Optimal Kernels for Unsupervised Learning

Sepp Hochreiter and Klaus Obermayer
Bernstein Center for Computational Neuroscience
and
Technische Universität Berlin
10587 Berlin, Germany
{hochreit,oby}@cs.tu-berlin.de

***Abstract*— We investigate the optimal kernel for sample-based model selection in unsupervised learning if maximum likelihood approaches are intractable. Given a set of training data and a set of data generated by the model, two kernel density estimators are constructed. A model is selected through gradient descent w.r.t. the model parameters on the integrated squared difference between the density estimators. Firstly we prove that convergence is optimal, i.e. that the cost function has only one global minimum w.r.t. the locations of the model samples, if and only if the kernel in the reparametrized cost function is a Coulomb kernel. As a consequence, Gaussian kernels commonly used for density estimators are suboptimal. Secondly we show that the absolute value of the difference between model and reference density convergences at least with $1/t$. Finally, we apply the new methods to distribution free ICA and to nonlinear ICA.**

I. Introduction

Unsupervised learning methods are often based on the so-called generative model approach. In this approach, one usually considers a parameterized family of probability distributions for the observable data. Model selection is typically performed using the likelihood of the training data or the Bayes posterior as a selection criterion. Examples for generative approaches are abundant, ranging from factor analysis [6], ICA [5], mixture models [8], and Boltzmann machines [3].

Here we consider classes of generative models, where a set $\boldsymbol{z}$ of hidden causes is responsible for the generation of an observation $\boldsymbol{x}$ (Fig. 1). The hidden causes assume

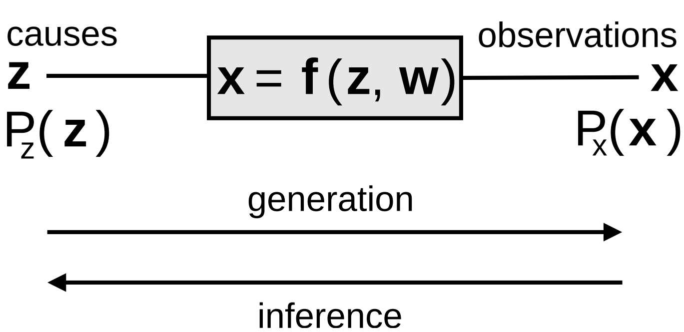

Fig. 1. The generative model framework.

values $\boldsymbol{z}$ according to a probability distribution $P_z(\boldsymbol{z})$. Every cause vector $\boldsymbol{z}$ is then transformed via a nonlinear function $f(\boldsymbol{z}, \boldsymbol{w})$, parameterized by a weight vector $\boldsymbol{w}$, to generate an observation $\boldsymbol{x}$. For the following we denote the distribution of the generated observations by $P_x(\boldsymbol{x})$. If a model has been selected, then its most common application is inference, i.e. the reconstruction of the source values $\boldsymbol{z}$ which have generated an observation $\boldsymbol{x}$. In order to unambiguously infer the source values, however, the inverse function $f^{-1}(\boldsymbol{z}, \boldsymbol{w})$ must exist and must be computable.

Model selection is often performed using a maximum likelihood method. In order to select the optimal set of parameters, the likelihood of an observation $\boldsymbol{x}$,

$$P_x(\boldsymbol{x}) \ = \ \int d\boldsymbol{z} \ \delta\left(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{z}, \boldsymbol{w})\right) \ P_z(\boldsymbol{z}) \ , \qquad (1)$$

is calculated and the likelihood of the full set $\{\boldsymbol{x}^i\}$, $i = 1, ..., N$ of observations, $L = \prod_{i=1}^{N} P_x(\boldsymbol{x}^i$, is maximized. If an inverse function exists and can be analytically calculated, then the integral in eq. (1) can be evaluated and one obtains

$$P_x(\boldsymbol{x}) \ = \ \left| \frac{d\boldsymbol{f}^{-1}}{d\boldsymbol{x}} \right| \ P_z\left(\boldsymbol{f}^{-1}(\boldsymbol{x})\right) \ . \qquad (2)$$

The straightforward application of the maximum likelihood (ML) method therefore requires the knowledge of the inverse function $f^{-1}(\boldsymbol{x}, \boldsymbol{w})$. Since the inverse function is also necessary for inference one may argue that it may be more adequate to parameterize the inverse function f^{-1} rather than the function f which describes the data generation process. This is, for example, done in many ICA applications. However, there exist problems (i) for which the inverse function does not exist (many-to-one mappings, e.g. in the case of incomplete measurements) or (ii) for which prior knowledge exists only for the generation process. For those cases, ML methods either fail or may be computationally intractable. There is another potential pitfall for the ML methods. Even if the inverse function is given, evaluation of eq. (2) requires knowledge of the cause densities $P_z(\boldsymbol{z})$. If they are not known or only partially known one has to resort to approximations (as in ICA).

In order to overcome above mentioned problems we suggest to use a sample based method for model selection. Let us first consider the case that the source densities are known but that the generative function f cannot easily be inverted. In this case we suggest to generate a sample $\{\boldsymbol{z}\}$ of causes, and to adjust the parameters $\boldsymbol{w}$ of the generative function f such that the location of the corresponding sample $\{\boldsymbol{x}\}$ of observations aligns with the observed data as good as possible. For the case that the source densities are only partially known, but an inverse function f^{-1} can be constructed, we suggest to generate a set $\{\boldsymbol{z}\}$ of sources by application of the inverse function f^{-1} to the set $\{\boldsymbol{x}\}$ observations. The parameters of the inverse function must then be adjusted in a way, that the set of sources aligns with a set of reference sources as good as possible. These two scenarios are depicted in Fig. 2.

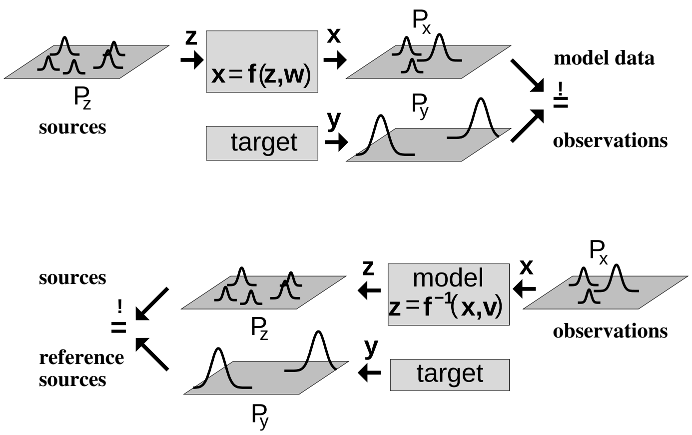

Fig. 2. Sample based methods for model selection. Top: The source densities are given, but f cannot be inverted. Model samples $\boldsymbol{x}$ are generated through cause samples $\boldsymbol{z}$. Bottom: f^{-1} can be computed and used for inference, but P_z is not fully known.

But what is the optimal way of doing the alignment? Here we suggest to endow the data points of the two sets with positive and negative electric charges, and to use a learning dynamics driven by Coulomb forces to move the generated data points to their correct position. Note, that movement is not "free" but constrained by the underlying changes in the model parameters $\boldsymbol{w}$ and $\boldsymbol{v}$. We show that this procedure corresponds to the minimization of the quadratic difference between two kernel density estimators for the densities $P_z(\boldsymbol{z})$ and $P_x(\boldsymbol{x})$, which are constructed from the sample locations. We then prove, that the choice of Coulomb's law is optimal in the sense, that the quadratic difference has only one global minimum w.r.t. the locations of the model samples. We finally show that the method provides excellent results when applied to ICA problems. Note that due to above mentioned optimality properties the Coulomb interaction is way superior compared to interactions derived from a standard Gaussian kernel.

II. Cost Functions and Optimization

Let us consider two sets of samples $\boldsymbol{x}^i$, $i = 1...N_x$, drawn from $p_x(.)$, and $\boldsymbol{y}^i$, $i = 1...N_y$, drawn from $p_y(.)$. We construct kernel density estimators (KDE) $\hat{p}_y$ and $\hat{p}_x$ using a kernel $k_d(.,.)$, because we assume that the true distributions are unknown or cannot be evaluated. Model selection, i.e. the selection of the parameters $\boldsymbol{w}$, is then performed minimizing the integrated squared difference (ISD) $\tilde{F}$ between both estimators:

$$\tilde{F}(k_d) \;=\; \tilde{F}\left(\hat{p}_y(.;k_d), \hat{p}_x(.;k_d)\right) \;=\; \int_T \Phi^2(\boldsymbol{a};k_d)d\boldsymbol{a}, \quad (3)$$

where

$$\Phi(\boldsymbol{a};k_d) \;:=\; \hat{p}_y(\boldsymbol{a};k_d) \;-\; \hat{p}_x(\boldsymbol{a};k_d) \;= \quad (4)$$
$$\frac{1}{N_y}\sum_{i=1}^{N_y} k_d\left(\boldsymbol{a},\boldsymbol{y}^i\right) \;-\; \frac{1}{N_x}\sum_{i=1}^{N_x} k_d\left(\boldsymbol{a},\boldsymbol{x}^i\right).$$

In the following, we will call Φ the **potential function**. If $\tilde{F} = 0$ then the estimate of the model output distribution is equal to the estimate of the reference distribution, and our goal of learning is reached.

Minimization of eq. (3), however, requires the evaluation of an integral which could be computationally expensive. We, therefore, define another kernel $k(.,.)$,

$$k\left(\boldsymbol{a},\boldsymbol{b}\right) \;=\; \int_T k_d\left(\boldsymbol{a},\boldsymbol{c}\right) k_d\left(\boldsymbol{b},\boldsymbol{c}\right) d\boldsymbol{c}, \quad (5)$$

for which we obtain a simpler expression $F(k) = \tilde{F}(k_d)$,

$$F(k) \;=\; F\left(\hat{p}_y(.;k), \hat{p}_x(.;k)\right) \;:= \quad (6)$$
$$\frac{1}{2}\left(\frac{1}{N_y}\sum_{i=1}^{N_y}\Phi(\boldsymbol{y}^i;k) \;-\; \frac{1}{N_x}\sum_{i=1}^{N_x}\Phi(\boldsymbol{x}^i;k)\right) \;=$$
$$\frac{1}{2}\left(\frac{1}{N_y^2}\sum_{i=1}^{N_y}\sum_{j=1}^{N_y} k\left(\boldsymbol{y}^i,\boldsymbol{y}^j\right) \;-\; \frac{2}{N_yN_x}\sum_{i=1}^{N_y}\sum_{j=1}^{N_x} k\left(\boldsymbol{y}^i,\boldsymbol{x}^j\right) \;+\right.$$
$$\left.\frac{1}{N_x^2}\sum_{i=1}^{N_x}\sum_{j=1}^{N_x} k\left(\boldsymbol{x}^i,\boldsymbol{x}^j\right)\right).$$

In the following we will call $F(k)$ the **energy function**. We now define the positive (semi)definiteness of a kernel:

Definition 1: A kernel $k : T^2 \to \mathbb{R}$ is called **positive semidefinite**, if for all $N_x \in \mathbb{N}$ and $\boldsymbol{x}^1,\ldots,\boldsymbol{x}^{N_x} \in T$ the matrix $K : K_{ij} = k(\boldsymbol{x}^i,\boldsymbol{x}^j)$ is positive semidefinite.
If k is positive semidefinite then $F \geq 0$, and we obtain the following theorem:

Theorem 1 (Equivalence of energy and ISD): Suppose the data is contained in a subset $T \subseteq \mathbb{R}^d$. Let $k_d, k :\; T \times T \to \mathbb{R}$ be kernels for which (*) $k\left(\boldsymbol{a},\boldsymbol{b}\right) = \int_T k_d\left(\boldsymbol{a},\boldsymbol{c}\right) k_d\left(\boldsymbol{b},\boldsymbol{c}\right) d\boldsymbol{c}$. Then the equality $\tilde{F}(k_d) \;=\; F(k)$ holds if

(A) $\boldsymbol{k_d}$ given: (*) converges.
(B) $\boldsymbol{k}$ given: T is compact; k is symmetric, continuous, and positive semidefinite.

Proof (sketch). (A) is straightforward and for (B) we use Mercers theorem: $\forall \boldsymbol{a},\boldsymbol{b} \in T$ there exists an expansion $k(\boldsymbol{a},\boldsymbol{b}) \;=\; \sum_{n=1}^{\infty}\lambda_n\; e_n(\boldsymbol{a})\; e_n(\boldsymbol{b}), \;\; \forall n : \;\; \lambda_n \;\geq\; 0$, for which convergence is absolute and uniform. λ_n and e_n are the eigenvalues and eigenfunctions of the k-induced Hilbert-Schmidt operator. We define $\forall \boldsymbol{a},\boldsymbol{b} \in T \;:\;\; k_d(\boldsymbol{a},\boldsymbol{b}) \;:= \sum_{n=1}^{\infty}(\lambda_n)^{\frac{1}{2}}\;\; e_n(\boldsymbol{a})\;\; e_n(\boldsymbol{b})$. $k_d \;\in\; L^2(T \times T)$ because k induces a trace class (nuclear) operator with trace $\sum_{n=1}^{\infty}\lambda_n \;= \int_T k(\boldsymbol{a},\boldsymbol{a})d\boldsymbol{a}$ (cf. [9], p. 267). ■

Theorem 1 offers a big advantage. It says that for every symmetric, continuous, and positive semidefinite kernel $k(.,.)$

there exist a kernel k_d for the density estimate. Therefore, it suffices to select $k(.,.)$, and there is no need for performing the integration in eq. (3). But what kernel $k(.,.)$ should be selected? We will provide an answer to this question in Section III.

The cost function F can be minimized by gradient descent. We obtain

$$\Delta \boldsymbol{w} \ = \ -\epsilon \ \boldsymbol{\nabla}_{\boldsymbol{w}} F \ = \ - \ \epsilon \frac{1}{N_x} \sum_{i=1}^{N_x} \left(\frac{\partial \boldsymbol{x}^i}{\partial \boldsymbol{w}} \right)^T \ \mathbf{E}\left(\boldsymbol{x}^i\right) \ , \quad (7)$$

where ϵ is the learning rate, $\partial \boldsymbol{x}^i / \partial \boldsymbol{w}$ is the Jacobian of $\boldsymbol{x}^i = f(\boldsymbol{z}^i; \boldsymbol{w})$, and $-\boldsymbol{\nabla}_{\boldsymbol{x}^i} F = -1/N_x \ \mathbf{E}(\boldsymbol{x}^i) = = 1/N_x \ \boldsymbol{\nabla}_{\boldsymbol{x}^i} \Phi\left(\boldsymbol{x}^i\right)$ ($\Phi(.) \equiv \Phi(.;k)$). We will call $\mathbf{E}(\boldsymbol{a}) := - \ \boldsymbol{\nabla}_{\boldsymbol{a}} \Phi\left(\boldsymbol{a}\right)$ the **field** at $\boldsymbol{a}$.

III. Optimal Kernels and Convergence Properties

In order to perform the analysis of the learning rule we consider the continuous case, i.e. the case where the number of samples goes to infinity. Let $\rho(\boldsymbol{a}) := p_y(\boldsymbol{a}) - p_x(\boldsymbol{a})$ be the difference between the distributions $p_y(.)$ and $p_x(.)$ from which the samples are drawn. Then the potential and the energy is given by

$$\begin{aligned} \Phi(\boldsymbol{a}) \ &:= \ \int \rho(\boldsymbol{b}) \ k\left(\boldsymbol{a}, \boldsymbol{b}\right) \ d\boldsymbol{b} \ , \\ F\left(\rho\right) \ &:= \ \frac{1}{2} \int \ \rho(\boldsymbol{a}) \ \Phi(\boldsymbol{a}) \ d\boldsymbol{a} \ = \\ &\frac{1}{2} \int\int \ \rho(\boldsymbol{a}) \ \rho(\boldsymbol{b}) \ k\left(\boldsymbol{a}, \boldsymbol{b}\right) \ d\boldsymbol{b} \ d\boldsymbol{a} \ . \end{aligned}$$

We now consider a simpler optimization problem than stated in eq. (7). Let us consider the optimization of F as a function of the sample locations $\boldsymbol{x}$ under the assumption, that samples $\boldsymbol{x}$ can move freely and are not constrained by the underlying model $f(.;\boldsymbol{w})$. Using the continuity equation [7]

$$\dot{\rho} \ = \ -\boldsymbol{\nabla} \cdot \left(\rho \ \boldsymbol{v}\right) \quad (8)$$

for particle densities, with particles moving with "velocity" $\boldsymbol{v} = - \ \boldsymbol{\nabla}_{\boldsymbol{a}} F = \operatorname{sign}(\rho(\boldsymbol{a})) \ \mathbf{E}$, we obtain

$$\begin{aligned} \dot{\rho}(\boldsymbol{a}) \ &= \ - \operatorname{sign}(\rho(\boldsymbol{a})) \ \boldsymbol{\nabla} \cdot \left(\rho(\boldsymbol{a}) \ \mathbf{E}(\boldsymbol{a})\right) \ = \\ & \ - \ \boldsymbol{\nabla} \cdot \left(|\rho(\boldsymbol{a})| \ \mathbf{E}(\boldsymbol{a})\right) \ . \end{aligned} \quad (9)$$

Let $\|\rho\|_\infty = \max_{\boldsymbol{a}} |\rho(\boldsymbol{a})|$ be the maximum norm. In order to analyze the convergence properties we define:

Definition 2 (Uniform learning convergence): Learning **converges uniformly** if $\|\rho\|_\infty(t) \leq U(t)$ for $0 \leq t$, where U is a positive strictly monotonous decreasing function of time t with $\lim_{t \to \infty} U(t) = 0$.

At the global maximum a_{max} of $|\rho|$: $\boldsymbol{\nabla}_{\boldsymbol{a}} \rho(\boldsymbol{a}_{max}) = 0$ and eq. (10) reduces to

$$|\dot{\rho}(\boldsymbol{a}_{max})| \ = \ - \ \rho(\boldsymbol{a}_{max}) \ \boldsymbol{\nabla} \cdot \left(\mathbf{E}(\boldsymbol{a}_{max})\right) \ . \quad (10)$$

Uniform learning convergence requires that at the global maximum $\boldsymbol{a}_{max}$ $\operatorname{sign}(\dot{\rho}(\boldsymbol{a}_{max})) = - \operatorname{sign}(\rho(\boldsymbol{a}_{max}))$ and, therefore, $\operatorname{sign}(\boldsymbol{\nabla} \cdot (\mathbf{E}(\boldsymbol{a}_{max}))) = \operatorname{sign}(\rho(\boldsymbol{a}_{max}))$. The next theorem characterizes the kernels for which uniform learning convergence is obtained.

Theorem 2 (Poisson Equation): Assume that the kernel $k(\boldsymbol{a}, \boldsymbol{b}) : T \times T \rightarrow \mathbb{R}$ is continuously differentiable, symmetric, and positive definite and that $\boldsymbol{\nabla}_{\boldsymbol{a}}^2 k\left(\boldsymbol{a}, \boldsymbol{b}\right) \in L^2(T \times T)$. Assume further that forces are symmetric: $\boldsymbol{\nabla}_{\boldsymbol{a}} k(\boldsymbol{a}, \boldsymbol{b}) = - \ \boldsymbol{\nabla}_{\boldsymbol{b}} k(\boldsymbol{a}, \boldsymbol{b})$.

If uniform convergence holds for each ρ then k must be of the following form: k can be partitioned into kernels $k = \sum_l k_{U(\lambda_l)}$, where the $U(\lambda_l)$ form a partition of T and the $k_{U(\lambda_l)}$ obey the following Dirichlet problems on $U(\lambda_l)$ (Poisson equation):

$$\boldsymbol{\nabla}_{\boldsymbol{a}}^2 (- \ k_{U(\lambda_l)}(\boldsymbol{a}, \boldsymbol{b})) \ = \ \lambda_l \ \delta_{U(\lambda_l)}(\boldsymbol{a} - \boldsymbol{b}) \ , \quad (11)$$

where $\delta_{U(\lambda_l)}$ is the delta function restricted to $U(\lambda_l)$ and $0 \leq \lambda_l$.

Proof. `http://ni.cs.tu-berlin.de/~hochreit/papers/ijcnnsupplementary.pdf` provides the proof. ■

The most important outcome of Theorem 2 is that uniform convergence implies (under weak assumption on the kernel k) that k must obey eq. (11). Other kernels do not allow uniform convergence for arbitrary ρ. If a kernel is chosen, which does not fulfill eq. (11), additional local optima may be introduced, and gradient based optimization methods lead to inferior optimization results. Clearly, those kernels should be avoided. If a proper kernel is chosen it follows from uniform convergence, that the cost function F has only one global minimum w.r.t. the particle locations $\boldsymbol{x}$. All local optima of the cost function F w.r.t. the model parameters $\boldsymbol{w}$ are then only a property of the model class $f(.;\boldsymbol{w})$.

We now solve the Dirichlet problem eq. (11). For simplicity we set $U(\lambda_l) = U(\lambda) = T$. In order to obtain an unique solution we set $k(\boldsymbol{a}, \boldsymbol{b}) = 0$ for $\boldsymbol{a} \in \partial T$, where ∂T is the boundary. Let $\mathrm{S}\left(\mathcal{S}_R\right)$ denote the surface area of the d-dimensional sphere $\mathcal{S}_R$ at $\mathbf{0}$ with radius R. Then the following corollary holds:

Corollary 1 (Coulomb Kernel): If (1) $U(\lambda) = T$, (2) for all $\boldsymbol{a} \in \partial T$ the kernel is $k(\boldsymbol{a}, \boldsymbol{b}) = 0$, (3) ∂T is smooth, and (4) $T \cup \partial T$ is simple connected then there exists an unique solution k for the Dirichlet problem eq. (11). For $T = \mathbb{R}^d$ this solution is given by

$$k(\boldsymbol{a}, \boldsymbol{b}) \quad = \quad \lambda \left\{ \begin{array}{ll} - \ \mathrm{S}\left(\mathcal{S}_1\right) \ \ln(\|\boldsymbol{a} - \boldsymbol{b}\|) & d = 2 \\ \frac{1}{\mathrm{S}(\mathcal{S}_1) \ (d-2)} \ \frac{1}{\|\boldsymbol{a} - \boldsymbol{b}\|^{d-2}} & d > 2 \ . \end{array} \right.$$

Proof (sketch). The corollary follows from Theorem 2 and the properties of the Dirichlet boundary problem. $k_{U(\lambda_l)}$ is up to a constant factor the Green's function of the Laplace operator. The constraint "$T \cup \partial T$ is simple connected" assures that for any unbounded T we obtain $T = \mathbb{R}^d$, otherwise, we can enclose a region of $\mathbb{R}^d \setminus T$ by a curve in $T \cup \partial T$ through ∞. ■

The kernel k is is the basis of electrostatic and gives rise to forces between charged particles which obey Coulomb's law.

Therefore we will call k a Coulomb kernel. The next theorem addresses the speed of learning convergence, still under the assumption that there are no constraints on the motion of data points (which is true for models which are sufficiently complex). Remember, that the goal of learning was to push ρ, the difference between the model output and the reference distribution, towards zero.

Theorem 3: For the Coulomb kernel defined above the following equation holds (t denotes the time starting at $t = 0$):

$$\|\rho\|_\infty(t) \; = \; \frac{1}{\lambda \; t \; + \; \left(\|\rho\|_\infty(0)\right)^{-1}} \; . \qquad (12)$$

Proof (sketch). At the extremal points $\boldsymbol{a}$: $|\dot{\rho}(\boldsymbol{a})| \; = \; \rho(\boldsymbol{a}) \; \boldsymbol{\nabla} \cdot (\mathbf{E}(\boldsymbol{a})) \; = \; - \; \lambda \; \rho(\boldsymbol{a})^2 \; = \; - \; \lambda \; |\rho(\boldsymbol{a})|^2$. This differential equation finishes the proof. ■

The Coulomb kernel and the kernel k_R possess a weak singularity and are not positive definite. A positive definite kernel, however, can be constructed if $\|\boldsymbol{a} - \boldsymbol{b}\|$ is replaced by $\sqrt{\|\boldsymbol{a} - \boldsymbol{b}\|^2 \; + \; \epsilon}$, where ϵ is a smoothing parameter. This kernels are called Plummer kernels k_P. They are widely used in computational physics, but have recently also been introduced as a useful kernel for support vector learning [4]. Because $k_P\left(\boldsymbol{x}^i, \boldsymbol{x}^i\right)$ does not depend on $\boldsymbol{x}^i$, i.e. $\boldsymbol{\nabla}_{\boldsymbol{x}^i} k_P\left(\boldsymbol{x}^i, \boldsymbol{x}^i\right) \; = \; 0$, the learning dynamics does hardly change for small ϵ.

IV. Experiments: Independent Component Analysis

Here we apply our new sample-based method to independent component analysis (ICA [5], [1], [2]) in a framework depicted in Fig. 2, bottom. ICA is a method that builds a representation of the observed data in which the statistical dependence between the components is minimal. ICA methods assume that the observed data have been generated by a *linear* mixing process of source signals which are assumed to be statistically independent from one another. The source signals should then be recovered by the ICA method. Linear ICA approaches estimate the inverse of the mixing matrix where an independence criterion serves as objective.

Standard ICA algorithms rely on certain properties of the source densities, e.g. that they are unimodal, super-Gaussian or that they have mean zero. Our approach, however, generalizes these ICA approaches because it is distribution independent. It thus extends the application of ICA methods to a broader range of real world problems.

In the new ICA method we repeat following steps: (1.) Compute model output $\boldsymbol{z}$ from observations $\boldsymbol{x}$. (2.) Draw the target source samples $\boldsymbol{y}$. (3.) Compute electric field $\mathbf{E}$. (4.) Use field $\mathbf{E}$ and eq. (7) to compute $\Delta \boldsymbol{w}$. (5.) update the weights. Step 2. draws a sample from a distribution where the components are statistically independent. This reference (target) distribution is constructed to be the product of the marginal distributions of the model's causes $\boldsymbol{z}$. In our numerical simulations we randomly recombine components of $\boldsymbol{z}$ to generate samples $\boldsymbol{y}$ from the reference distribution, i.e. each component of $\boldsymbol{y}$ was obtained from an independently, randomly chosen $\boldsymbol{z}$. Because the choice of one component of $\boldsymbol{y}$ is independent from the choice of the other components we generate a proper reference distribution.

A. Sub-Gaussian Source Distribution

Standard independent component analysis methods work well for super-Gaussian (peaky) source distributions. Our distribution free algorithm, however, is also suited for sub-Gaussian source distributions, like the multimodal source distributions used in two experiments of this section.

1) 3-D Sub-Gaussian sources: The source distributions are $x_1 \; \sim \; \frac{1}{2} \; N(1.4, 0.05) \; + \; \frac{1}{2} \; N(-0.8, 0.05)$, $x_2 \; \sim \; \frac{1}{3} \; N(1.5, 0.05) + \frac{2}{3} \; N(-1.5, 0.05)$, $x_3 \; \sim \; \frac{1}{3} \; N(1.8, 0.05) + \frac{1}{3} \; N(0.4, 0.05) + \frac{1}{3} \; N(-1.1, 0.05)$. These source distribution are mixed through a linear, randomly generated mixing matrix (matrix entries are from a uniform distribution on [-1,1]). We then trained a linear demixing model with 1000 fixed examples and a learning rate of 0.00001 for 1000 epochs. Figure 3 shows the sources, mixtures, and recovered sources. The demixing result was almost perfect which is indicated by the product of the mixing matrix with the demixing matrix which is close to a identity matrix subject to permutation and scaling.

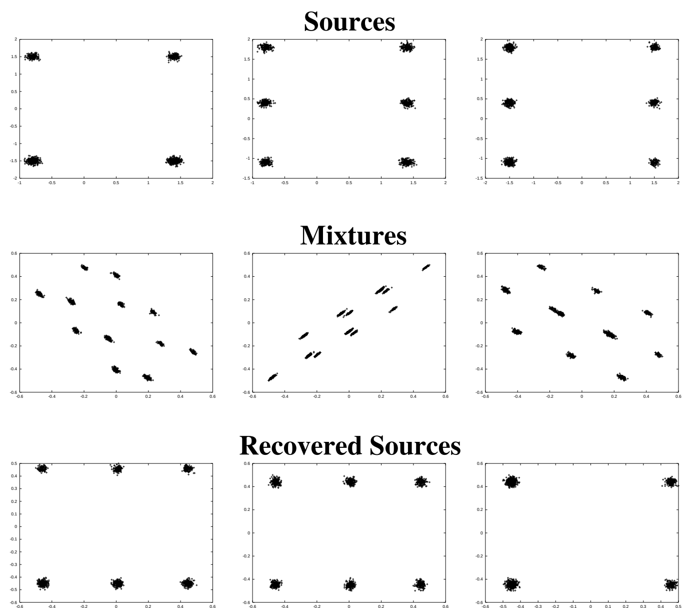

Fig. 3. Demixing a 3-D mixture of sub-Gaussian distributions. Sources (first row), mixtures (second row), and recovered sources (third row) are projected on a 2-D plane for visualization.

The fixed demixing matrix multiplied with the mixing matrix gives:

0.0010	-0.0008	**0.3117**
-0.0003	**0.3024**	-0.0007
0.4039	0.0002	-0.0005

2) 4-D Sub-Gaussian sources: In another experiment we mixed multimodal normal distributions with super-Gaussians. The demixing model and the parameters of the learning rule were exactly as in the previous experiment. The sources are: $x_1 \; \sim \; \frac{1}{2} \; N(0.4, 0.2) \; + \; \frac{1}{2} \; N(-0.8, 0.2)$, $x_2 \; \sim \; \frac{1}{3} \; N(0.4, 0.1) \; + \; \frac{2}{3} \; N(-0.3, 0.1)$, $x_3 \; = \; \mathrm{sign}(y_1) y_1^4$; $y_1 \; \sim \; N(1, 2)$; $x_4 \; = \; y_2^3$; $y_2 \; \sim \; N(0, 1)$.

-0.0129	-0.0130	-0.0123	**-0.1473**
0.4325	0.0050	0.0124	0.0123
0.0078	**0.6470**	0.0125	0.0046
-0.0123	-0.0133	**-0.1521**	-0.0146

The mixing matrix multiplied by the demixing matrix (left) is almost a permutation matrix. Standard ICA algorithms fail at this ICA task due to the multimodal source densities.

B. Nonlinear Mixing and De-mixing

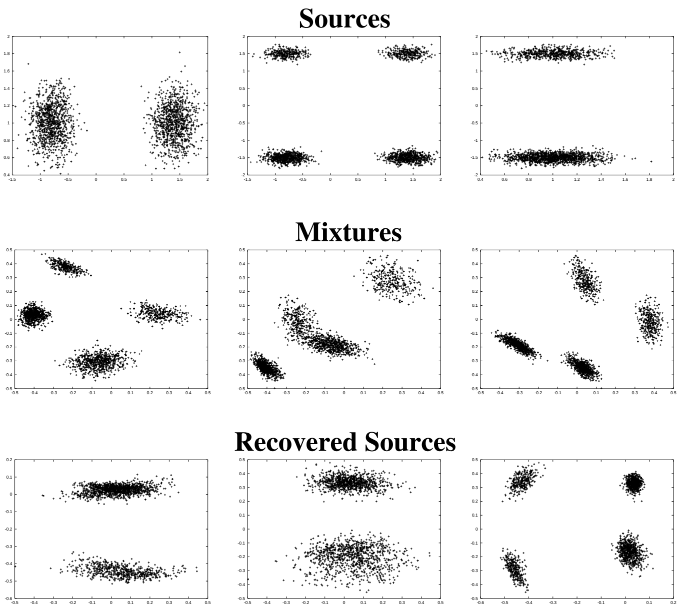

Fig. 4. Demixing a 3-D nonlinear superimposure of 3 sources. The figure shows projected sources (top row), mixtures (center row), and recovered sources (bottom row).

To demonstrate that our approach also works for nonlinear mixing problems we applied it to a 3-D mixture task. Here our goal was to extract the sources. The sources are $x_1 \sim \frac{1}{2}\, N(1.4, 0.2) + \frac{1}{2}\, N(-0.8, 0.2);\ \ x_2 \sim N(1.0, 0.2), x_3 \sim \frac{1}{3}\, N(1.5, 0.1) + \frac{2}{3}\, N(-1.5, 0.1)$.

The mixing functions f_1 to f_3 are highly nonlinear:
$f_1(\boldsymbol{x}) = \log(3 + x_1)(3x_1 + 5x_2 + 2x_3)$,
$f_2(\boldsymbol{x}) = \left(2 + \exp\left(-\frac{1}{2}x_2^2\right)\right)(-8x_1 + 4x_2 + 6x_3)$,
$f_3(\boldsymbol{x}) = \left(5 + \frac{1}{2}x_3\right)^2 (3x_1 - 7x_2 + 5x_3)$.

For demixing we used a sigmoid 3-layered neural network with 50 hidden units. We trained 100000 epochs with a learning rate of 0.0001. The results depicted in Figure 4 are good given the fact that nonlinear ICA may not have a unique solution, and the results are much better than results obtained by simple linear models if the independence is measured by the entropy. The improved performance compared to the linear model results from large weights in the nonlinear neural network, which produce useful nonlinearities for approximating the inverse mixing function.

Acknowledgments

This work was funded in part by BMBF project no. 10025304.

References

[1] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[2] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[3] G. E. Hinton and T. E. Sejnowski. Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing*, volume 1, pages 282–317. MIT Press, Cambridge, MA, 1986.

[4] S. Hochreiter, M. C. Mozer, and K. Obermayer. Coulomb classifiers: Generalizing support vector machines via an analogy to electrostatic systems. In S. Beckers, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 545–552. MIT Press, Cambridge, MA, 2003.

[5] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[6] K. G. Jöreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32:443–482, 1967.

[7] M. Schwartz. *Principles of Electrodynamics*. Dover Publications, NY, 1987. Republication of McGraw-Hill Book 1972.

[8] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.

[9] D. Werner. *Funktionalanalysis*. Springer-Verlag, Berlin, 3. edition, 2000.