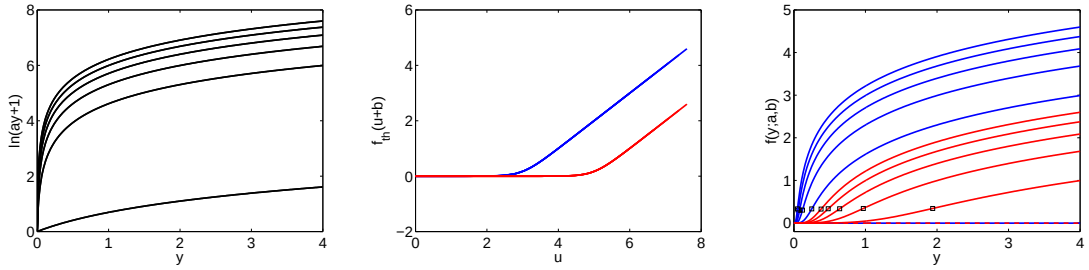
Results For the visualization of the first-layer feature detectors $\mathbf{w}_i$, note that the inner product $\mathbf{w}_i^T\mathbf{x}$ is equal to $(\mathbf{w}_i^T\mathbf{V})\mathbf{I} = \tilde{\mathbf{w}}_i^T\mathbf{I}$. The $\mathbf{w}_i \in \mathbb{R}^n$ are coordinate vectors with respect to the basis defined in Subsection 5.1, while the $\tilde{\mathbf{w}}_i \in \mathbb{R}^n$ are the coordinate vectors with respect to the pixel basis. The latter vectors can thus be visualized as images. This is done in Figure 9(a). Another way to visualize the first-layer feature detectors $\mathbf{w}_i$ is to show the images which yield the largest feature output while satisfying the constraints in Eq. (34). These optimal stimuli are proportional to $\mathbf{V}^-(\mathbf{w}_i - < \mathbf{w}_i >)$, where $< \mathbf{w}_i > \in \mathbb{R}$ is the average value of the elements in the vector $\mathbf{w}_i$. They are shown in Figure 9(b). Both visualizations show that the first layer computes "Gabor-like" features, which is in line with previous research on natural image statistics.

Figure 10 shows a random selection of the learned second-layer weights Q_{ik}. Figure 10(a) shows that the weights are extremely sparse: only few of them are nonzero. Note that the sparseness of the second-layer weights was obtained without any norm constraints or regularization on $\mathbf{Q}$. From Figure 10(b), we see that the learned second-layer weights Q_{ik} are such that they combine first-layer features of similar orientation, which are centered at nearby locations ("complex cells"). The same figure shows also a condensed representation of the feature detectors using icons. This form of visualization is used in Figure 11 to visualize all the second-layer feature detectors.

Figure 12(a) shows the learned nonlinearities $f(.; a_k, b_k)$. Note that we incorporated the learned normalizing parameter c as an offset c/n for each nonlinearity. The learned thresholding is similar for feature outputs of mid- and high-frequency feature detectors (black curves). For the feature detectors tuned to low frequencies, the thresholds tend to be smaller (green curves). In the following, we comment on the shape of the thresholding curves shown in black. For values of y smaller than two, the nonlinearities are convex (red rectangle); they show a squashing behavior. Looking at the distribution of the second-layer outputs y_k in Figure 10(b), we see that, up for the case of y_k close to zero, it is more likely that noise rather than natural images was the input when the second-layer feature outputs y_k are smaller than 2. The squashing nonlinearities map thus more often noise input to small values than natural images so that $\ln p_m(\mathbf{u}; \hat{\boldsymbol{\theta}}_T)$ tends to be larger when $\mathbf{u}$ is a natural image than when it is noise (see Subsection 5.2). One could, however, think that the thresholding nonlinearities are suboptimal because they ignore the fact that natural images lead, compared to noise, rather often to y_k close to zero. An optimal nonlinearity should, unlike the thresholding nonlinearities, assign a large value to both large and small y_k while mapping intermediate values of y_k to small numbers. The next subsection shows that we can learn such a nonlinearity from the data.

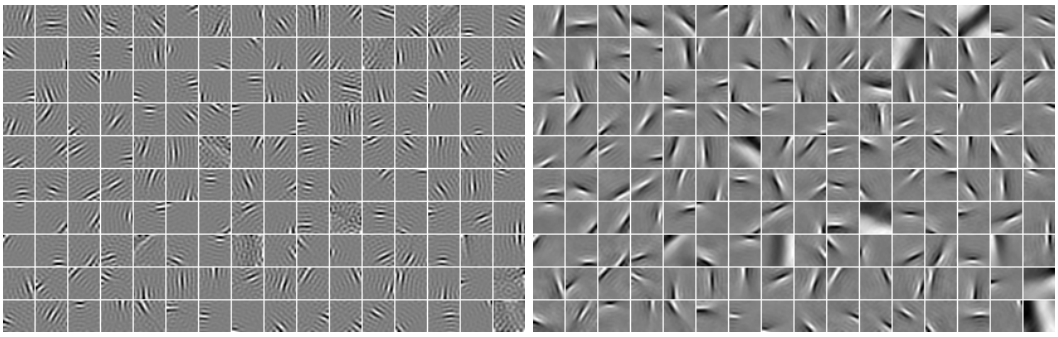
5.4 Two-layer model with flexible family of nonlinearities

In the previous subsection, the family of nonlinearities f in Eq. (37) was rather limited. Here, we look for f in the larger family of cubic splines where we consider the location of the knots to be fixed (regression splines represented with B-spline basis functions, see for



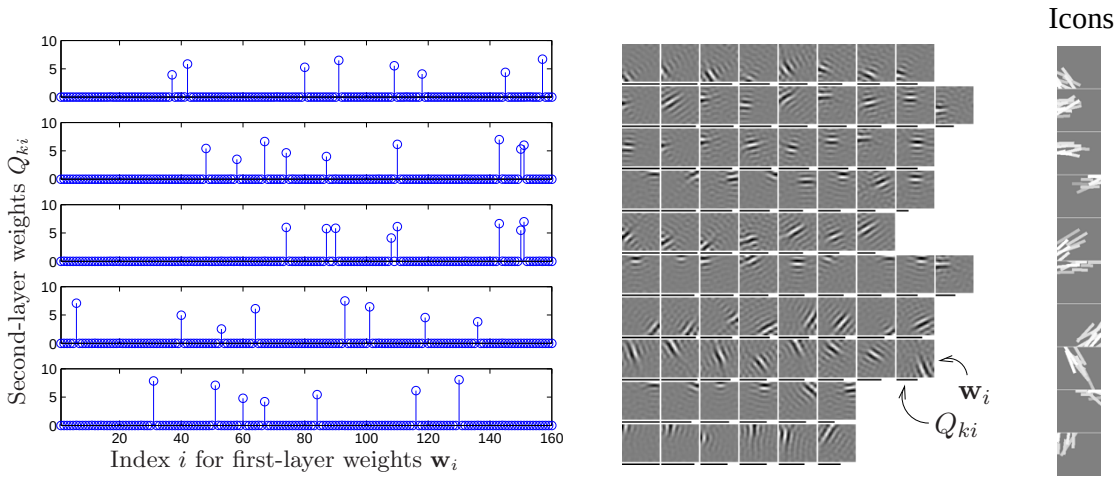
(a) Compression (b) Rectification (c) Resulting nonlinearity

Figure 8: Family of nonlinearities used in Subsection 5.3 for the modelling of natural images. The family is $f(y; a, b) = f_{\text{th}}(\ln(ay + 1) + b)$, $y \geq 0$. The function is composed of a compressive nonlinearity $\ln(ay + 1)$, shown in (a), and a smooth rectification function $f_{\text{th}}(u + b)$ shown in (b). Figure (c) shows examples of $f(y; a, b)$ for different values of a and b. Parameter b sets the threshold, and parameter a controls the steepness of the function. Since the scale of the weights in Eq. (37) is not restrained, the parameters a_k do not need to be learned explicitly. After learning, they can be identified by dividing y_k in Eq. (37) by a_k so that its expectation is 1. The formula for the thresholding function is $f_{\text{th}}(u) = 0.25 \ln(\cosh(2u)) + 0.5u + 0.17$. The curves shown in blue are for $b = -3$ and $a \in \{1, 50, 100, 200, \ldots, 500\}$. For the curves in red, $b = -5$. The squares indicate where f changes from convex to concave.



(a) Feature detectors (b) Optimal stimuli

Figure 9: Two-layer model with thresholding nonlinearities: visualization of the learned first-layer feature detectors $\mathbf{w}_i$. (a) The feature detectors in the pixel basis. (b) The corresponding optimal stimuli. The feature detectors in the first layer are "Gabor-like" (localized, oriented, bandpass). Comparison of the two figures shows that feature detectors which appear noisy in the pixel basis are tuned to low-frequency input.



(a) Raw result

(b) Graphical visualization

Figure 10: Two-layer model with thresholding nonlinearities: Random selection of second layer units. (a) Second-layer weights Q_{ki} for five different k (five different rows of the matrix $\mathbf{Q}$) are shown. The weights are extremely sparse so that in the sum $\sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2$ only few selected squared first-layer outputs are added together. (b) Every row shows one second-layer feature detector. The first-layer feature detectors $\mathbf{w}_i$ are shown as image patches like in Figure 9, and the black bar under each patch indicates the strength Q_{ki} by which a certain $\mathbf{w}_i$ is pooled by the k-th second-layer feature detector. The numerical values Q_{ki} for the first five rows are shown in Figure (a). The right-most column shows a condensed visualization. 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}.

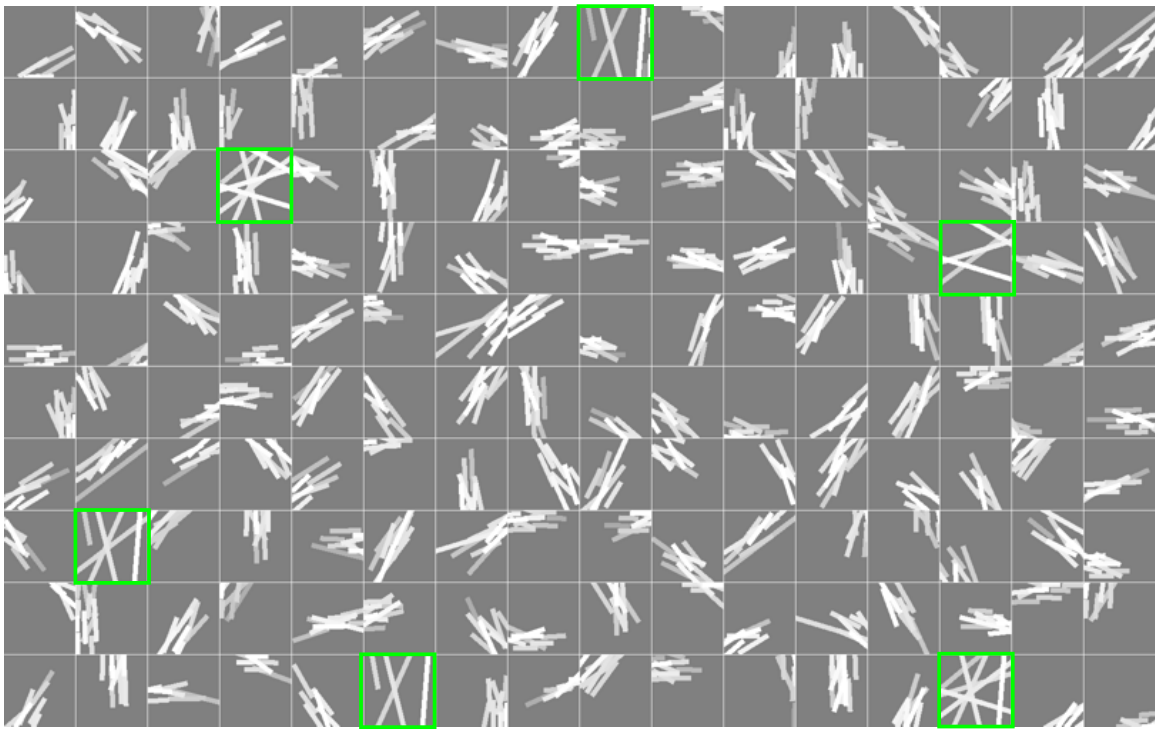
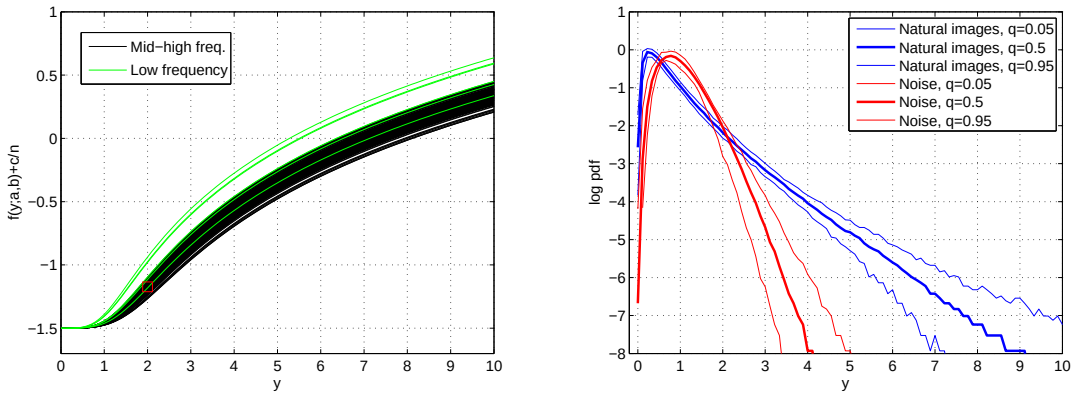


Figure 11: Two-layer model with thresholding nonlinearities: Visualization of the first- and second-layer feature detectors with icons. In the second layer, first-layer features of similar orientations are pooled together. See Figure 10 for details of how the icons are created. The feature detectors marked in green are tuned to low frequencies. The corresponding nonlinearities in Figure 12 and 13 are also shown in green.



(a) Learned nonlinearities

(b) Distribution of second-layer outputs y_k

Figure 12: Two-layer model with thresholding nonlinearities: Learned nonlinearities and interpretation. Natural images tend to have larger second-layer outputs y_k than noise input since the two processing layers, visualized in Figures 9 to 11, detect structure inherent to natural images. Thresholding the y_k provides a way to assign to natural images large values in the model pdf and to noise small values. In Figure (a), the nonlinearities acting on pooled low-frequency feature detectors are shown in green, those for medium and high frequency feature detectors in black. The bold curve in Figure (b) shows the median, the other curves the 5% and 95% quantiles. Curves in blue relate to natural images, curves in red to noise.

example Hastie et al., 2009, ch. 5). First, we improve the model of the previous section by optimizing f only, keeping the features fixed. Then, we consider the joint learning of features and nonlinearity.

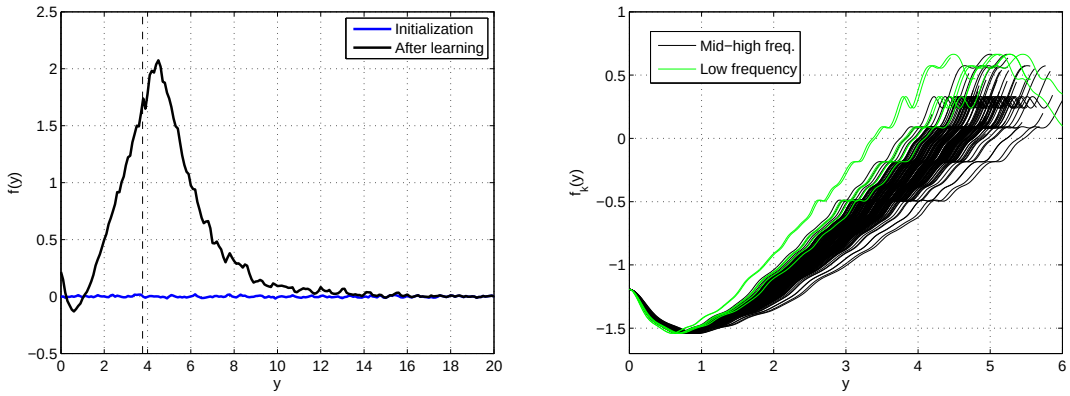
5.4.1 Refinement of the thresholding model

We take here a simple approach and leave the feature extraction layers that were obtained for the thresholding model fixed, and learn only the cubic spline nonlinearity f. The model is thus

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{n} f(y_k;\boldsymbol{\theta}) + c, \qquad y_k = \sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2, \tag{38}$$

where the vector $\boldsymbol{\theta}$ contains the parameters for the nonlinearity f and the normalizing parameter c. The knots of the spline are set to have an equal spacing of 0.1 on the interval $[0\ 20]$. Outside that interval, we define f to stay constant. With that specification, we can write f in terms of 203 B-spline basis functions. The parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{204}$ contains then the 203 coefficients for the basis functions and the parameter c.

Results Figure 13(a) shows the learned nonlinearity (black curve) and its random initialization (blue curve). The dashed line around $y = 4$ indicates the border of validity of the nonlinearity since 99% of the y_k fall, for natural image input, to the left of the dashed line. The salient property of the emerging nonlinearity is the "dip" after zero which makes



(a) Learned nonlinearity

(b) Learned effective nonlinearities

Figure 13: Refinement of the thresholding model of Subsection 5.3. Only the nonlinearity was learned, the features were kept fixed. (a) Learned spline (black curve) and the initialization (blue curve). The dashed line indicates the border of validity of the learned nonlinearity since 99% of the y_k fall, for natural image input, to the left of it. (b) The different scales of the y_k give rise to a set of effective nonlinearities f_k, as shown in Eq. (39). Nonlinearities acting on low-frequency feature detectors are shown in green, the others in black.

f non-monotonic. Figure 13(b) shows the effective nonlinearities when the different scales of the second layer outputs y_k are taken into account: We calculate the scale σ_k by taking the average value of y_k over the natural images. The different scales σ_k then define different nonlinearities. Incorporating the normalizing parameter c into the nonlinearity, we obtain the set of effective nonlinearities $f_k(y)$,

$$f_k(y) = f(\sigma_k y) + c/n,\ k = 1, \dots n. \tag{39}$$

For the nonlinearities f_k, the dip occurs between zero and two. Inspection of Figure 12(b) shows that the optimal nonlinearities f_k take, unlike the thresholding nonlinearities, the distribution of the second-layer outputs y_k fully into account: The region where the dip occurs is just the region where noise input is more likely than natural image input. The learned nonlinearities are thus more effective than the thresholding nonlinearities in assigning to natural images and noise input large and small values, respectively.

5.4.2 Joint learning of features and nonlinearities: one-layer model

In the models so far, we have made the assumption that $\ln p_m$ is a an even function of the first-layer feature outputs. Here, we consider a one-layer model of the form

$$\ln p_m(\mathbf{x}; \boldsymbol{\theta}) = \sum_{k=1}^{n} f(\mathbf{w}_k^T \mathbf{x}; a_1, a_2, \dots) + c, \tag{40}$$

where the nonlinearity f is a cubic spline. No symmetry assumption is made in this model. The parameters are the feature weights $\mathbf{w}_k \in \mathbb{R}^n$, $c \in \mathbb{R}$ for the normalization of the pdf,

as well as the $a_i \in \mathbb{R}$ for the parameterization of the nonlinearity f. For the modeling of the nonlinearity, we must also define its domain. Its domain is related to the range of its arguments $\mathbf{w}_k^T\mathbf{x}$. To avoid ambiguities in the model specification, we constrain the vectors $\mathbf{w}_k$ such that

$$\max_k \mathrm{E}\left\{(\mathbf{w}_k^T\mathbf{x})^2\right\} = 1, \tag{41}$$

where the expectation is taken over the natural images. Defining f as a cubic spline on the whole real line is impossible since the number of parameters a_i would become intractable. With the constraint in Eq. (41), it is enough to define f only on the interval $[-10\ 10]$ as a cubic spline. For that, we use a knot sequence with an equal spacing of 0.1. Outside the interval, we define f to stay constant. With this specifications, we can write f in terms of B-spline basis functions with 203 coefficients $a_1, \ldots, a_{203}$.

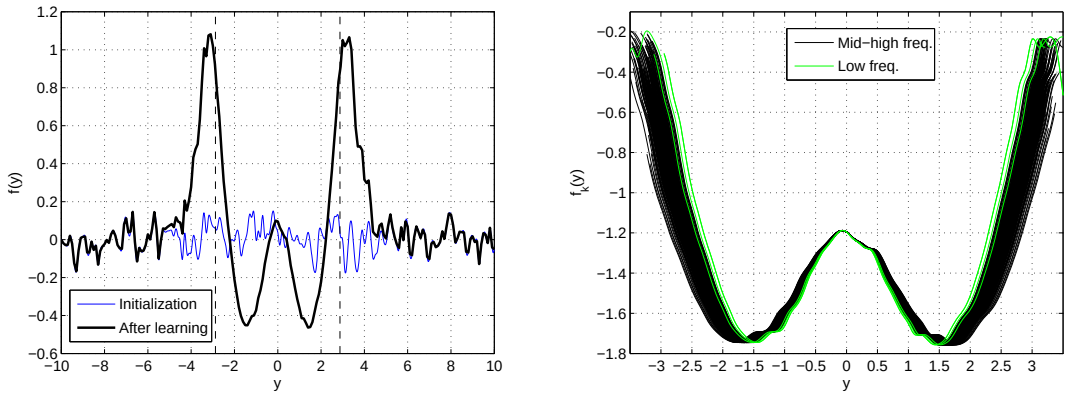
Results As in the two-layer model, the learned features are "Gabor-like" (results not shown). We observed, however, a smaller number of feature detectors that are tuned to low frequencies. Figure 14(a) shows the learned nonlinearity f (black curve) and the random initialization (blue curve). The dashed lines indicate the interval where 99% of the feature outputs occur for natural image input. The learned nonlinearity should thus only be considered valid on that interval. Since no norm constraint was imposed on the $\mathbf{w}_i$, the different scales of the vectors define, like in Eq. (39), effectively different nonlinearities f_k: $f_k(u) = f(\sigma_k u) + c/n$ where σ_k is a normalization factor such that $\mathrm{Var}(\mathbf{w}_k^T\mathbf{x}/\sigma_k) = 1$. The nonlinearities f_k are plotted in Figure 14(b). They have two striking properties: First, they are even functions. Note that no such constraint was imposed, so the symmetry of the nonlinearities is due to symmetry in the natural images. This result validates the assumption of the previous section. It also updates a previous result of ours where we have searched in a more restrictive space of functions and no symmetric nonlinearity emerged (Gutmann and Hyvärinen, 2009). Second, the f_k are not monotonic. The reason for the non-monotonic behavior is the same as for the nonlinearities in Figure 13: The learned nonlinearities are such that natural images are most often mapped to large numbers while noise is mapped to small numbers. The non-monotonicity has also a simple interpretation in terms of sparse coding. The absolute values of the feature outputs are often very large or very small in natural images, which means that their distribution is sparse.

5.4.3 Joint learning of features and nonlinearities: two-layer model

We extend here the spline-based one-layer model to a two-layer model. The model is thus like in Subsection 5.3 but the nonlinearity f is a cubic spline,

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{n} f(y_k; a_1, a_2, \ldots) + c, \qquad y_k = \sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2. \tag{42}$$

The parameters $\boldsymbol{\theta}$ are as in Subsection 5.3 the $\mathbf{w}_i \in \mathbb{R}^n$, $Q_{ki} \geq 0$, and $c \in \mathbb{R}$. But additionally, we have as parameters also the $a_i \in \mathbb{R}$ which are the coefficients of the B-spline basis functions of the cubic spline f. As in the one-layer model, we need to constrain the range of the arguments y_k of the spline f. A possible way to achieve this is to impose Eq. (41) as a constraint for the first-layer outputs feature $\mathbf{w}_i^T\mathbf{x}$, and to constrain the columns of the matrix $\mathbf{Q}$ to have norm one. We used the learned parameters of the one-layer model to



(a) Learned nonlinearity

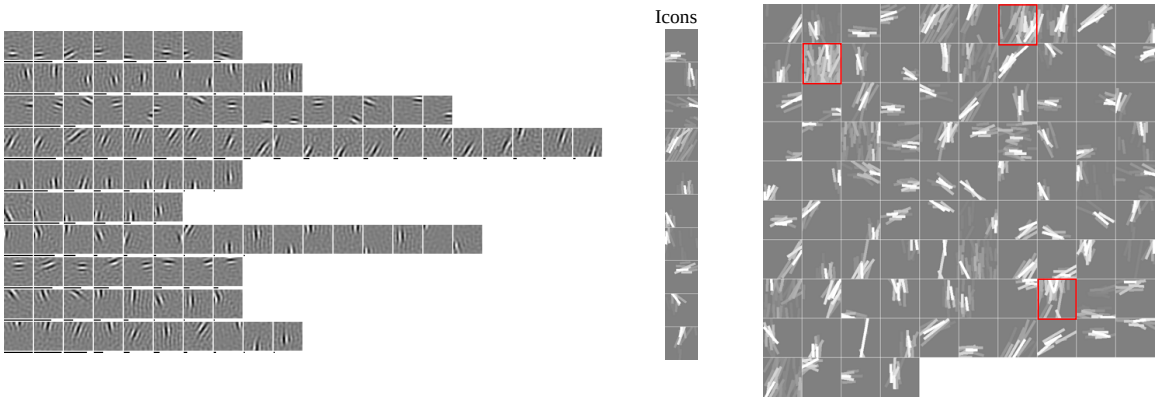
(b) Learned effective nonlinearities

Figure 14: One-layer model with spline nonlinearities. (a) Learned spline (black curve) and the initialization (blue curve). The dashed lines indicate the interval where 99% of the feature outputs occur for natural image input. The learned nonlinearity should thus only be considered valid on that interval. (b) The different norms of the learned $\mathbf{w}_i$ give rise to a set of nonlinearities f_k. Nonlinearities acting on low-frequency feature detectors are shown in green, the others in black.

initialize the two-layer model. It turned then out that imposing Eq. (41) was enough for the learning to work and no norm constraint for the columns of $\mathbf{Q}$ was necessary. The results were very similar whether there were norm constraints or not. In the following, we report the results without any norm constraints.

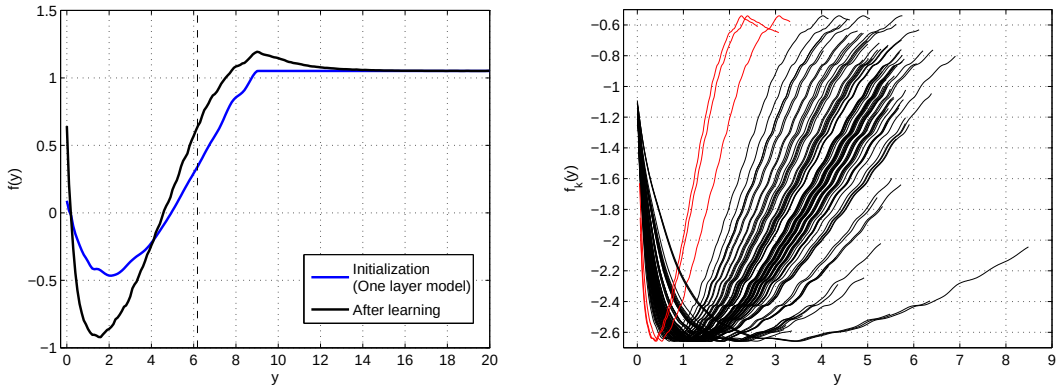
Results Figure 15 visualizes the learned parameters $\mathbf{w}_i$ and Q_{ki} in the same way as for the two-layer model with thresholding nonlinearities (Subsection 5.3: Figure 10 and 11). The learned feature extraction stage is qualitatively, up to two differences, very similar. The first difference is that many second-layer weights Q_{ki} shrunk to zero: 66 out of 160 rows of the matrix $\mathbf{Q}$ had so small values that we could omit them while accounting for 99.9% of the sum $\sum_{ki} Q_{ki}$. The second difference is that the pooling in the second layer is sometimes less sparse. In that case, the second layer still combines first-layer feature detectors of the same orientation but they are not all centered at the same location (examples are marked with red in Figure 15).

The learned nonlinearity f is shown in Figure 16(a) (black curve) together with its initialization based on the one-layer model (blue curve). The dashed line indicates the 99% quantile for all the feature outputs y_k for natural image input. The behavior of f on the right of the dashed line is thus due to only few training data and the nonlinearity should only be considered valid on the interval to left of the line. The nonlinearity from the one-layer model is altered so that higher values are assigned to small and large inputs while intermediate inputs are mapped to smaller numbers. Figure 16(b) shows the effective nonlinearities f_k, defined in Eq. (39), when the different scales of the second-layer outputs are taken into account. Compared to the previous models, the scales are far more diverse so that qualitatively different nonlinearities emerge.



(a) Pooling in the second layer (b) Representation with icons

Figure 15: Two-layer model with spline nonlinearities. (a) Random selection of the learned second-layer units. (b) Representation of all the learned second-layer feature detectors as iconic images. The corresponding nonlinearities f_k are shown in Figure 16.



(a) Learned nonlinearity (b) Learned effective nonlinearities

Figure 16: Two-layer model with spline nonlinearities. (a) Learned spline (black curve) and the initialization (blue curve). The dashed line indicates the 99% quantile for all the feature outputs y_k for natural images. The nonlinearity should only be considered valid on the interval to left of the line. (b) The different scales of the y_k give rise to a set of nonlinearities f_k. Nonlinearities marked in red correspond to the red-colored features in Figure 15.

5.5 Model comparison

In the previous sections, we have learned one and two-layer models for natural images, both with thresholding nonlinearities and with splines. We make here a simple model comparison.

A quantitative comparison is done by calculating for a validation set the value of the objective function J_T of Eq. (13), which is used in noise-contrastive estimation. The sample size of the validation set was $T_d = 100000$, and ν was set to 10, as in the estimation of the models. For the same validation data, we also computed the rescaled log-likelihood (average, sign-inverted log-loss) $L_T = 1/T_d \sum_t \ln p_m(\mathbf{x}_t; \hat{\boldsymbol{\theta}}_T)$ for the different models.

A qualitative comparison can be performed by sampling from the models. We choose here a simplistic approach to sampling: we drew random samples that followed the noise distribution p_n (uniform on the sphere), and used them as initial points in the optimization of the various log-densities $\ln p_m(\mathbf{x}; \hat{\boldsymbol{\theta}}_T)$ with respect to $\mathbf{x}$ under the constraint of Eq. (34). We used the same initial points for all models. The optimization with respect to the coordinates $\mathbf{x}$ can be considered as computing the maximum-a-posteriori estimate of the image under an additive noise model with strong noise variance. Given an optimal point $\hat{\mathbf{x}}$, a "sample" is obtained via Eq. (35) as $\hat{\mathbf{I}} = \mathbf{V}^{-}\hat{\mathbf{x}}$.

The ICA model with Laplacian sources is a simple model for natural images (see for example Hyvärinen et al., 2009, ch. 7). In Section 3.2, we considered the estimation of the unnormalized version of that model. We include the unnormalized model, as defined in Eq. (23), into our comparison. The normalized model cannot be used here since the partition function for the ICA model is calculated by integration over the whole space and applies thus not to our data.[6] This is a one-layer model with a fixed nonlinearity, given by $f(u) = -\sqrt{2}|u|$. We refer to this model as one-layer model with "Laplacian nonlinearity".

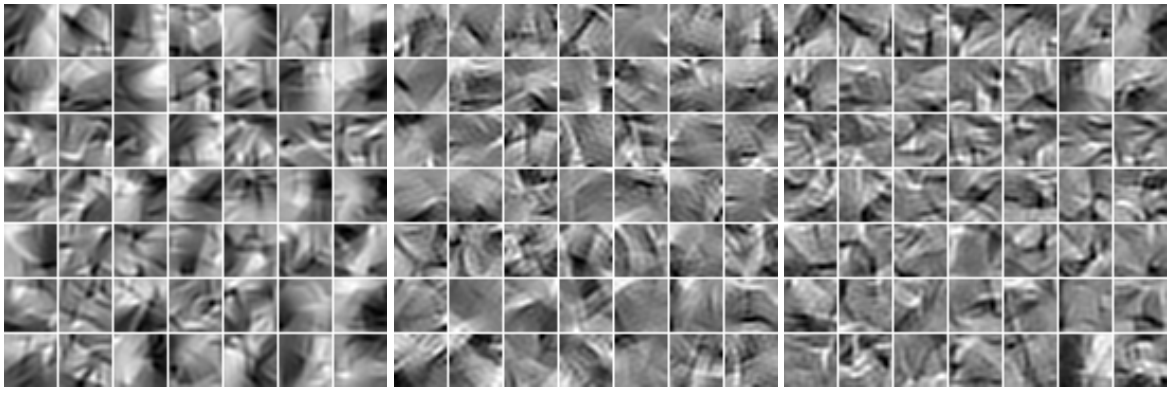
Results Table 1 shows that the spline-based two-layer model, defined in Eq. (42), gives the largest value of the objective function J_T, and also L_T. It is closely followed by the model in Eq. (38) where we learned the nonlinearity f only while keeping the feature extraction stage fixed. The one-layer models with thresholding or Laplacian nonlinearities have the smallest objectives J_T and L_T. The two models achieve the objectives in different ways. For the thresholding model, the absolute value of the feature outputs $\mathbf{w}_i^T\mathbf{x}$ must be large to yield a large objective while for the model with the Laplacian nonlinearity $f(\mathbf{w}_i^T\mathbf{x}) = -\sqrt{2}|\mathbf{w}_i^T\mathbf{x}|$, the feature outputs must have small absolute values. The two models consider thus different aspects of the, for natural images, typically sparsely distributed feature outputs $\mathbf{w}_i^T\mathbf{x}$. The one-layer model with spline nonlinearity combines both aspects, see Figure 14, and yields also the higher scores in the comparison. The same reason explains why spline-based two-layer models have higher scores than the two-layer model with fixed nonlinearity.

Figure 17 shows samples from the various models p_m. The models with large objectives in Table 1 lead to samples with particularly clear structure. The emergence of structure can be explained in terms of sparse coding. Samples which lead to sparse activations of the feature detectors are typically highly structured. In the sampling, sparseness of the feature outputs is faciliated by the nonlinearities in the models, and through the competition between the features by means of the sphere-constraint in Eq. (34) on the coordinates $\mathbf{x}$.

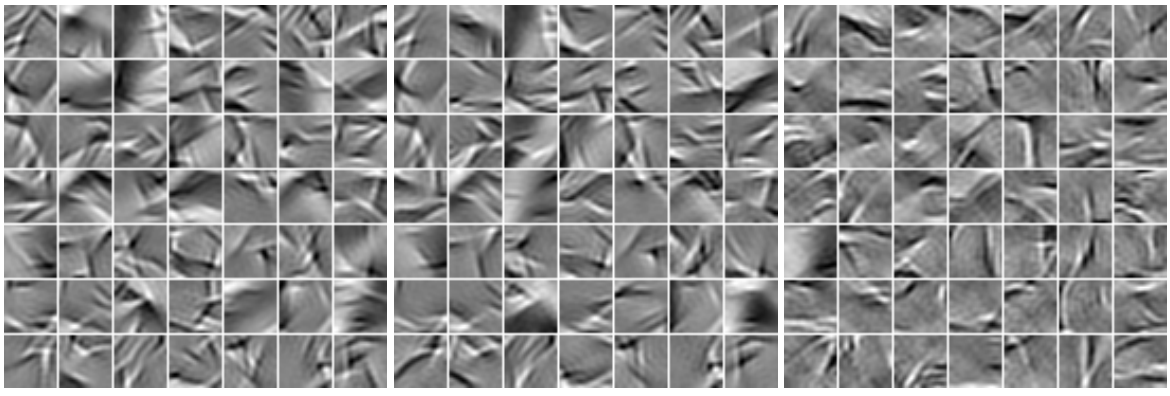
6. In simulations with the normalized model, we only obtained noisy feature detectors $\mathbf{w}_i$.

	One-layer model			Two-layer model		
	Thresholding	Laplacian nonl.	Spline	Thresholding	Refinement	Spline
J_T	-1.88	-1.52	-1.07	-0.877	-0.627	**-0.616**
L_T	-224.2	-223.6	-220.7	-221.7	-214.2	**-213.5**

Table 1: Quantitative model comparison. The objective J_T of Eq. (13) and the (rescaled) log-likelihood L_T are used to measure the performance. Larger values indicate better performance. The features for the one-layer models with thresholding and Laplacian nonlinearity were not shown in the paper. The "one-layer, thresholding" model is identical to the "two-layer, thresholding" model when the second layer is fixed to the identity matrix. With Laplacian nonlinearity we mean the function $f(u) = -\sqrt{2}|u|$. The "two-layer, tresholding" model was presented in Subsection 5.3. The "two-layer, refinement", "one-layer, spline", and "two-layer, spline" models were in that order treated in Subsection 5.4. All models are for random variables that lie on a n-dimensional sphere $\mathbb{S}^n$.



(a) One-layer, thresholding (b) One-layer, Laplacian nonl. (c) One-layer, spline



(d) Two-layer, thresholding (e) Two-layer, refinement (f) Two-layer, spline

Figure 17: Sampling from the learned models for natural images. For every model, we used the same random initialization. The samples from the different models can thus explicitely be compared to each other. All models are for random variables that lie on a n-dimensional sphere $\mathbb{S}^n$.

6. Conclusions

In this paper, we have considered the problem of estimating unnormalized statistical models for which the normalizing partition function cannot be computed in closed form. Direct estimation by maximization of likelihood is then not an option. The main contribution of the paper is a new estimation method for unnormalized models. A further contribution is made in the modeling of natural image statistics.

We have proven that our new estimation method, noise-contrastive estimation, provides a consistent estimator for both normalized and unnormalized statistical models. The assumptions that must be fulfilled to have consistency are not stronger than the assumptions that are needed in maximum likelihood estimation. We have further derived the asymptotic distribution of the estimation error which shows that, in the limit of arbitrarily many contrastive noise samples, the estimator performs like the maximum likelihood estimator. The new method has a very intuitive interpretation in terms of supervised learning: The estimation is performed by discriminating between the observed data and some artificially generated noise by means of logistic regression.

All theoretical results were illustrated and validated on artificial data where ground truth is known. We have also used artificial data to assess the balance between statistical and computational performance. In particular, we have compared the new estimation method to a number of other estimation methods for unnormalized models: Simulations suggest that noise-contrastive estimation strikes a highly competitive trade-off. We have used the mean squared error of the estimated parameters as statistical performance measure. It should be noted that this is only one possible criterion among many (see Hyvärinen, 2008, for a recently proposed alternative measure of performance).

Noise-contrastive estimation as presented here extends the previous definition given by Gutmann and Hyvärinen (2010) since it allows for more noise samples than data points. We considered such a generalization also previously (Pihlaja et al., 2010). Unlike in that preliminary version, our method here is asymptotically Fisher-efficient for all admissible noise densities when the number of noise samples becomes arbitrarily large. Pihlaja et al. (2010) established links of noise-contrastive estimation to importance sampling which remain valid for this paper.

We applied noise-contrastive estimation to the modeling of natural images. Besides validating the method on a large scale two-layer model, we have, as a new contribution to natural image statistics, presented nonparametric extensions. In previous models, the output nonlinearity in the pdf was hand-picked. Here, we have parameterized it as a spline and learned it from the data. The statistical models were all unnormalized and had several ten-thousands of parameters which demonstrates that our new estimation method can handle demanding large-scale problems.

References

C. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

C.J. Geyer. On the convergence of monte carlo maximum likelihood calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):261–274, 1994.

M. Gutmann and A. Hyvärinen. Learning features by contrasting natural images with noise. In *Proceedings of the 19th International Conference on Artificial Neural Networks (ICANN)*, 2009.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

A. Hyvärinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

A. Hyvärinen. Optimal approximation of signal priors. *Neural Computation*, 20:3087–3110, 2008.

A. Hyvärinen, P.O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Comp.*, 13(7):1527–1558, 2001a.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, 2001b.

A. Hyvärinen, J. Hurri, and P.O. Hoyer. *Natural Image Statistics*. Springer, 2009.

Y. Karklin and M. Lewicki. A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. *Neural Computation*, 17:397–423, 2005.

D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

U. Köster and A. Hyvärinen. A two-layer model of natural stimuli estimated with score matching. *Neural Computation*, 22(9):2308–2333, 2010.

J. Lücke and M. Sahani. Maximal causes for non-linear component extraction. *Journal of Machine Learning Research*, 9:1227 – 1267, 2008.

D. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

J. Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

S.G. Nash. A survey of truncated-newton methods. *Journal of Computational and Applied Mathematics*, 124:45–59, 2000.

B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

S. Osindero and G. Hinton. Modeling image patches with a directed hierarchy of markov random fields. In *Advances in Neural Information Processing Systems 20*, pages 1121–1128. MIT Press, 2008.

S. Osindero, M. Welling, and G. E. Hinton. Topographic product models applied to natural scene statistics. *Neural Computation*, 18 (2), 2006.

M. Pihlaja, M. Gutmann, and A. Hyvärinen. A family of computationally efficient and simple estimators for unnormalized statistical models. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

M.A. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *Proceedings of the 23rd Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

C.E. Rasmussen. Conjugate gradient algorithm, version 2006-09-08. 2006.

Y. Teh, M. Welling, S. Osindero, and G. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, 2004.

J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1394):359–366, 1998.

L. Wasserman. *All of statistics*. Springer, 2004.

Appendix A. Proofs of the theorems

We give here detailed proofs for Theorem 1, 2 and 3 on nonparametric estimation, consistency and the asymptotic distribution of the estimator, respectively.

A.1 Preliminaries

In the proofs, we use often the following properties of the function $r_\nu(u)$,

$$r_\nu(u) = \frac{1}{1+\nu\exp(-u)}, \tag{43}$$

which was introduced in Eq. (9):

$$\begin{aligned}
1 - r_\nu(u) &= r_{\frac{1}{\nu}}(-u) && (44)\\
\frac{\partial r_\nu(u)}{\partial u} &= r_{\frac{1}{\nu}}(-u) r_\nu(u) && (45)\\
\frac{\partial}{\partial u} \ln r_\nu(u) &= r_{\frac{1}{\nu}}(-u) && (46)\\
\frac{\partial^2}{\partial u^2} \ln r_\nu(u) &= -r_{\frac{1}{\nu}}(-u) r_\nu(u) && (47)\\
\frac{\partial}{\partial u} \ln[1 - r_\nu(u)] &= -r_\nu(u) && (48)\\
\frac{\partial^2}{\partial u^2} \ln[1 - r_\nu(u)] &= -r_{\frac{1}{\nu}}(-u) r_\nu(u) && (49)
\end{aligned}$$

The functions $h(\mathbf{u};\boldsymbol{\theta}) = r_\nu\left(G(\mathbf{u};\boldsymbol{\theta})\right)$ and $1 - h(\mathbf{u};\boldsymbol{\theta}) = r_{\frac{1}{\nu}}\left(-G(\mathbf{u};\boldsymbol{\theta})\right)$ are equal to

$$h(\mathbf{u};\boldsymbol{\theta}) = \frac{p_m(\mathbf{u};\boldsymbol{\theta})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \qquad 1 - h(\mathbf{u};\boldsymbol{\theta}) = \frac{\nu p_n(\mathbf{u})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \tag{50}$$

see Eq. (5) and Eq. (6). It follows that

$$\nu p_n(\mathbf{u}) r_\nu\left(G(\mathbf{u};\boldsymbol{\theta})\right) = \frac{\nu p_n(\mathbf{u}) p_m(\mathbf{u};\boldsymbol{\theta})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \tag{51}$$

$$p_d(\mathbf{u}) r_{\frac{1}{\nu}}\left(-G(\mathbf{u};\boldsymbol{\theta})\right) = \frac{\nu p_n(\mathbf{u}) p_d(\mathbf{u})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \tag{52}$$

which are key properties for the proofs below.

The first and second order derivatives are used in the following Taylor expansions

$$\begin{aligned}
\ln r_\nu(u + \epsilon u_1 + \epsilon^2 u_2) &= \ln r_\nu(u) + \epsilon r_{\frac{1}{\nu}}(-u) u_1 + \\
&\quad \epsilon^2 \left[r_{\frac{1}{\nu}}(-u) u_2 - \frac{1}{2} r_{\frac{1}{\nu}}(-u) r_\nu(u) u_1^2 \right] + \\
&\quad O(\epsilon^3), \\
\ln\left[1 - r_\nu(u + \epsilon u_1 + \epsilon^2 u_2)\right] &= \ln[1 - r_\nu(u)] - \epsilon r_\nu(u) u_1 + \\
&\quad \epsilon^2 \left[-r_\nu(u) u_2 - \frac{1}{2} r_{\frac{1}{\nu}}(-u) r_\nu(u) u_1^2 \right] + \\
&\quad O(\epsilon^3). && (53)
\end{aligned}$$

A.2 Proof of Theorem 1 (nonparametric estimation)

A.2.1 Lemma

The Taylor expansions in Eq. (53) are used to prove the following lemma.

Lemma 8 *For $\epsilon > 0$ and $\phi(\mathbf{x})$ a perturbation of the log-pdf $f_m(\mathbf{x}) = \ln p_m(\mathbf{x})$,*

$$\begin{aligned}\tilde{J}(f_m + \epsilon\phi) &= \tilde{J}(f_m) + \epsilon \int [p_d(\mathbf{u}) r_{\frac{1}{\nu}}(-f_m(\mathbf{u}) + \ln p_n(\mathbf{u})) - \\ &\quad \nu p_n(\mathbf{u}) r_\nu(f_m(\mathbf{u}) - \ln p_n(\mathbf{u}))]\phi(\mathbf{u})\mathrm{d}\mathbf{u} - \\ &\quad \frac{\epsilon^2}{2} \int r_{\frac{1}{\nu}}(-f_m(\mathbf{u}) + \ln p_n(\mathbf{u})) r_\nu(f_m(\mathbf{u}) - \ln p_n(\mathbf{u})) \\ &\quad (p_d(\mathbf{u}) + \nu p_n(\mathbf{u}))\phi(\mathbf{u})^2 \mathrm{d}\mathbf{u} + O(\epsilon^3).\end{aligned}$$

Proof The proof is obtained by evaluating the objective function $\tilde{J}$ in Eq. (16) at $f_m + \epsilon\phi$, and making then use of the Taylor expansions in Eq. (53) with $u = f_m(\mathbf{x}) - \ln p_n(\mathbf{x})$, $u_1 = \phi(\mathbf{x})$ and $u_2 = 0$. ■

A.2.2 Proof of the theorem

Proof A necessary condition for optimality is that in the expansion of $\tilde{J}(f_m + \epsilon\phi)$, the term of order ϵ is zero for any perturbation ϕ. This happens if and only if

$$p_d(\mathbf{u}) r_{\frac{1}{\nu}}(-f_m(\mathbf{u}) + \ln p_n(\mathbf{u})) = \nu p_n(\mathbf{u}) r_\nu(f_m(\mathbf{u}) - \ln p_n(\mathbf{u})). \tag{54}$$

With Eq. (51) and Eq. (52), this implies that $\tilde{J}$ has an extremum at p_m if and only if

$$\frac{\nu p_n(\mathbf{u}) p_d(\mathbf{u})}{p_m(\mathbf{u}) + \nu p_n(\mathbf{u})} = \frac{\nu p_n(\mathbf{u}) p_m(\mathbf{u})}{p_m(\mathbf{u}) + \nu p_n(\mathbf{u})}. \tag{55}$$

That is, as $\nu > 0$, $p_m(\mathbf{u}) = p_d(\mathbf{u})$ at all points $\mathbf{u}$ where $p_n(\mathbf{u}) \neq 0$. At points where $p_n(\mathbf{u}) = 0$, the equation is trivially fulfilled. Hence, $p_m = p_d$, or $f_m = \ln p_d$, leads to an extremum of $\tilde{J}$.

Inserting $f_m = \ln p_d$ into $\tilde{J}$ in Lemma 8 leads to

$$\tilde{J}(\ln p_d + \epsilon\phi) = \tilde{J}(\ln p_d) - \frac{\epsilon^2}{2} \left\{ \int \frac{\nu p_n(\mathbf{u}) p_d(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})} \phi(u)^2 \mathrm{d}\mathbf{u} \right\} + O(\epsilon^3). \tag{56}$$

Since the term of order ϵ^2 is negative for all choices of ϕ, the extremum is a maximum. The assumption that $p_n(\mathbf{u}) \neq 0$ whenever $p_d(\mathbf{u}) \neq 0$ shows that $f_m = \ln p_d$ is the only extremum and completes the proof. ■

A.3 Proof of Theorem 2 (consistency)

A.3.1 Lemmata

The Taylor expansions in Eq. (53) are used to prove the following lemma which is like Lemma 8 for $\tilde{J}$ but for the objective function J in Eq. (15).

Lemma 9 *For $\epsilon > 0$ and $\boldsymbol{\varphi} \in \mathbb{R}^m$,*

$$\begin{aligned} J(\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi}) &= J(\boldsymbol{\theta}) + \epsilon \int u_1 \left[p_d(\mathbf{u})(1 - h(\mathbf{u};\boldsymbol{\theta})) - \nu p_n(\mathbf{u}) h(\mathbf{u};\boldsymbol{\theta})\right] \mathrm{d}\mathbf{u} + \\ &\quad \epsilon^2 \left\{ \int -\frac{1}{2} u_1^2 (1 - h(\mathbf{u};\boldsymbol{\theta})) h(\mathbf{u};\boldsymbol{\theta}) \left(p_d(\mathbf{u}) + \nu p_n(\mathbf{u})\right) \mathrm{d}\mathbf{u} + \right. \\ &\quad \left. \int u_2 \left(p_d(\mathbf{u})(1 - h(\mathbf{u};\boldsymbol{\theta})) - \nu p_n(\mathbf{u}) h(\mathbf{u};\boldsymbol{\theta})\right) \mathrm{d}\mathbf{u} \right\} + O(\epsilon^3), \end{aligned} \tag{57}$$

where

$$u_1 = \boldsymbol{\varphi}^T \mathbf{g}(\mathbf{u};\boldsymbol{\theta}), \tag{58}$$

$$u_2 = \frac{1}{2} \boldsymbol{\varphi}^T \mathbf{H}_G(\mathbf{u};\boldsymbol{\theta}) \boldsymbol{\varphi}. \tag{59}$$

The term $\mathbf{g}(\mathbf{u};\boldsymbol{\theta})$ is $\nabla G(\mathbf{u};\boldsymbol{\theta})$, and $\mathbf{H}_G$ denotes the Hessian matrix of $G(\mathbf{u};\boldsymbol{\theta})$ where the derivatives are taken with respect to $\boldsymbol{\theta}$.

Proof With the definition of J in Eq. (15), we have

$$\begin{aligned} J(\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi}) &= \int \ln \left[r_\nu \left(G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})\right)\right] p_d(\mathbf{u}) \mathrm{d}\mathbf{u} + \\ &\quad \nu \int \ln \left[1 - r_\nu \left(G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})\right)\right] p_n(\mathbf{u}) \mathrm{d}\mathbf{u}. \end{aligned} \tag{60}$$

Developing $G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})$ till terms of order ϵ^2 yields

$$G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi}) = G(\mathbf{u};\boldsymbol{\theta}) + \epsilon \boldsymbol{\varphi}^T \mathbf{g}(\mathbf{u};\boldsymbol{\theta}) + \epsilon^2 \frac{1}{2} \boldsymbol{\varphi}^T \mathbf{H}_G(\mathbf{u};\boldsymbol{\theta}) \boldsymbol{\varphi} + O(\epsilon^3). \tag{61}$$

Defining u_1 and u_2 as in the lemma, we obtain

$$\ln r_\nu \left(G(\mathbf{u};\boldsymbol{\theta} + \epsilon\nu)\right) = \ln r_\nu \left(G(\mathbf{u};\boldsymbol{\theta}) + \epsilon u_1 + \epsilon^2 u_2 + O(\epsilon^3)\right). \tag{62}$$

Using now the Taylor expansions in Eq. (53) for $u = G(\mathbf{u};\boldsymbol{\theta})$, and the identities $h(\mathbf{u};\boldsymbol{\theta}) = r_\nu \left(G(\mathbf{u};\boldsymbol{\theta})\right)$ as well as $1 - h(\mathbf{u};\boldsymbol{\theta}) = r_{\frac{1}{\nu}} \left(-G(\mathbf{u};\boldsymbol{\theta})\right)$ proves the lemma. ■

Lemma 10 *If $p_n(\mathbf{u}) \neq 0$ whenever $p_d(\mathbf{u}) \neq 0$ and if*

$$\boldsymbol{\mathcal{I}}_\nu = \int \mathbf{g}(\mathbf{u}) \mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{63}$$

is full rank, where

$$P_\nu(\mathbf{u}) = \frac{\nu p_n(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}, \tag{64}$$

$$\mathbf{g}(\mathbf{u}) = \nabla_{\boldsymbol{\theta}} \ln p_m(\mathbf{u};\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}, \tag{65}$$

then

$$J(\boldsymbol{\theta}^\star) > J(\boldsymbol{\theta}^\star + \boldsymbol{\varphi}) \quad \forall \boldsymbol{\varphi} \neq \mathbf{0}. \tag{66}$$

Proof A necessary condition for optimality is that in the expansion of $J(\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})$ in Lemma 9, the term of order ϵ is zero for any $\boldsymbol{\varphi}$. This happens if

$$p_d(\mathbf{u})(1 - h(\mathbf{u}; \boldsymbol{\theta})) = \nu p_n(\mathbf{u}) h(\mathbf{u}; \boldsymbol{\theta}), \tag{67}$$

that is, if

$$\frac{\nu p_n(\mathbf{u}) p_d(\mathbf{u})}{p_m(\mathbf{u}; \boldsymbol{\theta}) + \nu p_n(\mathbf{u})} = \frac{\nu p_n(\mathbf{u}) p_m(\mathbf{u}; \boldsymbol{\theta})}{p_m(\mathbf{u}; \boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \tag{68}$$

where we have used Eq. (51) and Eq. (52) as in the proof for Lemma 8. The assumption that $\nu > 0$ and $p_d(.) = p_m(.; \boldsymbol{\theta}^\star)$ implies together with the above equation that the term of order ϵ is zero if $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$.

The objective function $J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi})$ becomes thus

$$\begin{aligned} J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi}) &= J(\boldsymbol{\theta}^\star) - \frac{\epsilon^2}{2} \int u_1^2 (1 - h(\mathbf{u}; \boldsymbol{\theta}^\star)) h(\mathbf{u}; \boldsymbol{\theta}^\star) \\ &\qquad (p_d(\mathbf{u}) + \nu p_n(\mathbf{u}))\, \mathrm{d}\mathbf{u} + O(\epsilon^3). \end{aligned} \tag{69}$$

The terms $h(\mathbf{u}; \boldsymbol{\theta}^\star)$ and $1 - h(\mathbf{u}; \boldsymbol{\theta}^\star)$ are with Eq. (50)

$$h(\mathbf{u}; \boldsymbol{\theta}^\star) = \frac{p_d(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}, \qquad 1 - h(\mathbf{u}; \boldsymbol{\theta}^\star) = \frac{\nu p_n(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}. \tag{70}$$

The expression for $J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi})$ becomes then

$$J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi}) = J(\boldsymbol{\theta}^\star) - \frac{\epsilon^2}{2} \boldsymbol{\varphi}^T \left[\int \mathbf{g}(\mathbf{u}) \mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u} \right] \boldsymbol{\varphi} + O(\epsilon^3) \tag{71}$$

by inserting the definition of u_1 evaluated at $\boldsymbol{\theta}^\star$, and making use of the definitions for $P_\nu(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$ in the statement of the lemma. The term of order ϵ^2 defines the nature of the extremum at $\boldsymbol{\theta}^\star$. If $\boldsymbol{\mathcal{I}}_\nu$ is positive definite, $J(\boldsymbol{\theta}^\star)$ is a maximum. As $\boldsymbol{\mathcal{I}}_\nu$ is a positive semi-definite matrix, it is positive definite if it is full rank.

Depending on the parameterization, there might be other values $\check{\boldsymbol{\theta}}$ which make the term of order ϵ zero. Note that, by definition, $J(\boldsymbol{\theta}) = \tilde{J}(\ln p_m(.; \boldsymbol{\theta}))$ for any $\boldsymbol{\theta}$ so that $J(\check{\boldsymbol{\theta}}) = \tilde{J}(\ln p_m(.; \check{\boldsymbol{\theta}}))$ and $J(\boldsymbol{\theta}^\star) = \tilde{J}(\ln p_m(.; \boldsymbol{\theta}^\star)) = \tilde{J}(\ln p_d)$. Now, by Theorem 1, $J(\check{\boldsymbol{\theta}}) < J(\boldsymbol{\theta}^\star)$ for a suitable noise density p_n so that J attains a global maximum at $\boldsymbol{\theta}^\star$. ■

A.3.2 Proof of the theorem

The proof of consistency goes along the same lines as the proof of consistency for MLE (see for example Wasserman, 2004, ch. 9).

Proof To prove consistency, we have to show that given $\epsilon > 0$, $P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon)$ tends to zero as $T_d \to \infty$. In what follows, it is sometimes useful to make the underlying probability space explicit and write $P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon)$ as $P(\{\omega : ||\hat{\boldsymbol{\theta}}_T(\omega) - \boldsymbol{\theta}^\star|| > \epsilon\})$.

Since, by Lemma 10, $J(\boldsymbol{\theta}^\star)$ is a global maximum, $||\boldsymbol{\theta} - \boldsymbol{\theta}^\star|| > \epsilon$ implies that there is a $\delta(\epsilon)$ such that $J(\boldsymbol{\theta}) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)$. Hence,

$$\{\omega : ||\hat{\boldsymbol{\theta}}_T(\omega) - \boldsymbol{\theta}^\star|| > \epsilon\} \subset \{\omega : J(\hat{\boldsymbol{\theta}}_T(\omega)) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)\} \tag{72}$$

and thus

$$P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon) < P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)). \tag{73}$$

Next, we investigate what happens to $P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon))$ when T_d goes to infinity. We have

$$\begin{aligned} J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T) &= J(\boldsymbol{\theta}^\star) - J_T(\boldsymbol{\theta}^\star) + J_T(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T) &(74)\\ &\leq J(\boldsymbol{\theta}^\star) - J_T(\boldsymbol{\theta}^\star) + J_T(\hat{\boldsymbol{\theta}}_T) - J(\hat{\boldsymbol{\theta}}_T) &(75)\end{aligned}$$

as $\hat{\boldsymbol{\theta}}_T$ has been defined as the argument which maximizes J_T. By use of the triangle inequality, we obtain further

$$|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| \leq |J(\boldsymbol{\theta}^\star) - J_T(\boldsymbol{\theta}^\star)| + |J_T(\hat{\boldsymbol{\theta}}_T) - J(\hat{\boldsymbol{\theta}}_T)|, \tag{76}$$

and

$$|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| \leq 2 \sup_{\boldsymbol{\theta}} |J(\boldsymbol{\theta}) - J_T(\boldsymbol{\theta})|, \tag{77}$$

from which follows that

$$P(|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| > \delta(\epsilon)) \leq P(2 \sup_{\boldsymbol{\theta}} |J(\boldsymbol{\theta}) - J_T(\boldsymbol{\theta})| > \delta(\epsilon)). \tag{78}$$

Using the assumption that $J_T(\boldsymbol{\theta})$ converges in probability uniformly over $\boldsymbol{\theta}$ to $J(\boldsymbol{\theta})$, we obtain that for sufficiently large T_d

$$P(|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| > \delta(\epsilon)) < \epsilon_2 \tag{79}$$

for any $\epsilon_2 > 0$. As $J(\boldsymbol{\theta}^\star) > J(\boldsymbol{\theta})$ for any $\boldsymbol{\theta}$, we have thus the result that

$$P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)) < \epsilon_2 \tag{80}$$

for any $\epsilon_2 > 0$. The probability $P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon))$ can thus be made arbitrarily small by choosing T_d large enough. Combining this result with Eq. (73), we conclude that $P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon)$ tends to zero as $T_d \to \infty$. ■

A.4 Proof of Theorem 3 (asymptotic normality)

A.4.1 Lemmata

In the following lemma, we use the definitions of the score function $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta})$ and $\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}; \boldsymbol{\theta}^\star)$, as well as the definition of the Hessian $\mathbf{H}_G$ which were given in Lemma 9 and Lemma 10.

Lemma 11

$$0 = \nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) + \mathbf{H}_J(\boldsymbol{\theta}^\star)(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star) + O(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2) \tag{81}$$

where

$$\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) = \frac{1}{T_d}\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t) - \nu\frac{1}{T_n}\sum_{t=1}^{T_n} h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t), \tag{82}$$

$$\begin{aligned}\mathbf{H}_J(\boldsymbol{\theta}^\star) = \frac{1}{T_d}\sum_{t=1}^{T_d}\big\{&-(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))h(\mathbf{x}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{x}_t)\mathbf{g}(\mathbf{x}_t)^T + \\ &(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{H}_G(\mathbf{x}_t;\boldsymbol{\theta}^\star)\big\} - \\ \nu\frac{1}{T_n}\sum_{t=1}^{T_n}\big\{&(1-h(\mathbf{y}_t;\boldsymbol{\theta}^\star))h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\mathbf{g}(\mathbf{y}_t)^T + \\ &h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{H}_G(\mathbf{y}_t;\boldsymbol{\theta}^\star)\big\}.\end{aligned} \tag{83}$$

Proof Using the chain rule, it follows from the relations in Section A.1 that

$$\nabla_{\boldsymbol{\theta}} \ln h(\mathbf{x}_t;\boldsymbol{\theta}) = (1-h(\mathbf{x}_t;\boldsymbol{\theta}))\mathbf{g}(\mathbf{x}_t;\boldsymbol{\theta}) \tag{84}$$

$$\nabla_{\boldsymbol{\theta}} \ln\left[1-h(\mathbf{y}_t;\boldsymbol{\theta})\right] = -h(\mathbf{y}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{y}_t;\boldsymbol{\theta}). \tag{85}$$

The derivative $\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta})$ of $J_T(\boldsymbol{\theta})$, defined in Eq. (14) as

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\sum_{t=1}^{T_d}\ln h(\mathbf{x}_t;\boldsymbol{\theta}) + \nu\frac{1}{T_n}\sum_{t=1}^{T_n}\ln\left[1-h(\mathbf{y}_t;\boldsymbol{\theta})\right], \tag{86}$$

is

$$\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}))\mathbf{g}(\mathbf{x};\boldsymbol{\theta}) - \nu\frac{1}{T_n}\sum_{t=1}^{T_n} h(\mathbf{y}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{y}_t;\boldsymbol{\theta}). \tag{87}$$

As $\hat{\boldsymbol{\theta}}_T$ is the value of $\boldsymbol{\theta}$ which maximizes $J_T(\boldsymbol{\theta})$, we must have $\nabla_{\boldsymbol{\theta}} J_T(\hat{\boldsymbol{\theta}}_T) = 0$. Doing a Taylor series around $\hat{\boldsymbol{\theta}}_T$, we have

$$0 = \nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) + \mathbf{H}_J(\boldsymbol{\theta}^\star)(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star) + O((||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2). \tag{88}$$

Half of the lemma is proved when $\nabla_{\boldsymbol{\theta}} J_T$ is evaluated at $\boldsymbol{\theta}^\star$. To prove the other half, we need to calculate the Hessian $\mathbf{H}_J$ at $\boldsymbol{\theta}^\star$. The k-th row of the Hessian $\mathbf{H}_J(\boldsymbol{\theta})$ is $\nabla_{\boldsymbol{\theta}} F_k(\boldsymbol{\theta})^T$ where F_k is the k-th element of the vector $\nabla_{\boldsymbol{\theta}} J_T$. Denoting by g_k is the k-th element of the score function $\mathbf{g}$, we have

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} F_k(\boldsymbol{\theta}) = \frac{1}{T_d}\sum_{t=1}^{T_d}&\{-\nabla_{\boldsymbol{\theta}} h(\mathbf{x}_t;\boldsymbol{\theta})g_k(\mathbf{x}_t;\boldsymbol{\theta}) + (1-h(\mathbf{x}_t;\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}} g_k(\mathbf{x}_t;\boldsymbol{\theta})\} \\ -\nu\frac{1}{T_n}\sum_{t=1}^{T_n}&\{\nabla_{\boldsymbol{\theta}} h(\mathbf{y}_t;\boldsymbol{\theta})g_k(\mathbf{y}_t;\boldsymbol{\theta}) + h(\mathbf{y}_t;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} g_k(\mathbf{x}_t;\boldsymbol{\theta})\}.\end{aligned} \tag{89}$$

Using the chain rule, it follows from the relations in Section A.1 that

$$\nabla_{\boldsymbol{\theta}} h(\mathbf{u};\boldsymbol{\theta}) = (1-h(\mathbf{u};\boldsymbol{\theta}))h(\mathbf{u};\boldsymbol{\theta})\mathbf{g}(\mathbf{u};\boldsymbol{\theta}). \tag{90}$$

Hence,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} F_k(\boldsymbol{\theta}) &= \frac{1}{T_d}\sum_{t=1}^{T_d}\{-(1-h(\mathbf{x}_t;\boldsymbol{\theta}))h(\mathbf{x}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{x}_t;\boldsymbol{\theta})g_k(\mathbf{x}_t;\boldsymbol{\theta})+ \\ &\quad (1-h(\mathbf{x}_t;\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}} g_k(\mathbf{x}_t;\boldsymbol{\theta})\} - \\ &\quad \nu\frac{1}{T_n}\sum_{t=1}^{T_n}\{(1-h(\mathbf{y}_t;\boldsymbol{\theta}))h(\mathbf{y}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{y}_t;\boldsymbol{\theta})g_k(\mathbf{y}_t;\boldsymbol{\theta})+ \\ &\quad h(\mathbf{y}_t;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} g_k(\mathbf{y}_t;\boldsymbol{\theta})\}, \end{aligned} \tag{91}$$

which proves the lemma. ■

For the next lemma, recall the definition of $\boldsymbol{\mathcal{I}}_\nu$ given in Lemma 10 or Theorem 2.

Lemma 12 $\mathbf{H}_J(\boldsymbol{\theta}^\star)$ *converges in probability to* $-\boldsymbol{\mathcal{I}}_\nu$ *as the sample size* T_d *tends to infinity.*

Proof As $T_n = \nu T_d$, T_n also tends to infinity when T_d tends to infinity. As the sample sizes become arbitrarily large, the sample averages become integration over the corresponding densities so that

$$\begin{aligned}\lim_{T_d\to\infty}\mathbf{H}_J(\boldsymbol{\theta}^\star) &\xrightarrow{P} \int -(1-h(\mathbf{x};\boldsymbol{\theta}^\star))h(\mathbf{x};\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^T p_d(\mathbf{x})\mathrm{d}\mathbf{x} + \\ &\quad \int (1-h(\mathbf{x};\boldsymbol{\theta}^\star))\mathbf{H}_G(\mathbf{x};\boldsymbol{\theta}^\star)p_d(\mathbf{x})\mathrm{d}\mathbf{x} - \\ &\quad \int (1-h(\mathbf{y};\boldsymbol{\theta}^\star))h(\mathbf{y};\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y})\mathbf{g}(\mathbf{y})^T \nu p_n(\mathbf{y})\mathrm{d}\mathbf{y} - \\ &\quad \int h(\mathbf{y};\boldsymbol{\theta}^\star)\mathbf{H}_G(\mathbf{y};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{y})\mathrm{d}\mathbf{y}. \end{aligned} \tag{92}$$

Reordering of the terms and changing the names of the integration variables to $\mathbf{u}$ gives

$$\begin{aligned}\lim_{T_d\to\infty}\mathbf{H}_J(\boldsymbol{\theta}^\star) &\xrightarrow{P} -\int (1-h(\mathbf{u};\boldsymbol{\theta}^\star))h(\mathbf{u};\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T(p_d(\mathbf{u})+\nu p_n(\mathbf{u}))\mathrm{d}\mathbf{u} + \\ &\quad \int \left((1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) - h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u})\right)\mathbf{H}_G(\mathbf{u};\boldsymbol{\theta}^\star)\mathrm{d}\mathbf{u}. \end{aligned} \tag{93}$$

With Eq. (51) and Eq. (52), we have

$$(1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) = h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u}), \tag{94}$$

$$(1-h(\mathbf{u};\boldsymbol{\theta}^\star))h(\mathbf{u};\boldsymbol{\theta}^\star)(p_d(\mathbf{u})+\nu p_n(\mathbf{u})) = \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u})+\nu p_n(\mathbf{u})}. \tag{95}$$

Hence,

$$\lim_{T_d\to\infty}\mathbf{H}_J(\boldsymbol{\theta}^\star) \xrightarrow{P} -\int \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u})+\nu p_n(\mathbf{u})}\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T\mathrm{d}\mathbf{u}, \tag{96}$$

which is $-\boldsymbol{\mathcal{I}}_\nu$. ■

Lemma 13 *The expectation* $\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$ *is zero.*

Proof We calculate

$$\begin{aligned}
\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d}\sum_{t=1}^{T_d} \mathrm{E}\,\mathbf{g}(\mathbf{x}_t)(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star)) - \\
&\quad \nu\frac{1}{T_n}\sum_{t=1}^{T_n} \mathrm{E}\,\mathbf{g}(\mathbf{y}_t)h(\mathbf{y}_t;\boldsymbol{\theta}^\star) \qquad (97)\\
&= \mathrm{E}\,\mathbf{g}(\mathbf{x})(1-h(\mathbf{x};\boldsymbol{\theta}^\star)) - \nu\,\mathrm{E}\,\mathbf{g}(\mathbf{y})h(\mathbf{y};\boldsymbol{\theta}^\star) \qquad (98)\\
&= \int \mathbf{g}(\mathbf{u})(1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u})\mathrm{d}\mathbf{u} - \\
&\quad \nu\int \mathbf{g}(\mathbf{u})h(\mathbf{u};\boldsymbol{\theta}^\star)p_n(\mathbf{u})\mathrm{d}\mathbf{u}, \qquad (99)
\end{aligned}$$

where the second equality follows from the iid assumption of the sample X and Y, respectively. Reordering leads to

$$\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) = \int \mathbf{g}(\mathbf{u})\left((1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) - h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u})\right)\mathrm{d}u, \qquad (100)$$

which is, with Eq. (94), zero. ■

Lemma 14 *The variance* $\mathrm{Var}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$ *is*

$$\frac{1}{T_d}\left(\boldsymbol{\mathcal{I}}_\nu - \left(1+\frac{1}{\nu}\right)\mathrm{E}(P_\nu\mathbf{g})\,\mathrm{E}(P_\nu\mathbf{g})^T\right),$$

where $\boldsymbol{\mathcal{I}}_\nu$*,* P_ν *and* $\mathbf{g}$ *were defined in Lemma 10, and the expectation is taken over the data pdf* p_d*.*

Proof As the expectation $\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$ is zero, the variance is given by $\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)^T$. Multiplying out gives

$$\begin{aligned}
\mathrm{Var}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)^T\right] - \\
&\quad \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)^T\right] - \\
&\quad \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)^T\right] + \\
&\quad \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)^T\right]. \qquad (101)
\end{aligned}$$

Since the samples are all independent from each other, we have

$$\begin{aligned}
\operatorname{Var} \nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d^2} \sum_{t=1}^{T_d} \mathrm{E}\left[(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))^2 \mathbf{g}(\mathbf{x}_t)\mathbf{g}(\mathbf{x}_t)^T\right] + \\
&\quad \frac{1}{T_d^2} \sum_{\substack{t,\tau=1 \\ t\neq\tau}}^{T_d} \mathrm{E}\left[(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\right] \mathrm{E}\left[(1-h(\mathbf{x}_\tau;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_\tau)^T\right] - \\
&\quad \frac{1}{T_d^2} \sum_{t=1}^{T_d}\sum_{\tau=1}^{T_n} \mathrm{E}\left[(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\right] \mathrm{E}\left[h(\mathbf{y}_\tau;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_\tau)^T\right] - \\
&\quad \frac{1}{T_d^2} \sum_{t=1}^{T_n}\sum_{\tau=1}^{T_d} \mathrm{E}\left[h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\right] \mathrm{E}\left[(1-h(\mathbf{x}_\tau;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_\tau)^T\right] + \\
&\quad \frac{1}{T_d^2} \sum_{\substack{t,\tau=1 \\ t\neq\tau}}^{T_n} \mathrm{E}\left[h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\right] \mathrm{E}\left[h(\mathbf{y}_\tau;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_\tau)^T\right] + \\
&\quad \frac{1}{T_d^2} \sum_{t=1}^{T_n} \mathrm{E}\left[h(\mathbf{y}_t;\boldsymbol{\theta}^\star)^2 \mathbf{g}(\mathbf{y}_t)\mathbf{g}(\mathbf{y}_t)^T\right]. \qquad (102)
\end{aligned}$$

As we assume that all $\mathbf{x}_t$, and also $\mathbf{y}_t$, are identically distributed, the above expression simplifies to

$$\begin{aligned}
\operatorname{Var} \nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d} \int (1-h(\mathbf{u};\boldsymbol{\theta}^\star))^2 \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T p_d(\mathbf{u})\mathrm{d}\mathbf{u} + \\
&\quad \frac{T_d^2 - T_d}{T_d^2}\mathbf{m}_x\mathbf{m}_x^T - \frac{T_d T_n}{T_d^2}\mathbf{m}_x\mathbf{m}_y^T - \\
&\quad \frac{T_d T_n}{T_d^2}\mathbf{m}_y\mathbf{m}_x^T + \frac{T_n^2 - T_n}{T_d^2}\mathbf{m}_y\mathbf{m}_y^T + \\
&\quad \frac{T_n}{T_d^2} \int h(\mathbf{u};\boldsymbol{\theta}^\star)^2 \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T p_n(\mathbf{u})\mathrm{d}\mathbf{u}, \qquad (103)
\end{aligned}$$

where

$$\mathbf{m}_x = \int (1-h(\mathbf{u};\boldsymbol{\theta}^\star))\mathbf{g}(u)p_d(\mathbf{u})\mathrm{d}\mathbf{u}, \qquad (104)$$

$$\mathbf{m}_y = \int h(\mathbf{u};\boldsymbol{\theta}^\star)\mathbf{g}(u)p_n(\mathbf{u})\mathrm{d}\mathbf{u}. \qquad (105)$$

Denoting by A the sum of the first and last line of Eq. (103), we have

$$A = \frac{1}{T_d} \int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T \left[(1-h(\mathbf{u};\boldsymbol{\theta}^\star))^2 p_d(\mathbf{u}) + h(\mathbf{u};\boldsymbol{\theta}^\star)^2 \nu p_n(\mathbf{u})\right] \mathrm{d}\mathbf{u} \qquad (106)$$

since $T_n = \nu T_d$. Now, Eq. (50) and $p_m(\mathbf{u};\boldsymbol{\theta}^\star) = p_d(\mathbf{u})$ imply that

$$\begin{aligned}
(1-h(\mathbf{u};\boldsymbol{\theta}^\star))^2 p_d(\mathbf{u}) + h(\mathbf{u};\boldsymbol{\theta}^\star)^2 \nu p_n(\mathbf{u}) &= \frac{\nu p_n(\mathbf{u}) p_d(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})} \qquad (107) \\
&= P_\nu p_d(\mathbf{u}), \qquad (108)
\end{aligned}$$

so that

$$A = \frac{1}{T_d}\int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu p_d(\mathbf{u})\mathrm{d}\mathbf{u} \quad (109)$$
$$= \frac{1}{T_d}\boldsymbol{\mathcal{I}}_\nu. \quad (110)$$

Denote by B the second line of Eq. (103). Rearranging the terms, we have

$$B = \mathbf{m}_x \int \left[(1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) - h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u})\right]\mathbf{g}(\mathbf{u})^T\mathrm{d}\mathbf{u} - \frac{1}{T_d}\mathbf{m}_x\mathbf{m}_x^T. \quad (111)$$

Again, Eq. (50) and $p_m(\mathbf{u};\boldsymbol{\theta}^\star) = p_d(\mathbf{u})$ imply that

$$(1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) = h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u}) \quad (112)$$
$$= \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u})+\nu p_n(\mathbf{u})} \quad (113)$$
$$= P_\nu p_d(\mathbf{u}), \quad (114)$$

so that the first line in Eq. (111) is zero and

$$\mathbf{m}_x = \int P_\nu \mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u}. \quad (115)$$

The term B is thus

$$B = -\frac{1}{T_d}\int P_\nu \mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u}\int P_\nu \mathbf{g}(\mathbf{u})^T p_d(\mathbf{u})\mathrm{d}\mathbf{u}. \quad (116)$$

Denote by C the third line of Eq. (103). Rearranging the terms, we have with $T_n = \nu T_d$

$$C = -\frac{\nu}{T_d}\mathbf{m}_y\mathbf{m}_y^T + \nu\mathbf{m}_y(\nu\mathbf{m}_y^T - \mathbf{m}_x^T). \quad (117)$$

The term $\nu\mathbf{m}_y$ is with Eq. (50) and $p_m(\mathbf{u};\boldsymbol{\theta}^\star) = p_d(\mathbf{u})$

$$\nu\mathbf{m}_y = \int P_\nu \mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u}, \quad (118)$$

so that $\nu\mathbf{m}_y = \mathbf{m}_x$, and hence

$$C = -\frac{1}{\nu T_d}(\nu\mathbf{m}_y)(\nu\mathbf{m}_y^T) \quad (119)$$
$$= \frac{1}{\nu}B. \quad (120)$$

All in all, the variance $\mathrm{Var}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$ is thus

$$\mathrm{Var}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) = A + B + C \quad (121)$$
$$= \frac{1}{T_d}\left(\boldsymbol{\mathcal{I}}_\nu - (1+\frac{1}{\nu})\,\mathrm{E}\left(P_\nu\mathbf{g}\right)\mathrm{E}\left(P_\nu\mathbf{g}^T\right)\right), \quad (122)$$

where

$$\mathrm{E}\left(P_\nu \mathbf{g}\right) = \int P_\nu \mathbf{g}(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}. \tag{123}$$

■

A.4.2 Proof of the theorem

We are now ready to give the proof of Theorem 3.

Proof Up to terms of order $O(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2)$, we have with Lemma 11

$$\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star) = -\mathbf{H}_J^{-1}\sqrt{T_d}\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star). \tag{124}$$

By Lemma 12, $\mathbf{H}_J \xrightarrow{P} -\boldsymbol{\mathcal{I}}_\nu$ for large sample sizes T_d. Using Lemma 13 and Lemma 14, we see that

$$\sqrt{T_d}\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$$

converges in distribution to a normal distribution of mean zero and covariance matrix

$$\boldsymbol{\mathcal{I}}_\nu - \left(1 + \frac{1}{\nu}\right) \mathrm{E}(P_\nu \mathbf{g})\, \mathrm{E}(P_\nu \mathbf{g})^T,$$

which implies that $\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$ converges in distribution to a normal distribution of mean zero and covariance matrix $\boldsymbol{\Sigma}$,

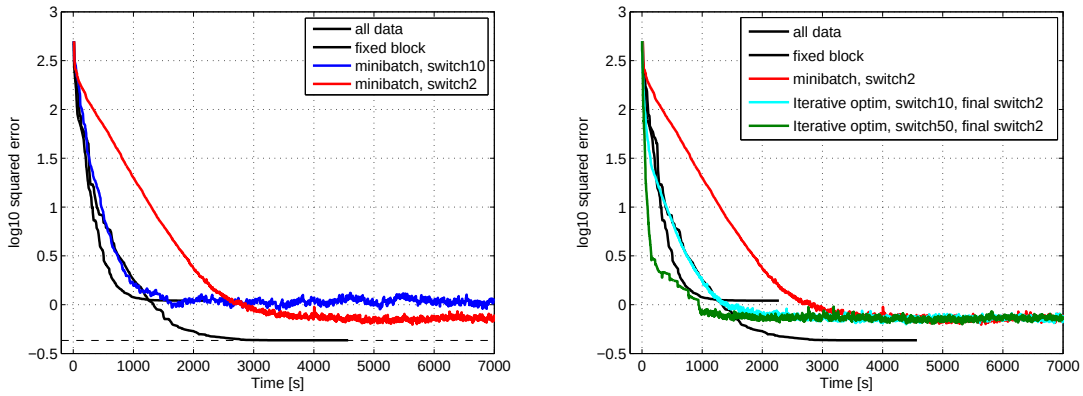
$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{I}}_\nu^{-1} - \left(1 + \frac{1}{\nu}\right) \boldsymbol{\mathcal{I}}_\nu^{-1}\, \mathrm{E}(P_\nu \mathbf{g})\, \mathrm{E}(P_\nu \mathbf{g})^T \boldsymbol{\mathcal{I}}_\nu^{-1}. \tag{125}$$

■

Appendix B. Reducing computation time with minibatches and an iterative increase of noise sample size

The objective function J_T in Eq. (13) is defined through an sample average. In an iterative optimization scheme, often not all the data is used to compute the average. The reason for using minibatches can lie in memory considerations or in the desire to speed up the computations. We analyze here what statistical cost (reduction of estimation accuracy) such a optimization scheme implies. Furthermore, we show that optimizing J_T for increasingly larger values of ν reduces computation time without affecting estimation accuracy. The presented strategy is for optimization with the nonlinear conjugate gradient algorithm of Rasmussen (2006). Truncated-Newton methods are an alternative to nonlinear conjugate gradient algorithms (Nash, 2000; Martens, 2010). The presented optimization strategy might also be applicable for that class of optimization schemes.

As working example, we consider the unnormalized Gaussian distribution of Subsection 3.1 for $n = 40$. Estimating the information matrix and the normalizing parameter means estimating 821 parameters. We use $T_d = 50000$, and $\nu = 10$. We assume further



(a) Increasing accuracy

(b) Increasing accuracy and speed of convergence

Figure 18: Optimization with minibatches and an iterative increase of noise sample size reduces computation time.

that, for whatever reason, it is not feasible to work with all the data points at the same time but only with $\tilde{T}_d = 25000$ samples (although for the present example, it is of course possible to use all the data.)

Results The lower black curve in Figure 18(a) shows the performance for the hypothetical situation when we could use all the data. The mean squared error (MSE) reaches the level which Corollary 4 predicts (dashed curve). This is the lowest error than can be obtained with noise-contrastive estimation for $\nu = 10$ and $T_d = 50000$. The upper black curve in the same figure shows the MSE when only $\tilde{T}_d = 25000$ data points are used in the optimization. This clearly leads to less precise estimates. The performance can, however, be improved by randomly choosing a new minibatch of size $\tilde{T}_d$ after two updates of the parameters (red curve). The improved performance comes, however, at the cost of slowing down convergence. If the minibatch is switched at a lower rate, e.g. after 10 updates, the speed of convergence stays the same but the accuracy does not improve (blue curve).

Figure 18(b) shows the proposed optimization strategy, which we also use in Section 5 for the simulations with natural image data: We iteratively optimize J_T for increasingly larger values of ν. Whenever we increase ν to $\nu + 1$, we also take a new minibatch. When ν reaches its maximal value, which is here $\nu = 10$, we switch the minibatch after two parameter updates. For the other values of ν, we switch the minibatches at a lower rate (after 50 iterations for the results shown as green curve, after 10 iterations for the blue curve). This allows to speed up convergence while achieving the same precision as in the optimization based on minibatches alone (red curve in Figure (a) and (b)).