Nonparametric Canonical Correlation Analysis

Tomer Michaeli[1] Weiran Wang[2] Karen Livescu[2]
[1]Technion–Israel Institute of Technology, Haifa, Israel
[2]TTI-Chicago, Chicago, IL 60637, USA
tomer.m@technion.ac.il {weiranwang,klivescu}@ttic.edu

February 9, 2016

Abstract

Canonical correlation analysis (CCA) is a classical representation learning technique for finding correlated variables in multi-view data. Several nonlinear extensions of the original linear CCA have been proposed, including kernel and deep neural network methods. These approaches seek maximally correlated projections among families of functions, which the user specifies (by choosing a kernel or neural network structure), and are computationally demanding. Interestingly, the theory of nonlinear CCA, without functional restrictions, had been studied in the population setting by Lancaster already in the 1950s, but these results have not inspired practical algorithms. We revisit Lancaster's theory to devise a practical algorithm for nonparametric CCA (NCCA). Specifically, we show that the solution can be expressed in terms of the singular value decomposition of a certain operator associated with the joint density of the views. Thus, by estimating the population density from data, NCCA reduces to solving an eigenvalue system, superficially like kernel CCA but, importantly, without requiring the inversion of any kernel matrix. We also derive a partially linear CCA (PLCCA) variant in which one of the views undergoes a linear projection while the other is nonparametric. Using a kernel density estimate based on a small number of nearest neighbors, our NCCA and PLCCA algorithms are memory-efficient, often run much faster, and perform better than kernel CCA and comparable to deep CCA.

1 Introduction

A common task in data analysis is to reveal the common variability in multiple views of the same phenomenon, while suppressing view-specific noise factors. Canonical correlation analysis (CCA) [Hotelling, 1936] is a classical statistical technique that targets this goal. In CCA, linear projections of two random vectors are sought, such that the resulting low-dimensional vectors are maximally correlated. This tool has found widespread use in various fields, including recent application to natural language processing [Dhillon et al., 2011], speech recognition [Arora and Livescu, 2013], genomics [Witten and Tibshirani, 2009], and cross-modal retrieval [Gong et al., 2014].

One of the shortcomings of CCA is its restriction to linear mappings, since many real-world multi-view datasets exhibit highly nonlinear relationships. To overcome this limitation, several extensions of CCA have been proposed for finding maximally correlated *nonlinear* projections. In kernel CCA (KCCA) [Akaho, 2001, Melzer et al., 2001, Bach and Jordan, 2002, Hardoon et al., 2004], these nonlinear mappings are chosen from two reproducing kernel Hilbert spaces (RKHS). In deep CCA (DCCA) [Andrew et al., 2013], the projections are obtained from two deep neural networks that are trained to output maximally correlated vectors. Nonparametric CCA-type methods, which are not limited to specific function classes, include the alternating conditional expectations (ACE) algorithm and its extensions [Breiman and Friedman, 1985, Balakrishnan et al., 2012, Makur et al., 2015]. Nonlinear CCA methods are advantageous over linear CCA in a range of applications [Hardoon et al., 2004, Melzer et al., 2001, Wang et al., 2015b]. However, existing nonlinear CCA approaches are very computationally demanding, and are often impractical to apply on large data.

Interestingly, the problem of finding the most correlated nonlinear projections of two random variables has been studied by Lancaster [1958] and Hannan [1961], long before the derivation of KCCA, DCCA and ACE. They characterized the optimal projections in the population setting, without restricting the solution to an RKHS or to have any particular parametric form. However, these theoretical results have not inspired practical algorithms.

In this paper, we revisit Lancaster's theory, and use it to devise a practical algorithm for *nonparametric CCA* (NCCA). Specifically, we show that the solution to the nonlinear CCA problem can be expressed in terms of the singular value decomposition (SVD) of a certain operator, which is defined via the population density. Therefore, to obtain a practical method, we estimate the density from training data and use the estimate in the solution. The resulting algorithm reduces to solving an eigenvalue system with a particular kernel that depends on the joint distribution between the views. While superficially similar to other eigenvalue methods, it is fundamentally different from them and in particular has crucial advantages over KCCA. For example, unlike KCCA, NCCA does not involve computing the inverse of any matrix, making it computationally feasible on large data where KCCA (even using approximation techniques) is impractical. We elucidate this and other contrasts in Sec. 3 below. We show that NCCA achieves state-of-the art performance, while being much more computationally efficient than KCCA and DCCA.

In certain situations, nonlinearity is needed for one view but not for the other. In such cases, it may be advantageous to constrain the projection of the second view to be linear. An additional contribution of this paper is the derivation of a closed-form solution to this *partially linear CCA* (PLCCA) problem in the population setting. We show that PLCCA has essentially the same form as linear CCA, but with the optimal linear predictor term in CCA replaced by an optimal nonlinear predictor in PLCCA. Thus, moving from the population setting to sample data entails simply using nonlinear regression to estimate this predictor. The resulting algorithm is efficient and, as we demonstrate on realistic data, sometimes matches DCCA and significantly outperforms CCA and KCCA.

2 Background

We start by reviewing the original CCA algorithm [Hotelling, 1936]. Let $X \in \mathbb{R}^{D_x}$ and $Y \in \mathbb{R}^{D_y}$ be two random vectors (views). The goal in CCA is to find a pair of L-dimensional projections $\mathbf{W}_1^\top X, \mathbf{W}_2^\top Y$ that are maximally correlated, but where different dimensions within each view are constrained to be uncorrelated. Assuming for notational simplicity that X and Y have zero mean, the CCA problem can be written as[1]

$$\max_{\mathbf{W}_1,\mathbf{W}_2} \mathbb{E}\left[\left(\mathbf{W}_1^\top X\right)^\top \left(\mathbf{W}_2^\top Y\right)\right] \tag{1}$$
$$\text{s.t. } \mathbb{E}\left[\left(\mathbf{W}_1^\top X\right)\left(\mathbf{W}_1^\top X\right)^\top\right] = \mathbb{E}\left[\left(\mathbf{W}_2^\top Y\right)\left(\mathbf{W}_2^\top Y\right)^\top\right] = \mathbf{I},$$

where the maximization is over $\mathbf{W}_1 \in \mathbb{R}^{D_x \times L}, \mathbf{W}_2 \in \mathbb{R}^{D_y \times L}$. This objective has been extensively studied and is known to be optimal in several senses: It maximizes the mutual information for certain distributions $p(\mathbf{x}, \mathbf{y})$ [Borga, 2001], maximizes the likelihood for certain latent variable models [Bach and Jordan, 2005], and is equivalent to the information bottleneck method when $p(\mathbf{x}, \mathbf{y})$ is Gaussian [Chechik et al., 2005].

The CCA solution can be expressed as $(\mathbf{W}_1, \mathbf{W}_2) = (\boldsymbol{\Sigma}_{xx}^{-1/2}\mathbf{U}, \boldsymbol{\Sigma}_{yy}^{-1/2}\mathbf{V})$, where $\boldsymbol{\Sigma}_{xx} = \mathbb{E}[XX^\top]$, $\boldsymbol{\Sigma}_{yy} = \mathbb{E}[YY^\top]$, $\boldsymbol{\Sigma}_{xy} = \mathbb{E}[XY^\top]$, and $\mathbf{U} \in \mathbb{R}^{D_x \times L}$ and $\mathbf{V} \in \mathbb{R}^{D_y \times L}$ are the top L left and right singular vectors of the matrix $\mathbf{T} = \boldsymbol{\Sigma}_{xx}^{-1/2}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1/2}$ (see [Mardia et al., 1979]). In practice, the joint distribution $p(\mathbf{x}, \mathbf{y})$ is rarely known, and only paired multi-view samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ are available, so the population covariances are replaced by their empirical estimates.[2]

To facilitate the analogy with partially linear CCA (Sec. 3.2), we note that the CCA solution can also be expressed in terms of the optimal predictor (in the mean squared error sense) of X from Y, given by $\hat{X} = \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}Y$, and its covariance $\boldsymbol{\Sigma}_{\hat{x}\hat{x}} = \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$. Specifically, $\mathbf{U}$ corresponds to the eigenvectors of $\mathbf{K} = \mathbf{T}\mathbf{T}^\top = \boldsymbol{\Sigma}_{xx}^{-1/2}\boldsymbol{\Sigma}_{\hat{x}\hat{x}}\boldsymbol{\Sigma}_{xx}^{-1/2}$, and, by algebraic manipulation, the optimal projections can be written as

$$\mathbf{W}_1^\top X = \mathbf{U}^\top \boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}} X, \quad \mathbf{W}_2^\top Y = \mathbf{D}^{-\frac{1}{2}} \mathbf{U}^\top \boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}} \hat{X}, \tag{2}$$

where $\mathbf{D}$ is a diagonal matrix with the top L eigenvalues of $\mathbf{K}$ on its diagonal.

Since the representation power of linear mappings is limited, several nonlinear extensions of problem (1) have been proposed. These methods find two maximally correlated *nonlinear* projections $\mathbf{f} : \mathbb{R}^{D_x} \to \mathbb{R}^L$ and $\mathbf{g} : \mathbb{R}^{D_y} \to \mathbb{R}^L$ by

[1] Here and throughout, expectations are with respect to the joint distribution of all random variables (capital letters) appearing within the square brackets of the expectation operator $\mathbb{E}$.

[2] $\boldsymbol{\Sigma}_{xy} \approx \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i\mathbf{y}_i^\top$ and similarly for $\boldsymbol{\Sigma}_{xx}$ and $\boldsymbol{\Sigma}_{yy}$.

solving

$$\max_{\mathbf{f}\in\boldsymbol{\mathcal{A}},\mathbf{g}\in\boldsymbol{\mathcal{B}}} \; \mathbb{E}\big[\mathbf{f}(X)^\top \mathbf{g}(Y)\big] \qquad (3)$$
$$\text{s.t.} \;\; \mathbb{E}\big[\mathbf{f}(X)\mathbf{f}(X)^\top\big] = \mathbb{E}\big[\mathbf{g}(Y)\mathbf{g}(Y)^\top\big] = \mathbf{I},$$

where $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are two families of (possibly nonlinear) measurable functions. Observe that if $(\mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{y}))$ is a solution to (3), then $(\mathbf{R}\mathbf{f}(\mathbf{x}), \mathbf{R}\mathbf{g}(\mathbf{y}))$ is also a solution, for any orthogonal matrix $\mathbf{R}$. This ambiguity can be removed by adding the additional constraints $\mathbb{E}[f_i(X)g_j(Y)] = 0, \forall i \neq j$ (see, *e.g.*, Hardoon et al. [2004]). Here we do not pursue this route, and simply focus on one solution among this family of solutions.

Alternating conditional expectations (ACE): The ACE method [Breiman and Friedman, 1985] treats the case of a single projection ($L = 1$), where $\boldsymbol{\mathcal{B}}$ is the class of all zero-mean scalar-valued functions $g(Y)$, and $\boldsymbol{\mathcal{A}}$ is the class of additive models $f(X) = \sum_{\ell=1}^{D_x} \gamma_\ell \phi_\ell(X_\ell)$ with zero-mean scalar-valued functions $\phi_\ell(X_\ell)$. The ACE algorithm minimizes the objective (3) by iteratively computing the conditional expectation of each view given the other. Recently, Makur et al. [2015] extended ACE to multiple dimensions by whitening the vector-valued $\mathbf{f}(X)$ and $\mathbf{g}(Y)$ during each iteration. In practice, the conditional expectations are estimated from training data using nonparametric regression. Since this computationally demanding step has to be repeatedly applied until convergence, ACE and its extensions are impractical to apply on large data.

Kernel CCA (KCCA): In KCCA [Lai and Fyfe, 2000, Akaho, 2001, Melzer et al., 2001, Bach and Jordan, 2002, Hardoon et al., 2004], $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are two reproducing kernel Hilbert spaces (RKHSs) associated with user-specified kernels $k_x(\cdot,\cdot)$ and $k_y(\cdot,\cdot)$. By the representer theorem, the projections can be written in terms of the training samples as $f_\ell(\mathbf{x}) = \sum_{i=1}^N \alpha_{i,\ell} k_x(\mathbf{x}, \mathbf{x}_i)$ and $g_\ell(\mathbf{y}) = \sum_{i=1}^N \beta_{i,\ell} k_x(\mathbf{y}, \mathbf{y}_i)$ with some coefficients $\{\alpha_{i,\ell}\}$ and $\{\beta_{i,\ell}\}$. Letting $\mathbf{K}_x = [k_x(\mathbf{x}_i, \mathbf{x}_j)]$ and $\mathbf{K}_y = [k_y(\mathbf{y}_i, \mathbf{y}_j)]$ denote the $N \times N$ kernel matrices, the optimal coefficients can be computed from the top L eigenvectors of the matrix $(\mathbf{K}_x + r_x\mathbf{I})^{-1}\mathbf{K}_y(\mathbf{K}_y + r_y\mathbf{I})^{-1}\mathbf{K}_x$, where r_x and r_y are positive regularization parameters. Computation of the exact solution is intractable for large datasets due to the memory cost of storing the kernel matrices and the time complexity of solving dense eigenvalue systems. Several approximate techniques have been proposed, largely based on low-rank kernel matrix approximations [Bach and Jordan, 2002, Hardoon et al., 2004, Arora and Livescu, 2012, Lopez-Paz et al., 2014].

Deep CCA (DCCA): In the more recently proposed DCCA approach [Andrew et al., 2013], $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are the families of functions that can be implemented using two deep neural networks of predefined architecture. As a parametric method, DCCA scales better than approximate KCCA for large datasets [Wang et al., 2015b].

Population solutions: Lancaster [1958] studied a variant of problem (3), where $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are the families of *all* measurable functions. This setting may seem too unrestrictive. However, it turns out that in the population setting, the optimal projections are well-defined even without imposing smoothness in any way. Lancaster characterized the optimal (possibly nonlinear) mappings f_i and g_i for one-dimensional X and Y ($D_x = D_y = 1$). In particular, he showed that if X, Y are jointly Gaussian, then the optimal projections are Hermite polynomials. Eagleson [1964] extended this analysis to the Gamma, Poisson, binomial, negative binomial, and hypergeometric distributions. Hannan [1961] gave Lancaster's characterization a functional analysis interpretation, which confirmed its validity also for multi-dimensional views.

Our approach: Lancaster's population solution has never been used for devising a practical CCA algorithm that works with sample data. Here, we revisit Lancaster's result, extend it to a semi-parametric setting, and devise practical algorithms that work with sample data. Clearly, in the finite-sample setting, it is necessary to impose smoothness. Our approach to imposing smoothness is different from KCCA, which formulates the problem as one of finding the optimal smooth solution (in an RKHS) and then approximates it from samples. Here, we first derive the optimal solution among all (not necessarily smooth) measurable functions, and then approximate it by using smoothed versions of the true densities, which we estimate from data. As we show, the resulting algorithm has significant advantages over KCCA.

3 Nonparametric and partially linear CCA

We treat the following two variants of the nonlinear CCA problem (3): (i) *Nonparametric CCA* in which both $\mathcal{A}$ and $\mathcal{B}$ are the sets of all (nonparametric) measurable functions; (ii) *Partially linear CCA* (PLCCA), in which $\mathcal{A}$ is the set of all linear functions $\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$, and $\mathcal{B}$ is the set of all (nonparametric) measurable functions $\mathbf{g}(\mathbf{y})$. We start by deriving closed-form solutions in the population setting, and then plug in an empirical estimate of $p(\mathbf{x}, \mathbf{y})$.

3.1 Nonparametric CCA (NCCA)

Let $\mathcal{A}$ and $\mathcal{B}$ be the sets of all (nonparametric) measurable functions of X and Y, respectively. Note that the coordinates of $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{y})$ are constrained to satisfy $\mathbb{E}[f_i^2(X)] = \mathbb{E}[g_i^2(Y)] = 1$, so that we may write (3) as an optimization problem over the Hilbert spaces

$$\mathcal{H}_x = \left\{ q : \mathbb{R}^{D_x} \to \mathbb{R} \;\middle|\; \mathbb{E}[q^2(X)] < \infty \right\},$$
$$\mathcal{H}_y = \left\{ u : \mathbb{R}^{D_y} \to \mathbb{R} \;\middle|\; \mathbb{E}[u^2(Y)] < \infty \right\},$$

which are endowed with the inner products $\langle q, r\rangle_{\mathcal{H}_x} = \mathbb{E}[q(X)r(X)]$ and $\langle u, v\rangle_{\mathcal{H}_y} = \mathbb{E}[u(Y)v(Y)]$. To do so, we express the correlation between $f_i(X)$ and $g_i(Y)$ as

$$\mathbb{E}[f_i(X)g_i(Y)] = \int f_i(\mathbf{x}) \left(\int g_i(\mathbf{y}) s(\mathbf{x}, \mathbf{y}) p(\mathbf{y}) d\mathbf{y} \right) p(\mathbf{x}) d\mathbf{x} = \langle f_i, \mathcal{S} g_i \rangle_{\mathcal{H}_x}, \tag{4}$$

where[3]

$$s(\mathbf{x}, \mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \tag{5}$$

and $\mathcal{S} : \mathcal{H}_y \to \mathcal{H}_x$ is the operator defined by[4] $(\mathcal{S}u)(\mathbf{x}) = \int u(\mathbf{y}) s(\mathbf{x}, \mathbf{y}) p(\mathbf{y}) d\mathbf{y}$. Thus, problem (3) can be written as

$$\max_{\substack{\langle f_i, f_j\rangle_{\mathcal{H}_x} = \delta_{ij} \\ \langle g_i, g_j\rangle_{\mathcal{H}_y} = \delta_{ij}}} \sum_{i=1}^{L} \langle \mathcal{S} g_i, f_i \rangle_{\mathcal{H}_x}, \tag{6}$$

where δ_{ij} is Kronecker's delta function.

When $\mathcal{S}$ is a compact operator, the solution to problem (6) can be expressed in terms of its SVD (see *e.g.*, [Bolla, 2013, Proposition A.2.8]). Specifically, in this case $\mathcal{S}$ possesses a discrete set of singular values $\sigma_1 \geq \sigma_2 \geq \ldots$ and corresponding left and right singular functions $\psi_i \in \mathcal{H}_x, \phi_i \in \mathcal{H}_y$, and the maximal value of the objective in (6) is precisely $\sigma_1 + \ldots + \sigma_L$ and is attained with

$$f_i(\mathbf{x}) = \psi_i(\mathbf{x}), \quad g_i(\mathbf{y}) = \phi_i(\mathbf{y}). \tag{7}$$

That is, the optimal projections are the singular functions of $\mathcal{S}$ and the canonical correlations are its singular values: $\mathbb{E}[f_i(X)g_i(Y)] = \sigma_i$.

The NCCA solution (7), has several interesting interpretations. First, note that $\log s(\mathbf{x}, \mathbf{y})$ is the *pointwise mutual information* (PMI) between X and Y, which is a common measure of statistical dependence. Since the optimal projections are the top singular functions of $s(\mathbf{x}, \mathbf{y})$, the NCCA solution may be interpreted as an embedding which preserves as much of the (exponentiation of the) PMI between X and Y as possible. Second, note that the operator $\mathcal{S}$ corresponds to the *optimal predictor* (in mean square error sense) of one view based on the other, as $(\mathcal{S}g_i)(\mathbf{x}) = \mathbb{E}[g_i(Y)|X = \mathbf{x}]$ and $(\mathcal{S}^* f_i)(\mathbf{y}) = \mathbb{E}[f_i(X)|Y = \mathbf{y}]$. Therefore, the NCCA projections can also be thought of as approximating the best predictors of each view based on the other. Finally, note that rather than using SVD, the NCCA solution can be also expressed in terms of the *eigen-decomposition* of a certain operator. Specifically, the optimal view

[3] Formally, $s(\mathbf{x}, \mathbf{y})$ is the Radon-Nikodym derivative of the joint probability measure w.r.t. the product of marginal measures, assuming the former is absolutely continuous w.r.t. the latter.

[4] To see that $\mathcal{S}u \in \mathcal{H}_x$ for every $u \in \mathcal{H}_y$, note that $(\mathcal{S}u)(\mathbf{x}) = \mathbb{E}[u(Y)|X = \mathbf{x}]$ and thus $\|\mathcal{S}u\|^2_{\mathcal{H}_x} = \mathbb{E}[(\mathbb{E}[u(Y)|X])^2] \leq \mathbb{E}[u^2(Y)] = \|u\|^2_{\mathcal{H}_y} < \infty$.

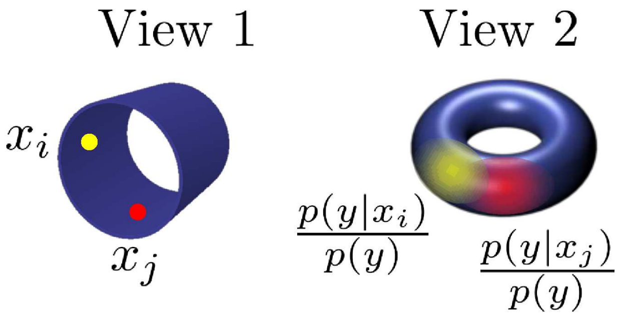

Figure 1: In NCCA, the similarity $k(\mathbf{x}, \mathbf{x}')$ between $\mathbf{x}$ and $\mathbf{x}'$ in view 1 is given by the inner product between the functions $p(\mathbf{y}|\mathbf{x})/p(\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x}')/p(\mathbf{y})$ over the domain of view 2.

1 projections are the eigenfunctions of $\mathcal{K} = \mathcal{S}\mathcal{S}^*$ (and the view 2 projections are eigenfunctions of $\mathcal{S}^*\mathcal{S}$), which is the operator defined by $(\mathcal{K}q)(\mathbf{x}) = \int q(\mathbf{x})k(\mathbf{x}, \mathbf{x}')p(\mathbf{x})d\mathbf{x}$, with the kernel

$$k(\mathbf{x}, \mathbf{x}') = \int s(\mathbf{x}, \mathbf{y})s(\mathbf{x}', \mathbf{y})p(\mathbf{y})d\mathbf{y}. \tag{8}$$

This shows that NCCA resembles other spectral dimensionality reduction algorithms, in that the projections are the eigenfunctions of some kernel. However, in NCCA, the kernel is not specified by the user. From (8), we see that $k(\mathbf{x}, \mathbf{x}')$ corresponds to the inner product between $s(\mathbf{x}, \cdot)$ and $s(\mathbf{x}', \cdot)$ (equivalently $p(\mathbf{y}|\mathbf{x})/p(\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x}')/p(\mathbf{y})$). Therefore, as visualized in Fig. 1, in NCCA $\mathbf{x}$ is considered similar to $\mathbf{x}'$ if the conditional distribution of Y given $X = \mathbf{x}$ is similar to that of Y given $X = \mathbf{x}'$.

A sufficient condition for $\mathcal{S}$ to be compact is that it be a Hilbert-Schmidt operator, *i.e.*, that

$$\iint |s(x, y)|^2 p(x)dx\, p(y)dy < \infty.$$

Substituting (5), this condition can be equivalently written as $\mathbb{E}[s(X, Y)] < \infty$. This can be thought of as a requirement that the statistical dependence between X and Y should not be too strong. In this case, the singular values σ_i tend to zero as i tends to ∞. Furthermore, the largest singular value of $\mathcal{S}$ is always $\sigma_1 = 1$ and is associated with the constant functions $\psi_1(\mathbf{x}) = \phi_1(\mathbf{y}) = 1$. To see this, note that for any pair of unit-norm functions $\psi \in \mathcal{H}_x, \phi \in \mathcal{H}_y$, we have that $\langle \psi, \mathcal{S}\phi \rangle_{\mathcal{H}_x} = \mathbb{E}[\psi(X)\phi(Y)] \leq \sqrt{\mathbb{E}[\psi^2(X)]\mathbb{E}[\phi^2(Y)]} = 1$ and this bound is clearly attained with $\psi(\mathbf{x}) = \phi(\mathbf{y}) = 1$. Thus, we see that the first nonlinear CCA projections are always constant functions $f_1(\mathbf{x}) = g_1(\mathbf{y}) = 1$. These projections are perfectly correlated, but carry no useful information on the common variability in X and Y. Therefore, in practice, we discard them. The rest of the projections are orthogonal to the first and therefore have zero mean: $\mathbb{E}[f_\ell(X)] = \mathbb{E}[g_\ell(Y)] = 0$ for $\ell \geq 2$.

3.2 Partially linear CCA (PLCCA)

The above derivation of NCCA can be easily adapted to cases in which $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are different families of functions. As an example, we next derive PLCCA, in which $\boldsymbol{\mathcal{A}}$ is the set of all *linear* functions of X while $\boldsymbol{\mathcal{B}}$ is still the set of all (nonparametric) measurable functions of Y.

Let $\mathbf{f}(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$, where $\mathbf{W} \in \mathbb{R}^{D_x \times L}$. In this case, the constraint that $\mathbb{E}[\mathbf{f}(X)\mathbf{f}(X)^\top] = \mathbf{I}$ corresponds to the restriction that $\mathbf{W}^\top \boldsymbol{\Sigma}_{xx} \mathbf{W} = \mathbf{I}$. By changing variables to $\tilde{\mathbf{W}} = \boldsymbol{\Sigma}_{xx}^{1/2}\mathbf{W}$ and denoting the ith column of $\tilde{\mathbf{W}}$ by $\tilde{\mathbf{w}}_i$, the constraint simplifies to $\tilde{\mathbf{w}}_i^\top \tilde{\mathbf{w}}_j = \delta_{ij}$. Furthermore, we can write the objective (3) as

$$\sum_{i=1}^{L} \mathbb{E}\left[\tilde{\mathbf{w}}_i^\top \boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}} X g_i(Y)\right] = \sum_{i=1}^{L} \tilde{\mathbf{w}}_i^\top \mathbb{E}\left[\boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}} \mathbb{E}[X|Y]\, g_i(Y)\right] = \sum_{i=1}^{L} \tilde{\mathbf{w}}_i^\top \mathcal{S}_{\text{PL}} g_i, \tag{9}$$

where $\mathcal{S}_{\text{PL}} : \mathcal{H}_y \to \mathbb{R}^{D_x}$ is the operator defined by $\mathcal{S}_{\text{PL}} u = \boldsymbol{\Sigma}_{xx}^{-1/2} \int \mathbb{E}[X|Y = \mathbf{y}]\, u(\mathbf{y})p(\mathbf{y})d\mathbf{y}$. Therefore, Problem

(3) now takes the form

$$\max_{\substack{\tilde{\mathbf{w}}_i^\top \tilde{\mathbf{w}}_j=\delta_{ij}\\ \langle g_i,g_j\rangle_{\mathcal{H}_y}=\delta_{ij}}} \sum_{i=1}^{L} \tilde{\mathbf{w}}_i^\top \mathcal{S}_{\text{PL}}\, g_i, \tag{10}$$

which is very similar to (6). Note that here the domain of the operator $\mathcal{S}_{\text{PL}}$ is infinite dimensional (the space $\mathcal{H}_y$), but its range is finite-dimensional (the Euclidian space $\mathbb{R}^{D_x}$). Therefore, $\mathcal{S}_{\text{PL}}$ is guaranteed to be compact without any restrictions on the joint probability $p(\mathbf{x},\mathbf{y})$. The optimal $\tilde{\mathbf{w}}_i$'s are thus the top L singular vectors of $\mathcal{S}_{\text{PL}}$ and the optimal g_i's are the top L right singular functions of $\mathcal{S}_{\text{PL}}$.

The PLCCA solution can be expressed in more convenient form by noting that the optimal $\tilde{\mathbf{w}}_i$'s are also the top L eigenvectors of the matrix $\mathbf{K}_{\text{PL}} = \mathcal{S}_{\text{PL}}\mathcal{S}_{\text{PL}}^*$, given by

$$\mathbf{K}_{\text{PL}} = \mathbb{E}\left[\left(\boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}}\mathbb{E}[X|Y]\right)\left(\boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}}\mathbb{E}[X|Y]\right)^{\top}\right] = \boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}}\boldsymbol{\Sigma}_{\hat{x}\hat{x}}\boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}}. \tag{11}$$

Here, $\boldsymbol{\Sigma}_{\hat{x}\hat{x}} = \mathbb{E}[\mathbb{E}[X|Y]\mathbb{E}[X|Y]^\top]$ denotes the covariance of $\hat{X} = \mathbb{E}[X|Y]$, the optimal predictor of X from Y. Denoting the top L eigenvectors of $\mathbf{K}_{\text{PL}}$ by $\mathbf{U}$, and reverting the change of variables, we get that $\mathbf{W} = \boldsymbol{\Sigma}_{xx}^{-1/2}\mathbf{U}$.

Having determined the optimal $\mathbf{f}(\mathbf{x}) = \mathbf{W}^\top\mathbf{x}$, we can compute the optimal $\mathbf{g}(\mathbf{y})$ using the following lemma[5].

Lemma 3.1. *Assume that* $\mathbb{E}[\mathbb{E}[\mathbf{f}(X)|Y]\mathbb{E}[\mathbf{f}(X)|Y]^\top]$ *is a non-singular matrix. Then the function* $\mathbf{g}$ *optimizing* (3) *for a fixed* $\mathbf{f}$ *is given by*

$$\mathbf{g}(Y) = \left(\mathbb{E}\left[\mathbb{E}[\mathbf{f}(X)|Y]\mathbb{E}[\mathbf{f}(X)|Y]^\top\right]\right)^{-\frac{1}{2}}\mathbb{E}[\mathbf{f}(X)|Y]. \tag{12}$$

Substituting $\mathbf{f}(\mathbf{x}) = \mathbf{W}^\top\mathbf{x} = \mathbf{U}^\top\boldsymbol{\Sigma}_{xx}^{-1/2}\mathbf{x}$ into (12), we obtain that the partially linear CCA projections are

$$\mathbf{W}^\top X = \mathbf{U}^\top\boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}}X, \quad \mathbf{g}(Y) = \mathbf{D}^{-\frac{1}{2}}\mathbf{U}^\top\boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}}\hat{X}, \tag{13}$$

where $\mathbf{D}$ is the diagonal $L \times L$ matrix that has the top L eigenvalues of $\mathbf{K}_{\text{PL}}$ on its diagonal.

Comparing (13) with (2), we see that PLCCA has the exact same form as CCA. The only difference is that here $\hat{X}$ is the optimal *nonlinear predictor* of X from Y (a nonlinear function of Y), whereas in CCA, $\hat{X}$ corresponded to the best linear predictor of X from Y (a linear function of Y).

3.3 Practical implementations

The NCCA and PLCCA solutions require knowing the joint probability density $p(\mathbf{x},\mathbf{y})$ of the views. Given a set of training data $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^N$ drawn independently from $p(\mathbf{x},\mathbf{y})$, we can estimate $p(\mathbf{x},\mathbf{y})$ and plug it into our formulas. There are many ways of estimating this density. We next present the algorithms resulting from using one particular choice, namely the kernel density estimates (KDEs)

$$\begin{aligned}
\hat{p}(\mathbf{x}) &= \tfrac{1}{N}\sum\nolimits_{i=1}^{N} w\left(\|\mathbf{x}-\mathbf{x}_i\|^2/\sigma_x^2\right), \\
\hat{p}(\mathbf{y}) &= \tfrac{1}{N}\sum\nolimits_{i=1}^{N} w\left(\|\mathbf{y}-\mathbf{y}_i\|^2/\sigma_y^2\right), \\
\hat{p}(\mathbf{x},\mathbf{y}) &= \tfrac{1}{N}\sum\nolimits_{i=1}^{N} w\left(\|\mathbf{x}-\mathbf{x}_i\|^2/\sigma_x^2 + \|\mathbf{y}-\mathbf{y}_i\|^2/\sigma_y^2\right),
\end{aligned} \tag{14}$$

where $w(t) \propto e^{-t/2}$ is the Gaussian kernel, and σ_x and σ_y are the kernel widths of the two views.

We note that, theoretically, KDEs suffer from the curse of dimensionality, and use of other density estimation methods is certainly possible. However, we make two important observations. First, real-world data sets often have low-dimensional manifold structure, and the KDE accuracy is affected only by the intrinsic dimensionality. As shown

[5] A simpler version of this lemma, in which $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ and $\mathbf{g}$ is linear, appeared in Eldar and Oppenheim [2003]. The proof of Lemma 3.1 is provided in the Supplemntary Material and follows closely that of [Eldar and Oppenheim, 2003, Theorem 1].

in [Ozakin and Gray, 2009], if the data lies on an r-dimensional manifold, then the KDE converges to the true density at a rate of[6] $\mathcal{O}(n^{-\frac{4}{r+4}})$. Indeed, KDEs have been shown to work well in practice in relatively high dimensions [Georgescu et al., 2003], as is also confirmed in our experiments. Second, the NCCA algorithm resulting from working with KDEs involves the same Gaussian affinity matrices used in (Gaussian kernel) KCCA. Thus, intuitively, the amount of smoothness required for obtaining accurate results in high dimensions is similar for NCCA and KCCA. Nevertheless, NCCA has a clear advantage over KCCA in terms of both performance and computation.

PLCCA Using the above KDEs, the conditional expectation $\hat{\mathbf{x}}(\mathbf{y}) = \mathbb{E}[X|Y = \mathbf{y}]$ needed for the PLCCA solution (13) reduces to the Nadaraya-Watson nonparametric regression [Nadaraya, 1964, Watson, 1964]

$$\hat{\mathbf{x}}(\mathbf{y}) = \frac{\sum_{i=1}^{N} w\left(\|\mathbf{y}-\mathbf{y}_i\|^2/\sigma_y^2\right)\mathbf{x}_i}{\sum_{i=1}^{N} w\left(\|\mathbf{y}-\mathbf{y}_i\|^2/\sigma_y^2\right)}. \tag{15}$$

The population moments $\mathbf{\Sigma}_{\hat{x}\hat{x}} = \mathbb{E}[\hat{X}\hat{X}^\top]$ and $\mathbf{\Sigma}_{xx} = \mathbb{E}[XX^\top]$ can then be replaced by the empirical moments of $\{\hat{\mathbf{x}}(\mathbf{y}_i)\}$ and $\{\mathbf{x}_i\}$.

NCCA The quadratic form $\langle \mathcal{S}g_i, f_i\rangle_{\mathcal{H}_x}$ is given by $\mathbb{E}[(\mathcal{S}g_i)(X)f_i(X)]$ and is approximated by $\frac{1}{N}\sum_{\ell=1}^{N}(\mathcal{S}g_i)(\mathbf{x}_\ell)f(\mathbf{x}_\ell)$. Furthermore, $(\mathcal{S}g_i)(\mathbf{x}_\ell)$ is equal to $\mathbb{E}[s(\mathbf{x}_\ell, Y)g_i(Y)]$ and thus can be approximated by $\frac{1}{N}\sum_{m=1}^{N} s(\mathbf{x}_\ell,\mathbf{y}_m)g(\mathbf{y}_m)$, where $s(\mathbf{x}_\ell,\mathbf{y}_m) = \frac{p(\mathbf{x}_\ell,\mathbf{y}_m)}{p(\mathbf{x}_\ell)p(\mathbf{y}_m)}$. Therefore, defining the $N \times N$ matrix $\mathbf{S} = [s(\mathbf{x}_\ell,\mathbf{y}_m)]$, and stacking the projections of the data points into the $N \times 1$ vectors $\mathbf{f}_i = \frac{1}{\sqrt{N}}(f_i(\mathbf{x}_1),\ldots,f_i(\mathbf{x}_N))^\top$ and $\mathbf{g}_i = \frac{1}{\sqrt{N}}(g_i(\mathbf{y}_1),\ldots,g_i(\mathbf{y}_N))^\top$, the NCCA objective can be approximated by $\frac{1}{N}\sum_{i=1}^{L}\mathbf{f}_i^\top\mathbf{S}\mathbf{g}_i$. Similarly, the NCCA constraints become $\mathbf{f}_i^\top\mathbf{f}_j = \mathbf{g}_i^\top\mathbf{g}_j = \delta_{ij}$. This implies that the optimal $\mathbf{f}_i$ and $\mathbf{g}_i$ are the top L singular vectors of $\mathbf{S}$. Recall that in the continuous formulation, the first pair of singular functions are constant functions. Therefore, in practice, we compute the top $L+1$ singular vectors of $\mathbf{S}$ and discard the first one. To construct the matrix $\mathbf{S}$ we use the kernel density estimates (14) for joint and marginal probability distributions over $(\mathbf{x},\mathbf{y})$.

The NCCA implementation, with the specific choice of Gaussian KDEs, is given in Algorithm 1. If the input dimensionality is too high, we first perform PCA on the inputs for more robust density estimates. To make our algorithm computationally efficient, we truncate the Gaussian affinities $\mathbf{W}_{ij}^x$ to zero if $\mathbf{x}_i$ is not within the k-nearest neighbors of $\mathbf{x}_j$ (similarly for view 2). This leads to a sparse matrix $\mathbf{S}$, whose SVD can be computed efficiently.

To obtain out-of-sample mapping for a new view 1 test sample $\mathbf{x}$, we use the Nyström method [Williams and Seeger, 2001], which avoids recomputing SVD. Specifically, recall that the view 1 projections are the eigenfunctions of the positive definite kernel $k(\mathbf{x},\mathbf{x}')$ of (8). Computing this kernel function between $\mathbf{x}$ and the training samples leads to (notice the corresponding view 2 input of $\mathbf{x}$ is not needed)

$$k(\mathbf{x},\mathbf{x}_i) = \sum_{m=1}^{N} s(\mathbf{x},\mathbf{y}_m)s(\mathbf{x}_i,\mathbf{y}_m). \tag{16}$$

Thus, applying the Nyström method, the projections of $\mathbf{x}$ can be approximated as

$$f_i(\mathbf{x}) = \frac{1}{\sigma_i^2}\sum_{n=1}^{N} k(\mathbf{x},\mathbf{x}_n)f_i(\mathbf{x}_n) = \frac{1}{\sigma_i}\sum_{n=1}^{N} s(\mathbf{x},\mathbf{y}_n)g_i(\mathbf{y}_n)$$

for $i = 1,\ldots,L+1$, where σ_i is the ith singular value of $\mathbf{S}$. The second equality follows from substituting (16) and using the fact that $\mathbf{f}_i$ and $\mathbf{g}_i$ are singular vectors of $\mathbf{S}$. Note again that since the affinity matrices are sparse, the mappings are computed via fast sparse matrix multiplication.

Relationship with KCCA Notice that NCCA is not equivalent to KCCA with any kernel. KCCA requires two kernels, each of which only sees one view; the NCCA kernel (8) depends on both views through their joint distribution. In terms of practical implementation, our KDE-based NCCA solves a different eigenproblem and does not involve any full matrix inverses. Indeed, both methods compute the SVD of the matrix $\mathbf{Q}_x^{-1}\mathbf{W}^x\mathbf{W}^y\mathbf{Q}_y^{-1}$. However, in NCCA,

[6] This requires normalizing the KDE differently, but the scaling cancels out in $s(\mathbf{x},\mathbf{y}) = p(\mathbf{x},\mathbf{y})/p(\mathbf{x})p(\mathbf{y})$.

Algorithm 1 Nonparametric CCA with Gaussian KDE

```
Input: Training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, test sample $\mathbf{x}$.
Construct affinity matrices for each view
    $\mathbf{W}^x_{ij} \leftarrow \exp\left\{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma_x^2}\right\}, \ \mathbf{W}^y_{ij} \leftarrow \exp\left\{-\frac{\|\mathbf{y}_i-\mathbf{y}_j\|^2}{2\sigma_y^2}\right\}.$
Normalize $\mathbf{W}^x$ to be right stochastic and $\mathbf{W}^y$ to be left stochastic, i.e.,
    $\mathbf{W}^x_{ij} \leftarrow \mathbf{W}^x_{ij} / \sum_{l=1}^N \mathbf{W}^x_{il}, \quad \mathbf{W}^y_{ij} \leftarrow \mathbf{W}^y_{ij} / \sum_{l=1}^N \mathbf{W}^y_{lj}.$
Form the matrix $\mathbf{S} \leftarrow \mathbf{W}^x \mathbf{W}^y$.
Compute $\mathbf{U} \in \mathbb{R}^{N \times (L+1)}, \mathbf{V} \in \mathbb{R}^{N \times (L+1)}$, the first $L+1$ left and right singular vectors of $\mathbf{S}$, with corresponding
    singular values $\sigma_1, \ldots, \sigma_{L+1}$.
Output: At train time, compute the projections $i = 1, \ldots, L+1$ of the training samples as
    $f_i(\mathbf{x}_n) \leftarrow \sqrt{N}\mathbf{U}_{n,i}, \quad g_i(\mathbf{y}_n) \leftarrow \sqrt{N}\mathbf{V}_{n,i}.$
    At test time, calculate a new row of $\mathbf{W}^x$ for $\mathbf{x}$ as
    $\mathbf{W}^x_{N+1,j} \leftarrow \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_j\|^2}{2\sigma_x^2}\right\},$
    $\mathbf{W}^x_{N+1,j} \leftarrow \mathbf{W}^x_{N+1,j} / \sum_{l=1}^N \mathbf{W}^x_{N+1,l}$
    and a new row of $\mathbf{S}$ as $\mathbf{S}_{N+1} \leftarrow \mathbf{W}^x_{N+1}\mathbf{W}^y$, and compute the projections of $\mathbf{x}$ as
    $f_i(\mathbf{x}) \leftarrow \frac{1}{\sigma_i} \sum_{n=1}^N \mathbf{S}_{N+1,n} \, g_i(\mathbf{y}_n), \quad i = 1, \ldots, L+1.$
```

$\mathbf{Q}_x, \mathbf{Q}_y$ are diagonal matrices containing the sums of rows/columns of $\mathbf{W}^x/\mathbf{W}^y$, whereas in KCCA, $\mathbf{Q}_x = \mathbf{W}^x + r_x\mathbf{I}$, $\mathbf{Q}_y = \mathbf{W}^y + r_y\mathbf{I}$, for some positive regularization parameters r_x, r_y. Moreover, in NCCA this factorization gives the projections, whereas in KCCA it gives the coefficients in the RKHS.

An additional key distinction is that NCCA does not require regularization in order to be well defined. In contrast, KCCA must use regularization, as otherwise the matrix it factorizes collapses to the identity matrix, and the resulting projections are meaningless. This is due to the fact that KCCA attempts to estimate covariances in the infinite-dimensional feature space, whereas NCCA is based on estimating probability densities in the primal space.

The resulting computational differences are striking. The number of training samples N is often such that the $N \times N$ matrices in either NCCA or KCCA cannot even be stored in memory. However, these matrices are sparse, with only kN entries if we retain k neighbors. Therefore, in NCCA the storage problem is alleviated and matrix multiplication and eigendecomposition are $O(kN^2)$ operations instead of $O(N^3)$. In KCCA, one cannot take advantage of truncated kernel affinities, because of the need to compute the inverses of kernel matrices, which are in general not sparse, so direct computation is often infeasible in terms of both memory and time. Low-rank KCCA approximations (as used in our experiments below) with rank M have a time complexity $O(M^3 + M^2N)$, which is still challenging with typical ranks in the thousands or tens of thousands.

4 Related work

Several recent multi-view learning algorithms use products or sums of single-view affinity matrices, diffusion matrices, or Markov transition matrices. The combined kernels constructed in these methods resemble our matrix $\mathbf{S} = \mathbf{W}^x\mathbf{W}^y$. Such an approach has been used, for example, for multi-view spectral clustering [de Sa, 2005, Zhou and Burges, 2007, Kumar et al., 2011], metric fusion [Wang et al., 2012], common manifold learning [Lederman and Talmon, 2014], and

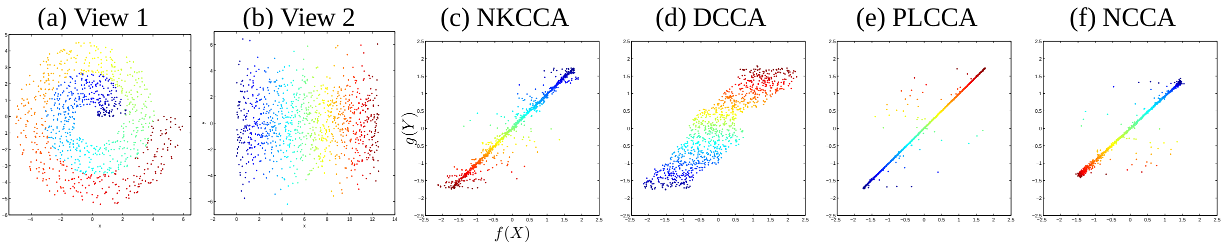

Figure 2: Dimensionality reduction obtained by nonlinear CCAs on a synthetic dataset.

multi-view nonlinear system identification [Boots and Gordon, 2012]. Note, however, that in NCCA the matrix $\mathbf{S}$ corresponds to the product $\mathbf{W}^x\mathbf{W}^y$ *only when using a separable Gaussian kernel* for estimating the joint density $p(\mathbf{x}, \mathbf{y})$. If a non-separable density estimate is used, then the matrix $\mathbf{S}$ no longer resembles the previously proposed multi-view kernels. Furthermore, although algorithmically similar, NCCA arises from a completely different motivation: It maximizes the correlation between the views, whereas these other methods do not.

5 Experiments

In the following experiments, we compare PLCCA/NCCA with linear CCA, two kernel CCA approximations using random Fourier features (FKCCA, Lopez-Paz et al. [2014]) and Nyström approximation (NKCCA, Williams and Seeger [2001]) as described in Wang et al. [2015b], and deep CCA (DCCA, Andrew et al. [2013]).

Illustrative example We begin with the 2D synthetic dataset (1000 training samples) in Fig. 2(a,b), where samples of the two input manifolds are colored according to their common degree of freedom. Clearly, a linear mapping in view 1 cannot unfold the manifold to align the two views, and linear CCA indeed fails (results not shown). We extract a one-dimensional projection for each view using different nonlinear CCAs, and plot the projection $g(\mathbf{y})$ vs. $f(\mathbf{x})$ of test data (a different set of 1000 random samples from the same distribution) in Fig. 2(c-f). Since the second view is essentially a linear manifold (plus noise), for NKCCA we use a linear kernel in view 2 and a Gaussian kernel in view 1, and for DCCA we use a linear network for view 2 and two hidden layers of 512 ReLU units for view 1. Overall, NCCA achieves better alignment of the views while compressing the noise (variations not described by the common degree of freedom). While DCCA also succeeds in unfolding the view 1 manifold, it fails to compress the noise.

Table 1: Total canonical correlation on the XRMB 'JW11-s' test set and run time of each algorithm. The maximum possible canonical correlation is 112 (the view 2 input dimensionality). PLCCA/NCCA run time is given as neighbor search time + optimization time.

	CCA	FKCCA	NKCCA	DCCA	PLCCA	NCCA
Total Correlation	21.7	99.2	105.6	107.6	79.4	107.9
Run Time (sec)	2.3	510.7	1449.8	10044.0	40.7 + 0.8	69.4 + 79.0

X-Ray Microbeam Speech Data The University of Wisconsin X-Ray Micro-Beam (XRMB) corpus [Westbury, 1994] consists of simultaneously recorded speech and articulatory measurements. Following Andrew et al. [2013] and Lopez-Paz et al. [2014], the acoustic view inputs are 39D Mel-frequencey cepstral coefficients and the articulatory view inputs are horizontal/vertical displacement of 8 pellets attached to different parts of the vocal tract, each then concatenated over a 7-frame context window, for speaker 'JW11'. As in [Lopez-Paz et al., 2014], we randomly shuffle the frames and generate splits of $30K/10K/11K$ frames for training/tuning/testing, and we refer to the result as the 'JW11-s' setup (random splits better satisfy the i.i.d. assumption of train/tune/test data than splits by utterances as in [Andrew et al., 2013]). We extract $112D$ projections with each algorithm and measure the total correlation between the two views of the test set, after an additional $112D$ linear CCA. As in prior work, for both FKCCA and NKCCA we use rank-6000 approximations for the kernel matrices; for DCCA we use two ReLU [Nair and Hinton, 2010] hidden layers of width $1800/1200$ for view 1/2 respectively and run stochastic optimization with minibatch size 750

Table 2: Clustering accuracy, SVM error rate, and run times (same format as in Table 1) on the noisy MNIST projected test set.

	Baseline	CCA	FKCCA	NKCCA	DCCA	PLCCA	NCCA
Clustering Accuracy (%)	47.1	72.3	95.6	96.7	99.1	98.4	99.2
Classification Error (%)	13.3	18.9	3.9	3.1	0.9	1.3	0.7
Run Time (sec)	0	161.9	1270.1	5890.3	16212.7	4932.1 + 5.7	9052.6 + 38.3

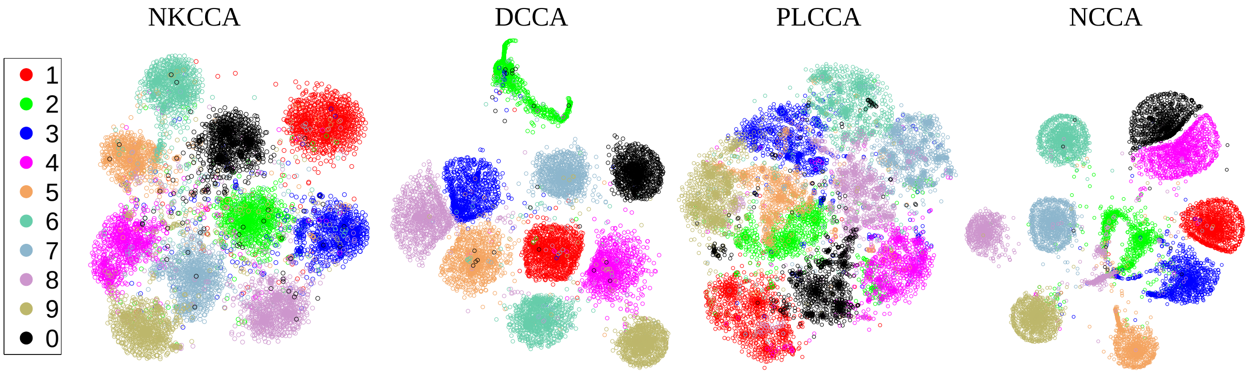

Figure 3: 2D t-SNE visualization of the noisy MNIST test set.

as in [Wang et al., 2015a] for 100 epochs. Kernel widths for FKCCA/NKCCA, learning rate and momentum for DCCA, kernel widths and neighborhood sizes for NCCA/PLCCA are selected by grid search based on total tuning set correlation. Sensitivity to their values is mild over a large range; *e.g.*, setting the kernel widths to 30-60% of the sample L_2 norm gives similarly good results. For NCCA/PLCCA, input dimensionalities are first reduced by PCA to 20% of the original ones (except that PLCCA does not apply PCA for view 2 in order to extract a $112D$ projection). The total correlation achieved by each algorithm is given in Table 1. We also report the running time (in seconds) of the algorithms (measured with a single thread on a workstation with a 3.2GHz CPU and 56G main memory), each using its optimal hyperparameters, and including the time for exact 15-nearest neighbor search for NCCA/PLCCA. Overall, NCCA achieves the best canonical correlation while being much faster than the other nonlinear methods.

Noisy MNIST handwritten digits dataset We now demonstrate the algorithms on a noisy MNIST dataset, generated identically to that of Wang et al. [2015b] but with a larger training set. View 1 inputs are randomly rotated images (28×28, gray scale) from the original MNIST dataset [LeCun et al., 1998], and the corresponding view 2 inputs are randomly chosen images with the same identity plus additive uniform pixel noise. We generate $450K/10K/10K$ pairs of images for training/tuning/testing (Wang et al. [2015b] uses a $50K$-pair training set). This dataset satisfies the multi-view assumption that given the label, the views are uncorrelated, so that the most correlated subspaces should retain class information and exclude the noise. Following Wang et al. [2015b], we extract a low-dimensional projection of the view 1 images with each algorithm, run spectral clustering to partition the splits into 10 classes (with clustering parameters tuned as in [Wang et al., 2015b]), and compare the clustering with ground-truth labels and report the clustering accuracy. We also train a one-vs.-one linear SVM [Chang and Lin, 2011] on the projections with highest cluster accuracy for each algorithm (we reveal labels of 10% of the training set for fast SVM training) and report the classification error rates. The tuning procedure is as for XRMB except that we now select the projection dimensionality from $\{10, 20, 30\}$. For NCCA/PLCCA we first reduce dimensionality to 100 by PCA for density estimation and exact nearest neighbor search, and use a randomized algorithm [Halko et al., 2011] to compute the SVD of the $450K \times 450K$ matrix $\mathbf{S}$; for RKCCA/NKCCA we use an approximation rank of 5000; for DCCA we use 3 ReLU hidden layers of 1500 units in each view and train with stochastic optimization of minibatch size 4500. Clustering and classification results on the original 784D view 1 inputs are recorded as the baseline. Table 2 shows the clustering accuracy and classification error rates on the test set, as well as training run times, and Figure 3 shows t-SNE embeddings [van der Maaten and Hinton, 2008] of several algorithms with their optimal hyper-parameters. NCCA and DCCA achieve near perfect class separation.

Discussion Several points are worth noting regarding the experiments. First, the computation for NCCA and PLCCA is dominated by the exact kNN search; approximate search [Arya et al., 1998, Andoni and Indyk, 2006] should make NCCA/PLCCA much more efficient. Second, we have not explored the space of choices for density estimates; alternative choices, such as adaptive KDE [Terrell and Scott, 1992], could also further improve performance. Our current choice of KDE would seem to require large training sets for high-dimensional problems. Indeed, with less training data we do observe a drop in performance, but NCCA still outperforms KCCA; for example, using a 50K subset of the MNIST training set—an order of magnitude less data—the classification error rates when using FKCCA/NKCCA/DCCA/NCCA are 5.9/5.2/2.9/4.7%.

6 Conclusion

We have presented closed-form solutions to the nonparametric CCA (NCCA) and partially linear CCA (PLCCA) problems. As opposed to kernel CCA, which restricts the nonparametric projections to lie in a predefined RKHS, we have addressed the unconstrained setting. We have shown that the optimal nonparametric projections can be obtained from the SVD of a kernel defined via the pointwise mutual information between the views. This leads to a simple algorithm that outperforms KCCA and matches deep CCA on multiple datasets, while being more computationally efficient than either for moderate-sized data sets. Future work includes leveraging approximate nearest neighbor search and alternative density estimates.

A Proof of Lemma 3.1

Let the eigen-decomposition of the second-order moment of $\mathbb{E}[\mathbf{f}(X)|Y]$ be $\mathbb{E}[\mathbb{E}[\mathbf{f}(X)|Y]\mathbb{E}[\mathbf{f}(X)|Y]^\top] = \mathbf{ADA}^\top$ and define $U = \mathbf{A}^\top\mathbb{E}[\mathbf{f}(X)|Y]$ and $\tilde{\mathbf{g}}(Y) = \mathbf{A}^\top\mathbf{g}(Y)$. Then the objective in (3) can be written as $\mathbb{E}[\mathbf{f}(X)^\top\mathbf{g}(Y)] = \mathbb{E}[\mathbb{E}[\mathbf{f}(X)|Y]^\top\mathbf{g}(Y)] = \mathbb{E}[(\mathbf{A}^\top\mathbb{E}[\mathbf{f}(X)|Y])^\top(\mathbf{A}^\top\mathbf{g}(Y))] = \mathbb{E}[U^\top\tilde{\mathbf{g}}(Y)]$. Similarly, the constraint $\mathbf{I} = \mathbb{E}[\mathbf{g}(Y)\mathbf{g}(Y)^\top]$ can be expressed as $\mathbf{I} = \mathbf{A}^\top\mathbf{A} = \mathbb{E}[(\mathbf{A}^\top\mathbf{g}(Y))(\mathbf{A}^\top\mathbf{g}(Y))^\top] = \mathbb{E}[\tilde{\mathbf{g}}(Y)\tilde{\mathbf{g}}(Y)^\top]$. Therefore, the optimization problem (3) can be written in terms of $\tilde{\mathbf{g}}$ as

$$\max_{\tilde{\mathbf{g}}} \mathbb{E}\left[U^\top\tilde{\mathbf{g}}(Y)\right] \quad \text{s.t.} \quad \mathbb{E}\left[\tilde{\mathbf{g}}(Y)\tilde{\mathbf{g}}(Y)^\top\right] = \mathbf{I}. \tag{17}$$

Our objective is the sum of correlations in all L dimensions. Let us consider the correlation in the jth dimension. From the Cauchy-Schwartz inequality, we have

$$\mathbb{E}[U_j\tilde{g}_j(Y)] \le \sqrt{\mathbb{E}\left[U_j^2\right]\mathbb{E}[\tilde{g}_j(Y)^2]} = \sqrt{\mathbb{E}\left[U_j^2\right]}$$

with equality if and only if $\tilde{g}_j(Y) = c_jU_j$ for some scalar c_j with probability 1. Note that choosing each $\tilde{g}_j(Y)$ to be proportional to U_j is valid, since the dimensions of U are uncorrelated (as $\mathbb{E}[UU^\top] = \mathbf{A}^\top\mathbb{E}\left[\mathbb{E}[\mathbf{f}(X)|Y]\mathbb{E}[\mathbf{f}(X)|Y]^T\right]\mathbf{A} = \mathbf{D}$). In order for each $\tilde{g}_j(Y)$ to have unit second order moment, we must have $c_j = 1/\sqrt{\mathbb{E}[U_j^2]} = 1/\sqrt{\mathbf{D}_{jj}}$. Therefore, $\tilde{\mathbf{g}}(Y) = \mathbf{D}^{-1/2}U$ so that $\mathbf{g}(Y) = \mathbf{AD}^{-\frac{1}{2}}\mathbf{A}^\top U = (\mathbb{E}[\mathbb{E}[\mathbf{f}(X)|Y]\mathbb{E}[\mathbf{f}(X)|Y]^\top])^{-1/2}\mathbb{E}[\mathbf{f}(X)|Y]$, proving the lemma.

Acknowledgement

Thanks to Nathan Srebro, Ryota Tomioka, and Yochai Blau for fruitful discussions. This research was supported by NSF grant IIS-1321015. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency.

References

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

A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proc. of the 30th Int. Conf. Machine Learning (ICML 2013)*, pages 1247–1255, 2013.

R. Arora and K. Livescu. Kernel CCA for multi-view learning of acoustic features using articulatory measurements. In *Symposium on Machine Learning in Speech and Language Processing (MLSLP)*, 2012.

R. Arora and K. Livescu. Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains. In *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP'13)*, 2013.

S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

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

F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Dept. of Statistics, University of California, Berkeley, 2005.

S. Balakrishnan, K. Puniyani, and J. Lafferty. Sparse additive functional and kernel CCA. In *Proc. of the 29th Int. Conf. Machine Learning (ICML 2012)*, pages 911–918, 2012.

M. Bolla. *Spectral Clustering and Biclustering: Learning Large Graphs and Contingency Tables*. John Wiley & Sons, 2013.

B. Boots and G. Gordon. Two manifold problems with applications to nonlinear system identification. In *Proc. of the 29th Int. Conf. Machine Learning (ICML 2012)*, pages 623–630, 2012.

M. Borga. Canonical correlation: A tutorial. 2001.

L. Breiman and J. H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598, 1985.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2(3):27, 2011.

G. Chechik, A. Globerson, N. Tishby, and Y. Weiss. Information bottleneck for Gaussian variables. *Journal of Machine Learning Research*, 6:165–188, 2005.

V. de Sa. Spectral clustering with two views. In *Workshop on Learning with Multiple Views (ICML'05)*, pages 20–27, 2005.

P. Dhillon, D. Foster, and L. Ungar. Multi-view learning of word embeddings via CCA. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24, pages 199–207, 2011.

G. Eagleson. Polynomial expansions of bivariate distributions. *The Annals of Mathematical Statistics*, pages 1208–1215, 1964.

Y. C. Eldar and A. V. Oppenheim. MMSE whitening and subspace whitening. *IEEE Trans. Info. Theory*, 49(7): 1846–1851, 2003.

B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Proc. 9th Int. Conf. Computer Vision (ICCV'03)*, pages 456–463, Nice, France, Oct. 14–17 2003.

Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *European Conference on Computer Vision*, 2014.

N. Halko, P.-G. Martinsson, Y. Shkolnisky, and M. Tygert. An algorithm for the principal component analysis of large data sets. *SIAM J. Sci. Comput.*, 33(5):2580–2594, 2011.

E. J. Hannan. The general theory of canonical correlation and its relation to functional analysis. *Journal of the Australian Mathematical Society*, 2(02):229–242, 1961.

D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

A. Kumar, P. Rai, and H. Daumé III. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24, pages 1413–1421, 2011.

P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *Int. J. Neural Syst.*, 10(5):365–377, 2000.

H. Lancaster. The structure of bivariate distributions. *The Annals of Mathematical Statistics*, pages 719–736, 1958.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

R. R. Lederman and R. Talmon. Common manifold learning using alternating-diffusion. Technical Report YALEU/DCS/TR-1497, 2014.

D. Lopez-Paz, S. Sra, A. Smola, Z. Ghahramani, and B. Schoelkopf. Randomized nonlinear component analysis. In *Proc. of the 31st Int. Conf. Machine Learning (ICML 2014)*, pages 1359–1367, 2014.

A. Makur, F. Kozynski, S.-L. Huang, and L. Zheng. An efficient algorithm for information decomposition and extraction. In *53rd Annual Allerton Conference on Communication, Control, and Computing*, 2015.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

T. Melzer, M. Reiter, and H. Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *Proc. of the 11th Int. Conf. Artificial Neural Networks (ICANN'01)*, pages 353–360, 2001.

E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proc. of the 27th Int. Conf. Machine Learning (ICML 2010)*, pages 807–814, June 21–25 2010.

A. Ozakin and A. Gray. Submanifold density estimation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 1375–1382, 2009.

G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

L. J. P. van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605, 2008.

B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu. Unsupervised metric fusion by cross diffusion. In *Proc. of the 2012 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'12)*, pages 2997–3004, 2012.

W. Wang, R. Arora, K. Livescu, and J. Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP'15)*, 2015a.

W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *Proc. of the 32st Int. Conf. Machine Learning (ICML 2015)*, 2015b.

G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

J. R. Westbury. *X-Ray Microbeam Speech Production Database User's Handbook Version 1.0*, 1994.

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 682–688, 2001.

D. M. Witten and R. J. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, 8(1):1–27, 2009.

D. Zhou and C. J. C. Burges. Spectral clustering and transductive learning with multiple views. In *Proc. of the 24th Int. Conf. Machine Learning (ICML'07)*, pages 1159–1166, 2007.