Journal of Machine Learning Research 3 (2002) 1-48 Submitted 11/01; Published 07/02

# Kernel Independent Component Analysis

**Francis R. Bach** FBACH@CS.BERKELEY.EDU
*Computer Science Division*
*University of California*
*Berkeley, CA 94720, USA*

**Michael I. Jordan** JORDAN@CS.BERKELEY.EDU
*Computer Science Division and Department of Statistics*
*University of California*
*Berkeley, CA 94720, USA*

**Editor:** John Shawe-Taylor

## Abstract

We present a class of algorithms for independent component analysis (ICA) which use contrast functions based on canonical correlations in a reproducing kernel Hilbert space. On the one hand, we show that our contrast functions are related to mutual information and have desirable mathematical properties as measures of statistical dependence. On the other hand, building on recent developments in kernel methods, we show that these criteria and their derivatives can be computed efficiently. Minimizing these criteria leads to flexible and robust algorithms for ICA. We illustrate with simulations involving a wide variety of source distributions, showing that our algorithms outperform many of the presently known algorithms.

**Keywords:** kernel methods, independent component analysis, blind source separation, mutual information, Gram matrices, canonical correlations, semiparametric models, integral equations, Stiefel manifold, incomplete Cholesky decomposition

## 1. Introduction

Independent component analysis (ICA) is the problem of recovering a latent random vector $x = (x_1, \ldots, x_m)^\top$ from observations of $m$ unknown linear functions of that vector. The components of $x$ are assumed to be mutually independent. Thus, an observation $y = (y_1, \ldots, y_m)^\top$ is modeled as:

$$y = Ax, \tag{1}$$

where $x$ is a latent random vector with independent components, and where $A$ is an $m \times m$ matrix of parameters. Given $N$ independently, identically distributed observations of $y$, we hope to estimate $A$ and thereby to recover the latent vector $x$ corresponding to any particular $y$ by solving a linear system.

By specifying distributions for the components $x_i$, one obtains a parametric model that can be estimated via maximum likelihood (Bell and Sejnowski, 1995, Cardoso, 1999). Working with $W = A^{-1}$ as the parameterization, one readily obtains a gradient or fixed-point algorithm that yields an estimate $\hat{W}$ and provides estimates of the latent components via $\hat{x} = \hat{W}y$ (Hyvärinen et al., 2001).

In practical applications, however, one does not generally know the distributions of the components $x_i$, and it is preferable to view the ICA model in Eq. (1) as a *semiparametric model* in which the distributions of the components of $x$ are left unspecified (Bickel et al., 1998). Maximizing the likelihood in the semiparametric ICA model is essentially equivalent to minimizing the mutual information between the components of the estimate $\hat{x} = \hat{W}y$ (Cardoso, 1999). Thus it is natural to view mutual information as a *contrast function* to be minimized in estimating the ICA model. Moreover, given that the mutual information of a random vector is nonnegative, and zero if and only if the components of the vector are independent, the use of mutual information as a function to be minimized is well motivated, quite apart from the link to maximum likelihood (Comon, 1994).

Unfortunately, the mutual information is difficult to approximate and optimize on the basis of a finite sample, and much research on ICA has focused on alternative contrast functions (Amari et al., 1996, Comon, 1994, Hyvärinen and Oja, 1997). These have either been derived as expansion-based approximations to the mutual information, or have had a looser relationship to the mutual information, essentially borrowing its key property of being equal to zero if and only if the arguments to the function are independent.

The earliest ICA algorithms were (in retrospect) based on contrast functions defined in terms of expectations of a single fixed nonlinear function, chosen in an ad-hoc manner (Jutten and Herault, 1991). More sophisticated algorithms have been obtained by careful choice of a single fixed nonlinear function, such that the expectations of this function yield a robust approximation to the mutual information (Hyvärinen and Oja, 1997). An interesting feature of this approach is that links can be made to the parametric maximum likelihood formulation, in which the nonlinearities in the contrast function are related to the assumed densities of the independent components. All of these developments have helped to focus attention on the choice of particular nonlinearities as the key to the ICA problem.

In the current paper, we provide a new approach to the ICA problem based not on a single nonlinear function, but on an entire function space of candidate nonlinearities. In particular, we work with the functions in a reproducing kernel Hilbert space, and make use of the "kernel trick" to search over this space efficiently. The use of a function space makes it possible to adapt to a variety of sources and thus makes our algorithms more robust to varying source distributions, as illustrated in Section 7.

We define a contrast function in terms of a rather direct measure of the dependence of a set of random variables. Considering the case of two univariate random variables $x_1$ and $x_2$, for simplicity, and letting $\mathcal{F}$ be a vector space of functions from $\mathbb{R}$ to $\mathbb{R}$, define the *$\mathcal{F}$-correlation* $\rho_{\mathcal{F}}$ as the maximal correlation between the random variables $f_1(x_1)$ and $f_2(x_2)$, where $f_1$ and $f_2$ range over $\mathcal{F}$:

$$\rho_{\mathcal{F}} = \max_{f_1, f_2 \in \mathcal{F}} \operatorname{corr}(f_1(x_1), f_2(x_2)) = \max_{f_1, f_2 \in \mathcal{F}} \frac{\operatorname{cov}(f_1(x_1), f_2(x_2))}{(\operatorname{var} f_1(x_1))^{1/2} (\operatorname{var} f_2(x_2))^{1/2}}.$$

Clearly, if the variables $x_1$ and $x_2$ are independent, then the $\mathcal{F}$-correlation is equal to zero. Moreover, if the set $\mathcal{F}$ is large enough, the converse is also true. For example, it is well known that if $\mathcal{F}$ contains the Fourier basis (all functions of the form $x \mapsto e^{i\omega x}$ where $\omega \in \mathbb{R}$), then $\rho_{\mathcal{F}} = 0$ implies that $x_1$ and $x_2$ are independent.

To obtain a computationally tractable implementation of the $\mathcal{F}$-correlation, we make use of reproducing kernel Hilbert space (RKHS) ideas. Let $\mathcal{F}$ be an RKHS on $\mathbb{R}$, let $K(x, y)$ be

the associated kernel, and let $\Phi(x) = K(\cdot, x)$ be the feature map, where $K(\cdot, x)$ is a function in $\mathcal{F}$ for each $x$. We then have the well-known *reproducing property* (Saitoh, 1988):

$$f(x) = \langle \Phi(x), f \rangle, \quad \forall f \in \mathcal{F}, \ \forall x \in \mathbb{R}.$$

This implies:

$$\operatorname{corr}(f_1(x_1), f_2(x_2)) = \operatorname{corr}\left(\langle \Phi(x_1), f_1 \rangle, \langle \Phi(x_2), f_2 \rangle\right).$$

Consequently, the $\mathcal{F}$-correlation is the maximal possible correlation between one-dimensional linear projections of $\Phi(x_1)$ and $\Phi(x_2)$. This is exactly the definition of the first *canonical correlation* between $\Phi(x_1)$ and $\Phi(x_2)$ (Hotelling, 1936). This suggests that we can base an ICA contrast function on the computation of a canonical correlation in function space.

Canonical correlation analysis (CCA) is a multivariate statistical technique similar in spirit to principal component analysis (PCA). While PCA works with a single random vector and maximizes the variance of projections of the data, CCA works with a pair of random vectors (or in general with a set of $m$ random vectors) and maximizes correlation between sets of projections. While PCA leads to an eigenvector problem, CCA leads to a generalized eigenvector problem. Finally, just as PCA can be carried out efficiently in an RKHS by making use of the "kernel trick" (Schölkopf et al., 1998), so too can CCA (as we show in Section 3.2). Thus we can employ a "kernelized" version of CCA to compute a flexible contrast function for ICA.

There are several issues that must be faced in order to turn this line of reasoning into an ICA algorithm. First, we must show that the $\mathcal{F}$-correlation in fact has the properties that are required of a contrast function; we do this in Section 3.1. Second, we must show how to formulate the canonical correlation problem with $m$ random variables, and show how to solve the problem efficiently using kernel functions. This is easily done, as we show in Section 3.2. Third, our method turns out to require the computation of generalized eigenvectors of matrices of size $mN \times mN$. A naive implementation of our algorithm would therefore require $O(m^3N^3)$ operations. As we show in Section 4, however, by making use of incomplete Cholesky decomposition we are able to solve the kernelized CCA problem in time $O(N(h(N/\eta))^2)$, where $\eta$ is a precision parameter and $h(t)$ is a slowly growing function of $t$. Moreover, in computing the contrast function, the precision $\eta$ need only be linear in $N$; consequently, we have a linear time algorithm. Finally, our goal is not simply that of computing the contrast function, but of optimizing it, and for this we require derivatives of the contrast function. Although incomplete Cholesky factorization cannot be used directly for computing these derivatives, we are able to derive an algorithm for computing derivatives with similar linear complexity in $N$ (see Section 4.6).

There are a number of other interesting relationships between CCA and ICA that we explore in this paper. In particular, for Gaussian variables the CCA spectrum (i.e., all of the eigenvalues of the generalized eigenvector problem) can be used to compute the mutual information (essentially as a product of these eigenvalues). This suggests a general connection between our contrast function and the mutual information, and it also suggests an alternative contrast function for ICA, one based on all of the eigenvalues and not simply the maximal eigenvalue. We discuss this connection in Section 3.4.

The remainder of the paper is organized as follows. In Section 2, we present background material on CCA, RKHS methods, and ICA. Section 3 provides a discussion of the contrast

functions underlying our new approach to ICA, as well as a high-level description of our ICA algorithms. We discuss the numerical linear algebra underlying our algorithms in Section 4, the optimization methods in Section 5, and the computational complexity in Section 6. Finally, comparative empirical results are presented in Section 7, and we conclude in Section 8.

## 2. Background

In this section we provide enough basic background on canonical correlation, kernel methods and ICA so as to make the paper self-contained. For additional discussion of CCA see Borga et al. (1997), for kernel methods see Schölkopf and Smola (2001), and for ICA see Hyvärinen et al. (2001).

### 2.1 Canonical correlation

Given a random vector $x$, *principal component analysis (PCA)* is concerned with finding a linear transformation such that the components of the transformed vector are uncorrelated. Thus PCA diagonalizes the covariance matrix of $x$. Similarly, given two random vectors, $x_1$ and $x_2$, of dimension $p_1$ and $p_2$, *canonical correlation analysis (CCA)* is concerned with finding a pair of linear transformations such that one component within each set of transformed variables is correlated with a single component in the other set. Thus, the correlation matrix between $x_1$ and $x_2$ is reduced to a block diagonal matrix with blocks of size two, where each block is of the form $\begin{pmatrix} 1 & \rho_i \\ \rho_i & 1 \end{pmatrix}$. The $\rho_i$, at most $p = \min\{p_1, p_2\}$ of which are nonzero, are called the *canonical correlations*.

As in the case of PCA, CCA can be defined recursively, component by component. Indeed, the first canonical correlation can be defined as the maximum possible correlation between the two projections $\xi_1^\top x_1$ and $\xi_2^\top x_2$ of $x_1$ and $x_2$:

$$\begin{aligned}
\rho(x_1, x_2) &= \max_{\xi_1, \xi_2} \operatorname{corr}(\xi_1^\top x_1, \xi_2^\top x_2) \\
&= \max_{\xi_1, \xi_2} \frac{\operatorname{cov}(\xi_1^\top x_1, \xi_2^\top x_2)}{\left(\operatorname{var} \xi_1^\top x_1\right)^{1/2} \left(\operatorname{var} \xi_2^\top x_2\right)^{1/2}} \\
&= \max_{\xi_1, \xi_2} \frac{\xi_1^\top C_{12} \xi_2}{\left(\xi_1^\top C_{11} \xi_1\right)^{1/2} \left(\xi_2^\top C_{22} \xi_2\right)^{1/2}},
\end{aligned}$$

where $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$ denotes the covariance matrix of $(x_1, x_2)$. Taking derivatives with respect to $\xi_1$ and $\xi_2$, we obtain:

$$C_{12}\xi_2 = \frac{\xi_1^\top C_{12} \xi_2}{\xi_1^\top C_{11} \xi_1} \, C_{11}\xi_1$$

and

$$C_{21}\xi_1 = \frac{\xi_1^\top C_{12} \xi_2}{\xi_2^\top C_{22} \xi_2} \, C_{22}\xi_2.$$

Normalizing the vectors $\xi_1$ and $\xi_2$ by letting $\xi_1^\top C_{11}\xi_1 = 1$ and $\xi_2^\top C_{22}\xi_2 = 1$, we see that CCA reduces to the following generalized eigenvalue problem:

$$\begin{pmatrix} 0 & C_{12} \\ C_{21} & 0 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \rho \begin{pmatrix} C_{11} & 0 \\ 0 & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}. \tag{2}$$

This problem has $p_1 + p_2$ eigenvalues: $\{\rho_1, -\rho_1, \ldots, \rho_p, -\rho_p, 0, \ldots, 0\}$.

Note that the generalized eigenvector problem in Eq. (2) can also be written in following form:

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = (1+\rho) \begin{pmatrix} C_{11} & 0 \\ 0 & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix},$$

with eigenvalues $\{1+\rho_1, 1-\rho_1, \ldots, 1+\rho_p, 1-\rho_p, 1, \ldots, 1\}$. Note, moreover, that the problem of finding the maximal generalized eigenvalue, $\lambda_{max} = 1 + \rho_{max}$, where $\rho_{max}$ is the first canonical correlation, is equivalent to finding the minimal generalized eigenvalue, $\lambda_{min} = 1 - \rho_{max}$. In fact, this latter quantity is bounded between zero and one, and turns out to provide a more natural upgrade path when we consider the generalization to more than two variables. Thus henceforth our computational task will be that of finding *minimum* generalized eigenvalues.

#### 2.1.1 Generalizing to more than two variables

There are several ways to generalize CCA to more than two sets of variables (Kettenring, 1971). The generalization that we consider in this paper, justified in Appendix A, is the following. Given $m$ multivariate random variables, $x_1, \ldots, x_m$, we find the smallest generalized eigenvalue $\lambda(x_1, \ldots, x_m)$ of the following problem:

$$\begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1m} \\ C_{21} & C_{22} & \cdots & C_{2m} \\ \vdots & \vdots & & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mm} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & 0 & \cdots & 0 \\ 0 & C_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & C_{mm} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix}, \tag{3}$$

or, in short, $C\xi = \lambda D\xi$, where $C$ is the covariance matrix of $(x_1, x_2, \ldots, x_m)$ and $D$ is the block-diagonal matrix of covariances of the individual vectors $x_i$.

As we discuss in Appendix A, the minimal generalized eigenvalue has the fixed range $[0, 1]$, whereas the maximal generalized eigenvalue has a range dependent on the dimensions of the variables. Thus the minimal generalized eigenvalue is more convenient for our purposes.

### 2.2 Reproducing kernel Hilbert spaces

Let $K(x, y)$ be a Mercer kernel (Saitoh, 1988) on $\mathcal{X} = \mathbb{R}^p$, that is, a function for which the *Gram matrix* $K_{ij} = K(x_i, x_j)$ is positive semidefinite for any collection $\{x_i\}_{i=1,\ldots,N}$ in $\mathcal{X}$. Corresponding to any such kernel $K$ there is a map $\Phi$ from $\mathcal{X}$ to a *feature space* $\mathcal{F}$, such that:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle.$$

That is, the kernel can be used to evaluate an inner product in the feature space. This is often referred to as the "kernel trick."

One particularly attractive instantiation of such a feature space is the *reproducing kernel Hilbert space (RKHS)* associated with $K$. Consider the set of functions $\{K(\cdot, x) : x \in \mathcal{X}\}$, where the dot represents the argument to a given function and $x$ indexes the set of functions. Define a linear function space as the span of such functions. Such a function space is unique and can always be completed into a Hilbert space (Saitoh, 1988). The crucial property of these Hilbert spaces is the "reproducing property" of the kernel:

$$f(x) = \langle K(\cdot, x), f \rangle \quad \forall f \in \mathcal{F}. \tag{4}$$

Note in particular that if we define $\Phi(x) = K(\cdot, x)$ as a map from the input space into the RKHS, then we have:

$$\langle \Phi(x), \Phi(y) \rangle = \langle K(\cdot, x), K(\cdot, y) \rangle = K(x, y),$$

and thus $\Phi(x) = K(\cdot, x)$ is indeed an instantiation of the "kernel trick."

For concreteness we restrict ourselves mostly to translation-invariant kernels in this paper; that is, to kernel functions of the form $K(x, y) = k(x - y)$, where $k$ is a function from $\mathbb{R}^p$ to $\mathbb{R}$. In this case, the feature space $\mathcal{F}$ has infinite dimension and the RKHS can be described succinctly using Fourier theory (Girosi et al., 1995). Indeed, for a given function $k$, $\mathcal{F}$ is composed of functions $f \in L^2(\mathbb{R}^p)$ such that:

$$\int_{\mathbb{R}^p} \frac{|\hat{f}(\omega)|^2}{\nu(\omega)} d\omega < \infty, \tag{5}$$

where $\hat{f}(\omega)$ is the Fourier transform of $f$ and $\nu(\omega)$ is the Fourier transform of $k$ (which must be real and positive to yield a Mercer kernel). This interpretation shows that functions in the RKHS $\mathcal{F}$ have a Fourier transform that decays rapidly, implying that $\mathcal{F}$ is a space of smooth functions.

Finally, consider the case of an isotropic Gaussian kernel in $p$ dimensions:

$$K(x, y) = G_\sigma(x - y) = \exp\left(-\frac{1}{2\sigma^2}||x - y||^2\right).$$

In this case the Fourier transform is $\nu(\omega) = (2\pi\sigma^2)^{p/2} \exp\left(-\frac{\sigma^2}{2}||\omega||^2\right)$, and the feature space $\mathcal{F}_\sigma$ contains functions whose Fourier transform decays very rapidly. Alternatively, functions in $\mathcal{F}_\sigma$ can be seen as convolutions of functions of $L^2(\mathbb{R}^p)$ with a Gaussian kernel $G_{\sigma/\sqrt{2}}(x) = \exp\left(-\frac{1}{\sigma^2}||x||^2\right)$. Note that, as $\sigma$ increases from 0 to $\infty$, the functions $G_{\sigma/\sqrt{2}}$ range from an impulse to a constant function, and the function spaces $\mathcal{F}_\sigma$ decrease from $L^2(\mathbb{R}^p)$ to $\varnothing$.

### 2.3 Independent component analysis

The independent component analysis (ICA) problem that we consider in this paper is based on the following statistical model:

$$y = Ax, \tag{6}$$

where $x$ is a latent random vector with $m$ independent components, $A$ is an $m \times m$ matrix of parameters, assumed invertible, and $y$ is an observed vector with $m$ components. Based

on a set of $N$ independent, identically distributed observations of the vector $y$, we wish to estimate the parameter matrix $A$.[1] From the estimate of $A$ we can estimate the values of $x$ corresponding to any observed $y$ by solving a linear system of equations. The distribution of $x$ is assumed unknown, and we do not care to estimate this distribution. Thus we formulate ICA as a semiparametric model (Bickel et al., 1998).

Our goal is to find a maximum likelihood estimate of $A$. Let us first consider the population version of ICA, in which $p^*(y)$ denotes the true distribution of $y$, and $p(y)$ denotes the model. We wish to minimize the Kullback-Leibler (KL) divergence between the distributions $p^*$ and $p$: $D(p^*(y) \,||\, p(y))$. Define $W = A^{-1}$, so that $x = Wy$. Since the KL divergence is invariant with respect to an invertible transformation, we can apply $W$ to $y$ in both arguments of the KL divergence, which implies our problem is equivalent to that of minimizing $D(p^*(x) \,||\, p(x))$.

Let $\tilde{p}(x)$ denote the joint probability distribution obtained by taking the product of the marginals of $p^*(x)$. We have the following decomposition of the KL divergence (see Cover and Thomas, 1991):

$$D(p^*(x) \,||\, p(x)) = D(p^*(x) \,||\, \tilde{p}(x)) + D(\tilde{p}(x) \,||\, p(x)),$$

for any distribution $p(x)$ with independent components. Consequently, for a given $A$, the minimum over all possible $p(x)$ is attained precisely at $p(x) = \tilde{p}(x)$, and the minimal value is $D(p^*(x) \,||\, \tilde{p}(x))$, which is exactly the mutual information between the components of $x = Wy$. Thus, the problem of maximizing the likelihood with respect to $W$ is equivalent to the problem of minimizing the mutual information between the components of $x = Wy$.

ICA can be viewed as a generalization of principal components analysis (PCA). While PCA yields uncorrelated components, and is based solely on second moments, ICA yields independent components, and is based on the mutual information, which is in general a function of higher-order moments. Clearly an ICA solution is also a PCA solution, but the converse is not true. In practice, ICA algorithms often take advantage of this relationship, treating PCA as a preprocessing phase. Thus one *whitens* the random variable $y$, multiplying $y$ by a matrix $P$ such that $\tilde{y} = Py$ has an identity covariance matrix ($P$ can be chosen as the inverse of the square root of the covariance matrix of $y$). There is a computational advantage to this approach: once the data are whitened, the matrix $W$ is necessarily orthogonal (Hyvärinen et al., 2001). This reduces the number of parameters to be estimated, and, as we discuss in Section 5, enables the use of efficient optimization techniques based on the Stiefel manifold of orthogonal matrices.

In practice we do not know $p^*(y)$ and thus the estimation criteria—mutual information or KL divergence—must be replaced with empirical estimates. While in principle one could form an empirical mutual information or empirical likelihood, which is subsequently optimized with respect to $W$, the more common approach to ICA is to work with approximations to the mutual information (Amari et al., 1996, Comon, 1994, Hyvärinen, 1999), or to use alternative contrast functions (Jutten and Herault, 1991). For example, by using Edgeworth or Gram-Charlier expansions one can develop an approximation of the mutual

1. The identifiability of the ICA model has been discussed by Comon (1994). Briefly, the matrix $A$ is identifiable, up to permutation and scaling of its columns, if and only if at most one of the component distributions $p(x_i)$ is Gaussian.

information in terms of skew and kurtosis. Forming an empirical estimate of the skew and kurtosis via the method of moments, one obtains a function of $W$ that can be optimized.

We propose two new ICA contrast functions in this paper. The first is based on the $\mathcal{F}$-correlation, which, as we briefly discussed in Section 1, can be obtained by computing the first canonical correlation in a reproducing kernel Hilbert space. The second is based on computing not only the first canonical correlation, but the entire CCA spectrum, a quantity known as the "generalized variance." We describe both of these contrast functions, and their relationship to the mutual information, in the following section.

## 3. Kernel independent component analysis

We refer to our general approach to ICA, based on the optimization of canonical correlations in a reproducing kernel Hilbert space, as KernelICA. In this section we describe two contrast functions that exemplify our general approach, and we present the resulting KernelICA algorithms.

### 3.1 The $\mathcal{F}$-correlation

We begin by studying the $\mathcal{F}$-correlation in more detail. We restrict ourselves to two random variables in this section and present the generalization to $m$ variables in Section 3.2.3.

**Theorem 1** *Let $x_1$ and $x_2$ be random variables in $\mathcal{X} = \mathbb{R}^p$. Let $K_1$ and $K_2$ be Mercer kernels with feature maps $\Phi_1$, $\Phi_2$ and feature spaces $\mathcal{F}_1$, $\mathcal{F}_2 \subset \mathbb{R}^{\mathcal{X}}$. Then the canonical correlation $\rho_{\mathcal{F}}$ between $\Phi_1(x_1)$ and $\Phi_2(x_2)$, which is defined as*

$$\rho_{\mathcal{F}} = \max_{(f_1,f_2)\in\mathcal{F}_1\times\mathcal{F}_2} \operatorname{corr}(\langle\Phi_1(x_1), f_1\rangle, \langle\Phi_2(x_2), f_2\rangle),$$

*is equal to*

$$\rho_{\mathcal{F}} = \max_{(f_1,f_2)\in\mathcal{F}_1\times\mathcal{F}_2} \operatorname{corr}(f_1(x_1), f_2(x_2)). \tag{7}$$

**Proof** This is immediate from the reproducing property (4). ■

The choice of kernels $K_1$ and $K_2$ specifies the sets $\mathcal{F}_1$ and $\mathcal{F}_2$ of functions that we use to characterize independence, via the correlation between $f_1(x_1)$ and $f_2(x_2)$. While in general we can use different kernels for $x_1$ and $x_2$, for notational simplicity we restrict ourselves in the remainder of the paper to cases in which the two kernels and the two feature spaces are equal, denoting them as $K$ and $\mathcal{F}$, respectively.

Note that the larger $\mathcal{F}$ is, the larger the value of the $\mathcal{F}$-correlation. In particular, for translation-invariant kernels, the $\mathcal{F}$-correlation increases as $\sigma$ decreases. But for any value of $\sigma$, the $\mathcal{F}$-correlation turns out to provide a sound basis for assessing independence, as the following theorem makes precise in the case of the univariate Gaussian kernel:

**Theorem 2 (Independence and $\mathcal{F}$-correlation)** *If $\mathcal{F}$ is the RKHS corresponding to a Gaussian kernel on $\mathcal{X} = \mathbb{R}$, $\rho_{\mathcal{F}} = 0$ if and only if the variables $y_1$ and $y_2$ are independent.*

**Proof** We mentioned earlier that the first implication is trivial. Let us now assume that $\rho_{\mathcal{F}} = 0$. Since $\mathcal{F}$ is a vector space, we have:

$$\rho_{\mathcal{F}} = \max_{(f_1,f_2)\in\mathcal{F}\times\mathcal{F}} |\mathrm{corr}(f_1(x_1), f_2(x_2))|,$$

which implies $\mathrm{cov}(f_1(x_1), f_2(x_2)) = 0$, or, equivalently, $E(f_1(x_1)f_2(x_2)) = E(f_1(x_1))E(f_2(x_2))$, for all $f_1, f_2 \in \mathcal{F}$. For any given $\omega_0 \in \mathbb{R}$ and $\tau > 0$, the function $x \mapsto e^{-x^2/2\tau^2}e^{i\omega_0 x}$ has a Fourier transform equal to $\sqrt{2\pi}\tau e^{-\tau^2(\omega-\omega_0)^2/2}$, and thus satisfies the condition in Eq. (5) as long as $\tau > \sigma/\sqrt{2}$. Consequently, if $\tau > \sigma/\sqrt{2}$, the function belongs to $\mathcal{F}$ and we have, for all real $\omega_1$ and $\omega_2$:

$$E\left(e^{i\omega_1 x_1 + i\omega_2 x_2} e^{-(x_1^2+x_2^2)/2\tau^2}\right) = E\left(e^{i\omega_1 x_1} e^{-x_1^2/2\tau^2}\right) E\left(e^{i\omega_2 x_2} e^{-x_2^2/2\tau^2}\right).$$

Letting $\tau$ tend to infinity, we find that for all $\omega_1$ and $\omega_2$:

$$E\left(e^{i\omega_1 x_1 + i\omega_2 x_2}\right) = E\left(e^{i\omega_1 x_1}\right) E\left(e^{i\omega_2 x_2}\right)$$

which implies that $x_1$ and $x_2$ are independent (Durrett, 1996). ■

Note that when the function space has finite dimension, Theorem 2 does not hold. In particular, for polynomial kernels (e.g., Schölkopf and Smola, 2001), $\mathcal{F}$ is a finite-dimensional space of polynomials, and thus the $\mathcal{F}$-correlation does not characterize independence. Nonetheless, as we show in Section 5.3, polynomial kernels can still be usefully employed in the ICA setting to provide heuristic initialization procedures for optimizing a contrast function.

### 3.2 Estimation of the $\mathcal{F}$-correlation

To employ the $\mathcal{F}$-correlation as a contrast function for ICA, we need to be able to compute canonical correlations in feature space. We also need to be able to optimize the canonical correlation, but for now our focus is simply that of computing the canonical correlations in an RKHS. (We discuss the optimization problem in Section 5).

In fact our goal is not solely computational, it is also statistical. While thus far we have defined $\mathcal{F}$-correlation in terms of a population expectation, we generally do not have access to the population but rather to a finite sample. Thus we need to develop an empirical *estimate* of the $\mathcal{F}$-correlation. Indeed, the problem of working in an RKHS and the issue of developing a statistical estimate of a functional defined in an RKHS are closely tied—by considering the empirical version of the $\mathcal{F}$-correlation we work in a finite-dimensional subspace of the RKHS, and can exploit its useful geometric properties while avoiding issues having to do with its (possible) infinite dimensionality.

Thus we will develop a "kernelized" version of canonical correlation, which involves two aspects: working with an empirical sample, and working in a feature space. We proceed by first presenting a naive "kernelization" of the population $\mathcal{F}$-correlation. For reasons that we will discuss, this naive kernelization does not provide a generally useful estimator, but it does serve as a guide for developing a *regularized* kernelization that does provide a useful estimator. It is this regularized, kernelized canonical correlation that provides the foundation for the algorithms that we present in the remainder of the paper.

### 3.2.1 Kernelization of CCA

In the case of two variables the goal is to maximize the correlation between projections of the data in the feature space. A direct implementation would simply map each data point to feature space and use CCA in the feature space. This is likely to be very inefficient computationally, however, if not impossible, and we would prefer to perform all of our calculations in the input space.[2]

Let $\{x_1^1, \ldots, x_1^N\}$ and $\{x_2^1, \ldots, x_2^N\}$ denote sets of $N$ empirical observations of $x_1$ and $x_2$, respectively, and let $\{\Phi(x_1^1), \ldots, \Phi(x_1^N)\}$ and $\{\Phi(x_2^1), \ldots, \Phi(x_2^N)\}$ denote the corresponding images in feature space. Suppose (momentarily) that the data are centered in feature space (i.e., $\sum_{k=1}^N \Phi(x_1^k) = \sum_{k=1}^N \Phi(x_2^k) = 0$). We let $\hat{\rho}_{\mathcal{F}}(x_1, x_2)$ denote the empirical canonical correlation; that is, the canonical correlation based not on population covariances but on empirical covariances. Since, as we shall see, $\hat{\rho}_{\mathcal{F}}(x_1, x_2)$ depends only on the Gram matrices $K_1$ and $K_2$ of these observations, we also use the notation $\hat{\rho}_{\mathcal{F}}(K_1, K_2)$ to denote this canonical correlation.

As in kernel PCA (Schölkopf et al., 1998), the key point to notice is that we only need to consider the subspace of $\mathcal{F}$ that contains the span of the data. For fixed $f_1$ and $f_2$, the empirical covariance of the projections in feature space can be written:

$$\widehat{\mathrm{cov}}(\langle \Phi(x_1), f_1 \rangle, \langle \Phi(x_2), f_2 \rangle) = \frac{1}{N} \sum_{k=1}^N \left\langle \Phi(x_1^k), f_1 \right\rangle \left\langle \Phi(x_2^k), f_2 \right\rangle. \tag{8}$$

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ represent the linear spaces spanned by the $\Phi$-images of the data points. Thus we can write $f_1 = \sum_{k=1}^N \alpha_1^k \Phi(x_1^k) + f_1^\perp$ and $f_2 = \sum_{k=1}^N \alpha_2^k \Phi(x_2^k) + f_2^\perp$, where $f_1^\perp$ and $f_2^\perp$ are orthogonal to $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. We have:

$$\begin{aligned}
\widehat{\mathrm{cov}}(\langle \Phi(x_1), f_1 \rangle, \langle \Phi(x_2), f_2 \rangle) &= \frac{1}{N} \sum_{k=1}^N \left\langle \Phi(x_1^k), \sum_{i=1}^N \alpha_1^i \Phi(x_1^i) \right\rangle \left\langle \Phi(x_2^k), \sum_{j=1}^N \alpha_2^j \Phi(x_2^j) \right\rangle \\
&= \frac{1}{N} \sum_{k=1}^N \sum_{i=1}^N \sum_{j=1}^N \alpha_1^i K_1(x_1^i, x_1^k) K_2(x_2^j, x_2^k) \alpha_2^j \\
&= \frac{1}{N} \alpha_1^\top K_1 K_2 \alpha_2,
\end{aligned}$$

where $K_1$ and $K_2$ are the Gram matrices associated with the data sets $\{x_1^i\}$ and $\{x_2^i\}$, respectively. We also obtain:

$$\widehat{\mathrm{var}}\left(\langle \Phi(x_1), f_1 \rangle\right) = \frac{1}{N} \alpha_1^\top K_1 K_1 \alpha_1$$

and

$$\widehat{\mathrm{var}}\left(\langle \Phi(x_2), f_2 \rangle\right) = \frac{1}{N} \alpha_2^\top K_2 K_2 \alpha_2.$$

2. Melzer et al. (2001) and Akaho (2001) have independently derived the kernelized CCA algorithm for two variables that we present in this section. A similar but not identical algorithm, also restricted to two variables, has been described by Fyfe and Lai (2000).

Putting these results together, our kernelized CCA problem becomes that of performing the following maximization:

$$\hat{\rho}_{\mathcal{F}}(K_1, K_2) = \max_{\alpha_1,\ \alpha_2 \in \mathbb{R}^N} \frac{\alpha_1^\top K_1 K_2 \alpha_2}{(\alpha_1^\top K_1^2 \alpha_1)^{1/2} (\alpha_2^\top K_2^2 \alpha_2)^{1/2}}. \tag{9}$$

But this is equivalent to performing CCA on two vectors of dimension $N$, with covariance matrix equal to $\begin{pmatrix} K_1^2 & K_1 K_2 \\ K_2 K_1 & K_2^2 \end{pmatrix}$. Thus we see that we can perform a kernelized version of CCA by solving the following generalized eigenvalue problem:

$$\begin{pmatrix} 0 & K_1 K_2 \\ K_2 K_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \rho \begin{pmatrix} K_1^2 & 0 \\ 0 & K_2^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \tag{10}$$

based on the Gram matrices $K_1$ and $K_2$.

If the points $\Phi(x_i^k)$ are not centered, then although it is impossible to actually center them in feature space, it is possible to find the Gram matrix of the centered data points (Schölkopf et al., 1998). That is, if $K$ is the $N \times N$ Gram matrix of the non-centered data points, then the Gram matrix $\widetilde{K}$ of the centered data points is $\widetilde{K} = N_0 K N_0$ where $N_0 = I - \frac{1}{N}\mathbf{1}$ is a constant singular matrix.[3] Whenever we use a Gram matrix, we assume that it has been centered in this way.

### 3.2.2 Regularization

Unfortunately, the kernelized CCA problem in Eq. (9) does not provide a useful estimate of the population canonical correlation in general. This can easily be seen by considering a geometric interpretation of Eq. (9). In particular, if we let $V_1$ and $V_2$ denote the subspaces of $\mathbb{R}^N$ generated by the columns of $K_1$ and $K_2$, respectively, then we can rewrite Eq. (9) as:

$$\hat{\rho}_{\mathcal{F}}(K_1, K_2) = \max_{v_1 \in V_1, v_2 \in V_2} \frac{v_1^\top v_2}{(v_1^\top v_1)^{1/2} (v_2^\top v_2)^{1/2}} = \max_{v_1 \in V_1, v_2 \in V_2} \cos(v_1, v_2),$$

which is exactly the cosine of the angle between the two subspaces $V_1$ and $V_2$ (Golub and Loan, 1996). From this interpretation, it is obvious that if the matrices $K_1$ and $K_2$ were invertible, then the subspaces $V_1$ and $V_2$ would be equal to $\mathbb{R}^N$ and thus the angle would always be equal to zero, whatever $K_1$ and $K_2$ are. The matrices $K_1$ and $K_2$ do not have full rank, because they are centered Gram matrices. However, centering is equivalent to projecting the column spaces $V_1$ and $V_2$ onto the subspace orthogonal to the vector composed of all ones; therefore, if the non-centered Gram matrices are invertible (which occurs, for example, if a Gaussian kernel is used, and the data points are distinct), the two column spaces are identical and the angle between them is still equal to zero, resulting in a canonical correlation estimate that is always equal to one.

Thus the naive kernelization in Eq. (9) does not provide a useful estimator for general kernels. It does, however, provide a guide to the design of a useful *regularized* estimator, as we now discuss. Our regularization will penalize the RKHS norms of $f_1$ and $f_2$, and thus

3. The matrix $\mathbf{1}$ is an $N \times N$ matrix composed of ones. Note that $\mathbf{1}^2 = N\mathbf{1}$.

provide control over the statistical properties of KernelICA.[4] In particular, we define the *regularized $\mathcal{F}$-correlation* $\rho_{\mathcal{F}}^{\kappa}$ as:

$$\rho_{\mathcal{F}}^{\kappa} = \max_{f_1, f_2 \in \mathcal{F}} \frac{\operatorname{cov}(f_1(x_1), f_2(x_2))}{(\operatorname{var} f_1(x_1) + \kappa ||f_1||_{\mathcal{F}}^2)^{1/2} (\operatorname{var} f_2(x_2) + \kappa ||f_2||_{\mathcal{F}}^2)^{1/2}}, \tag{11}$$

where $\kappa$ is a small positive constant. Note that the regularized $\mathcal{F}$-correlation inherits the independence characterization property of the $\mathcal{F}$-correlation. In order to estimate it from a finite sample, we expand $\operatorname{var} f_1(x_1) + \kappa ||f_1||_{\mathcal{F}}^2$ up to second order in $\kappa$, to obtain:

$$\operatorname{var} f_1(x_1) + \kappa ||f_1||_{\mathcal{F}}^2 = \frac{1}{N} \alpha_1^\top K_1^2 \alpha_1 + \kappa \alpha_1^\top K_1 \alpha_1 \approx \frac{1}{N} \alpha_1^\top (K_1 + \frac{N\kappa}{2} I)^2 \alpha_1.$$

Thus the regularized kernel CCA problem becomes:

$$\hat{\rho}_{\mathcal{F}}^{\kappa}(K_1, K_2) = \max_{\alpha_1,\, \alpha_2 \in \mathbb{R}^N} \frac{\alpha_1^\top K_1 K_2 \alpha_2}{(\alpha_1^\top (K_1 + \frac{N\kappa}{2} I)^2 \alpha_1)^{1/2} (\alpha_2^\top (K_2 + \frac{N\kappa}{2} I)^2 \alpha_2)^{1/2}}, \tag{12}$$

with its equivalent formulation as a generalized eigenvalue problem as follows:

$$\begin{pmatrix} 0 & K_1 K_2 \\ K_2 K_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \rho \begin{pmatrix} (K_1 + \frac{N\kappa}{2} I)^2 & 0 \\ 0 & (K_2 + \frac{N\kappa}{2} I)^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}. \tag{13}$$

As we shall see in Section 4.3, the formulation of the regularized KCCA problem in Eq. (13) can be reduced to a (simple) eigenvalue problem that is well-behaved computationally. In addition, the regularized first canonical correlation has an important statistical property that the unregularized canonical correlation does not have—it turns out to be a *consistent* estimator of the regularized $\mathcal{F}$-correlation. (That is, when the number of samples $N$ tends to infinity, the estimate converges in probability to the population quantity. The proof of this result is beyond the scope of the paper).

### 3.2.3 Generalizing to more than two variables

The generalization of regularized kernelized canonical correlation to more than two sets of variables is straightforward, given our generalization of CCA to more than two sets of variables. We simply denote by $\mathcal{K}_\kappa$ the $mN \times mN$ matrix whose blocks are $(\mathcal{K}_\kappa)_{ij} = K_i K_j$, for $i \neq j$, and $(\mathcal{K}_\kappa)_{ii} = (K_i + \frac{N\kappa}{2} I)^2$, and we let $\mathcal{D}_\kappa$ denote the $mN \times mN$ block-diagonal

4. Intuitively, without any restriction on $||f_1||_{\mathcal{F}}$ and $||f_2||_{\mathcal{F}}$, it is possible to separate one data point $x_1^{k_0}$ from the other points $\{x_1^k\}$ with a function $f_1$, while separating $x_2^{k_0}$ from the other $\{x_2^k\}$ with a function $f_2$. For those $f_1$ and $f_2$, we get a correlation equal to one and thus obtain no information about the dependence of $x_1$ and $x_2$.

matrix with blocks $(K_i + \frac{N\kappa}{2} I)^2$. We obtain the following generalized eigenvalue problem:

$$\begin{pmatrix} (K_1 + \frac{N\kappa}{2} I)^2 & K_1 K_2 & \cdots & K_1 K_m \\ K_2 K_1 & (K_2 + \frac{N\kappa}{2} I)^2 & \cdots & K_2 K_m \\ \vdots & \vdots & & \vdots \\ K_m K_1 & K_m K_2 & \cdots & (K_m + \frac{N\kappa}{2} I)^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}$$
$$= \lambda \begin{pmatrix} (K_1 + \frac{N\kappa}{2} I)^2 & 0 & \cdots & 0 \\ 0 & (K_2 + \frac{N\kappa}{2} I)^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & (K_m + \frac{N\kappa}{2} I)^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}, \quad (14)$$

or $\mathcal{K}_\kappa \alpha = \lambda \mathcal{D}_\kappa \alpha$ for short. The minimal eigenvalue of this problem will be denoted $\hat{\lambda}^\kappa_{\mathcal{F}}(K_1, \ldots, K_m)$ and referred to as the *first kernel canonical correlation*. We also extend our earlier terminology and refer to this eigenvalue as an (empirical) $\mathcal{F}$-correlation.

Note that in the two-variable case we defined a function $\rho_{\mathcal{F}}(x_1, x_2)$ that depends on the covariances of the random variables $\Phi(x_1)$ and $\Phi(x_2)$, and we obtained an empirical contrast function $\hat{\rho}^\kappa_{\mathcal{F}}(x_1, x_2)$ from $\rho_{\mathcal{F}}(x_1, x_2)$ by substituting empirical covariances for population covariances and introducing regularization.[5] In the $m$-variable case, we have (thus far) defined only the empirical function $\hat{\lambda}^\kappa_{\mathcal{F}}(K_1, \ldots, K_m)$. In Appendix A.3, we study the properties of the population version of this quantity, $\lambda_{\mathcal{F}}(x_1, \ldots, x_m)$, by relating $\lambda_{\mathcal{F}}(x_1, \ldots, x_m)$ to a generalized notion of "correlation" among $m$ variables. By using this definition, we show that $\lambda_{\mathcal{F}}(x_1, \ldots, x_m)$ is always between zero and one, and is equal to one if and only if the variables $x_1, \ldots, x_m$ are pairwise independent. Thus we obtain an analog of Theorem 2 for the $m$-variable case.

For reasons that will become clear in Section 3.4, where we discuss a relationship between canonical correlations and mutual information, it is convenient to define our contrast functions in terms of the negative logarithm of canonical correlations. Thus we define a contrast function $I_{\lambda_{\mathcal{F}}}(x_1, \ldots, x_m) = -\frac{1}{2} \log \lambda_{\mathcal{F}}(x_1, \ldots, x_m)$ and ask to minimize this function. The result alluded to in the preceding paragraph shows that this quantity is nonnegative, and equal to zero if and only if the variables $x_1, \ldots, x_m$ are pairwise independent.

For the empirical contrast function, we will use the notation $\hat{I}_{\lambda_{\mathcal{F}}}(K_1, \ldots, K_m) = -\frac{1}{2} \log \hat{\lambda}^\kappa_{\mathcal{F}}(K_1, \ldots, K_m)$, emphasizing the fact that this contrast function depends on the data only through the Gram matrices.

### 3.2.4 Relationship to ACE

Given a response variable $y$ and predictor variables $x_1, \ldots, x_p$, the alternating conditional expectation (ACE) algorithm (Breiman and Friedman, 1985) minimizes

$$e = \frac{E[\theta(y) - \sum_{j=1}^p \phi_j(x_j)]^2}{\operatorname{var} \theta(y)}$$

5. In fact the word "contrast function" is generally reserved for a quantity that depends only on data and parameters, and is to be extremized in order to obtain parameter estimates. Thus $\hat{\rho}^\kappa_{\mathcal{F}}(x_1, x_2)$ is a contrast function. By also refering to the population version $\rho_{\mathcal{F}}(x_1, x_2)$ as a "contrast function," we are abusing terminology. But this is a standard abuse in the ICA literature, where, for example, the mutual information is viewed as a "contrast function."

Algorithm KernelICA-KCCA

**Input:** Data vectors $y^1, y^2, \ldots, y^N$
Kernel $K(x, y)$

1. Whiten the data
2. Minimize (with respect to $W$) the contrast function $C(W)$ defined as:
   a. Compute the centered Gram matrices $K_1, K_2, \ldots, K_m$ of the estimated sources $\{x^1, x^2, \ldots, x^N\}$, where $x^i = Wy^i$
   b. Define $\hat{\lambda}^{\kappa}_{\mathcal{F}}(K_1, \ldots, K_m)$ as the minimal eigenvalue of the generalized eigenvector equation $\mathcal{K}_\kappa \alpha = \lambda \mathcal{D}_\kappa \alpha$
   c. Define $C(W) = \hat{I}_{\lambda_{\mathcal{F}}}(K_1, \ldots, K_m) = -\frac{1}{2} \log \hat{\lambda}^{\kappa}_{\mathcal{F}}(K_1, \ldots, K_m)$

**Output:** $W$

Figure 1: A high-level description of the KernelICA-KCCA algorithm for estimating the parameter matrix $W$ in the ICA model.

with respect to the real-valued functions $\theta$, $\phi_1, \ldots, \phi_p$. The transformations obtained produce the best-fitting additive models. For the bivariate case, $p = 1$, minimization of $e$ can be shown to be equivalent to maximization of the correlation $\mathrm{corr}(\theta(y), \phi(x))$. It is thus equivalent to the $\mathcal{F}$-correlation problem. In the population case, the optimization can be done in $L^2(\mathbb{R})$, while in presence of a finite sample, the function spaces that are considered by Breiman and Friedman (1985) are smooth function spaces similar to the RKHS we use in this paper. A strong link exists between kernel CCA and ACE in this case: one step of the ACE algorithm is equivalent to one step of a power method algorithm for computing the largest generalized eigenvalue in Eq. (10) (Buja, 1990, Hastie and Tibshirani, 1990).[6]

### 3.3 The KernelICA-KCCA algorithm

Let us now apply the machinery that we have developed to the ICA problem. Given a set of data vectors $y^1, y^2, \ldots, y^N$, and given a parameter matrix $W$, we set $x^i = Wy^i$, for each $i$, and thereby form a set of estimated source vectors $\{x^1, x^2, \ldots, x^N\}$. The $m$ components of these vectors yield a set of $m$ Gram matrices, $K_1, K_2, \ldots, K_m$, and these Gram matrices (which depend on $W$) define the contrast function $C(W) = \hat{I}_{\lambda_{\mathcal{F}}}(K_1, \ldots, K_m)$. We obtain an ICA algorithm by minimizing this function with respect to $W$.

A high-level description of the resulting algorithm, which we refer to as KernelICA-KCCA, is provided in Figure 1.

6. We note in passing that the implementation of kernel CCA using low-rank approximations of Gram matrices that we present in Section 4.4 transfers readily to the general setting of *generalized additive models* based on kernel smoothers (Hastie and Tibshirani, 1990), thus enabling a fast implementation of the fitting procedure for such models.

Note that KernelICA-KCCA is not simply a "kernelization" of an extant ICA algorithm. Instead, we use kernel ideas to characterize independence and thus to define a new contrast function for ICA. As with alternative approaches to ICA, we minimize this contrast function with respect to the demixing matrix $W$.

We still have a significant amount of work to do to turn the high-level description in Figure 1 into a practical algorithm. The numerical linear algebra and optimization procedures that complete our description of the algorithm are presented in Sections 4 and 5. Before turning to those details, however, we turn to the presentation of an alternative contrast function based on generalized variance.

### 3.4 Kernel generalized variance

As we have discussed, the mutual information provides a natural contrast function for ICA, because of its property of being equal to zero if and only if the components are independent, and because of the link to the semiparametric likelihood. As we show in this section, there is a natural generalization of the $\mathcal{F}$-correlation that has a close relationship to the mutual information. We develop this generalization in this section, and use it to define a second ICA contrast function.

Our generalization is inspired by an interesting link that exists between canonical correlations and mutual information in the case of Gaussian variables. As we show in Appendix A, for jointly-Gaussian variables $x_1$ and $x_2$, the mutual information, $I(x_1, x_2)$, can be written as follows:

$$I(x_1, x_2) = -\frac{1}{2}\sum_{i=1}^{p} \log(1 - \rho_i^2),$$

where $\rho_i$ are the canonical correlations. Thus CCA can be used to compute the mutual information between a pair of Gaussian variables. Moreover, Appendix A also shows that this link can be extended to the mutual information between $m$ variables. Thus, the $m$-way mutual information between $m$ Gaussian random variables, $I(x_1, x_2, \ldots, x_m)$, can be obtained from the set of eigenvalues obtained from the generalized eigenvector problem $C\xi = \lambda D\xi$ that we defined in Section 2.1.1. In particular, Appendix A shows the following:

$$I(x_1, x_2, \ldots, x_m) = -\frac{1}{2}\log\frac{\det C}{\det D} = -\frac{1}{2}\sum_{i=1}^{P}\log\lambda_i, \tag{15}$$

where $\lambda_i$ are the generalized eigenvalues of $C\xi = \lambda D\xi$.

This result suggests that it may be worth considering a contrast function based on more than the first canonical correlation, and holds open the possibility that such a contrast function, if based on the nonlinearities provided by an RKHS, might provide an approximation to the mutual information between non-Gaussian variables.

Let us define the *generalized variance* associated with the generalized eigenvector problem $C\xi = \lambda D\xi$ as the ratio $(\det C)/(\det D)$. The result in Eq. (15) shows that for Gaussian variables the mutual information is equal to minus one-half the logarithm of the generalized variance.

We make an analogous definition in the kernelized CCA problem, defining the *kernel generalized variance* to be the product of the eigenvalues of the generalized eigenvector

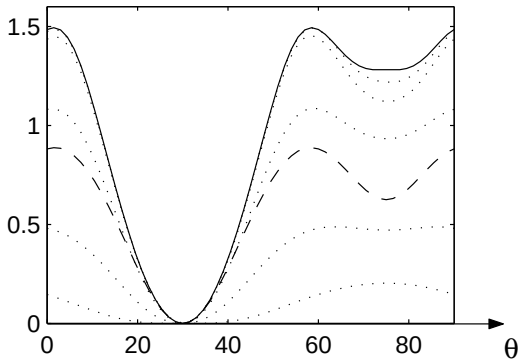


Figure 2: The mutual information $I(x_1, x_2)$ (dashed), the approximation $I_{\delta_{\mathcal{F}}}(x_1, x_2)$ for $\sigma = .25, .5, 1, 2, 4$ (dotted), and the limit $J(x_1, x_2)$ as $\sigma$ tends to zero (solid). The abscissa is the angle of the first independent component in a two-source ICA problem. As $\sigma$ decreases, $I_{\delta_{\mathcal{F}}}$ increases towards $J$. See the text for details.

problem in Eq. (14), or equivalently the ratio of determinants of the matrices in this problem. That is, given the generalized eigenvector problem $\mathcal{K}_\kappa \alpha = \lambda \mathcal{D}_\kappa \alpha$ of Eq. (14), we define:

$$\hat{\delta}_{\mathcal{F}}^{\kappa}(K_1, \ldots, K_m) = \frac{\det \mathcal{K}_\kappa}{\det \mathcal{D}_\kappa}$$

as the kernel generalized variance. We also define a contrast function $\hat{I}_{\delta_{\mathcal{F}}}(K_1, \ldots, K_m)$:

$$\hat{I}_{\delta_{\mathcal{F}}}(K_1, \ldots, K_m) = -\frac{1}{2} \log \hat{\delta}_{\mathcal{F}}^{\kappa}(K_1, \ldots, K_m),$$

by analogy with the mutual information for the Gaussian case.

Although we have proceeded by analogy with the Gaussian case, which is of little interest in the ICA setting, it turns out that $\hat{I}_{\delta_{\mathcal{F}}}(K_1, \ldots, K_m)$ has as its population counterpart a function $I_{\delta_{\mathcal{F}}}(x_1, \ldots, x_m)$ that is actually closely related to the mutual information between the original non-Gaussian variables in the input space. The proof of this result is sketched in Appendix B for the Gaussian kernel. In particular, in the case of two variables ($m = 2$), we show that, as $\sigma$ tends to zero, $I_{\delta_{\mathcal{F}}}(x_1, \ldots, x_m)$ tends to a limit $J(x_1, \ldots, x_m)$ that is equal to the mutual information up to second order, when we expand the mutual information around distributions that factorize.

Our result is illustrated in Figure 2. We compute the mutual information for a whitened ICA problem with two known sources and two components, as the angle $\theta$ of the estimated first component ranges from 0 to 90 degrees, with the independent component occurring at 30 degrees. The graph plots the true mutual information $I(x_1, x_2)$, the approximation $I_{\delta_{\mathcal{F}}}(x_1, x_2)$, for various values of $\sigma$, and the limit $J(x_1, x_2)$. The close match of the shape of $J(x_1, x_2)$ and the mutual information is noteworthy.

Algorithm KernelICA-KGV

**Input:** Data vectors $y^1, y^2, \ldots, y^N$
Kernel $K(x, y)$

1. Whiten the data
2. Minimize (with respect to $W$) the contrast function $C(W)$ defined as:
   a. Compute the centered Gram matrices $K_1, K_2, \ldots, K_m$ of the estimated sources $\{x^1, x^2, \ldots, x^N\}$, where $x^i = Wy^i$
   b. Define $\hat{\delta}_{\mathcal{F}}^{\kappa}(K_1, \ldots, K_m) = \det \mathcal{K}_\kappa / \det \mathcal{D}_\kappa$
   c. Define $C(W) = \hat{I}_{\delta_{\mathcal{F}}}(K_1, \ldots, K_m) = -\frac{1}{2} \log \hat{\delta}_{\mathcal{F}}^{\kappa}(K_1, \ldots, K_m)$

**Output:** $W$

Figure 3: A high-level description of the KernelICA-KGV algorithm for estimating the parameter matrix $W$ in the ICA model.

### 3.5 The KernelICA-KGV algorithm

In the previous section, we defined an alternative contrast function, $\hat{I}_{\delta_{\mathcal{F}}}(x_1, \ldots, x_m)$, in terms of the generalized variance associated with the generalized eigenvector problem $\mathcal{K}_\kappa \alpha = \lambda \mathcal{D}_\kappa \alpha$. Essentially, instead of computing only the first eigenvalue of this problem, as in the case of the $\mathcal{F}$-correlation contrast function, we compute the entire spectrum. As we show in Section 6, this does not increase the practical running time complexity of the algorithm. Based on this contrast function, we define a second KernelICA algorithm, the KernelICA-KGV algorithm outlined in Figure 3.

In summary, we have defined two KernelICA algorithms, both based on contrast functions defined in terms of the eigenvalues of the generalized eigenvector problem $\mathcal{K}_\kappa \alpha = \lambda \mathcal{D}_\kappa \alpha$. We now turn to a discussion of the computational methods by which we evaluate and optimize these contrast functions.

## 4. Computational issues

The algorithms that we have presented involve finding generalized eigenvalues of matrices of dimension $mN \times mN$, where $N$ is the number of data points and $m$ the number of sources. A naive implementation of these algorithms would therefore scale as $O(m^3 N^3)$, a computational complexity whose cubic growth in the number of data points would be a serious liability in applications to large data sets. As noted by several researchers, however, the spectrum of Gram matrices tends to show rapid decay, and low-rank approximations of Gram matrices can therefore often provide sufficient fidelity for the needs of kernel-based algorithms (Smola and Schölkopf, 2000, Williams and Seeger, 2001). Indeed, building on these observations, we describe an implementation of KernelICA whose computational complexity is linear in the number of data points.

We have two goals in this section. The first is to overview theoretical results that support the use of low-rank approximations to Gram matrices. Our presentation of these results will be concise, with a detailed discussion deferred to Appendix C. Second, we present a KernelICA implementation based on low-rank approximations obtained from incomplete Cholesky decomposition. We show both how to compute the KernelICA contrast functions, and how to compute derivatives of the contrast functions.

### 4.1 Theory

In Appendix C, we present theoretical results that show that in order to achieve a given required precision $\eta$, the rank $M$ of an approximation to a Gram matrix $K$ can be chosen as $M = h(N/\eta)$, where $h(t)$ is a function that depends on the underlying distribution $p(x)$ of the data. Moreover, the growth of $h(t)$ as $t$ tends to infinity depends only on the decay of $p(x)$ as $|x|$ tends to infinity. In particular, in the univariate case with a Gaussian kernel, when this decay is exponential (Gaussian-like), we have $h(t) = O(\log t)$. When the decay is polynomial, as $x^{-d}$, then $h(t) = O(t^{1/d+\varepsilon})$, for arbitrary $\varepsilon > 0$.

These results imply that if we require a constant precision $\eta$, it suffices to find an approximation of rank $M = O(\log N)$, for exponentially-decaying input distributions, and rank $M = O(N^{1/d+\varepsilon})$ for polynomially-decaying input distributions. These results are applicable to any method based on Gram matrices over input spaces of small dimension, and thus we can expect that kernel algorithms should generally be able to achieve a substantial reduction in complexity via approximations whose rank grows slowly with respect to $N$.

We will show separately in Section 4.3, however, that in the context of $\mathcal{F}$-correlation and the KGV, the precision $\eta$ can be taken to be linear in $N$. This implies that the rank of the approximation can be taken to be bounded by a constant in the ICA setting, and provides an even stronger motivation for basing an implementation of KernelICA on low-rank approximations.

### 4.2 Incomplete Cholesky decomposition

We aim to find low-rank approximations of Gram matrices of rank $M \ll N$. Note that even calculating a full Gram matrix is to be avoided because it is already an $O(N^2)$ operation. Fortunately, the fact that Gram matrices are positive semidefinite is a rather strong constraint, allowing approximations to Gram matrices to be found in $O(M^2N)$ operations. Following Fine and Scheinberg (2001) and Cristianini et al. (2002), the particular tool that we employ here is the incomplete Cholesky decomposition, commonly used in implementations of interior point methods for linear programming (Wright, 1999). Alternatives to incomplete Cholesky decomposition are provided by methods based on the Nyström approximation (Smola and Schölkopf, 2000, Williams and Seeger, 2001).

A positive semidefinite matrix $K$ can always be factored as $GG^\top$, where $G$ is an $N \times N$ matrix. This factorization can be found via Cholesky decomposition (which is essentially a variant of Gaussian elimination). Our goal, however, is to find a matrix $\widetilde{G}$ of size $N \times M$, for small $M$, such that the difference $K - \widetilde{G}\widetilde{G}^\top$ has norm less than a given value $\eta$. This can be achieved via incomplete Cholesky decomposition.

Incomplete Cholesky decomposition differs from standard Cholesky decomposition in that all pivots that are below a certain threshold are simply skipped. If $M$ is the number

of non-skipped pivots, then we obtain a lower triangular matrix $\widetilde{G}$ with only $M$ nonzero columns. Symmetric permutations of rows and columns are necessary during the factorization if we require the rank to be as small as possible (Golub and Loan, 1996). In that case, the stopping criterion involves the sum of remaining pivots.

An algorithm for incomplete Cholesky decomposition is presented in Figure 4. The algorithm involves picking one column of $K$ at a time, choosing the column to be added by greedily maximizing a lower bound on the reduction in the error of the approximation. After $l$ steps, we have an approximation of the form $\widetilde{K}_l = \widetilde{G}_l \widetilde{G}_l^\top$, where $G_l$ is $N \times l$. The ranking of the $N-l$ vectors that might be added in the following step is done by comparing the diagonal elements of the remainder matrix $K - \widetilde{G}_l \widetilde{G}_l^\top$. Each of these elements requires $O(l)$ operations to compute. Moreover, the update of $\widetilde{G}_l$ has a cost of $O(lN)$, so that the overall complexity is $O(M^2 N)$.

The incomplete Cholesky method has many attractive features. Not only is its time complexity $O(M^2 N)$, but also the only elements of $K$ that are needed in memory are the diagonal elements (which are equal to one for Gaussian kernels[7]). Most of the other elements are never used and those that are needed can be computed on demand. The storage requirement is thus $O(MN)$. Also, the number $M$ can be chosen online such that the approximation is as tight as desired.[8]

### 4.3 Solving Kernel CCA

Before turning to a discussion of how to use incomplete Cholesky decomposition to compute the eigenstructure needed for our ICA contrast functions, we discuss the generalized eigenvector problem of Eq. (14), $\mathcal{K}_\kappa \alpha = \lambda \mathcal{D}_\kappa \alpha$, in more detail.

Owing to the regularization, we can apply a classical method for solving such a problem by finding a matrix $\mathcal{C}$ such that $\mathcal{D}_\kappa = \mathcal{C}^\top \mathcal{C}$, defining $\beta = \mathcal{C}\alpha$, and thereby transforming the problem into a standard eigenvector problem $\mathcal{C}^{-\top} \mathcal{K}_\kappa \mathcal{C}^{-1} \beta = \lambda \beta$. Our kernelized CCA problem thus reduces to finding the smallest eigenvalue of the matrix:

$$\widetilde{\mathcal{K}}_\kappa = \mathcal{D}_\kappa^{-1/2} \mathcal{K}_\kappa \mathcal{D}_\kappa^{-1/2} = \begin{pmatrix} I & r_\kappa(K_1) r_\kappa(K_2) & \cdots & r_\kappa(K_1) r_\kappa(K_m) \\ r_\kappa(K_2) r_\kappa(K_1) & I & \cdots & r_\kappa(K_2) r_\kappa(K_m) \\ \vdots & \vdots & & \vdots \\ r_\kappa(K_m) r_\kappa(K_1) & r_\kappa(K_m) r_\kappa(K_2) & \cdots & I \end{pmatrix} \tag{16}$$

where $r_\kappa(K_i) = K_i (K_i + \frac{N\kappa}{2} I)^{-1} = (K_i + \frac{N\kappa}{2} I)^{-1} K_i$. If we have an eigenvector $\tilde{\alpha}$ of $\widetilde{\mathcal{K}}_\kappa$, then we have a generalized eigenvector defined by $\alpha_i = (K_i + \frac{N\kappa}{2} I)^{-1} \tilde{\alpha}_i$, with the same eigenvalue. In the case of the KGV problem, we need to compute $\det \widetilde{\mathcal{K}}_\kappa$.

Our regularization scheme has the effect of shrinking each eigenvalue of $K_i$ towards zero or one, via the function $\lambda \mapsto \lambda/(\lambda + \frac{N\kappa}{2})$. Consequently, all eigenvalues less than a small

7. Centering, which would make the diagonal elements different from one and make the other elements harder to compute, can be done easily after the Cholesky decomposition.
8. Note that no theoretical bound is available concerning the relation between $M$ and the optimal rank $M$ for a given precision. In our empirical work, however, we always obtained a rank very close to the optimal one. We believe this is due to the fact that our Gram matrices have a spectrum that decays very rapidly. Indeed, as pointed out by Wright (1999), a significant eigengap ensures that incomplete Cholesky has small numerical error and yields a good approximation.

Algorithm IncompleteCholesky

**Input:** $N \times N$ semidefinite positive matrix $K$
precision parameter $\eta$

1. Initialization: $i = 1$, $K' = K$, $P = I$, for $j \in [1, N]$, $G_{jj} = K_{jj}$
2. While $\sum_{j=i}^{N} G_{jj} > \eta$
   - Find best new element: $j^* = \arg\max_{j \in [i,N]} G_{jj}$
   - Update permutation $P$:
     set $P_{ii} = 0$, $P_{j^*j^*} = 0$ and $P_{ij^*} = 1$, $P_{j^*i} = 1$
   - Permute elements $i$ and $j^*$ in $K'$:
     column $K'_{1:N,i} \leftrightarrow K'_{1:N,j^*}$
     row $K'_{i,1:N} \leftrightarrow K'_{j^*,1:N}$
   - Update (due to new permutation) the already calculated elements of $G$: $G_{i,1:i} \leftrightarrow G_{j^*,1:i}$
   - Set $G_{ii} = \sqrt{K'_{ii}}$
   - Calculate $i^{th}$ column of $G$:
     $G_{i+1:n,i} = \frac{1}{G_{ii}} \left( K'_{i+1:N,i} - \sum_{j=1}^{i-1} G_{i+1:N,j} G_{ij} \right)$
   - Update only diagonal elements:
     for $j \in [i+1, N]$, $G_{jj} = K_{jj} - \sum_{k=1}^{i} G_{jk}^2$
   - $i \leftarrow i + 1$
3. Output $P$, $G$ and $M = i - 1$

**Output:** an $N \times M$ lower triangular matrix $G$ and a permutation matrix $P$ such that $||PKP^\top - GG^\top|| \leqslant \eta$

Figure 4: An algorithm for incomplete Cholesky decomposition. The notation $G_{a:b,c:d}$ refers to the matrix extracted from $G$ by taking the rows $a$ to $b$ and columns $c$ to $d$.

fraction of $\frac{N\kappa}{2}$ (we use the fraction $10^{-3}$ in our simulations) will numerically be discarded. This implies that in our search for low-rank approximations, we need only keep eigenvalues greater than $\eta = 10^{-3}\frac{N\kappa}{2}$. As detailed in Appendix C, this has the numerical effect of making our Gram matrices of constant numerical rank as $N$ increases.

### 4.4 Algorithms for KCCA and KGV

We now show how to use incomplete Cholesky decomposition to solve the KCCA and KGV problems. As we have seen, these problems reduce to eigenvalue computations involving the regularized matrix $\widetilde{\mathcal{K}}_\kappa$ in Eq. (16).

Using the incomplete Cholesky decomposition, for each matrix $K_i$ we obtain the factorization $K_i \approx G_i G_i^\top$, where $G_i$ is an $N \times M_i$ matrix with rank $M_i$, where $M_i \ll N$. We perform a singular value decomposition of $G_i$, in time $O(M_i^2 N)$, to obtain an $N \times M_i$ matrix $U_i$ with orthogonal columns (i.e., such that $U_i^\top U_i = I$), and an $M_i \times M_i$ diagonal matrix $\Lambda_i$ such that:

$$K_i \approx G_i G_i^\top = U_i \Lambda_i U_i^\top.$$

Let $M = \frac{1}{m}\sum_{i=1}^m M_i$ denote the average value of the ranks $M_i$.

In order to study how to use these matrices to perform our calculations, let $V_i$ denote the orthogonal complement of $U_i$, such that $(U_i\ V_i)$ is an $N \times N$ orthogonal matrix. We have:

$$K_i \approx U_i \Lambda_i U_i^\top = (U_i\ V_i) \begin{pmatrix} \Lambda_i & 0 \\ 0 & 0 \end{pmatrix} (U_i\ V_i)^\top$$

If we now consider the regularized matrices $r_\kappa(K_i)$, we have:

$$r_\kappa(K_i) = (K_i + \frac{N\kappa}{2} I)^{-1} K_i = (U_i\ V_i) \begin{pmatrix} R_i & 0 \\ 0 & 0 \end{pmatrix} (U_i\ V_i)^\top = U_i R_i U_i^\top,$$

where $R_i$ is the diagonal matrix obtained from the diagonal matrix $\Lambda_i$ by applying the function $\lambda \mapsto \frac{\lambda}{\lambda + N\kappa/2}$ to its elements. As seen before, this function softly thresholds the eigenvalues less than $\frac{N\kappa}{2}$. We now have the following decomposition:

$$\widetilde{\mathcal{K}}_\kappa = \mathcal{U}\mathcal{R}_\kappa\, \mathcal{U}^\top + \mathcal{V}\mathcal{V}^\top = (\mathcal{U}\ \mathcal{V}) \begin{pmatrix} \mathcal{R}_\kappa & 0 \\ 0 & I \end{pmatrix} (\mathcal{U}\ \mathcal{V})^\top,$$

where $\mathcal{U}$ is $mN \times mM$, $\mathcal{V}$ is $mN \times (mN - mM)$, $\mathcal{R}_\kappa$ is $mM \times mM$, and $(\mathcal{U}\ \mathcal{V})$ is orthogonal:

$$\mathcal{U} = \begin{pmatrix} U_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & U_m \end{pmatrix} \qquad \mathcal{V} = \begin{pmatrix} V_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & V_m \end{pmatrix}$$

$$\mathcal{R}_\kappa = \begin{pmatrix} I & R_1 U_1^\top U_2 R_2 & \cdots & R_1 U_1^\top U_m R_m \\ R_2 U_2^\top U_1 R_1 & I & \cdots & R_2 U_2^\top U_m R_m \\ \vdots & \vdots & & \vdots \\ R_m U_m^\top U_1 R_1 & R_m U_m^\top U_2 R_2 & \cdots & I \end{pmatrix}.$$

The $mN$ (nonnegative) eigenvalues of $\widetilde{\mathcal{K}}_\kappa$ sum to $\mathrm{tr}(\widetilde{\mathcal{K}}_\kappa) = mN$. If $\widetilde{\mathcal{K}}_\kappa \neq I$ then at least one of these eigenvalues must be less than 1. Consequently, since $\widetilde{\mathcal{K}}_\kappa$ is similar to $\begin{pmatrix} \mathcal{R}_\kappa & 0 \\ 0 & I \end{pmatrix}$, the smallest eigenvalue of $\widetilde{\mathcal{K}}_\kappa$ (with eigenvector $\alpha \in \mathbb{R}^{mN}$) is equal to the smallest eigenvalue of $\mathcal{R}_\kappa$ (with eigenvector $\beta \in \mathbb{R}^{mM}$), and the two eigenvectors are related through:

$$\alpha = \mathcal{U}\beta \;\Rightarrow\; \beta = \mathcal{U}^\top \alpha.$$

This allows us to compute the KCCA criterion. For the KGV criterion, we trivially have $\det \widetilde{\mathcal{K}}_\kappa = \det \begin{pmatrix} \mathcal{R}_\kappa & 0 \\ 0 & I \end{pmatrix} = \det \mathcal{R}_\kappa$.

We thus have reduced the size of our matrices from $mN \times mN$ to $mM \times mM$. Once we have borne the cost of such a low-rank decomposition, the further complexity is greatly reduced. In the case of the first canonical correlation (the smallest eigenvalue of $\widetilde{\mathcal{K}}_\kappa$) we simply need to find the smallest eigenvalue of $\mathcal{R}_\kappa$, which has a cost of $O(m^2M^2)$. In the case of the generalized variance, we need to compute $\det \widetilde{\mathcal{K}}_\kappa = \det \mathcal{R}_\kappa$, which costs $O(\xi(mM))$, where $\xi(s)$ is the complexity of multiplying two $s \times s$ matrices, which is less than $O(s^3)$ when using Strassen's algorithm (Cormen et al., 1990). Since in our situation we have $M = O(1)$, the complexities are reduced to $O(m^2)$ and $O(\xi(m))$.

### 4.5 Free parameters

The KernelICA algorithms have two free parameters: the regularization parameter $\kappa$ and the width $\sigma$ of the kernel (assuming identical Gaussian kernels for each source). In our experimental work we found that the KernelICA algorithms were reasonably robust to the settings of these parameters. Our choices were to set $\kappa = 2 \times 10^{-3}$, $\sigma = 1/2$ for large samples ($N > 1000$) and $\kappa = 2 \times 10^{-2}$, $\sigma = 1$ for smaller samples ($N \leqslant 1000$).

For finite $N$, a value of $\sigma$ that is overly small leads to diagonal Gram matrices and our criteria become trivial. On the other hand, for large $N$ the KGV approaches the mutual information as $\sigma$ tends to zero, and this suggests choosing $\sigma$ as small as possible. Still another consideration, however, is computational—for small $\sigma$ the spectra of the Gram matrices decay less rapidly and the computational complexity grows. This can be mitigated by an appropriate choice of $\kappa$; in particular, the algorithm could choose $\kappa$ so that the number of retained eigenvalues for each Gram matrix is held constant. It is also possible to use cross-validation to set $\kappa$ (Leurgans et al., 1993). Clearly, there are several tradeoffs at play here, and the development of theoretical guidelines for the choice of the parameters is deferred to future work.

### 4.6 Derivatives

Our approach to ICA involves optimizing a contrast function defined in terms of a set of $m$ Gram matrices, where $m$ is the number of components. These matrices are functions of the weight matrix $W$, and thus our contrast functions are defined on a manifold of dimension $m(m-1)/2$ (see Section 5). For small $m$ (less than $m = 8$ in our simulations), the optimization can be based on simple techniques such as naive first-difference approximations of derivatives, or optimization methods that require only function evaluations. Such techniques are not viable for large problems, however, and in general we must turn to

derivative-based optimization techniques, where it is required that derivatives are computed efficiently.

The derivatives of Gram matrices are not semidefinite matrices in general, and thus we cannot directly invoke the low-rank decomposition algorithms that we have discussed in previous sections. Fortunately, however, in the case of Gaussian kernels, it is possible to express these matrix derivatives as a difference between two low-rank positive semidefinite matrices, and we can apply the incomplete Cholesky decomposition to each of these matrices separately. The details of this computation are provided in Appendix D.

Another possibility to compute derivatives efficiently is to use first-difference approximations of derivatives, as mentioned earlier, but with the help of the particular structure of the problem. Indeed, a naive computation would require $m(m-1)/2$ computations of the objective function, and yield a complexity of $O(m^3N)$. As shown in Appendix D, we manage to reduce the complexity to $O(m^2N)$ by reducing the number of incomplete Cholesky decompositions to be performed.

## 5. Optimization

An ICA contrast function is ultimately a function of the parameter matrix $W$. Estimating ICA parameters and independent components means minimizing the contrast function with respect to $W$. As noted by Amari (1998), the fact that $W$ is an orthogonal matrix in the ICA problem (once the data are whitened) endows the parameter space with additional structure, and this structure can be exploited by optimization algorithms. The particular formalism that we pursue here is that of a *Stiefel manifold*. In this section, we first review the properties of this manifold. We then discuss convergence properties of our algorithm. Finally, we present two efficient techniques to tame local minima problems.

### 5.1 The Stiefel manifold

The set of all $m \times m$ matrices $W$ such that $W^\top W = I$ is an instance of a *Stiefel manifold* (Edelman et al., 1999). Our optimization problem is thus the minimization of a function $C(W)$ on the Stiefel manifold. The familiar optimization algorithms of Euclidean spaces—gradient descent, steepest descent and conjugate gradient—can all be performed on a Stiefel manifold. The basic underlying quantities needed for optimization are the following:

- The *gradient* of a function $C(W)$ is defined as

$$\nabla C = \frac{\partial C}{\partial W} - W \left( \frac{\partial C}{\partial W} \right)^\top W,$$

  where $\frac{\partial C}{\partial W}$ is the derivative of $C$ with respect to $W$; that is, an $m \times m$ matrix whose element $(i, j)$ is $\frac{\partial C}{\partial w_{ij}}$ .

- The *tangent space* is equal to the space of all matrices $H$ such that $W^\top H$ is skew-symmetric. It is of dimension $m(m-1)/2$ and equipped with the canonical metric $||H||_c = \frac{1}{2} \operatorname{tr}(H^\top H)$.

- The *geodesic* starting from $W$ in the direction $H$ (in the tangent space at $W$) is determined as $G_{W,H}(t) = W \exp(tW^\top H)$, where the matrix exponential can be calculated efficiently after having diagonalized (in the complex field) the skew-symmetric matrix $W^\top H$.

In the simulations that we report in Section 7, we used steepest descent, with line search along the geodesic. Note that in our case, the calculation of the gradient is more costly (from 5 to 10 times) than the evaluation of the function $C$. Consequently, conjugate gradient techniques are particularly appropriate, because they save on the number of computed derivatives by computing more values of the functions.

### 5.2 Convergence issues

Since our algorithm is simply steepest descent (with line search) on an almost-everywhere differentiable function $C(W)$, the algorithm converges to a local minimum of $C(W)$, for any starting point. However, the ICA contrast functions have multiple local minima, and restarts are generally necessary if we are to find the global optimum. Empirically, the number of restarts that were needed was found to be small when the number of samples is sufficiently large so as to make the problem well-defined.

We have also developed two initialization heuristics that have been found to be particularly useful in practice for large-scale problems. These heuristics find a matrix $W_0$ sufficiently close to the global optimum so that the direct optimization of $C(W)$ gives the desired optimum without the need for restarts. We describe these heuristics in the following two sections.

### 5.3 Polynomial kernels

In the case of the KernelICA-KCCA algorithm, the contrast function $C(W)$ that is minimized is itself obtained through a minimization of a functional over the feature space of functions $\mathcal{F}$, as seen in the case of two variables in Eq. (7). This space of function is defined through the kernel $K(x, y)$. Intuitively, the larger the space of functions $\mathcal{F}$ is, the less smooth the function $C(W)$ should be with respect to variations of $W$, since a larger $\mathcal{F}$ can more closely adapt to any $W$.

Thus, by using a finite-dimensional feature space, we might hope that the optimization function will be smoother and thus local minima will disappear. As we noted in Section 3.1, objective functions based on kernels with finite-dimensional feature spaces do not characterize independence as well as kernels with infinite-dimensional feature spaces, so we should expect inaccurate solutions. Nevertheless, we have found empirically that such solutions are often found in the basin of attraction of the global minima for the full KernelICA-KCCA or KernelICA-KGV algorithm. Thus we can use polynomial kernels to initialize the algorithm.

The classical polynomial kernel is of the form $K(x, y) = (r + sxy)^d$. Its RKHS $\mathcal{F}_d$ is the space of polynomials of degree less or equal to $d$ and the RKHS norm $||f||^2$ is a weighted sum of squares of the coefficients of the polynomials: $|| \sum_{k=0}^d a_k x^k ||^2_{\mathcal{F}_d} = \sum_{k=0}^d \alpha_k |a_k|^2$. When a polynomial is considered as function in $L^2(\mathbb{R})$, the $k$-th coefficient is equal to $\frac{1}{k!} d^k f/dx^k(0)$, so we can write for $f \in \mathcal{F}_d$, $||f||^2_{\mathcal{F}_d} = \sum_{k=0}^d \frac{\alpha_k}{(k!)^2} \left| d^k f/dx^k(0) \right|^2$. Although

attractive computationally, we see that the traditional polynomial kernel does not lead to a natural norm for its RKHS. Instead, in our simulations, we use the Hermite polynomial kernels (Vapnik, 1998), which correspond to a more natural norm in a space of polynomials: If we let $h_k(x)$ denote the $k$-th Hermite polynomials (Szegö, 1975), we define the following Hermite polynomial kernel of order $d$, $K(x,y) = \sum_{k=0}^{d} e^{-x^2/2\sigma^2} e^{-y^2/2\sigma^2} \frac{h_k(x/\sigma) h_k(y/\sigma)}{2^k k!}$. Using classical properties of orthogonal polynomials, this kernel can be computed as efficiently as the traditional polynomial kernel. The RKHS associated with this kernel is the space of functions that can be written as $e^{-x^2/2\sigma^2} P(x)$ where $P(x)$ is a polynomial of degree less or equal to $d$, and the norm in that space is the $L^2$ norm. In the simulations that we report in Section 7, when we use finite dimensional kernels as a first step in the global optimization, we use Hermite polynomial kernels with $\sigma = 1.5$ and $d = 3$.

### 5.4 One-unit contrast functions

*One-unit contrast functions*—objective functions similar to projection pursuit criteria that are designed to discover single components—have been usefully employed in the ICA setting (Hyvärinen et al., 2001). These functions involve optimization on the sphere (of dimension $m-1$) rather than the orthogonal group (of dimension $m(m-1)/2$); this helps tame problems of local minima. In the KernelICA setting, the KCCA or KGV contrast between one univariate component and its orthogonal subspace provides a natural one-unit contrast function. Two issues still have to be resolved to implement such an approach: the choice of the kernel on the component of dimension $m-1$, and the combination of components obtained from one-unit contrasts into a full ICA solution.

In this paper, we assume that we are given the same kernel $K(x,y)$ for all desired univariate components. To define a kernel on vectors of dimension $m-1$, two natural choices arise. The first is to take the product of the $m-1$ univariate kernels. This is equivalent to choosing the feature space of functions on $\mathbb{R}^{m-1}$ to be the tensor product of the $m-1$ (identical) feature spaces of functions on $\mathbb{R}$ (Vapnik, 1998). In the case of a Gaussian kernel, this amounts to using an isotropic Gaussian kernel $K(x,y) = e^{-||x-y||^2/2\sigma^2}$. This choice has the advantage that the kernel is invariant to $(m-1)$-dimensional orthogonal transformations, so that the contrast function is invariant with respect to the choice of the basis of the orthogonal subspace. However, the spectrum of the Gram matrix decays more slowly than for the one-dimensional Gaussian, making this method impractical for large $m$.

The other natural choice is to take the sum of the kernels; this is equivalent to letting each component of the $m-1$ vector have its own Gram matrix. Here, the decay of each of the $m-1$ Gram matrices is fast, so that the computation of the contrast function scales well with $m$. However, care must be taken in the choice of the representative of the orthogonal subspace: to obtain a valid function it is necessary to define a unique representative of the orthogonal subspace which depends continuously on the univariate component $w$. This is easily done by considering the image of the canonical basis $(e_1, \ldots, e_m)$ of $\mathbb{R}^m$ by the rotation that transforms the first vector $e_1$ of this basis into $w$, and reduces to identity on the orthogonal complement of the span of $\{w, e_1\}$.

To combine one-dimensional component into a full ICA solution we use a "deflation" technique similar to the one used by Hyvärinen and Oja (1997). We first find a $m$-dimensional component $w_1$, then project the data into the subspace (of dimension $m-1$)

orthogonal to $w_1$ and search for a component $w_2$ in this subspace. This process is iterated until $m-1$ components are found (which implies that the $m$-th is also found by orthogonality). A full ICA solution $W$ is then obtained by combining the $m$ one-dimensional components.

## 6. Computational complexity

Let $N$ denote the number of samples, and let $m$ denote the number of sources. $M$ is the maximal rank considered by our low-rank decomposition algorithms for the kernels. We assume that $m \leqslant N$.

- Performing PCA on the input variables is $O(m^2N)$—calculating the covariance matrix is $O(m^2N)$, diagonalizing the $m \times m$ matrix is $O(m^3) = O(m^2N)$, and scaling is $O(m^2N)$.

- KCCA using incomplete Cholesky decomposition is $O(m^2M^2N)$—calculating the decomposition $m$ times is $m \times O(NM^2)$, then forming the matrix $\mathcal{R}_\kappa$ is $\frac{m(m-1)}{2} \times O(M^2N) = O(m^2M^2N)$, and finding its smallest eigenvalue is $O((mM)^2)$. However, in practice, the incomplete Cholesky decompositions are the bottleneck of these computations, so that the empirical complexity is in fact $O(mM^2N)$.

- KGV using incomplete Cholesky decomposition is $O(m^2M^2N+m^3M^3)$, which is usually $O(m^2M^2N)$ because $N$ is generally greater than $mM$—calculating the decomposition $m$ times is $m \times O(M^2N)$, then forming the matrix $\mathcal{R}_\kappa$ is $\frac{m(m-1)}{2} \times O(M^2N) = O(m^2M^2N)$, and computing its determinant is $O((mM)^3)$. As for KCCA, the empirical complexity is $O(mM^2N)$.

- Computation of the derivatives is $O(m^2M^2N)$—at most $3m^2$ incomplete Cholesky decompositions to perform, and then matrix multiplications with lower complexity, for the direct approach, while $O(m^2)$ computations of the contrast functions are needed for the first-order difference approach.

## 7. Simulation results

We have conducted an extensive set of simulation experiments using data obtained from a variety of source distributions. The sources that we used (see Figure 5) included subgaussian and supergaussian distributions, as well as distributions that are nearly Gaussian. We studied unimodal, multimodal, symmetric, and nonsymmetric distributions. All distributions are scaled to have zero mean and unit variance.

We also varied the number of components, from 2 to 16, the number of training samples, from 250 to 4000, and studied the robustness of the algorithms to varying numbers of outliers.

Comparisons were made with three existing ICA algorithms: the FastICA algorithm (Hyvärinen and Oja, 1997), the JADE algorithm (Cardoso, 1999), and the extended Infomax algorithm (Lee et al., 1999).

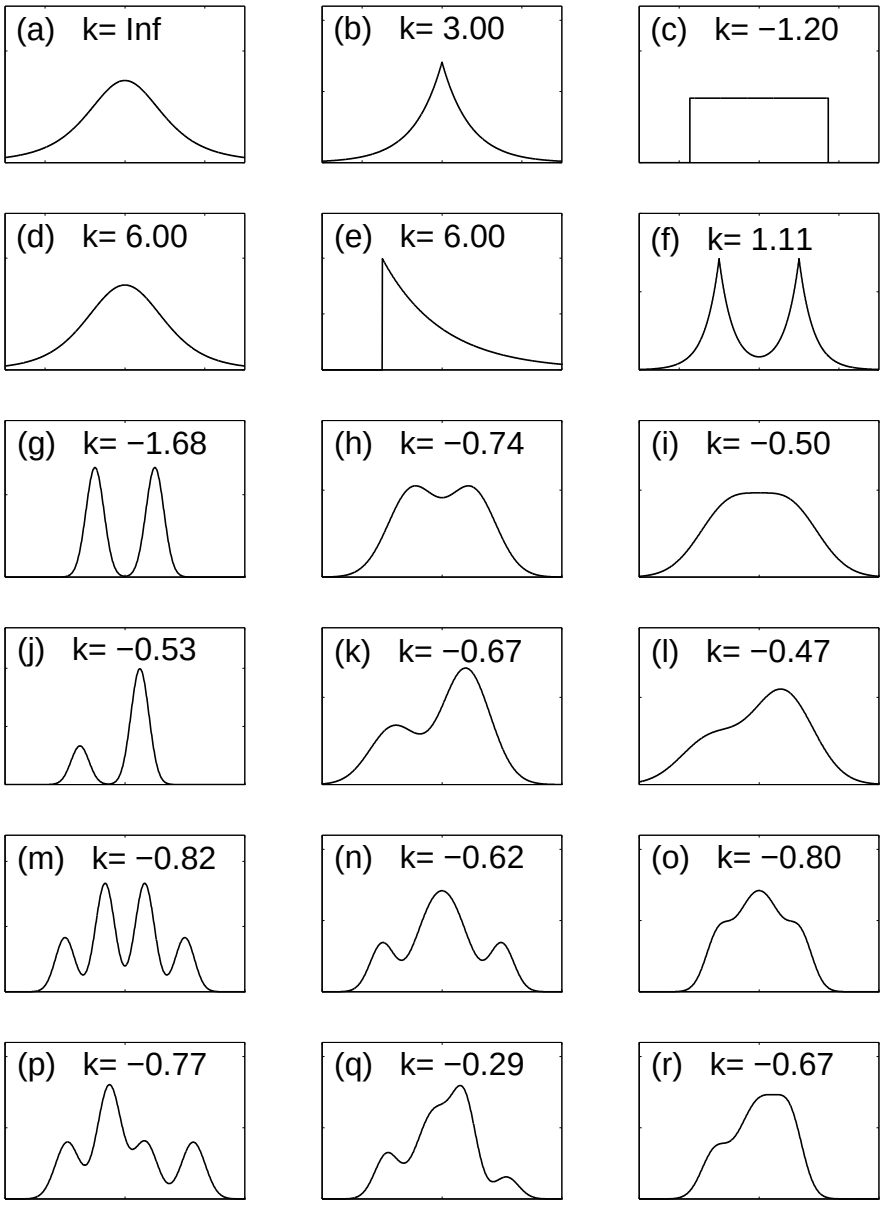


Figure 5: Probability density functions of sources with their kurtoses: (a) Student with 3 degrees of freedom; (b) double exponential; (c) uniform; (d) Student with 5 degrees of freedom; (e) exponential; (f) mixture of two double exponentials; (g)-(h)-(i) symmetric mixtures of two Gaussians: multimodal, transitional and unimodal; (j)-(k)-(l) nonsymmetric mixtures of two Gaussians, multimodal, transitional and unimodal; (m)-(n)-(o) symmetric mixtures of four Gaussians: multimodal, transitional and unimodal; (p)-(q)-(r) nonsymmetric mixtures of four Gaussians: multimodal, transitional and unimodal.

### 7.1 Experimental setup

All of our experiments made use of the same basic procedure for generating data: (1) $N$ samples of each of the $m$ sources were generated according to their probability density functions (pdfs) and placed into an $m \times N$ matrix $X$, (2) a random mixing matrix $A$ was chosen, with random but bounded condition number (between 1 and 2), (3) a matrix $\widetilde{Y}$ of dimension $m \times N$ was formed as the mixture $\widetilde{Y} = AX$, (4) the data were whitened by multiplying $\widetilde{Y}$ by the inverse $P$ of the square root of the sample covariance matrix, yielding an $m \times N$ matrix of whitened data $Y$. This matrix was the input to the ICA algorithms.

Each of the ICA algorithms outputs a demixing matrix $W$ which can be applied to the matrix $Y$ to recover estimates of the independent components. To evaluate the performance of an algorithm, we compared $W$ to the known truth, $W_0 = A^{-1}P^{-1}$, using the following metric:

$$d(V, W) = \frac{1}{2m} \sum_{i=1}^{m} \left( \frac{\sum_{j=1}^{m} |a_{ij}|}{\max_j |a_{ij}|} - 1 \right) + \frac{1}{2m} \sum_{j=1}^{m} \left( \frac{\sum_{i=1}^{m} |a_{ij}|}{\max_i |a_{ij}|} - 1 \right), \tag{17}$$

where $a_{ij} = (VW^{-1})_{ij}$. This metric, introduced by (Amari et al., 1996), is invariant to permutation and scaling of the columns of $V$ and $W$, is always between 0 and $(m-1)$, and is equal to zero if and only if $V$ and $W$ represent the same components. We thus measure the performance of an algorithm by the value $d(W, W_0)$, which we refer to in the following sections as the "Amari error."

### 7.2 ICA algorithms

We briefly overview the other ICA algorithms that we used in our simulations. The FastICA algorithm (Hyvärinen and Oja, 1997) uses a deflation scheme to compute components sequentially. For each component an one-unit contrast function, based on an approximation to the negentropy of a component, is maximized. This function can be viewed as a measure of nongaussianity. The JADE algorithm (Cardoso, 1999) is a cumulant-based method that uses joint diagonalization of a set of fourth-order cumulant matrices. It uses algebraic properties of fourth-order cumulants to define a contrast function that is minimized using Jacobi rotations. The extended Infomax algorithm (Lee et al., 1999) is a variation on the Infomax algorithm (Bell and Sejnowski, 1995) that can deal with either subgaussian or supergaussian components, by adaptively switching between two nonlinearities.

The last three algorithms were used with their default settings. Thus, no fine tuning was performed to increase their performance according to the various sources that we tested. We did the same for the KernelICA algorithms[9], fixing the Gaussian kernel width to $\sigma = 1$ and the regularization parameter to $\kappa = 2 \times 10^{-2}$ for samples $N$ less than 1000 and using $\sigma = 1/2$, $\kappa = 2 \times 10^{-3}$ for larger samples. We used the two initialization techniques presented in Section 5 in a first stage (optimization of an one-unit contrast with a deflation scheme, with a Hermite polynomial kernel of degree $d = 3$ and width $\sigma = 1.5$) and initialized the second stage (optimization of the full $m$-way contrast functions based on the Gaussian kernel) with the result of the first stage. For numbers of components $m$ smaller than 16, the desired solution is reached with a limited number of restarts for the first stage ($\approx m/2$ on average). For $m = 16$, however, the global minimum is not found consistently after a few

9. A MATLAB implementation can be downloaded at `http://www.cs.berkeley.edu/~fbach/`.

restarts. We obtained better results (results that are reported in the last line of Table 2) by using the three other algorithms to initialize our optimization procedure.

### 7.3 Influence of source distributions

In a first series of experiments we tested the various algorithms on a two-component ICA problem, with 250 and 1000 samples, and with all 18 possible source distributions. We studied two kinds of ICA problem. In the first ICA problem, the two source distributions were identical. For each of the 18 sources (a to r), we replicated the experiment 100 times and calculated the average Amari error. The results are reported in Table 1. The table also shows the average across these $18 \times 100$ simulations (the line denoted **mean**). In the second ICA problem, we chose two sources uniformly at random among the 18 possibilities. A total of 1000 replicates were performed, with the average over replications presented in the line denoted **rand** in Table 1.

The results for the KernelICA algorithms show a consistent improvement, up to 50%, over the other algorithms. Comparing just between the two KernelICA algorithms, KernelICA-KGV shows small but consistent performance improvements over KernelICA-KCCA.

In addition, the performance of KernelICA is robust with respect to the source distributions. Performance is similar across multimodal (f, g, j, m, p), unimodal (a, b, d, e, i, l, o, r) and transitional (c, h, k, n, q) distributions. The KernelICA algorithms are particularly insensitive to asymmetry of the pdf when compared to the other algorithms (see, e.g., case q).

### 7.4 Increasing number of components

In a second series of experiments, we tested the algorithms in simulations with 2, 4, 8 and 16 components. Source distributions were chosen at random from the 18 possible sources in Figure 5. The results are presented in Table 2, where we see that KernelICA yields a smaller Amari error than the other ICA algorithms in all cases.

### 7.5 Robustness to Gaussianity

ICA algorithms are known to have difficulties when the sources are nearly Gaussian. To address this issue, we studied two-component ICA problems with identical source distributions. These distributions were chosen at random among a set of mixtures of Gaussians which are at various distances from Gaussianity. This set includes both supergaussian (positive kurtosis) and subgaussian distributions (negative kurtosis). Figure 6 shows the average performance of the algorithms as the kurtosis approaches zero, from above and from below.[10] We see that the performance of all algorithms degrades as the kurtosis approaches zero, and that the KernelICA algorithms are more robust to near-Gaussianity than the other algorithms.

10. Although zero kurtosis does not characterize Gaussianity, kurtosis is often used to score the non-Gaussianity of distributions (e.g., Hyvärinen et al., 2001)

| pdfs | F-ica | Jade | Imax | KCCA | KGV |
|---|---|---|---|---|---|
| a | 7.2 | 6.4 | 50.0 | 7.7 | 6.5 |
| b | 13.1 | 11.0 | 58.4 | 7.9 | 7.1 |
| c | 4.7 | 3.6 | 15.2 | 5.5 | 4.4 |
| d | 12.8 | 10.9 | 54.3 | 13.8 | 11.8 |
| e | 8.8 | 8.0 | 70.3 | 3.7 | 3.3 |
| f | 7.1 | 5.2 | 9.5 | 3.4 | 3.2 |
| g | 3.4 | 2.7 | 19.2 | 2.7 | 2.5 |
| h | 13.4 | 9.1 | 29.7 | 9.8 | 8.3 |
| i | 24.9 | 18.4 | 36.2 | 23.5 | 21.6 |
| j | 20.6 | 16.5 | 52.1 | 3.2 | 3.1 |
| k | 13.4 | 8.1 | 28.4 | 6.0 | 5.0 |
| l | 27.7 | 20.0 | 35.0 | 12.6 | 10.2 |
| m | 8.6 | 6.2 | 25.6 | 13.6 | 10.6 |
| n | 12.9 | 9.1 | 34.3 | 15.1 | 9.2 |
| o | 9.6 | 6.9 | 24.6 | 11.9 | 9.6 |
| p | 9.2 | 6.0 | 27.4 | 8.1 | 6.2 |
| q | 41.2 | 34.4 | 40.6 | 11.8 | 8.2 |
| r | 14.3 | 9.2 | 33.9 | 9.0 | 8.0 |
| **mean** | **14.1** | **10.6** | **35.8** | **9.4** | **7.7** |
| **rand** | **10.8** | **8.6** | **30.4** | **6.9** | **5.4** |

| pdfs | F-ica | Jade | Imax | KCCA | KGV |
|---|---|---|---|---|---|
| a | 4.4 | 3.7 | 1.8 | 3.7 | 3.0 |
| b | 5.8 | 4.1 | 3.4 | 3.7 | 2.9 |
| c | 2.3 | 1.9 | 2.0 | 2.7 | 2.4 |
| d | 6.4 | 6.1 | 6.9 | 7.1 | 5.7 |
| e | 4.9 | 3.9 | 3.2 | 1.7 | 1.5 |
| f | 3.6 | 2.7 | 1.0 | 1.7 | 1.5 |
| g | 1.7 | 1.4 | 0.5 | 1.4 | 1.3 |
| h | 5.5 | 3.9 | 3.2 | 4.3 | 3.6 |
| i | 8.7 | 7.2 | 6.8 | 7.8 | 6.5 |
| j | 6.7 | 4.6 | 57.6 | 1.4 | 1.3 |
| k | 5.7 | 4.0 | 3.5 | 3.2 | 2.6 |
| l | 12.1 | 7.2 | 10.4 | 4.8 | 4.2 |
| m | 3.6 | 2.9 | 4.2 | 6.3 | 4.6 |
| n | 5.4 | 3.5 | 30.6 | 7.6 | 3.0 |
| o | 4.7 | 3.3 | 4.4 | 5.1 | 4.3 |
| p | 4.1 | 3.1 | 7.4 | 3.8 | 3.0 |
| q | 22.9 | 15.8 | 40.9 | 5.1 | 3.9 |
| r | 6.6 | 4.4 | 4.9 | 4.3 | 3.6 |
| **mean** | **6.4** | **4.6** | **10.7** | **4.2** | **3.3** |
| **rand** | **5.3** | **4.3** | **6.9** | **3.0** | **2.4** |

Table 1: The Amari errors (multiplied by 100) for two-component ICA with 250 samples (left) and 1000 samples (right). For each pdf (from a to r), averages over 100 replicates are presented. The overall mean is calculated in the row labeled **mean**. The **rand** row presents the average over 1000 replications when two (generally different) pdfs were chosen uniformly at random among the 18 possible pdfs.

| $m$ | $N$ | # repl | F-ica | Jade | Imax | KCCA | KGV |
|---|---|---|---|---|---|---|---|
| 2 | 250 | 1000 | 11 | 9 | 30 | 7 | 5 |
| | 1000 | 1000 | 5 | 4 | 7 | 3 | 2 |
| 4 | 1000 | 100 | 18 | 13 | 25 | 12 | 11 |
| | 4000 | 100 | 8 | 7 | 11 | 6 | 4 |
| 8 | 2000 | 50 | 26 | 22 | 123 | 30 | 20 |
| | 4000 | 50 | 18 | 16 | 41 | 16 | 8 |
| 16 | 4000 | 25 | 42 | 38 | 130 | 31 | 19 |

Table 2: The Amari errors (multiplied by 100) for $m$ components with $N$ samples: $m$ (generally different) pdfs were chosen uniformly at random among the 18 possible pdfs. The results are averaged over the stated number of replications.

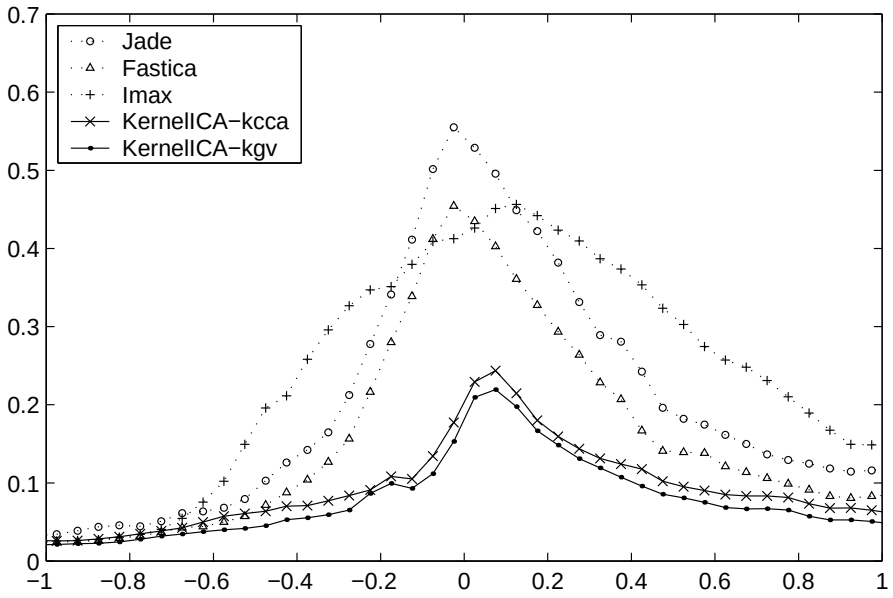


Figure 6: Robustness to near-Gaussianity. The solid lines plot the Amari error of the KernelICA algorithms as the kurtosis approaches zero. The dotted lines plot the performance of the other three algorithms.

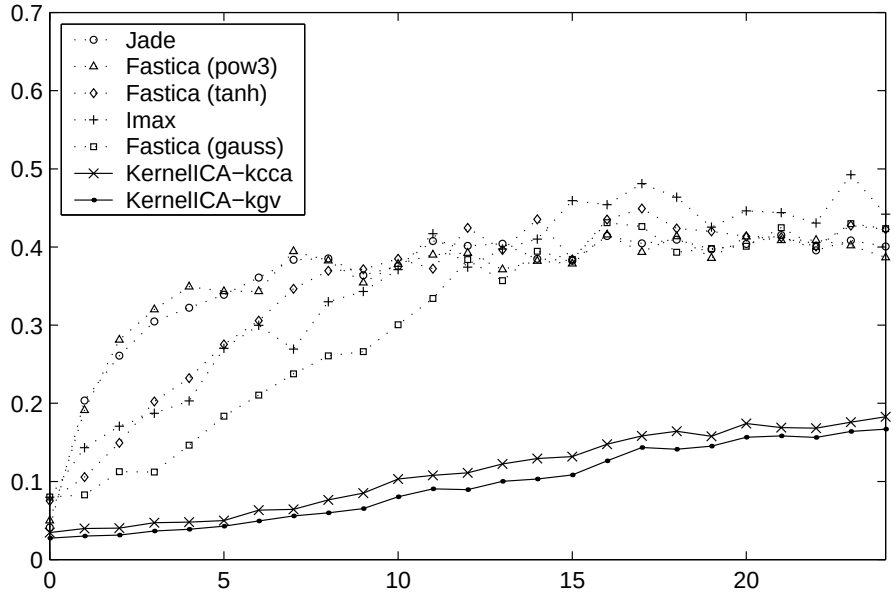


Figure 7: Robustness to outliers. The abscissa displays the number of outliers and the ordinate shows the Amari error.

### 7.6 Robustness to outliers

Outliers are also an important concern for ICA algorithms, given that ICA algorithms are based in one way or another on high-order statistics. Direct estimation of third and fourth degree polynomials can be particularly problematic in this regard, and many ICA algorithms are based on nonlinearities that are more robust to outliers. In particular, in the case of the FastICA algorithm, the hyperbolic tangent and Gaussian nonlinearities are recommended in place of the default polynomial when robustness is a concern (Hyvärinen and Oja, 1997).

We simulated outliers by randomly choosing up to 25 data points to corrupt. This was done by adding the value $+5$ or $-5$ (chosen with probability $1/2$) to a single component in each of the selected data points. We performed 100 replications using source distributions chosen uniformly at random from the 18 possible sources.

The results are shown in Figure 7. We see that the KernelICA methods are significantly more robust to outliers than the other ICA algorithms, including FastICA with the hyperbolic tangent and Gaussian nonlinearities.

### 7.7 Running time

The performance improvements that we have demonstrated in this section come at a computational cost—KernelICA is slower than the other algorithms we studied. The running time is, however, still quite reasonable in the examples that we studied. For example, for $N = 1000$ samples, and $m = 2$ components, it takes 0.05 seconds to evaluate our contrast functions, and 0.25 second to evaluate their derivatives (using Matlab with a Pentium 700 MHz processor). Moreover, the expected scaling of $O(mN)$ for the computations of KCCA and KGV was observed empirically in our experiments.

## 8. Conclusions

We have presented a new approach to ICA based on kernel methods. While most current ICA algorithms are based on using a single nonlinear function—or a small parameterized set of functions—to measure departures from independence, our approach is a more flexible one in which candidate nonlinear functions are chosen adaptively from a reproducing kernel Hilbert space. Our approach thus involves a search in this space, a problem which boils down to finding solutions to a generalized eigenvector problem. Such a search is not present in other ICA algorithms, and our approach to ICA is thus more demanding computationally than the alternative approaches. But the problem of measuring (and minimizing) departure from independence over all possible non-Gaussian source distributions is a difficult one, and we feel that the flexibility provided by our approach is appropriately targeted. Moreover, our experimental results show that the approach is more robust than other ICA algorithms with regards to variations in source densities, degree of non-Gaussianity, and presence of outliers. Our algorithms are thus particularly appropriate in situations where little is known about the underlying sources. It is also worth noting that current algorithms can provide fast approximate solutions that can be improved by KernelICA.

A number of ideas that are closely related to our own have been presented in recent work by other authors. Related work has been presented by Fyfe and Lai (2000), who propose the use of a kernelized version of canonical correlation analysis as an ICA algorithm

(for two-component problems). Canonical correlation analysis in and of itself, however, is simply a feature extraction technique—it can be viewed as an extension of PCA to two variables. CCA does not define an ICA contrast function and it should not be expected to find independent components in general. Indeed, in the experiments presented by Fyfe and Lai (2000), independent components were not always present among the first canonical variates. It is important to emphasize that in our approach, canonical correlation is used to define an ICA contrast function, and this contrast function is subsequently optimized with respect to the parameters of the model to derive an ICA algorithm.

Harmeling et al. (2002) have recently described work on kernel-based ICA methods whose focus is complementary to ours. They show how linear ICA methods in feature space can be used to solve nonlinear ICA problems (problems of the general form $y = f(x)$, for nonlinear $f$). Their method finds a certain number of candidate nonlinear functions of the data as purported independent components. These candidates, however, do not have any optimizing property in terms of an ICA contrast function that allows them to be ranked and evaluated, and in Harmeling et al. (2002) the authors simply pick those components that are closest to the known sources (in simulations in which these sources are known). A possible solution to this problem may lie in combining their approach with ours, using KernelICA in the subspace of feature space identified by their method, and identifying components sequentially.

There are a number of other recent lines of work that are similar to KernelICA in being based on a flexible treatment of the underlying source densities. These include methods that represent estimated source densities via kernel density estimation (Vlassis and Motomura, 2001, Boscolo et al., 2001) and Gaussian mixtures (Attias, 1999, Welling and Weber, 2001), and methods that minimize asymptotic variances of estimators (Pham and Garat, 1997). These methods differ from KernelICA along various dimensions, including statistical foundations (frequentist vs. Bayesian, semiparametric vs. parametric), computational requirements and extensibility, and further work will be needed to disentangle their relative advantages and disadvantages. All of these approaches, however, have in common with KernelICA the emphasis on source adaptivity as a key to the most challenging ICA problems.

The current paper provides a general, flexible foundation for algorithms that measure and minimize departure from independence, and can serve as a basis for exploring various extensions of the basic ICA methodology. There are several directions which we are currently exploring.

First, our approach generalizes in a straightforward manner to multidimensional ICA (Cardoso, 1998), which is a variant of ICA with multivariate components. Indeed, the Gram matrices in our methods can be based on kernel functions computed on vector variables, and the rest of our approach goes through as before. A possible difficulty is that the spectrum of such Gram matrices may not decay as fast as the univariate case, and this may impact the running time complexity.

Second, we can take advantage of the versality of kernels to extend the current ICA model. Although the model we have presented here is based on an assumption of exchangeability (that is, the samples can be randomly permuted without affecting the KernelICA contrast function), in applications where the data are clearly not exchangeable, such as speech processing, it is possible to define kernels on "frames"—short overlapping sequences.

By using frames, the local temporal structure of speech or music is taken into account, and contrast functions that are much more discriminative can be designed. Also, kernels can be defined on data that are not necessarily numerical (e.g., the "string kernels" of Lodhi et al., 2002), and it is interesting to explore the possibility that our kernel-based approach may allow generalizations of ICA to problems involving more general notions of "sources" and "mixtures."

Third, a more thoroughgoing study of the statistical properties of KernelICA is needed. In particular, while we have justified our contrast functions in terms of their mathematical properties in the limiting case of an infinite number of samples, we have not yet fully studied the finite-sample properties of these contrast functions, including the bias and variance of the resulting estimators of the parameters. Nor have we studied the statistical adaptivity of our method as a semiparametric estimation method, comparing its theoretical rate of convergence to that of a method which knows the exact source distributions. Such analyses are needed not only to provide deeper insight into the ICA problem and our proposed solution, but also to give guidance for choosing the values of the free parameters $\sigma$ and $\kappa$ in our algorithm.

Finally, in this paper, we establish a connection between kernel-based quantities and information-theoretic concepts, a connection that may extend beyond its utility for defining contrast functions for ICA. In particular, in recent work we have used the kernel generalized variance to provide a contrast function for a model in which the sources are no longer independent, but factorize according to a tree-structured graphical model (Bach and Jordan, 2002).

## Appendix A. Canonical correlation and its generalizations

In the following appendices, we expand on several of the topics discussed in the paper. This material should be viewed as optional, complementing and amplifying the ideas presented in the paper, but not necessary for a basic understanding of the KernelICA algorithms.

This first section provides additional background on canonical correlation, complementing the material in Section 2.1. In particular, we review the relationship between CCA and mutual information for Gaussian variables, and we motivate the generalization of CCA to more than two variables.

### A.1 CCA and mutual information

For Gaussian random variables there is a simple relationship between canonical correlation analysis and the mutual information. Consider two multivariate Gaussian random variables $x_1$ and $x_2$, of dimension $p_1$ and $p_2$, with covariance matrix $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$. The mutual information, $M(x_1, x_2) = \int p(x_1, x_2) \log[p(x_1, x_2)/p(x_1)p(x_2)]dx_1 dx_2$, is readily computed (Kullback, 1959):

$$M(x_1, x_2) = -\frac{1}{2} \log \left( \frac{\det C}{\det C_{11} \det C_{22}} \right). \tag{18}$$

The determinant ratio appearing in this expression, $\det C/(\det C_{11} \det C_{22})$, is known as the "generalized variance."

As we discussed in Section 2.1, CCA reduces to the computation of eigenvalues of the following generalized eigenvector problem:

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = (1+\rho) \begin{pmatrix} C_{11} & 0 \\ 0 & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}. \tag{19}$$

The eigenvalues appear in pairs: $\{1-\rho_1, 1+\rho_1, \ldots, 1-\rho_p, 1+\rho_p, 1, \ldots, 1\}$, where $p = \min\{p_1, p_2\}$ and where $(\rho_1, \ldots, \rho_p)$ are the canonical correlations.

For invertible $B$, the eigenvalues of a generalized eigenvector problem $Ax = \lambda Bx$ are the same as the eigenvalues of the eigenvector problem $B^{-1}Ax = \lambda x$. Thus the ratio of determinants in Eq. (18) is equal to the product of the generalized eigenvalues of Eq. (19). This yields:

$$M(x_1, x_2) = -\frac{1}{2} \log \prod_{i=1}^{p} (1-\rho_i)(1+\rho_i) = -\frac{1}{2} \sum_{i=1}^{p} \log(1-\rho_i^2). \tag{20}$$

Thus we see that for Gaussian variables, the canonical correlations $\rho_i$ obtained from CCA can be used to compute the mutual information.

While Eq. (20) is an exact result (for Gaussian variables), it also motivates us to consider approximations to the mutual information. Noting that all of the terms in Eq. (20) are positive, suppose that we retain only the largest term in that sum, corresponding to the first canonical correlation. The following theorem, which is easily proved, shows that this yields an approximation to the mutual information.

**Theorem 3** *Let $x_1$ and $x_2$ be Gaussian random variables of dimension $p_1$ and $p_2$, respectively. Letting $\rho(x_1, x_2)$ denote the maximal canonical correlation between $x_1$ and $x_2$, and defining $M_\rho(x_1, x_2) = -\frac{1}{2}\log(1-\rho^2(x_1, x_2))$, we have:*

$$M_\rho(x_1, x_2) \leqslant M(x_1, x_2) \leqslant \min\{p_1, p_2\} M_\rho(x_1, x_2). \tag{21}$$

*Moreover, $M_\rho(x_1, x_2)$ is the maximal mutual information between one-dimensional linear projections of $x_1$ and $x_2$. Also, these bounds are tight—for each of the inequalities, one can find $x_1$ and $x_2$ such that the inequality is an equality.*

### A.2 Generalizing to more than two variables

We generalize CCA to more than two variables by preserving the relationship that we have just discussed between mutual information and CCA for Gaussian variables. (For alternative generalizations of CCA, see Kettenring, 1971).

Consider $m$ multivariate Gaussian random variables, $x_1, \ldots, x_m$, where $x_i$ has dimension $p_i$. Let $C_{ij}$ denote the $p_i \times p_j$ covariance matrix between $x_i$ and $x_j$, and $C$ the overall covariance matrix whose $(i,j)$th block is $C_{ij}$. The mutual information, $M(x_1, \ldots, x_m)$, is readily computed in terms of $C$ (Kullback, 1959):

$$M(x_1, \ldots, x_m) = -\frac{1}{2} \log \left( \frac{\det C}{\det C_{11} \cdots \det C_{mm}} \right). \tag{22}$$

We again refer to the ratio appearing in this expression, $\det C / (\det C_{11} \cdots \det C_{mm})$, as the "generalized variance."

As in the two-variable case, the generalized variance can be obtained as the product of the eigenvalues of a certain generalized eigenvector problem. In particular, we define the following problem:

$$\begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1m} \\ C_{21} & C_{22} & \cdots & C_{2m} \\ \vdots & \vdots & & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mm} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & 0 & \cdots & 0 \\ 0 & C_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & C_{mm} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix}, \quad (23)$$

which we also write as $C\xi = \lambda D\xi$, where $D$ is the block-diagonal matrix whose diagonal blocks are the $C_{ii}$. Given the definition of $C$ and $D$, the ratio of determinants of $C$ and $D$ is clearly equal to the generalized variance. Thus we have:

$$M(x_1, \ldots, x_m) = -\frac{1}{2} \sum_{i=1}^{P} \log \lambda_i, \quad (24)$$

where $\lambda_i$ are the generalized eigenvalues of $C\xi = \lambda D\xi$. Defining CCA as the solution of the generalized eigenvector problem $C\xi = \lambda D\xi$, we again obtain the mutual information in terms of a sum of functions of generalized eigenvalues.[11]

If we wish to obtain a single "maximal canonical correlation," we can proceed by analogy to the two-variable case and take the largest (positive) term in the sum in Eq. (24). Thus, we define the first canonical correlation $\lambda(x_1, \ldots, x_m)$ as the smallest generalized eigenvalue of $C\xi = \lambda D\xi$. We define $M_\lambda(x_1, \ldots, x_m) = -\frac{1}{2} \log \lambda(x_1, \ldots, x_m)$ as an approximation to the mutual information based on this eigenvalue. The following theorem, proved by making use of Jensen's inequality, shows that this approximation yields upper and lower bounds on the mutual information, in the case of Gaussian variables:

**Theorem 4** *Let $(x_1, \ldots, x_m)$ be multivariate Gaussian random variables, where $x_i$ has dimension $p_i$. We have the following lower and upper bounds on the mutual information $M = M(x_1, \ldots, x_m)$:*

$$M_\lambda + \frac{\lambda - 1}{2} \leqslant M_\lambda - \frac{P-1}{2} \log \frac{P - \lambda}{P - 1} \leqslant M \leqslant P M_\lambda, \quad (25)$$

*where $M_\lambda = M_\lambda(x_1, \ldots, x_m)$, $\lambda = \lambda(x_1, \ldots, x_m)$, and $P = \sum_i p_i$.*

Which of the properties of the classical definition of the first canonical correlation generalize to the $m$-variable definition? As we have already noted, the eigenvalues occur in pairs in the two-variable case, while they do not in the $m$-variable case. This implies that the specialization of the $m$-variable definition to $m = 2$, $\lambda(x_1, x_2)$, does not reduce exactly to the classical definition, $\rho(x_1, x_2)$. But the difference is unimportant; indeed, we have $\rho(x_1, x_2) = 1 - \lambda(x_1, x_2)$. A more important aspect of the two-variable case is the fact (cf. Theorem 3) that there is a relationship between $\rho(x_1, x_2)$ and one-dimensional projections of $x_1$ and $x_2$. This relationship is an important one, lying at the heart of the properties of $\mathcal{F}$-correlation. In the following section, we prove that such a relation exists in the $m$-way case as well.

11. Note that the $\lambda_i$ are all nonnegative and sum to $P = \sum_i p_i$. Note also that the $\lambda_i$ do not occur in pairs as they do in the two-variable case. Moreover, the terms in the sum in Eq. (24) are not all positive.

### A.3 $\mathcal{F}$-correlation and independence

Let $y_1, \ldots, y_m$ be univariate random variables, with correlation matrix $\widetilde{C}$, a matrix whose $(i,j)$th element is $\mathrm{corr}(y_i, y_j)$. We define $\nu(y_1, \ldots, y_m)$ to be the minimal eigenvalue of $\widetilde{C}$. Note that a correlation matrix is symmetric positive semidefinite with trace equal to $m$, and thus the eigenvalues are nonnegative and sum to $m$. This implies that $\nu(y_1, \ldots, y_m)$ must always be between zero and one, and is equal to one if and only if $\widetilde{C} = I$. That is, $\nu(y_1, \ldots, y_m) = 1$ if and only if the variables $y_1, \ldots, y_m$ are uncorrelated. The function $\nu$, a function of $m$ univariate random variables, plays a similar role as the correlation between two random variables, as shown in the following theorem:

**Theorem 5** *Let $x_1, \ldots, x_m$ be $m$ multivariate random variables. Let $\lambda(x_1, \ldots, x_m)$ be the first canonical correlation, defined as the smallest generalized eigenvalue of Eq. (23). Then $\lambda(x_1, \ldots, x_m)$ is the minimal possible value of $\nu(y_1, \ldots, y_m)$, where $y_1, \ldots, y_m$ are one-dimensional projections of $x_1, \ldots, x_m$:*

$$\lambda(x_1, \ldots, x_m) = \min_{\xi_1, \ldots, \xi_m} \nu(\xi_1^\top x_1, \ldots, \xi_m^\top x_m). \tag{26}$$

*In addition, $\lambda(x_1, \ldots, x_m) = 1$ if and only if the variables $x_1, \ldots, x_m$ are uncorrelated.*

**Proof** Let $\widetilde{C}(\xi_1^\top x_1, \ldots, \xi_m^\top x_m)$ denote the correlation matrix between $(\xi_1^\top x_1, \ldots, \xi_m^\top x_m)$. If the vectors $\xi_i$ have unit norm then the $(i,j)$th element of $\widetilde{C}(\xi_1^\top x_1, \ldots, \xi_m^\top x_m)$ is just $\xi_i^\top \widetilde{C}_{ij} \xi_j$, where $\widetilde{C}_{ij}$ is the correlation matrix between $x_i$ and $x_j$. We then have:

$$\begin{aligned}
\min_{\xi_1, \ldots, \xi_m} \nu(\xi_1^\top x_1, \ldots, \xi_m^\top x_m) &= \min_{||\xi_1|| = \cdots = ||\xi_m|| = 1} \nu(\xi_1^\top x_1, \ldots, \xi_m^\top x_m) \\
&= \min_{||\xi_1|| = \cdots = ||\xi_m|| = 1} \quad \min_{\beta \in \mathbb{R}^m,\ ||\beta|| = 1} \beta^\top \widetilde{C}(\xi_1^\top x_1, \ldots, \xi_m^\top x_m) \beta \\
&= \min_{||\xi_1|| = \cdots = ||\xi_m|| = ||\beta|| = 1} \sum_{i,j=1}^{m} (\beta_i \xi_i)^\top \widetilde{C}_{ij} (\beta_j \xi_j)
\end{aligned}$$

Minimizing over all possible $\xi_1, \ldots, \xi_m$ and $\beta$ of unit norm is the same as minimizing over all possible $\zeta_i$ such that $\sum_{i=1}^m ||\zeta_i||^2 = 1$, by simply mapping $((\xi_i), \beta)$ to $(\zeta_i)$ by $\zeta_i = \beta_i \xi_i$ and mapping $(\zeta_i)$ to $((\xi_i), \beta)$, by $\beta_i = ||\zeta_i||$ and $\xi_i = \zeta_i / ||\zeta_i||$. Consequently, we have:

$$\min_{\xi_1, \ldots, \xi_m} \nu(\xi_1^\top x_1, \ldots, \xi_m^\top x_m) = \min_{||\zeta|| = 1} \sum_{i,j=1}^{m} \zeta_i^\top \widetilde{C}_{ij} \zeta_j = \min_{||\zeta|| = 1} \zeta^\top \widetilde{C} \zeta = \lambda(x_1, \ldots, x_m),$$

which proves the first part of Theorem 5. Let us now prove the second part.

If the variables $x_1, \ldots, x_m$ are uncorrelated, then any linear projections will also be uncorrelated, so $\nu(\xi_1^\top x_1, \ldots, \xi_m^\top x_m)$ is constant equal to one, which implies by Eq. (26) that $\lambda(x_1, \ldots, x_m) = 1$. Conversely, if $\lambda(x_1, \ldots, x_m) = 1$, then since $\nu$ is always between zero and one, using Eq. (26), for all $\xi_i$, $\nu(\xi_1^\top x_1, \ldots, \xi_m^\top x_m)$ must be equal to one, and consequently, the univariate random variables $\xi_1^\top x_1, \ldots, \xi_m^\top x_m$ are uncorrelated. Since this is true for all one-dimensional linear projections, $x_1, \ldots, x_m$ must be uncorrelated. ■

Applying this theorem to a reproducing kernel Hilbert space $\mathcal{F}$, we see that the $\mathcal{F}$-correlation between $m$ variables is equal to zero if and only if for all functions $f_1, \ldots, f_m$ in $\mathcal{F}$, the variables $f_1(x_1), \ldots, f_m(x_m)$ are uncorrelated. Consequently, assuming a Gaussian kernel, we can use the same line of reasoning as in Theorem 2 to prove that the $\mathcal{F}$-correlation is zero if and only if the variables $x_1, \ldots, x_m$ are pairwise independent.

Concerning our second contrast function, the KGV, Theorem 4 shows that the KGV is always an upper bound of a constant function $\varphi(\lambda)$ of the first canonical correlation $\lambda$. Since $\varphi$ is nonnegative and equal to zero if and only if $\lambda = 1$, this shows that if the KGV is equal to zero, then the first canonical correlation is also zero, and the variables $x_1, \ldots, x_m$ are pairwise independent. As in the KCCA case, the converse is trivially true. Thus, the KGV also defines a valid contrast function.

## Appendix B. Kernel generalized variance and mutual information

In Section 3.4 we noted that there is a relationship between the kernel generalized variance (KGV) and the mutual information in the bivariate case. In particular, we claim that in the population case (where no regularization is needed), the KGV approaches a limit as the kernel width approaches zero, and that in the bivariate case, this limit is equal to the mutual information, up to second order, expanding around independence. A full proof of this result is beyond the scope of this paper, but in this section we provide a sketch of the proof. We first introduce a new information criterion for discrete multinomial random variables, show its link to the mutual information and then extend it to continuous variables.

### B.1 Multinomial and Gaussian variables

We begin by establishing a relationship between a pair of multinomial random variables and a pair of Gaussian random variables with the same covariance structure. Let $x$ and $y$ be multinomial random variables of dimension $p$ and $q$, respectively, and let $P$ denote the $p \times q$ joint probability matrix whose $(i,j)$th element, $P_{ij}$, is equal to $P(x = i, y = j)$. As usual, we also represent these variables as unit basis vectors, $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$, such that $P(X_i = 1, Y_j = 1) = P_{ij}$.

Let $p_x$ denote a $p$-dimensional vector representing the marginal probability distribution of $x$, and let $p_y$ denote a $q$-dimensional vector representing the marginal probability distribution of $y$. In the following, if $r$ is a $R \times 1$ vector, $\mathrm{diag}(r)$ denotes the diagonal $R \times R$ matrix with diagonal $r$. The covariance structure of $(X, Y)$ can be written as follows, where $D_{p_x} = \mathrm{diag}(p_x)$ and $D_{p_y} = \mathrm{diag}(p_y)$:

$$E(XY^\top) = P, \quad E(X) = p_x, \quad E(Y) = p_y, \quad E(XX^\top) = D_{p_x}, \quad E(YY^\top) = D_{p_y},$$

which implies $C_{XY} = P - p_x p_y^\top$, $C_{XX} = D_{p_x} - p_x p_x^\top$, and $C_{YY} = D_{p_y} - p_y p_y^\top$. Let $X^G$ and $Y^G$ denote Gaussian random variables that have the same covariance structure as $X$ and $Y$. It is easy to show that the mutual information between $X^G$ and $Y^G$, which we denote as $I^G$, is equal to $I^G = -\frac{1}{2} \log \det(I - C^\top C) = -\frac{1}{2} \log \det(I - CC^\top)$, where $C = D_{p_x}^{-1/2}(P - p_x p_y^\top) D_{p_y}^{-1/2}$.

We now expand $I^G$ near independence, that is, when we assume $P_{ij} = p_{xi} p_{yj}(1 + \varepsilon_{ij})$ where $\varepsilon$ is a matrix with small norm. In this case, the matrix $C$ defined as $C = D_{p_x}^{-1/2}(P -$

$p_x p_y^\top) D_{p_y}^{-1/2} = D_{p_x}^{1/2} \varepsilon D_{p_y}^{1/2}$, has also a small norm and we can expand $I^G$ as follows:

$$I^G = -\frac{1}{2} \log\det(I - CC^\top) \approx \frac{1}{2} \operatorname{tr}(CC^\top),$$

Indeed, if we let $s_i$ denote the singular values of $C$ (which are close to zero because the matrix $C$ has small norm), we have:

$$-\frac{1}{2} \log\det(I - CC^\top) = -\frac{1}{2} \sum_i \log(1 - s_i^2) \approx \frac{1}{2} \sum_i s_i^2 = \frac{1}{2} \operatorname{tr}(CC^\top).$$

Thus, we obtain:

$$I^G \approx \frac{1}{2} \operatorname{tr}(D_{p_x} \varepsilon D_{p_y} \varepsilon^\top) = \frac{1}{2} \sum_{ij} \varepsilon_{ij}^2 p_{xi} p_{yj}. \tag{27}$$

Let us now expand the mutual information $I = I(x, y)$, using the Taylor expansion $(1 + \varepsilon)\log(1 + \varepsilon) \approx \varepsilon + \varepsilon^2/2$:

$$I = \sum_{ij} p_{xi} p_{yj} (1 + \varepsilon_{ij}) \log(1 + \varepsilon_{ij}) \approx \sum_{ij} p_{xi} p_{yj} (\varepsilon_{ij} + \frac{1}{2} \varepsilon_{ij}^2) = \frac{1}{2} \sum_{ij} \varepsilon_{ij}^2 p_{xi} p_{yj}, \tag{28}$$

using $\sum_{ij} \varepsilon_{ij} p_{xi} p_{yj} = \sum_{ij} (1 + \varepsilon_{ij}) p_{xi} p_{yj} - \sum_{ij} p_{xi} p_{yj} = \sum_{ij} P_{ij} - \sum_{ij} p_{xi} p_{yj} = 1 - 1 = 0$.

In summary, for multinomial random variables $x$ and $y$, we have defined the quantity $I^G(x, y)$ in terms of the mutual information between Gaussian variables with the same covariance. We have shown that this quantity is equal up to second order to the actual mutual information, $I(x, y)$, when we expand "near independence." We now extend these results, defining the quantity $I^G$, which we will refer to as the *Gaussian mutual information (GMI)*, for continuous univariate variables.

### B.2 A new information measure for continuous random variables

Let $x$ and $y$ be two continuous random variables. Their mutual information, $I(x, y) = \int p(x, y) \log[p(x, y)/p(x)p(y)] dxdy$, can be defined as the upper bound of the mutual information between all discretizations of $x$ and $y$ (Kolmogorov, 1956). Behind this definition lies the crucial fact that when refining the partitions of the sample space used to discretize $x$ and $y$, the discrete mutual information must increase.

By analogy, we generalize the GMI to continuous variables: the GMI $I^G(x, y)$ is defined to be the supremum of $I^G(x_d, y_d)$ for discretizations $(x_d, y_d)$ of $x$ and $y$. In order to have a proper definition, we need to check that when we refine the partitions, then the discrete GMI can only increase. It is easy to check that the associated Gaussian random variables before the refinement are linear combinations of the associated Gaussian random variables after the refinement, which implies that the refinement can only increase the mutual information between the associated Gaussian random variables. But this implies that $I^G$ can only increase during a refinement.

Another property of the Gaussian mutual information that we will need in the following section, one that is also shared by the classical mutual information, is that it is equal to the limit of the discrete mutual information, when the discretization is based on a uniform mesh whose spacing tends to zero. From this, it can be shown that the equality of $I$ and $I^G$ up to

second order around independence still holds for continuous variables. Using the singular value decomposition for bivariate distributions (Buja, 1990), an alternate justification of this expansion could be derived.

### B.3 Relation with kernel generalized variance

Let us consider the feature space $\mathcal{F}_\sigma$ associated with a Gaussian kernel $K(x,y) = \frac{1}{\sqrt{2\pi}\sigma} G\left(\frac{x-y}{\sigma}\right)$ where $G(x) = e^{-x^2/2}$. Let us denote $G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} G\left(\frac{x}{\sigma}\right)$, such that $\int G_\sigma(x)dx = 1$. As we saw in Section 2, the space $\mathcal{F}_\sigma$ can be viewed as the completion of the space of finite linear combinations of functions of the form $G_\sigma(x - x^i)$ where $x^i \in \mathbb{R}$. Let $\{x^i\}$ be a mesh of uniformly distributed points in $\mathbb{R}$ with spacing $h$. Using these fixed points, we define $\mathcal{F}_\sigma\{x^i\}$ to be the (finite-dimensional) linear span of the functions $f_i = G_\sigma(x - x^i)$. Similarly we define a mesh $\{y^j\}$ for the second random variable, and let $\mathcal{F}_\sigma\{y^j\}$ denote the linear span of the functions $g_j = G_\sigma(x - y^j)$.

The contrast function $I_{\delta\mathcal{F}}(\sigma)$ based on the KGV is defined as the mutual information between Gaussian random variables that have the same covariance structure as $\Phi(x)$ and $\Phi(y)$. Let $I_{\delta\mathcal{F}}(h,\sigma)$ be the mutual information between finite-dimensional Gaussian random variables that have the same covariance structure as the projections of $\Phi(x)$ and $\Phi(y)$ onto $\mathcal{F}_\sigma\{x^i\}$ and $\mathcal{F}_\sigma\{y^j\}$.

As the spacing $h$ tends to zero and as the number of points tends to infinity, the spaces $\mathcal{F}_\sigma\{x^i\}$ and $\mathcal{F}_\sigma\{y^j\}$ tend to the feature space $\mathcal{F}_\sigma$, so that $I_{\delta\mathcal{F}}(h,\sigma)$ tends to $I_{\delta\mathcal{F}}(\sigma)$. We now relate the quantity $I_{\delta\mathcal{F}}(h,\sigma)$ to the Gaussian mutual information $I^G(x,y)$. We have:

$$\begin{aligned} E\langle f_i, \Phi(x)\rangle\langle g_j, \Phi(y)\rangle &= \int G_\sigma(x - x^i) G_\sigma(y - y^j) p(x,y) dx\, dy \\ &= [G_\sigma(x) G_\sigma(y) * p(x,y)](x^i, y^j) \\ &= p_{G_\sigma}(x^i, y^j), \end{aligned}$$

where $p_{G_\sigma}$, a smoothed version of $p$, is well defined as a probability density function because $G_\sigma$ is normalized. Similar formulas can be obtained for the other expectations:

$$E\langle f_i, \Phi(x)\rangle = (p_{G_\sigma})_x(x^i), \qquad E\langle g_j, \Phi(x)\rangle = (p_{G_\sigma})_y(y^j)$$

and covariances:

$$E\langle f_i, \Phi(x)\rangle\langle f_j, \Phi(x)\rangle \propto \delta_{ij} \times (p_{G_\sigma})_x(x^i) \text{ if } \sigma \ll h \ll 1.$$

These identities ensure that, as $h$ and $\sigma$ tends to zero, the covariance structure of the projections of $\Phi(x)$ and $\Phi(y)$ onto $\mathcal{F}_\sigma\{x^i\}$ and $\mathcal{F}_\sigma\{y^j\}$ is equivalent to the covariance obtained through the discretization on the mesh $\{x^i, y^j\}$ of random variables having joint distribution $p_{G_\sigma}$. This implies that, as $h$ and $\sigma$ tends to zero, $I_{\delta\mathcal{F}}(h,\sigma)$ is equivalent to the Gaussian mutual information of the variables $x$ and $y$, smoothed by $G_\sigma$. Moreover, as the smoothing parameter $\sigma$ tends to zero, $p_{G_\sigma}$ tends to $p$, and we see that $I_{\delta\mathcal{F}}(\sigma)$ tends to $I^G$. Thus as $\sigma$ tends to zero, the KGV tends to the Gaussian mutual information.

## Appendix C. Spectrum of Gram matrices

The computational efficiency of our algorithms relies on the approximation of Gram matrices by matrices of very low rank.[12] In this section we present theoretical results from functional analysis that justify the use of such approximations. For simplicity, we restrict ourselves to Gaussian kernels, but many of these results can be generalized to other translation-invariant kernels.

The rank of approximations to Gram matrices depends on the decay of the distribution of the eigenspectrum of these matrices. As pointed out by Williams and Seeger (2000), for one-dimensional input spaces the eigenvalues decay geometrically if the input density is Gaussian. We discuss a generalization of this result in this section, showing that the decay of the spectrum depends in general on the decay of the tails of the underlying distribution $p(x)$ of the data.

The study of the spectrum of Gram matrices calculated from a kernel $K(x, y)$ is usually carried out by studying the spectrum of an associated integral operator, and using the Nyström method to relate these spectra (Baker, 1977). We briefly review the relevant machinery.

### C.1 Integral Operators and Nyström method

Let $K \in L^2(\mathbb{R}^d \times \mathbb{R}^d)$ denote a symmetric kernel and $p(x)$ the probability density of a random variable on $\mathbb{R}^d$. We assume that $p$ is bounded and that the integral $\int_{\mathbb{R}^d \times \mathbb{R}^d} |K(x, y)|p(x)dxdy$ is finite. We define the *integral operator* $T$, from $L^2(\mathbb{R}^d)$ to $L^2(\mathbb{R}^d)$, as follows:

$$T : \phi(y) \mapsto \int_{\mathbb{R}^d} K(x, y)p(x)\phi(x)dx. \tag{29}$$

$T$ is called a Hilbert-Schmidt operator (Brezis, 1980). It is known that the spectrum of such an operator is a sequence of real numbers tending to zero, where the spectrum is defined as the set of $\lambda_i$ for which there exists $\phi_i \in L^2(\mathbb{R}^d)$ such that $T\phi_i = \lambda_i\phi_i$:

$$\int_{\mathbb{R}^d} K(x, y)p(x)\phi_i(x)dx = \lambda_i\phi_i(y). \tag{30}$$

The eigenvalues $\lambda_i$ and eigenvectors $\phi_i$ are often approximated using the "Nyström method," which relates them to the spectra of Gram matrices of points sampled from $p$. That is, the expectation in Eq. (29) is approximated by the sample mean $T\phi(y) \approx \frac{1}{N}\sum_{k=1}^{N} K(x_k, y)\phi(x_k)$, where $x_k$ are $N$ data points sampled from $p$. Substituting this into the definition of an eigenvector in Eq. (30) and evaluating at $y = x_j$, we get:

$$\frac{1}{N}\sum_{k=1}^{N} K(x_k, x_j)\phi_i(x_k) \approx \lambda_i\phi_i(x_j), \tag{31}$$

and thus $\Phi_i = (\phi_i(x_1), \ldots, \phi_i(x_N))^\top$ is an eigenvector of the Gram matrix $K = (K(x_i, x_j))$ with eigenvalue $N\lambda_i$:

$$\frac{1}{N}K\Phi_i = \lambda_i\Phi_i.$$

12. Note that a (non-centered) Gram matrix is always invertible (e.g., Schölkopf and Smola, 2001), given distinct sample points and a Gaussian kernel, so any low-rank representation of such a matrix is necessarily an approximation.

| Decay of $p(x)$ | Bound of $h(t)$ | Decay of $\lambda_n$ |
|---|---|---|
| compact support | $o(\log t)$ | $e^{-An \log n}$ |
| $e^{-x^2/2}$ | $\log t$ | $e^{-An}$ |
| $\|x\|^{-d},\ d > 2$ | $t^{1/d+\varepsilon}$ | $n^{-d+\varepsilon}$ |

Table 3: Bounds for the number of eigenvalues greater than $\eta$, as a function $h(t)$ of $t = 1/\eta$, and the $n$-th eigenvalue $\lambda_n$ of the integral operator $T$. The number of eigenvalues greater than $\eta$, for an $N \times N$ Gram matrix, is bounded by $h(t)$, where $t = N/\eta$ (see text for details).

Consequently, the eigenvalues of the Gram matrix $K$ are approximately equal to $N\lambda$, where $\lambda$ ranges over eigenvalues of the integral operator. It is also possible to approximate the eigenfunctions $\phi_i$ using this approach (see Baker, 1977).

Two problems arise: How fast does the spectrum of the integral operator decay for various kernels $K$ and densities $p$? How close are the eigenvalues of the Gram matrices to $N$ times the eigenvalues of the integral operator? In the following section, we overview some theoretical results that give asymptotic bounds for the decay of the spectra of integral operators, and we provide empirical results that relate the eigenvalues of Gram matrices to the eigenvalues of the integral operator.

### C.2 Spectra of integral operators

Widom (1963, 1964) provides some useful results regarding the spectra of the operator $T$ defined in Eq. (29) for translation-invariant kernels of the form $k(x-y)$. He shows that the rate of decay of the spectrum depends only on the rate of decay of the Fourier tranform $\nu(\omega)$ of $k$, and of the rate of decay of the probability density function of the underlying input variable $x$. Moreover, he provides asymptotic equivalents for many cases of interest. Most of the results can be generalized to multivariate kernels. For the case of Gaussian kernels, we summarize some of the pertinent results in Table 3. Note that except for heavy-tailed distributions (those with polynomial decay), the spectrum vanishes at least geometrically.

### C.3 Nyström approximation

We now provide empirical results about how the spectra of Gram matrices relate to the spectrum of the associated integral operator. We study the Gaussian distribution, where an exact result can be calculated, and the Student distribution with three degrees of freedom, where a function of the form $\lambda_n = \frac{a}{(b+n)^4}$ can be fit tightly to the spectrum.[13] In both cases, we used distributions with unit variance.

We sampled $N$ data points from these distributions, for $N$ ranging from $2^3$ to $2^{13}$, and computed the spectra of the resulting Gram matrices. The results are plotted in Figure 8. We see that the spectrum of an $N \times N$ Gram matrix, which we denote as $\lambda_{k,N}$, is composed

13. Note that this is consistent with the bounds in Table 3, since the Student distribution with three degrees of freedom has a density that decays as $|x|^{-4}$.

of two regimes. For eigenvalues $\lambda_{k,N}$ up to a given rank $k_0(N)$, the eigenvalues are very close to their limiting value $\lambda_k/N$, where $\lambda_k$ is the $k$-th eigenvalue of the associated integral operator. After $k_0(N)$, the spectrum decays very rapidly.

The important point is that the spectra of the Gram matrices decay at least as rapidly as $N$ times the eigenvalues of the integral operators. Consequently, we need only consider low-rank approximations of order $M = h(N/\eta)$, where, as $t = N/\eta$ tends to infinity, $h(t)$ grows as described in Table 3. Given that we choose the precision to be proportional to $N$, i.e. $\eta = \eta_0 N$, the number of eigenvalues we need to consider is bounded by a constant that depends solely on the input distribution.

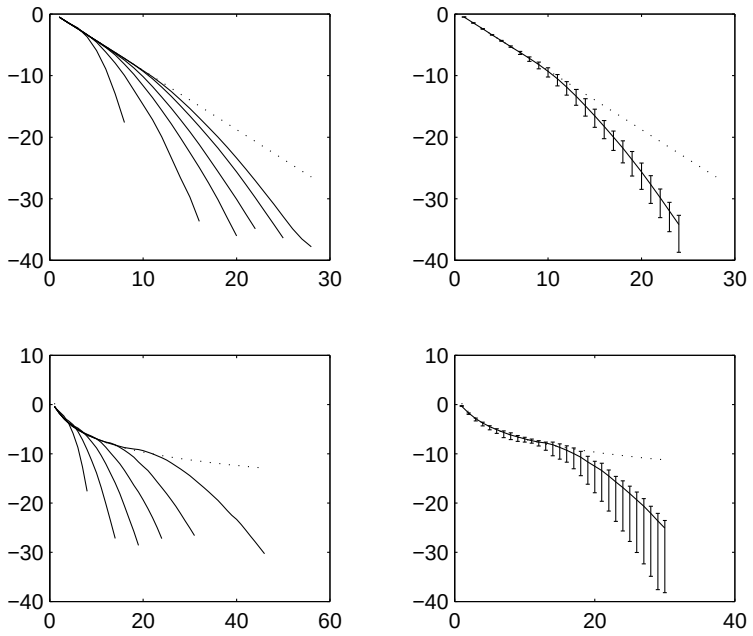


Figure 8: Spectra for two different input densities (top: Gaussian, bottom: Student distribution with three degrees of freedom). The dashed lines are the exact or fitted (see text for details) logarithm of the spectra $\log \lambda_k$, plotted as a function of the eigenvalue number $k$. (Left) The solid lines represent $\log \frac{1}{N}\lambda_{k,N}$, for $N = 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}$. (Right) For $N = 2^{11} = 2048$, the solid line represents $\log \frac{1}{N}\lambda_{k,N}$, plotted as a function of $k$, while the lower and upper ends of the error bars represent the minimum and the maximum of $\log \frac{1}{N}\lambda_{k,N}$ across 20 replications.

## Appendix D. Derivatives

In this section we provide a discussion of the computation of the derivatives of our contrast functions. The computation of these derivatives is a straightforward application of the chain rule, where the core subroutine is the computation of the derivatives of the Gram matrices.

This latter computation is not entirely straightforward, however, and it is our focus in this section. Note that the computation of the derivatives of a Gram matrix arises outside of the ICA setting, and this material may therefore have utility for other kernel-based methods.

The key problem is that although the Gram matrix $K$ is symmetric and positive semidefinite, its derivative with respect to some underlying variable is symmetric but not in general positive or negative semidefinite. Consequently, incomplete Cholesky decomposition cannot be used directly to find low-rank approximations of derivatives.

Fortunately, for Gaussian kernels, it is possible to express the derivatives as sum and/or difference of positive semidefinite matrices that themselves are Gram matrices, and to which incomplete Cholesky decomposition can be applied. More precisely, if $w \in \mathbb{R}^m$ is a row of our parameter matrix $W$, then the Gram matrix that we have to differentiate has its $(a,b)$th element equal to $\exp\left\{-\frac{1}{2\sigma^2}(w^\top x_a - w^\top x_b)^2\right\}$. Without loss of generality, let us differentiate this expression around $w = (1,0,\ldots,0)^\top$. We obtain:

$$\left(\partial_{w_j} K\right)_{ab} = -\frac{1}{\sigma^2}(x_{a1} - x_{b1})(x_{aj} - x_{bj})e^{-\frac{1}{2\sigma^2}(x_{a1}-x_{b1})^2}. \tag{32}$$

This is not a Gram matrix, because the Fourier transform of $x \mapsto x_1 x_j e^{-x_1^2/2\sigma^2}$ is not real-valued and nonnegative. We instead proceed by decomposing the derivative as a difference of Gram matrices. Two cases arise:

- If $j = 1$, from Eq. (32), we have a matrix whose elements are of the form $f(x_{a1} - x_{b1})$ where $f(x) = x^2 e^{-x^2/2\sigma^2}$. Let $\hat{f}$ be the Fourier transform of $f$. The Fourier transform of $g(x) = e^{-x^2/2\sigma^2}$ is $\nu(\omega) = \sqrt{2\pi}\sigma e^{-\omega^2\sigma^2/2}$, and we have:

$$\begin{aligned}
\hat{f}(\omega) &= -\frac{d^2}{d\omega^2}(\nu(\omega)) = -\frac{d^2}{d\omega^2}(\sqrt{2\pi}\sigma e^{-\omega^2\sigma^2/2}) \\
&= \sigma^2(1-\sigma^2\omega^2)\sqrt{2\pi}\sigma e^{-\omega^2\sigma^2/2} \\
&= \sigma^2\nu(\omega) - \sigma^4\omega^2\sqrt{2\pi}\sigma e^{-\omega^2\sigma^2/2} \\
&= \sigma^2\nu(\omega) - \hat{h}(\omega)
\end{aligned} \tag{33}$$

  The function $h = \sigma^2 g - f$ has a nonnegative Fourier transform, which implies that the matrix whose elements are $\sigma^2 g(x_{a1} - x_{b1}) - f(x_{a1} - x_{b1})$ is positive semidefinite. Since $g(x)$ also induces a positive semidefinite matrix, we have managed to decompose our derivative.

- If $j \neq 1$, from Eq. (32), we have a matrix induced by a function of the form $f(x_{a1} - x_{b1}, x_{aj} - x_{bj})$, where $f(x,y) = xye^{-x^2/2\sigma^2}$. We use the following trick to reduce the problem to the previous case. For a positive real number $\gamma$, we write:

$$xy = \frac{1}{2}(\sigma^2 - x^2) + \frac{1}{2}(\gamma^2 - y^2) - \frac{1}{2}(\sigma^2 + \gamma^2 - (x+y)^2). \tag{34}$$

  Thus we can decompose the function $f_\gamma(x,y) = xye^{-x^2/2\sigma^2}e^{-y^2/2\gamma^2}$ as before, letting $f_\gamma(x,y) = h_\sigma(x,y) + h_\gamma(x,y) - h_{\sigma,\gamma}(x,y)$ where $h_\sigma(x,y) = \frac{1}{2}(\sigma^2 - x^2)e^{-x^2/2\sigma^2}e^{-y^2/2\gamma^2}$, $h_\gamma(x,y) = \frac{1}{2}(\gamma^2 - y^2)e^{-x^2/2\sigma^2}e^{-y^2/2\gamma^2}$ and $h_{\sigma,\gamma}(x,y) = \frac{1}{2}(\sigma^2 + \gamma^2 - (x+y)^2)e^{-x^2/2\sigma^2}e^{-y^2/2\gamma^2}$ all have real positive Fourier transforms. To approximate $f$ based on $f_\gamma$, we note that:

$$|f(x,y) - f_\gamma(x,y)| \leqslant |xy|e^{-x^2/2\sigma^2}y^2/2\gamma^2. \tag{35}$$

Given that our goal is to obtain a descent direction and not an exact value for the derivative, we can chose a large value of $\gamma$ (we used $\gamma = 50$ in our simulations) and obtain satisfactory results.[14]

In summary, we have managed to decompose the derivatives of Gram matrices in terms of the difference of two matrices to which we can apply our low-rank decomposition algorithm. The final time complexity is $O(m^2M^2N)$ for the derivatives of the contrast functions we use.

A similar reduced complexity can be achieved through first-difference approximations of the derivatives, when the particular structure of the optimization is used. Indeed, since the derivatives have $m(m-1)/2$ components, we would need as many evaluations of the contrast functions, which would take $O(m^3M^2N)$. Nonetheless, each new evaluation requires to compute incomplete Cholesky decompositions only for two components (all others are held fixed to compute a partial derivative), so the complexity can be reduced to the one of $m$ evaluations of the constrast functions, that is $O(m^2M^2N)$.

## Acknowledgments

We would like to thank Alexander Smola for useful comments on kernel canonical correlation analysis and regularization. We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642), and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637).

## References

S. Akaho. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. Tokyo: Springer-Verlag, 2001.

S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems, 8*. Cambridge, MA: MIT Press, 1996.

H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

F. R. Bach and M. I. Jordan. Tree-dependent component analysis. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI 2002)*. San Francisco, CA: Morgan Kaufmann, 2002.

C. Baker. *The Numerical Treatment of Integral Equations*. Oxford, UK: Clarendon Press, 1977.

14. Note that the variables $x$ and $y$ have unit variance, and thus by the Chebyshev bound $y^2$ is unlikely to be larger than $\gamma = 50$.

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

P. J. Bickel, C. A. J. Klaassen, Y. Ritov, and J. A. Wellner. *Efficient and Adaptive Estimation for Semiparametric Models*. New York: Springer-Verlag, 1998.

M. Borga, H. Knutsson, and T. Landelius. Learning canonical correlations. In *Proceedings of the Tenth Scandinavian Conference on Image Analysis (SCIA'97)*, 1997.

R. Boscolo, H. Pan, and V. P. Roychowdhury. Non-parametric ICA. In *Proceedings of the Third International Conference on Independent Component Analysis and Blind Source Separation (ICA 2001)*, 2001.

L. Breiman and J. H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80:580–598, 1985.

H. Brezis. *Analyse Fonctionelle*. Paris: Masson, 1980.

A. Buja. Remarks on functional canonical variates, alternating least squares methods, and ACE. *Annals of Statistics*, 18(3):1032–1069, 1990.

J.-F. Cardoso. Multidimensional independent component analysis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, 1998.

J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157–192, 1999.

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

T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: John Wiley & Sons, 1991.

N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2-3):127–152, 2002.

R. Durrett. *Probability: Theory and Examples*. Belmont, CA: Duxbury Press, 1996.

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, 1999.

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

C. Fyfe and P. L. Lai. ICA using kernel canonical correlation analysis. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, 2000.

F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1996.

S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller. Kernel feature spaces and nonlinear blind source separation. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems, 14*. Cambridge, MA: MIT Press, 2002.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. London: Chapman and Hall, 1990.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. New York: John Wiley & Sons, 2001.

A. Hyvärinen and E. Oja. A fast fixed point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

C. Jutten and J. Herault. Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

J. R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.

A. N. Kolmogorov. On the Shannon theory of information transmission in the case of continuous signals. *IRE Transactions on Information Theory*, 2(4):102–108, 1956.

S. Kullback. *Information Theory and Statistics*. New York: John Wiley & Sons, 1959.

T.-W. Lee, M. Girolami, and T. J. Sejnowski. Independent component analysis using an extended Infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, 11(2):417–441, 1999.

S. Leurgans, R. Moyeed, and B. W. Silverman. Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society*, B, 55(3):725–740, 1993.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

T. Melzer, M. Reiter, and H. Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2001)*. New York: Springer-Verlag, 2001.

D. T. Pham and P. Garat. Blind separation of mixtures of independent sources through a quasi-maximum likelihood approach. *IEEE Transactions on Signal Processing*, 45(7): 1712–1725, 1997.

S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Harlow, UK: Longman Scientific & Technical, 1988.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. Cambridge, MA: MIT Press, 2001.

B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(3):1299–1319, 1998.

A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In P. Langley, editor, *Proceedings of Seventeenth International Conference on Machine Learning (ICML 2000)*. San Francisco, CA: Morgan Kaufmann, 2000.

G. Szegö. *Orthogonal Polynomials*. Providence, RI: American Mathematical Society, 1975.

V. N. Vapnik. *Statistical Learning Theory*. New York: John Wiley & Sons, 1998.

N. Vlassis and Y. Motomura. Efficient source adaptivity in independent component analysis. *IEEE Transactions on Neural Networks*, 12(3):559–566, 2001.

M. Welling and M. Weber. A constrained EM algorithm for independent component analysis. *Neural Computation*, 13(3):677–689, 2001.

H. Widom. Asymptotic behavior of the eigenvalues of certain integral equations. *Transactions of the American Mathematical Society*, 109:278–295, 1963.

H. Widom. Asymptotic behavior of the eigenvalues of certain integral equations II. *Archive for Rational Mechanics and Analysis*, 17:215–229, 1964.

C. K. I. Williams and M. Seeger. Effect of the input density distribution on kernel-based classifiers. In P. Langley, editor, *Proceedings of Seventeenth International Conference on Machine Learning (ICML 2000)*. San Francisco, CA: Morgan Kaufmann, 2000.

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems, 13*. Cambridge, MA: MIT Press, 2001.

S. Wright. Modified Cholesky factorizations in interior-point algorithms for linear programming. *SIAM Journal on Optimization*, 9(4):1159–1191, 1999.