Large-Scale Sparse Kernel Canonical Correlation Analysis

Anonymous Authors[1]

Abstract

This paper presents gradKCCA, a large-scale sparse non-linear canonical correlation method. Like Kernel Canonical Correlation Analysis (KCCA), our method finds non-linear correlations through kernel functions, but unlike KCCA, our method does not incorporate a kernel matrix, a known bottleneck for scaling up kernel methods. gradKCCA corresponds to solving KCCA with the additional constraint that the canonical projection directions in the kernel-induced feature space have pre-images in the original data space. Firstly, this modification allows us to very efficiently maximize kernel canonical correlation through an alternating projected gradient algorithm working in the original data space. Secondly, we can control the sparsity of the projection directions by constraining the ℓ_1 norm of the pre-images of the projection directions, facilitating the interpretation of the discovered patterns, which is not available through KCCA. Our empirical experiments demonstrate that gradKCCA outperforms state-of-the-art CCA methods in terms of speed and robustness to noise both in simulated and real-world datasets.

1. Introduction

Canonical correlation analysis (CCA) (Hotelling, 1935; 1936) discovers multivariate relations in two-view datasets, by finding linear combinations of variables, the canonical projections, that produce maximum correlation between the two views. The CCA problem can be solved using the singular value decomposition (SVD), or in the form of a standard or generalized eigenvalue problem (Uurtio et al., 2017).

The standard CCA model has been extended to a non-linear setup by kernel methods (Bach & Jordan, 2002; Hardoon et al., 2004; Lopez-Paz et al., 2014; Wang & Livescu, 2016; Uurtio et al., 2018) and deep learning (Andrew et al., 2013). The advantages of kernelizing the CCA problem, or applying a deep neural network, include the possibility to uncover non-linear multivariate relations. However, the non-linear CCA formulations generally come at the expense of losing the information of which of the variables in the original data matrices are relevant, and typically produce dense models, which hinders interpretability and predictive performance when the underlying relations are sparse. For kernel-based CCA, the quadratic size of the kernel matrix in the number of examples also makes scaling up to big data challenging, although this can be solved through approximations such as random Fourier features (Lopez-Paz et al., 2014) and Nyström approximation (Uurtio et al., 2018).

This paper proposes a new KCCA variant, named gradKCCA, that is designed for sparse non-linear canonical correlation analysis in large datasets. The contributions of the paper are the following:

- A new model for kernel-based non-linear CCA that maximizes canonical correlation in the kernel-induced feature spaces through the gradients of the pre-images of the projection directions, without relying on a kernel matrix. A sparsity-inducing variant of the model is achieved through controlling the ℓ_1 norms of the pre-images of the projection directions.

- A theoretical analysis of the new model compared to KCCA showing that in the training set, gradKCCA correlation can be bounded between the correlation of KCCA and the correlation achieved by an approximate pre-image for KCCA.

- An efficient alternating projected gradient algorithm that has the time-complexity of $O(nd^2)$ per epoch, where n is the number training examples and d is the maximum number of variables in the two views.

- An experimental study including several non-linear CCA variants demonstrating (a) the superior speed and scalability of the method on big data, (b) tolerance to high amounts of irrelevant variables, (c) predictive performance in line with the state-of-the-art on large real world datasets.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Background

In the following, we briefly review the main approaches proposed for non-linear canonical correlation analysis. These methods will be experimentally compared to our proposed method in Section 5.

Notation. Matrices will be denoted by bold upper-case letters (e.g. $\mathbf{X}$, $\mathbf{Y}$ for data matrices) and vectors by bold lower-case letters. $\mathbf{I}$ denotes the identity matrix, $\mathbf{1}$ a is a vector of all ones, $\|.\|_p$ denotes the l_p norm, $^\top$ denotes transpose, and $\langle , \rangle$ denotes inner product. Multiplication and entry-wise multiplication are denoted by $\cdot$ and $\odot$ respectively. We denote sets with non-bold upper-case letters.

2.1. Canonical correlation analysis (CCA)

Canonical correlation analysis (Hotelling, 1935; 1936) finds related variables in two-view datasets. Let the two views be given by $\mathbf{X} \in \mathbb{R}^{n\times p}$ and $\mathbf{Y} \in \mathbb{R}^{n\times q}$, where n, p, and q are the sample size and the numbers of variables in $\mathbf{X}$ and $\mathbf{Y}$ respectively. The related variables are determined from the entries of the coefficient vectors $\mathbf{u} \in \mathbb{R}^p$ and $\mathbf{v} \in \mathbb{R}^q$ that satisfy

$$\max_{\mathbf{u},\mathbf{v}} \quad \frac{\langle \mathbf{Xu}, \mathbf{Yv} \rangle}{||\mathbf{Xu}||_2||\mathbf{Yv}||_2}. \tag{1}$$

In general, the non-convex problem in (1) can be solved through the singular value decomposition (SVD), or in the form of a standard or generalized eigenvalue problem (Uurtio et al., 2017).

The standard techniques work well when the underlying statistical patterns are linear and there are fewer variables than examples. However, it is not possible to model non-linear correlations well using standard CCA.

2.2. Kernel Canonical Correlation Analysis (KCCA)

Similarly to the original CCA problem (1), kernel CCA (KCCA) (Bach & Jordan, 2002; Hardoon et al., 2004) is solved through an eigenvalue problem. The n examples are transformed to Hilbert spaces $\phi_x : \mathbb{R}^p \mapsto \mathcal{H}_x$ and $\phi_y : \mathbb{R}^q \mapsto \mathcal{H}_y$ where ϕ denotes a feature map. The similarities between the observations in the Hilbert spaces are given by symmetric positive semi-definite kernels $\mathbf{K}^x_{i,j} = \mathbf{K}^x(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi_x(\mathbf{x}_i), \phi_x(\mathbf{x}_j) \rangle_{\mathcal{H}_x}$ and $\mathbf{K}^\mathbf{y}_{i,j} = \mathbf{K}^y(\mathbf{y}_i, \mathbf{y}_j) = \langle \phi_y(\mathbf{y}_i), \phi_y(\mathbf{y}_j) \rangle_{\mathcal{H}_y}$ where $i, j = 1, 2, \ldots, n$, $\mathbf{x} \in \mathbb{R}^p$ for view $\mathbf{X}$, and $\mathbf{y} \in \mathbb{R}^q$ for view $\mathbf{Y}$ respectively. The kernel canonical correlation is then measured through

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \rho_{\text{kcca}} = \quad \frac{\langle \mathbf{K}^x\boldsymbol{\alpha}, \mathbf{K}^y\boldsymbol{\beta} \rangle}{\|\mathbf{K}^x\boldsymbol{\alpha}\|_2 \|\mathbf{K}^y\boldsymbol{\beta}\|_2} \tag{2}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the coefficient vectors to be optimized by KCCA.

The computation of the kernel matrices in KCCA becomes costly at large sample sizes. Matrix decomposition approaches such as the incomplete Cholesky decomposition (Bach & Jordan, 2002; Hardoon et al., 2004) and the partial Gram-Schmidt orthogonalization (PGSO) (Hardoon et al., 2004) have been proposed to speed up the computations. When applying the PGSO in KCCA, every KCCA projection is computed as the span of a subset of the projections of a set of training examples that are selected by performing PGSO of the training vectors in the feature space.

2.3. Randomized Non-Linear CCA (RCCA)

The kernel computations of KCCA can also be approximated by random Fourier features (Rahimi & Recht, 2008), as proposed by (Lopez-Paz et al., 2014). In randomized non-linear CCA (RCCA) the kernels are approximated by random Fourier features. For example, as explained in (Lopez-Paz et al., 2014), the examples in view $\mathbf{X} \in \mathbb{R}^{n\times p} \mapsto \mathbf{z}(\mathbf{X}) \in \mathbb{R}^{n\times m}$ through the following map

$$\begin{aligned} & \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m \sim \mathcal{P}(\mathbf{w}) \\ \mathbf{z}_i \quad = \quad & [\cos(\mathbf{w}_i^\top \mathbf{x}_1 + b_i), \ldots, \cos(\mathbf{w}_i^\top \mathbf{x}_n + b_i)] \end{aligned}$$

where $\mathbf{z} \in \mathbb{R}^n$ for $i = 1, 2, \ldots, m$ and $\mathcal{P}(\mathbf{w})$ is a multivariate normal distribution, where $\mathbf{w} \in \mathbb{R}^d$. b_i is sampled from a uniform distribution $U[0, 2\pi]$. The m denotes the number of random Fourier features. The approximated kernel matrix $\mathbf{K} \approx \hat{\mathbf{K}}$ is given by

$$\hat{\mathbf{K}} = \frac{1}{m}\mathbf{z}(\mathbf{X})\mathbf{z}(\mathbf{X})^\top = \frac{1}{m}\sum_{i=1}^{m} \mathbf{z}_i\mathbf{z}_i^\top.$$

The kernel canonical correlations are found through the generalized eigenvalue problem, as in KCCA. In (Lopez-Paz et al., 2014), RCCA was shown to converge to KCCA solution when the number of random Fourier features increases.

2.4. Kernel Non-Linear Orthogonal Iterations (KNOI)

The approximation of RCCA to KCCA relies on the dimensionality of the random Fourier feature space. In some cases, a sufficiently good approximation may require a large number of features. This incurs high memory requirements to solve the CCA problem in the random Fourier feature space. A memory-efficient stochastic optimization algorithm of RCCA was proposed in (Wang & Livescu, 2016). The approach was named kernel non-linear orthogonal iterations (KNOI). KNOI employs ideas from the Augmented Approximate Gradient (AppGrad) (Ma et al., 2015) and Nonlinear Orthogonal Iterations (NOI) (Wang et al., 2015) techniques. The idea of KNOI is to update the CCA projections in small mini-batches, using a convex combination of the previous and current estimates. The estimates are then applied to update the cross-view least squares regression problems. In

(Wang & Livescu, 2016), KNOI was shown to outperform RCCA in terms of test correlation and running time, with GPU support.

2.5. Deep Canonical Correlation Analysis (DCCA)

Large-scale CCA problems can also be solved using deep neural networks. The deep learning methods apply the standard SVD technique to compute the maximal correlation between the outputs of two neural networks. In deep CCA (DCCA), (Andrew et al., 2013), the n observations, $\mathbf{x}_i \in \mathbb{R}^p$ and $\mathbf{y}_i \in \mathbb{R}^q$ for $i = 1, 2, \ldots, n$, are transformed several times through a neural network. At every layer, an observation, for example $\mathbf{x}$, is transformed by $\mathbf{h} = s(\mathbf{S}\mathbf{x} + \mathbf{b})$, where $\mathbf{S} \in \mathbb{R}^{\text{output} \times \text{input}}$ is a matrix of weights, $\mathbf{b} \in \mathbb{R}^{\text{output}}$ is a vector of bias, and $s : \mathbb{R} \mapsto \mathbb{R}$ is generally a non-linear function applied to every entry. The aim is to learn the optimal parameters $\mathbf{S}$ and $\mathbf{b}$ for both views such that the correlation between the transformed observations is maximized. Let $\mathbf{H}_x \in \mathbb{R}^{\text{output} \times n}$ and $\mathbf{H}_y \in \mathbb{R}^{\text{output} \times n}$ denote the matrices that have the final transformed output vectors in their columns. Let $\tilde{\mathbf{H}}_x = \mathbf{H}_x - \frac{1}{n}\mathbf{H}_x\mathbf{1}$ denote the centered data matrix and let $\hat{\mathbf{C}}_{xy} = \frac{1}{m-1}\tilde{\mathbf{H}}_x\tilde{\mathbf{H}}_y^T$, $\hat{\mathbf{C}}_{xx} = \frac{1}{m-1}\tilde{\mathbf{H}}_x\tilde{\mathbf{H}}_x^T + r_x\mathbf{I}$, and $\hat{\mathbf{C}}_{yy} = \frac{1}{m-1}\tilde{\mathbf{H}}_y\tilde{\mathbf{H}}_y^T + r_y\mathbf{I}$ where r_x and r_y are regularization constants, denote the cross-covariance and within-set covariance matrices. The correlation of the k^{th} components is the k^{th} singular value of the matrix

$$\mathbf{T} = \hat{\mathbf{C}}_{xx}^{-1/2}\hat{\mathbf{C}}_{xy}\hat{\mathbf{C}}_{yy}^{-1/2}.$$

2.6. CCA through Hilbert-Schmidt Independence Criterion (SCCA-HSIC)

Recently, as another direction of research, the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005) has been proposed as a dependence metric for the CCA projections (Chang et al., 2013; Uurtio et al., 2018). The sparse non-linear variant, named SCCA-HSIC, solves the following optimization problem (Uurtio et al., 2018):

$$\begin{aligned} \max_{\mathbf{u},\mathbf{v}} \quad & \frac{\text{trace}(\mathbf{K}^{\mathbf{u}}\hat{\mathbf{K}}^{\mathbf{v}})}{(n-1)^2} \\ s.t. \quad & \|\mathbf{u}\|_1 \leq s_x \\ & \|\mathbf{v}\|_1 \leq s_y \end{aligned} \tag{3}$$

where the values of the associated kernel matrices are given by $\mathbf{K}^{\mathbf{u}}_{ij} = \langle \mathbf{u}^T\mathbf{x}_i, \mathbf{u}^T\mathbf{x}_i \rangle$ and $\mathbf{K}^{\mathbf{v}}_{ij} = \langle \mathbf{v}^T\mathbf{y}_i, \mathbf{v}^T\mathbf{y}_i \rangle$. Due to the $O(n^2)$ size of the kernel matrices, SCCA-HSIC is approximated for large-scale experiments by the Nyström approximation of the kernel matrices (Williams & Seeger, 2001; Zhang et al., 2018).

3. gradKCCA: a method for sparse non-linear CCA

In this paper, we consider solving a KCCA problem with the additional constraint that the projection directions $\mathbf{w}_x \in \mathcal{H}_x$ and $\mathbf{w}_y \in \mathcal{H}_y$ need to have pre-images in the original data spaces, that is, $\phi_x^{-1}(\mathbf{w}_x) \in \mathbb{R}^p$ and $\phi_y^{-1}(\mathbf{w}_y) \in \mathbb{R}^q$. This requirement can alternatively be expressed as requiring the existence of a such $\mathbf{u} \in \mathbb{R}^p$ that $\phi_x(\mathbf{u}) = \mathbf{w}_x$ (resp. $\mathbf{v} \in \mathbb{R}^q$ s.t. $\phi(\mathbf{v}) = \mathbf{w}_y$). The resulting CCA problem is written as an optimization over the variables $\mathbf{u} \in \mathbb{R}^p, \mathbf{v} \in \mathbb{R}^q$:

$$\rho = \max_{\mathbf{u},\mathbf{v}} \frac{\sum_{i=1}^n \langle \phi_x(\mathbf{x}_i), \phi_x(\mathbf{u}) \rangle \cdot \langle \phi_y(\mathbf{y}_i), \phi_y(\mathbf{v}) \rangle}{\|(\langle \phi_x(\mathbf{x}_i), \phi_x(\mathbf{u}) \rangle)_{i=1}^n\| \|(\langle \phi_y(\mathbf{y}_i), \phi_y(\mathbf{v}) \rangle)_{i=1}^n\|} \tag{4}$$

Replacing the inner products with kernel functions $k^x(\mathbf{x}, \mathbf{z}) = \langle \phi_x(\mathbf{x}), \phi_x(\mathbf{z}) \rangle$, $k^y(\mathbf{y}, \mathbf{z}) = \langle \phi_y(\mathbf{y}), \phi_y(\mathbf{z}) \rangle$ and denoting the resulting score vectors as $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$ and $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$ we can write (4) as

$$\rho_{\text{gradKCCA}} = \max_{\mathbf{u},\mathbf{v}} \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\| \|\mathbf{k}^y(\mathbf{v})\|} \tag{5}$$

Note that the optimization problem (5) is the same as CCA (1) and KCCA (2) when $\mathbf{k}^x$ and $\mathbf{k}^y$ are the linear kernel functions. When $\mathbf{k}^x$ and $\mathbf{k}^y$ are non-linear kernels, the model constrains the projection directions to lie on the image of the feature map (see Section 4 for details). In terms of computation, (5) differs from KCCA by the time and space complexity of evaluating and maintaining the model, which is linear in the number of examples for gradKCCA, compared to the quadratic complexity of KCCA, while still allowing us to measure non-linear correlations in high-dimensional kernel-induced feature spaces.

We regularize the objective (5) by constraining the ℓ_P norm (with $P = 1, 2$) of $\mathbf{u}$ and $\mathbf{v}$. Hence, with regularization, the final optimization problem is as follows

$$\begin{aligned} \rho_{\text{gradKCCAreg}} \quad = \quad & \max_{\mathbf{u},\mathbf{v}} \rho(\mathbf{u}, \mathbf{v}) \coloneqq \frac{\mathbf{k}^x(\mathbf{u})^T \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\| \|\mathbf{k}^y(\mathbf{v})\|} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_u} \leq s_u \\ & \|\mathbf{v}\|_{P_v} \leq s_v, \end{aligned} \tag{6}$$

where s_u and s_v are user-defined positive scalars. The solution to (6) results in sparse projection vectors by setting $P_u = 1$ and/or $P_v = 1$. When the values of s_u and s_v are decreased the sparsity increases.

To maximize the correlation in (5), we use an alternating projected gradient approach (Algorithm 1). The algorithm computes the gradients $\nabla\rho_{\mathbf{u}} = \frac{\partial\rho(\mathbf{u},\mathbf{v})}{\partial\mathbf{u}}$ (resp. $\nabla\rho_{\mathbf{v}}$) of the correlation ρ_{gradKCCA} with respect to $\mathbf{u}$ and $\mathbf{v}$ in alternating fashion. Each gradient computation entails computing the

gradients of the kernel function $\nabla_{\mathbf{u}}\mathbf{k}^x(\mathbf{u})$ between the training data and the current $\mathbf{u}$, (See Supplementary material), which can be done in time $O(nd^2)$ time per epoch where $d = \max(p, q)$ in the case of polynomial and RBF kernels. In addition, for large-scale data, a mini-batch approach can be used to facilitate optimization.

Algorithm 1 gradKCCA

Input: $\mathbf{X}$, $\mathbf{Y}$, M (components), R (repetitions), δ (convergence limit), P_x and P_y (norms of $\mathbf{u}$ and $\mathbf{v}$) s_x and s_y (l_1 or l_2 norm constraints for $\mathbf{u}$ and $\mathbf{v}$), d_x and d_y (hyperparameters for $\mathbf{k}^x$ and $\mathbf{k}^y$)
Output: $\mathbf{U}$, $\mathbf{V}$
for all $m = \{1, 2, \ldots, M\}$ **do**
  for all $r = \{1, 2, \ldots, R\}$ **do**
    Initialize $\mathbf{u}_{mr}$ and $\mathbf{v}_{mr}$
    Compute $\mathbf{k}^x(\mathbf{u})$, $\mathbf{k}^y(\mathbf{v})$
    repeat
      Compute $\rho_{\text{old}} = \rho(\mathbf{u}, \mathbf{v})$
      Compute $\nabla\rho_{\mathbf{u}} = \frac{\partial\rho(\mathbf{u},\mathbf{v})}{\partial\mathbf{u}}$
      Update $\mathbf{u}_{mr} = \prod_{\|.\|_{P_x} \leq s_x}(\mathbf{u}_{mr} + \gamma\nabla_{\mathbf{u}})$ (The step size γ determined by line search)
      Re-compute $\mathbf{k}^x(\mathbf{u})$
      Compute $\nabla\rho_{\mathbf{v}} = \frac{\partial\rho(\mathbf{u},\mathbf{v})}{\partial\mathbf{v}}$
      Update $\mathbf{v}_{mr} = \prod_{\|.\|_{P_y} \leq s_y}(\mathbf{v}_{mr} + \gamma\nabla_{\mathbf{v}})$ (The step size γ determined by line search)
      Re-compute $\mathbf{k}^y(\mathbf{v})$
      Compute $\rho_{\text{current}} = \rho(\mathbf{u}, \mathbf{v})$
    until $|\rho_{\text{old}} - \rho_{\text{current}}|/|\rho_{\text{old}} + \rho_{\text{current}}| < \delta$
    $\rho_r = \rho_{\text{current}}$, $\mathbf{u}_r = \mathbf{u}_{mr}$, $\mathbf{v}_r = \mathbf{v}_{mr}$
  end for
  Select $r^* = \arg\max_r \rho_r$
  Store $\mathbf{U}(:, m) = \mathbf{u}_{r^*}$, $\mathbf{V}(:, m) = \mathbf{v}_{r^*}$
  Deflate $\mathbf{X}^{(m)}$, $\mathbf{Y}^{(m)}$ by $\mathbf{U}(:, m)$ and $\mathbf{V}(:, m)$
end for
Return: $\mathbf{U}$, $\mathbf{V}$

In Algorithm 1, the ℓ_1 or ℓ_2 norm constraints are implemented as follows. Let $\prod_{\|.\|_P \leq s}(\mathbf{u})$ denote the projection of $\mathbf{u}$ onto an ℓ_p norm ball. We solve

$$\begin{aligned}\prod_{\|.\|_P \leq s}(\mathbf{u}) \quad &= \quad \min_{\mathbf{u}_{\text{project}}} \|\mathbf{u}_{\text{project}} - \mathbf{u}\|_2 \\ &\text{s.t.} \qquad \|\mathbf{u}_{project}\|_P \leq s\end{aligned}$$

When $P = 2$, $\prod_{\|.\|_P \leq s}(\mathbf{u}) = \frac{s\mathbf{u}}{\|\mathbf{u}\|_2}$. When $P = 1$, we follow the method of ℓ_1 ball projection described in (Duchi et al., 2008).

3.1. Finding multiple canonical vectors

Given up to the m^{th} estimate of canonical projection $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$, the following orthogonal canonical projection vectors can be obtained by solving (6) for $\mathbf{u}_{m+1}$ and $\mathbf{v}_{m+1}$ with additional constraint that $\{\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_{m+1}\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{m+1}\}$ are mutually orthogonal. In Algorithm 1, deflation (Mackey, 2009) is applied to obtain multiple multivariate relations. As an example for $\mathbf{u}$, upon finding the m^{th} estimate for $\mathbf{u}_m$ from the current data matrix $\mathbf{X}^{(m)}$, the following orthogonal projection vector $\mathbf{u}_{(m+1)}$ can be obtained by solving $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} - \frac{\mathbf{u}_m\mathbf{u}_m^T\mathbf{X}^{(m)^T}}{\mathbf{u}_m^T\mathbf{u}_m}$. The same procedure is applied on $\mathbf{v}$. In this way, the resulting sets of vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots\}$ are mutually orthogonal. An alternative approach for finding orthogonal projection directions, is to look for uncorrelated projections (Hardoon et al., 2004). These can obtained by solving (6) for $\mathbf{u}_{m+1}$ and $\mathbf{v}_{m+1}$ with additional constraint that the score vectors $\{\mathbf{k}^x(\mathbf{u}_1), \ldots, \mathbf{k}^x(\mathbf{u}_m)\}$ and $\{\mathbf{k}^y(\mathbf{v}_1), \ldots, \mathbf{k}^y(\mathbf{v}_m)\}$ are mutually orthogonal.

Instead of using the successive formulation of Algorithm 1, we can join the successive sub-problems into one and solve (7) for matrices $\mathbf{U} \in \mathbb{R}^{p\times M}$ and $\mathbf{V} \in \mathbb{R}^{q\times M}$ which contains M vectors $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M)$ and $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M)$.

$$\begin{aligned}\max_{\mathbf{U},\mathbf{V}} \quad & \sum_{i=1}^{M} \frac{\mathbf{k}^x(\mathbf{u}_i)^\top\mathbf{k}^y(\mathbf{v}_i)}{||\mathbf{k}^x(\mathbf{u}_i)||||\mathbf{k}^y(\mathbf{v}_i)||} \\ s.t. \quad & \mathbf{k}^x(\mathbf{u}_i)^T\mathbf{k}^x(\mathbf{u}_j) = 0, \quad \forall i, j,\ i \neq j \\ & \mathbf{k}^y(\mathbf{v}_i)^T\mathbf{k}^y(\mathbf{v}_j) = 0, \quad \forall i, j,\ i \neq j \\ & \mathbf{k}^x(\mathbf{u}_i)^T\mathbf{k}^y(\mathbf{v}_j) = 0, \quad \forall i, j,\ i \neq j \qquad (7)\end{aligned}$$

The non-linear equality constraints in (7) are considerably harder to solve than (6) with the deflation approach. One can soften constraints by adding them to the objective function using the concept of penalty method and then solve the unconstrained problem by gradient descent. We leave following this direction for future work.

4. Relation between gradKCCA and KCCA

Let $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ be the optimal solution of KCCA (2). Then the optimal canonical vectors (Kuss & Graepel, 2003) are

$$\mathbf{w}_x^* = \sum_i \boldsymbol{\alpha}_i^*\phi_x(\mathbf{x}_i) \in \mathcal{L}(\phi_x, \mathbf{X}) \subseteq \mathcal{H}_x \text{ and}$$

$$\mathbf{w}_y^* = \sum_i \boldsymbol{\beta}_i^*\phi_y(\mathbf{y}_i) \in \mathcal{L}(\phi_y, \mathbf{Y}) \subseteq \mathcal{H}_y.$$

where $\mathcal{L}(\phi_x, \mathbf{X})$ and $\mathcal{L}(\phi_y, \mathbf{Y})$ are spaces spanned by feature maps corresponding to training samples only.

Let $\mathbf{u}^*$ and $\mathbf{v}^*$ be the optimum solution for (5) and hence $\phi_x(\mathbf{u}^*)$ and $\phi_y(\mathbf{v}^*)$ are the optimal canonical vectors. Please note that, $\phi_x(\mathbf{u}^*)$ (resp. $\phi_y(\mathbf{v}^*)$) does not necessarily lie within the span $\mathcal{L}(\phi_x, \mathbf{X})$ (resp. $\mathcal{L}(\phi_y, \mathbf{Y})$) of the training data in view $\mathbf{X}$ (resp. view $\mathbf{Y}$).

First we show that KCCA (2) correlation ρ_{kcca} upper-bounds the correlation found by gradKCCA (5), ρ_{gradKCCA}.

Lemma 4.1. *$\rho_{gradKCCA} \leq \rho_{kcca}$*

Proof. Considering the above notations, in general $\phi_x(\mathbf{u}^*)$ or $\phi_y(\mathbf{v}^*)$ may not lie in $\mathcal{L}(\phi_x, \mathbf{X})$ and $\mathcal{L}(\phi_y, \mathbf{Y})$. If $\phi_x(\mathbf{u}^*) \notin \mathcal{L}(\phi_x, \mathbf{X})$, $\phi_x(\mathbf{u}^*)$ can be split into the following two components

$$\phi_x(\mathbf{u}^*) = \phi_x(\mathbf{u}^*)^{\|} + \phi_x(\mathbf{u}^*)^{\perp}$$

where $\phi_x(\mathbf{u}^*)^{\|} \in \mathcal{L}(\phi_x, \mathbf{X}))$ and $\phi_x(\mathbf{u}^*)^{\perp}$ is orthogonal $\mathcal{L}(\phi_x, \mathbf{X})$, that is, $\langle \phi_x(\mathbf{u}^*)^{\perp}, \phi(x_i)\rangle = 0$ for any $i = 1, \ldots, n$. The score vector $\mathbf{k}^x(\mathbf{u}^*)$ satisfies

$$\begin{aligned}\mathbf{k}^x(\mathbf{u}^*) &= \Big[\langle \phi_x(\mathbf{x}_i), \phi_x(\mathbf{u}^*)\rangle\Big]_{i=1}^n = \\ &\Big[\langle \phi_x(\mathbf{x}_i), \phi_x(\mathbf{u}^*)^{\|}\rangle\Big]_{i=1}^n + \Big[\langle \phi_x(\mathbf{x}_i), \phi_x(\mathbf{u}^*)^{\perp}\rangle\Big]_{i=1}^n \\ &= \Big[\langle \phi_x(\mathbf{x}_i), \phi_x(\mathbf{u}^*)^{\|}\rangle\Big]_{i=1}^n + \mathbf{0}.\end{aligned}$$

In other words, the orthogonal part has no contribution to the score vector $\mathbf{k}^x(\mathbf{u}^*)$. An analogous fact holds true for the score vector $\mathbf{k}^y(\mathbf{v}^*)$. Consequently, the value of the canonical correlation ρ_{gradKCCA} is also not affected by the parts orthogonal to the span. Hence there exist some $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ such that $\phi_x(\mathbf{u}^*)^{\|} = \sum_i \boldsymbol{\alpha}_i \phi_x(\mathbf{x}_i)$ and $\phi_y(\mathbf{v}^*)^{\|} = \sum_i \boldsymbol{\beta}^{\top} \phi_y(\mathbf{y}_i)$ that achieve the same correlation as gradKCCA. Since KCCA (2) returns the optimal $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ the claim follows. □

Since for non-linear maps ϕ_x, $\mathcal{L}(\phi_x, \mathbf{X})$ does not in general exactly coincide with the image of the feature map ϕ_x, it may be that some vectors in $\mathbf{w}_x \in \mathcal{L}(\phi_x, \mathbf{X})$ and $\mathbf{w}_y \in \mathcal{L}(\phi_y, \mathbf{Y})$, the pre-images $\phi_x^{-1}(\mathbf{w}_x)$ and $\phi_x^{-1}(\mathbf{w}_y)$ do not not exist (Weston et al., 2004). In these cases, in general the (approximate) pre-image $\mathbf{x}$ for is chosen such that the squared distance of $\mathbf{w}_x$ and $\phi_x(\mathbf{u})$ is minimized.

$$\tilde{\mathbf{u}}(\mathbf{w}_x) = \arg\min_{\mathbf{u}\in\mathbb{R}^p} \|\mathbf{w}_x - \phi_x(\mathbf{u})\|_2 \quad (8)$$

$$\tilde{\mathbf{v}}(\mathbf{w}_y) = \arg\min_{\mathbf{v}\in\mathbb{R}^q} \|\mathbf{w}_y - \phi_y(\mathbf{v})\|_2 \quad (9)$$

The correlation achieved by computing the pre-image and mapping back to the feature space is given by

$$\rho_{\text{preimage}} = \frac{\mathbf{k}^x(\tilde{\mathbf{u}}(\mathbf{w}_x))^{\top}\mathbf{k}^y(\tilde{\mathbf{v}}(\mathbf{w}_y))}{||\mathbf{k}^x(\tilde{\mathbf{u}}(\mathbf{w}_x))||||\mathbf{k}^y(\tilde{\mathbf{v}}(\mathbf{w}_y))||} \quad (10)$$

This pre-image gives a lower-bound on ρ_{gradKCCA}:

Lemma 4.2. *$\rho_{gradKCCA} \geq \rho_{preimage}$*

Proof. Follows from the fact that (5) gives the the maximum correlation for all $\mathbf{u} \in \mathbb{R}^p$ and $\mathbf{v} \in \mathbb{R}^q$, including $\mathbf{u} = \tilde{\mathbf{u}}(\mathbf{w}_x)$ and $\mathbf{v} = \tilde{\mathbf{v}}(\mathbf{w}_y)$ □

This pre-image can be efficiently computed from an optimal KCCA solution. Namely, substituting the KCCA optimum $\mathbf{w}_x^* = \sum_i \alpha_i^* \phi_x(x_i)$ into (8) we obtain

$$\begin{aligned}\tilde{\mathbf{u}}(\mathbf{w}_x^*) &= \arg\min_{\mathbf{u}} ||\sum_i \alpha_i^* \phi(\mathbf{x}_i) - \phi(\mathbf{u})||^2 = \\ &= \arg\min_{\mathbf{u}} (||\sum_i \alpha_i^* \phi(\mathbf{x}_i)||^2 - 2\langle \sum_i \alpha_i^* \phi(\mathbf{x}_i), \phi(\mathbf{u})\rangle \\ &\qquad + ||\phi(\mathbf{u})||^2) \\ &= \arg\min_{\mathbf{u}} \boldsymbol{\alpha}^{*T}\mathbf{K}^x\boldsymbol{\alpha}^* - 2\boldsymbol{\alpha}^{*T}\mathbf{k}^x(\mathbf{u}) + \mathbf{k}^x(\mathbf{u}, \mathbf{u})\end{aligned}$$

which can be solved by e.g. by gradient descent on $\mathbf{u}$. We use this approach in our experiment section to compute pre-images for KCCA.

Using the above pre-image problem, we can further bound the distance of gradKCCA from the KCCA optimum. Let us define the bounds on pre-image errors as

$$\begin{aligned}B_{\mathbf{X}} &= \max_{\mathbf{w}_x \in \mathcal{L}(\phi_x, \mathbf{X})} \|\mathbf{w}_x - \phi_x(\tilde{\mathbf{u}}(\mathbf{w}_x))\|_2 \\ B_{\mathbf{Y}} &= \max_{\mathbf{w}_y \in \mathcal{L}(\phi_y, \mathbf{Y})} \|\mathbf{w}_y - \phi_x(\tilde{\mathbf{v}}(\mathbf{w}_y))\|_2. \quad (11)\end{aligned}$$

Theorem 4.3. *Let us further assume that norm in $\mathcal{H}_x$ and $\mathcal{H}_y$ are upper bounded by M_x and M_y, that is,*

$$\begin{aligned}&\forall\, \phi_x(\mathbf{u}) \in \mathcal{H}_x,\ \|\phi_x(\mathbf{u})\| \leq M_x \\ \textit{and}\quad &\forall\, \phi_y(\mathbf{v}) \in \mathcal{H}_y,\ \|\phi_y(\mathbf{v})\| \leq M_y. \quad (12)\end{aligned}$$

Then,

$$\rho_{gradKCCA} \geq \rho_{preimage} \geq \rho_{kcca} - \left(\frac{B_y}{M_y} + \frac{B_x}{M_x}\right)$$

Proof. See Section 2 in the Supplementary Material. □

5. Experiments

5.1. Predictive Performance

We assess the performance of gradKCCA, and compare it with KCCA, DCCA, RCCA, KNOI and SCCA-HSIC, in terms of robustness to noise and scalability. In the experiments, we test two different sets of input relations, monotonic algebraic relations and non-monotonic trigonometric relations. For the monotonic algebraic relations, we use the linear kernel for gradKCCA and KCCA. For the non-monotonic trigonometric relations we use the quadratic kernel for both.

For the other kernel-based methods, that is RCCA, KNOI, and SCCA-HSIC, we only apply the Gaussian kernel, which is a requirement for SCCA-HSIC due to its universality (Gretton et al., 2008), and which has been applied previously together with random Fourier features for RCCA and KNOI (Lopez-Paz et al., 2014; Wang & Livescu, 2016).

Experiments on Monotonic Relations. In the experiments, where the linear kernels are tested for gradKCCA and KCCA, we apply the following monotonic algebraic relations. We generate three different polynomial relations of the form $f(x) = x^d$, where $x \in \mathbb{R}_+$ and $d = 1, 2, 3$, that is linear, quadratic, and cubic relations, respectively. We also generate two transcendental relations, the exponential and the logarithmic relations, of the form $f(x) = \exp(x)$ and $f(x) = \log(x)$ respectively, where $x \in \mathbb{R}_+$. We consider a two-to-two relation, that is, the two first columns of $\mathbf{Y}$ and $\mathbf{X}$ satisfy $\mathbf{y}^1 + \mathbf{y}^2 = f(\mathbf{x}^1 + \mathbf{x}^2) + \boldsymbol{\xi}$ where $\boldsymbol{\xi} \sim \mathcal{N}(0, 0.05)$ denotes a vector of normal distributed noise. The variables in the columns of $\mathbf{X}$ and $\mathbf{Y}$ are generated from a random uni-variate uniform distribution, $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p \sim U[0, 1]$ and $\mathbf{y}^3, \mathbf{y}^4, \ldots, \mathbf{y}^q \sim U[0, 1]$.

Experiments on Non-Monotonic Relations. As non-monotonic relations, we generate the following trigonometric relations that cannot be approximated using a linear kernel. Let every entry of $\boldsymbol{\theta} \in \mathbb{R}^n \sim U[-2\pi, 2\pi]$. The variables, in the columns of $\mathbf{X}$ and $\mathbf{Y}$, are generated by

$$\begin{aligned} \mathbf{X} &= \left(a_1 \cdot \sin \tfrac{\mathbf{x}_2}{a_2} + \boldsymbol{\xi} \quad \cdots \quad \boldsymbol{\theta}_p\right) \\ \mathbf{Y} &= \left(b_1 \cdot \cos \tfrac{\mathbf{x}_2}{b_2} + \boldsymbol{\xi} \quad c_1 \cdot \cos \tfrac{\mathbf{x}_2}{c_2} + \boldsymbol{\xi} \quad \cdots \quad \boldsymbol{\theta}_q\right) \end{aligned}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(0, 0.05)$ denotes a vector of normal distributed noise. In the $\mathbf{X}$ view, the first and second variables are related, and the first and second variables in the $\mathbf{Y}$ view are related with the second variable of $\mathbf{X}$ view. We generate three different variants of this setting. In the first relation, we put $a_1 = 0.2$, $b_1 = 0.5$, and $c_1 = 0.4$ and $a_2 = 1$, $b_2 = 1$, and $c_2 = 0.4$. In the second relation, we put $a_1 = 3$, $b_1 = 3$, and $c_1 = 6$ and $a_2 = 1$, $b_2 = 1$, and $c_2 = 0.4$. In the third relation, we put $a_1 = 2$, $b_1 = 2$, and $c_1 = 4$ and $a_2 = 2$, $b_2 = 2$, and $c_2 = 4$.

Hyperparameter Settings. In all of the experiments, we randomly select 500 examples from the simulated training dataset for hyperparameter tuning by repeated nested cross-validation. For gradKCCA and SCCA-HSIC, we tune the ℓ_1 sparsity constraints. For KCCA, we tune the regularization hyperparameters. For DCCA, we tune the neural network structure, that is the number of layers and the number of hidden units. We apply the sigmoid activation function for the transformations. For RCCA and KNOI, we tune the number of random Fourier features. For SCCA-HSIC, RCCA, and KNOI, we set the standard deviations of the Gaussian kernels by the median heuristic.

Evaluation Metrics. We assess the performances of the methods in terms of test correlation, F1 score, and Area Under the Curve (AUC). The test correlations are computed using (5), where the learned $\mathbf{u}$ and $\mathbf{v}$ are applied on test examples.

For gradKCCA, SCCA-HSIC and KCCA, we assess the correctness of $\mathbf{u}$ and $\mathbf{v}$ using the F1 score and AUC. For KCCA, we use the output of the preimage algorithm as the predicted $\mathbf{u}$ and $\mathbf{v}$. F1 score is defined as $\text{F1} = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$, where TP, FP, and FN denote the true positives, false positives, and false negatives respectively. In our case, the labels refer to the zero/non-zero coefficient values in the estimated $\mathbf{u}$ and $\mathbf{v}$ and the respective ground truth vectors. For the computations of the two metrics, the estimated $\mathbf{u}$ and $\mathbf{v}$ are binarized such that all entries less than or equal to 0.05 are put to zero and all the rest to 1.

5.1.1. Varying the Ground Truth Sparsity

We evaluate the robustness of gradKCCA, KCCA, DCCA, KNOI and SCCA-HSIC when the sparsity of the ground truth, that is the number of noise variables, increases. We do not include RCCA in these small-scale experiments since it results in the same kernel canonical correlation as KCCA at a sufficient number of random Fourier features.

For every multivariate function, given in Section 5.1, we repeat the following scheme 10 times. We generate two data matrices $\mathbf{X}$ and $\mathbf{Y}$ of sizes $n \times p$ and $n \times q$, where $n = 1000$. The dimensions tested are $p = q = \{5, 50, 100, 250\}$ and $\{5, 10, 50, 100\}$, for the monotonic and non-monotonic relations respectively.

Monotonic Relations. The results are shown in Figure 1. Since a linear kernel function is applied, gradKCCA and KCCA perform similarly, in terms of training and test correlation. The F1 score and AUC show that gradKCCA is slightly more robust to noise than KCCA. This could be due to the constraint that the projection direction lies on the image of the feature map. DCCA, KNOI, and SCCA-HSIC do not tolerate noise as well as gradKCCA and KCCA. For KNOI, this could be explained by the stochastic approximation algorithm that uses mini-batches of the training examples.

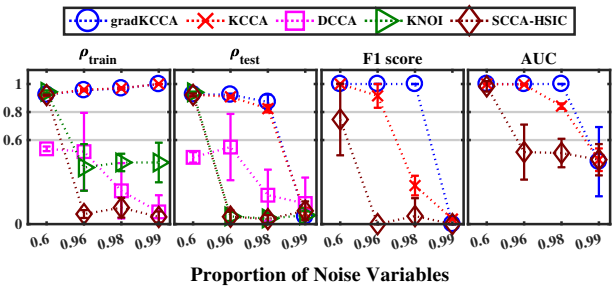

Figure 1. **Experiment 5.1.1 on Monotonic Relations.**

Non-Monotonic Relations. The results are shown in Figure 2. In terms of test correlations, the quadratic kernel gradKCCA and SCCA-HSIC perform equally well. However, gradKCCA is more accurate than SCCA-HSIC in identifying the related variables in terms of F1 score. As has been shown in Section 4, in the case of higher degree kernels, the canonical correlation using the computed the pre-images of KCCA is lower than the kernel canonical correlation. DCCA and KNOI are shown not to be tolerant to irrelevant noise variables in this context.

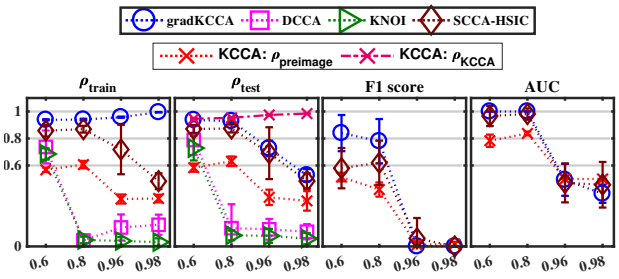

Figure 2. **Experiment 5.1.1 on Non-Monotonic Relations.**

5.1.2. Varying the Sample Size

In the scalability experiments, we compare gradKCCA with the large-scale CCA variants, that is DCCA, RCCA, KNOI, and Nyström approximated SCCA-HSIC. For every multivariate function, given in Section 5.1, we repeat the following scheme five times. We generate two data matrices $\mathbf{X}$ and $\mathbf{Y}$ of sizes $n \times p$ and $n \times q$, where $n = \{10^3, 10^4, 10^5, 10^6\}$. The number of inducing variables for the Nyström approximated SCCA-HSIC is set to $\{0.7, 0.1, 0.05\}$ for the first three sample sizes respectively. The dimensions are $p = q = 20$.

Monotonic Relations. The results are shown in Figure 3. In terms of training and test (kernel) canonical correlations, RCCA performs the best. In terms of training and test canonical correlations, gradKCCA and SCCA-HSIC perform equally well. However, in terms of running time gradKCCA outperforms all the methods. SCCA-HSIC is the slowest method, and is not run for one million examples.

Non-Monotonic Relations. The results are shown in Figure 4. As in the experiment on monotonic relations, gradKCCA is the fastest method. DCCA extracts non-monotonic trigonometric relations better than the monotonic algebraic ones, equally well as RCCA. In contrast to the previous experiment, gradKCCA is better in extracting the relations than SCCA-HSIC, which is seen both in terms of test correlation and F1 score. In the small sample size regime ($n = 1000$) DCCA, KNOI and SCCA-HSIC exhibit considerably more overfitting than RCCA and gradKCCA.

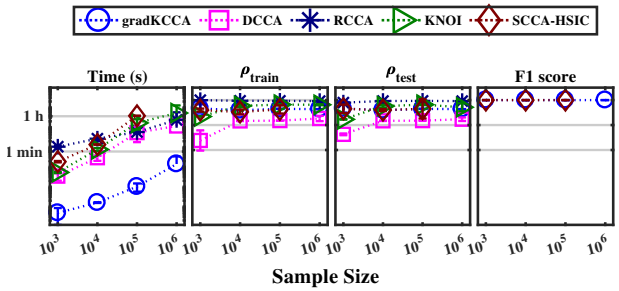

Figure 3. **Experiment 5.1.2 on Monotonic Relations.** The performances of gradKCCA, DCCA, RCCA, KNOI, and SCCA-HSIC are shown when the sample sizes $n = \{10^3, 10^4, 10^5, 10^6\}$ are tested. Due to the slow computation time, SCCA-HSIC is only run until 10^5.

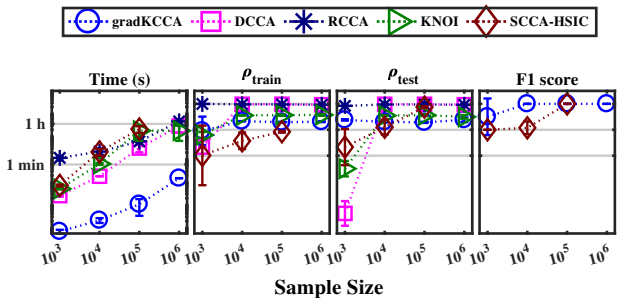

Figure 4. **Experiment 5.1.2 on Non-Monotonic Relations.** The performances of gradKCCA, DCCA, RCCA, KNOI, and SCCA-HSIC are shown when the sample sizes $n = \{10^3, 10^4, 10^5, 10^6\}$ are tested. Due to the slow computation time, SCCA-HSIC is only run until 10^5.

5.2. Real-World Datasets

We evaluate the performances of gradKCCA, DCCA, RCCA, KNOI, and SCCA-HSIC in terms of test correlation of the first CCA component and computation time on the publicly available MNIST handwritten digits (LeCun, 1998) and MediaMill (Snoek et al., 2006) datasets. We extract the first component 5 times to obtain an average. We tune the hyperparameters in similar manner as in Section 5.1.

MNIST Handwritten Digits. The MNIST handwritten digits dataset contains 60 000 training and 10 000 testing images of handwritten digits. As in (Andrew et al., 2013), we analyze the relations of the left and right halves of images of the handwritten digits. Every image consists of 28×28 matrix of pixels. Every pixel represents a grey scale value

Table 1. **MNIST Dataset.** Comparison of the test correlations of gradKCCA, DCCA, RCCA, KNOI and SCCA-HSIC.

	ρ_{TEST}	TIME (S)
GRADKCCA	0.989 ± 0.001	12.54 ± 1.17
DCCA	0.987 ± 0.000	$1.21 \cdot 10^3 \pm 48.46$
RCCA	0.906 ± 0.025	$4.21 \cdot 10^2 \pm 12.45$
KNOI	0.901 ± 0.013	$1.93 \cdot 10^3 \pm 38.12$
SCCA-HSIC	0.976 ± 0.041	$5.14 \cdot 10^4 \pm 55.32$

Table 2. **MediaMill Dataset.** Comparison of the test correlations of gradKCCA, DCCA, RCCA, KNOI and SCCA-HSIC.

	ρ_{TEST}	TIME (S)
GRADKCCA	0.825 ± 0.021	25.67 ± 0.27
DCCA	0.823 ± 0.035	$3.08 \cdot 10^2 \pm 79.12$
RCCA	0.771 ± 0.042	$2.35 \cdot 10^2 \pm 16.02$
KNOI	0.732 ± 0.026	$4.78 \cdot 10^2 \pm 64.25$
SCCA-HSIC	0.781 ± 0.037	$2.31 \cdot 10^4 \pm 86.44$

ranging from one to 256. The 14 left and right columns are separated to form the two views, making 392 features in each view.

The test correlations of the views, and computation times, are given in Table 1. gradKCCA outperforms in terms of speed, and finds a test correlation of highest value. DCCA is the second best to compute the test correlation and SCCA-HSIC the third, but with the slowest computation time.

MediaMill Dataset. The MediaMill dataset consists of 43907 examples that correspond to keyframes taken from video shots. Every keyframe is represented by 120 visual features and 101 text annotations, or labels, respectively.

The test correlations between the visual and annotation views, together with the computation times, are given in Table 2. As in the MNIST experiment, gradKCCA is the fastest. DCCA extracts almost as high test correlation value as gradKCCA but is much slower.

6. Discussion

Our experiments showed that the gradKCCA model is orders of magnitude faster to optimize than competing nonlinear CCA models. On real-world datasets, gradKCCA obtained top predictive performance, and again proved to be the fastest method by a large margin. In our experiments with artifical data, we also showed that gradKCCA is very robust to a large amount respect to irrelevant, noise variables in the data. Also, the method obtains good performance already with modest amount of data, compared to several of the competing methods.

The gradKCCA model can be seen as a variant of KCCA that gives access to the pre-images of the canonical projection directions. Our results in Section 4 show that the correlations on the training set are upper bounded by KCCA and lower-bounded by correlations that are obtained after taking the pre-image of the KCCA solution. In our experiments, these relations were generally seen to translate to the test set as well. Thus, gradKCCA can be seen to give a good compromise in situations where the model's interpretation in terms of relevant variables is of interest.

The general approach used here that removes the need to maintain a kernel matrix by moving the optimization to the pre-image space is not limited to CCA models. We envision that many other kernel methods could be scaled up to big data using the same principle. On the other hand, it is clear that the new model puts more burden for finding a good kernel so that the requirement of the kernel space model having an exact pre-image does not become too restrictive.

7. Conclusions

In this paper, we have put forward a new method for finding canonical correlations in paired datasets. The proposed method, gradKCCA, adapts KCCA by requiring the projection directions in the feature space to have pre-images in the original space. This modeling choice has the by-product that a kernel matrix does not need to be computed or maintained. This allows fast optimization through gradient approaches, which scales up to big datasets effortlessly. Moreover due the approach, the sparsity projection directions can be controlled which gives the method robustness against irrelevant variables.

References

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. Deep canonical correlation analysis. In *ICML*, pp. 1247–1255, 2013.

Bach, F. and Jordan, M. Kernel independent component analysis. *JMLR*, 3(Jul):1–48, 2002.

Chang, B., Kruger, U., Kustra, R., and Zhang, J. Canonical correlation analysis based on hilbert-schmidt independence criterion and centered kernel target alignment. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 316–324, 2013.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279. ACM, 2008.

Gretton, A., Bousquet, O., Smola, A., and Scholkopf, B.

Measuring statistical dependence with hilbert-schmidt norms. In *ALT*, volume 16, pp. 63–78. Springer, 2005.

Gretton, A., Fukumizu, K., Teo, C., Song, L., Schölkopf, B., and Smola, A. A kernel statistical test of independence. In *Advances in neural information processing systems*, pp. 585–592, 2008.

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

Hotelling, H. The most predictable criterion. *Journal of educational Psychology*, 26(2):139, 1935.

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

Kuss, M. and Graepel, T. The geometry of kernel canonical correlation analysis. *Technical Report 108, Max-Planck Institute for Biol. Cybernetics*, 2003.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Lopez-Paz, D., Sra, S., Smola, A., Ghahramani, Z., and Schölkopf, B. Randomized nonlinear component analysis. In *International Conference on Machine Learning*, pp. 1359–1367, 2014.

Ma, Z., Lu, Y., and Foster, D. Finding linear structure in large datasets with scalable canonical correlation analysis. In *International Conference on Machine Learning*, pp. 169–178, 2015.

Mackey, L. W. Deflation methods for sparse pca. In *Advances in neural information processing systems*, pp. 1017–1024, 2009.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Snoek, C. G., Worring, M., Van Gemert, J. C., Geusebroek, J.-M., and Smeulders, A. W. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM international conference on Multimedia*, pp. 421–430. ACM, 2006.

Uurtio, V., Monteiro, J., Kandola, J., Shawe-Taylor, J., Fernandez-Reyes, D., and Rousu, J. A tutorial on canonical correlation methods. *ACM Comput. Surv.*, 50(6): 95:1–95:33, November 2017. ISSN 0360-0300.

Uurtio, V., Bhadra, S., and Rousu, J. Sparse non-linear cca through hilbert-schmidt independence criterion. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1278–1283, Nov 2018. doi: 10.1109/ICDM.2018.00172.

Wang, W. and Livescu, K. Large-scale approximate kernel canonical correlation analysis. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

Wang, W., Arora, R., Livescu, K., and Srebro, N. Stochastic optimization for deep cca via nonlinear orthogonal iterations. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pp. 688–695. IEEE, 2015.

Weston, J., Schölkopf, B., and Bakir, G. H. Learning to find pre-images. In *Advances in neural information processing systems*, pp. 449–456, 2004.

Williams, C. and Seeger, M. Using the nyström method to speed up kernel machines. In *Adv Neural Inf Process Syst*, number EPFL-CONF-161322, pp. 682–688, 2001.

Zhang, Q., Filippi, S., Gretton, A., and Sejdinovic, D. Large-scale kernel methods for independence testing. *Stat Comput*, 28(1):113–130, 2018.