Fast Kernel-Based Independent Component Analysis*

Hao Shen†, Stefanie Jegelka†and Arthur Gretton‡

Abstract

Recent approaches to Independent Component Analysis (ICA) have used kernel independence measures to obtain highly accurate solutions, particularly where classical methods experience difficulty (for instance, sources with near-zero kurtosis). FastKICA (Fast HSIC-based Kernel ICA) is a new optimisation method for one such kernel independence measure, the Hilbert-Schmidt Independence Criterion (HSIC). The high computational efficiency of this approach is achieved by combining geometric optimisation techniques, specifically an approximate Newton-like method on the orthogonal group, with accurate estimates of the gradient and Hessian based on an incomplete Cholesky decomposition. In contrast to other efficient kernel-based ICA algorithms, FastKICA is applicable to any twice differentiable kernel function. Experimental results for problems with large numbers of sources and observations indicate that FastKICA provides more accurate solutions at a given cost than gradient descent on HSIC. Comparing with other recently published ICA methods, FastKICA is competitive in terms of accuracy, relatively insensitive to local minima when initialised far from independence, and more robust towards outliers. An analysis of the local convergence properties of FastKICA is provided.

1 Introduction

The problem of Independent Component Analysis (ICA) involves the recovery of linearly mixed, statistically independent sources, in the absence of information about the source distributions beyond their mutual independence [12, 22]. The performance of ICA algorithms thus depends on the choice of the contrast

*Copyright (c) 2008 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

†H. Shen is with the Institute for Data Processing, Technische Universität München, 80290 München, Germany. (email: hao.shen@tum.de).

‡S. Jegelka and A. Gretton are with Department of Empirical Inference for Machine Learning and Perception, Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany. (email: {Stefanie.Jegelka,Arthur.Gretton}@tuebingen.mpg.de).

function measuring the degree of statistical independence of the recovered signals, and on the optimisation technique used to obtain the estimated mixing coefficients.

Classical approaches, also referred to as parametric ICA approaches, construct their independence criteria according to certain hypothetical properties of the probability distributions, either by an explicit parametric model of these distributions via maximum likelihood [8], or by maximising certain statistics of the unmixed sources (often measures of non-Gaussianity, such as the kurtosis) [9, 22]. These approaches can therefore be less powerful than methods which explicitly model the source distributions, and can even fail completely when the modelling assumptions are not satisfied (e.g. a kurtosis-based contrast will not work for sources with zero kurtosis).

More recently, several approaches to ICA have been proposed that directly optimise nonparametric independence criteria. One option is to minimise the mutual information between the sources, as in [26, 38, 7, 10]. Another approach is to use a characteristic function-based measure of mutual independence due to Kankainen [24] based on the pairwise criterion of Feuerverger [15], which was applied to ICA in [14, 11], and to ICA with post-nonlinear mixing in [2, 3].

Finally, a variety of kernel independence criteria have been employed in ICA. These criteria measure dependence using the spectrum of a covariance operator between mappings of the variables to high dimensional feature spaces, specifically reproducing kernel Hilbert spaces (RKHSs) [31]. The various kernel independence criteria differ in the way they summarise the covariance operator spectrum, and in the normalisation they use. They include the kernel canonical correlation [5], the kernel generalised variance [5], the spectral norm of the covariance operator (COCO) [19], the kernel mutual information [19], and the Hilbert-Schmidt Independence Criterion (HSIC) [18]. A biased empirical estimate of the HSIC statistic is in fact identical (as a function of its kernel argument) to the characteristic function-based criterion of [15], which is in turn identical to the ℓ_2 distance between Parzen window estimates of the joint density and the product of the marginals: see Rosenblatt [30]. When a Gaussian kernel is used and the sample size is fixed, the three statistics correspond exactly. As pointed out elsewhere [15, 24], however, the characteristic function-based statistic is more general than Rosenblatt's, since it admits a wider range of kernels while remaining an independence measure (a further difference is that the kernel bandwidth may remain fixed for increasing sample size). Likewise, there exist universal kernels (in the sense of [37]: that is, kernels for which HSIC is zero iff the variables are independent, for any probability distribution [19, Theorem 6]) which have no equivalence with the characteristic function-based criterion of [15, 24]: examples are given in [37, Section 3] and [27, Section 3].[1] Thus, the RKHS criterion is a more general dependence measure than the characteristic function criterion, which is in turn more general than the ℓ_2 distance between Parzen window density estimates. Since the HSIC-based algorithm performs as

[1] The RKHS approach also allows dependence testing on more general structures such as strings and graphs [20].

well as or better than the remaining kernel dependence criteria for large sample sizes [18] on the benchmark data of [5], we use it as the contrast function in our present algorithm.

While the above studies report excellent demixing accuracy, efficient optimisation of these dependence measures for ICA remains an ongoing problem,[2] and a barrier to using nonparametric methods when the number of sources, m, is large. The main focus of the present work is thus on more efficient optimisation of kernel dependence measures. ICA is generally decomposed into two subproblems [12, 11]: signal decorrelation or whitening, which is straightforward and is not discussed further, and optimisation over the set of orthogonal matrices (the orthogonal group, $O(m)$), which is a differentiable manifold, and for which the bulk of the computation is required. The approach of [5, 18, 19, 11] is to perform gradient descent on $O(m)$ in accordance with [13], choosing the step width by a Golden search. This is inefficient on two counts: gradient descent can require a very large number of steps for convergence even on relatively benign cost functions, and the Golden search requires many costly evaluations of the dependence measure. Although [23] propose a cheaper local quadratic approximation to choose the step size, this does not address the question of better search direction choice. An alternative solution is to use a Jacobi-type method [14, 26, 38], where the original optimisation problem on $O(m)$ is decomposed into a sequence of one-dimensional sub-problems over a set of pre-determined curves, parameterised by the Jacobi angles. While the theoretical convergence properties of a Jacobi approach as compared with direct optimisation on $O(m)$ are beyond the scope of this work, we perform an empirical evaluation against algorithms employing optimisation over Jacobi angles in our experiments.

In the present study, we develop an approximate Newton-like method for optimising the HSIC-based ICA contrast over $O(m)$, namely Fast HSIC-based Kernel ICA (FastKICA). A key feature of our approach is its computational efficiency, due to both the Newton-like optimisation and accurate low rank approximations of the independence measure and its derivatives. Importantly, these techniques do not require particular mathematical properties of the kernel (e.g. compact support, or that it be Laplace), but can be applied directly for any twice differentiable kernel function. The optimisation strategy follows recent studies on Newton-like methods for numerical optimisation on smooth manifolds in [21]. Approximate Newton-like algorithms have previously been developed in the case of classical ICA contrast functions [34, 32], where the authors use the diagonal structure of the Hessian at independence to greatly reduce complexity and computational cost. These earlier methods share the significant property of local quadratic convergence to a solution with correct source separation. We show the HSIC-based ICA contrast likewise has a diagonal Hessian at independence (this analysis originally appeared in [33]; note also that the diagonal property does not hold for the multivariate characteristic function-based counterpart [24] to the HSIC-based contrast), and that FastKICA is locally

[2] Most of the effort in increasing efficiency has gone into cheaply and accurately approximating the independence measures [5, 10, 11, 23].

quadratically convergent to a correct unmixing matrix. Moreover, our experiments suggest that in the absence of a good initialisation, FastKICA converges more often to a correct solution than gradient descent methods. Previous kernel algorithms require either a large number of restarts or a good initial guess [5, 11, 19]. The current work is built on an earlier presentation by the authors in [35]. Compared with [35], the present study contains proofs of the main theorems (which were omitted in [35] due to space constraints); a proof of local quadratic convergence in the neighbourhood of the global solution; additional experiments on ICA performance vs "smoothness" of the departure from independence; and experiments on outlier resistance, for which our method strongly outperforms the other tested approaches.

The paper is organised as follows. In Section 2, we briefly introduce the instantaneous noise-free ICA model, the HSIC-based ICA contrast, and a Newton-like method on $O(m)$. In Section 3, we analyse the critical point condition and the structure of the Hessian of this contrast. We describe our ICA method, FastKICA, in Section 4, and prove local quadratic convergence. We also present an efficient implementation of FastKICA, based on the incomplete Cholesky decomposition [16]. Finally, our experiments in Section 5 compare FastKICA with several competing nonparametric approaches: RADICAL [26], MILCA [38], mutual information-based ICA (MICA) [29], and KDICA [10]. Experiments address performance and runtimes on large-scale problems, performance for decreasing smoothness of the departure of the mixture from independence (which makes demixing more difficult for algorithms that assume smooth source densities), and outlier resistance. Matlab code for FastKICA may be downloaded at `www.kyb.mpg.de/bs/people/arthur/fastkica.htm`

2 Preliminaries: ICA, HSIC and Newton-like Method on $O(m)$

2.1 Linear Independent Component Analysis

The instantaneous noise-free ICA model takes the form

$$Z = AS, \tag{1}$$

where $S \in \mathbb{R}^{m\times n}$ is a matrix containing n observations of m sources, $A \in \mathbb{R}^{m\times m}$ is the mixing matrix (assumed here to have full rank),[3] and $Z \in \mathbb{R}^{m\times n}$ contains the observed mixtures. Denote as s and z single columns of the matrices S and Z, respectively, and let s_i be the i-th source in s. ICA is based on the assumption that the components s_i of s, for all $i = 1 \dots m$, are mutually statistically independent. This ICA model (1) is referred to as instantaneous as a way of describing the dual assumptions that the observation vector z depends only on

[3] In other words, we do not address the more difficult problems of undercomplete or overcomplete ICA (corresponding to more mixtures than sources, or fewer mixtures than sources, respectively).

the source vector s at that instant, and the source samples s are drawn independently and identically from $\Pr_s$. As a consequence of the first assumption, the mixture samples z are likewise drawn independently and identically from $\Pr_z$.

The task of ICA is to recover the independent sources via an estimate B of the inverse of the mixing matrix A, such that the recovered signals $Y = BAS$ have mutually independent components. It is well known that if at most one of the sources s is Gaussian, the mixing matrix A can be identified up to an ordering and scaling of the recovered sources [12]. This means the unmixing matrix B is the inverse of A up to an $m \times m$ permutation matrix P and an $m \times m$ diagonal (scaling) matrix D, i.e., $B = PDA^{-1}$. To reduce the computational complexity the mixtures Z are usually pre-whitened via principal component analysis (PCA) [12, 11]. Whitening corresponds to finding a matrix $V \in \mathbb{R}^{m \times m}$ such that $W = VZ = VAS \in \mathbb{R}^{m \times n}$ with $\mathbb{E}[ww^\top] = I$, where W are referred to as the whitened observations. While this pre-whitening step is less statistically efficient than solving directly for the unconstrained mixing matrix [9, Section VI.B], the optimisation problem in the pre-whitened case is easier. Assuming the sources s_i have zero mean and unit variance, we find $VA \in \mathbb{R}^{m \times m}$ to be orthogonal. Therefore, the whitened noise-free ICA unmixing model becomes

$$Y = X^\top W, \tag{2}$$

where $X \in \mathbb{R}^{m \times m}$ is an orthogonal unmixing matrix (i.e., $X^\top X = I$), and $Y \in \mathbb{R}^{m \times n}$ contains our estimates of the sources. Let $O(m)$ denote the orthogonal group:

$$O(m) := \{X \in \mathbb{R}^{m \times m} | X^\top X = I\}. \tag{3}$$

We focus in the remainder of this work on the problem of finding $X \in O(m)$ so as to recover the mutually statistically independent sources via the model (2). Thus, we next describe our measure of independence.

2.2 The Hilbert-Schmidt Independence Criterion

The Hilbert-Schmidt Independence Criterion (HSIC) is a bivariate independence measure obtained as the squared Hilbert-Schmidt (HS) norm of the covariance operator between mappings to RKHSs [18], and generalises the characteristic function-based criterion originally proposed by Feuerverger [15]. The Hilbert space $\mathcal{F}$ of functions from a compact subset $\mathcal{U} \subset \mathbb{R}$ to $\mathbb{R}$ is an RKHS if at each $u \in \mathcal{U}$, the point evaluation operator $\delta_u : \mathcal{F} \to \mathbb{R}$, which maps $f \in \mathcal{F}$ to $f(u) \in \mathbb{R}$, is a continuous linear functional. To each point $u \in \mathcal{U}$, there corresponds an element $\alpha_u \in \mathcal{F}$, also called the feature map, such that $\langle \alpha_u, \alpha_{u'} \rangle_{\mathcal{F}} = \psi(u, u')$, where $\psi : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a unique positive definite kernel. We also define a second RKHS $\mathcal{G}$ with respect to $\mathcal{U}$, with feature map $\beta_v \in \mathcal{G}$ and corresponding kernel $\langle \beta_v, \beta_{v'} \rangle_{\mathcal{G}} = \widehat{\psi}(v, v')$.

Let $\Pr_{u,v}$ be a joint measure on $(\mathcal{U} \times \mathcal{U}, \Gamma \times \Lambda)$ (here Γ and Λ are Borel σ-algebras on $\mathcal{U}$), with associated marginal measures $\Pr_u$ and $\Pr_v$. The covariance

operator $C_{uv} : \mathcal{G} \to \mathcal{F}$ is defined as

$$\langle f, C_{uv}(g) \rangle_{\mathcal{F}} = \mathbb{E}[f(u)g(v)] - \mathbb{E}[f(u)]\mathbb{E}[g(v)] \tag{4}$$

for all $f \in \mathcal{F}$ and $g \in \mathcal{G}$. The squared HS norm of the covariance operator C_{uv}, denoted as HSIC, is then

$$\|C_{uv}\|^2_{\mathrm{HS}} = \mathbb{E}_{u,u',v,v'}\left[\psi(u,u')\,\widehat{\psi}(v,v')\right] \tag{5a}$$

$$+ \mathbb{E}_{u,u'}\left[\psi(u,u')\right]\mathbb{E}_{v,v'}\left[\widehat{\psi}(v,v')\right] \tag{5b}$$

$$- 2\mathbb{E}_{u,v}\left[\mathbb{E}_{u'}\left[\psi(u,u')\right]\mathbb{E}_{v'}\left[\widehat{\psi}(v,v')\right]\right] \tag{5c}$$

(see [18] for details), where $(u, v) \sim \Pr_{u,v}$ and $(u', v') \sim \Pr_{u,v}$ are independent random variables drawn from the same distribution, and $\mathbb{E}[\cdot]$ denotes the expectation over the corresponding random variables. As long as the kernels $\psi(u, \cdot) \in \mathcal{F}$ and $\widehat{\psi}(u, \cdot) \in \mathcal{G}$ are universal in the sense of [37], e.g., the Gaussian and Laplace kernels, $\|C_{uv}\|^2_{\mathrm{HS}} = 0$ if and only if u and v are statistically independent [18, Theorem 4]. In this work, we confine ourselves to a Gaussian kernel, and use the same kernel for both $\mathcal{F}$ and $\mathcal{G}$,

$$\psi(a, b) = \widehat{\psi}(a, b) := \phi(a - b) = \exp\left(-\tfrac{(a-b)^2}{2\lambda^2}\right). \tag{6}$$

As discussed in the introduction, the empirical expression for HSIC in [18] is identical to Feuerverger's independence criterion [15] and Rosenblatt's ℓ^2 independence statistic [30] for a Gaussian kernel at a given sample size.

We now construct an HSIC-based ICA contrast for more than two random variables. In the ICA model (1), the components s_i of the sources s are mutually statistically independent if and only if their probability distribution factorises as $\Pr_s = \prod_{i=1}^m \Pr_{s_i}$. Although the random variables are pairwise independent if they are mutually independent, where pairwise independence is defined as $\Pr_{s_i} \Pr_{s_j} = \Pr_{s_i,s_j}$ for all $i \neq j$, the reverse does not generally hold: pairwise independence does not imply mutual independence. Nevertheless, Theorem 11 of [12] shows that in the ICA setting, unmixed components can be uniquely identified using only the pairwise independence between components of the recovered sources Y, since pairwise independence between components of Y in this case implies their mutual independence (and thus recovery of the sources S).[4] Hence, by summing all unique pairwise HSIC measures, an HSIC-based

[4]That said, in the finite sample setting, the statistical performance obtained by optimizing over a pairwise independence criterion might differ from that of a mutual independence criterion.

contrast function over the estimated signals $Y \in \mathbb{R}^{m\times n}$ is defined as

$$
\begin{aligned}
&H : O(m) \to \mathbb{R}, \\
&\quad H(X) := \sum_{1\le i<j\le m} \mathbb{E}_{k,l}\left[\phi\left(x_i^\top \overline{w}_{kl}\right)\phi\left(x_j^\top \overline{w}_{kl}\right)\right] && \text{(7a)}\\
&\qquad + \mathbb{E}_{k,l}\left[\phi\left(x_i^\top \overline{w}_{kl}\right)\right]\mathbb{E}_{k,l}\left[\phi\left(x_j^\top \overline{w}_{kl}\right)\right] && \text{(7b)}\\
&\qquad - 2\mathbb{E}_k\left[\mathbb{E}_l\left[\phi\left(x_i^\top \overline{w}_{kl}\right)\right]\mathbb{E}_l\left[\phi\left(x_j^\top \overline{w}_{kl}\right)\right]\right], && \text{(7c)}
\end{aligned}
$$

where $X := [x_1, \ldots, x_m] \in O(m)$, $\overline{w}_{kl} = w_k - w_l \in \mathbb{R}^m$ denotes the difference between k-th and l-th samples of the whitened observations, and $\mathbb{E}_{k,l}[\cdot]$ represents the empirical expectation over all k and l.

2.3 Newton-like Methods on $O(m)$

In this section, we briefly review some basic concepts regarding Newton-like methods on the orthogonal group $O(m)$. We refer to [6, 36] for an excellent introduction to differential geometry, and to [1] for an introduction to optimisation algorithms on differentiable manifolds. We will review both the classical approach to Newton-type optimization on smooth manifolds [13], and then describe a more recently developed Newton-like method [21], which we apply on $O(m)$.

We consider the orthogonal group $O(m)$ as an $m(m-1)/2$ dimensional embedded submanifold of $\mathbb{R}^{m\times m}$, and denote the set of all $m \times m$ skew-symmetric matrices by $\mathfrak{so}(m) := \{\Omega \in \mathbb{R}^{m\times m} | \Omega = -\Omega^\top\}$. Note that $\mathfrak{so}(m)$ is isomorphic to $\mathbb{R}^{m(m-1)/2}$, written $\mathfrak{so}(m) \cong \mathbb{R}^{m(m-1)/2}$. The tangent space $T_X O(m)$ of $O(m)$ at point $X \in O(m)$ is given by

$$
T_X O(m) := \left\{ \Xi \in \mathbb{R}^{m\times m} \,\middle|\, \Xi = X\Omega,\ \Omega \in \mathfrak{so}(m) \right\}. \tag{8}
$$

A typical approach in developing a Newton-type method for optimising a smooth function $H\colon O(m) \to \mathbb{R}$ is to endow the manifold $O(m)$ with a Riemannian structure: see [13]. Rather than moving along a straight line as in the Euclidean case, a Riemannian Newton iteration moves along a geodesic[5] in $O(m)$. For a given tangent space direction $\Xi = X\Omega \in T_X O(m)$, the geodesic γ_X through $X \in O(m)$ with respect to the Riemannian metric $\langle X\Omega_1, X\Omega_2\rangle := -\operatorname{tr}\Omega_1\Omega_2$, for $X\Omega_1, X\Omega_2 \in T_X O(m)$, is

$$
\gamma_X : \mathbb{R} \to O(m), \qquad \varepsilon \mapsto X \exp\left(\varepsilon X^\top \Xi\right), \tag{9}
$$

with $\gamma_X(0) = X$ and $\dot{\gamma}_X(0) = \Xi$. Here, $\exp(\cdot)$ denotes matrix exponentiation. It is well known that this method enjoys the significant property of local quadratic convergence.

More recently, a novel Newton-like method on smooth manifolds was proposed [21]. This method has lower complexity than the classical approach, but

[5] The geodesic is a concept on manifolds analogous to the straight line in a Euclidean space . It allows parallel transportation of tangent vectors on manifolds.

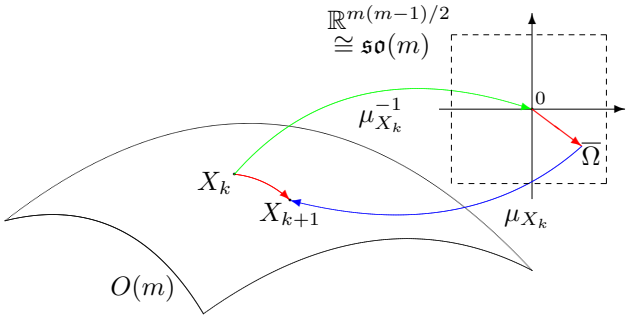


Figure 1: Illustration of a Newton-like method on $O(m)$.

retains the property of local quadratic convergence. We adapt the general formulation from [21] to the present setting, the orthogonal group $O(m)$.

For every point $X \in O(m)$, there exists a smooth map

$$\mu_X \colon \mathbb{R}^{m(m-1)/2} \to O(m), \quad \mu_X(0) = X, \tag{10}$$

which is a local diffeomorphism around $0 \in \mathbb{R}^{m(m-1)/2}$, i.e., both the map μ_X and its inverse are locally smooth around 0. Let $X^* \in O(m)$ be a nondegenerate critical point of a smooth contrast function $H \colon O(m) \to \mathbb{R}$. If there exists an open neighborhood $\mathcal{U}(X^*) \subset O(m)$ of $X^* \in O(m)$ and a smooth map

$$\widetilde{\mu} : \mathcal{U}(X^*) \times \mathbb{R}^{m(m-1)/2} \to O(m), \tag{11}$$

such that $\widetilde{\mu}(X, \overline{\Omega}) = \mu_X(\overline{\Omega})$ for all $X \in \mathcal{U}(X^*)$ and $\overline{\Omega} \in \mathbb{R}^{m(m-1)/2}$, we call $\{\mu_X\}_{X \in \mathcal{U}(X^*)}$ a locally smooth family of parametrisations around X^*.

Let $X_k \in O(m)$ be the k-th iteration point of a Newton-like method for minimising the contrast function H. A local cost function can then be constructed by composing the original function H with the local parametrisation μ_{X_k} around X_k, i.e., $H \circ \mu_{X_k} \colon \mathbb{R}^{m(m-1)/2} \to \mathbb{R}$, which is a smooth function locally defined on the Euclidean space $\mathbb{R}^{m(m-1)/2}$. Thus, one Euclidean Newton step $\overline{\Omega} \in \mathbb{R}^{m(m-1)/2}$ for H, expressed in local coordinates, is the solution of the linear equation

$$\mathcal{H}(H \circ \mu_{X_k})(0)\overline{\Omega} = -\nabla(H \circ \mu_{X_k})(0), \tag{12}$$

where $\nabla(H \circ \mu_{X_k})(0)$ and $\mathcal{H}(H \circ \mu_{X_k})(0)$ are respectively the gradient and Hessian of $H \circ \mu_{X_k}$ at $0 \in \mathbb{R}^{m(m-1)/2}$ with respect to the standard Euclidean inner product on the parameter space $\mathbb{R}^{m(m-1)/2}$. Finally, projecting the Newton step $\overline{\Omega}$ in the local coordinates back to $O(m)$ using the local parametrisation[6] μ_{X_k} completes a basic iteration of a Newton-like method on $O(m)$ (see Fig. 1 for an illustration of the method).

[6] In the general setting, the second parametrisation can be from a different family to the first: see [21] for details. This is not the case for our algorithm, however.

We now describe our choice of local parametrisation μ_X of $O(m)$, which follows directly from the geodesic expression (9). We define $\Omega = (\omega_{ij})_{i,j=1}^m \in \mathfrak{so}(m)$ as before and let $\overline{\Omega} = (\omega_{ij})_{1\leq i<j\leq m} \in \mathbb{R}^{m(m-1)/2}$ in a lexicographical order. A local parametrisation of $O(m)$ around a point $X \in O(m)$ is given by

$$\mu_X \colon \mathbb{R}^{m(m-1)/2} \cong \mathfrak{so}(m) \to O(m), \quad \overline{\Omega} \mapsto X \exp(\Omega), \tag{13}$$

which is a local diffeomorphism around $0 \in \mathbb{R}^{m(m-1)/2}$, i.e. $\mu_X(0) = X$.

To summarise, a Newton-like method for minimising the contrast function $H \colon O(m) \to \mathbb{R}$ can be stated as follows:

Newton-like method on $O(m)$
Step 1: Given an initial guess $X_0 \in O(m)$ and set $k = 0$.
Step 2: Calculate $H \circ \mu_{X_k} \colon \mathbb{R}^{m(m-1)/2} \cong \mathfrak{so}(m) \to \mathbb{R}$.
Step 3: Compute the Euclidean Newton step, i.e., solve the linear system for $\overline{\Omega} \in \mathbb{R}^{m(m-1)/2}$, $\mathcal{H}(H \circ \mu_{X_k})(0)\overline{\Omega} = -\nabla(H \circ \mu_{X_k})(0)$.
Step 4: Set $X_{k+1} = \nu_{X_k}(\overline{\Omega})$.
Step 5: If $\|X_{k+1} - X_k\|_{\mathrm{F}}$ is small enough, stop. Otherwise, set $k = k + 1$ and go to Step 2.

Here, $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm of matrices. According to Theorem 1 in [21], this Newton-like method is locally quadratically convergent to $X^* \in O(m)$. For an approximate Newton-like method on $O(m)$, we replace the true Hessian $\mathcal{H}(H \circ \mu_X)(0)$ by an approximate Hessian $\widetilde{\mathcal{H}}(H \circ \mu_X)(0)$ which is more efficient to compute. We will prove that local quadratic convergence is still obtained with this approximation.

3 The HSIC-Based ICA Contrast at Independence

In this section we examine the critical points of the HSIC-based ICA contrast at independence. It turns out that any correct unmixing matrix which is a global minimum of the HSIC-based contrast is a nondegenerate critical point.

Let $X = [x_1, \ldots, x_m] \in O(m)$. By the chain rule, the first derivative of H in the direction $\Xi = [\xi_1, \ldots, \xi_m] \in T_X O(m)$ is

$$\begin{aligned}
&\mathrm{D}\, H(X)\Xi = \tfrac{\mathrm{d}}{\mathrm{d}\,\varepsilon}(H \circ \gamma_X)(\varepsilon)\big|_{\varepsilon=0} \\
&= \sum_{i,j=1;i\neq j}^{m} \mathbb{E}_{k,l}\left[\phi'\left(x_i^\top \overline{w}_{kl}\right)\xi_i^\top \overline{w}_{kl}\phi\left(x_j^\top \overline{w}_{kl}\right)\right] && \text{(14a)} \\
&\quad + \mathbb{E}_{k,l}\left[\phi'\left(x_i^\top \overline{w}_{kl}\right)\xi_i^\top \overline{w}_{kl}\right]\mathbb{E}_{k,l}\left[\phi\left(x_j^\top \overline{w}_{kl}\right)\right] && \text{(14b)} \\
&\quad - 2\mathbb{E}_k\left[\mathbb{E}_l\left[\phi'\left(x_i^\top \overline{w}_{kl}\right)\xi_i^\top \overline{w}_{kl}\right]\mathbb{E}_l\left[\phi\left(x_j^\top \overline{w}_{kl}\right)\right]\right]. && \text{(14c)}
\end{aligned}$$

Setting the above derivative to zero, we can characterise the critical points of the HSIC-based ICA contrast function defined in (7). Obviously, the critical point condition depends not only on the statistical characteristics of the sources, but also on the properties of the kernel function. It is hard to characterise all critical points of HSIC in full generality: thus, we deal only with those critical points that occur at the ICA solution.

Lemma 1. *Let $X^* \in O(m)$ be a correct unmixing matrix of the model* (2)*. Then X^* is a nondegenerate critical point of the HSIC-based ICA contrast* (7)*, i.e., $\mathrm{D}\, H(X^*)\Xi = 0$ for arbitrary $\Xi \in T_{X^*}O(m)$.*

Proof. We recall that $\|C_{uv}\|^2_{\mathrm{HS}} \geq 0$ and $\|C_{uv}\|^2_{\mathrm{HS}} = 0$ if and only if $\Pr_{u,v} = \Pr_u \Pr_v$. In other words, $X^* \in O(m)$ is a global minimum of the HSIC-based contrast function. Theorem 4 in [18] shows that any small displacement of $X^* \in O(m)$ along the geodesics will result in an increase in the score of the contrast function H between the recovered sources (only larger perturbations, e.g. a swap of the source orderings, would yield independent sources). According to Theorem 4.2 in [17], the Hessian of H at X^* is positive definite. Hence, any correct separation point $X^* \in O(m)$ is a nondegenerate critical point of the HSIC-based contrast function H. □

In what follows, we investigate the structure of the Hessian of the HSIC-based contrast H at independence, i.e. at a correct unmixing matrix $X^* \in O(m)$. We first compute the second derivative of H at $X \in O(m)$ in direction $\Xi = [\xi_1, \ldots, \xi_m] \in T_X O(m)$,

$$
\begin{aligned}
&\mathrm{D}^2 H(X)(\Xi,\Xi) = \tfrac{\mathrm{d}^2}{\mathrm{d}\,\varepsilon^2}(H \circ \gamma_X)(\varepsilon)\Big|_{\varepsilon=0} \\
&= \sum_{i,j=1;i\neq j}^{m} \mathbb{E}_{k,l}\big[\phi''\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \overline{w}_{kl}\overline{w}_{kl}^\top \xi_i \phi\big(x_j^\top \overline{w}_{kl}\big)\big] && \text{(15a)}\\
&\quad - \mathbb{E}_{k,l}\big[\phi'\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \Xi X^\top \overline{w}_{kl} \phi\big(x_j^\top \overline{w}_{kl}\big)\big] && \text{(15b)}\\
&\quad + \mathbb{E}_{k,l}\big[\phi'\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \overline{w}_{kl}\overline{w}_{kl}^\top \xi_j \phi'\big(x_j^\top \overline{w}_{kl}\big)\big] && \text{(15c)}\\
&\quad + \mathbb{E}_{k,l}\big[\phi''\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \overline{w}_{kl}\overline{w}_{kl}^\top \xi_i\big]\, \mathbb{E}_{k,l}\big[\phi\big(x_j^\top \overline{w}_{kl}\big)\big] && \text{(15d)}\\
&\quad - \mathbb{E}_{k,l}\big[\phi'\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \Xi X^\top \overline{w}_{kl}\big]\, \mathbb{E}_{k,l}\big[\phi\,\big(x_j^\top \overline{w}_{kl}\big)\big] && \text{(15e)}\\
&\quad + \mathbb{E}_{k,l}\big[\phi'\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \overline{w}_{kl}\big]\, \mathbb{E}_{k,l}\big[\phi'\big(x_j^\top \overline{w}_{kl}\big)\, \xi_j^\top \overline{w}_{kl}\big] && \text{(15f)}\\
&\quad - 2\mathbb{E}_k\big[\mathbb{E}_l\big[\phi''\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \overline{w}_{kl}\overline{w}_{kl}^\top \xi_i\big]\mathbb{E}_l\big[\phi\big(x_j^\top \overline{w}_{kl}\big)\big]\big] && \text{(15g)}\\
&\quad + 2\mathbb{E}_k\big[\mathbb{E}_l\big[\phi'\big(x_i^\top \overline{w}_{kl}\big)\, \xi_i^\top \Xi X^\top \overline{w}_{kl}\big]\mathbb{E}_l\big[\phi\big(x_j^\top \overline{w}_{kl}\big)\big]\big] && \text{(15h)}\\
&\quad - 2\mathbb{E}_k\big[\mathbb{E}_l\big[\phi'\big(x_i^\top \overline{w}_{kl}\big)\xi_i^\top \overline{w}_{kl}\big]\mathbb{E}_l\big[\phi'\big(x_j^\top \overline{w}_{kl}\big)\xi_j^\top \overline{w}_{kl}\big]\big]. && \text{(15i)}
\end{aligned}
$$

Let $\Omega = [\omega_1, \ldots, \omega_m] = (\omega_{ij})_{i,j=1}^m \in \mathfrak{so}(m)$. For $X = X^*$, a tedious but straightforward computation (see Appendix A) gives

$$
\mathrm{D}^2 H(X)(X\Omega, X\Omega)\big|_{X=X^*} = \sum_{1\leq i<j\leq m}^{m} \omega_{ij}^2\,(\kappa_{ij} + \kappa_{ji})\,, \tag{16}
$$

where

$$\kappa_{ij} = 2\mathbb{E}_k[\mathbb{E}_l[\overline{s}_{kli}]\mathbb{E}_l[\phi'(\overline{s}_{kli})]]\mathbb{E}_k[\mathbb{E}_l[\overline{s}_{klj}]\mathbb{E}_l[\phi'(\overline{s}_{klj})]] \tag{17a}$$
$$+ \mathbb{E}_{k,l}[\phi''(\overline{s}_{kli})]\mathbb{E}_{k,l}[(\overline{s}_{klj})^2\phi(\overline{s}_{klj})] \tag{17b}$$
$$+ 2\mathbb{E}_{k,l}[\phi''(\overline{s}_{kli})]\mathbb{E}_{k,l}[\phi(\overline{s}_{klj})] \tag{17c}$$
$$- 2\mathbb{E}_{k,l}[\phi''(\overline{s}_{kli})]\mathbb{E}_k\left[\mathbb{E}_l[(\overline{s}_{klj})^2]\mathbb{E}_l[\phi(\overline{s}_{klj})]\right] \tag{17d}$$
$$- \mathbb{E}_{k,l}[\phi'(\overline{s}_{kli})\overline{s}_{kli}]\mathbb{E}_{k,l}[\phi'(\overline{s}_{klj})\overline{s}_{klj}]. \tag{17e}$$

Remark 1. *Without loss of generality, let* $\overline{\Omega} = (\omega_{ij})_{1\leq i<j\leq m} \in \mathbb{R}^{m(m-1)/2}$ *in a lexicographical order. The quadratic form* (16) *is a sum of pure squares, which indicates that the Hessian of the contrast function* H *in* (7) *at the desired critical point* $X^* \in O(m)$*, i.e., the symmetric bilinear form* $\mathcal{H}H(X^*)\colon T_{X^*}O(m) \times T_{X^*}O(m) \to \mathbb{R}$*, is diagonal with respect to the standard basis of* $\mathbb{R}^{m(m-1)/2}$*. Furthermore, following the arguments in Lemma 1, the expressions* $\kappa_{ij} + \kappa_{ji}$ *are positive.* □

4 Fast HSIC Based ICA and its Implementation

Having defined our independence criterion and its behaviour at independence, we now describe an efficient Newton-like method for minimising $H(X)$. We begin in Section 4.1 with an overview of the method, including the approximate Hessian used to speed up computation. The subsequent two sections describe how the Hessian (Section 4.2), as well as $H(X)$ (Section 4.3) and its gradient (Section 4.4), can be computed much faster using the incomplete Cholesky decomposition.

4.1 Fast HSIC Based ICA Method

We compute the gradient and Hessian of $H \circ \mu_X$ in the parameter space $\mathbb{R}^{m(m-1)/2}$. By analogy with equation (14), the first derivative of $H \circ \mu_X$ at $0 \in \mathbb{R}^{m(m-1)/2}$ is

$$\begin{aligned} \mathrm{D}(H \circ \mu_X)(0)\overline{\Omega} &= \tfrac{\mathrm{d}}{\mathrm{d}\,\varepsilon}(H \circ \mu_X)(\varepsilon\overline{\Omega})\big|_{\varepsilon=0} \\ &= \sum_{1\leq i<j\leq m} \omega_{ij}(\tau_{ij} - \tau_{ji}), \end{aligned} \tag{18}$$

where

$$\begin{aligned} \tau_{ij} = &\sum_{r=1;r\neq i}^{m} \mathbb{E}_{k,l}\left[\phi'\left(x_i^\top \overline{w}_{kl}\right) x_j^\top \overline{w}_{kl}\phi\left(x_r^\top \overline{w}_{kl}\right)\right] \\ &+ \mathbb{E}_{k,l}\left[\phi'\left(x_i^\top \overline{w}_{kl}\right) x_j^\top \overline{w}_{kl}\right] \mathbb{E}_{k,l}\left[\phi\left(x_r^\top \overline{w}_{kl}\right)\right] \\ &- 2\mathbb{E}_k\left[\mathbb{E}_l\left[\phi'\left(x_i^\top \overline{w}_{kl}\right) x_j^\top \overline{w}_{kl}\right] \mathbb{E}_l\left[\phi\left(x_r^\top \overline{w}_{kl}\right)\right]\right]. \end{aligned} \tag{19}$$

Let $Q := (q_{ij})_{i,j=1}^m \in \mathfrak{so}(m)$ and $\nabla(H \circ \mu_X)(0) = (q_{ij})_{1 \leq i < j \leq m} \in \mathbb{R}^{m(m-1)/2}$. Then the Euclidean gradient $\nabla(H \circ \mu_X)(0)$ is given entry-wise as $q_{ij} = \tau_{ij} - \tau_{ji}$ for all $1 \leq i < j \leq m$.

Similarly, the Hessian of the contrast function H can be computed directly from (15). The diagonal property of the Hessian does not hold true for arbitrary $X \in O(m)$, however, and it is clearly too expensive to compute the true Hessian at each step of our optimization. Nevertheless, since the Hessian is diagonal at X^* (Eq. (16)), a diagonal approximation of the Hessian makes sense in a neighborhood of X^*. Thus, we propose a diagonal matrix for the Hessian of $H \circ \mu_X$ for an arbitrary $X \in O(m)$ at $0 \in \mathbb{R}^{m(m-1)/2}$; the result is a symmetric bilinear form $\mathcal{H}(H \circ \mu_X)(0) \colon \mathbb{R}^{m(m-1)/2} \times \mathbb{R}^{m(m-1)/2} \to \mathbb{R}$. Due to the smoothness of both the contrast function H in (7) and the map $\widetilde{\mu}$ on $O(m)$ in (11), the Hessian $\mathcal{H}(H \circ \mu_X)(0)$ is smooth in an open neighborhood $\mathcal{U}(X^*) \subset O(m)$ of $X^* \in O(m)$. Replacing the correct unmixing components s in (17) by the current estimates $y = X^\top w$, i.e. $\overline{y}_{kli} = x_i^\top \overline{w}_{kl}$, gives

$$\mathcal{H}(H \circ \mu_X)(0)(\overline{\Omega}, \overline{\Omega}) \approx \sum_{1 \leq i < j \leq m}^{m} \omega_{ij}^2 \left(\widetilde{\kappa}_{ij} + \widetilde{\kappa}_{ji}\right), \tag{20}$$

where

$$\begin{aligned}
\widetilde{\kappa}_{ij} = {} & 2\mathbb{E}_k[\mathbb{E}_l[\overline{y}_{kli}]\mathbb{E}_l[\phi'(\overline{y}_{kli})]]\mathbb{E}_k\big[\mathbb{E}_l[\overline{y}_{klj}]\mathbb{E}_l[\phi'(\overline{y}_{klj})]\big] && \text{(21a)}\\
& + \mathbb{E}_{k,l}[\phi''(\overline{y}_{kli})]\mathbb{E}_{k,l}[(\overline{y}_{klj})^2\phi(\overline{y}_{klj})] && \text{(21b)}\\
& + 2\mathbb{E}_{k,l}[\phi''(\overline{y}_{kli})]\mathbb{E}_{k,l}[\phi(\overline{y}_{klj})] && \text{(21c)}\\
& - 2\mathbb{E}_{k,l}[\phi''(\overline{y}_{kli})]\mathbb{E}_k\left[\mathbb{E}_l[(\overline{y}_{klj})^2]\mathbb{E}_l[\phi(\overline{y}_{klj})]\right] && \text{(21d)}\\
& - \mathbb{E}_{k,l}[\phi'(\overline{y}_{kli})\overline{y}_{kli}]\mathbb{E}_{k,l}[\phi'(\overline{y}_{klj})\overline{y}_{klj}]. && \text{(21e)}
\end{aligned}$$

We emphasise that (by Remark 1) the Hessian at a correct unmixing matrix X^* is positive definite (i.e. the terms $\widetilde{\kappa}_{ij} + \widetilde{\kappa}_{ji}$ are positive). Thus, the approximate Newton-like direction $\widetilde{\Omega} \in \mathbb{R}^{m(m-1)/2}$ with

$$\widetilde{\omega}_{ij} = \tfrac{\tau_{ij} - \tau_{ji}}{\widetilde{\kappa}_{ij} + \widetilde{\kappa}_{ji}} \tag{22}$$

for $1 \leq i < j \leq m$ is smooth, and is well defined within $\mathcal{U}(X^*)$. We call the Newton-like method arising from this approximation Fast HSIC-based Kernel ICA (FastKICA).

Although the approximation (20) can differ substantially from the true Hessian at an arbitrary $X \in O(m)$, they coincide at X^*. Thus, while the diagonal approximation is exact at the correct solution, it becomes less accurate as we move away from this solution. To ensure FastKICA is nonetheless well behaved as the global solution is approached, we provide the following local convergence result (we investigate the performance of our algorithm given arbitrary initialisation in our numerical experiments: see Section 5).

Corollary 1. *Let $X^* \in O(m)$ be a correct unmixing matrix. Then FastKICA is locally quadratically convergent to X^*.*

Proof. By considering the approximate Newton-like direction $\widetilde{\Omega}$ in (22) as $\widetilde{\Omega}\colon \mathcal{U}(X^*) \to \mathfrak{so}(m)$, each iteration of FastKICA can be written as the map

$$\mathcal{A}\colon \mathcal{U}(X^*) \subset O(m) \to O(m), \quad X \mapsto X \exp(\widetilde{\Omega}(X)). \tag{23}$$

A tedious but direct computation shows that $\widetilde{\Omega}(X^*) = 0$, and X^* is a fixed point of $\mathcal{A}$. To prove the local quadratic convergence of FastKICA, we employ Lemma 2.9 in [25], bearing in mind that the required smoothness conditions (that the function be at least twice differentiable) are fulfilled by $H(X)$. According to this lemma, we only need to show that the first derivative of $\mathcal{A}$,

$$\mathrm{D}\,\mathcal{A}\colon T_X O(m) \to T_{\mathcal{A}(X)} O(m), \tag{24}$$

vanishes at a fixed point X^*. Thus we compute directly

$$\mathrm{D}\,\mathcal{A}(X^*)(X^*\Omega) = X^*\Omega + X^* \,\mathrm{D}\,\widetilde{\Omega}(X^*)(X^*\Omega). \tag{25}$$

For the expression in (25) to vanish is equivalent to

$$\Omega = -\,\mathrm{D}\,\widetilde{\Omega}(X^*)(X^*\Omega), \tag{26}$$

i.e., for all $1 \leq i < j \leq m$,

$$\mathrm{D}\,\widetilde{\omega}_{ij}(X^*)(X^*\Omega) = -\omega_{ij}. \tag{27}$$

Denoting the numerator $\tau_{ij} - \tau_{ji}$ in (22) by $\widetilde{\omega}_{ij}^{(n)}(X)$ and the denominator $\widetilde{\kappa}_{ij} + \widetilde{\kappa}_{ji}$ by $\widetilde{\omega}_{ij}^{(d)}(X)$, we compute

$$\begin{aligned} \mathrm{D}\,\widetilde{\omega}_{ij}(X^*)(X^*\Omega) = &- \frac{\widetilde{\omega}_{ij}^{(n)}(X^*)}{(\widetilde{\omega}_{ij}^{(d)}(X^*))^2} \,\mathrm{D}\,\widetilde{\omega}_{ij}^{(d)}(X^*)(X^*\Omega) \\ &+ \frac{\mathrm{D}\,\widetilde{\omega}_{ij}^{(n)}(X^*)(X^*\Omega)}{\widetilde{\omega}_{ij}^{(d)}(X^*)}. \end{aligned} \tag{28}$$

It can be shown that the first summand in (28) is equal to zero. Finally, following an argument almost identical to that in Appendix A,

$$\mathrm{D}\,\widetilde{\omega}_{ij}^{(n)}(X^*)(X^*\Omega) = -\widetilde{\omega}_{ij}^{(d)}(X^*)\omega_{ij}. \tag{29}$$

Thus, we conclude condition (26) holds true at X^*, i.e., $\mathrm{D}\,\mathcal{A}(X^*)$ vanishes as required. The result follows. □

We emphasise that the local convergence result is proved in the population setting. In the finite sample case, this theoretical convergence rate might not be achievable [28]. For this reason, we experimentally compare the convergence behavior of FastKICA with a gradient-based algorithm (see Section 5.1). We will see that FastKICA converges significantly faster than gradient descent in practice.

As an additional caveat to Corollary 1, there is in general no practical strategy to guarantee that our ICA algorithm will be initialised within the neighbourhood $\mathcal{U}(X^*)$. Nevertheless, our experiments in Section 5 demonstrate that FastKICA often converges to the global solution even from arbitrary initialisation points, and is much more reliable in this respect than simple gradient descent.

4.2 Incomplete Cholesky Estimate of the Hessian

In the following, we present an explicit formulation of the Hessian, and derive its approximation using an incomplete Cholesky decomposition of the kernel matrices. Sections 4.3 and 4.4 provide implementations of the contrast and gradient, respectively, that re-use the factors of the Cholesky decomposition.

First, we rewrite the pairwise HSIC terms (7) in a more convenient matrix form. Let K_i denote the kernel matrix for the i-th estimated source, i.e., its (k,l)-th entry is $\phi(\overline{y}_{kli}) = \phi(x_i^\top \overline{w}_{kl})$. We further denote by $M \in \mathbb{R}^{n\times n}$ the centring operation, $M = I - \frac{1}{n}1_n 1_n^\top$, with 1_n being an $n \times 1$ vector of ones.

Lemma 2 (HSIC in terms of kernels [18])**.** *An empirical estimate of HSIC for two estimated sources y_i, y_j is*

$$H_{uv}(X) := \frac{1}{(n-1)^2}\mathrm{tr}(MK_iMK_j). \tag{30}$$

This empirical estimate is biased, however the bias in this estimate decreases as $1/n$, and thus drops faster than the variance (which decreases as $1/\sqrt{n}$: see [18]).

The first and second derivatives of the Gaussian kernel function ϕ in (6) are

$$\begin{cases} \phi'(a,b) &= -\frac{a-b}{\lambda^2}\phi(a,b), \\ \phi''(a,b) &= \frac{(a-b)^2}{\lambda^4}\phi(a,b) - \frac{1}{\lambda^2}\phi(a,b). \end{cases} \tag{31}$$

Substituting the above terms into the approximate Hessian of $H \circ \mu_X$ at $0 \in \mathfrak{so}(m)$, as computed in (20), yields

$$\mathcal{H}(H\circ\mu_X)(0)(\Omega,\Omega) \approx \sum_{1\le i<j\le m} \omega_{ij}^2 \Delta_{ij}, \tag{32}$$

where

$$\Delta_{ij} = \tfrac{2}{\lambda^2}\left(\beta_i\zeta_j + \zeta_i\beta_j\right) + \tfrac{4}{\lambda^4}\left(\zeta_i\zeta_j - \eta_i\eta_j\right), \tag{33}$$

and

$$\begin{cases} \beta_i &= \mathbb{E}_{k,l}[\phi(\overline{y}_{kli})], \\ \zeta_i &= \mathbb{E}_{k,l}[\phi(\overline{y}_{kli})y_{ki}y_{li}], \\ \eta_i &= \mathbb{E}_{k,l}[\phi(\overline{y}_{kli})y_{ki}^2]. \end{cases} \tag{34}$$

Here, $y_{ki} = e_i^\top y_k$ is the i-th entry of the k-th sample of the estimates Y, selected by the i-th standard basis vector e_i of $\mathbb{R}^m$.

We now outline how the incomplete Cholesky decomposition [16] helps to estimate the approximate Hessian efficiently. An incomplete Cholesky decomposition of the Gram matrix K_i yields a low-rank approximation $K_i \approx G_iG_i^\top$ that greedily minimises $\mathrm{tr}(K_i - G_iG_i^\top)$. The cost of computing the $n \times d$ matrix G_i is $O(nd^2)$, with $d \ll n$. Greater values of d result in a more accurate reconstruction of K_i. As pointed out in [5, Appendix C], however, the spectrum of a Gram matrix based on the Gaussian kernel generally decays rapidly, and

a small d yields a very good approximation (the experiments in [23] provide further empirical evidence). With approximate Gram matrices, the empirical estimates of the three terms in Equation (34) become

$$\begin{aligned}\hat{\beta}_i &= \frac{1}{n^2}(1_n^\top G_i)(1_n^\top G_i)^\top,\\ \hat{\zeta}_i &= \frac{1}{n^2}(y_i^\top G_i)(y_i^\top G_i)^\top,\\ \hat{\eta}_i &= \frac{1}{n^2}((y_i \odot y_i)^\top G_i)(G_i^\top 1_n),\end{aligned}$$

where y_i is the sample vector for the ith estimated source, and $\odot$ the entry-wise product of vectors.

4.3 Incomplete Cholesky Estimate of HSIC

Lemma 2 states an estimate of the pairwise HSIC as the trace of a product of centred kernel matrices MK_iM, MK_jM. Reusing the Cholesky decomposition from above, i.e. $MK_iM = MG_iG_i^\top M =: \widetilde{G}_i\widetilde{G}_i^\top$, where $\widetilde{G}_i$ is $n \times d_i$, we arrive at an equivalent trace of a much smaller $d_j \times d_j$ matrix for each pair of estimated sources (y_i, y_j), and avoid any product of $n \times n$ matrices:

$$\begin{aligned}\widehat{H}(X) &= \frac{1}{(n-1)^2}\sum_{1\le i<j\le m}\operatorname{tr}\left(\widetilde{G}_i\widetilde{G}_i^\top\widetilde{G}_j\widetilde{G}_j^\top\right)\\ &= \frac{1}{(n-1)^2}\sum_{1\le i<j\le m}\operatorname{tr}\left(\left(\widetilde{G}_j^\top\widetilde{G}_i\right)\left(\widetilde{G}_i^\top\widetilde{G}_j\right)\right).\end{aligned}$$

4.4 Incomplete Cholesky Estimate of the Gradient

We next provide an approximation of the gradient in terms of the same Cholesky approximation $K = GG^\top$. Williams and Seeger [39] describe an approximation of K based on an index set I of length d with unique entries from $\{1, \ldots, n\}$:

$$K \approx K' := K_{:,I}K_{I,I}^{-1}K_{I,:}, \tag{35}$$

where $K_{:,I}$ is the Gram matrix with the rows unchanged, and the columns chosen from the set I; and $K_{I,I}$ is the $d \times d$ submatrix with both rows and columns restricted to I. We adapt the approach of Fine and Scheinberg [16] and choose the indices I in accordance with an incomplete Cholesky decomposition to minimise $\operatorname{tr}(K - K')$.

For simplicity of notation, we will restrict ourselves to the gradient of HSIC for a pair (y_i, y_j) and ignore the normalisation by $(n-1)^{-2}$. The gradient of $\widehat{H}(X)$ is merely the sum of all pairwise gradients. Let K be the Gram matrix for y_i and L for y_j. With centring matrices $\widetilde{K} = MKM$ and $\widetilde{L} = MLM$ the

(unnormalised) differential of the pairwise HSIC is [23]

$$\begin{aligned} d\mathrm{tr}\left(\widetilde{K}'\widetilde{L}'\right) &= \mathrm{tr}\left(\widetilde{K}'d(L')\right) + \mathrm{tr}\left(\widetilde{L}'d(K')\right) \\ &= \mathrm{tr}\left(\widetilde{K}'d(L')\right) + \mathrm{tr}\left(\widetilde{L}'d(K_{:,I}K_{I,I}^{-1}K_{I,:})\right) \\ &= \mathrm{tr}\left(\widetilde{K}'d(L')\right) \\ &\quad + 2\mathrm{vec}(\widetilde{L}'K_{:,I}K_{I,I}^{-1})^\top \mathrm{vec}(dK_{:,I}) \\ &\quad - \mathrm{vec}(K_{I,I}^{-1}K_{I,:}\widetilde{L}'K_{:,I}K_{I,I}^{-1})^\top \mathrm{vec}(dK_{I,I}). \end{aligned} \tag{36}$$

Our expression for $d(K')$ is derived in Appendix B.1. The expansion of dL' is analogous. The matrix decompositions shrink the size of the factors in (36). An appropriate ordering of the matrix products allows us to avoid ever generating or multiplying an $n \times n$ matrix. In addition, note that for a column x of X,

$$\mathrm{vec}(dK_{:,I}) = d\mathrm{vec}(K_{:,I}) = \left(\partial \mathrm{vec}(K_{:,I})/\partial x^\top\right)\ d\mathrm{vec}(x).$$

The partial matrix derivative $\partial \mathrm{vec}(K_{:,I})/\partial x^\top$ is defined in Appendix B.2 and has size $nd \times m$, whereas the derivative of the full K has $n^2 \times m$ entries. Likewise, $\partial \mathrm{vec}(K_{I,I})/\partial x^\top$ is only $d^2 \times m$. We must also account for the rapid decay of the spectrum of Gram matrices with Gaussian kernels [5, discussion in Appendix C], since this can cause the inverse of $K_{I,I}$ to be ill-conditioned. We therefore add a small ridge of 10^{-6} to $K_{I,I}$, although we emphasise that our algorithm is insensitive to this value.

We end this section with a brief note on the overall computational cost of FastKICA. As discussed in [23, Section 1.5], the gradient and Hessian are computable in $\mathcal{O}(nm^3d^2)$ operations. A more detailed breakdown of how we arrive at this cost may be found in [23], bearing in mind that the Hessian has the same cost as the gradient thanks to our diagonal approximation.

5 Numerical Experiments

In our experiments, we demonstrate four main points: First, if no alternative algorithm is used to provide an initial estimate of X, FastKICA is resistant to local minima, and often converges to the correct solution. This is by contrast with gradient descent, which is more often sidetracked to local minima. In particular, if we choose sources incompatible with the initialising algorithm (so that it fails completely), our method can nonetheless find a good solution.[7] Second, when a good initial point is given, the Newton-like algorithm converges faster than gradient descent. Third, our approach runs sufficiently quickly on large-scale problems to be used either as a standalone method (when a good

[7]Note the criterion optimised by FastKICA is also the statistic of an independence test [15, 20]. This test can be applied directly to the values of HSIC between pairs of unmixed sources, to verify the recovered signals are truly independent; no separate hypothesis test is required.

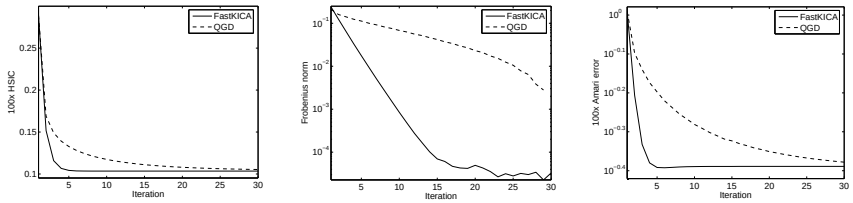


Figure 2: Convergence measured by HSIC, by the Frobenius norm of the difference between the i-th iterate $X^{(i)}$ and $X^{(30)}$, and by the Amari error. FastKICA converges faster. The plots show averages over 25 runs with 40,000 samples from 16 sources.

initialisation is impossible or unlikely), or to fine tune the solution obtained by another method. While not the fastest method tested, demixing performance of FastKICA achieves a reasonable compromise between speed and solution quality, as demonstrated by its performance as the departure of the mixture from independence becomes non-smooth. Finally, FastKICA shows better resistance to outliers than alternative approaches.

Our artificial data for Sections 5.1 (comparison of FastKICA with gradient descent) and 5.2 (computational cost benchmarks) were generated in accordance with [19, Table 3], which is similar to the artificial benchmark data of [5]. Each source was chosen randomly with replacement from 18 different distributions having a wide variety of statistical properties and kurtoses. Sources were mixed using a random matrix with condition number between one and two. Section 5.3 describes our experiments on source smoothness vs performance, and Section 5.4 contains our experiments on outlier resistance. We used the Amari divergence, defined by [4], as an index of ICA algorithm performance (we multiplied this quantity by 100 to make the performance figures more readable). In all experiments, the precision of the incomplete Cholesky decomposition was $10^{-6}n$. Convergence was measured by the difference in HSIC values over consecutive iterations.

5.1 Comparison with Gradient Descent

We first compare the convergence of FastKICA with a simple gradient descent method [23]. In order to find a suitable step width along the gradient mapped to $O(m)$, the latter uses a quadratic interpolation of HSIC along the geodesic. This requires HSIC to be evaluated at two additional points. Both FastKICA and quadratic gradient descent (QGD) use the same gradient and independence measure. Figure 2 compares the convergence for both methods (on the same data) of HSIC, the Amari error, and the Frobenius norm of the difference between the i-th iterate $X^{(i)}$ and the solution $X^{(30)}$ reached after 30 iterations. The results are averaged over 25 runs. In each run, 40,000 observations from 16 artificial, randomly drawn sources were generated and mixed. We initialised

both methods with FastICA [22], and used a kernel width of $\lambda = 0.5$. As illustrated by the plots, FastKICA approaches the solution much faster than QGD. We also observe that the number of iterations to convergence decreases when the sample size grows. The plot of the Frobenius norm suggests that convergence of FastKICA is only linear for the artificial data set. While local quadratic convergence is guaranteed in the population setting, the required properties for Corollary 1 do not hold exactly in the finite sample setting, which can reduce the convergence rate [28].

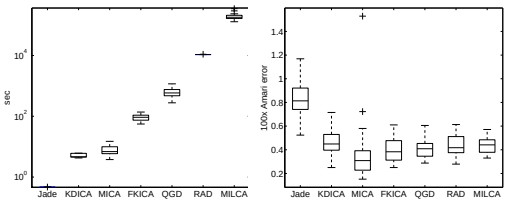
For arbitrary initialisations, FastKICA is still applicable with multiple restarts, although a larger kernel width is more appropriate for the initial stages of the search (local fluctuations in FastKICA far from independence are then smoothed out, although the bias in the location of the global minimum increases). We set $\lambda = 1.0$ and a convergence threshold of 10^{-8} for both FastKICA and QGD. For 40,000 samples from 8 artificial sources, FastKICA converged on average for 37% of the random restarts with an average error ($\times$ 100) of 0.54 ± 0.01, whereas the QGD did not yield any useful results at all (mean error $\times$ 100: 74.14 ± 1.39). Here, averages are over 10 runs with 20 random initialisations each. The solution obtained with FastKICA can be refined further by shrinking the kernel width after initial convergence, to reduce the bias.

5.2 Performance and Cost vs Other Approaches

Our next results compare the performance and computational cost of FastKICA, Jade [9], KDICA [10], MICA [29], MILCA [38], RADICAL [26], and quadratic gradient descent (QGD) [23]. The timing experiments for all methods except MILCA were performed on the same dual AMD Opteron (2x AMD Opteron(tm) Processor 250, 64KiB L1 cache, 1MiB L2 cache, GiB System Memory). Since MILCA was much slower, its tests were run in parallel on 64 bit cluster nodes with 2–16 processors and 7.8–94.6 GB RAM, running Ubuntu 7.04: these nodes were generally faster than the one used for the other algorithms, and the runtime of MILCA is consequently an underestimate, relative to the remaining methods.

We demixed 8 sources and 40,000 observations of the artificial data. The run times include the initialisation by Jade for FastKICA, QGD, and KDICA, and are averaged over 10 repetitions for each data set (except for the slower methods RADICAL and MILCA). QGD was run for 10 iterations, and the convergence threshold for FastKICA was 10^{-5} ($\lambda = 0.5$). MICA was allowed a maximum number of 50 iterations, since with the default 10 iterations, the Amari error was more than twice as high (0.84 ± 0.74), and worse than that all other algorithms besides Jade.

Figure 3 displays the error and time for 24 data sets. We see that demixing performance is very similar across all nonparametric approaches, which perform well on this benchmark. MICA has the best median performance, albeit with two more severely misconverged solutions. While small, the median performance difference is statistically significant according to a level 0.05 sign test. FastKICA and QGD provide the next best result, and exhibit a small but statistically significant performance difference compared with KDICA, RADICAL



	Error $\times 100$	Time [s]
Jade	0.83±0.17	0.46±0.002
KDICA	0.47±0.12	5.09±0.8
MICA	0.37±0.28	8.10±3.4
FKICA	0.40±0.10	91.27±22.9
QGD	0.41±0.08	618.17±208.7
RAD	0.44±0.09	$10.72 \cdot 10^3$±50.38
MILCA	0.44±0.06	$18.97 \cdot 10^4 \pm 5.1 \cdot 10^4$

Figure 3: Comparison of run times (left) and performance (middle) for various ICA algorithms. FastKICA is faster than MILCA, RADICAL, and gradient descent with quadratic approximation, and its results compare favorably to the other methods. KDICA is even faster, but performs less well than FastKICA. Both KDICA and MICA have higher variance than FastKICA. The values are averages over 24 data sets.

and Jade, but not MILCA. The time differences between the various algorithms are much larger than their performance differences. In this case, the ordering is Jade, KDICA, MICA, FastKICA, QGD, RADICAL, and MILCA. The additional evaluations of HSIC for the quadratic approximation make QGD slower per iteration than FastKICA. As shown above, FastKICA also converges in fewer iterations than QGD, requiring 4.32 iterations on average.

We also compared KDICA and FastKICA when random initialisations were used. We see in Figure 5(a) that FastKICA solutions have a clear bivariate distribution, with a large number of initialisations reaching an identical global minimum: indeed, the correct solution is clearly distinguishable from local optima on the basis of its HSIC value. By contrast, KDICA appears to halt at a much wider variety of local minima even for these relatively simple data, as evidenced by the broad range of Amari errors in the estimated unmixing matrices. Thus, in the absence of a good initialising estimate (where classical methods fail), FastKICA is to be preferred. We will further investigate misconvergence behaviour of the different ICA algorithms in the next section, for a more difficult (non-smooth) demixing problem.

5.3 ICA performance as a function of problem smoothness

While the foregoing experiments provide a good idea of computational cost, they do not address the performance of the various ICA methods as a function of the statistical properties of the sources (rather, the performance is an average over sources with a broad variety of behaviours, and is very similar across the various nonparametric approaches). In the present section, we focus specifically on how well the ICA algorithms perform as a function of the smoothness of the ICA problem. Our source distributions take the form of Gaussians with sinusoidal perturbations, and are proportional to $g(x)(1 + \sin(2\pi\nu\beta x))$, where $g(x)$ is a Gaussian probability density function with unit variance, β is the maximum perturbing frequency considered and ν is a scaling factor ranging from 0.05 to

1.05 with spacing 0.05 (the choice of β will be addressed later). The sinusoidal perturbation is multiplied by $g(x)$ to ensure the resulting density expression is everywhere non-negative. Plots of the source probability density function and its characteristic function are given in Figure 4. Bearing in mind that purely Gaussian sources can only be resolved up to rotation [12], the resulting ICA problem becomes more difficult (for algorithms making smoothness assumptions on the sources) as the departure from Gaussianity is encoded at increasing frequencies, which are harder to distinguish from random noise for a given sample size.[8] By contrast, the sources used in the previous section (taken from [19, Table 3]) yield very similar demixing performance when comparing across the nonparametric algorithms used in our benchmarks. One reason for this similarity is that the departure from Gaussianity of these sources has substantial amplitude at low frequencies, resulting in ICA problems of similar difficulty for MICA, KDICA, and FastKICA.[9] We remark that linear mixtures where the departure from independence occurs only at high frequencies are not typical of real-life ICA problems. That said, such mixtures represent an important failure mode of ICA algorithms that make smoothness assumptions on the source densities (as for MICA, KDICA, and FastKICA). Thus, our purpose in this section is to compare the decay in unmixing performance across the various ICA algorithms as this failure mode is approached.

We decide on the base frequency β of the perturbation with reference to the parameters of MICA, to simplify the discussion of performance. The MICA algorithm optimizes a sum of entropies of each mixture component, where the entropies are computed using discretised empirical estimates of the mixture probability distributions. We can express the distribution estimates by first convolving the mixture sample by a B-spline kernel (of order 3, although other orders are also possible) and then downsampling to get probability estimates at the gridpoints. If we consider the population setting and perform these operations in the frequency domain, this corresponds to multiplying the Fourier transform of the source density by that of the B-spline, and then aliasing the frequency components that exceed the grid Nyquist frequency.

Given a baseline bandwidth b_0, the grid spacing is computed as a function of the sample size n according to $b_n = b_0 \times 2.107683/n^{0.2}$; without loss of generality, we set $b_0 = 1$. The spline kernel bandwidth is also scaled by this factor, such that the zeros in the kernel spectrum occur at integer multiples of $f_m := n^{0.2}/(2.107683)$. To use these factors in setting β, consider two sources consisting of perturbed Gaussians with identical β. The characteristic function of the two mixtures resulting from a rotation with angle $\pi/4$ has a distinctive peak at $\beta\sqrt{2}$ when $\nu = 1$. Thus, by setting $\beta = n^{0.2}/(2.107683\sqrt{2})$, this peak

[8] This perturbation to the Gaussian distribution differs from that used for designing contrast functions in classical ICA studies, which employ Edgeworth [12] or Gram-Charlier [4] expansions. We shall see, however, that our perturbed sources are better suited to characterizing the interaction between ICA performance and the choice of kernel, for methods using kernel density estimates (MICA, and KDICA) and for FastKICA.

[9] As we shall see, the spacings-based entropy estimates of RADICAL and the graph-based mutual information estimates of MILCA behave quite differently.

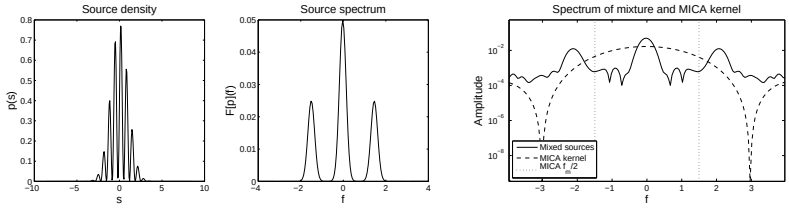


Figure 4: **Left:** Source probability density function for a perturbation at frequency $0.7 f_m/\sqrt{2}$, where f_m is the frequency of the first zero in the spectrum of the spline kernel used by MICA. **Middle:** Characteristic function of the source, showing peaks at the perturbation frequency. **Right:** Empirical (smoothed) characteristic function of the mixture of two sources with angle $\pi/4$. Two peaks are seen at locations $0.7 f_m$. The spectrum of the MICA kernel (a 3rd order B-spline) is superposed. The dashed vertical lines are at $f_m/2$, which is the Nyquist frequency for the grid used by MICA. Thus, perturbations exceeding this frequency will be aliased.

will fall at the first minimum of the spline kernel spectrum. An illustration is provided in Figure 4. Note in particular the decay of the spline spectrum towards its first zero at f_m: any component of the mixture characteristic function at this frequency will be severely attenuated, and thus we expect that demixing sources with ν approaching 1 will be difficult (the sources will appear Gaussian).

We sampled 25 data sets consisting of two sources with $n = 1,000$ for each value of ν, and mixed the sources with orthonormal matrices. To ensure comparable results, we used the same set of 25 mixing matrices across all ν. The algorithms were run for a maximum of 50 iterations. The convergence threshold for FastKICA was a 0.5% change of HSIC, and the bandwidth $\lambda = 0.5$. MICA used a bandwidth of $b_0 = 1$. For these bandwidth choices, we emphasise that both FKICA and MICA reached chance level performance (i.e. complete failure) at the same source perturbing frequency, corresponding to $\nu = 1$, making the behaviour of these two methods across the ν range directly comparable. In other words, we report the relative performance of the two algorithms over the frequency range for which they operate at better than chance level. The Amari errors in Figure 5(b) are averages over 10 random initialisations and the 25 data sets for each frequency. Each algorithm was initialised with the same 10 random orthonormal matrices.

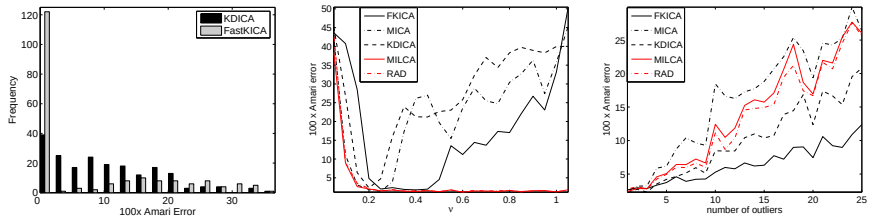
We note first of all that FastKICA has a longer ν interval in which the average Amari error is very low, compared with MICA and KDICA. In addition, as ν rises above 0.5, the average error of FastKICA is consistently below that of MICA and KDICA. On the other hand, for the lowest perturbing frequencies, KDICA and MICA perform better than FastKICA. The two most computationally costly methods, RADICAL and MILCA, perform best, with a low Amari divergence over all the high ν values tested. This is as expected, for two reasons: first, both methods perform an exhaustive search over all pairs of

Jacobi angles, and are not susceptible to local minima. Second, RADICAL is based on a spacings estimate of entropy, and MILCA on a k-nn estimate of the mutual information: in other words, both methods adapt automatically to the scale of the variations in the mixture densities. That said, efficient optimization techniques have yet to be developed for RADICAL and MILCA.

We next examine in more detail the convergence behaviour leading to the drop in average performance of FastKICA, MICA, and KDICA as ν rises. First, as noted in the previous section, KDICA can be sensitive to local minima: thus its average performance degrades even for low values of ν as a large number of initialisations result in misconvergence. The behaviour of MICA is more complex, with performance dropping at $\nu \approx 0.4$ but recovering for $\nu \approx 0.55$ (two more such oscillations occur at higher ν). At $\nu \approx 0.4$, the MICA entropy score develops a local minimum at a rotation of $\pi/4$ from the true unmixing matrix, resulting in a substantial number of initializations converging to this incorrect solution, as well as a group of correct solutions (this local minimum is also seen for other b_0 values, but at different onset values of ν). The local minimum becomes less pronounced at $\nu \approx 0.55$, but then strengthens again at $\nu \approx 0.65$. By contrast, the results for FastKICA at moderate values of ν more closely follow the histogram of Figure 5(a), with a large fraction of solutions at the global optimum, and the remaining misconverged solutions having a range of Amari errors. Taking the best solution over all 10 initializations (as measured by the HSIC or entropy score), rather than the average solution, the results of MICA and FastKICA at larger ν both remain indistinguishable from RADICAL and MILCA until $\nu \approx 0.8$. For $\nu > 0.8$, performance worsens towards chance level as ν rises to 1, at which point the global optimum of both contrast functions occurs at a random angle. The onset of this performance drop can be increased for both FastKICA and MICA by decreasing b_0 or λ, respectively; but at a cost of worse mean performance due to more pronounced local minima. On the other hand, the best KDICA result continues to perform as well as RADICAL and MILCA, since the slow decaying Fourier transform of its Laplace kernel makes it sensitive to higher frequencies.

5.4 Resistance to Outliers

In our final experiment, we investigate the effect of outlier noise added to the observations. We selected two generating distributions from the benchmark data in [19, Table 3], randomly and with replacement. After combining these signals with a randomly generated matrix with condition number between 1 and 2, we generated a varying number of outliers by adding ± 5 (with equal probability) to both signals at random locations. We evaluated HSIC using a Gaussian kernel of size $\lambda = 0.5$. For FastKICA and MICA, we chose the best result out of 3 random initialisations, according to HSIC or the estimated entropy, respectively. The initialisation for KDICA was the FastKICA result, since the KDICA is sensitive to poor initialisation (as seen in Section 5.2). Results are shown in Figure 5(c). It is clear that FastKICA substantially outperforms the alternatives in outlier resistance.



(a) Misconvergence (b) Performance vs. smoothness (c) Outlier results

Figure 5: (a) Comparison of performance for arbitrary initialisations (n = 40,000, $m = 8$). Amari error histograms are shown for FastKICA vs. KDICA with mixed artificial sources (10 data sets, 20 initialisations each). FastKICA reaches a global minimum far more often than KDICA. (b) Amari error depending on the sine frequencies for $n = 1000$ samples and two sources. (c) Effect of outliers on the performance of the ICA algorithms, for two sources of length $n = 1000$, drawn independently with replacement from [19, Table 3], and corrupted at random observations with outliers at ± 5 (where each sign has probability 0.5). Each point represents an average over 100 independent experiments. The number of corrupted observations in both signals is given on the horizontal axis.

6 Conclusion

We demonstrate that an approximate Newton-like method, FastKICA, can improve the speed and performance of kernel/characteristic function-based ICA methods. We emphasise that FastKICA is applicable even if no good initialisation is at hand. With a modest number of restarts and a kernel width that shrinks near independence (on our data, from $\lambda = 1.0$ to $\lambda = 0.5$), the correct global optimum is consistently found. A good initialisation results in more rapid convergence, and we do not need to adapt the kernel size. Our method demonstrates much better outlier resistance than recently published competing approaches. Moreover, our optimization method can be applied to any twice differentiable RKHS kernel, rather than relying on the specific properties of particular kernels (to be Laplace in the case of [10], or to be a spline kernel with compact support in [29]).

Several directions for future work are suggested by the present study. First, the kernel bandwidth used is currently chosen heuristically. It would be of interest to develop more principled methods for choosing this bandwidth based on properties of the data. Second, it is notable that ICA methods based on spacings estimates of entropy, or nearest-neighbour estimates of mutual information, perform very well for ICA problems where the departure from independence is encoded at high frequencies. Unfortunately, efficient optimization techniques have yet to be developed for ICA using these dependence measures.

A Evaluation of the Second Derivative of the HSIC Based ICA Contrast

We assume the unmixing matrix is correct, and thus $X = X^*$. The term (15a) for a fixed pair (i, j) can be computed as

$$\begin{aligned} &\mathbb{E}_{k,l}\left[\phi''\left(\overline{s}_{kli}\right)\omega_i^\top \overline{s}_{kl}\overline{s}_{kl}^\top \omega_i \phi\left(\overline{s}_{klj}\right)\right] \\ =& \sum_{r,t=1;r,t\neq i}^{m} \omega_{ir}\omega_{it}\mathbb{E}_{k,l}\left[\phi''\left(\overline{s}_{kli}\right)\overline{s}_{klr}\overline{s}_{klt}\phi\left(\overline{s}_{klj}\right)\right]. \end{aligned} \tag{37}$$

Under the assumptions of independence and whitened mixtures, the corresponding (r, t) expression can be written

$$\begin{cases} 0, & r \neq t; \\ 2\mathbb{E}_{k,l}\left[\phi''\left(\overline{s}_{kli}\right)\phi\left(\overline{s}_{klj}\right)\right], & r = t \neq i, j; \\ \mathbb{E}_{k,l}[\phi''\left(\overline{s}_{kli}\right)]\mathbb{E}_{k,l}\left[\overline{s}_{klj}^2\phi\left(\overline{s}_{klj}\right)\right], & r = t = j. \end{cases} \tag{38}$$

Thus the term (15a) can be further simplified as

$$\begin{aligned} (15a):\ &\mathbb{E}_{k,l}\left[\phi''\left(\overline{s}_{kli}\right)\omega_i^\top \overline{s}_{kl}\overline{s}_{kl}^\top \omega_i \phi\left(\overline{s}_{klj}\right)\right] \\ =& \sum_{r=1;r\neq i,j}^{m} 2\omega_{ir}^2\mathbb{E}_{k,l}[\phi''\left(\overline{s}_{kli}\right)]\mathbb{E}_{k,l}[\phi\left(\overline{s}_{klj}\right)] \\ &+ \omega_{ij}^2\mathbb{E}_{k,l}[\phi''\left(\overline{s}_{kli}\right)]\mathbb{E}_{k,l}\left[\overline{s}_{klj}^2\phi\left(\overline{s}_{klj}\right)\right]. \end{aligned} \tag{39}$$

By applying the same techniques, the remaining terms (15b)–(15i) become

$$\begin{aligned} (15b):\ &\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{kli}\right)\omega_i^\top \Omega\overline{s}_{kl}\phi\left(\overline{s}_{klj}\right)\right] \\ =& \sum_{r=1;r\neq i}^{m} -\omega_{ir}^2\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{kli}\right)\overline{s}_{kli}\right]\mathbb{E}_{k,l}\left[\phi\left(\overline{s}_{klj}\right)\right], \end{aligned} \tag{40}$$

$$\begin{aligned} (15c):\ &\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{kli}\right)\omega_i^\top \overline{s}_{kl}\overline{s}_{kl}^\top \omega_j \phi'\left(\overline{s}_{klj}\right)\right] \\ =& -\omega_{ij}^2\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{kli}\right)\overline{s}_{kli}\right]\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{klj}\right)\overline{s}_{klj}\right], \end{aligned} \tag{41}$$

$$\begin{aligned} (15d):\ &\mathbb{E}_{k,l}\left[\phi''\left(\overline{s}_{kli}\right)\omega_i^\top \overline{s}_{kl}\overline{s}_{kl}^\top \omega_i\right]\mathbb{E}_{k,l}\left[\phi\left(\overline{s}_{klj}\right)\right] \\ =& \sum_{r=1;r\neq i}^{m} 2\omega_{ir}^2\mathbb{E}_{k,l}\left[\phi''\left(\overline{s}_{kli}\right)\right]\mathbb{E}_{k,l}\left[\phi\left(\overline{s}_{klj}\right)\right], \end{aligned} \tag{42}$$

$$\begin{aligned} (15e):\ &\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{kli}\right)\omega_i^\top \Omega\overline{s}_{kl}\right]\mathbb{E}_{k,l}\left[\phi\left(\overline{s}_{klj}\right)\right] \\ =& \sum_{r=1;r\neq i}^{m} -\omega_{ir}^2\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{kli}\right)\overline{s}_{kli}\right]\mathbb{E}_{k,l}\left[\phi\left(\overline{s}_{klj}\right)\right], \end{aligned} \tag{43}$$

$$(15f):\ \mathbb{E}_{k,l}\left[\phi'(\overline{s}_{kli})\,\omega_i^\top \overline{s}_{kl}\right]\mathbb{E}_{k,l}\left[\phi'(\overline{s}_{klj})\,\omega_j^\top \overline{s}_{kl}\right] = 0, \tag{44}$$

$$
\begin{aligned}
&(15\text{g}): \mathbb{E}_k\left[\mathbb{E}_l\left[\phi''\left(\overline{s}_{kli}\right)\omega_i^\top \overline{s}_{kl}\overline{s}_{kl}^\top\omega_i\right]\mathbb{E}_l\left[\phi\left(\overline{s}_{klj}\right)\right]\right] \\
&\quad = \sum_{r=1;r\neq i,j}^{m} 2\omega_{ir}^2\mathbb{E}_{k,l}\left[\phi''\left(\overline{s}_{kli}\right)\right]\mathbb{E}_{k,l}\left[\phi\left(\overline{s}_{klj}\right)\right] \\
&\qquad + \omega_{ij}^2\mathbb{E}_{k,l}[\phi''(\overline{s}_{kli})]\,\mathbb{E}_k\left[\mathbb{E}_l\left[\overline{s}_{klj}^2\right]\mathbb{E}_l\left[\phi(\overline{s}_{klj})\right]\right],
\end{aligned} \tag{45}
$$

$$
\begin{aligned}
&(15\text{h}): \mathbb{E}_k\left[\mathbb{E}_l\left[\phi'\left(\overline{s}_{kli}\right)\omega_i^\top \Omega\overline{s}_{kl}\right]\mathbb{E}_l\left[\phi\left(\overline{s}_{klj}\right)\right]\right] \\
&\quad = \sum_{r=1;r\neq i}^{m} -\omega_{ir}^2\mathbb{E}_{k,l}\left[\phi'\left(\overline{s}_{kli}\right)\overline{s}_{kli}\right]\mathbb{E}_{k,l}\left[\phi\left(\overline{s}_{klj}\right)\right],
\end{aligned} \tag{46}
$$

$$
\begin{aligned}
&(15\text{i}): \mathbb{E}_k\left[\mathbb{E}_l\left[\phi'(\overline{s}_{kli})\,\omega_i^\top \overline{s}_{kl}\right]\mathbb{E}_l\left[\phi'(\overline{s}_{klj})\,\omega_j^\top \overline{s}_{kl}\right]\right] \\
&= -\omega_{ij}^2\mathbb{E}_k[\mathbb{E}_l[\overline{s}_{kli}]\mathbb{E}_l[\phi'(\overline{s}_{kli})]]\mathbb{E}_k[\mathbb{E}_l[\overline{s}_{klj}]\mathbb{E}_l[\phi'(\overline{s}_{klj})]].
\end{aligned} \tag{47}
$$

Substituting (39)–(47) into equation (15), the result in equation (16) follows directly from $\omega_{ij} = -\omega_{ji}$.

B Derivation of the approximate gradient and the matrix partial derivatives

In this appendix, we first derive the differential dK' of the Cholesky approximation to the Gram matrix K, and use it to obtain the differential of the approximate HSIC in (36). We then give an expression for the differential of the factors of K', which involves the entry-wise derivative of the Gram matrix with respect to a column x of X. Details have been published in [23].

B.1 Differential of the incomplete Cholesky approximation to HSIC

Recall that the differential of the low-rank approximation to HSIC is

$$
d\mathrm{tr}\left(\widetilde{K}'\widetilde{L}'\right) = \mathrm{tr}\left(\widetilde{K}'d(L')\right) + \mathrm{tr}\left(\widetilde{L}'d(K')\right). \tag{48}
$$

We expand the differential of $K' = K_{:,I}K_{I,I}^{-1}K_{I,:}$ by the product rule and by rewriting $d(K_{I,I}^{-1}) = K_{I,I}^{-1}(dK_{I,:})K_{I,I}^{-1}$. Plugging the result into the second term $\mathrm{tr}\left(\widetilde{L}'d(K')\right)$ from (48) yields

$$
\begin{aligned}
&\mathrm{tr}\left(\widetilde{L}'d(K')\right) \\
&= \mathrm{tr}\left(\widetilde{L}'\mathrm{d}K_{:,I}K_{I,I}^{-1}K_{I,:}\right) + \mathrm{tr}\left(\widetilde{L}'K_{:,I}K_{I,I}^{-1}dK_{I,:}\right) \\
&\quad - \mathrm{tr}\left(\widetilde{L}'K_{:,I}K_{I,I}^{-1}(dK_{I,I})K_{I,I}^{-1}K_{I,:}\right).
\end{aligned} \tag{49}
$$

Using the symmetry of the Gram matrices, the second term on the right hand side can be transformed as

$$\begin{aligned}\operatorname{tr}\left(\widetilde{L}' K_{:,I} K_{I,I}^{-1} dK_{I,:}\right) &= \operatorname{tr}\left((dK_{:,I} K_{I,I}^{-1} K_{I,:})^\top \widetilde{L}'^\top\right) \\ &= \operatorname{tr}\left(\widetilde{L}' dK_{:,I} K_{I,I}^{-1} K_{I,:}\right),\end{aligned}$$

and (49) becomes

$$\begin{aligned}\operatorname{tr}\left(\widetilde{L}' d(K')\right) &= 2\operatorname{tr}\left(K_{I,I}^{-1} K_{I,:} \widetilde{L}' dK_{:,I}\right) - \operatorname{tr}\left(K_{I,I}^{-1} K_{I,:} \widetilde{L}' K_{:,I} K_{I,I}^{-1} dK_{I,I}\right) \\ &= 2\operatorname{vec}(\widetilde{L}' K_{:,I} K_{I,I}^{-1})^\top \operatorname{vec}(dK_{:,I}) \\ &\quad - \operatorname{vec}(K_{I,I}^{-1} K_{I,:} \widetilde{L}' K_{:,I} K_{I,I}^{-1})^\top \operatorname{vec}(dK_{I,I}).\end{aligned}$$

The other term $\operatorname{tr}\left(\widetilde{K}' d(L')\right)$ in (48) is equivalent.

B.2 Derivative of the Gram matrix with respect to X

The derivative of the Gram matrix entries with respect to a particular column x of the unmixing matrix X depends on the kernel. We employ a Gaussian kernel here, but one could easily obtain the derivatives of additional kernels: these can then be plugged straightforwardly into the equations in the previous section.

Lemma 3 (Derivative of K with respect to X)**.** *Let K be the Gram matrix computed with a Gaussian kernel, and let x be an $m \times 1$ column of the unmixing matrix, such that the (i,j)th entry of K is*

$$k_{ij} = \phi(y_i, y_j) = \exp\left[\frac{-1}{2\lambda^2} x^\top W_{ij} x\right],$$

where $W_{ij} = (w_i - w_j)(w_i - w_j)^\top$, and w_i is the ith sample of observations. Then the derivative of any k_{ij} with respect to x is

$$\frac{\partial k_{ij}}{\partial x^\top} = -\frac{k_{ij}}{\lambda^2} x^\top (w_i - w_j)(w_i - w_j)^\top.$$

Since the above derivative is a vector, we require appropriate notation to express the derivative of the entire Gram matrix in a tractable form. This is done using the $\operatorname{vec}(A)$ operation, which stacks the columns of the matrix A on top of each other. Thus, the resulting differential is

$$d(\operatorname{vec} K) = \underbrace{\left[\frac{\partial k_{11}}{\partial x}, \ldots, \frac{\partial k_{n1}}{\partial x}, \frac{\partial k_{12}}{\partial x}, \ldots, \frac{\partial k_{nn}}{\partial x}\right]^\top}_{\partial \operatorname{vec}(K)/\partial x^\top} d(\operatorname{vec} x), \tag{50}$$

where $d(\operatorname{vec} x) = dx$ and $\partial k_{11}/\partial x = (\partial k_{11}/\partial x^\top)^\top$. The derivatives of the submatrices $\partial \operatorname{vec}(K_{:,I})/\partial x^\top$ and $\partial \operatorname{vec}(K_{I,I})/\partial x^\top$ are submatrices of $\partial \operatorname{vec}(K)/\partial x^\top$, restricted to the rows $\partial k_{ij}/\partial x^\top$, with $1 \leq i \leq n$ and $j \in I$, or both $i, j \in I$, respectively.

Acknowledgment

The authors would like to thank Aiyou Chen for providing us with his KDICA code, and Alex Smola and Knut Hüper for valuable discussions. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

The present work started when H. Shen was with the Department of Information Engineering, The Australian National University, Australia and NICTA, Australia.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ, 2008.

[2] S. Achard, D.-T. Pham, and C. Jutten. Quadratic dependence measure for non-linear blind sources separation. In Fourth International Symposium on ICA and BSS, pages 263–268, 2003.

[3] S. Achard, C. Jutten, and D.-T. Pham. Mutual information and quadratic dependence for post nonlinear blind source separation. In EUSIPCO 13, 2005.

[4] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In NIPS 8, pages 757–763. The MIT Press, 1996.

[5] F. R. Bach and M. I. Jordan. Kernel independent component analysis. Journal of Machine Learning Research, 3:1–48, 2002.

[6] W. M. Boothby. An Introduction to Differentiable Manifolds and Riemannian Geometry, Revised. Academic Press, 2nd edition, 2002.

[7] R. Boscolo, H. Pan, and V. P. Roychowdhury. Independent component analysis based on nonparametric density estimation. IEEE Transactions on Neural Networks, 15(1):55–65, 2004.

[8] J.-F. Cardoso. Infomax and maximum likelihood for blind source separation. IEEE Letters on Signal Processing, 4:112–114, 1997.

[9] J.-F. Cardoso. Blind signal separation: Statistical principles. Proceedings of the IEEE, Special Issue on Blind Identification and Estimation, 86(10): 2009–2025, 1998.

[10] A. Chen. Fast kernel density independent component analysis. In Sixth International Conference on ICA and BSS, volume 3889, pages 24–31, Berlin/Heidelberg, 2006. Springer-Verlag.

[11] A. Chen and P. J. Bickel. Consistent independent component analysis and prewhitening. IEEE Transactions on Signal Processing, 53(10):3625–3632, 2005.

[12] P. Comon. Independent component analysis, a new concept? Signal Processing, 36(3):287–314, 1994.

[13] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications, 20(2):303–353, 1998.

[14] J. Eriksson and K. Koivunen. Characteristic-function based independent component analysis. Signal Processing, 83(10):2195–2208, 2003.

[15] A. Feuerverger. A consistent test for bivariate dependence. International Statistical Review, 61(3):419–433, 1993.

[16] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. Journal of Machine Learning Research, 2(22):243–264, 2001.

[17] D. Gabay. Minimizing a differentiable function over a differential manifold. Journal of Optimization Theory and Applications, 37(2):177–219, 1982.

[18] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In ALT, pages 63–78, Heidelberg, 2005. Springer-Verlag.

[19] A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. Journal of Machine Learning Research, 6:2075–2129, 2005.

[20] Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. In NIPS 20, pages 585–592. MIT Press, Cambridge, MA, 2008.

[21] K. Hüper and J. Trumpf. Newton-like methods for numerical optimisation on manifolds. In Proceedings of Thirty-eighth Asilomar Conference on Signals, Systems and Computers, pages 136–139, 2004.

[22] A. Hyvärinen, J. Karhunen, and E. Oja. Independent Component Analysis. Wiley, New York, 2001.

[23] S. Jegelka and A. Gretton. Large-Scale Kernel Machines, chapter Brisk kernel ICA. MIT Press, Cambridge, Massachusetts, 2007.

[24] A. Kankainen. Consistent testing of total independence based on empirical characteristic function. PhD thesis, University of Jyväskylä, 1995.

[25] M. Kleinsteuber. Jacobi-Type Methods on Semisimple Lie Algebras – A Lie Algebraic Approach to the Symmetric Eigenvalue Problem. PhD thesis, Bayerische Julius-Maximilians-Universität Würzburg, 2006.

[26] E. G. Learned-Miller and J. W. Fisher III. ICA using spacings estimates of entropy. Journal of Machine Learning Research, 4:1271–1295, 2003.

[27] C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. Journal of Machine Learning Research, 7:2651–2667, 2006.

[28] E. Oja and Z. Yuan. The FastICA algorithm revisited: Convergence analysis. IEEE Transactions on Neural Networks, 17(6):1370–1381, 2006.

[29] D.-T. Pham. Fast algorithms for mutual information based independent component analysis. IEEE Transactions on Signal Processing, 52(10):2690–2700, 2004.

[30] M. Rosenblatt. A quadratic measure of deviation of two-dimensional density estimates and a test of independence. The Annals of Statistics, 3(1): 1–14, 1975.

[31] B. Schölkopf and A. Smola. Learning with Kernels. MIT Press, Cambridge, MA, 2002.

[32] H. Shen and K. Hüper. Newton-like methods for parallel independent component analysis. In MLSP 16, pages 283–288, Maynooth, Ireland, 2006.

[33] H. Shen, S. Jegelka, and A. Gretton. Geometric analysis of Hilbert-Schmidt independence criterion based ICA contrast function. National ICT Australia Technical Report (PA006080), ISSN 1833-9646, October 2006.

[34] H. Shen, K. Hüper, and A.-K. Seghouane. Geometric optimisation and FastICA algorithms. In MTNS 17, pages 1412–1418, Kyoto, Japan, 2006.

[35] H. Shen, S. Jegelka, and A. Gretton. Fast kernel ICA using an approximate Newton method. In AISTATS 11, pages 476–483. Microtome, 2007.

[36] M. Spivak. A Comprehensive Introduction to Differential Geometry, Vol. 1 – 5. Publish or Perish, Inc, 3rd edition, 1999.

[37] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. Journal of Machine Learning Research, 2:67–93, 2002.

[38] H. Stögbauer, A. Kraskov, S. Astakhov, and P. Grassberger. Least dependent component analysis based on mutual information. Phys. Rev. E, 70 (6):066123, 2004.

[39] C. K. I. Williams and M. Seeger. Using the Nystrom method to speed up kernel machines. In NIPS 14, pages 682–688. The MIT Press, 2001.