Exact and Approximation Algorithms for Sparse PCA (SPCA)

Presenter: Yongchun Li (liyc@vt.edu, Ph.D. Student), Coauthor: Weijun Xie (wxie@vt.edu, Assistant Professor)
Department of Industrial and Systems Engineering, Virginia Tech

The 22nd Conference on Integer Programming and Combinatorial Optimization, 2021, Atlanta, Georgia.

VIRGINIA TECH

Introduction

Principal Component Analysis (PCA): A dimension reduction method in machine learning

$$\text{(PCA)} \quad w^* := \max_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1 \right\},$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the sample covariance matrix of n features.

Drawback of PCA: lack of interpretability when n is large (many features)! $\Longrightarrow$ **Sparse PCA**

Advantages of Sparse PCA (SPCA):

- Improve interpretability
- Dimension reduction
- Feature selection

Applications

Example 1. SPCA reduces the data complexity and determines the important factors for drug abuse analysis

Figure: Drug abuse

Example 2. It is shown that the selected genes by SPCA is stable and interpretable

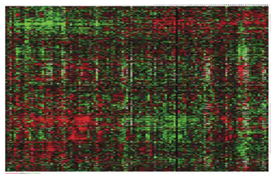


Figure: Gene expression

- **Other applications:** Image processing; Biological metabolomics; Anomaly localization ...

SPCA Model

Model Formulation

The SPCA is formulated by

$$\text{(SPCA)} \quad w^* := \max_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, ||\boldsymbol{x}||_0 = k \right\}.$$

- $\mathbf{A}$ is an $n \times n$ positive semi-definite matrix of rank d
- Zero norm constraint $||\boldsymbol{x}||_0 = k$ forces us to select only k important features
- SPCA is NP-hard (Magdon-Ismail, 2017)

"NP-hardness motivates us to develop efficient exact and approximation algorithms for SPCA"

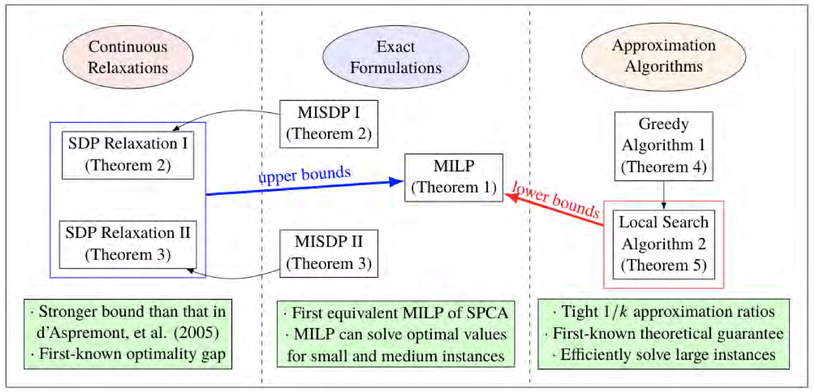
Existing Work of SPCA

- Exact algorithm: Branch and bound (Moghaddam et al., 2006, Berk and Bertsimas, 2019)
- Approximation algorithm: greedy (He et al., 2011) , local search (Guerrero et al., 2014), and truncation (Chan et al., 2016)

Research Gap

- Existing weak formulations
- No theoretical guarantees for greedy and local search algorithms

Summary of Contributions



Selected References

1. Magdon-Ismail, M. (2017). NP-hardness and inapproximability of sparse PCA. Information Processing Letters, 126, 35-38.
2. d'Aspremont, A., Ghaoui, L., Jordan, M., Lanckriet, G. (2005). A direct formulation for sparse PCA using semidefinite programming. Advances in neural information processing systems, 17, 41-48.
3. Chan, S. O., Papailliopoulos, D., Rubinstein, A. (2016, June). On the approximability of sparse pca. In Conference on Learning Theory (pp. 623-646).
4. Dey, S. S., Mazumder, R., Wang, G. (2018). A convex integer programming approach for optimal sparse PCA. arXiv preprint arXiv:1810.09062.

Exact Algorithms for Solving SPCA

Mixed Integer Linear Program (MILP)

Theorem 1. For any threshold $\epsilon > 0$, the following MILP is $O(\epsilon)$-approximate to SPCA, i.e., $\epsilon \le \hat{w}(\epsilon) - w^* \le \epsilon\sqrt{d}$.

$$\begin{aligned}
\text{(MILP)} \quad \hat{w}(\epsilon) := \max_{w,\boldsymbol{z}\in Z,\boldsymbol{y},\boldsymbol{\alpha},\boldsymbol{x}_\cdot,\boldsymbol{\delta}_\cdot,\boldsymbol{\mu}_\cdot,\boldsymbol{\sigma}} \quad & w \\
\text{s.t.} \quad & \boldsymbol{x} = \boldsymbol{\delta}_{i1} + \boldsymbol{\delta}_{i2}, ||\boldsymbol{\delta}_{i1}||_\infty \le z_i, ||\boldsymbol{\delta}_{i2}||_\infty \le 1 - z_i, \forall i \in [n], \boldsymbol{x} = \sum_{j\in[d]} \boldsymbol{\sigma}_j, ||\boldsymbol{\sigma}_j||_\infty \le y_j, \sigma_{jj} = y_j, \forall j \in [d], \\
& \sum_{j\in[d]} y_j = 1, \boldsymbol{x} = \boldsymbol{\mu}_{\ell 1} + \boldsymbol{\mu}_{\ell 2}, ||\boldsymbol{\mu}_{\ell 1}||_\infty \le \alpha_\ell, ||\boldsymbol{\mu}_{\ell 2}||_\infty \le 1 - \alpha_\ell, \forall \ell \in [m], \boldsymbol{\alpha} \in \{0,1\}^m, \boldsymbol{y} \in \{0,1\}^d, \\
& \left|\left| \sum_{i\in[n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1} - w_U \boldsymbol{x} + (w_U - w_L) \sum_{\ell\in[m]} 2^{-\ell} \boldsymbol{\mu}_{\ell 1} \right|\right|_\infty \le \epsilon, w = w_U - (w_U - w_L)\left(\sum_{i\in[m]} 2^{-i}\alpha_i \right),
\end{aligned}$$

where w_L, w_U separately denote the lower and upper bounds of SPCA.

" First equivalent MILP of SPCA (if $\epsilon \to 0$) "

- **Proof Idea:** Cholesky factorization, modeling disjunction, eigenvalue equation, and binary expression
- MILP can be efficiently solved by solvers like Gurobi or CPLEX

Mixed Integer Semidefinite Program (MISDP) I

Theorem 2. The SPCA admits an equivalent MISDP formulation

$$\text{(MISDP I)} \quad w^* := \max_{\substack{\boldsymbol{z}\in Z, \\ \boldsymbol{X},\boldsymbol{W}_1,\cdots,\boldsymbol{W}_n\in\mathcal{S}_+^d}} \left\{ \sum_{i\in[n]} \boldsymbol{c}_i^\top \boldsymbol{W}_i \boldsymbol{c}_i : \text{tr}(\boldsymbol{X}) = 1, \boldsymbol{X} \succeq \boldsymbol{W}_i, \text{tr}(\boldsymbol{W}_i) = z_i, \forall i \in [n] \right\},$$

and the continuous relaxation value $\overline{w}_1$ of MISDP I achieves a $\min\{k, n/k\}$ optimality gap of SPCA, i.e.,

$$w^* \le \overline{w}_1 \le \min\{k, n/k\} w^*.$$

- **Proof Idea:** Modeling disjunction, Cholesky factorization, trace inequality
- Using Benders decomposition, we obtain closed-form cuts for solving SPCA
- An equivalent saddle point problem for the continuous relaxation $\overline{w}_1$ ($O(1/T)$-subgradient method)

Mixed Integer Semidefinite Program (MISDP) II

Theorem 3. The SPCA can reduce to following MISDP formulation:

$$\text{(MISDP II)} \quad w^* := \max_{\boldsymbol{z}\in Z,\boldsymbol{X}\in\mathcal{S}_+^n} \left\{ \text{tr}(\boldsymbol{A}\boldsymbol{X}) : \text{tr}(\boldsymbol{X}) = 1, \sum_{j\in[n]} X_{ij}^2 \le X_{ii} z_i, \left(\sum_{j\in[n]} |X_{ij}| \right)^2 \le k X_{ii} z_i, \forall i \in [n] \right\},$$

and the continuous relaxation value $\overline{w}_2$ of MISDP I achieves a $\min\{k, n/k\}$ optimality gap of SPCA, i.e.,

$$w^* \le \overline{w}_2 \le \min\{k, n/k\} w^*.$$

- **Proof Idea:** SDP reformulation of the largest eigenvalue
- $\overline{w}_2$ is stronger than the continuous relaxation proposed by dAspremont et al. (2005).

Approximation Algorithms for Solving SPCA

Greedy Algorithm

Algorithm 1: Greedy Algorithm of SPCA

Input: Let $\hat{S}_G := \emptyset$ denote the chosen set
for $\ell = 1, \cdots, k$ **do**
  Compute $j^* \in \text{argmax}_{j\in[n]\setminus\hat{S}_G}$, add j^* to the set $\hat{S}_G$
Output: Set $\hat{S}_G$

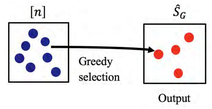


Figure: An illustration of greedy algorithm.

Theorem 4. The greedy Algorithm 1 yields a k^{-1}-approximation ratio for SPCA, i.e., the output $\hat{S}_G$ satisfies $\lambda_{\max}(\sum_{i\in\hat{S}_G} \boldsymbol{c}_i\boldsymbol{c}_i^\top) \ge k^{-1} w^*$, and the ratio is tight.

Local Search Algorithm

Algorithm 2: Local Search Algorithm

Input: Initialize a size k subset $\hat{S}_L \subseteq [n]$
for each pair $(i,j) \in \hat{S}_L \times ([n] \setminus \hat{S}_L)$ **do**
  if $\lambda_{\max}(\sum_{\ell\in\hat{S}_L\cup\{j\}\setminus\{i\}} \boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top) > \lambda_{\max}(\sum_{\ell\in\hat{S}_L} \boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top)$ **do**
    Update $\hat{S}_L := \hat{S}_L \cup \{j\} \setminus \{i\}$
Output: Set $\hat{S}_L$

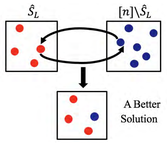


Figure: An illustration of local search algorithm.

Theorem 5. The local search Algorithm 2 yields a k^{-1}-approximation ratio for SPCA, i.e., the output $\hat{S}_G$ satisfies $\lambda_{\max}(\sum_{i\in\hat{S}_G} \boldsymbol{c}_i\boldsymbol{c}_i^\top) \ge k^{-1} w^*$, and the ratio is tight.

" First-known and tight approximation ratios of greedy and local search algorithms"

- **Proof Idea of Theorem 4 & 5:** convexity of largest eigenvalue function; tightness by constructing a worst-case example
- **Efficient Implementation:** Power iteration speeds up both algorithms significantly
- Greedy and local search algorithm can efficiently solve large-scale instances

Computational Results of SPCA

Case 1. A Benchmark Instance of n = 13: Exact Algorithms and Continuous Relaxations

n=13	SPCA	MISDP I		MISDP II		MILP		Gurobi		Benchmark		SDP Relaxation I			SDP Relaxation II		
k	w^*	w^*	time(s)	w^*	time(s)	$\hat{w}(\epsilon)$	time(s)	w^*	time(s)	$\overline{w}_3$	gap(%)	$\overline{w}_1$	gap(%)	time(s)	$\overline{w}_2$	gap(%)	time(s)
4	2.9375	2.9375	1	2.9375	2	2.9375	1	2.9375	1	3.0172	2.71	3.1065	5.75	0.51	2.9495	0.41	0.13
5	3.4062	3.4062	1	3.4062	2	3.4062	1	3.4062	1	3.4581	1.52	3.4868	2.37	0.55	3.4124	0.18	0.18
6	3.7710	3.7710	1	3.7710	2	3.7710	2	3.7710	1	3.8137	1.13	3.7859	0.39	0.52	3.7767	0.15	0.15
7	3.9962	3.9962	1	3.9962	1	3.9962	1	3.9962	3	4.0316	0.89	3.9962	0.00	0.43	3.9962	0.00	0.15
8	4.0686	4.0686	1	4.0686	2	4.0686	2	4.0686	12	4.1448	1.87	4.0805	0.29	0.29	4.0793	0.26	0.17
9	4.1386	4.1386	1	4.1386	2	4.1386	1	4.1387	30	4.2063	1.64	4.1386	0.00	0.00	4.1398	0.03	0.15
10	4.1726	4.1726	1	4.1726	1	4.1726	1	4.1441	83	4.2186	1.10	4.1763	0.09	0.09	4.1778	0.12	0.16

- By setting $\epsilon = 1e-4$, MILP beats the latest Gurobi, more stable and efficient
- The SDP continuous relaxation II ($\overline{w}_2$) is tighter than the existing bound from dAspremont et al. (2005)

Case 2. A Benchmark Instance of n = 13: Approximation Algorithms

n=13	SPCA	Truncation algorithm			Greedy Algorithm 1			Local Search Algorithm 2		
k	w^*	LB	gap(%)	time(s)	LB	gap(%)	time(s)	LB	gap(%)	time(s)
4	2.9375	2.8913	1.57	1e-3	2.9375	0.00	1e-3	2.9375	0.00	1e-2
5	3.4062	3.3951	0.32	1e-3	3.4062	0.00	1e-3	3.4062	0.00	1e-2
6	3.7710	3.7576	0.36	1e-3	3.7710	0.00	1e-2	3.7710	0.00	1e-2
7	3.9962	3.9929	0.08	1e-3	3.9962	0.00	1e-2	3.9962	0.00	1e-2
8	4.0686	4.0648	0.09	1e-3	4.0686	0.00	1e-2	4.0686	0.00	1e-2
9	4.1386	4.1313	0.18	1e-3	4.1386	0.00	1e-2	4.1386	0.00	1e-2
10	4.1726	4.0094	3.91	1e-3	4.1726	0.00	1e-2	4.1726	0.00	1e-2

- The greedy and local search algorithms yield optimal values
- Both outperform the best-known approximation algorithm in Chan et al. (2016)

Case 3. Four Larges Scale Instances of n = 79, 118, 500, 2000: The datasets are from (Dey et al., 2018)

Dataset	Case		Local Search Algorithm 2		SDP Relaxation I		SDP Relaxation II		MILP (ϵ=1e-4)		
	n	k	LB	time(s)	UB	time(s)	UB	time(s)	VAL	MIPgap(%)	time(s)
Eisen-1	79	10	17.3355	1	17.9144	14	17.7571	126	17.3355	0.00	34
	79	20	17.7195	1	18.1309	13	18.0362	85	17.7195	0.00	125
Eisen-2	118	10	11.7182	1	13.8732	89	-	-	11.7182	18.39	3600
	118	20	19.3228	1	22.9268	90	-	-	19.3228	18.65	3600
Colon	500	10	2641.2289	1	2901.1105	342	-	-	2641.2289	9.84	3600
	500	20	4255.6941	3	4833.1900	344	-	-	4255.6941	13.57	3600
Reddit	2000	10	1521.3083	9	1867.9965	1198	-	-	-	-	-
	2000	20	1684.3943	59	2184.2436	1241	-	-	-	-	-

- For medium case, MILP can solve optimal values for SPCA
- The local search performs very well in large scale instance, less than one minute!

Case 4. SPCA for Drug Abuse

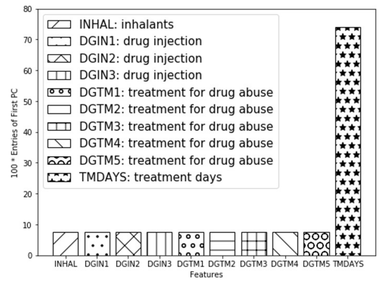


Figure: 10 features selected by local search Algorithm 2

- *Drugabuse* dataset: n= 2365 features (National survey, 2018)
- It is recognized that the selected 10 features are important for drug abuse analysis

Extensions

Rank-one Sparse SVD (R1-SSVD)

Non-symmetric matrix $\boldsymbol{A}$

$$w_{\text{SVD}}^* := \max_{\boldsymbol{u}\in\mathbb{R}^m,\boldsymbol{v}\in\mathbb{R}^n} \left\{ \boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v} : ||\boldsymbol{u}||_2 = ||\boldsymbol{v}||_2 = 1, \; ||\boldsymbol{u}||_0 = k_1, ||\boldsymbol{v}||_0 = k_2 \right\}.$$

Sparse Fair PCA (SFPCA)

Multiple matrices $\boldsymbol{A}_i$

$$w_F^* := \max_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \min_{i\in S} \boldsymbol{x}^\top \boldsymbol{A}_i \boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, ||\boldsymbol{x}||_0 \le k \right\}.$$

Problem	Exact MI Formulation	Optimality Gap
R1-SSVD	MISDP (I)	$\sqrt{mnk_1^{-1}k_2^{-1}}$
	MISDP (II)	$\min\{\sqrt{k_1k_2}, \sqrt{mnk_1^{-1}k_2^{-1}}\}$
	MILP	$[(k_1+k_2)(\sqrt{d}/2+1/2)-1]\sqrt{mnk_1^{-1}k_2^{-1}}$
SFPCA	MISDP	–

Problem	Approximation Algorithm	Approximation Ratio
R1-SSVD	Truncation algorithm	$\max\{\sqrt{k_1^{-1}}, \sqrt{k_2^{-1}}, \sqrt{k_1k_2m^{-1}n^{-1}}\}$
	Greedy algorithm	$\sqrt{k_1^{-1}k_2^{-1}}$
	Local Search algorithm	$\sqrt{k_1^{-1}k_2^{-1}}$

Figure: Theoretical results of R1-SSVD and SFPCA