Randomized Nonlinear Component Analysis

David Lopez-Paz\textsuperscript{1,2}  
Suvrit Sra\textsuperscript{1,3}  
Alexander J. Smola\textsuperscript{3,4}  
Zoubin Ghahramani\textsuperscript{2}  
Bernhard Schölkopf\textsuperscript{1}

\textsuperscript{1}Max Planck Institute for Intelligent Systems  
\textsuperscript{2}University of Cambridge  
\textsuperscript{3}Carnegie Mellon University  
\textsuperscript{4}Google

Abstract

Classical techniques such as Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) are ubiquitous in statistics. However, these techniques only reveal linear relationships in data. Although nonlinear variants of PCA and CCA have been proposed, they are computationally prohibitive in the large scale.

In a separate strand of recent research, randomized methods have been proposed to construct features that help reveal nonlinear patterns in data. For basic tasks such as regression or classification, random features exhibit little or no loss in performance, while achieving dramatic savings in computational requirements.

In this paper we leverage randomness to design scalable new variants of nonlinear PCA and CCA; our ideas also extend to key multivariate analysis tools such as spectral clustering or LDA. We demonstrate our algorithms through experiments on real-world data, on which we compare against the state-of-the-art. Code in R implementing our methods is provided in the Appendix.

1. Introduction

Principal Component Analysis (Pearson, 1901) and Canonical Correlation Analysis (Hotelling, 1936) are two of the most popular multivariate analysis methods. They have played a crucial role in a vast array of applications since their conception a century ago.

Principal Component Analysis (PCA) rotates a collection of correlated variables into their uncorrelated principal components (also known as factors or latent variables). Principal components owe their name to the following key property: the first principal component captures the maximum amount of variance in the data; successive components account for the maximum amount of remaining variance in dimensions orthogonal to the preceding ones. PCA is commonly used for dimensionality reduction, assuming that core properties of a high-dimensional sample are largely captured by a small number of principal components.

Canonical Correlation Analysis (CCA) computes linear projections of a pair of random variables such that their projections are maximally correlated. Analogous to principal components, the projections of the pair of random variables are mutually orthogonal and ordered by their amount of explained cross-correlation. CCA is widely used to learn from multiple modalities of data (Kakade & Foster, 2007), an ability particularly useful when some of the modalities are only available at training time, but keeping information about them at testing time is beneficial (Chaudhuri et al., 2009; Vapnik & Vashist, 2009).

The applications of PCA and CCA are ubiquitous. Some examples are feature extraction, time-series prediction, finance, medicine, meteorology, chemometrics, biology, neurology, natural language processing, speech recognition, computer vision or multimodal signal processing (Jolliffe, 2002).

Despite their success, an impediment of PCA and CCA to modern data analysis is that both reveal only linear relationships between the variables under study. To overcome this limitation, for both PCA and CCA several nonlinear extensions have been proposed. For PCA, these include Kernel Principal Component Analysis or KPCA (Schölkopf et al., 1999) and Autoencoder Neural Networks (Baldi & Hornik, 1989; Hinton & Salakhutdinov, 2006). For CCA, common extensions are Kernel Canonical Correlation Analysis or KCCA (Lai & Fyfe, 2000; Melzer et al., 2001; Bach & Jordan, 2002) and Deep Canonical Correlation Analysis (Andrew et al., 2013). However, these solutions tend to have...
Rather high computational complexity (often cubic in the sample size), are difficult to parallelize, and are not always accompanied by theoretical guarantees.

In a separate strand of recent research, randomized strategies have been introduced for constructing features that can help reveal nonlinear patterns in data when used in conjunction with linear algorithms (Rahimi & Recht, 2008; Le et al., 2013). For basic tasks such as regression or classification, using nonlinear random features incurs little or no loss in performance compared with exact kernel methods, while achieving dramatic savings in computational complexity (from cubic to linear in the sample size). Furthermore, random features are amenable to simple implementation and clean theoretical analysis.

The main contribution of this paper is to lay the foundations for nonlinear, randomized variants of PCA and CCA. Therefore, we dedicate key attention to studying the spectral properties of low-rank kernel matrices constructed as sums of random feature dot-products. Our analysis is powered by the recently developed matrix Bernstein inequality (Mackey et al., 2012). With little additional effort, our analysis extends to other popular multivariate analysis tools such as linear discriminant analysis, spectral clustering, or the randomized dependence coefficient.

We demonstrate the effectiveness of the proposed randomized methods by experimenting with several real-world data and comparing against the state-of-the-art on CCA: Deep Canonical Correlation Analysis (Andrew et al., 2013). As a novel application of the presented methods, we derive an implementation of the presented algorithms in just 15 lines of R code.

1.1. Related Work

There has been a recent stream of research in kernel approximations via randomized feature maps as the seminal work of Rahimi & Recht (2008). For instance, their extensions to dot-product kernels (Kar & Karnick, 2012) and polynomial kernels (Hamid et al., 2013); the development of advanced sampling techniques using Quasi-Monte-Carlo methods (Yang et al., 2014) or their accelerated computation via fast Walsh-Hadamard transforms (Le et al., 2013).

The use of randomized techniques for kernelized component analysis methods dates back to (Achlioptas et al., 2002), where three kernel sub-sampling strategies were suggested to speed up KPCA. On the other hand, (Avron et al., 2013) made use of randomized Walsh-Hadamard transforms to adapt linear CCA to large-scale datasets. The use of nonlinear random features is more scarce and has only appeared twice in previous literature. First, McWilliams et al. (2013) use the Nyström method to define a randomized feature map and perform CCA to achieve fast state-of-the-art semi-supervised learning. Second, Lopez-Paz et al. (2013) define the dependence statistic RDC as the largest canonical correlation between two sets of copula random projections.

2. Random Nonlinear Features

Before presenting our new methods, let us first recall a few key aspects of nonlinear random features.

Consider the class of functions

\[ F_p := \left\{ f(x) = \int_{\mathbb{R}^d} \alpha(w) \phi(x^T w)dw : |\alpha(w)| \leq C p(w) \right\}, \]

over which we wish to learn. Here, \( \alpha : \mathbb{R}^d \rightarrow \mathbb{R} \) is a nonlinear map of “weights”, while \( \phi : \mathbb{R} \rightarrow \mathbb{R} \) is a nonlinear map that satisfies \( |\phi(z)| \leq 1; x, w \) are vectors in \( \mathbb{R}^d \), \( p(w) \) is a probability density of the parameter vectors \( w \) and \( C \) is a regularizing constant. More simply, we may consider the finite version of \( f \):

\[ f_m(x) := \sum_{i=1}^m \alpha_i \phi(w_i^T x). \]

Kernel machines, Gaussian processes, AdaBoost, and neural networks are models that fit within this function class.

Let \( D = \{ (x_i, y_i) \}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R} \) be a finite sample of input-output pairs drawn from a distribution \( Q(X,Y) \). We seek to approximate a function \( f \) in class \( F_p \) by minimizing the empirical risk (over dataset \( D )\)

\[ R_{\text{emp}}(f) := \frac{1}{m} \sum_{i=1}^m c(f_m(x_i), y_i), \]

for a suitable loss function \( c(\hat{y}, y) \) that penalizes departure of \( f_m(x) \) from the true label \( y \); for us, the least-squares loss \( c(\hat{y}, y) = (\hat{y} - y)^2 \) is most convenient.

Jointly optimizing \( (3) \) over \( (\alpha, w_1, \ldots, w_m) \) used in defining \( f_m \), is a daunting task given the nonlinear nature of \( \phi \). The key insight of Rahimi & Recht (2008) is that we can instead randomly sample the parameters \( w_i \in \mathbb{R}^d \) from a data-independent distribution \( p(w) \) and construct an \( m \)-dimensional randomized feature map \( z(X) \) for the input data \( X \in \mathbb{R}^{n \times d} \) that obeys the following structure:

\[ w_1, \ldots, w_m \sim p(w), \]

\[ z_i := [\cos(w_i^T x_1), \ldots, \cos(w_i^T x_n)] \in \mathbb{R}^n, \]

\[ z(X) := [z_1 \ldots z_m] \in \mathbb{R}^{n \times m}. \]

Using the (precomputed) nonlinear random features \( z(X) \) ultimately transforms the nonconvex optimization of \( (3) \) into a least-squares problem of the form

\[ \min_{\alpha \in \mathbb{R}^d} \|y - z(X)\alpha\|_2^2, \quad \text{s.t.} \quad \|\alpha\|_\infty \leq C. \]
This form remarkably simplifies computation (in practice, we actually solve a regularized version of it), while incurring only a bounded error. Theorem 1 formalizes this claim.

**Theorem 1.** (Rahimi & Recht, 2008) Let $F_p$ be defined as in (1). Draw $D \sim Q(X, Y)$. Construct $z(\cdot)$ as in (4). Let $c : \mathbb{R}^2 \to \mathbb{R}_+$ be a loss-function $L$-Lipschitz in its first argument, then, for any $\delta > 0$, with probability at least $1 - 2\delta$ there exist some $\alpha = (\alpha_1, \ldots, \alpha_m)$ such that

$$\mathbb{E}_D [c(z(x)\alpha, y)] - \min_{f \in F_p} \mathbb{E}_D [c(f(x), y)] \leq O \left( \left( \frac{LC}{\sqrt{m}} + \frac{LC}{\sqrt{m}} \right) \sqrt{\log \frac{1}{\delta}} \right).$$

Solving (5) takes $O(ndm + m^2n)$ operations, while testing $t$ points on the fitted model takes $O(tdm)$ operations. Recent techniques that use subsampled Hadamard randomized transforms (Le et al., 2013) allow computation of the random features even faster, yielding $O(n \log(d)m + m^2n)$ operations to solve (5) and $O(t \log(d)m)$ to test $t$ new points.

It is of special interest that randomized algorithms are in many cases more robust than their deterministic analogues (Mahoney, 2011). However, the nature and quantification of the implicit regularization induced by randomness remains an open question.

### 2.1. Random Features and Kernel Matrices

Bochner’s theorem helps connect shift-invariant kernels (Schölkopf & Smola, 2002) and random nonlinear features. Let $k(x, y)$ be a real valued, normalized ($k(x, y) \leq 1$), shift-invariant kernel on $\mathbb{R}^d \times \mathbb{R}^d$. Then,

$$k(x, y) = \int_{\mathbb{R}^d} p(w)e^{-i w^T(x-y)}dw \approx \sum_{i=1}^m \frac{1}{\sqrt{m}} e^{-i w_i^T x} e^{i w_i^T y} = \sum_{i=1}^m \frac{1}{\sqrt{m}} \cos(w_i^T x) \cos(w_i^T y) = \langle \frac{1}{\sqrt{m}} z(x), \frac{1}{\sqrt{m}} z(y) \rangle,$$

where $z(x)$ is a random feature map as in (4) with $p(w)$ set to be the inverse Fourier transform of $k$ (Rahimi & Recht, 2008)—e.g., the Gaussian kernel $k(x, y) = \exp(-s\|x - y\|^2_2)$ can be approximated using $w_i \sim \mathcal{N}(0, 2sI)$.

For a data matrix $X \in \mathbb{R}^{n \times d}$ let $K \in \mathbb{R}^{n \times n}$ be its kernel matrix, i.e., $K_{ij} = k(x_i, x_j)$. When approximating the feature map of a kernel $k$ using random features, we may as well approximate the kernel matrix $K \approx \hat{K}$, where

$$\hat{K} := \frac{1}{m} z(X)z(X)^T = \frac{1}{m} \sum_{i=1}^m z_i z_i^T = \sum_{i=1}^m \hat{K}^{(i)}.$$ (6)

The focus of this paper is on building scalable kernel component analysis methods which not only exploit these approximations but are also accompanied by theoretical guarantees.

### 3. Principal Component Analysis (PCA)

Principal Component Analysis or PCA (Pearson, 1901; Jolliffe, 2002) is the orthogonal transformation of a set of $n$ observations of $d$ correlated variables $X \in \mathbb{R}^{n \times d}$ into a set of $n$ observations of $d$ uncorrelated principal components.

For a centered data matrix (zero mean columns) $X$, PCA requires computing the (full) singular value decomposition

$$X = U\Sigma F^T,$$

where $\Sigma$ is a diagonal matrix containing the singular values of $X$ in decreasing order. The principal components are computed via the linear transformation $XF$.

**Nonlinear** variants of PCA are also known; notably,

- Kernel PCA (Schölkopf et al., 1999) uses the kernel trick to embed data into a high-dimension Reproducing Kernel Hilbert Space, where regular PCA is performed. Computation of the principal components reduces to an eigenvalue problem, which takes $O(n^3)$ operations.

- Autoencoders (Hinton & Salakhutdinov, 2006) are artificial neural networks configured to learn their own input. They are trained to learn compressed representations of data. The transformation computed by a linear autoencoder with a bottleneck of size $r < d$ is the projection into the subspace spanned by the first $r$ principal components of the training data (Baldi & Hornik, 1989).

### 3.1. Randomized Nonlinear PCA (RPCA)

We propose RPCA, a nonlinear randomized variant of PCA. We may view RPCA as a low-rank approximation of KPCA when the latter is equipped with a shift-invariant kernel.

RPCA may be thus understood as (linear) PCA on a nonlinear mapping of the data. Schematically,

$$\{F, z(\cdot)\} =: \text{RPCA}(X) \equiv \text{PCA}(z(X)) \approx \text{KPCA}(X),$$

where $F \in \mathbb{R}^{m \times m}$ are the principal component loading vectors and $z : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times m}$ is a random feature map generated as in (4) (typically, $m \ll n$).

The computational complexity is $O(d^2n)$ for PCA and $O(m^2n)$ for RPCA; both are linear in the sample size $n$, and when $m \ll d$, the latter is much cheaper. When using nonlinear features as in (4), PCA loadings are no longer linear transformations but approximations of nonlinear functions belonging to the function class $F_p$ described in Section 2.

As a consequence of Bochner’s theorem (Section 2.1), RPCA converges to KPCA as the number of random features $m$ tends to infinity. This is because random feature dot-products converge uniformly to the exact kernel evaluations in expectation (Rahimi & Recht, 2008). Since the solution of KPCA is the eigensystem of the kernel matrix $K$.
for the data matrix $X \in \mathbb{R}^{n \times d}$, one may study how fast the approximation $\hat{K}$ made in (6) converges in operator norm to $K$ as $m$ grows.

To analyze this convergence we appeal to the recently proposed Matrix Bernstein Inequality. In the theorem below and henceforth $\|X\|$ denotes the operator norm.

**Theorem 2 (Matrix Bernstein, (Mackey et al., 2012)).** Let $Z_1, \ldots, Z_m$ be independent $d \times d$ random matrices. Assume that $\mathbb{E}[Z_i] = 0$ and that $\|Z_i\| \leq R$. Define the variance parameter $\sigma^2 := \max \{ \sum_i \mathbb{E}[Z_i^T Z_i], \sum_i \mathbb{E}[Z_i Z_i^T] \}$. Then, for all $t \geq 0$,

$$
P \left( \left\| \sum_i Z_i \right\| \geq t \right) \leq 2d \cdot \exp \left( \frac{-t^2}{3 \sigma^2 + 2Rt} \right).
$$

Furthermore,

$$
\mathbb{E} \left\| \sum_i Z_i \right\| \leq \sigma \sqrt{3 \log(2d) + R \log(2d)}.
$$

The convergence rate of RPCA to its exact kernel counterpart KPCA is expressed by the following theorem, which actually invokes the Hermitian matrix version of Theorem 3.1, and hence depends on $d$ instead of $2d$, and uses matrix squares when defining the variance $\sigma^2$.

**Theorem 3.** Assume access to the data $X \in \mathbb{R}^{n \times d}$ and a shift-invariant, even kernel $k$. Construct the kernel matrix $K_{ij} := k(x_i, x_j)$ and its approximation $\hat{K}$ using $m$ random features as per (6). Then,

$$
\mathbb{E} \left\| \hat{K} - K \right\| \leq \sqrt{\frac{3n^2 \log n}{m}} + \frac{2 \log n}{m}.
$$

**Proof.** We follow a derivation similar to (Tropp, 2012).

Denote by

$$
\hat{K} := \frac{1}{m} \sum_{i=1}^m z_i z_i^T = \sum_{i=1}^m \hat{K}^{(i)}
$$

the $n \times n$ sample kernel matrix, and by $K$ its population counterpart such that $\mathbb{E}[K] = K$. Note that $\hat{K}$ is the sum of the $m$ independent matrices $\hat{K}^{(i)}$, since our random features are sampled i.i.d. and the matrix $X$ is defined to be constant. Consider the individual error matrices

$$
E = \hat{K} - K = \sum_{i=1}^m E_i, \quad E_i = \frac{1}{m} (\hat{K}^{(i)} - K),
$$

each of which satisfies $\mathbb{E}[E_i] = 0$. Since we are using cosine features—see $z(x)$ in (4)—it follows that there exists a constant $B$ such that $\|z\|^2 \leq B$. Thus, we see that

$$
\|E_i\| = \frac{1}{m} \|z_i z_i^T - \mathbb{E}[z z^T]\| \leq \frac{1}{m} (\|z_i\|^2 + \mathbb{E}[\|z\|^2]) \leq \frac{2B}{m},
$$

because of the triangle inequality on the norm and Jensen’s inequality on the expected value. To bound the variance of $E$, bound first the variance of each of its summands $E_i$ (noting that $\mathbb{E}[z_i z_i^T] = K$):

$$
\mathbb{E}[E_i^2] = \frac{1}{m^2} \mathbb{E} [ (z_i z_i^T - K)^2 ] = \frac{1}{m^2} \mathbb{E} [ \|z_i z_i^T - z_i z_i^T K - K z_i z_i^T + K^2 \| ] \leq \frac{1}{m^2} [ B K - 2 K^2 + K^2 ] \leq \frac{B K}{m^2}.
$$

Next, taking all summands $E_i$ together we obtain

$$
\mathbb{E} \|E\|^2 \leq \sum_{i=1}^m \mathbb{E} E_i^2 \leq \frac{1}{m} B \|K\|.
$$

Where the first inequality follows by Jensen. We can now invoke the matrix Bernstein inequality (Theorem 3.1) on $E - \mathbb{E}[E]$ to obtain the bound

$$
\mathbb{E} \|\hat{K} - K\| \leq \sqrt{\frac{3B \|K\| \log n}{m}} + \frac{2 B \log n}{m}.
$$

Observe that random features and kernel evaluations are upper-bounded by 1; thus both $B$ and $\|K\|$ are upper-bounded by $n$, yielding the bound (7).

To obtain a characterization in relative-error, we can divide both sides of (7) by $\|K\|$. This results in a bound that depends on $n$ logarithmically (since $\|K\| = O(n)$). Moreover, additional information may be extracted from the tail-probability version of Theorem 3.1 for additional discussion on this aspect.

Before closing this section, we mention a byproduct of our above analysis.

**Extension to Spectral Clustering.** Spectral clustering (Luxburg, 2007) uses the spectrum of $K$ to perform dimensionality reduction before applying k-means. Therefore, the analysis of RPCA can be easily extended to obtain a randomized and nonlinear variant of spectral clustering.

**4. Canonical Correlation Analysis (CCA)**

Canonical Correlation Analysis or CCA (Hotelling, 1936) measures the correlation between two multidimensional random variables. Specifically, given two samples $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$, CCA computes a pair of canonical bases $F \in \mathbb{R}^{p \times r}$ and $G \in \mathbb{R}^{q \times r}$ such that

$$
\| \text{corr}(X F, Y G) - I \|_F \text{ is minimized},
$$

corr(X F, X F) = I, \quad \text{corr}(Y G, Y G) = I,
$$

where $I$ stands for the identity matrix. The correlations between the canonical variables $XF$ and $YG$ are referred to as canonical correlations, and up to $r = \max(\text{rank}(X), \text{rank}(Y))$ of them can be computed.
The canonical correlations $\rho_1^2, \ldots, \rho_r^2$ and basis vectors $f_1, \ldots, f_r \in \mathbb{R}^p$ and $g_1, \ldots, g_r \in \mathbb{R}^q$ form the eigensystem of the generalized eigenvalue problem (Bie et al., 2005):

$$
\begin{pmatrix}
0 & C_{XY} \\
C_{XY} & 0
\end{pmatrix}
\begin{pmatrix}
f \\
g
\end{pmatrix} =
\begin{pmatrix}
\rho_1^2 & 0 \\
0 & \rho_2^2
\end{pmatrix}
\begin{pmatrix}
C_{XX} + \gamma_x I & 0 \\
0 & C_{YY} + \gamma_y I
\end{pmatrix}
\begin{pmatrix}
f \\
g
\end{pmatrix},
$$

where $C_{XY}$ is the covariance $\text{cov}(X, Y)$, while the diagonal terms $\gamma I$ act as regularization.

In another words, CCA processes two different views of the same data (i.e., speech audio signals and paired speaker video frames) and returns their maximally correlated linear transformations. This is particularly useful when the two views are available at training time, but only one of them is available at test time (Kakade & Foster, 2007; Chaudhuri et al., 2009; Vapnik & Vashist, 2009).

Several nonlinear extensions of CCA have been proposed:

- **Kernel Canonical Correlation Analysis or KCCA (Lai & Fyfe, 2000; Melzer et al., 2001; Bach & Jordan, 2002)** uses the kernel trick to derive a nonparametric and nonlinear CCA algorithm. Its explicit computation takes time $O(n^3)$. If not regularized, KCCA is ill-posed and always returns $\rho_2^2 = 1$ canonical correlations.

- **Deep Canonical Correlation Analysis or DCCA (Andrew et al., 2013)** feeds the pair of input variables through a deep neural network. Transformation weights are learnt using gradient descent to maximize the correlation of the output mappings.

### 4.1. Randomized Nonlinear CCA (RCCA)

We now propose RCCA, a nonlinear and randomized variant of CCA. As will be shown, RCCA is a low-rank approximation of KCCA when the latter is equipped with a pair of shift-invariant kernels. RCCA can be understood as linear CCA performed on a pair of randomized nonlinear mappings (see 4): $z_x : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^{n \times m_x}, z_y : \mathbb{R}^{n \times q} \rightarrow \mathbb{R}^{n \times m_y}$ of the data $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$. Schematically,

$$
\text{RCCA}(X, Y) := \text{CCA}(z_x(X), z_y(Y)) \approx \text{KCCA}(X, Y).
$$

CCA requires $O((p^2 + q^2)n)$ operations while RCCA costs $O(\max(m_x, m_y)^2 n)$. Both are linear in the sample size $n$; if $\max(m_x, m_y) \ll \max(p, q)$, RCCA is much cheaper.

When performing RCCA, the basis vectors $f_1, \ldots, f_r \in \mathbb{R}^p$ and $g_1, \ldots, g_r \in \mathbb{R}^q$ become the basis functions $f_1, \ldots, f_r : \mathbb{R}^p \rightarrow \mathbb{R}$ and $g_1, \ldots, g_r : \mathbb{R}^q \rightarrow \mathbb{R}$, which approximate functions in the class $\mathcal{F}_p$ defined in Section 2.

As with PCA, we are interested in characterizing the convergence rate of RCCA to its exact kernel counterpart KCCA as $m_x$ and $m_y$ grow. The solution of KCCA is the eigensystem of the matrix $R^{-1} L$, where,

$$
R^{-1} := \begin{pmatrix}
(K_x + \gamma_x I)^{-1} & 0 \\
0 & (K_y + \gamma_y I)^{-1}
\end{pmatrix},
$$

$$
L := \begin{pmatrix}
0 & K_y \\
K_x & 0
\end{pmatrix},
$$

and $(\gamma_x, \gamma_y)$ are positive regularizers mandatory for avoiding spurious $\pm 1$ correlations (Bach & Jordan, 2002). Theorem 4 characterizes the convergence rate of RCCA to KCCA. Let $\hat{R}^{-1}$ and $\hat{L}$ be the approximations to (8) and (9) obtained by using $m$ random features; that is

$$
\hat{R}^{-1} := \begin{pmatrix}
(\hat{K}_x + \gamma_x I)^{-1} & 0 \\
0 & (\hat{K}_y + \gamma_y I)^{-1}
\end{pmatrix},
$$

$$
\hat{L} := \begin{pmatrix}
0 & \hat{K}_y \\
\hat{K}_x & 0
\end{pmatrix}.
$$

**Theorem 4.** Assume access to the datasets $X \in \mathbb{R}^{n \times p}$, $Y \in \mathbb{R}^{n \times q}$ and shift-invariant kernels $k_x, k_y$. Define the kernel matrices $(K_x)_{ij} := k_x(x_i, x_j), (K_y)_{ij} := k_y(y_1, y_1)$ and their approximations $\hat{K}_x, \hat{K}_y$ using $m_x, m_y$ random features as in (4), respectively. Let $L, R, \hat{L}, \hat{R}$ be as defined in (8–11), where $\gamma_x, \gamma_y > 0$ are regularization parameters. Furthermore, define $\gamma := \min(\gamma_x, \gamma_y), m := \min(m_x, m_y)$. Then,

$$
E \| \hat{R}^{-1} \hat{L} - R^{-1} L \| \leq \frac{1}{\gamma} \left( \sqrt{\frac{3m^2 \log 2n}{m} + \frac{2n \log 2n}{m}} \right).
$$

**Proof.** As the matrices are block-diagonal, we have

$$
E \| \hat{R}^{-1} \hat{L} - R^{-1} L \|
\leq \max(E \| (\hat{K}_x + \gamma_x I)^{-1} \hat{K}_y - (K_x + \gamma_x I)^{-1} K_y \|),
E \| (\hat{K}_y + \gamma_y I)^{-1} \hat{K}_x - (K_y + \gamma_y I)^{-1} K_x \|
$$

We analyze the first term of the maximum; the latter can be analyzed analogously. Let $A := (K_x + \gamma_x I)^{-1}$ and $A := (K_x + \gamma_x I)^{-1}$. Define the individual error terms

$$
E_i = \frac{1}{m_y}(\hat{A} \hat{K}_y^{(i)} - A K_y),
E = \sum_{i=1}^{m_y} E_i.
$$

Recall that the $m_x + m_y$ random features are sampled i.i.d. and that the data matrices $X, Y$ are constant. Therefore, the random matrices $K_x^{(1)}, \ldots, K_x^{(m_x)}, K_y^{(1)}, \ldots, K_y^{(m_y)}$ are i.i.d. random variables. Hence, their expectations factorize:

$$
E[E_i] = \frac{1}{m_y} (E[A] K_y - A K_y),
$$

where we used $E[\hat{K}_y^{(i)}] = K_y$. The deviation of the individual error matrices from their expectations is

$$
Z_i := E_i - E[E_i] = \frac{1}{m_y}(\hat{A} \hat{K}_y^{(i)} - E[A] K_y),
$$

$$
\sum_{i=1}^{m_y} E[Z_i] = -E[E_i] = \frac{1}{m_y} (E[A] K_y - A K_y).
$$
and the norm of this deviation is bounded as
\[ \|Z_i\| = \frac{1}{m_y} \|\hat{A} \hat{K}_y^{(i)} - \mathbb{E}[\hat{A}] K_y\| \leq \frac{2B}{m_y \gamma_x} =: R. \]

The inequality follows by applying Hölder twice after using the triangle inequality. We now turn to the issue of computing the variance, which is defined as
\[ \sigma^2 := \max \left\{ \left\| \sum_{i=1}^{m_y} \mathbb{E}[Z_i Z_i^T] \right\|, \left\| \sum_{i=1}^{m_y} \mathbb{E}[Z_i^T Z_i] \right\| \right\} . \]

Consider first second argument of the maximum above, for which we expand an individual term in the summand:
\[
Z_i^T Z_i = \frac{1}{m_y} \left( \hat{K}_y^{(i)} \hat{A}^2 \hat{K}_y^{(i)} + K_y \mathbb{E}[\hat{A}]^2 K_y \right. \\
\left. - \hat{K}_y^{(i)} \hat{A} \mathbb{E}[\hat{A}] K_y - \mathbb{E}[\hat{A}] K_y \hat{A} \hat{K}_y^{(i)} \right).
\]

Taking expectations we see that
\[
\mathbb{E}[Z_i^T Z_i] = \frac{1}{m_y} \left( \mathbb{E}[\hat{K}_y^{(i)} \hat{A}^2 \hat{K}_y^{(i)}] - K_y \mathbb{E}[\hat{A}]^2 K_y \right) \\
\leq \frac{1}{m_y} \mathbb{E}[\hat{K}_y^{(i)} \hat{A}^2 \hat{K}_y^{(i)}],
\]

where the inequality follows as \(K_y \mathbb{E}[\hat{A}]^2 K_y \geq 0\). Taking norms and invoking Jensen’s inequality we then obtain
\[
\left\| \mathbb{E}[Z_i^T Z_i] \right\| \leq \frac{B \|K_y\|}{m^2 \gamma^2}.
\]

A similar argument shows that
\[
\mathbb{E}[Z_i Z_i^T] \leq \frac{1}{m_y} \mathbb{E}[\hat{A} \hat{K}_y^{(i)}^2 \hat{A}] \Rightarrow \left\| \mathbb{E}[Z_i Z_i^T] \right\| \leq \frac{B \|K_y\|}{m^2 \gamma^2}.
\]

An invocation of Jensen on the definition of \(\sigma^2\) along with the two bounds above yields the worst-case estimate
\[
\sigma^2 \leq \frac{B \|K_y\|}{m_y \gamma^2}.
\]

We may now appeal to the matrix Bernstein inequality (Theorem 3.1) to obtain the bound
\[
\mathbb{E} \left[ \left\| \left( \hat{K}_x + \gamma_x I \right)^{-1} \hat{K}_y - \left( K_x + \gamma_x I \right)^{-1} K_y \right\| \right] \leq \\
\frac{1}{\gamma_x} \left( \sqrt{\frac{3n^2 \log 2n}{m_y} + \frac{2n \log 2n}{m_y}} \right).
\]

Finally, the claimed result follows by analogously bounding \(\mathbb{E} \left[ \left\| \left( K_y + \gamma_y I \right)^{-1} K_x - \left( K_y + \gamma_y I \right)^{-1} K_x \right\| \right]\) and taking maxima.

Before concluding this section, we briefly comment on two easy extensions of our above result.

### Extension to Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) seeks a linear combination of the features of the data \(X \in \mathbb{R}^{n \times d}\) such that the samples become maximally separable with respect to a paired labeling \(y\) with \(y_i \in \{1, \ldots, c\}\). LDA can be solved by CCA(\(X, T\), where \(T_{ij} = 1[\{y_i = j\}\) (Bie et al., 2005). Therefore, a similar analysis to the one of RCCA could be used to obtained a randomized nonlinear variant of LDA.

### Extension to RDC

The Randomized Dependence Coefficient or RDC (Lopez-Paz et al., 2013) is defined as the largest canonical correlation of RCCA when performed on the copula transformation of the data matrices of \(X\) and \(Y\). Our analysis applies straightforwardly to the one of RDC.

### 5. Experiments

We investigate the performance of RCCA in multiple experiments with real-world data against state-of-the-art algorithms. For brevity, we illustrate only RCCA as it is the more difficult of the two methods proposed in this paper. Interestingly, in Section 5.3, we provide a novel algorithm based on RCCA to perform learning using privileged information (Vapnik & Vashist, 2009).

We set our random features to have nonlinearity \(\phi(x) = \cos(x)\) and sample parameters according to \(w_i \sim \mathcal{N}(0, 2s I)\). This setup approximates the widely used spherical Gaussian kernel \(k(x, x') = \exp(-s\|x - x'\|^2)\) as \(m \to \infty\). The parameter \(s\) is set using Jaakkola’s heuristic for each tested dataset. RCCA requires no regularization.

#### 5.1. Empirical Validation of Bernstein Inequalities

We first turn to the issue of empirically validating the bounds obtained in Theorems 3 and 4. To do so, we perform simulations in which we separately vary the values of the sample size \(n\), the number of random projections \(m\), and the regularization parameter \(\gamma\). We use synthetic data matrices \(X \in \mathbb{R}^{n \times 10}\) and \(Y \in \mathbb{R}^{n \times 10}\) formed by i.i.d. normal entries. When not varying, the parameters are fixed to \(n = 1000, k = 1000\) and \(\gamma = 10^{-3}\).

Figure 1 depicts the value of the norms from equations (7, 12) as the parameters \(\{n, m, \gamma\}\) vary, when averaged over a total of 1000 random samples \(\{X_i, Y_i\}_{i=1}^{1000}\). The agreement with the theoretical analysis is remarkable: the sample size \(n\) and regularization parameter \(\gamma\) exhibit a linear effect, while increasing the number of random features \(m\) induces an \(O(m^{-1/2})\) reduction in error (the closest function \(\propto m^{-1/2}\) is overlaid in red for comparison).

#### 5.2. Canonical Correlation Analysis

We compare the performance of three variants of CCA on the task of learning features from two modalities of the same
Randomized Nonlinear Component Analysis

Figure 1. Error-norms as a function of a varying parameter, depicted in the x-axis. Left: RPCA. Right: RCCA.

data: linear CCA, Deep CCA (Andrew et al., 2013) and the proposed RCCA. We were unable to run exact KCCA on the proposed datasets due to its cubic computational complexity.

We replicate the two experiments presented in Galen et al. (2013). The task is to measure performance as the accumulated correlation between the canonical variables associated with the top canonical correlations on some unseen test data. The participating datasets are MNIST and XRMB, which are introduced in the following paragraphs.

MNIST Handwritten Digits. Learn correlated representations between the left and right halves of the MNIST dataset of handwritten digit images (LeCun & Cortes, 1998). These images have a width and height of 28 pixels; therefore, each of the two views of CCA consists on 392 features. 54000 samples are used for training, 10000 for testing and 6000 to cross-validate the parameters of each of the CCA variants.

X-Ray Microbeam Speech Data. Learn correlated representations of simultaneous acoustic and articulatory speech measurements (Westbury, 1994). The articulatory measurements describe the position of the speaker’s lips, tongue and jaws for seven consecutive frames, yielding a 112-dimensional vector at each point in time; the acoustic measurements consist on frequency cepstral coefficients (MFCCs) for the same frames, producing a 273-dimensional vector for each point in time. 30000 samples are used for training, 10000 for testing and 10000 to cross-validate the parameters of each of the CCA variants.

Summary of Results. Figures 2 and 3 show the sum of the largest canonical correlations obtained on the test set by each CCA variant, plotted against the number of random projections $m$ (set equally for both $X$ and $Y$) used by RCCA. Running RCCA with 2000 random projections takes two minutes on a 1.8GHz single core laptop computer when using the R code provided in Appendix A. Running a two-layered DCCA on MNIST takes on the order of 10 hours with Intel-MKL C code. The number of weights required to be stored for test time for RCCA is a hundred times lower than for DCCA for MNIST. In terms of explained correlation in the test set, RCCA performs best.

Parameter Selection. Note that no parameters were tuned for RCCA, since the random projection widths are heuristically set, and regularization was automatically provided by the use of randomness. The number of random features $m$ can be set to the maximum value that fits within the available computational budget. On the other hand, CCA has two parameters (regularizer for each view) and DCCA has ten (two autoencoder parameters for pretraining, number of hidden layers, number of hidden units and regularizers...
for each view). These were tuned via cross-validation using the grids described in Galen et al. (2013).

If desired, further speed improvements for RCCA could be achieved by distributing the computation of covariance matrices over several computers, and by making use of truncated SVD (Baglama & Reichel, 2006).

5.3. Learning Using Privileged Information

In Vapnik’s paradigm Learning Using Privileged Information or LUPI (Vapnik & Vashist, 2009) the learner has access to a set of additional features $X_*$ during training time. These features are understood as helpful high-level “teacher explanations” about each of the training samples. The challenge is to build algorithms that are able to extract information from this privileged information at training time to build a better classifier for test time. We propose to use RCCA to construct a highly dependent subspace between the regular features $X$ and the privileged features $X_*$. We experiment with the Animals with Attributes dataset.

In this dataset, the regular set of features $X$ is a set of 30000 pictures of 35 different animals. The set of privileged features $X_*$ is a set of 85 high-level binary attributes associated with each picture (such as eats-fish or can-fly). To extract information from $X_*$ at training time, we build a feature space formed by the concatenation of the 85, five-dimensional canonical variables $z_F^{(i)}(x)$ associated with each RCCA($X, [X_*, y]$), $i \in \{1, \ldots, 85\}$. The vector $y$ denotes the training labels.

![Figure 4. Classification accuracy on the LUPI Experiments.](image)

We perform 15 random training/test partitions of 1000 samples each. Each partition groups a random subset of 10 animals as class “0” and a second random subset of 10 animals as class “1”. Hence, each partition is a different, challenging binary classification problem. Figure 4 shows the test classification accuracy of a linear SVM when using as features the images’ SURF features or the RCCA “semi-privileged” features. As a side note, using directly the high-level attributes yielded 100% accuracy. The cost parameter of the linear SVM is cross-validated on the grid $[10^{-4}, \ldots, 10^4]$. We observe an average improvement of 14% in classification when using the RCCA basis instead of the image features alone. Results are statistically significant respect to a paired Wilcoxon test on a 95% confidence interval.

6. Conclusions

We presented a theoretical and empirical treatment of randomized, nonlinear variants of the classical multivariate analysis methods of Principal Component Analysis and Canonical Correlation Analysis. The use of nonlinear random features significantly reduces the computational complexity of these algorithms, whilst retaining their strong predictive performance. The proposed algorithms do not present any parameters to be tuned, except for the number of random features to be used; however, this can be understood as a convenient knob to straightforwardly trade-off between speed and accuracy.

Several questions of interest for further investigation remain open. For example: (i) how can we leverage knowledge from specific datasets to better design the random feature sampling distribution $p(w)$? (ii) how can we incorporate sparsity assumptions on the random features? (iii) can we form a hierarchy of layers of random features to build deep, more powerful component analysis methods?

A. R Source Code

```r
rcca_fit <- function(x,y,mx,my,sx,sy) {
  w <- matrix(sx*rnorm(ncol(x)*mx),ncol(x))
  m <- matrix(sy*rnorm(ncol(y)*my),ncol(y))
  C <- cancor(cos(x%*%w),cos(y%*%m))
  list(a=C$xcoef,b=C$ycoef,w=w,m=m)
}

rcca_eval <- function(rcca,x,y) {
  list(x=cos(x%*%rcca$w)%*%rcca$a,
       y=cos(y%*%rcca$m)%*%rcca$b)
}

rpca_fit <- function(x,m,s) {
  w <- matrix(s*rnorm(ncol(x)*m),ncol(x))
  list(w=w,pca=prcomp(cos(x%*%w)))
}

rpca_eval <- function(rpca,x) {
  predict(rpca$pca,cos(x%*%rpca$w))
}
```

B. Acknowledgments

DLP is thankful for the fruitful discussions had with Yarin Gal and Maxim Rabinovich. DLP is supported by Obra Social “la Caixa”.

References


