The University of Texas at Austin

A simple and provable algorithm for sparse diagonal CCA

Megasthenis Asteris
Anastasios Kyrillidis

Dept. of electrical and computer engineering
University of Texas, Austin

Oluwasanmi Koyejo
Russell Poldrack

Dept. of Psychology
Stanford University

What is CCA?

(Hotelling, 1936)

- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.

What is CCA?

(Hotelling, 1936)

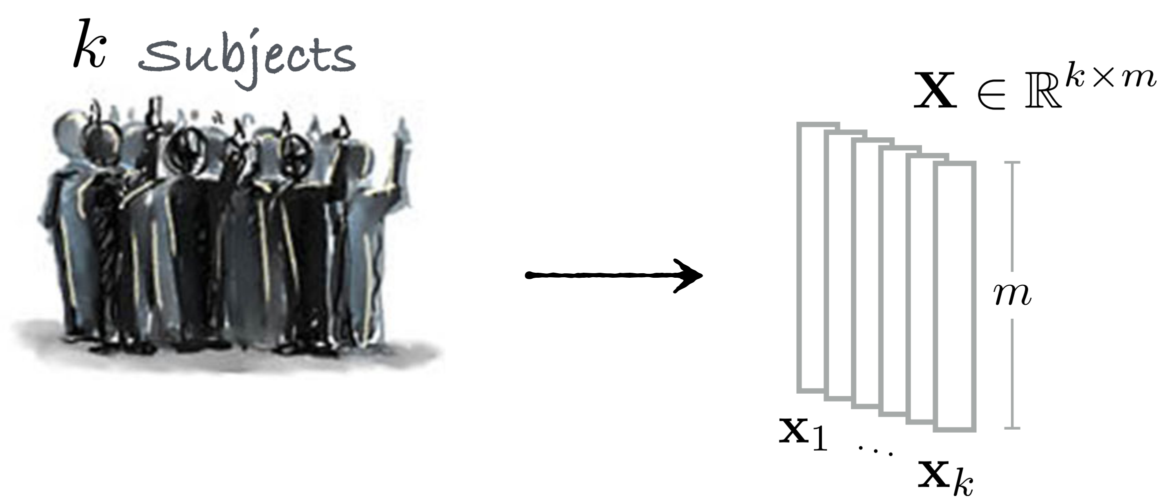
- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.

k subjects

What is CCA?

(Hotelling, 1936)

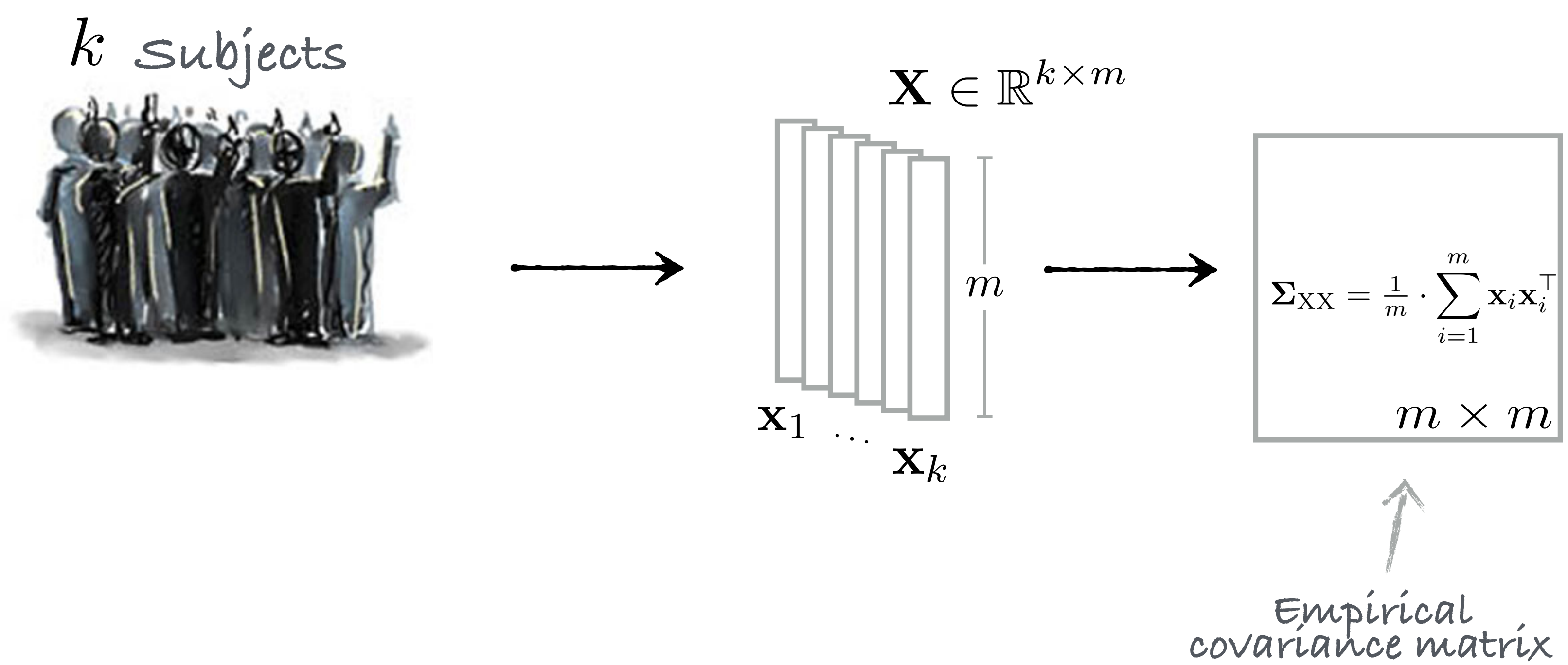
- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.



What is CCA?

(Hotelling, 1936)

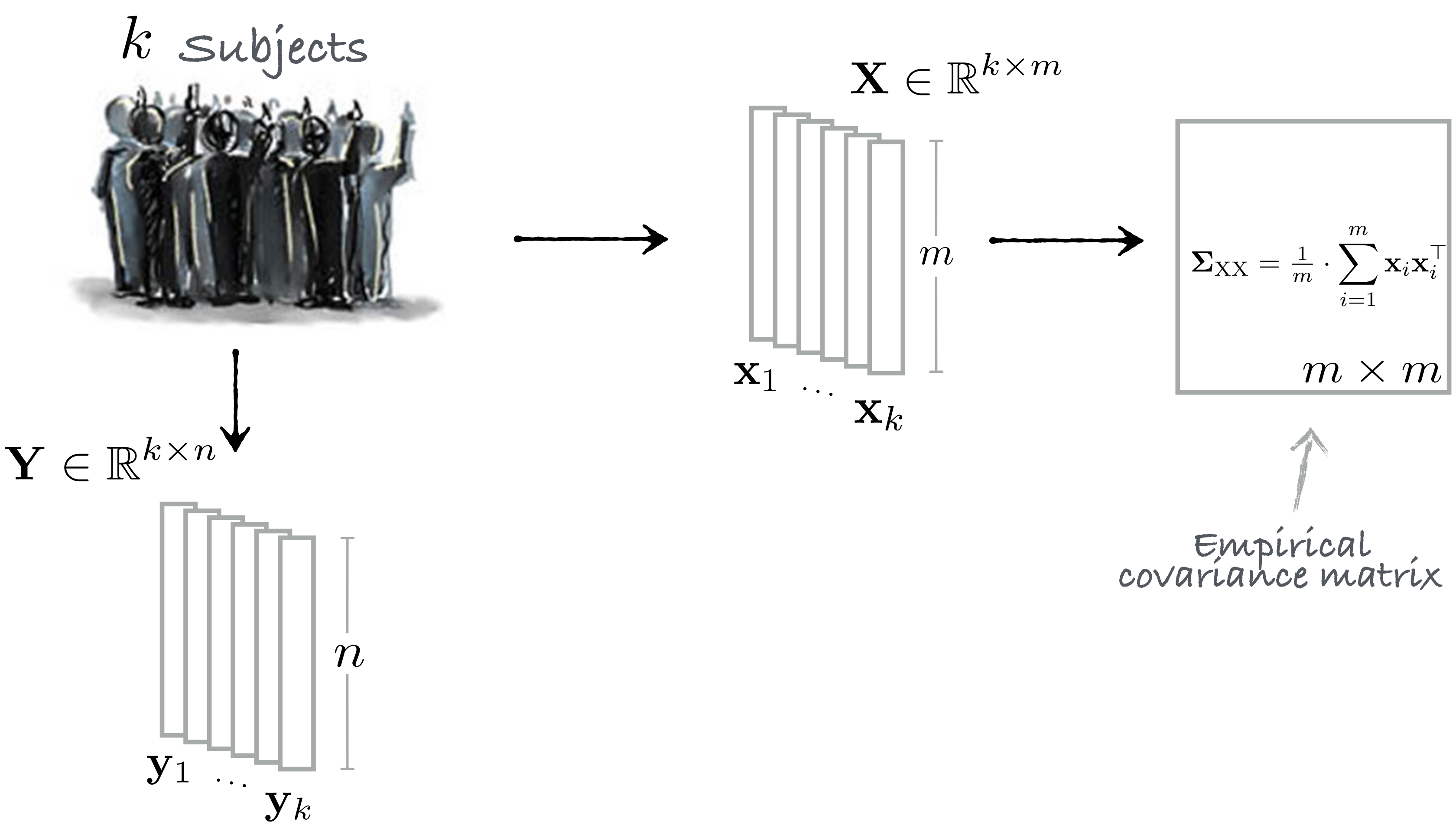
- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.



What is CCA?

(Hotelling, 1936)

- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.



What is CCA?

(Hotelling, 1936)

- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.

k subjects

$\mathbf{X} \in \mathbb{R}^{k \times m}$

$\mathbf{x}_1 \ \ldots \ \mathbf{x}_k$, m

$$\mathbf{\Sigma}_{\mathrm{XX}} = \frac{1}{m} \cdot \sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^\top$$

$m \times m$

Empirical covariance matrix

$\mathbf{Y} \in \mathbb{R}^{k \times n}$

$\mathbf{y}_1 \ \ldots \ \mathbf{y}_k$, n

$$\mathbf{\Sigma}_{\mathrm{YY}} = \frac{1}{n} \cdot \sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^\top$$

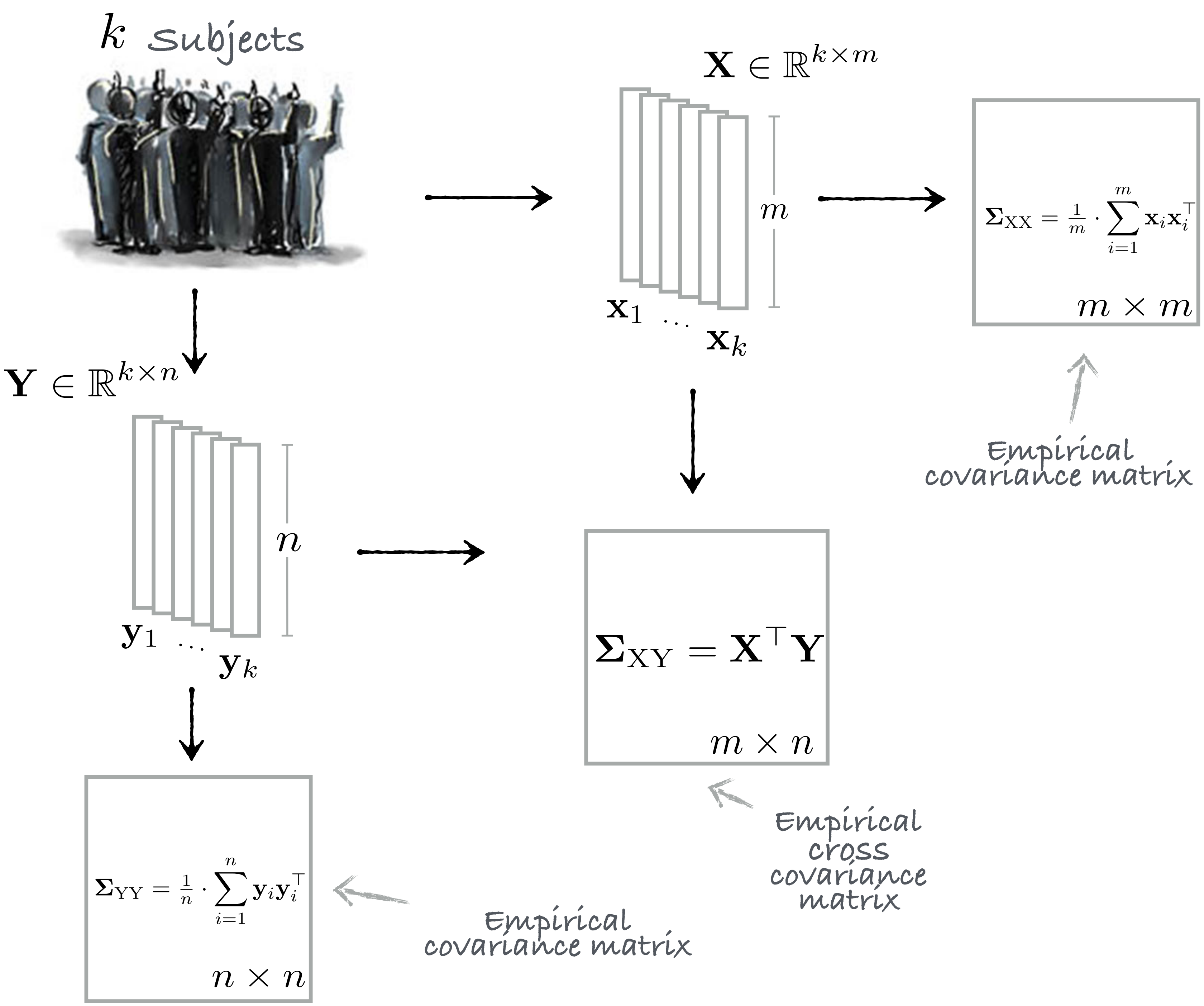
$n \times n$

Empirical covariance matrix

What is CCA?

(Hotelling, 1936)

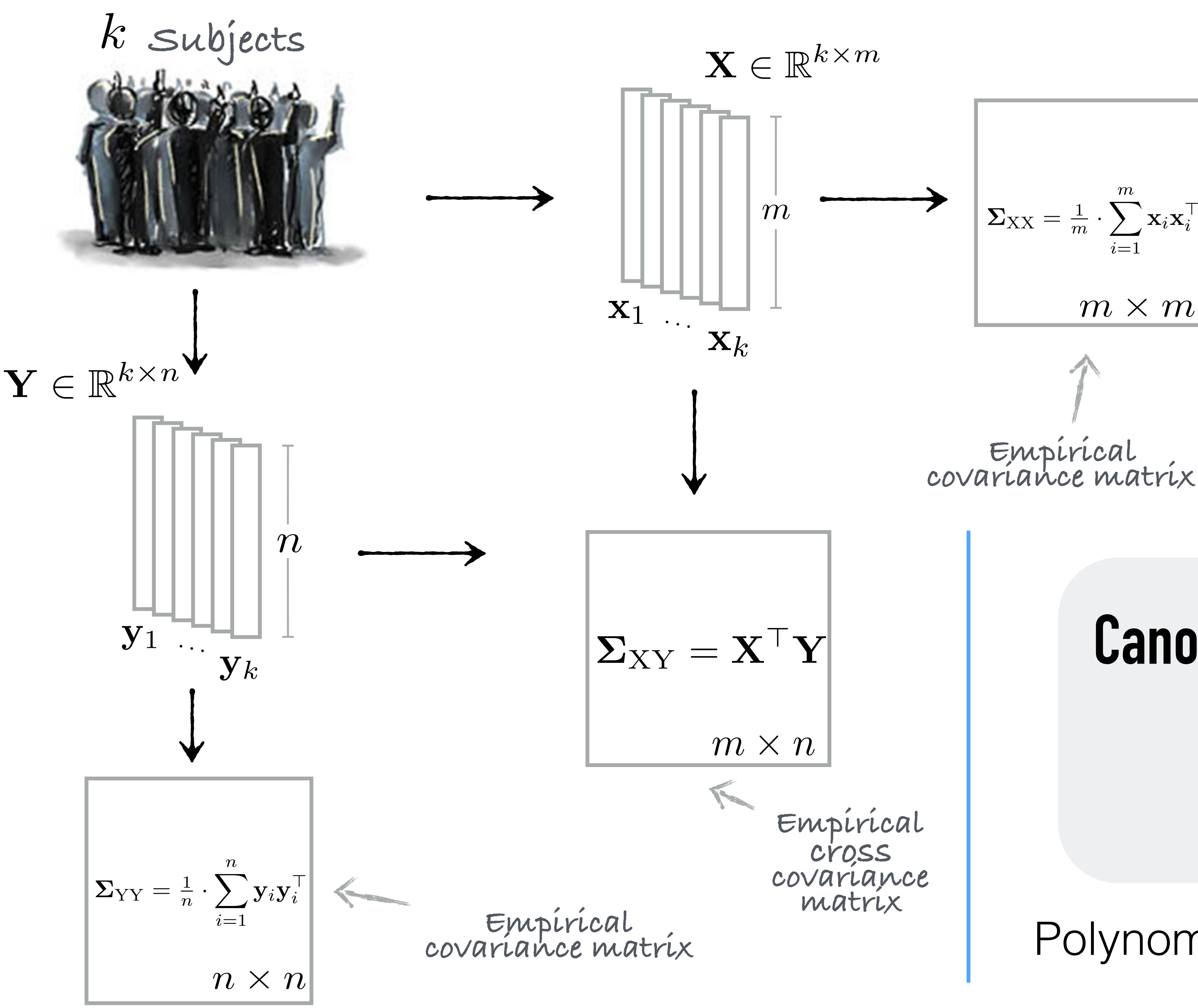
- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.



What is CCA?

(Hotelling, 1936)

- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.



Canonical Correlation Analysis (CCA)

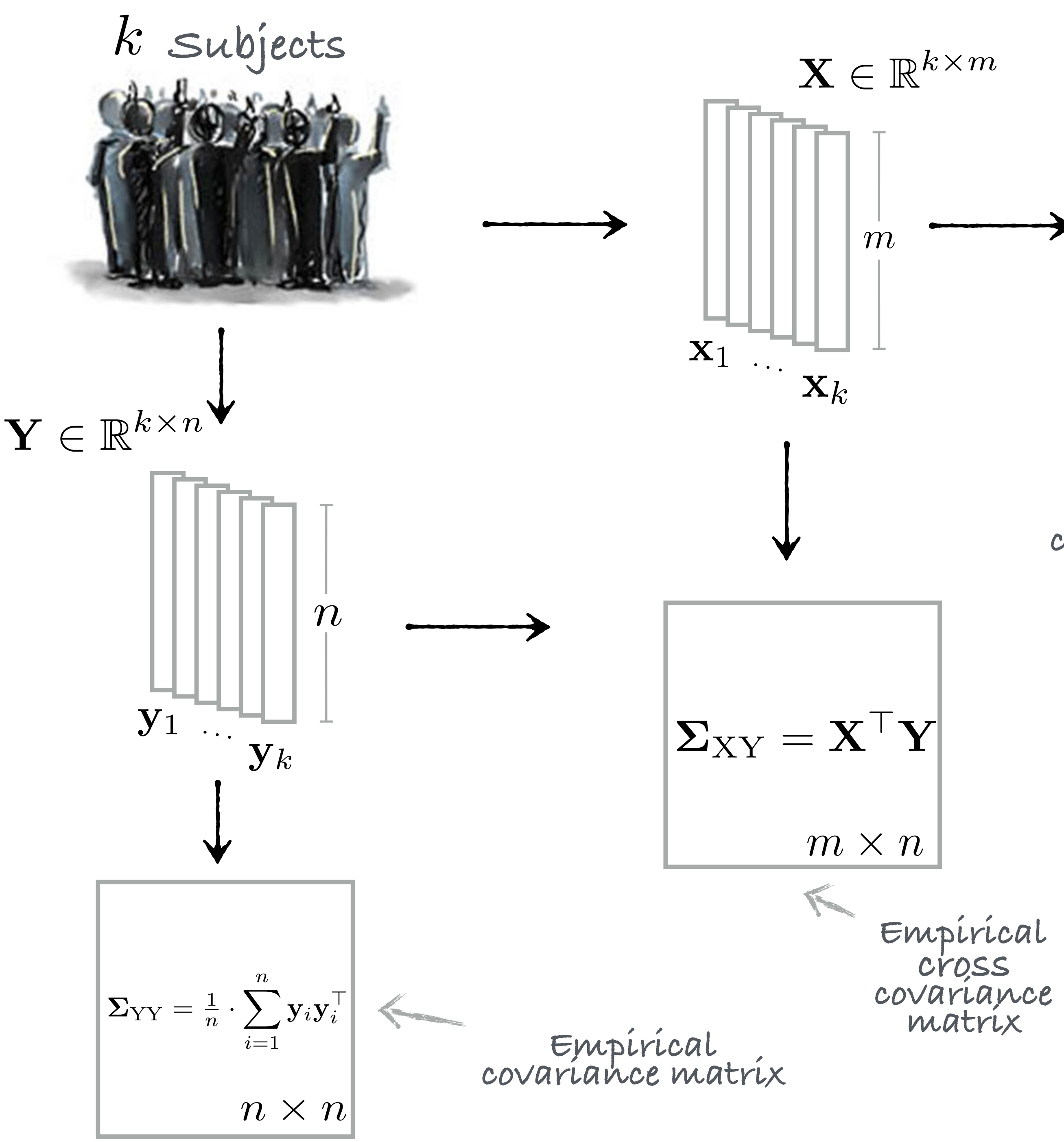
$$\max_{\substack{\mathbf{u}^\top \Sigma_{XX} \mathbf{u} = 1, \\ \mathbf{v}^\top \Sigma_{YY} \mathbf{v} = 1}} \mathbf{u}^\top \Sigma_{XY} \mathbf{v}$$

Polynomially solvable as a generalized EVD

What is CCA?

(Hotelling, 1936)

- **Input**: datapoints (samples) from two sets of variables.
- **Objective**: seek linear combination of original variables from each set that are **maximally correlated**.



Applications:

- Cognitive neuroscience (Posner et al., 1988; Poldrack, 2006)
- Genetics & molecular biology (Pollack et al., 2002; Morley et al., 2004; Stranger et al., 2007)
- Natural language processing (Dhillon et al., 2011)
- Speech recognition (Arora & Livescu, 2013)

Canonical Correlation Analysis (CCA)

$$\max_{\substack{\mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XX}} \mathbf{u}=1, \\ \mathbf{v}^\top \mathbf{\Sigma}_{\mathrm{YY}} \mathbf{v}=1}} \mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

Polynomially solvable as a generalized EVD

Necessity of sparsity in CCA

- **Common situation**: datasets of much more variables than # of samples.

  $\longrightarrow$ ***ill-posed objective***
- **Solution**: model regularization via constraints/regularizers

  $\longrightarrow$ ***common practice: sparsity***

Necessity of sparsity in CCA

- **Common situation**: datasets of much more variables than # of samples.
  $\longrightarrow$ ***ill-posed objective***
- **Solution**: model regularization via constraints/regularizers
  $\longrightarrow$ ***common practice: sparsity***

Sparse CCA

$$\max_{\substack{\mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XX}} \mathbf{u}=1,\ \|\mathbf{u}\|_0 \leq s_x,\\ \mathbf{v}^\top \mathbf{\Sigma}_{\mathrm{YY}} \mathbf{v}=1,\ \|\mathbf{v}\|_0 \leq s_y}} \mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

NP-hard

- Dates back to: (Thorndike, 1976; Thompson, 1984)
- Recent approaches:
  I. Convex relaxation & Lagrangian approaches (Torres et al., 2007; Hardoon & Shawe-Taylor, 2007; 2011; Gao et al., 2014)
  II. DC programming (Sriperumbudur et al., 2009)
  III. Greedy schemes (Wiesel et al., 2008)
  IV. Power method schemes (Tan et al, 2016)

Sparse diagonal CCA

(Witten et al., 2009; Parkhomenko et al. 2009)

Sparse diagonal CCA
(Witten et al., 2009; Parkhomenko et al. 2009)

- **Difficulties in solving sparse CCA**: handling sparsity + covariance constraints
- **(A) solution**: ``neglect" covariance constraints + use techniques used in PCA

Sparse CCA

$$\max_{\substack{\mathbf{u}^\top \boldsymbol{\Sigma}_{\mathrm{XX}} \mathbf{u}=1,\ \|\mathbf{u}\|_0 \leq s_x, \\ \mathbf{v}^\top \boldsymbol{\Sigma}_{\mathrm{YY}} \mathbf{v}=1,\ \|\mathbf{v}\|_0 \leq s_y}} \mathbf{u}^\top \boldsymbol{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

Sparse diagonal CCA

(Witten et al., 2009; Parkhomenko et al. 2009)

- **Difficulties in solving sparse CCA**: handling sparsity + covariance constraints
- **(A) solution**: ``neglect” covariance constraints + use techniques used in PCA

Sparse diagonal CCA

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\leq s_x,\\ \|\mathbf{v}\|_2=1,\ \|\mathbf{v}\|_0\leq s_y}} \mathbf{u}^\top \boldsymbol{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

(still) NP-hard

- Also known as **Sparse SVD**
  (Yang et al., 2011; Lee et al., 2010)
- ``Diagonal” as we approximate Sparse CCA as:

$$\boldsymbol{\Sigma}_{\mathrm{XX}} = \mathbf{I}_{m\times m} \quad \text{and} \quad \boldsymbol{\Sigma}_{\mathrm{YY}} = \mathbf{I}_{n\times n}$$

(Dudoit et al., 2002; Tibshirani et al., 2003)

Sparse diagonal CCA

(Witten et al., 2009; Parkhomenko et al. 2009)

- **Difficulties in solving sparse CCA**: handling sparsity + covariance constraints
- **(A) solution**: “neglect” covariance constraints + use techniques used in PCA

Sparse diagonal CCA

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\leq s_x, \\ \|\mathbf{v}\|_2=1,\ \|\mathbf{v}\|_0\leq s_y}} \mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

(still) NP-hard

- Also known as **Sparse SVD**
  (Yang et al., 2011; Lee et al., 2010)
- “Diagonal” as we approximate Sparse CCA as:

$$\mathbf{\Sigma}_{\mathrm{XX}} = \mathbf{I}_{m\times m} \quad \text{and} \quad \mathbf{\Sigma}_{\mathrm{YY}} = \mathbf{I}_{n\times n}$$

(Dudoit et al., 2002; Tibshirani et al., 2003)

Penalized matrix decomposition (PMD)

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_1\leq c_x, \\ \|\mathbf{v}\|_2=1,\ \|\mathbf{v}\|_1\leq c_y}} \mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

(Witten et al., 2009; Parkhomenko et al. 2009)

Sparse diagonal CCA

(Witten et al., 2009; Parkhomenko et al. 2009)

- **Difficulties in solving sparse CCA**: handling sparsity + covariance constraints
- **(A) solution**: “neglect” covariance constraints + use techniques used in PCA

Sparse diagonal CCA

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\leq s_x,\\ \|\mathbf{v}\|_2=1,\ \|\mathbf{v}\|_0\leq s_y}} \mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

(still) NP-hard

- Also known as **Sparse SVD**
  (Yang et al., 2011; Lee et al., 2010)
- “Diagonal” as we approximate Sparse CCA as:

$$\mathbf{\Sigma}_{\mathrm{XX}} = \mathbf{I}_{m\times m} \quad \text{and} \quad \mathbf{\Sigma}_{\mathrm{YY}} = \mathbf{I}_{n\times n}$$

(Dudoit et al., 2002; Tibshirani et al., 2003)

Penalized matrix decomposition (PMD)

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_1\leq c_x,\\ \|\mathbf{v}\|_2=1,\ \|\mathbf{v}\|_1\leq c_y}} \mathbf{u}^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

(Witten et al., 2009; Parkhomenko et al. 2009)

Sparse SVD

(Lee et al., 2010)

$$\min_{\mathbf{u},\mathbf{v}} \|\mathbf{\Sigma}_{\mathrm{XY}} - \mathbf{u}\mathbf{v}^\top\|_F^2 + \lambda_1 g_1(\mathbf{u}) + \lambda_2 g_2(\mathbf{v})$$

where $g_1,\ g_2$ are sparsity-inducing convex norms.

Sparse diagonal CCA: our approach

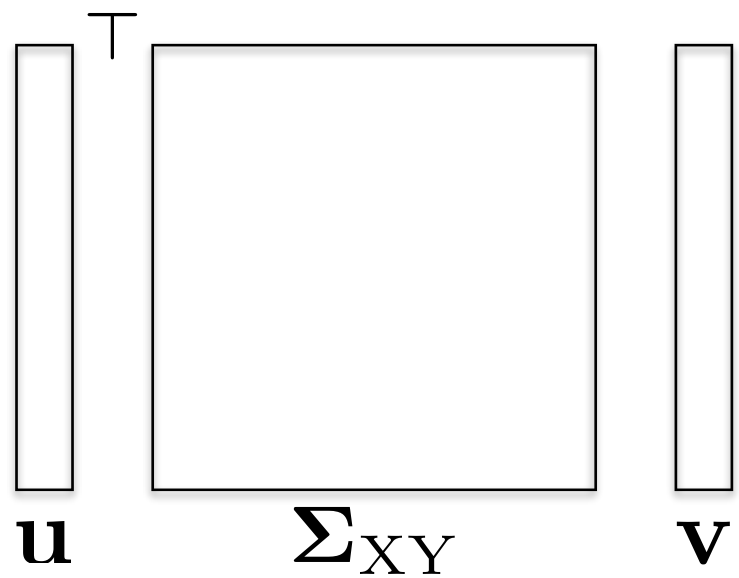
- **Intuition:** hardness lies in determining supports of $\mathbf{u}$ and $\mathbf{v}$.

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0\le s_y}} \mathbf{u}^\top \boldsymbol{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

Sparse diagonal CCA: our approach

- **Intuition:** hardness lies in determining supports of $\mathbf{u}$ and $\mathbf{v}$.

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0 \le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0 \le s_y}}$$



- Brute force approach finds **optimal** solution
- **Intractable:** $\binom{m}{s_x} \cdot \binom{n}{s_y}$

# of candidate supports

Sparse diagonal CCA: our approach

- **Intuition:** hardness lies in determining supports of $\mathbf{u}$ and $\mathbf{v}$.

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0 \leq s_x, \\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0 \leq s_y}} \mathbf{u}^\top \boldsymbol{\Sigma}_{\mathrm{XY}} \mathbf{v}$$

- Brute force approach finds **optimal** solution
- **Intractable:** $\binom{m}{s_x} \cdot \binom{n}{s_y}$

  # of candidate supports

Sparse diagonal CCA: our approach

- **Intuition:** hardness lies in determining supports of $\mathbf{u}$ and $\mathbf{v}$.

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0\le s_y}} \mathbf{u}^\top \Sigma_{\mathrm{XY}} \mathbf{v}$$

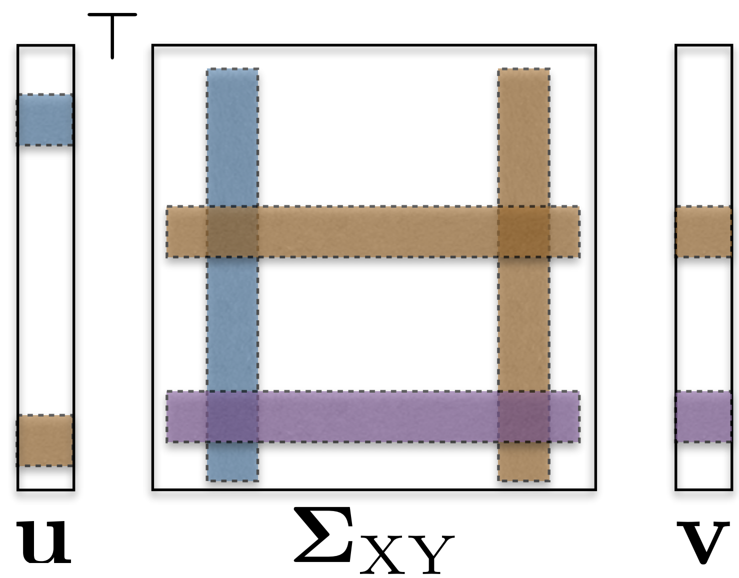
- Brute force approach finds **optimal** solution
- **Intractable:** $\binom{m}{s_x} \cdot \binom{n}{s_y}$

# of candidate supports

Sparse diagonal CCA: our approach

- **Intuition:** hardness lies in determining supports of $\mathbf{u}$ and $\mathbf{v}$.

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\leq s_x,\\ \|\mathbf{v}\|_2=1,\|\mathbf{v}\|_0\leq s_y}}$$



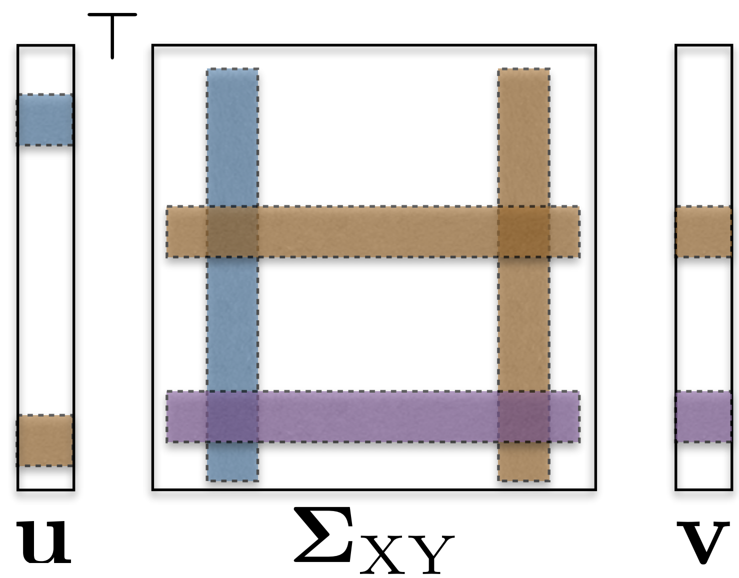
- Brute force approach finds **optimal** solution
- **Intractable:** $\binom{m}{s_x} \cdot \binom{n}{s_y}$

# of candidate supports

Sparse diagonal CCA: our approach

- **Intuition:** hardness lies in determining supports of $\mathbf{u}$ and $\mathbf{v}$.

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0\le s_y}}$$

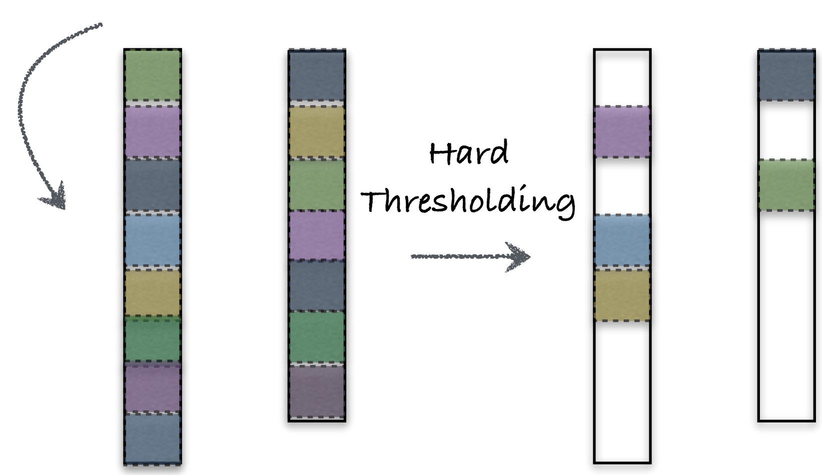


- Brute force approach finds **optimal** solution
- **Intractable:** $\binom{m}{s_x}\cdot\binom{n}{s_y}$

# of candidate supports

- ``**Greedy" approach:** hard-thresholding SVD

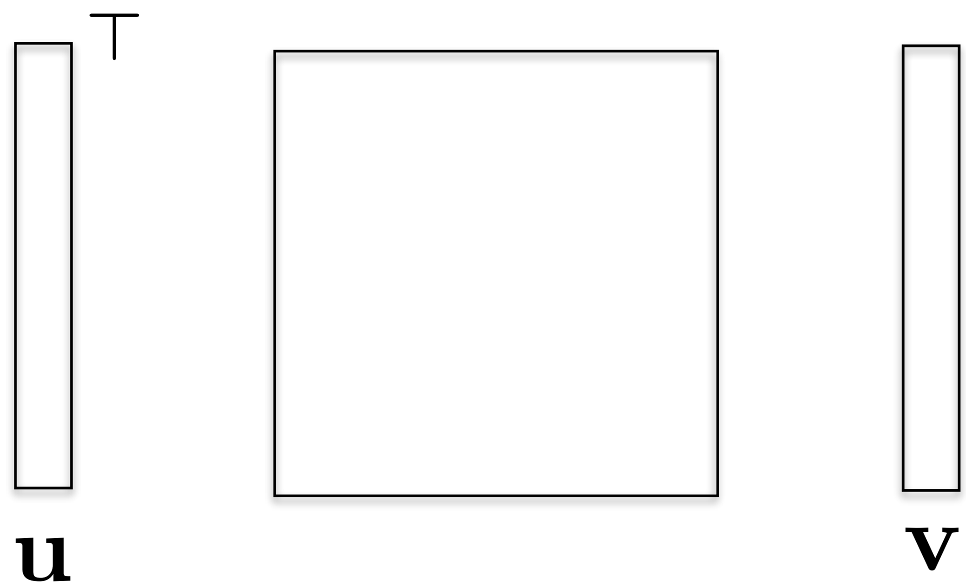
$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \cancel{\|\mathbf{u}\|_0\le s_x},\\ \|\mathbf{v}\|_2=1, \cancel{\|\mathbf{v}\|_0\le s_y}}} \mathbf{u}^\top \boldsymbol{\Sigma}_{\mathrm{XY}}\mathbf{v} \longrightarrow \texttt{[u,\textasciitilde,v]=svds(}\boldsymbol{\Sigma}_{\mathrm{XY}}\texttt{,1)}$$



- Polynomially solvable
- Heuristic: suboptimal guarantees and performance

Sparse diagonal CCA: our approach

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0 \le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0 \le s_y}}$$



Sparse diagonal CCA: our approach

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\leq s_x,\\ \|\mathbf{v}\|_2=1,\|\mathbf{v}\|_0\leq s_y}} \mathbf{u}^\top \overbrace{\underbrace{\mathbf{Q}_1}_{r} \mathbf{S} \mathbf{Q}_2^\top}^{\mathbf{B} \equiv (\boldsymbol{\Sigma}_{\mathrm{XY}})_r} \mathbf{v}$$

where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

Sparse diagonal CCA: our approach

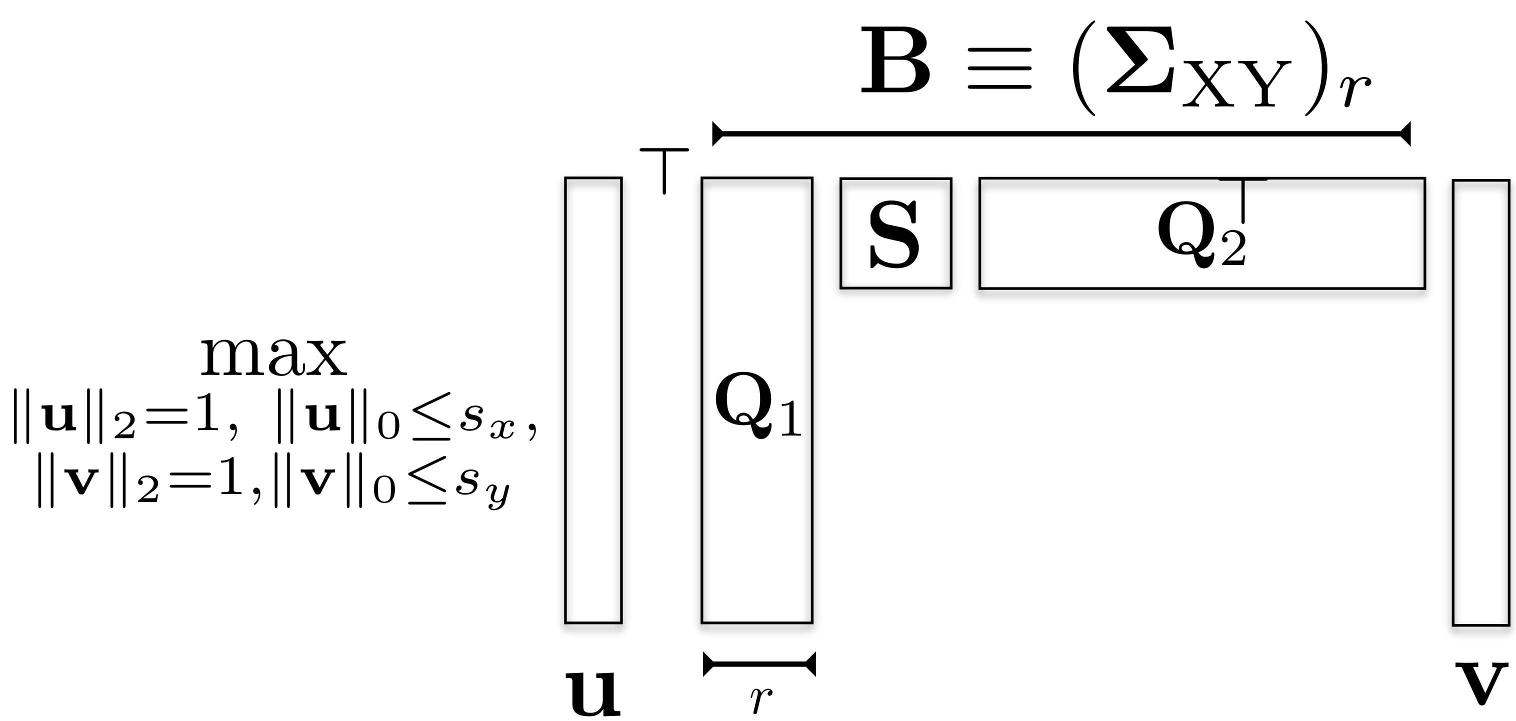
$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0 \le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0 \le s_y}} \mathbf{u}^\top \overbrace{\underbrace{\mathbf{Q}_1}_{r} \mathbf{S} \mathbf{Q}_2^\top}^{\mathbf{B} \equiv (\boldsymbol{\Sigma}_{\mathrm{XY}})_r} \mathbf{v}$$

where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

Intuition

- Explores a principal subspace of $\boldsymbol{\Sigma}_{\mathrm{XY}}$.
- Based on arguments that "*…the maximization of a low-rank Rayleigh quotient over combinatorial constraints is polynomially solvable.*" (Asteris et al., 2014; Kyrillidis et al., 2015)

Sparse diagonal CCA: our approach



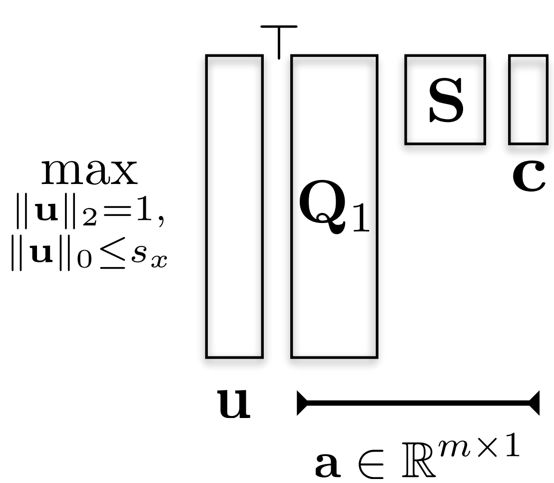
where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

Intuition

- Explores a principal subspace of $\mathbf{\Sigma}_{\mathrm{XY}}$.
- Based on arguments that "*…the maximization of a low-rank Rayleigh quotient over combinatorial constraints is polynomially solvable.*" (Asteris et al., 2014; Kyrillidis et al., 2015)

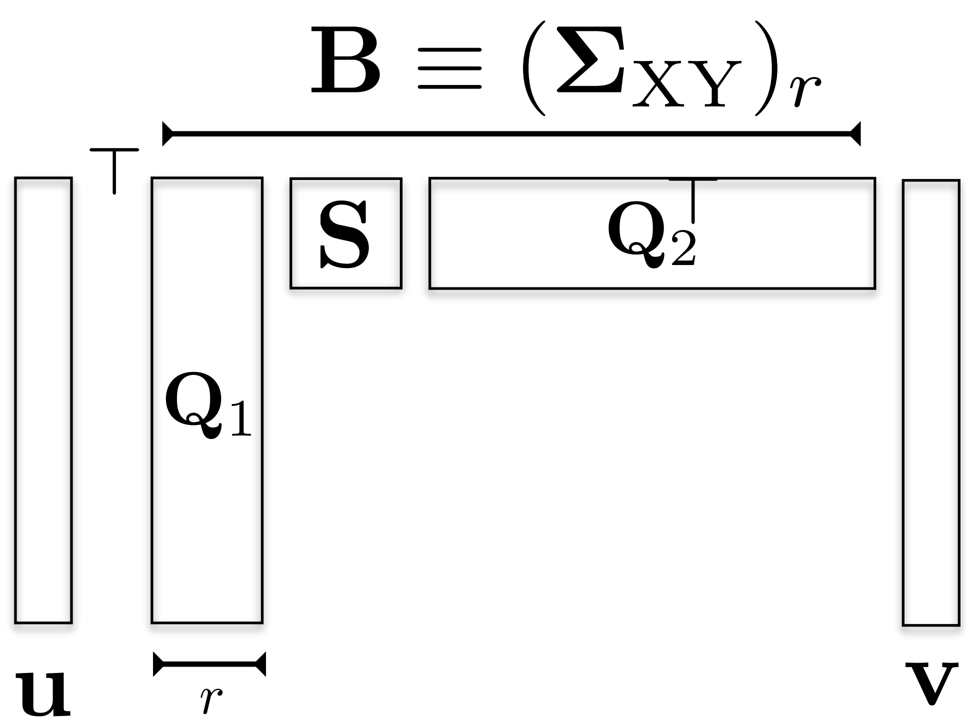
Approach: alternating optimization

- Step 1: select a $r \times 1$ random vector $\mathbf{c}$ to approximate $\mathbf{Q}_2^\top \mathbf{v} \in \mathbb{R}^{r\times 1}$ and compute:



Sparse diagonal CCA: our approach

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0\le s_y}}$$



where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

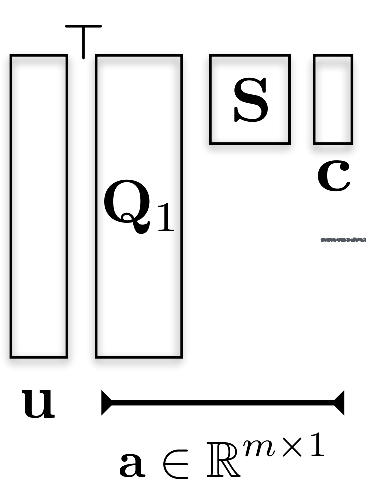
Intuition

- Explores a principal subspace of $\boldsymbol{\Sigma}_{\mathrm{XY}}$.
- Based on arguments that "*…the maximization of a low-rank Rayleigh quotient over combinatorial constraints is polynomially solvable.*" (Asteris et al., 2014; Kyrillidis et al., 2015)

Approach: alternating optimization

- Step 1: select a $r \times 1$ random vector $\mathbf{c}$ to approximate $\mathbf{Q}_2^\top \mathbf{v} \in \mathbb{R}^{r\times 1}$ and compute:

$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}}$$

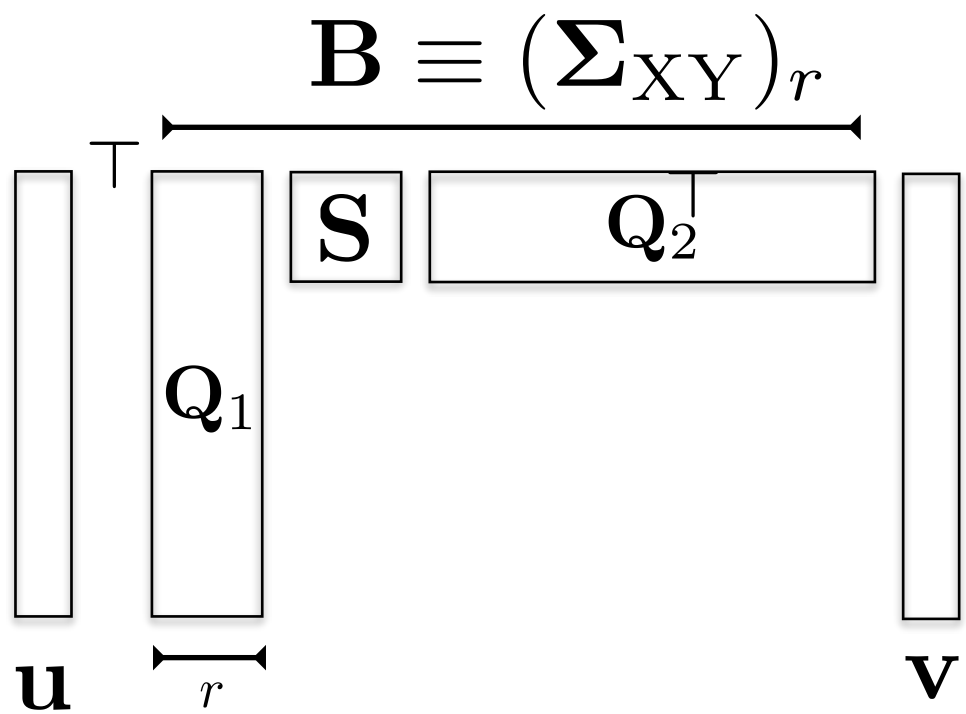


$$\longrightarrow \max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}} \mathbf{u}^\top \mathbf{a}$$

Easily computed in
O(m log(m)) time

Sparse diagonal CCA: our approach

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\leq s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0\leq s_y}}$$



where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

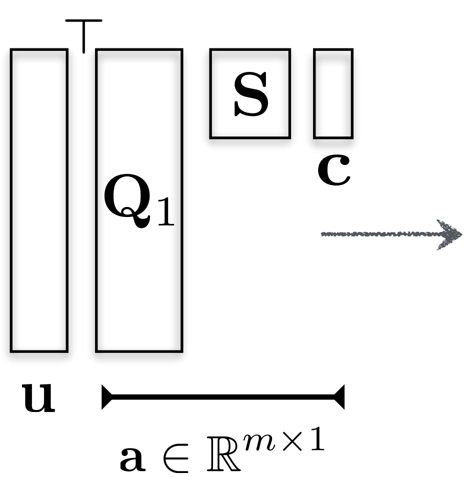
Intuition

- Explores a principal subspace of $\mathbf{\Sigma}_{\mathrm{XY}}$.
- Based on arguments that "*…the maximization of a low-rank Rayleigh quotient over combinatorial constraints is polynomially solvable.*" (Asteris et al., 2014; Kyrillidis et al., 2015)

Approach: alternating optimization

- Step 1: select a $r \times 1$ random vector $\mathbf{c}$ to approximate $\mathbf{Q}_2^\top \mathbf{v} \in \mathbb{R}^{r\times 1}$ and compute:

$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\leq s_x}}$$

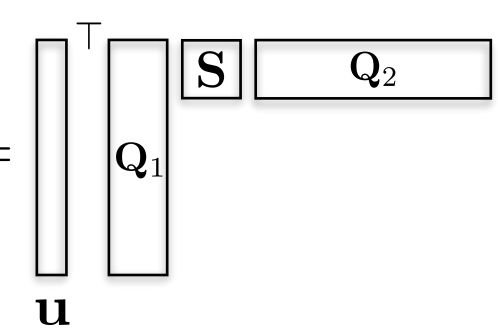


$$\rightarrow \quad \max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\leq s_x}} \mathbf{u}^\top \mathbf{a}$$

Easily computed in $O(m \log(m))$ time

- Step 2: Compute the maximizer of:

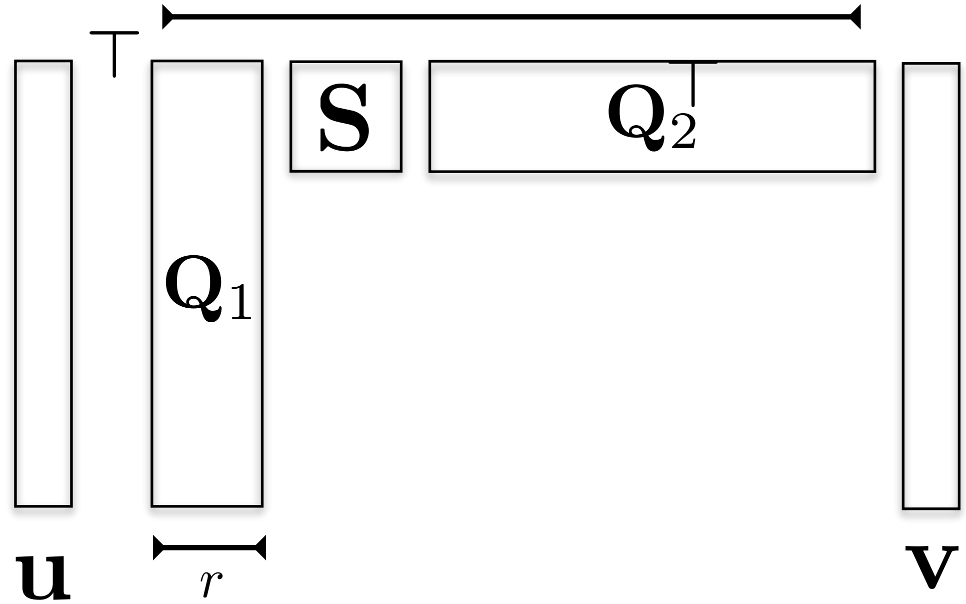
$$\max_{\substack{\|\mathbf{v}\|_2=1,\\ \|\mathbf{v}\|_0\leq s_y}} \mathbf{b}^\top \mathbf{v} \quad \text{where} \quad \mathbf{b}^\top =$$



Sparse diagonal CCA: our approach

$$\mathbf{B} \equiv (\boldsymbol{\Sigma}_{\mathrm{XY}})_r$$

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\le s_x,\\ \|\mathbf{v}\|_2=1,\|\mathbf{v}\|_0\le s_y}}$$



where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

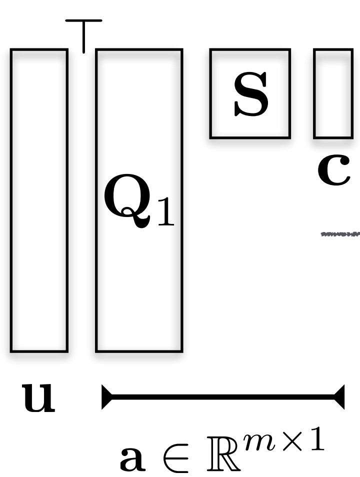
Intuition

- Explores a principal subspace of $\boldsymbol{\Sigma}_{\mathrm{XY}}$.
- Based on arguments that "*…the maximization of a low-rank Rayleigh quotient over combinatorial constraints is polynomially solvable.*" (Asteris et al., 2014; Kyrillidis et al., 2015)

Approach: alternating optimization

- Step 1: select a $r \times 1$ random vector $\mathbf{c}$ to approximate $\mathbf{Q}_2^\top \mathbf{v} \in \mathbb{R}^{r\times 1}$ and compute:

$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}}$$



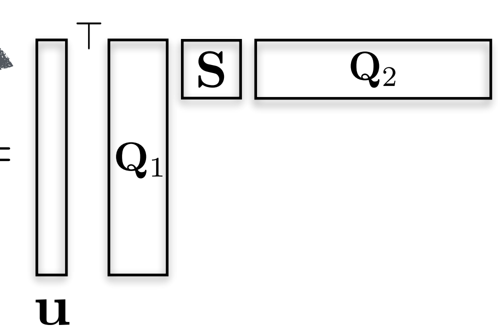
$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}} \mathbf{u}^\top \mathbf{a}$$

Easily computed in $O(m \log(m))$ time

- Step 2: Compute the maximizer of:

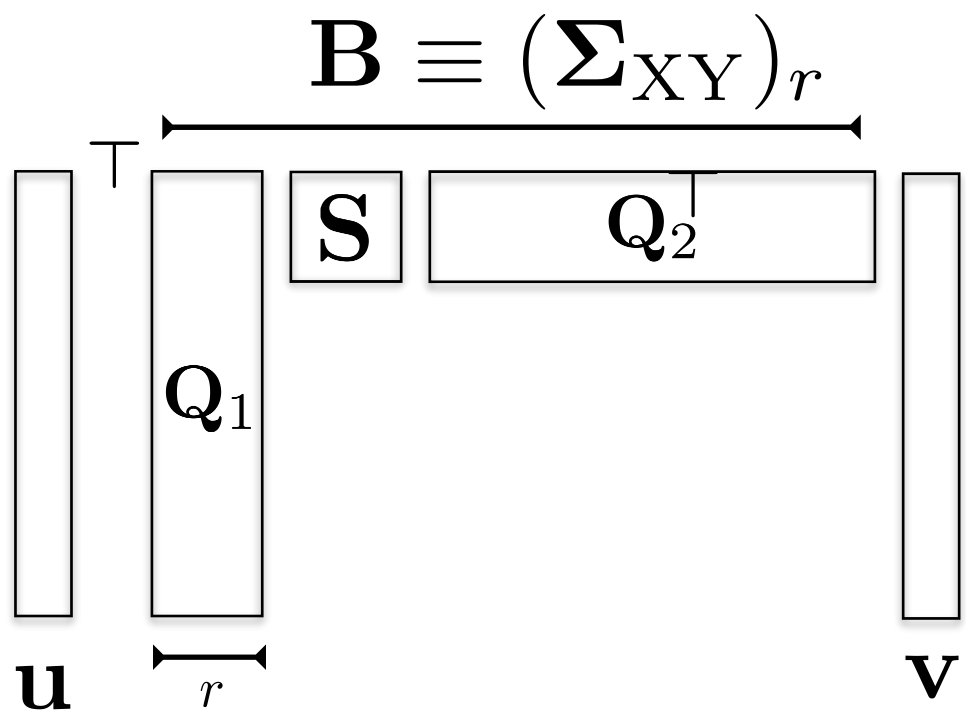
maximizer $\mathbf{u}$

$$\max_{\substack{\|\mathbf{v}\|_2=1,\\ \|\mathbf{v}\|_0\le s_y}} \mathbf{b}^\top \mathbf{v} \quad \text{where } \mathbf{b}^\top =$$



Sparse diagonal CCA: our approach

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\le s_x,\\ \|\mathbf{v}\|_2=1,\|\mathbf{v}\|_0\le s_y}}$$



where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

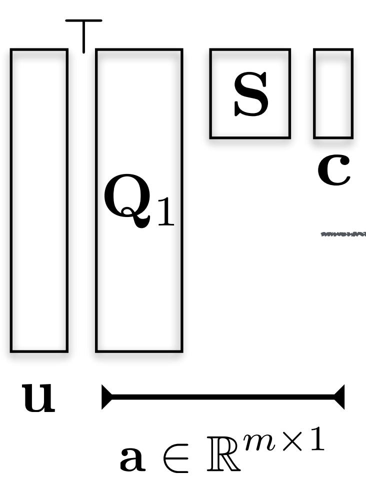
Intuition

- Explores a principal subspace of $\mathbf{\Sigma}_{\mathrm{XY}}$.
- Based on arguments that "*…the maximization of a low-rank Rayleigh quotient over combinatorial constraints is polynomially solvable.*" (Asteris et al., 2014; Kyrillidis et al., 2015)

Approach: alternating optimization

- Step 1: select a $r \times 1$ random vector $\mathbf{c}$ to approximate $\mathbf{Q}_2^\top \mathbf{v} \in \mathbb{R}^{r\times 1}$ and compute:

$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}}$$



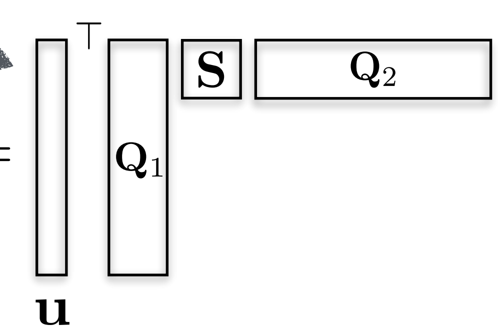
$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}} \mathbf{u}^\top \mathbf{a}$$

Easily computed in $O(m \log(m))$ time

maximizer $\mathbf{u}$

- Step 2: Compute the maximizer of:

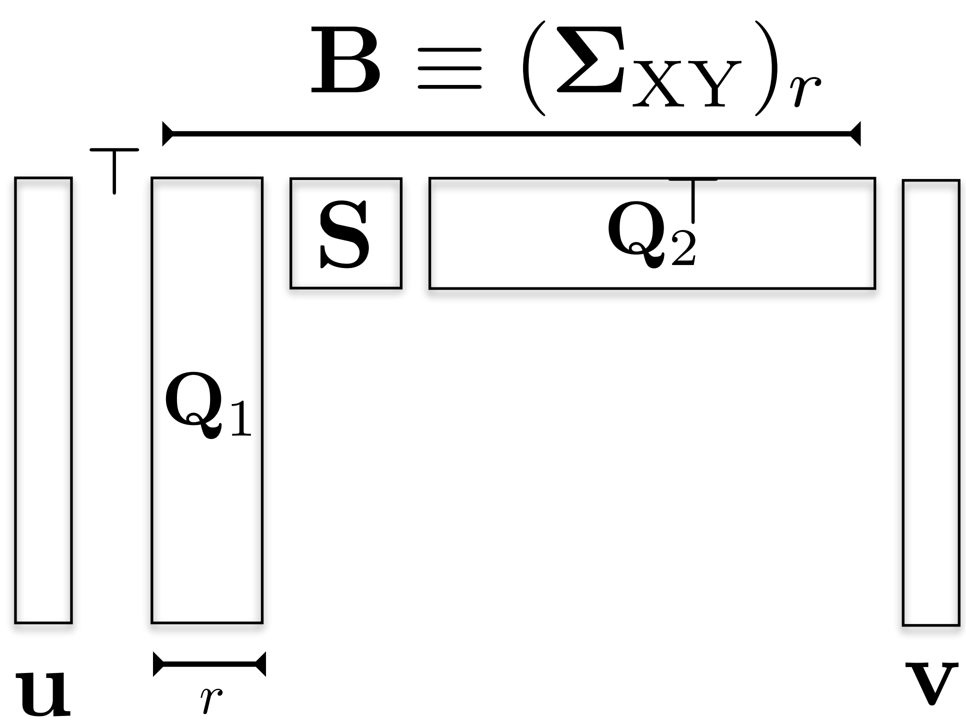
$$\max_{\substack{\|\mathbf{v}\|_2=1,\\ \|\mathbf{v}\|_0\le s_y}} \mathbf{b}^\top \mathbf{v} \quad \text{where } \mathbf{b}^\top =$$



Easily computed in $O(n \log(n))$ time

Sparse diagonal CCA: our approach

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0\le s_x,\\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0\le s_y}}$$



where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

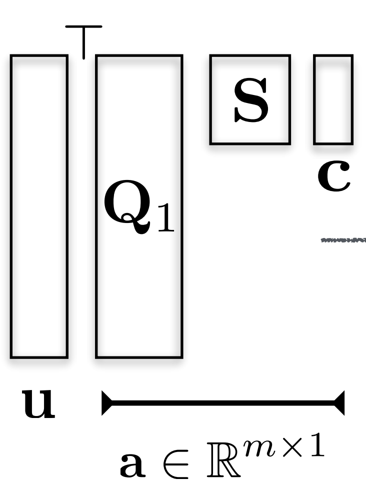
Intuition

- Explores a principal subspace of $\mathbf{\Sigma}_{\mathrm{XY}}$.
- Based on arguments that "*…the maximization of a low-rank Rayleigh quotient over combinatorial constraints is polynomially solvable.*" (Asteris et al., 2014; Kyrillidis et al., 2015)

Approach: alternating optimization

- Step 1: select a $r \times 1$ random vector $\mathbf{c}$ to approximate $\mathbf{Q}_2^\top \mathbf{v} \in \mathbb{R}^{r\times 1}$ and compute:

$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}}$$



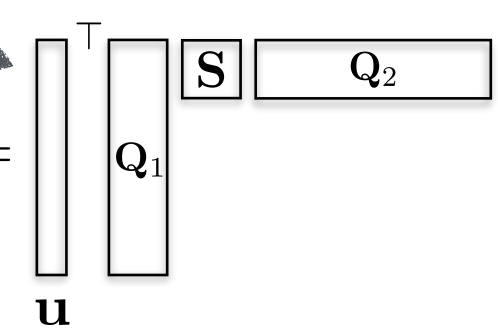
$$\max_{\substack{\|\mathbf{u}\|_2=1,\\ \|\mathbf{u}\|_0\le s_x}} \mathbf{u}^\top \mathbf{a}$$

Easily computed in $O(m \log(m))$ time

maximizer $\mathbf{u}$

- Step 2: Compute the maximizer of:

$$\max_{\substack{\|\mathbf{v}\|_2=1,\\ \|\mathbf{v}\|_0\le s_y}} \mathbf{b}^\top \mathbf{v} \quad \text{where } \mathbf{b}^\top =$$



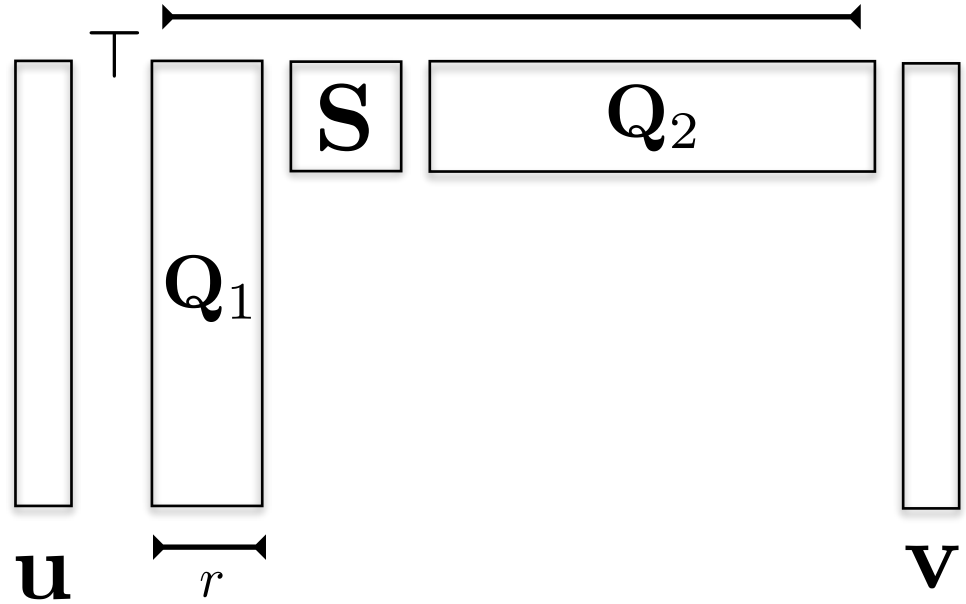
Easily computed in $O(n \log(n))$ time

…and repeat

Sparse diagonal CCA: our approach

$$\mathbf{B} \equiv (\boldsymbol{\Sigma}_{\mathrm{XY}})_r$$

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_0 \le s_x, \\ \|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0 \le s_y}}$$



where $\mathbf{A}_r$ is the best rank-r approximation of $\mathbf{A}$.

- If $\boldsymbol{\Sigma}_{\mathrm{XY}}$ is rank-1:

  our approach = greedy
- If $r = \min\{m, n\}$:

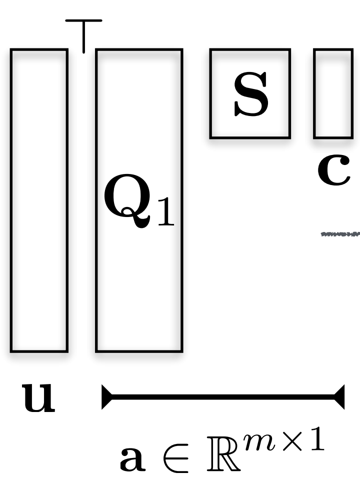
  our approach $\approx$ exhaustive
- If $1 \le r \ll \min\{m, n\}$:

  our approach spans the ground between

Approach: alternating optimization

- Step 1: select a $r \times 1$ random vector $\mathbf{c}$ to approximate $\mathbf{Q}_2^\top \mathbf{v} \in \mathbb{R}^{r \times 1}$ and compute:

$$\max_{\substack{\|\mathbf{u}\|_2=1, \\ \|\mathbf{u}\|_0 \le s_x}}$$



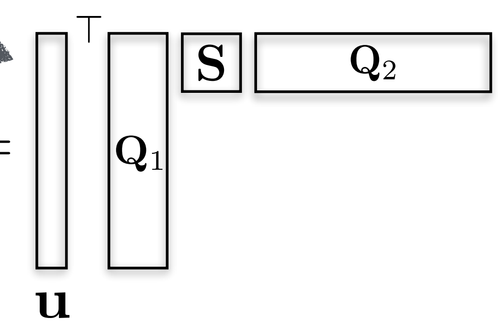
$$\max_{\substack{\|\mathbf{u}\|_2=1, \\ \|\mathbf{u}\|_0 \le s_x}} \mathbf{u}^\top \mathbf{a}$$

Easily computed in $O(m \log(m))$ time

- Step 2: Compute the maximizer of:

maximizer $\mathbf{u}$

$$\max_{\substack{\|\mathbf{v}\|_2=1, \\ \|\mathbf{v}\|_0 \le s_y}} \mathbf{b}^\top \mathbf{v} \quad \text{where } \mathbf{b}^\top =$$



Easily computed in $O(n \log(n))$ time

...and repeat

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \texttt{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$ $\quad \{ \mathbf{B} \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \}$
for $i = 1, \ldots, T$ **do**
  $\mathbf{c}_i \leftarrow \texttt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r\times r})\}$
  $\mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$
  $\mathbf{a}_i \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$
  $\mathbf{u}_i \leftarrow \arg\max_{\mathbf{u}\in\mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$
  $\mathbf{b}_i \leftarrow \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$
  $\mathbf{v}_i \leftarrow \arg\max_{\mathbf{v}\in\mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$
  $\mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$
end for
$i_0 \leftarrow \arg\max_{i\in[T]} \mathrm{obj}_i$
$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\boldsymbol{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

1: $\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V} \leftarrow \texttt{SVD}(\boldsymbol{\Sigma}_{\mathrm{XY}}, r)$ $\quad \{ \mathbf{B} \leftarrow \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \}$
2: **for** $i = 1, \ldots, T$ **do**
3: $\quad \mathbf{c}_i \leftarrow \texttt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r \times r})\}$
4: $\quad \mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$
5: $\quad \mathbf{a}_i \leftarrow \mathbf{U}\boldsymbol{\Sigma}\mathbf{c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$
6: $\quad \mathbf{u}_i \leftarrow \arg\max_{\mathbf{u} \in \mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$
7: $\quad \mathbf{b}_i \leftarrow \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$
8: $\quad \mathbf{v}_i \leftarrow \arg\max_{\mathbf{v} \in \mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$
9: $\quad \mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$
10: **end for**
11: $i_0 \leftarrow \arg\max_{i \in [T]} \mathrm{obj}_i$
12: $(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

SVD calculation in $O(mnr)$

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathtt{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$ $\{\ \mathbf{B} \leftarrow \mathbf{U\Sigma V}^\top\ \}$
for $i = 1, \dots, T$ **do**
  $\mathbf{c}_i \leftarrow \mathtt{randn(r)}$ $\{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r\times r})\}$
  $\mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$
  $\mathbf{a}_i \leftarrow \mathbf{U\Sigma c}_i$ $\{\mathbf{a}_i \in \mathbb{R}^m\}$
  $\mathbf{u}_i \leftarrow \arg\max_{\mathbf{u}\in\mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\{\mathsf{P}_{\mathcal{U}}(\cdot)\}$
  $\mathbf{b}_i \leftarrow \mathbf{V\Sigma U}^\top \mathbf{u}_i$ $\{\mathbf{b}_i \in \mathbb{R}^n\}$
  $\mathbf{v}_i \leftarrow \arg\max_{\mathbf{v}\in\mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\{\mathsf{P}_{\mathcal{V}}(\cdot)\}$
  $\mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$
end for
$i_0 \leftarrow \arg\max_{i\in[T]} \mathrm{obj}_i$
$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

SVD calculation in $O(mnr)$

Randomly select a direction in the low-rank subspace

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\text{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \texttt{SVD}(\mathbf{\Sigma}_{\text{XY}}, r)$ $\quad \{\ \mathbf{B} \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\ \}$
for $i = 1, \dots, T$ **do**
  $\mathbf{c}_i \leftarrow \texttt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r \times r})\}$
  $\mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$
  $\mathbf{a}_i \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$
  $\mathbf{u}_i \leftarrow \arg\max_{\mathbf{u} \in \mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$
  $\mathbf{b}_i \leftarrow \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$
  $\mathbf{v}_i \leftarrow \arg\max_{\mathbf{v} \in \mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$
  $\text{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$
end for
$i_0 \leftarrow \arg\max_{i \in [T]} \text{obj}_i$
$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

SVD calculation in O(mnr)

Randomly select a direction in the low-rank subspace

Matrix-vector multiplication

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathtt{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$ $\quad \{\, \mathbf{B} \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \,\}$
for $i = 1, \ldots, T$ **do**
  $\mathbf{c}_i \leftarrow \mathtt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r\times r})\}$
  $\mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$
  $\mathbf{a}_i \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$
  $\mathbf{u}_i \leftarrow \arg\max_{\mathbf{u}\in\mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$
  $\mathbf{b}_i \leftarrow \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$
  $\mathbf{v}_i \leftarrow \arg\max_{\mathbf{v}\in\mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$
  $\mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$
end for
$i_0 \leftarrow \arg\max_{i\in[T]} \mathrm{obj}_i$
$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

SVD calculation in O(mnr)

Randomly select a direction in the low-rank subspace

Matrix-vector multiplication

O(m) or O(m log m) - sorting operation

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathtt{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$ $\quad \{\ \mathbf{B} \leftarrow \mathbf{U\Sigma V}^\top\ \}$
for $i = 1, \dots, T$ **do**
$\quad \mathbf{c}_i \leftarrow \mathtt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r \times r})\}$
$\quad \mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$
$\quad \mathbf{a}_i \leftarrow \mathbf{U\Sigma c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$
$\quad \mathbf{u}_i \leftarrow \arg\max_{\mathbf{u} \in \mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$
$\quad \mathbf{b}_i \leftarrow \mathbf{V\Sigma U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$
$\quad \mathbf{v}_i \leftarrow \arg\max_{\mathbf{v} \in \mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$
$\quad \mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$
end for
$i_0 \leftarrow \arg\max_{i \in [T]} \mathrm{obj}_i$
$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

SVD calculation in O(mnr)

Randomly select a direction in the low-rank subspace

Matrix-vector multiplication

O(m) or O(m log m) - sorting operation

Matrix-vector multiplication

O(n) or O(n log n) - sorting operation

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

Algorithm		Computational complexity
$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathtt{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$	$\{\, \mathbf{B} \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \,\}$	SVD calculation in O(mnr)
for $i = 1, \ldots, T$ **do**		
$\quad \mathbf{c}_i \leftarrow \mathtt{randn(r)}$	$\{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r \times r})\}$	Randomly select a direction in the low-rank subspace
$\quad \mathbf{c}_i \leftarrow \mathbf{c}_i / \lVert \mathbf{c}_i \rVert_2$		
$\quad \mathbf{a}_i \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{c}_i$	$\{\mathbf{a}_i \in \mathbb{R}^m\}$	Matrix-vector multiplication
$\quad \mathbf{u}_i \leftarrow \arg\max_{\mathbf{u} \in \mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$	$\{\mathsf{P}_{\mathcal{U}}(\cdot)\}$	O(m) or O(m log m) - sorting operation
$\quad \mathbf{b}_i \leftarrow \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{u}_i$	$\{\mathbf{b}_i \in \mathbb{R}^n\}$	Matrix-vector multiplication
$\quad \mathbf{v}_i \leftarrow \arg\max_{\mathbf{v} \in \mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$	$\{\mathsf{P}_{\mathcal{V}}(\cdot)\}$	O(n) or O(n log n) - sorting operation
$\quad \mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$		
end for		
$i_0 \leftarrow \arg\max_{i \in [T]} \mathrm{obj}_i$		Picks the best pair of (u, v) - **PARALLELIZABLE**
$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$		

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathrm{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$ $\quad \{\, \mathbf{B} \leftarrow \mathbf{U\Sigma V}^\top \,\}$

for $i = 1, \ldots, T$ **do**

$\quad \mathbf{c}_i \leftarrow \texttt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r \times r})\}$

$\quad \mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$

$\quad \mathbf{a}_i \leftarrow \mathbf{U\Sigma c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$

$\quad \mathbf{u}_i \leftarrow \arg\max_{\mathbf{u} \in \mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$

$\quad \mathbf{b}_i \leftarrow \mathbf{V\Sigma U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$

$\quad \mathbf{v}_i \leftarrow \arg\max_{\mathbf{v} \in \mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$

$\quad \mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$

end for

$i_0 \leftarrow \arg\max_{i \in [T]} \mathrm{obj}_i$

$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

SVD calculation in O(mnr)

Randomly select a direction in the low-rank subspace

Matrix-vector multiplication

O(m) or O(m log m) - sorting operation

Matrix-vector multiplication

O(n) or O(n log n) - sorting operation

Picks the best pair of (u, v) - **PARALLELIZABLE**

Theorem 1. *For any real* $m \times n$ *matrix* $\mathbf{\Sigma}_{\mathrm{XY}}$, $\epsilon \in (0,1)$, *and* $r \leq \max\{m, n\}$, *Algorithm 1 with input* $\mathbf{\Sigma}_{\mathrm{XY}}$, r, *and* $\mathrm{T} = \widetilde{O}\big(2^{r \cdot \log_2(2/\epsilon)}\big)$ *outputs* $\mathbf{u}_\sharp \in \mathcal{U}$ *and* $\mathbf{v}_\sharp \in \mathcal{V}$ *such that*

$$\mathbf{u}_\sharp^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\sharp \geq \mathbf{u}_\star^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\star - \epsilon \cdot \sigma_1(\mathbf{\Sigma}_{\mathrm{XY}}) - 2\sigma_{r+1}(\mathbf{\Sigma}_{\mathrm{XY}}),$$

in time $\mathrm{T}_{\mathrm{SVD}}(r) + O\big(\mathrm{T} \cdot (\mathrm{T}_{\mathcal{U}} + \mathrm{T}_{\mathcal{V}} + r \cdot \max\{m, n\})\big)$.

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathtt{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$ $\quad\{\, \mathbf{B} \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \,\}$ ← SVD calculation in O(mnr)

for $i = 1, \dots, T$ **do**

$\quad \mathbf{c}_i \leftarrow \mathtt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r\times r})\}$ ← Randomly select a direction in the low-rank subspace

$\quad \mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$

$\quad \mathbf{a}_i \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$ ← Matrix-vector multiplication

$\quad \mathbf{u}_i \leftarrow \arg\max_{\mathbf{u}\in\mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$ ← O(m) or O(m log m) - sorting operation

$\quad \mathbf{b}_i \leftarrow \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$ ← Matrix-vector multiplication

$\quad \mathbf{v}_i \leftarrow \arg\max_{\mathbf{v}\in\mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$ ← O(n) or O(n log n) - sorting operation

$\quad \mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$

end for

$i_0 \leftarrow \arg\max_{i\in[T]} \mathrm{obj}_i$ ← Picks the best pair of (u, v) - **PARALLELIZABLE**

$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

Global guarantee with additive error

Theorem 1. *For any real $m \times n$ matrix $\mathbf{\Sigma}_{\mathrm{XY}}$, $\epsilon \in (0,1)$, and $r \leq \max\{m,n\}$, Algorithm 1 with input $\mathbf{\Sigma}_{\mathrm{XY}}$, r, and $\mathrm{T} = \widetilde{O}(2^{r\cdot\log_2(2/\epsilon)})$ outputs $\mathbf{u}_\sharp \in \mathcal{U}$ and $\mathbf{v}_\sharp \in \mathcal{V}$ such that*

$$\boxed{\mathbf{u}_\sharp^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\sharp \geq \mathbf{u}_\star^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\star - \epsilon \cdot \sigma_1(\mathbf{\Sigma}_{\mathrm{XY}}) - 2\sigma_{r+1}(\mathbf{\Sigma}_{\mathrm{XY}}),}$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O\big(\mathrm{T}\cdot(\mathrm{T}_{\mathcal{U}} + \mathrm{T}_{\mathcal{V}} + r\cdot\max\{m,n\})\big)$.

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathrm{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$ $\quad \{ \mathbf{B} \leftarrow \mathbf{U\Sigma V}^\top \}$ ← SVD calculation in O(mnr)

for $i = 1, \ldots, T$ **do**

$\quad \mathbf{c}_i \leftarrow \texttt{randn(r)}$ $\quad \{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r \times r})\}$ ← Randomly select a direction in the low-rank subspace

$\quad \mathbf{c}_i \leftarrow \mathbf{c}_i / \|\mathbf{c}_i\|_2$

$\quad \mathbf{a}_i \leftarrow \mathbf{U\Sigma c}_i$ $\quad \{\mathbf{a}_i \in \mathbb{R}^m\}$ ← Matrix-vector multiplication

$\quad \mathbf{u}_i \leftarrow \arg\max_{\mathbf{u} \in \mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$ $\quad \{\mathsf{P}_{\mathcal{U}}(\cdot)\}$ ← O(m) or O(m log m) - sorting operation

$\quad \mathbf{b}_i \leftarrow \mathbf{V\Sigma U}^\top \mathbf{u}_i$ $\quad \{\mathbf{b}_i \in \mathbb{R}^n\}$ ← Matrix-vector multiplication

$\quad \mathbf{v}_i \leftarrow \arg\max_{\mathbf{v} \in \mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$ $\quad \{\mathsf{P}_{\mathcal{V}}(\cdot)\}$ ← O(n) or O(n log n) - sorting operation

$\quad \mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$

end for

$i_0 \leftarrow \arg\max_{i \in [T]} \mathrm{obj}_i$ ← Picks the best pair of (u, v) - **PARALLELIZABLE**

$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$

Computational complexity

Global guarantee with additive error

Theorem 1. *For any real $m \times n$ matrix $\mathbf{\Sigma}_{\mathrm{XY}}$, $\epsilon \in (0,1)$, and $r \leq \max\{m,n\}$, Algorithm 1 with input $\mathbf{\Sigma}_{\mathrm{XY}}$, r, and $\mathrm{T} = \widetilde{O}(2^{r \cdot \log_2(2/\epsilon)})$ outputs $\mathbf{u}_\sharp \in \mathcal{U}$ and $\mathbf{v}_\sharp \in \mathcal{V}$ such that*

$$\boxed{\mathbf{u}_\sharp^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\sharp \geq \mathbf{u}_\star^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\star - \epsilon \cdot \sigma_1(\mathbf{\Sigma}_{\mathrm{XY}}) - 2\sigma_{r+1}(\mathbf{\Sigma}_{\mathrm{XY}}),}$$

in time $\mathrm{T}_{\mathsf{SVD}}(r) + O(\mathrm{T} \cdot (\mathrm{T}_{\mathcal{U}} + \mathrm{T}_{\mathcal{V}} + r \cdot \max\{m,n\}))$.

Theorem 2. *If $\mathcal{V} = \{\mathbf{v} : \|\mathbf{v}\|_2 = 1\}$, i.e., if no constraint is imposed on variable $\mathbf{v}$ besides unit length, then Algorithm 1 under the same configuration as that in Theorem 1 outputs $\mathbf{u}_\sharp \in \mathcal{U}$ and $\mathbf{v}_\sharp \in \mathcal{V}$ such that*

$$\mathbf{u}_\sharp^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\sharp \geq (1-\epsilon) \cdot \mathbf{u}_\star^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\star - 2 \cdot \sigma_{r+1}(\mathbf{\Sigma}_{\mathrm{XY}}).$$

Sparse diagonal CCA: our approach

Algorithm 1 SpanCCA

input : $\mathbf{\Sigma}_{\mathrm{XY}}$, a real $m \times n$ matrix.
$r \in \mathbb{N}_+$, the rank of the approximation to be used.
$\mathrm{T} \in \mathbb{N}_+$, the number of samples/iterations.

output $\mathbf{u}_\sharp \in \mathcal{U}$, $\mathbf{v}_\sharp \in \mathcal{V}$

Step	Comment	Computational complexity
$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \mathrm{SVD}(\mathbf{\Sigma}_{\mathrm{XY}}, r)$	$\{\, \mathbf{B} \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \,\}$	SVD calculation in O(mnr)
for $i = 1, \ldots, T$ **do**		
$\mathbf{c}_i \leftarrow \mathtt{randn(r)}$	$\{\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{r\times r})\}$	Randomly select a direction in the low-rank subspace
$\mathbf{c}_i \leftarrow \mathbf{c}_i / \Vert\mathbf{c}_i\Vert_2$		
$\mathbf{a}_i \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{c}_i$	$\{\mathbf{a}_i \in \mathbb{R}^m\}$	Matrix-vector multiplication
$\mathbf{u}_i \leftarrow \arg\max_{\mathbf{u}\in\mathcal{U}} \mathbf{a}_i^\top \mathbf{u}$	$\{\mathrm{P}_{\mathcal{U}}(\cdot)\}$	O(m) or O(m log m) - sorting operation
$\mathbf{b}_i \leftarrow \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \mathbf{u}_i$	$\{\mathbf{b}_i \in \mathbb{R}^n\}$	Matrix-vector multiplication
$\mathbf{v}_i \leftarrow \arg\max_{\mathbf{v}\in\mathcal{V}} \mathbf{b}_i^\top \mathbf{v}$	$\{\mathrm{P}_{\mathcal{V}}(\cdot)\}$	O(n) or O(n log n) - sorting operation
$\mathrm{obj}_i \leftarrow \mathbf{b}_i^\top \mathbf{v}_i$		
end for		
$i_0 \leftarrow \arg\max_{i\in[T]} \mathrm{obj}_i$		Picks the best pair of (u, v) - **PARALLELIZABLE**
$(\mathbf{u}_\sharp, \mathbf{v}_\sharp) \leftarrow (\mathbf{u}_{i_0}, \mathbf{v}_{i_0})$		

Computational complexity

Global guarantee with additive error

Theorem 1. *For any real $m \times n$ matrix $\mathbf{\Sigma}_{\mathrm{XY}}$, $\epsilon \in (0, 1)$, and $r \leq \max\{m, n\}$, Algorithm 1 with input $\mathbf{\Sigma}_{\mathrm{XY}}$, r, and $\mathrm{T} = \widetilde{O}\big(2^{r \cdot \log_2(2/\epsilon)}\big)$ outputs $\mathbf{u}_\sharp \in \mathcal{U}$ and $\mathbf{v}_\sharp \in \mathcal{V}$ such that*

$$\boxed{\mathbf{u}_\sharp^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\sharp \geq \mathbf{u}_\star^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\star - \epsilon \cdot \sigma_1(\mathbf{\Sigma}_{\mathrm{XY}}) - 2\sigma_{r+1}(\mathbf{\Sigma}_{\mathrm{XY}}),}$$

in time $\mathrm{T}_{\mathrm{SVD}}(r) + O\big(\mathrm{T} \cdot (\mathrm{T}_{\mathcal{U}} + \mathrm{T}_{\mathcal{V}} + r \cdot \max\{m, n\})\big)$.

Theorem 2. *If $\mathcal{V} = \{\mathbf{v} : \Vert\mathbf{v}\Vert_2 = 1\}$, i.e., if no constraint is imposed on variable $\mathbf{v}$ besides unit length, then Algorithm 1 under the same configuration as that in Theorem 1 outputs $\mathbf{u}_\sharp \in \mathcal{U}$ and $\mathbf{v}_\sharp \in \mathcal{V}$ such that*

$$\boxed{\mathbf{u}_\sharp^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\sharp \geq (1 - \epsilon) \cdot \mathbf{u}_\star^\top \mathbf{\Sigma}_{\mathrm{XY}} \mathbf{v}_\star - 2 \cdot \sigma_{r+1}(\mathbf{\Sigma}_{\mathrm{XY}}).}$$

Global guarantee with multiplicative approximation

Conclusions

- **Novel** algorithm for a special CCA case: diagonal CCA

`https://github.com/megasthenis/spancca`

Conclusions

- **Novel** algorithm for a special CCA case: diagonal CCA
- **Desirable features:**
  - Simple to implement
  - Parallelizable
  - Precise control on the sparsity of the extracted components
  - Theoretical global approximation guarantees

https://github.com/megasthenis/spancca

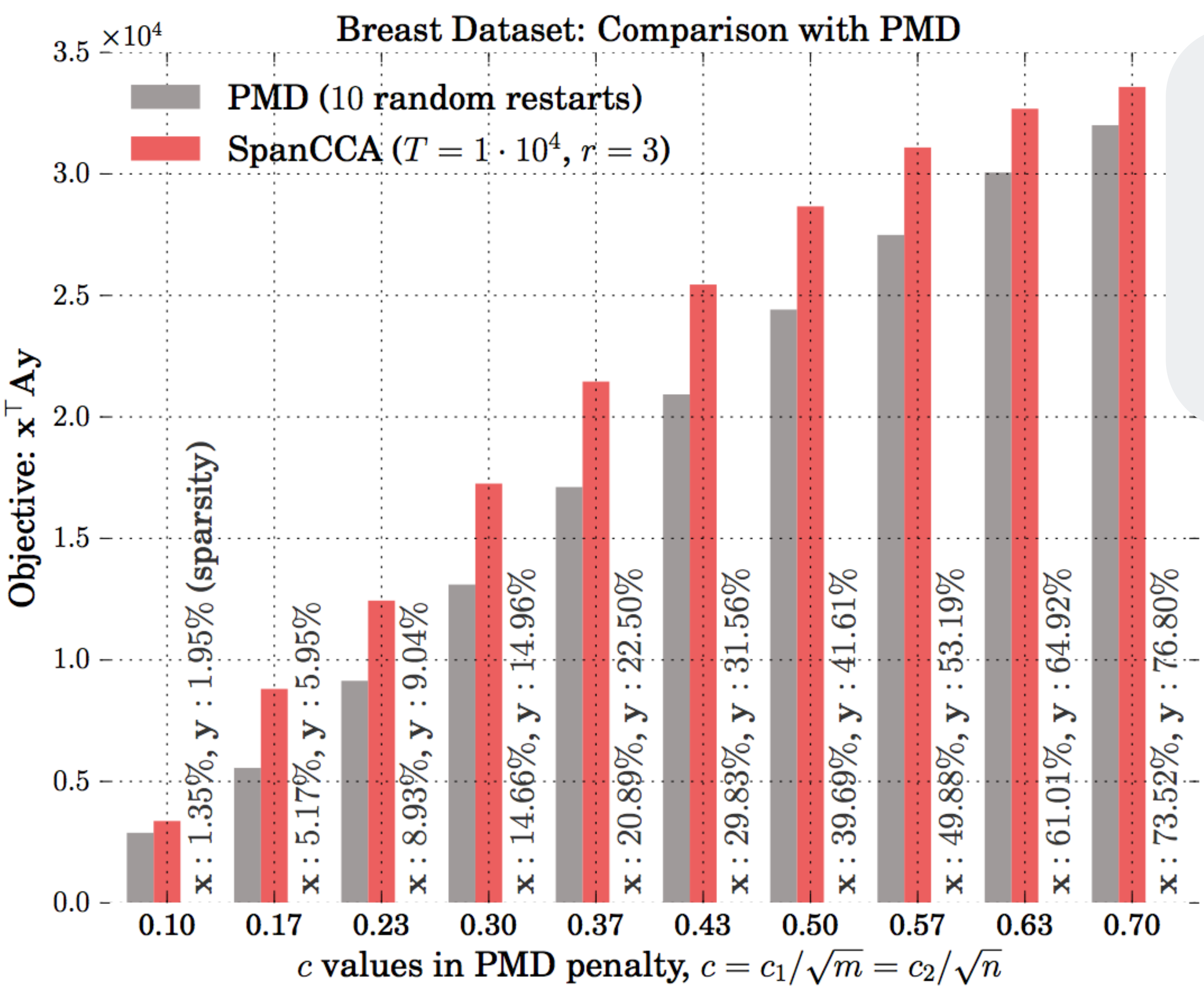
Conclusions

- **Novel** algorithm for a special CCA case: diagonal CCA
- **Desirable features:**
  - Simple to implement
  - Parallelizable
  - Precise control on the sparsity of the extracted components
  - Theoretical global approximation guarantees
- **Future work**
  - Handle original sparse CCA problem
  - Easy to extend to more-structured constraints: one needs to find the right projection operator.
  - Statistical guarantees?
  - More applications…

https://github.com/megasthenis/spancca

Applications

- Gene expression analysis
  - Joint analysis of DNA variants and gene expression measurements
  - **Goal:** identify correlations between expression levels of gene subsets and variation in related genes



Penalized matrix decomposition (PMD)

$$\max_{\substack{\|\mathbf{u}\|_2=1,\ \|\mathbf{u}\|_1\leq c_x,\\ \|\mathbf{v}\|_2=1,\ \|\mathbf{v}\|_1\leq c_y}} \mathbf{u}^\top \mathbf{\Sigma}_{XY}\mathbf{v}$$

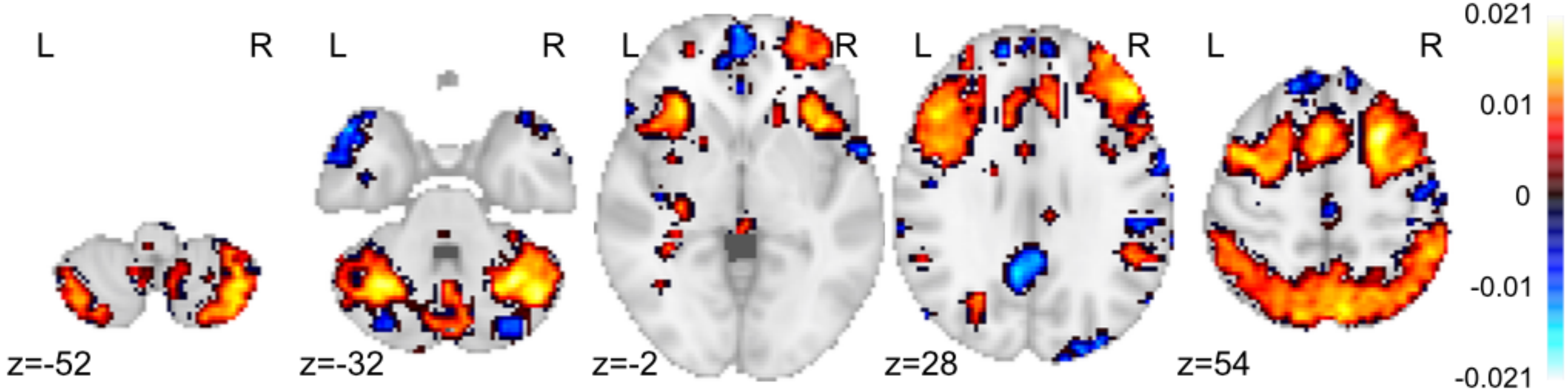
(Witten et al., 2009; Parkhomenko et al. 2009)

- Achieve better objective value
- Precise control over the sparsity levels

	Avg Exec. Time	Configuration
PMD	$\sim$ 44 seconds	10 rand. restarts
SpanCCA	$\sim$ 24 seconds	$T = 10^4, r = 3.$

Applications

- Cognitive neuroscience
  - Joint analysis of brain activation and behavioral observations
  - **Goal:** reveal associations between seemingly different tasks



Behavioral Factor & Weight	
PMAT24_A_CR	0.487
PicVocab_AgeAdj	0.448
ReadEng_AgeAdj	0.440
PicVocab_Unadj	0.433
ReadEng_Unadj	0.426

Bonus: parallelization