

# Large-Scale Sparse Kernel Canonical Correlation Analysis

Viivi Uurtio<sup>1</sup>, Sahely Bhadra<sup>2</sup>, and Juho Rousu<sup>1</sup>

<sup>1</sup> Department of Computer Science, Aalto University  
Helsinki Institute for Information Technology HIIT

<sup>2</sup> Indian Institute of Technology (IIT), Palakkad

June 11, 2019

A?



IIT PALAKKAD



From large two-view datasets, it is not straightforward to identify which of the variables are related

From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

→ In the non-linear and/or large-scale variants, we cannot access the  $\mathbf{u}$  and  $\mathbf{v}$

From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

→ In the non-linear and/or large-scale variants, we cannot access the  $\mathbf{u}$  and  $\mathbf{v}$

Scalability  $\mathbf{u}$  and  $\mathbf{v}$

From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

→ In the non-linear and/or large-scale variants, we cannot access the  $\mathbf{u}$  and  $\mathbf{v}$

	Scalability	$\mathbf{u}$ and $\mathbf{v}$
Kernel CCA		

From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

→ In the non-linear and/or large-scale variants, we cannot access the  $\mathbf{u}$  and  $\mathbf{v}$

Kernel CCA  
RF KCCA

Scalability



$\mathbf{u}$  and  $\mathbf{v}$





From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

→ In the non-linear and/or large-scale variants, we cannot access the  $\mathbf{u}$  and  $\mathbf{v}$

Kernel CCA  
RF KCCA  
KNOI

Scalability



$\mathbf{u}$  and  $\mathbf{v}$











From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

→ In the non-linear and/or large-scale variants, we cannot access the  $\mathbf{u}$  and  $\mathbf{v}$

	Scalability	$\mathbf{u}$ and $\mathbf{v}$
Kernel CCA		
RF KCCA		
KNOI		
Deep CCA		

From large two-view datasets, it is not straightforward to identify which of the variables are related

$$\frac{\langle \mathbf{X}\mathbf{u}, \mathbf{Y}\mathbf{v} \rangle}{\|\mathbf{X}\mathbf{u}\|_2 \|\mathbf{Y}\mathbf{v}\|_2}$$

→ In standard CCA, we identify the related variables from  $\mathbf{u}$  and  $\mathbf{v}$

→ In the non-linear and/or large-scale variants, we cannot access the  $\mathbf{u}$  and  $\mathbf{v}$

	Scalability	$\mathbf{u}$ and $\mathbf{v}$
Kernel CCA		
RF KCCA		
KNOI		
Deep CCA		
SCCA-HSIC		

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

$$\text{Let } \mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n \text{ and } \mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$$

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

Let  $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$  and  $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \rho_{\text{gradKCCA}}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\|_2 \|\mathbf{k}^y(\mathbf{v})\|_2} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_x} \leq s_u \text{ and } \|\mathbf{v}\|_{P_y} \leq s_v \end{aligned}$$

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

Let  $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$  and  $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \rho_{\text{gradKCCA}}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\|_2 \|\mathbf{k}^y(\mathbf{v})\|_2} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_x} \leq s_u \text{ and } \|\mathbf{v}\|_{P_y} \leq s_v \end{aligned}$$

Maximum through alternating projected gradient ascent

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

Let  $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$  and  $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \rho_{\text{gradKCCA}}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\|_2 \|\mathbf{k}^y(\mathbf{v})\|_2} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_x} \leq s_u \text{ and } \|\mathbf{v}\|_{P_y} \leq s_v \end{aligned}$$

Maximum through alternating projected gradient ascent

Optimization steps for  $\mathbf{u}$ :



gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

Let  $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$  and  $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \rho_{\text{gradKCCA}}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\|_2 \|\mathbf{k}^y(\mathbf{v})\|_2} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_x} \leq s_u \text{ and } \|\mathbf{v}\|_{P_y} \leq s_v \end{aligned}$$

Maximum through alternating projected gradient ascent

Optimization steps for  $\mathbf{u}$ :

→ Compute the gradient  $\nabla_{\rho_{\mathbf{u}}} = \frac{\partial \rho(\mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}$

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

Let  $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$  and  $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \rho_{\text{gradKCCA}}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\|_2 \|\mathbf{k}^y(\mathbf{v})\|_2} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_x} \leq s_u \text{ and } \|\mathbf{v}\|_{P_y} \leq s_v \end{aligned}$$

Maximum through alternating projected gradient ascent

Optimization steps for  $\mathbf{u}$ :

→ Compute the gradient  $\nabla_{\rho_{\mathbf{u}}} = \frac{\partial \rho(\mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}$

→ Step-size using line search:  $\max_{\gamma} \rho(\mathbf{u} + \gamma \nabla_{\rho_{\mathbf{u}}})$

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

Let  $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$  and  $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \rho_{\text{gradKCCA}}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\|_2 \|\mathbf{k}^y(\mathbf{v})\|_2} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_x} \leq s_u \text{ and } \|\mathbf{v}\|_{P_y} \leq s_v \end{aligned}$$

Maximum through alternating projected gradient ascent

Optimization steps for  $\mathbf{u}$ :

- Compute the gradient  $\nabla_{\rho_{\mathbf{u}}} = \frac{\partial \rho(\mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}$
- Step-size using line search:  $\max_{\gamma} \rho(\mathbf{u} + \gamma \nabla_{\rho_{\mathbf{u}}})$
- Gradient step towards maximum:  $\mathbf{u}_{\text{grad}} = \mathbf{u} + \gamma^* \nabla_{\rho_{\mathbf{u}}}$

gradKCCA is a kernel matrix free method that efficiently optimizes  $\mathbf{u}$  and  $\mathbf{v}$

Let  $\mathbf{k}^x(\mathbf{u}) = (k^x(\mathbf{x}_i, \mathbf{u}))_{i=1}^n$  and  $\mathbf{k}^y(\mathbf{v}) = (k^y(\mathbf{y}_i, \mathbf{v}))_{i=1}^n$

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \rho_{\text{gradKCCA}}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{k}^x(\mathbf{u})^\top \mathbf{k}^y(\mathbf{v})}{\|\mathbf{k}^x(\mathbf{u})\|_2 \|\mathbf{k}^y(\mathbf{v})\|_2} \\ \text{s.t.} \quad & \|\mathbf{u}\|_{P_x} \leq s_u \text{ and } \|\mathbf{v}\|_{P_y} \leq s_v \end{aligned}$$

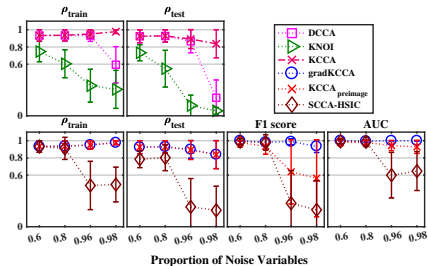
Maximum through alternating projected gradient ascent

Optimization steps for  $\mathbf{u}$ :

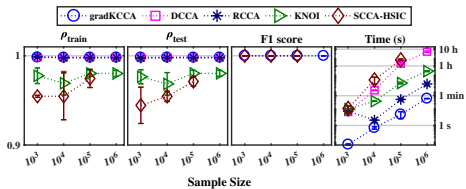
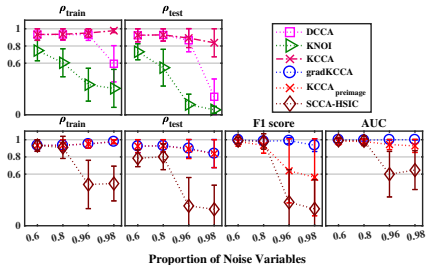
- Compute the gradient  $\nabla_{\rho_{\mathbf{u}}} = \frac{\partial \rho(\mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}$
- Step-size using line search:  $\max_{\gamma} \rho(\mathbf{u} + \gamma \nabla_{\rho_{\mathbf{u}}})$
- Gradient step towards maximum:  $\mathbf{u}_{\text{grad}} = \mathbf{u} + \gamma^* \nabla_{\rho_{\mathbf{u}}}$
- Project onto  $\ell_P$  ball:  $\mathbf{u} = \prod_{\|\cdot\|_{P_x} \leq s_x} \mathbf{u}_{\text{grad}}$

Experiments demonstrate noise tolerance, scalability, and superior speed of gradKCCA

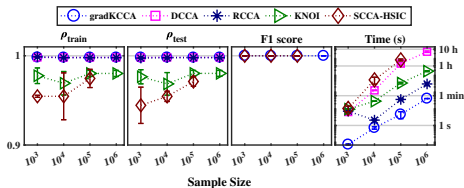
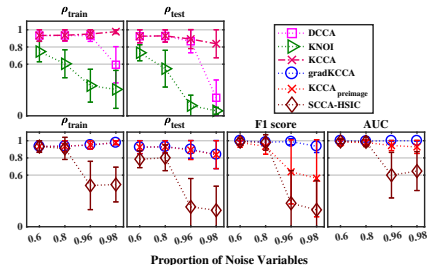
# Experiments demonstrate noise tolerance, scalability, and superior speed of gradKCCA



# Experiments demonstrate noise tolerance, scalability, and superior speed of gradKCCA



# Experiments demonstrate noise tolerance, scalability, and superior speed of gradKCCA



MediaMill

	$\rho_{TRAIN}$	$\rho_{TEST}$	TIME (s)
GRADKCCA	0.666 $\pm$ 0.004	0.657 $\pm$ 0.007	8 $\pm$ 4
DEEP CCA	0.643 $\pm$ 0.005	0.633 $\pm$ 0.003	1280 $\pm$ 112
RF KCCA	0.633 $\pm$ 0.001	0.626 $\pm$ 0.005	23 $\pm$ 9
KNOI	0.652 $\pm$ 0.001	0.645 $\pm$ 0.003	218 $\pm$ 73
SCCA-HSIC	0.627 $\pm$ 0.004	0.625 $\pm$ 0.002	1804 $\pm$ 143



# Thanks and meet me at the poster!

## Considerations can be sent to

✉ viivi.uurtio@aalto.fi

## MATLAB codes available on

<https://github.com/aalto-ics-kepaco/gradKCCA>

**Large-Scale Sparse Kernel Canonical Correlation Analysis**

Viivi Uurtio<sup>1</sup>, Sahel Shahmoradian<sup>2</sup>, and John Rønne<sup>3</sup>  
<sup>1</sup> Helsinki Institute of Information Technology (HIIT),  
<sup>2</sup> Department of Computer Science, Aalto University,  
<sup>3</sup> Institute of Technology (TKK), Finland  
v.uurtio@aalto.fi

**gradKCCA: Sparse kernel-based non-linear CCA**

- maximizes canonical correlation in the kernel-induced feature space through the gradient of the percentage of non-zero entries
- does not rely on a kernel matrix
- operates on the gradient of the kernel to achieve the same result using the  $L_1$  norm as the percentage of the projection directions

$\arg \max_{\substack{U, V \in \mathbb{R}^{n \times p} \\ \|U\|_F = \|V\|_F = 1}} \text{corr}(U, V)$

**Algorithms: Alternating projected gradient ascent**

```
1: Input:  $A \in \mathbb{R}^{n \times p}$ ,  $B \in \mathbb{R}^{n \times q}$ ,  $\lambda > 0$ ,  $\mu > 0$ ,  $\epsilon > 0$ ,  $\tau > 0$ ,  $\eta > 0$   
2: Initialize:  $U \in \mathbb{R}^{n \times p}$ ,  $V \in \mathbb{R}^{n \times q}$ ,  $\alpha = 1$   
3: while  $\|U - U_{\text{old}}\|_F + \|V - V_{\text{old}}\|_F > \epsilon$  or  $\text{corr}(U, V) < \tau$   
4:    $U_{\text{old}} = U$ ,  $V_{\text{old}} = V$   
5:    $U = \text{argmax}_{\|U\|_F = 1} \text{corr}(U, V)$   
6:    $V = \text{argmax}_{\|V\|_F = 1} \text{corr}(U, V)$   
7:    $U = U + \alpha \lambda (U - U_{\text{old}})$   
8:    $V = V + \alpha \lambda (V - V_{\text{old}})$   
9:    $\alpha = \text{argmin}_{\alpha > 0} \text{corr}(U + \alpha \lambda (U - U_{\text{old}}), V + \alpha \lambda (V - V_{\text{old}}))$   
10: end while
```

**Simulated non-linear relations**

**First components from real-world datasets**

Dataset	gradKCCA	Kernel CCA	Kernel PCA	Kernel LDA
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98
mnist	0.98	0.98	0.98	0.98

**Simulated non-quadratic relations**

**Concluding remarks**

- links the related variables in the data space necessarily when using a linear or a non-linear kernel
- unlike KCCA, gradKCCA does not rely on a kernel matrix, which results in superior speed
- due to more consistency in the percentage given subspaces against treatment variations

This work has been in part supported by Academy of Finland grants: 312043 (KAC) and 311248 (GradKCCA)

Open source MATLAB code: <https://github.com/aalto-ics-kepaco/gradKCCA>