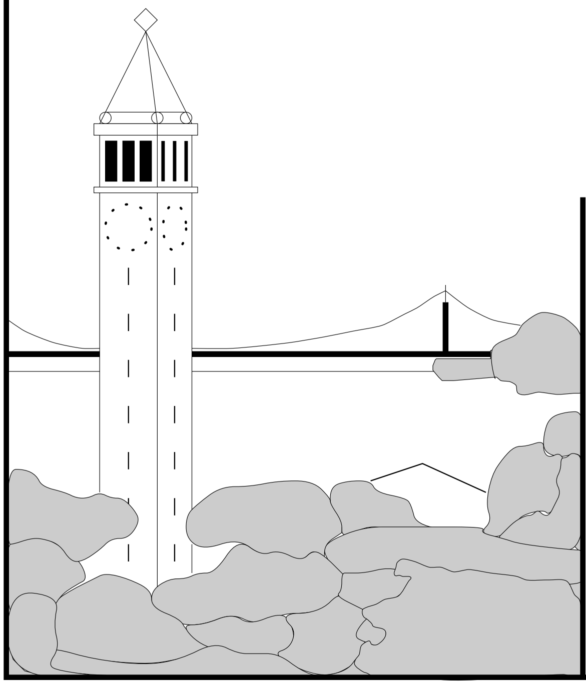
Kernel Independent Component Analysis

Francis R. Bach
fbach@cs.berkeley.edu
Division of Computer Science
University of California
Berkeley, CA 94720, USA

Michael I. Jordan
jordan@cs.berkeley.edu
Division of Computer Science and Department of Statistics
University of California
Berkeley, CA 94720, USA



Report No. UCB/CSD-01-1166

November 2001

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Kernel Independent Component Analysis

Francis R. Bach
fbach@cs.berkeley.edu
Division of Computer Science
University of California
Berkeley, CA 94720, USA

Michael I. Jordan
jordan@cs.berkeley.edu
Division of Computer Science and Department of Statistics
University of California
Berkeley, CA 94720, USA

November 2001

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.

1 Introduction

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

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 $\Re$ to $\Re$, 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 $\omega \in \Re$), 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 $\Re$, 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):

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

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, $\mathcal{F}$-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 η is a precision parameter and $h(t)$ is a slowly growing function of t. Moreover, in computing the contrast function, the precision η 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 ρ_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 $\xi_1^T x_1$ and $\xi_2^T x_2$ of x_1 and x_2:

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

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 ξ_1 and ξ_2, we obtain:

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

and

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

Normalizing the vectors ξ_1 and ξ_2 by letting $\xi_1^T C_{11}\xi_1 = 1$ and $\xi_2^T C_{22}\xi_2 = 1$, 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 ρ_{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:

$$\begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & 0 & \cdots & 0 \\ 0 & C_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & C_{nn} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix}, \tag{3}$$

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 $\mathcal{X} = \Re^p$, 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 Φ 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 to translation-invariant kernels in this paper; that is, to kernel functions of the form $K(x, y) = k(x - y)$. In this case the RKHS can be described succinctly using Fourier theory (Girosi et al., 1995, Smola et al., 1998). Indeed, for a given function k, $\mathcal{F}$ is composed of functions $f \in L^2(\Re^p)$ such that:

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

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 with a Gaussian kernel $G_{\sigma/\sqrt{2}}(x) = \exp\left(-\frac{1}{\sigma^2}||x||^2\right)$. Note that, as σ increases from 0 to ∞, the functions $G_{\sigma/\sqrt{2}}$ range from an impulse to a constant function, and the spaces $\mathcal{F}_\sigma$ decrease from $L^2(\Re^p)$ 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.

[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.

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 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 3.2.1.

Theorem 1 *Let x_1 and x_2 be random variables in $\mathcal{X} = \Re^p$. Let K_1 and K_2 be Mercer kernels with feature maps Φ_1, Φ_2 and feature spaces $\mathcal{F}_1$, $\mathcal{F}_2 \subset \Re^{\mathcal{X}}$. 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. For Gaussian kernels in particular, the $\mathcal{F}$-correlation increases as σ decreases. But for any value of σ, the $\mathcal{F}$-correlation turns out to provide a sound basis for assessing independence, as the following theorem makes precise:

Theorem 2 (Independence and $\mathcal{F}$-correlation) *If $\mathcal{F}$ is the RKHS corresponding to a Gaussian kernel, $\rho_{\mathcal{F}} = 0$ 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:

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

which implies $\text{cov}(f_1(x_1), f_2(x_2)) = 0$, 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 $\omega_0 \in \Re$ 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 $\tau > \sigma/\sqrt{2}$, it belongs to $\mathcal{F}$ and we have, for all real ω_1 and ω_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 τ tend to infinity, we find that for all ω_1 and ω_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). ■

3.2 Kernelization of CCA

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.[2] (We discuss the optimization problem in Section 5).

In the case of two variables the goal is to maximize the correlation between projections of the data in the feature space. A naive implementation would simply map each data point to feature space and use CCA directly 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.

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 $\{\Phi_1(x_1^1), \ldots, \Phi_1(x_1^N)\}$ and $\{\Phi_2(x_2^1), \ldots, \Phi_2(x_2^N)\}$ denote the corresponding images in feature space. Suppose (momentarily) that the data are centered in feature space (i.e., $\sum_{k=1}^N \Phi_1(x_1^k) = \sum_{k=1}^N \Phi_2(x_2^k) = 0$). We let $\hat{\rho}_{\mathcal{F}}(x_1, x_2)$ denote the

[2] Melzer et al. (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).

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:

$$\widehat{\mathrm{cov}}(\langle \Phi_1(x_1), f_1 \rangle, \langle \Phi_2(x_2), f_2 \rangle) = \frac{1}{N} \sum_{k=1}^{N} \left\langle \Phi_1(x_1^k), f_1 \right\rangle \left\langle \Phi_2(x_2^k), f_2 \right\rangle . \tag{8}$$

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ represent the linear spaces spanned by the Φ-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:

$$\begin{aligned}
\widehat{\mathrm{cov}}(\langle \Phi_1(x_1), f_1 \rangle, \langle \Phi_2(x_2), f_2 \rangle) &= \frac{1}{N} \sum_{k=1}^{N} \left\langle \Phi_1(x_1^k), \sum_{i=1}^{N} \alpha_1^i \Phi(x_1^i) \right\rangle \left\langle \Phi_2(x_2^k), \sum_{j=1}^{N} \alpha_2^j \Phi(x_2^j) \right\rangle \\
&= \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_1^i K_1(x_1^i, x_1^k) K_2(x_2^j, x_2^k) \alpha_2^j \\
&= \frac{1}{N} \alpha_1^T K_1 K_2 \alpha_2,
\end{aligned}$$

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:

$$\widehat{\mathrm{var}}\left(\langle \Phi_1(x_1), f_1 \rangle \right) = \frac{1}{N} \alpha_1^T K_1 K_1 \alpha_1$$

and

$$\widehat{\mathrm{var}}\left(\langle \Phi_2(x_2), f_2 \rangle \right) = \frac{1}{N} \alpha_2^T K_2 K_2 \alpha_2 .$$

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

$$\hat{\rho}_{\mathcal{F}}(K_1, K_2) = \max_{\alpha_1,\ \alpha_2 \in \Re^N} \frac{\alpha_1^T K_1 K_2 \alpha_2}{(\alpha_1^T K_1^2 \alpha_1)^{1/2} (\alpha_2^T K_2^2 \alpha_2)^{1/2}} .$$

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:

$$\begin{pmatrix} 0 & K_1 K_2 \\ K_2 K_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \rho \begin{pmatrix} K_1^2 & 0 \\ 0 & K_2^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \tag{9}$$

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 matrix.[3] Whenever we use a Gram matrix, we assume that it has been centered in this way.

3.2.1 Generalizing to more than two variables

The generalization of 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 $\mathcal{K}$ the $mN \times mN$ matrix whose blocks are $\mathcal{K}_{ij} = K_i K_j$, and we let $\mathcal{D}$ denote the $mN \times mN$ block-diagonal matrix with blocks K_i^2. We obtain the following generalized eigenvalue problem:

$$\begin{pmatrix} K_1^2 & K_1K_2 & \cdots & K_1K_m \\ K_2K_1 & K_2^2 & \cdots & K_2K_m \\ \vdots & \vdots & & \vdots \\ K_mK_1 & K_mK_2 & \cdots & K_m^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} = \lambda \begin{pmatrix} K_1^2 & 0 & \cdots & 0 \\ 0 & K_2^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & K_m^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}, \tag{10}$$

or $\mathcal{K}\alpha = \lambda \mathcal{D}\alpha$ for short. The minimal eigenvalue of this problem will be denoted $\hat{\lambda}_{\mathcal{F}}(K_1, \ldots, K_m)$ and referred to as the *first kernel canonical correlation*. We also extend our earlier terminology and refer to this eigenvalue as an $\mathcal{F}$-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 x_1 and x_2, and we obtained an empirical contrast function $\hat{\rho}_{\mathcal{F}}(x_1, x_2)$ from $\rho_{\mathcal{F}}(x_1, x_2)$ by substituting empirical covariances for population covariances.[4] In the m-variable case, we have (thus far) defined only the empirical function $\hat{\lambda}_{\mathcal{F}}(K_1, \ldots, K_m)$. 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 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 $M_{\lambda_{\mathcal{F}}}(x_1, \ldots, x_m) = -\frac{1}{2} \log \lambda_{\mathcal{F}}(x_1, \ldots, x_m)$ 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 independent, thus mimicking a key property of the mutual information.

---

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

[4]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 $\hat{\rho}_{\mathcal{F}}(x_1, x_2)$ 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. Compute the centered Gram matrices $K_1, K_2, \ldots, K_m$ of the estimated sources $\{x^1, x^2, \ldots, x^N\}$, where $x^i = Wy^i$
3. Define $\hat{\lambda}_{\mathcal{F}}(K_1, \ldots, K_m)$ as the first eigenvalue of the generalized eigenvector equation $\mathcal{K}\alpha = \lambda\mathcal{D}\alpha$
4. Minimize $\hat{M}_{\lambda_{\mathcal{F}}}(K_1, \ldots, K_m) = -\frac{1}{2}\log\hat{\lambda}_{\mathcal{F}}(K_1, \ldots, K_m)$ with respect to W

Output: W

Figure 1: A high-level description of the KERNELICA-KCCA algorithm for estimating the parameter matrix W in the ICA model. Note that steps (2) through (4) are executed iteratively until a convergence criterion is met.

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

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 define a contrast function $\hat{M}_{\lambda_{\mathcal{F}}}(K_1, \ldots, K_m)$. 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.

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, $M(x_1, x_2)$, can be written as follows:

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

where ρ_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, $M(x_1, x_2, \ldots, x_m)$, 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:

$$M(x_1, x_2, \ldots, x_m) = -\frac{1}{2} \log \frac{\det C}{\det D} = -\frac{1}{2} \sum_{i=1}^{p} \log \lambda_i, \tag{11}$$

where λ_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. (11) 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 problem in Eq. (10), or equivalently the ratio of determinants of the matrices in this problem. That is, given the generalized eigenvector problem $\mathcal{K}\alpha = \lambda \mathcal{D}\alpha$, we define:

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

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

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

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 $\hat{M}_{\delta_{\mathcal{F}}}(K_1, \ldots, K_m)$ has as its population counterpart a function $M_{\delta_{\mathcal{F}}}(x_1, \ldots, x_m)$ that is actually closely related to the mutual information between the original non-Gaussian variables in the input space. This result, whose proof is sketched in Appendix B, holds for translation-invariant kernels of the form

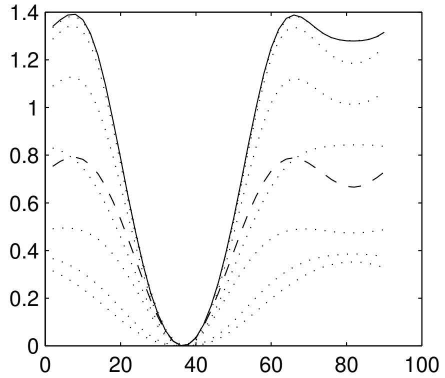


Figure 2: The mutual information (dashed), the approximation $M_{\delta_{\mathcal{F}}}(x_1, x_2)$ for $\sigma = .5, 1, 2, 4, 8, 16$ (dotted), and the limit $\mathcal{I}(x_1, x_2)$ as σ tends to zero (solid). The abscissa is the angle of the first independent component in a two-source ICA problem. As σ decreases, $M_{\delta_{\mathcal{F}}}$ increases towards $\mathcal{I}$. See the text for details.

$K_\sigma(x, y) = G((x-y)/\sigma)$. In particular, we show that, as σ tends to zero, $M_{\delta_{\mathcal{F}}}(x_1, \ldots, x_m)$ tends to a limit $\mathcal{I}(x_1, \ldots, x_m)$ that is independent of the particular choice of G, for G that satisfies positivity and decay conditions. Moreover, this limit is equal to the mutual information up to second order, when we expand the mutual information around distributions that are "nearly independent."

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 of the estimated first component ranges from 0 to 90 degrees, with the independent component occurring at 36.5 degrees. The graph plots the true mutual information, the approximation $M_{\delta_{\mathcal{F}}}(x_1, x_2)$, for various values of σ, and the limit $\mathcal{I}(x_1, x_2)$. The close match of the shape of $\mathcal{I}(x_1, x_2)$ and the mutual information is noteworthy.

3.5 The KernelICA-KGV algorithm

In the previous section, we defined an alternative contrast function, $\hat{M}_{\delta_{\mathcal{F}}}(x_1, \ldots, x_m)$, in terms of the generalized variance associated with the generalized eigenvector problem $\mathcal{K}\alpha = \lambda\mathcal{D}\alpha$. 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.

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 $\mathcal{K}\alpha = \lambda\mathcal{D}\alpha$. We now turn to a discussion of the computational methods by which we evaluate and optimize these contrast functions.

Algorithm KernelICA-KGV

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

1. Whiten the data
2. Compute the centered Gram matrices $K_1, K_2, \ldots, K_m$ of the estimated sources $\{x^1, x^2, \ldots, x^N\}$, where $x^i = Wy^i$
3. Define $\hat{\delta}_{\mathcal{F}}(K_1, \ldots, K_m) = \det \mathcal{K} / \det \mathcal{D}$
4. Minimize $\hat{M}_{\delta_{\mathcal{F}}}(K_1, \ldots, K_m) = -\frac{1}{2} \log \hat{\delta}_{\mathcal{F}}(K_1, \ldots, K_m)$ with respect to W

Output: W

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

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 (Williams and Seeger, 2001, Smola and Schölkopf, 2000). 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 η, 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, when this decay is exponential (Gaussian-like), we have $h(t) = O(\log t)$. When the decay is polynomial, i.e., x^{-d}, then $h(t) = O(t^{1/d+\varepsilon})$, for arbitrary $\varepsilon > 0$.

These results imply that if we require a constant precision η, 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, 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 separately show in Section 4.3, however, that in the context of $\mathcal{F}$-correlation and the KGV, the precision η 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), 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 (Williams and Seeger, 2001, Smola and Schölkopf, 2000). A difficulty with these methods in our context, however, is that they require the matrix being approximated to be a Gram matrix with a specified kernel (and not merely a general positive semidefinite matrix). This suffices for the computation of the ICA contrast functions, but cannot be used for the computation of the derivatives of the contrast functions, which involve matrices that cannot be expressed as Gram matrices.

Let us turn to a short description of incomplete Cholesky decomposition. A positive semidefinite matrix K can always be factored as GG^T, 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 dimension $N \times M$ matrix, for small M, such that the difference $K - \widetilde{G}\widetilde{G}^T$ has norm less than a given value η. 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, 1983). 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 $\widetilde{K}_l = \widetilde{G}_l\widetilde{G}_l^T$, 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 $K - \widetilde{G}_l\widetilde{G}_l^T$. 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 $O(M^2N)$.

The incomplete Cholesky method has many attractive features. Not only is its time complexity $O(M^2N)$, but also the only elements of K that are needed in memory are the diagonal elements (which are equal to one for Gaussian kernels[5]). 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.[6]

4.3 Regularization

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 that we must solve in more detail.

We must solve the generalized eigenvector problem $\mathcal{K}\alpha = \lambda\mathcal{D}\alpha$. The classical method for solving such a problem involves finding a matrix $\mathcal{C}$ such that $\mathcal{D} = \mathcal{C}^T\mathcal{C}$, defining $\beta = \mathcal{C}\alpha$, and thereby transforming the problem into a standard eigenvector problem $\mathcal{C}^{-T}\mathcal{K}\mathcal{C}^{-1}\beta = \lambda\beta$. Unfortunately, however, our matrix $\mathcal{D}$ is singular (due to the centering). We thus need to "regularize" $\mathcal{D}$, replacing $\mathcal{K}$ and $\mathcal{D}$ by $\mathcal{K}_\kappa$ and $\mathcal{D}_\kappa$ defined as:

$$\mathcal{K}_\kappa = \begin{pmatrix} (K_1+\kappa I)^2 & K_1K_2 & \cdots & K_1K_m \\ K_2K_1 & (K_2+\kappa I)^2 & \cdots & K_2K_m \\ \vdots & \vdots & & \vdots \\ K_mK_1 & K_mK_2 & \cdots & (K_m+\kappa I)^2 \end{pmatrix}$$

$$\mathcal{D}_\kappa = \begin{pmatrix} (K_1+\kappa I)^2 & 0 & \cdots & 0 \\ 0 & (K_2+\kappa I)^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & (K_2+\kappa I)^2 \end{pmatrix}$$

where κ is a small positive constant. The regularization parameter κ not only makes our problem well-posed numerically, but also provides control over the statistical properties of KernelICA, by controlling the capacity of the space of functions used in forming the contrast function. The constant κ is one of the two user-defined numerical parameters of our algorithm. (The other is σ, the width of the kernel. We discuss the setting of these parameters in Section 4.5.)

[5] Centering, which would make the diagonal elements different from one, can be done easily after the Cholesky decomposition.

[6] Note that no bound is available concerning the relation between M and the optimal 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 in Wright (1999), a significant eigengap ensures that incomplete Cholesky has small numerical error and yields a good approximation.

Algorithm IncompleteCholesky

Input: $N \times N$ matrix K
precision parameter η

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 $j \in \ [i+1, N]$, $G_{jj} = K_{jj} - \sum_{k=1}^{i} G_{jk}^2$
3. Output P, G and $M = i$

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

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.

Our kernelized CCA problem now reduces to finding the smallest eigenvalues of the matrix:

$$\widetilde{\mathcal{K}}_\kappa = \mathcal{D}_\kappa^{-1/2}\mathcal{K}_\kappa\mathcal{D}_\kappa^{-1/2} = \begin{pmatrix} I & r_\kappa(K_1)r_\kappa(K_2) & \cdots & r_\kappa(K_1)r_\kappa(K_m) \\ r_\kappa(K_2)r(K_1) & I & \cdots & r_\kappa(K_2)r_\kappa(K_m) \\ \vdots & \vdots & & \vdots \\ r_\kappa(K_m)r(K_1) & r_\kappa(K_m)r_\kappa(K_2) & \cdots & I \end{pmatrix} \tag{12}$$

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

For nonzero κ these regularized criteria are smooth even when the K_i lie in the space of singular semidefinite matrices, which is important because we want to use the criteria in gradient-based optimization algorithms.

Our regularization scheme has the effect of shrinking each eigenvalue of K_i towards zero, via the function $\lambda \mapsto \lambda/(\lambda+\kappa)$. Consequently, all eigenvalues less than a small fraction of κ (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 $\eta = 10^{-3}\kappa$.

In order to understand how to set κ with respect to N, it is useful to return to the setting of $\mathcal{F}$-correlation with two variables. In that case, $\hat{\rho}_\mathcal{F}$ is an estimator of the $\mathcal{F}$-correlation $\rho_\mathcal{F}(x_1,x_2)$, which is the maximal correlation between $f_1(x_1)$ and $f_2(x_2)$ for f_1, f_2 in the corresponding feature spaces. Using the notation of Section 3.2, we have: $\alpha_1^T(K_1+\kappa I)^2\alpha_1 = \alpha_1^T K_1^2\alpha_1 + 2\kappa\alpha_1^T K_1^2\alpha_1 + \kappa^2\alpha_1^T\alpha_1$. Ignoring second-order terms in κ, we have:

$$\alpha_1^T(K_1+\kappa I)^2\alpha_1 \approx N \operatorname{var} f_1(x_1) + 2\kappa||f_1||^2.$$

Consequently, using regularization, $\hat{\rho}_\mathcal{F}$ is an estimator of the following quantity:

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

In order to make our criteria asymptotically independent of the number of samples N, we thus let $\kappa = \kappa_0 N$, where κ_0 is a user-defined parameter. 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. (12).

Using the incomplete Cholesky decomposition, for each matrix K_i we obtain the factorization $K_i \approx G_i G_i^T$, 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 $N \times M_i$ orthogonal matrix U_i, and an $M_i \times M_i$ diagonal matrix Λ_i such that

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

Let $M = \frac{1}{n}\sum_{i=1}^{n} M_i$ 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:

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

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

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

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

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

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:

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

$$\mathcal{R}_\kappa = \begin{pmatrix} I & R_1 U_1^T U_2 R_2 & \cdots & R_1 U_1^T U_m R_m \\ R_2 U_2^T U_1 R_1 & I & \cdots & R_2 U_2^T U_m R_m \\ \vdots & \vdots & & \vdots \\ R_m U_m^T U_1 R_1 1 & R_m U_m^T U_2 R_2 & \cdots & I \end{pmatrix}.$$

The nN (non negative) eigenvalues of $\widetilde{\mathcal{K}}_\kappa$ sum to $\mathrm{tr}(\widetilde{\mathcal{K}}_\kappa) = nN$. 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 $\alpha \in \Re^{nN}$) is equal to the smallest eigenvalue of $\mathcal{R}_\kappa$ (with eigenvector $\beta \in \Re^{mM}$), and the two eigenvectors are related through:

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

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 $O(m^2 M^2)$. 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 (less than $O(s^3)$). Thus the complexity is greatly reduced: for $M = O(1)$, the complexities after decomposition are $O(m^2)$ and $O(\xi(n))$.

4.5 Free parameters

The KernelICA algorithms have two free parameters: the regularization parameter κ_0 and the covariance σ 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 let $\kappa_0 = 10^{-3}$, and to set $\sigma = 1/2$ for large samples ($N > 1000$) and $\sigma = 1$ for smaller samples ($N < 1000$).

For finite N, a value of σ 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 σ tends to zero, and this suggests choosing σ as small as possible. Still another consideration, however, is computational—for small σ the spectra of the Gram matrices decay less rapidly and the computational complexity grows. This can be mitigated by an appropriate choice of κ_0; in particular, the algorithm could choose κ_0 so that the number of retained eigenvalues for each Gram matrix is held constant. 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 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.

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 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.

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*.

5.1 The Stiefel manifold

The set of all $m \times m$ matrices W such that $W^T W = I$ is an instance of a *Stiefel manifold* (Edelman et al., 1999). Our optimization problem is thus the minimization of a function $F(W)$ 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 F is defined as

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

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

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

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

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

The ICA contrast functions do not have a single global optimum, and thus restarts are generally necessary to attempt 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 also found in our simulations that restarts could be avoided by initializing the optimization procedure at the solution found by simpler ICA algorithms (see Section 7).

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 $O(m^2 N)$—calculating the covariance matrix is $O(m^2 N)$, diagonalizing the $m \times m$ matrix is $O(m^3) = O(m^2 N)$, and scaling is $O(m^2 N)$

- 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 $O((mM)^2)$

- 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 finding its smallest eigenvalue is $O((mM)^3) = O(m^3NM^2)$

- Computation of the derivative of the first kernel canonical correlation is $O(m^2M^2N)$—at most $3m^2$ incomplete Cholesky decompositions to perform, and then matrix multiplications with lower complexity

- Computation of the derivative of the KGV is $O(m^3M^2N)$—at most $3m^2$ incomplete Cholesky decompositions to perform, and then matrix multiplications with maximal complexity $O(m^3M^2N)$

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.

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, 1998, Hyvärinen and Oja, 1997), the JADE algorithm (Cardoso, 1999), and the extended Infomax algorithm (Lee et al., 1999).

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:

$$d(V, W) = \frac{1}{2m} \sum_{i=1}^{m} \left(\frac{\sum_{j=1}^{n} |a_{ij}|}{\max_j |a_{ij}|} - 1 \right) + \frac{1}{2m} \sum_{j=1}^{m} \left(\frac{\sum_{i=1}^{n} |a_{ij}|}{\max_i |a_{ij}|} - 1 \right), \tag{13}$$

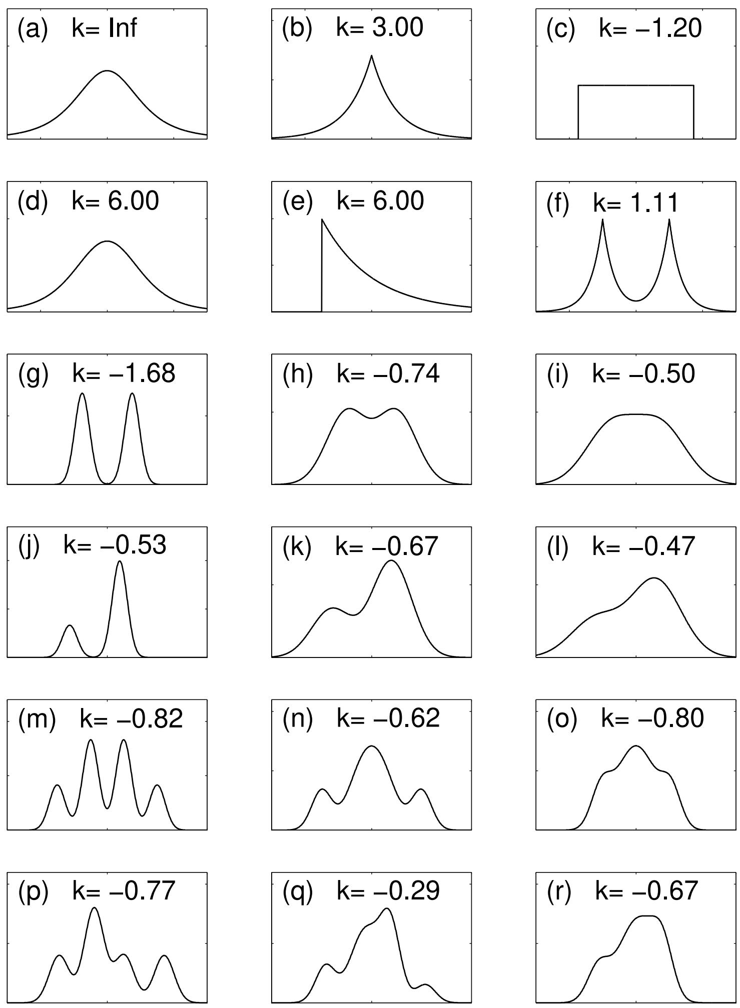


Figure 5: Probability density functions of sources with their kurtoses: (a) Student with 3 degrees of freedom; (b) double exponential; (c) uniform; (d) Student with 5 degrees of freedom; (e) exponential; (f) mixture of two double exponentials (g)-(h)-(i) symmetric mixtures of two Gaussians: multimodal,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

where $a_{ij} = (VW^{-1})_{ij}$. This metric, introduced in (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, 1998, Hyvärinen and Oja, 1997) uses a "deflation" scheme to calculate components sequentially. For each component a "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 three other 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 chose to do the same for the KernelICA algorithms, fixing the Gaussian kernel width to $\sigma = 1/2$ and the regularization parameter to $\kappa_0 = 10^{-3}$, except in the case of a small sample size, i.e., $N = 256$, where we used $\sigma = 1$.

7.3 Influence of source distributions

In a first series of experiments we tested the various algorithms on a two-component ICA problem, with 256 and 1024 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 18×100 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, ranging from 10% to 50%, over the other algorithms. Comparing just between the 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 n).

pdfs	Fica	Jade	Imax	Kcca	Kgv
a	8.5	7.6	56.6	7.6	6.5
b	10.7	8.7	61.4	7.7	6.4
c	4.8	3.6	15.7	5.3	4.7
d	3.5	2.7	22.1	2.8	2.7
e	12.7	8.4	33.3	10.2	9.0
f	23.8	18.1	45.4	22.0	20.0
g	18.3	14.5	55.4	3.1	2.9
h	12.1	8.6	36.6	5.9	4.8
i	22.7	16.1	42.0	12.3	9.5
j	12.1	10.9	61.3	14.1	11.4
k	9.3	7.7	69.4	3.8	3.4
l	6.8	5.1	22.2	3.5	3.0
m	8.0	5.8	31.1	13.1	9.9
n	12.9	9.3	37.3	15.0	8.8
o	10.0	6.1	25.1	11.4	9.1
p	9.7	7.0	26.7	7.0	5.5
q	41.7	32.8	44.3	10.6	9.2
r	14.0	9.4	37.3	8.2	6.9
mean	**13.4**	**10.1**	**40.2**	**9.1**	**7.4**
rand	**10.9**	**9.6**	**27.6**	**7.3**	**6.0**

pdfs	Fica	Jade	Imax	Kcca	Kgv
a	4.6	4.4	3.1	4.7	3.3
b	5.9	5.1	3.8	4.3	2.7
c	2.4	1.6	2.0	2.7	1.7
d	1.9	1.4	1.4	1.5	1.4
e	5.2	4.0	3.3	4.5	3.4
f	10.7	7.1	6.8	10.2	9.6
g	7.8	6.0	54.9	1.5	1.4
h	6.2	4.2	3.8	3.0	2.6
i	10.7	8.1	11.4	5.0	4.4
j	5.9	5.0	7.1	7.7	6.0
k	5.4	4.2	4.5	1.7	1.4
l	3.3	2.6	1.5	1.5	1.3
m	4.0	2.7	4.4	2.3	1.3
n	5.5	4.0	28.9	2.9	1.8
o	4.1	2.9	3.9	5.0	3.3
p	3.7	2.8	10.3	2.3	1.8
q	17.4	12.4	41.1	3.9	2.6
r	6.2	4.6	5.0	4.2	3.1
mean	**6.2**	**4.6**	**11.0**	**3.8**	**2.9**
rand	**6.4**	**4.7**	**10.5**	**3.2**	**2.3**

Table 1: The Amari errors (multiplied by 100) for two-component ICA with 256 samples (left) and 1024 samples (right). For each pdf (from a to r), averages over 20 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	Fica	Jade	Imax	Kcca	Kgv
2	250	1000	11	9	28	7	6
	1000	1000	6	5	11	3	2
4	1000	100	19	15	35	13	9
	4000	20	9	6	22	5	3
8	1000	20	34	39	87	31	32
	4000	20	19	16	51	12	7
16	4000	10	40	36	99	22	13

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.

7.4 Increasing number of components

In a second series of experiments, we tested the algorithms in simulations with 4, 8 and 16 components. Source distributions were chosen at random from the 18 possible sources in Figure 5. The KernelICA algorithms were initialized using the outputs of other algorithms and no restarts were performed. We also performed experiments using random initialization and random restarts and obtained essentially identical results.

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 a two-component ICA problem using two types of symmetric distributions that were nearly Gaussian. The first type was supergaussian with small positive kurtosis, obtained by using a mixtures of two Gaussians with identical variance and nearly identical means. The second type was subgaussian with small negative kurtosis, and were obtained by using a mixtures of Gaussians with identical means and nearly identical variances.

Figure 6 shows the performance of the algorithms as the kurtosis approaches zero, from above and from below. We see that the KernelICA algorithms are more robust to near-Gaussianity than the other algorithms.

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).

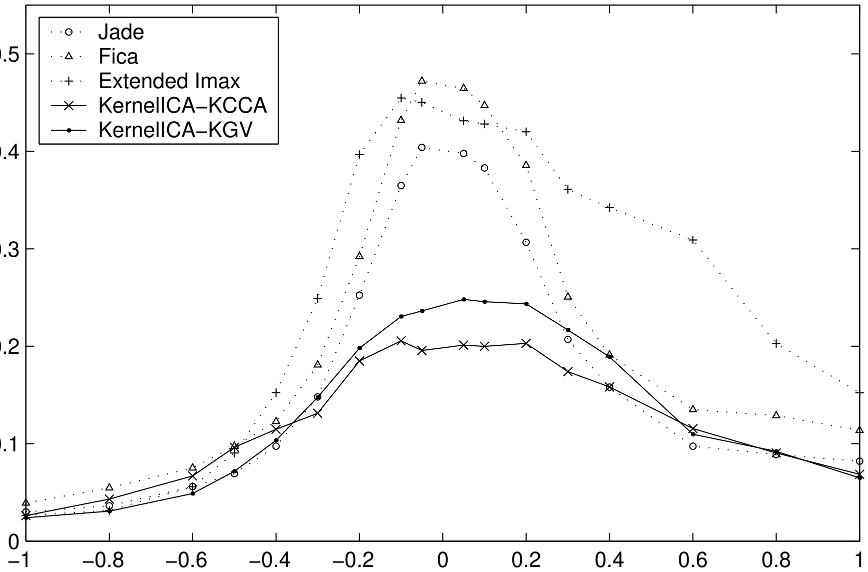


Figure 6: Robustness to near-Gaussianity. The solid lines plot the error of the KernelICA algorithms as the kurtosis approaches zero. The dotted lines plot the performance of the other three algorithms.

We simulated outliers by randomly choosing up to thirty 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 0.1 seconds to evaluate our contrast functions, and 1 second to evaluate their derivatives (using Matlab with a Pentium 500 MHz processor). Moreover, the expected scaling of $O(m^2N)$ 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

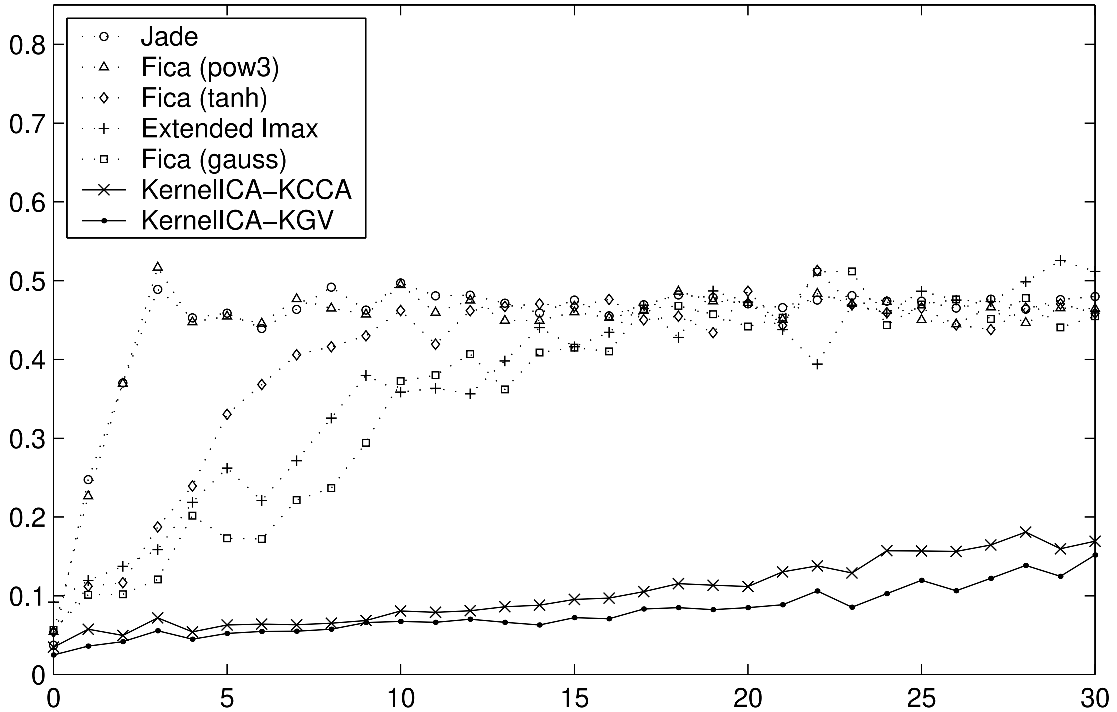


Figure 7: Robustness to outliers. The abscissa displays the number of outliers and the ordinate shows the Amari error.

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.

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.

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, kernels can be defined on data that are not necessarily numerical (e.g., the "string kernels" of Lodhi et al., 2001), 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, the optimization procedures that we have discussed in this paper can be extended and improved in several ways. For example, so-called “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 to tame problems of local minima. In the KernelICA setting, the KCCA or KGV between one univariate component and its orthogonal subspace provides a natural one-unit contrast function. Note also that methods exist to combine components obtained from one-unit contrasts into a full ICA solution. These include “deflationary orthogonalization,” in which components are found one by one and orthogonality is ensured by successive Gram-Schmidt projections, and “symmetric orthogonalization,” in which components are found in parallel and orthogonality is ensured by symmetric orthogonalization $(WW^T)^{-1/2}W$ of the demixing matrix W. Finally, another optimization technique that has been used in the context of ICA is Jacobi rotation (Cardoso, 1999), which can be used to perform optimization of any function on the group of orthogonal matrices. All of these techniques are applicable in principle to our approach.

Finally, a more thoroughgoing study of the statistical properties of KernelICA are 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 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 σ and κ_0 in our algorithm.

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):

$$M(x_1, x_2) = -\frac{1}{2} \log \left(\frac{\det C}{\det C_{11} \det C_{22}} \right). \tag{14}$$

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:

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = (1+\rho) \begin{pmatrix} C_{11} & 0 \\ 0 & C_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}. \tag{15}$$

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. (14) is equal to the product of the generalized eigenvalues of Eq. (15). This yields:

$$M(x_1, x_2) = -\frac{1}{2} \log \prod_{i=1}^{p} (1 - \rho_i)(1 + \rho_i) = -\frac{1}{2} \sum_{i=1}^{p} \log(1 - \rho_i^2). \tag{16}$$

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

While Eq. (16) 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. (16) 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:*

$$M_\rho(x_1, x_2) \leqslant M(x_1, x_2) \leqslant \min\{p_1, p_2\} M_\rho(x_1, x_2). \tag{17}$$

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):

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

We again refer to the ratio appearing in this expression, $\det C / (\det C_{11} \cdots \det C_{nn})$, 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:

$$\begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & 0 & \cdots & 0 \\ 0 & C_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & C_{nn} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{pmatrix}, \tag{19}$$

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:

$$M(x_1, \ldots, x_m) = -\frac{1}{2} \sum_{i=1}^{P} \log \lambda_i, \tag{20}$$

where λ_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.[7]

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. (20). 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 $M = M(x_1, \ldots, x_m)$:*

$$M_\lambda + \frac{\lambda - 1}{2} \leqslant M_\lambda - \frac{P-1}{2} \log \frac{P - \lambda}{P - 1} \leqslant M \leqslant P M_\lambda, \tag{21}$$

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

[7] Note that the λ_i are all nonnegative and sum to $P = \sum_i p_i$. Note also that the λ_i do not occur in pairs as they do in the two-variable case. Moreover, the terms in the sum in Eq. (20) are not all positive.

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.

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 $\text{corr}(y_i, y_j)$. We define $\nu(y_1, \ldots, y_m)$ to be the minimal eigenvalue of $\widetilde{C}$. Note that a correlation matrix is symmetric 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 ν, 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. (19). 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 $x_1, \ldots, x_m$:*

$$\lambda(x_1, \ldots, x_m) = \min_{\xi_1, \ldots, \xi_m} \nu(\xi_1^T x_1, \ldots, \xi_m^T x_m). \tag{22}$$

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

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

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

Minimizing over all possible $\xi_1, \ldots, \xi_m$ and β of unit norm is the same as minimizing over all possible ζ_i such that $\sum_{i=1}^{n} ||\zeta_i||^2 = 1$. Consequently, by assembling all ζ_i's in one vector ζ, necessarily of unit norm, we have:

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

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 $\nu(\xi_1^T x_1, \ldots, \xi_m^T x_m)$ is constant equal to one, which implies by Eq. (22) that $\lambda(x_1, \ldots, x_m) = 1$. Conversely, if $\lambda(x_1, \ldots, x_m) = 1$, then since ν is always between zero and one, using Eq. (22), for all ξ_i, $\nu(\xi_1^T x_1, \ldots, \xi_m^T x_m)$ must be equal to one, and consequently, the univariate random variables $\xi_1^T x_1, \ldots, \xi_m^T x_m$ 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 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 λ. Since φ 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 independent. As in the KCCA case, the converse is trivially true. Thus, the KGV also defines a valid contrast function.

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 particular, we claim that the KGV approaches a limit as the kernel width approaches zero, and that 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 restrict ourselves to two univariate random variables and Gaussian kernels for simplicity.

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, $X \in \Re^p$ and $Y \in \Re^q$, 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. The covariance structure of (X, Y) can be written as follows, where $D_{p_x} = \text{diag}(p_x)$ and $D_{p_y} = \text{diag}(p_y)$:

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

which implies $C_{XY} = P - p_x p_y^T$, $C_{XX} = D_{p_x} - p_x p_x^T$, and $C_{YY} = D_{p_y} - p_y p_y^T$. 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 $\mathcal{I}$, is equal to $\mathcal{I} = -\frac{1}{2}\log\det(I - C^TC) = -\frac{1}{2}\log\det(I - CC^T)$, where $C = D_{p_x}^{-1/2}(P - p_xp_y^T)D_{p_y}^{-1/2}$.

Near independence, that is, if we assume $P_{ij} = p_{xi}p_{yj}(1+\varepsilon_{ij})$ where ε is a matrix with small norm, we can expand $\mathcal{I}$ as follows:

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

where $C = D_{p_x}^{-1/2}(P - p_xp_y^T)D_{p_y}^{-1/2} = D_{p_x}^{1/2}\varepsilon D_{p_y}^{1/2}$. Thus, we have:

$$\mathcal{I} \approx \frac{1}{2}\operatorname{tr}(D_{p_x}\varepsilon D_{p_y}\varepsilon^T) = \frac{1}{2}\sum_{ij}\varepsilon_{ij}^2 p_{xi}p_{yj}. \tag{23}$$

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

$$M = \sum_{ij} p_{xi}p_{yj}(1+\varepsilon_{ij})\log(1+\varepsilon_{ij}) \approx \sum_{ij} p_{xi}p_{yj}(\varepsilon_{ij} + \frac{1}{2}\varepsilon_{ij}^2) \approx \frac{1}{2}\sum_{ij}\varepsilon_{ij}^2 p_{xi}p_{yj}, \tag{24}$$

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 $\mathcal{I}(x,y)$ 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, $M(x,y)$, when we expand "near independence." We now extend these results, defining the quantity $\mathcal{I}$, 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, $M(x,y) = \int p(x,y)\log[p(x,y)/p(x)p(y)]dxdy$, 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 $\mathcal{I}(x,y)$ is defined to be the supremum of $\mathcal{I}(x_d,y_d)$ 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 $\mathcal{I}$ 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.

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 $x^i \in \Re$. Let $\{x^i\}$ be a mesh of uniformly distributed points in $\Re$ 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 $M_{\delta_\mathcal{F}}(\sigma)$ 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 $M_{\delta_\mathcal{F}}(h,\sigma)$ 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 $M_{\delta_\mathcal{F}}(h,\sigma)$ tends to $M_{\delta_\mathcal{F}}(\sigma)$.[8] We now relate the quantity $M_{\delta_\mathcal{F}}(h,\sigma)$ to the Gaussian mutual information. 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_σ}, a smoothed version of p, is well defined as a probability density function because G_σ 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:

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

These identities ensure that, as h and σ 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_σ}. This implies that, as h and σ tends to zero, $M_{\delta_\mathcal{F}}(h,\sigma)$ is equivalent to the Gaussian mutual information of the variables x and y, smoothed by G_σ. Moreover, as the smoothing parameter σ tends to zero, p_{G_σ} tends to p, and we see that $M_{\delta_\mathcal{F}}(\sigma)$ tends to $\mathcal{I}$. Thus the KGV tends to the Gaussian mutual information, as claimed in Section 3.4.

C Spectrum of Gram matrices

The computational efficiency of our algorithms relies on the approximation of Gram matrices by matrices of very low rank.[9] In this section we present theoretical results from functional

[8] This limiting process can be made rigorous, but doing so is outside of the scope of the paper.

[9] Note that a (non-centered) Gram matrix is always invertible, given distinct sample points and a Gaussian kernel, so any low-rank representation of such a matrix is necessarily an approximation.

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

$$T : \phi(y) \mapsto \int_{\Re^d} K(x, y)p(x)\phi(x)dx. \tag{25}$$

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 λ_i for which there exists $\phi_i \in L^2(\Re^d)$ such that $T\phi_i = \lambda_i\phi_i$:

$$\int_{\Re^d} K(x, y)p(x)\phi_i(x)dx = \lambda_i\phi_i(y). \tag{26}$$

The eigenvalues λ_i and eigenvectors ϕ_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. (25) 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. (26) and evaluating at $y = x_j$, we get:

$$\frac{1}{N}\sum_{k=1}^{N} K(x_k, x_j)\phi_i(x_k) \approx \lambda_i\phi_i(x_j), \tag{27}$$

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

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

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

Decay of $p(x)$	Bound of $n(\eta)$	Decay of λ_n
compact support	$o(\log(1/\eta))$	$e^{-An\log n}$
$e^{-x^2/2}$	$\log(1/\eta)$	e^{-An}
$\lvert x\rvert^{-d},\ d>2$	$\eta^{-\varepsilon-1/d}$	$n^{-d+\varepsilon}$

Table 3: Bounds for the number $N(\eta)$ of eigenvalues greater than η, and the n-th eigenvalue λ_n of the integral operator T

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. (25) 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 spectra 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.[10] 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 the $N\times N$ Gram matrix, which we denote as $\lambda_{k,N}$, is composed 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 λ_k/N, where λ_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 $M = h(\eta/N)$, where as η/N tends to zero, $h(t)$ grows

[10] 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 $|x|^{-4}$.

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.

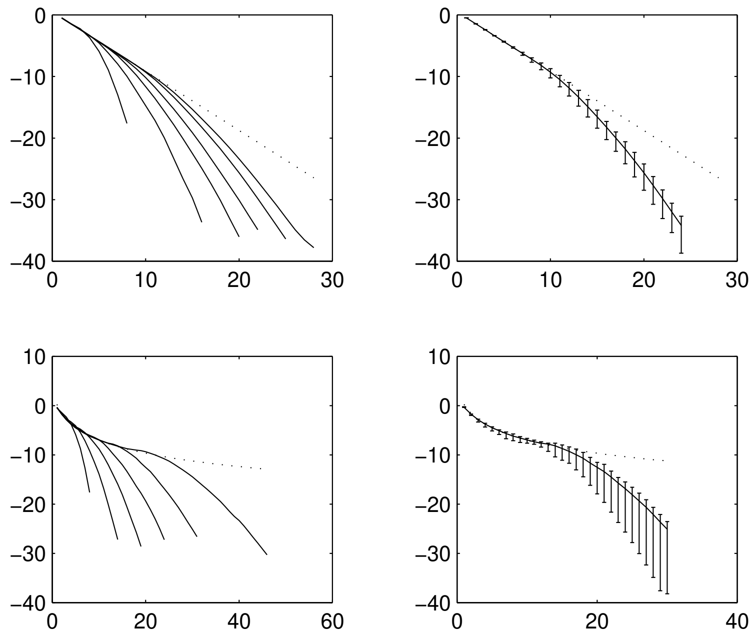


Figure 8: Spectra for two different input densities (top: Gaussian, bottom: Student distribution with three degrees of freedom). The dashed lines are the exact or fitted (see text for details) logarithm of the spectra $\log \lambda_k$, plotted as a function of the eigenvalue number k. (Left) The solid 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.

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 $w \in \Re^n$ 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 $\exp\left\{-\frac{1}{2\sigma^2}(w^T x_a - w^T x_b)^2\right\}$. Without loss of generality, let us differentiate this expression around $w = (1, 0, \ldots, 0)^T$. We obtain:

$$\left(\partial_{w_j} K\right)_{ab} = -\frac{1}{2\sigma^2}(x_{a1} - x_{b1})(x_{aj} - x_{bj})e^{-\frac{1}{2\sigma^2}(x_{a1}-x_{b1})^2}. \tag{28}$$

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. (28), 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:

$$\begin{aligned}
\hat{f}(\omega) &= -\frac{d^2}{d\omega^2}(\nu(\omega)) = -\frac{d^2}{d\omega^2}(\sqrt{2\pi}\sigma e^{-\omega^2\sigma^2/2}) \\
&= \sigma^2(1 - \sigma^2\omega^2)\sqrt{2\pi}\sigma e^{-\omega^2\sigma^2/2} \\
&= \sigma^2\nu(\omega) - \sigma^4\omega^2\sqrt{2\pi}\sigma e^{-\omega^2\sigma^2/2} \\
&= \sigma^2\nu(\omega) - \hat{h}(\omega)
\end{aligned} \tag{29}$$

  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. (28), 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 γ, we write:

$$xy = \frac{1}{2}(\sigma^2 - x^2) + \frac{1}{2}(\gamma^2 - y^2) - \frac{1}{2}(\sigma^2 + \gamma^2 - (x+y)^2). \tag{30}$$

  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 $f_\gamma(x,y) = g_\sigma(x,y) + g_\gamma(x,y) - h(x,y)$ where g_σ, g_γ and h all have real positive Fourier transforms. To approximate f based on f_γ, we note that:

$$|f(x,y) - f_\gamma(x,y)| \leqslant |xy|e^{-x^2/2\sigma^2}y^2/2\gamma^2. \tag{31}$$

  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 γ (we used $\gamma = 50$ in our simulations) and obtain satisfactory results.[11]

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

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 KCCA criterion and $O(m^3M^2N)$ for the KGV criterion.

Acknowledgments

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

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

Amari, S., Cichocki, A., and Yang, H. H. (1996). A new learning algorithm for blind signal separation. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 757–763, Cambridge, MA. MIT Press.

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

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

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

Borga, M., Knutsson, H., and Landelius, T. (1997). Learning canonical correlations. In *Proceedings of the 10th Scandinavian Conference on Image Analysis*, Lappeenranta, Finland.

Brezis, H. (1980). *Analyse Fonctionelle.* Masson, Paris, France.

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

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

Comon, P. (1994). Independent Component Analysis, a new concept? *Signal Processing*, 36(3):287–314. Special issue on Higher-Order Statistics.

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

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

Edelman, A., Arias, T. A., and Smith, S. T. (1999). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353.

Fine, S. and Scheinberg, K. (2001). Efficient SVM training using low-rank kernel representation. Technical Report RC 21911, IBM T. J. Watson Research Center.

Fyfe, C. and Lai, P. L. (2000). ICA using kernel canonical correlation analysis. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, pages 279–284.

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

Golub, G. H. and Loan, C. F. V. (1983). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD.

Harmeling, S., Ziehe, A., Kawanabe, M., and Müller, K.-R. (2002). Kernel feature spaces and nonlinear blind source separation. In *Advances in Neural Information Processing Systems*, volume 14, Cambridge, MA. MIT Press.

Hotelling, H. (1936). Relation between two sets of variates. *Biometrika*, 28:322–377.

Hyvärinen, A. (1998). The FastICA MATLAB toolbox. Available at http://www.cis.hut./projects/ica/fastica/.

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

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

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

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

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

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

Kullback, S. (1959). *Information Theory and Statistics*. John Wiley, New York, NY.

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

Lodhi, H., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2001). Text classification using string kernels. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA. MIT Press.

Melzer, T., Reiter, M., and Bischof, H. (2001). Nonlinear feature extraction using generalized canonical correlation analysis. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2001)*, pages 350–357.

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

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

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

Smola, A. J. and Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In *Proceedings of 17th International Conference on Machine Learning (ICML)*, pages 911–918. Morgan Kaufmann, San Francisco, CA.

Smola, A. J., Schölkopf, B., and Müller, K.-R. (1998). The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649.

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

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

Williams, C. K. I. and Seeger, M. (2000). Effect of the input density distribution on kernel-based classifiers. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*.

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

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