Randomized Nonlinear Component Analysis

David Lopez-Paz   DLOPEZ@TUE.MPG.DE
Max-Planck-Institute for Intelligent Systems, University of Cambridge

Suvrit Sra   SUVRIT@TUE.MPG.DE
Max-Planck-Institute for Intelligent Systems, Carnegie Mellon University

Alexander J. Smola   ALEX@SMOLA.ORG
Carnegie Mellon University, Google Research

Zoubin Ghahramani   ZOUBIN@ENG.CAM.AC.UK
University of Cambridge

Bernhard Schölkopf   BS@TUE.MPG.DE
Max-Planck-Institute for Intelligent Systems

Abstract

Classical methods such as Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) are ubiquitous in statistics. However, these techniques are only able to reveal linear relationships in data. Although nonlinear variants of PCA and CCA have been proposed, these 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 drastic savings in computational requirements.

In this paper we leverage randomness to design scalable new variants of nonlinear PCA and CCA; our ideas 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. A simple R implementation of the presented algorithms is provided.

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 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 transformations 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 for modern data analysis is that both reveal only linear relationships between the variables under study. To overcome this

limitation, several nonlinear extensions have been proposed for both PCA and CCA. 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; 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 or are not accompanied by theoretical guarantees.

In a separate strand of recent research, randomized strategies have been introduced to construct 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 drastic 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 attention to studying the spectral properties of low-rank kernel matrices constructed as sums of random feature dot-products. Our analysis relies on the recently developed matrix Bernstein inequality (Mackey et al., 2014). With little additional effort, our analysis extends to other popular multivariate analysis tools such as linear discriminant analysis, spectral clustering and 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 Deep Canonical Correlation Analysis (Andrew et al., 2013). As a novel application of the presented methods, we derive an algorithm to learn using privileged information (Vapnik & Vashist, 2009) and a scalable strategy to train nonlinear autoencoder neural networks. Additional numerical simulations are provided to validate the tightness of the concentration bounds derived in our theoretical analysis. Lastly, the presented methods are very simple to implement; we provide R source code at:

<div align="center">http://lopezpaz.org/code/rca.r</div>

1.1. Related Work

There has been a recent stream of research in kernel approximations via randomized feature maps since the seminal work of Rahimi & Recht (2008). For instance, their extensions to dot-product kernels (Kar & Karnick, 2012) and polynomial kernels (Hamid et al., 2014); 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, Lopez-Paz et al. (2013) defined the dependence statistic RDC as the largest canonical correlation between two sets of copula random projections. Second, McWilliams et al. (2013) used the Nyström method to define a randomized feature map and performed CCA to achieve state-of-the-art semi-supervised learning.

2. Random Nonlinear Features

We start our presentation by recalling a few key aspects of nonlinear random features.

Consider the class $\mathcal{F}_p$ of functions whose weights decay faster than some sampling distribution p; formally:

$$\mathcal{F}_p := \left\{ f(\boldsymbol{x}) = \int_{\mathbb{R}^d} \alpha(\boldsymbol{w})\phi(\boldsymbol{x}^T\boldsymbol{w})\mathrm{d}\boldsymbol{w} : |\alpha(\boldsymbol{w})| \leq Cp(\boldsymbol{w}) \right\}, \tag{1}$$

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

$$f_m(\boldsymbol{x}) := \sum\nolimits_{i=1}^m \alpha_i \phi(\boldsymbol{w}_i^T \boldsymbol{x}). \tag{2}$$

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

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

$$R_{\mathrm{emp}}(f) := \frac{1}{m} \sum\nolimits_{i=1}^m c(f_m(\boldsymbol{x}_i), y_i), \tag{3}$$

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

Jointly optimizing (3) over $(\boldsymbol{\alpha}, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_m)$ used in defining f_m, is a daunting task given the nonlinear nature of ϕ. The key insight of Rahimi & Recht (2008) is that we can instead randomly sample the parameters $\boldsymbol{w}_i \in \mathbb{R}^d$ from a data-independent distribution $p(\boldsymbol{w})$ and construct an m-dimensional randomized feature map $\boldsymbol{z}(\boldsymbol{X})$ for the input

data $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ that obeys the following structure:

$$\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m \sim p(\boldsymbol{w}),$$
$$\boldsymbol{z}_i := [\cos(\boldsymbol{w}_i^T \boldsymbol{x}_1 + b_i), \ldots, \cos(\boldsymbol{w}_i^T \boldsymbol{x}_n + b_i)] \in \mathbb{R}^n,$$
$$\boldsymbol{z}(\boldsymbol{X}) := [\boldsymbol{z}_1 \cdots \boldsymbol{z}_m] \in \mathbb{R}^{n \times m}. \quad (4)$$

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

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \|\boldsymbol{y} - \boldsymbol{z}(\boldsymbol{X})\boldsymbol{\alpha}\|_2^2, \quad \text{s.t. } \|\boldsymbol{\alpha}\|_\infty \leq C. \quad (5)$$

This form remarkably simplifies computation (in practice, we solve a regularized version of it), while incurring only a bounded error. Theorem 1 formalizes this claim.

Theorem 1. *(Rahimi & Recht, 2008) Let $\mathcal{F}_p$ be defined as in (1). Draw $\mathcal{D} \sim Q(\boldsymbol{X}, Y)$. Construct $\boldsymbol{z}(\cdot)$ 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 $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ such that*

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

Solving (5) takes $O(ndm + m^2 n)$ 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 faster computation of the random features, yielding $O(n \log(d)m + m^2 n)$ 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) because of the *implicit regularization* induced by randomness.

2.1. Random Features, Nyström and Kernel Matrices

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

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

where $p(\boldsymbol{w})$ is set to be the inverse Fourier transform of k and $b_i \sim \mathcal{U}(0, 2\pi)$ (Rahimi & Recht, 2008)—e.g., the Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-s\|\boldsymbol{x} - \boldsymbol{y}\|_2^2)$ can be approximated using $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, 2s\boldsymbol{I})$.

Let $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ be the kernel matrix of some data $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, i.e., $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. When approximating the kernel k using m random Fourier features, we may as well approximate the kernel matrix $\boldsymbol{K} \approx \hat{\boldsymbol{K}}$, where

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

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.

Importantly, our analysis extends straight-forwardly to features constructed using the Nyström method (Williams & Seeger, 2001) when its basis are bounded and sampled at random. Recent theoretical and empirical evidence suggest the superiority of the Nyström method when compared to the aforementioned Fourier features (Yang et al., 2012). However, we did not experience large differences (Section 5), while random Fourier features are faster to compute and do not need to be stored at test time (Le et al., 2013).

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 $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ into a set of n observations of d uncorrelated *principal components*.

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

$$\boldsymbol{X} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{F}^T,$$

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

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 randomized nonlinear mapping of the data. Schematically,

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

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

The computational complexity is $O(n^3)$ for KPCA, $O(d^2 n)$ for PCA and $O(m^2 n)$ for RPCA. PCA and RPCA are both linear in the sample size n. When using nonlinear features as in (4), PCA loadings are no longer linear transformations but approximations of nonlinear functions belonging to the function class $\mathcal{F}_p$ described in Section 2.

As a consequence of Bochner's theorem (Section 2.1), the RPCA kernel matrix will approximate the one of 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 $\boldsymbol{K}$ for the data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, one may study how fast the approximation $\hat{\boldsymbol{K}}$ made in (6) converges in operator (or spectral) norm to $\boldsymbol{K}$ as m grows.

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

Theorem 2 (Matrix Bernstein, (Mackey et al., 2014)). *Let $\boldsymbol{Z}_1, \ldots \boldsymbol{Z}_m$ be independent $d \times d$ random matrices. Assume that $\mathbb{E}[\boldsymbol{Z}_i] = 0$ and that $\|\boldsymbol{Z}_i\| \leq R$. Define the variance parameter $\sigma^2 := \max\{\|\sum_i \mathbb{E}[\boldsymbol{Z}_i^T \boldsymbol{Z}_i]\|, \|\sum_i \mathbb{E}[\boldsymbol{Z}_i \boldsymbol{Z}_i^T]\|\}$. Then, for all $t \geq 0$,*

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

Furthermore,

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

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 σ^2.

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

$$\mathbb{E}\|\hat{\boldsymbol{K}} - \boldsymbol{K}\| \leq \sqrt{\frac{3n^2 \log n}{m}} + \frac{2n \log n}{m}. \tag{7}$$

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

Denote by

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

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

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

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

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

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

$$\begin{aligned}
\mathbb{E}[\boldsymbol{E}_i^2] &= \frac{1}{m^2} \mathbb{E}\left[(\boldsymbol{z}_i \boldsymbol{z}_i^T - \boldsymbol{K})^2\right] \\
&= \frac{1}{m^2} \mathbb{E}\left[\|\boldsymbol{z}_i\|^2 \boldsymbol{z}_i \boldsymbol{z}_i^T - \boldsymbol{z}_i \boldsymbol{z}_i^T \boldsymbol{K} - \boldsymbol{K}\boldsymbol{z}_i \boldsymbol{z}_i^T + \boldsymbol{K}^2\right] \\
&\preceq \frac{1}{m^2}\left[B\boldsymbol{K} - 2\boldsymbol{K}^2 + \boldsymbol{K}^2\right] \preceq \frac{B\boldsymbol{K}}{m^2}.
\end{aligned}$$

Next, taking all summands $\boldsymbol{E}_i$ together we obtain

$$\|\mathbb{E}[\boldsymbol{E}^2]\| \leq \left\|\sum_{i=1}^{m} \mathbb{E}\boldsymbol{E}_i^2\right\| \leq \frac{1}{m} B\|\boldsymbol{K}\|.$$

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

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

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

To obtain a characterization in relative-error, we can divide both sides of (7) by $\|\boldsymbol{K}\|$. This results in a bound that depends on n logarithmically (since $\|\boldsymbol{K}\| = O(n)$). Moreover, additional information may be extracted from the tail-probability version of Theorem . Please refer to Section 5.1 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 may 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

$$\|\mathrm{corr}(XF, YG) - I\|_F \text{ is minimized,}$$
$$\mathrm{corr}(XF, XF) = I, \quad \mathrm{corr}(YG, YG) = I,$$

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 $r = \max(\mathrm{rank}(X), \mathrm{rank}(Y))$ 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):

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

where C_{XY} is the covariance $\mathrm{cov}(X, Y)$, while the diagonal terms γI 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; Bach & Jordan, 2002) uses the kernel trick to derive a nonparametric, nonlinear regularized CCA algorithm. Its exact computation takes time $O(n^3)$.
- 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 approxima-

tion 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): $z_x : \mathbb{R}^{n \times p} \to \mathbb{R}^{n \times m_x}$, $z_y : \mathbb{R}^{n \times q} \to \mathbb{R}^{n \times m_y}$ of the data $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$. Schematically,

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

The computational complexity is $O(n^3)$ for KCCA, $O((p^2 + q^2)n)$ for CCA and $O((m_x^2 + m_y^2)n)$ for RCCA. CCA and RCCA are both linear in the sample size n.

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 $f_1, \ldots, f_r : \mathbb{R}^p \to \mathbb{R}$ and $g_1, \ldots, g_r : \mathbb{R}^q \to \mathbb{R}$, 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,

$$R^{-1} := \begin{pmatrix} (K_x + \gamma_x I)^{-1} & 0 \\ 0 & (K_y + \gamma_y I)^{-1} \end{pmatrix}, \quad (8)$$

$$L := \begin{pmatrix} 0 & K_y \\ K_x & 0 \end{pmatrix}, \quad (9)$$

and (γ_x, γ_y) are positive regularizers mandatory to avoid spurious ± 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

$$\hat{R}^{-1} := \begin{pmatrix} (\hat{K}_x + \gamma_x I)^{-1} & 0 \\ 0 & (\hat{K}_y + \gamma_y I)^{-1} \end{pmatrix}, \quad (10)$$

$$\hat{L} := \begin{pmatrix} 0 & \hat{K}_y \\ \hat{K}_x & 0 \end{pmatrix}. \quad (11)$$

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 $(K_x)_{ij} := k_x(x_i, x_j)$, $(K_y)_{ij} := k_y(y_i, y_j)$ 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,*

$$\mathbb{E}\|\hat{R}^{-1}\hat{L} - R^{-1}L\| \leq \frac{1}{\gamma} \left(\sqrt{\frac{3n^2 \log 2n}{m}} + \frac{2n \log 2n}{m} \right).$$
$$(12)$$

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

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

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

$$\boldsymbol{E}_i = \tfrac{1}{m_y}\big(\hat{\boldsymbol{A}}\hat{\boldsymbol{K}}_y^{(i)} - \boldsymbol{A}\boldsymbol{K}_y\big), \quad \boldsymbol{E} = \sum_{i=1}^{m_y} \boldsymbol{E}_i.$$

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

$$\mathbb{E}\left[\boldsymbol{E}_i\right] = \tfrac{1}{m_y}\big(\mathbb{E}[\hat{\boldsymbol{A}}]\boldsymbol{K}_y - \boldsymbol{A}\boldsymbol{K}_y\big),$$

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

$$\boldsymbol{Z}_i := \boldsymbol{E}_i - \mathbb{E}\left[\boldsymbol{E}_i\right] = \tfrac{1}{m_y}\big(\hat{\boldsymbol{A}}\hat{\boldsymbol{K}}_y^{(i)} - \mathbb{E}[\hat{\boldsymbol{A}}]\boldsymbol{K}_y\big),$$

and the norm of this deviation is bounded as

$$\|\boldsymbol{Z}_i\| = \frac{1}{m_y}\|\hat{\boldsymbol{A}}\hat{\boldsymbol{K}}_y^{(i)} - \mathbb{E}[\hat{\boldsymbol{A}}]\boldsymbol{K}_y\| \le \frac{2B}{m_y \gamma_x} =: R.$$

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

$$\sigma^2 := \max\left\{\left\|\sum_{i=1}^{m_y} \mathbb{E}[\boldsymbol{Z}_i \boldsymbol{Z}_i^T]\right\|, \left\|\sum_{i=1}^{m_y} \mathbb{E}[\boldsymbol{Z}_i^T \boldsymbol{Z}_i]\right\|\right\}.$$

Consider first second argument of the maximum above, for which we expand an individual term in the summand:

$$\boldsymbol{Z}_i^T \boldsymbol{Z}_i = \frac{1}{m_y^2}\Big(\hat{\boldsymbol{K}}_y^{(i)}\hat{\boldsymbol{A}}^2\hat{\boldsymbol{K}}_y^{(i)} + \boldsymbol{K}_y\mathbb{E}[\hat{\boldsymbol{A}}]^2\boldsymbol{K}_y$$
$$- \hat{\boldsymbol{K}}_y^{(i)}\hat{\boldsymbol{A}}\mathbb{E}[\hat{\boldsymbol{A}}]\boldsymbol{K}_y - \mathbb{E}[\hat{\boldsymbol{A}}]\boldsymbol{K}_y\hat{\boldsymbol{A}}\hat{\boldsymbol{K}}_y^{(i)}\Big).$$

Taking expectations we see that

$$\mathbb{E}[\boldsymbol{Z}_i^T \boldsymbol{Z}_i] = \frac{1}{m_y^2}\Big(\mathbb{E}[\hat{\boldsymbol{K}}_y^{(i)}\hat{\boldsymbol{A}}^2\hat{\boldsymbol{K}}_y^{(i)}] - \boldsymbol{K}_y\mathbb{E}[\hat{\boldsymbol{A}}]^2\boldsymbol{K}_y\Big)$$
$$\preceq \frac{1}{m_y^2}\mathbb{E}[\hat{\boldsymbol{K}}_y^{(i)}\hat{\boldsymbol{A}}^2\hat{\boldsymbol{K}}_y^{(i)}],$$

where the inequality follows as $\boldsymbol{K}_y\mathbb{E}[\hat{\boldsymbol{A}}]^2\boldsymbol{K}_y \succeq 0$. Taking norms and invoking Jensen's inequality we then obtain

$$\big\|\mathbb{E}[\boldsymbol{Z}_i^T \boldsymbol{Z}_i]\big\| \le \frac{B\|\boldsymbol{K}_y\|}{m^2\gamma^2}.$$

A similar argument shows that

$$\mathbb{E}[\boldsymbol{Z}_i \boldsymbol{Z}_i^T] \preceq \frac{1}{m_y^2}\mathbb{E}[\hat{\boldsymbol{A}}(\hat{\boldsymbol{K}}_y^{(i)})^2\hat{\boldsymbol{A}}] \Rightarrow \|\mathbb{E}[\boldsymbol{Z}_i \boldsymbol{Z}_i^T]\| \le \frac{B\|\boldsymbol{K}_y\|}{m^2\gamma^2}.$$

An invocation of Jensen on the definition of σ^2 along with the two bounds above yields the worst-case estimate

$$\sigma^2 \le \frac{B\|\boldsymbol{K}_y\|}{m_y \gamma^2}.$$

We may now appeal to the matrix Bernstein inequality (Theorem 3.1) to obtain the bound

$$\mathbb{E}\|(\hat{\boldsymbol{K}}_x + \gamma_x \boldsymbol{I})^{-1}\hat{\boldsymbol{K}}_y - (\boldsymbol{K}_x + \gamma_x \boldsymbol{I})^{-1}\boldsymbol{K}_y\| \le$$
$$\frac{1}{\gamma_x}\left(\sqrt{\frac{3n^2 \log 2n}{m_y}} + \frac{2n \log 2n}{m_y}\right).$$

The result follows by analogously bounding $\mathbb{E}\|(\hat{\boldsymbol{K}}_y + \gamma_y \boldsymbol{I})^{-1}\hat{\boldsymbol{K}}_x - (\boldsymbol{K}_y + \gamma_y \boldsymbol{I})^{-1}\boldsymbol{K}_x\|$ 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 $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ such that the samples become maximally separable with respect to a paired labeling $\boldsymbol{y}$ with $y_i \in \{1, \ldots, c\}$. LDA can be solved by $\text{CCA}(\boldsymbol{X}, \boldsymbol{T})$, where $\boldsymbol{T}_{ij} = \mathbb{I}\{y_i = j\}$ (Bie et al., 2005). Therefore, a similar analysis to the one of RCCA could be used to obtain 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 $\boldsymbol{X}$ and $\boldsymbol{Y}$. Our analysis applies to the further understanding of RDC.

5. Experiments

We investigate the performance of RCCA in multiple experiments with real-world data against state-of-the-art algorithms. Section 5.3 provides a novel algorithm based on RCCA to perform learning using privileged information (Vapnik & Vashist, 2009). Section 5.4 introduces the use of RPCA as a tool to train autoencoders in a scalable manner.

We set our random (Fourier) features to approximate the Gaussian kernel, as described in the second paragraph of Section 2.1. We also compare to the Nyström method, set to construct an $m-$dimensional feature space formed by the evaluations of the Gaussian kernel on m random points from the training set (Yang et al., 2012). Gaussian kernel widths $\{s_x, s_y\}$ are set using the median heuristic.

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 γ. We use synthetic data matrices $\boldsymbol{X} \in \mathbb{R}^{n \times 10}$ and $\boldsymbol{Y} \in \mathbb{R}^{n \times 10}$ formed by i.i.d. normal entries. When not varying, the parameters are fixed to $n = 1000$, $m = 1000$ and $\gamma = 10^{-3}$.
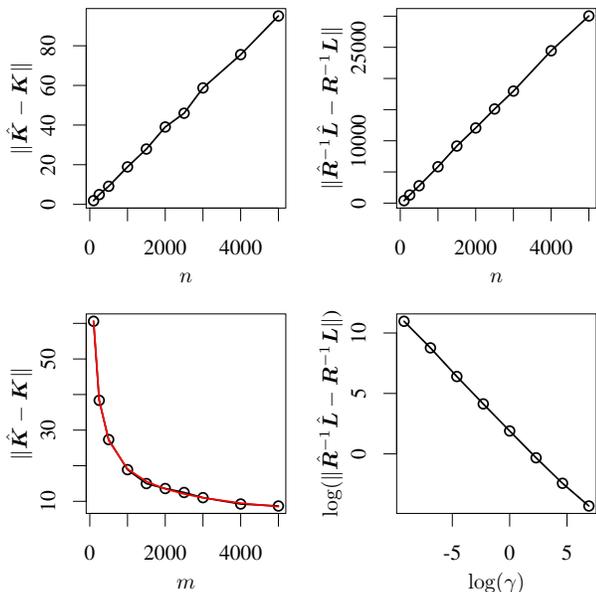
Figure 1. Error-norms as a function of a varying parameter, depicted in the x-axis. Left: RPCA. Right: RCCA.

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 $\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}_{i=1}^{1000}$. The simulations agree with the presented theoretical analysis: the sample size n and regularization parameter γ exhibit a linear effect, while increasing the number of random features m induces an $O(m^{-1/2})$ reduction in error (the closest function $O(m^{-1/2})$ is overlaid in red for comparison).

5.2. Canonical Correlation Analysis

We compare three variants of CCA on the task of learning correlated features from two modalities of the same data: linear CCA, state-of-the-art Deep CCA (Andrew et al., 2013) and the proposed (Fourier and Nyström based) RCCA. We were unable to run exact KCCA on the proposed datasets due to its cubic complexity; other low-rank approximations such as the one of Arora & Livescu (2012) were shown inferior to DCCA, and hence omitted in our analysis.

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

MNIST Handwritten Digits. Learn correlated representations between the left and right halves of the MNIST images (LeCun & Cortes, 1998). Each image has a width and height of 28 pixels; therefore, each of the two views of CCA

consists on 392 features. 54000 random samples are used for training, 10000 for testing and 6000 to cross-validate the parameters of (D)CCA.

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 are the MFCCs for the same frames, producing a 273-dimensional vector for each point in time. 30000 random samples are used for training, 10000 for testing and 10000 to cross-validate the parameters of (D)CCA.

RCCA on **MNIST** (50 largest canonical correlations)

m_x, m_y	Fourier		Nyström	
	corr.	minutes	corr.	minutes
1000	36.31	5.55	41.68	5.29
2000	39.56	19.45	43.15	18.57
3000	40.95	41.98	43.76	41.25
4000	41.65	73.80	44.12	75.00
5000	41.89	112.80	44.36	115.20
6000	42.06	153.48	**44.49**	156.07

RCCA on **XRMB** (112 largest canonical correlations)

m_x, m_y	Fourier		Nyström	
	corr.	minutes	corr.	minutes
1000	68.79	2.95	81.82	3.07
2000	82.62	11.45	93.21	12.05
3000	89.35	26.31	98.04	26.07
4000	93.69	48.89	100.97	50.07
5000	96.49	79.20	103.03	81.6
6000	98.61	120.00	**104.47**	119.4

	linear CCA		DCCA	
	corr.	minutes	corr.	minutes
MNIST	28.0	0.57	39.7	787.38
XRMB	16.9	0.11	92.9	4338.32

Table 1. Sum of largest test canonical correlations and running times by all CCA variants in the MNIST and XRMB datasets.

Summary of Results. Table 1 shows the sum of the largest canonical correlations (corr.) obtained by each CCA variant and their running times (minutes, single 1.8GHz core) on the MNIST and XRMB test sets. Given enough random projections ($m = m_x = m_y$), RCCA is able to explain the most amount of test correlation while running drastically faster than DCCA[1]. Moreover, when using ran-

---

[1] Running times for DCCA correspond to a single cross-validation iteration of its ten hyper-parameters. DCCA has 2 layers for MNIST and 8 layers for XRMB.

dom features (i) the number of weights required to be stored at test time for RCCA is up to two orders of magnitude lower than for DCCA and (ii) the use of Fastfood multiplications (Le et al., 2013) allows much faster model evaluation.

Parameter Selection. No parameters were tuned for RCCA: the kernel widths were heuristically set and CCA regularization is implicitly provided by the use of randomness (thus set to 10^{-8}). The number of random features m can be set to the maximum value that fits within the available (training or test time) computational budget. On the contrary, previous state-of-the-art DCCA has ten parameters (two autoencoder parameters for pretraining, number of hidden layers, number of hidden units and CCA regularizers for each view), which were cross-validated using the grids described in Andrew et al. (2013). Cross-validating RCCA parameters did not significantly improve performance.

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

5.3. Learning Using Privileged Information

In Vapnik's *Learning Using Privileged Information* (LUPI) paradigm (Vapnik & Vashist, 2009) the learner has access to a set of *privileged* features or information $X_\star$, exclusive of training time. These features are understood as helpful high-level "teacher explanations" about each of the training samples. The challenge is to build algorithms able to extract information from this privileged features at training time in order to build a better classifier at test time. We propose to use RCCA to construct a highly correlated subspace between the regular features X and the privileged features $X_\star$, accessible at test time through a nonlinear transformation of X.

We experiment with the *Animals-with-Attributes* dataset (Lampert et al., 2009). In this dataset, the regular features X are the SURF descriptors of 30000 pictures of 35 different animals; the privileged features $X_\star$ are 85 high-level binary attributes associated with each picture (such as *eats-fish* or *can-fly*). To extract information from $X_\star$ at training time, we build a feature space formed by the concatenation of the 85, five-dimensional top canonical variables $z_x(X)F_{1:5}^{(i)}$ associated with each RCCA$(X, [X_\star^{(i)}, y])$, $i \in \{1, \ldots, 85\}$. The vector y denotes the training labels.

We perform 14 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 experiment is a different, challenging binary classification problem. Figure 2 shows the test classification accuracy of a linear SVM when using as features the images' SURF descriptors or the RCCA
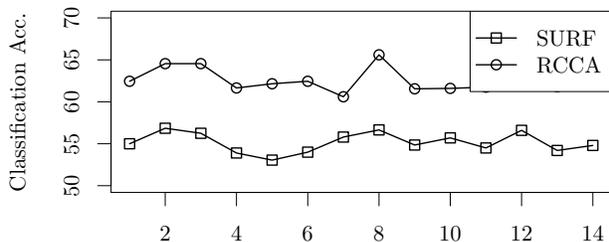
Figure 2. Classification accuracy on the LUPI Experiments.

"semi-privileged" features. As a side note, directly using the high-level attributes yields 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. The SVM+ algorithm (Vapnik & Vashist, 2009) did not improve on the regular SVM using SURF descriptors.

5.4. Randomized Autoencoders

RPCA can be used for scalable training of nonlinear autoencoders. The process involves (i) mapping the observed data $Y \in \mathbb{R}^{D \times n}$ into the latent factors $X \in \mathbb{R}^{d \times n}$ using the top d nonlinear principal components from RPCA and (ii) reconstructing Y from X using D nonlinear regressors.

Figure 3. Autoencoder reconstructions of unseen test images for the MNIST (top) and CIFAR-10 (bottom) datasets.

Figure 3 shows the reconstruction of *unseen* MNIST and CIFAR-10 images after being compressed with RPCA. The number random projections was set to $m = 2000$. The number of latent dimensions was set to $d = 20$ for MNIST, and $d = 40$ (first row) or $d = 100$ (second row) for CIFAR-10. Training took under 200 seconds for each full dataset.

References

Achlioptas, D., McSherry, F., and Schölkopf, B. Sampling techniques for kernel methods. *NIPS*, 2002.

Andrew, G., Arora, R., Livescu, K., and Bilmes, J. Deep canonical correlation analysis. *ICML*, 2013.

Arora, R. and Livescu, K. Kernel CCA for multi-view learning of acoustic features using articulatory measurements. *MLSLP*, 2012.

Avron, H., Boutsidis, C., Toledo, S., and Zouzias, A. Efficient dimensionality reduction for canonical correlation analysis. *ICML*, 2013.

Bach, F. R. and Jordan, M. I. Kernel independent component analysis. *JMLR*, 2002.

Baglama, J. and Reichel, L. Restarted block lanczos bidiagonalization methods. *Numerical Algorithms*, 2006.

Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 1989.

Bie, T. De, Cristianini, N., and Rosipal, R. Eigenproblems in pattern recognition. *Handbook of Geometric Computing*, 2005.

Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K. Multi-view clustering via canonical correlation analysis. *ICML*, 2009.

Hamid, R., Xiao, Y., Gittens, A., and DeCoste, D. Compact random feature maps. *ICML*, 2014.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 2006.

Hotelling, H. Relations Between Two Sets of Variates. *Biometrika*, 1936.

Jolliffe, I. T. *Principal Component Analysis*. Springer, 2002.

Kakade, S. M. and Foster, D. P. Multi-view regression via canonical correlation analysis. *COLT*, 2007.

Kar, P. and Karnick, H. Random feature maps for dot product kernels. *arXiv:1201.6530*, 2012.

Lai, P. and Fyfe, C. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 2000.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by betweenclass attribute transfer. *CVPR*, 2009.

Le, Q., Sarlos, T., and Smola, A. Fastfood – Approximating kernel expansions in loglinear time. *ICML*, 2013.

LeCun, Y. and Cortes, C. The MNIST database of handwritten digits. 1998.

Lopez-Paz, D., Hennig, P., and Schölkopf, B. The Randomized Dependence Coefficient. *NIPS*, 2013.

Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 2007.

Mackey, L., Jordan, M. I., Chen, R. Y., Farrell, B., and Tropp, J. A. Matrix Concentration Inequalities via the Method of Exchangeable Pairs. *Annals of Probability*, 2014.

Mahoney, M. W. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 2011.

McWilliams, B., Balduzzi, D., and Buhmann, J. Correlated random features for fast semi-supervised learning. *NIPS*, 2013.

Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 1901.

Rahimi, A. and Recht, B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *NIPS*, 2008.

Schölkopf, B. and Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2002.

Schölkopf, B., Smola, A., and Müller, K. R. Kernel principal component analysis. *Advances in kernel methods - Support vector learning*, 1999.

Tropp, J. A. User-Friendly Tools for Random Matrices: An Introduction. *NIPS Tutorials*, 2012.

Vapnik, V. and Vashist, A. A new learning paradigm: Learning using privileged information. *Neural Networks*, 2009.

Westbury, J. R. X-Ray microbeam speech production database user's handbook version 1.0. 1994.

Williams, C. and Seeger, M. Using the Nyström method to speed up kernel machines. *NIPS*, 2001.

Yang, J., Sindhwani, V., Avron, H., and Mahoney, M. W. Quasi-Monte Carlo Feature Maps for Shift-Invariant Kernels. *ICML*, 2014.

Yang, T., Li, Y., Mahdavi, M., Jin, R., and Zhou, Z. Nyström method vs random Fourier features: A theoretical and empirical comparison. *NIPS*, 2012.