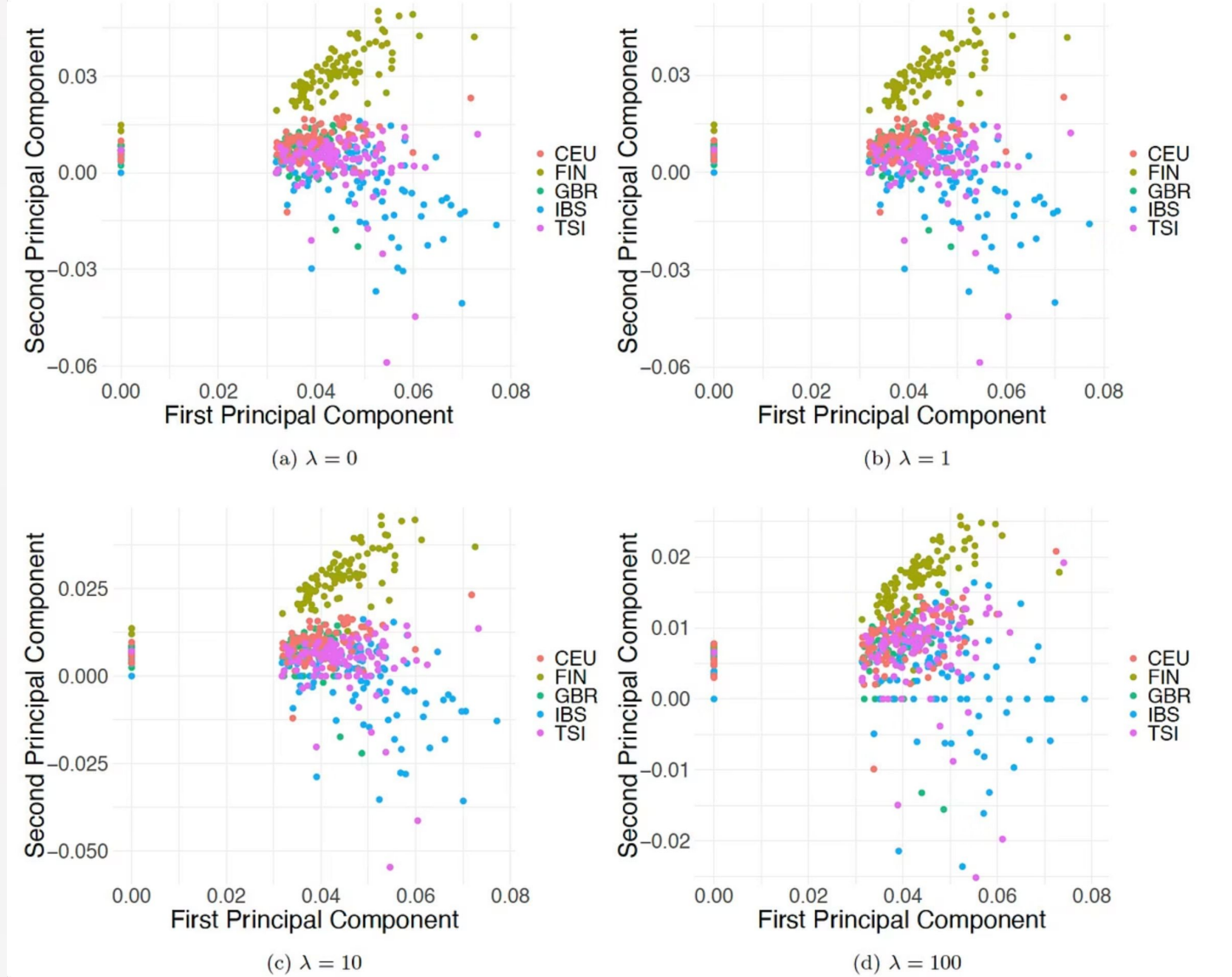


Penalized Principal Component Analysis Using Smoothing

Rebecca Hurwitz, PhD Student

Department of Biomedical Data Science

Stanford University



Agenda

1. Introduction to PCA
2. Limitations of PCA
3. Penalized Eigenvalue Problem (PEP)
4. Smoothed PEP
5. Higher-Order Eigenvectors
6. Experimental Studies & Results
7. Conclusions & Future Directions
8. Acknowledgments

Introduction to PCA

- Principal Component Analysis is a fundamental technique for dimensionality reduction
- It transforms large datasets into lower-dimensional representations while aiming to preserve a maximal amount of information
- Statistical genomics: eigenvectors from PCA are routinely used to adjust and correct for population stratification

Limitations of PCA

Classic PCA eigenvectors are typically dense, which can make them difficult to interpret

In high-dimensional settings (many features, few samples), sparsity-inducing methods are needed to "improve estimation accuracy" and provide "better interpretable eigenvectors through variable selection"

The Penalized Eigenvalue Problem (PEP)

- To address the issue of dense eigenvectors, the PEP reformulates the computation of the first eigenvector as an optimization problem

$$v = \arg \max_{v \in \mathbb{R}^p} v^\top Q v \quad \text{subject to} \quad v^\top C v \leq 1,$$

- Enforces sparsity via a LASSO-type L1 penalty

$$v_\lambda = \arg \max_{v: \|v\|=1} \left[v^\top Q v - \lambda \|v\|_1 \right]$$

Key challenge: the L1 penalty is non-differentiable

Proposed Solution: Smoothed PEP

The primary contribution of this paper is the application of smoothing to the original LASSO-type L1 penalty within the PEP framework

- How smoothing works: a smooth surrogate is used instead of the non-differentiable L1 norm
- Smoothed absolute value function, derived from an entropy-prox function:

$$f_e^\mu(z) = \mu \log \left(\frac{1}{2} e^{-z/\mu} + \frac{1}{2} e^{z/\mu} \right),$$

- Proposed smoothed penalized eigenvalue problem (PEP)::

$$v_\lambda^\mu = \arg \max_{v: \|v\|=1} \left[v^\top Q v - \lambda \sum_{i=1}^p f_e^\mu(v_i) \right].$$

Computing Higher-Order Eigenvectors & Enforcing Sparsity

- Higher-order eigenvectors are calculated using Singular Value Decomposition (SVD) and deflation, where the leading eigencomponent is subtracted to reveal subsequent eigenvectors
- We then use an iterative solving approach, which enhances numerical accuracy by starting with a large smoothing parameter and gradually decreasing it
- Sparsity is then enforced through thresholding (smoothed PEP solutions are not inherently sparse)

Experimental Studies

1. Population Stratification on 1000 Genomes Project Data

2. Polygenic Risk Score Application (SARS-CoV-2 Mortality)

3. Clustering in Iris Benchmark Dataset

4. Comparison with State-of-the-Art Sparse PCA Algorithms

Experimental Studies

1. Population Stratification on 1000 Genomes Project Data

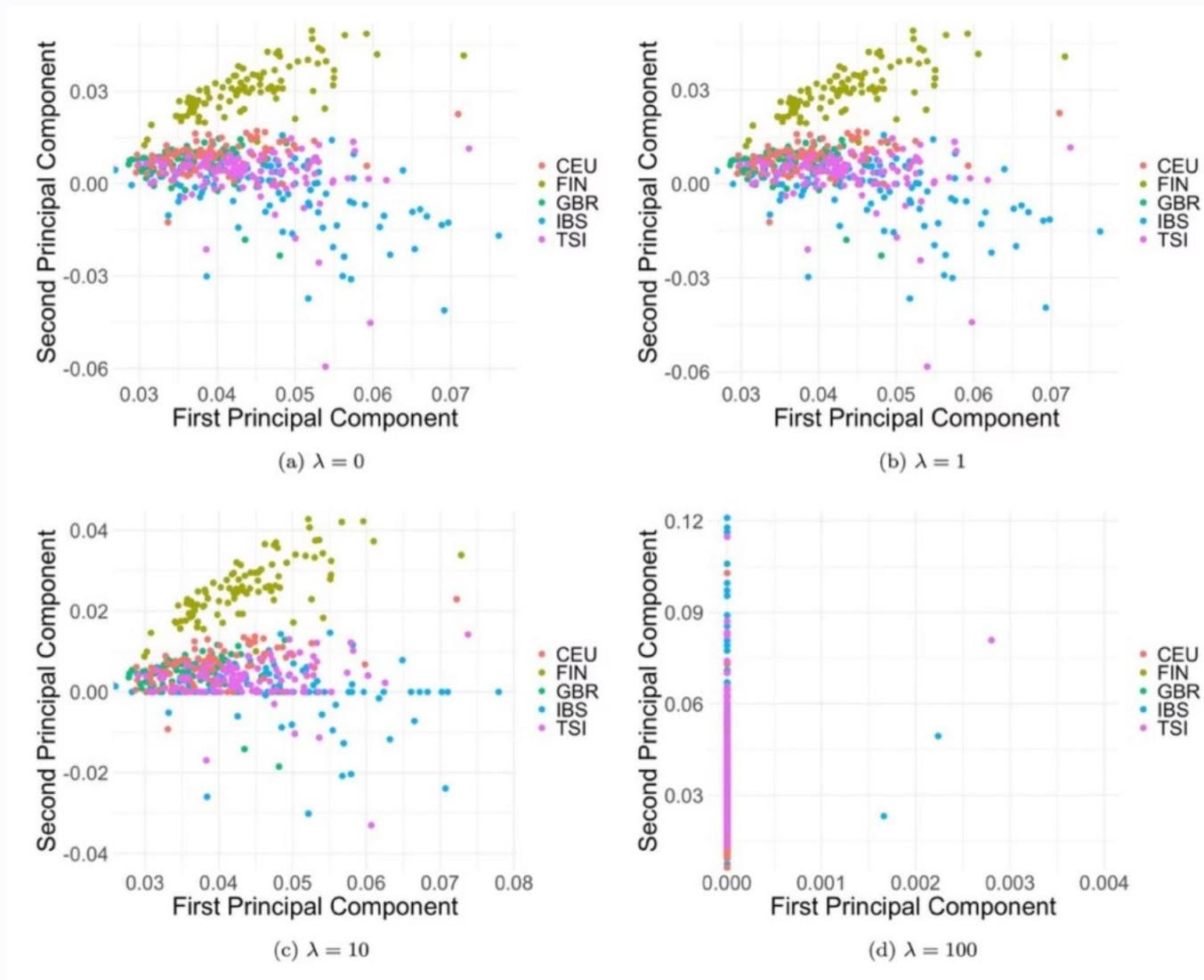
2. Polygenic Risk Score Application (SARS-CoV-2 Mortality)

3. Clustering in Iris Benchmark Dataset

4. Comparison with State-of-the-Art Sparse PCA Algorithms

Results: Population Stratification

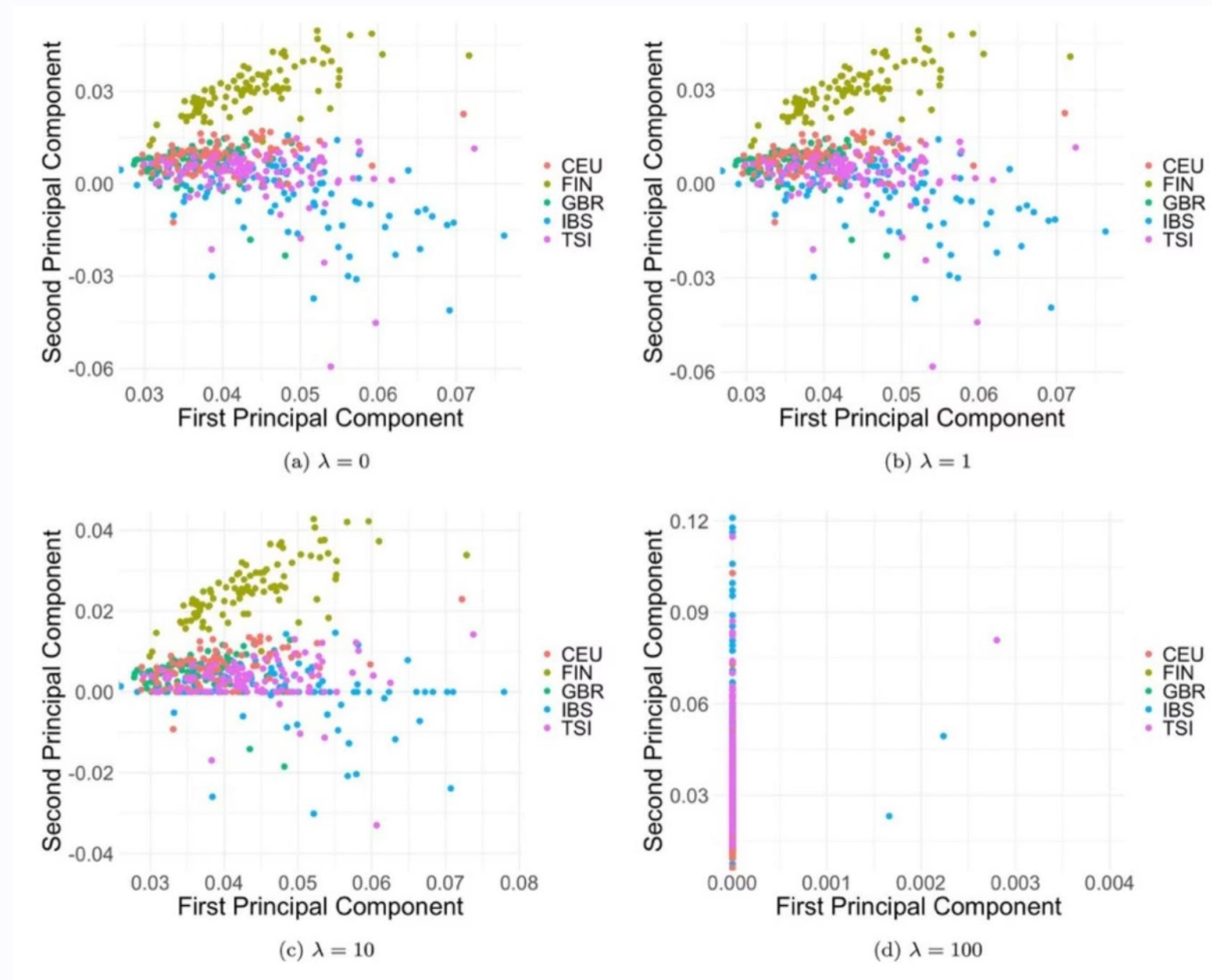
Unsmoothed PEP



Shows good stratification for $\lambda=0$, 1, and 10, but discernibility decreases significantly at $\lambda=100$.

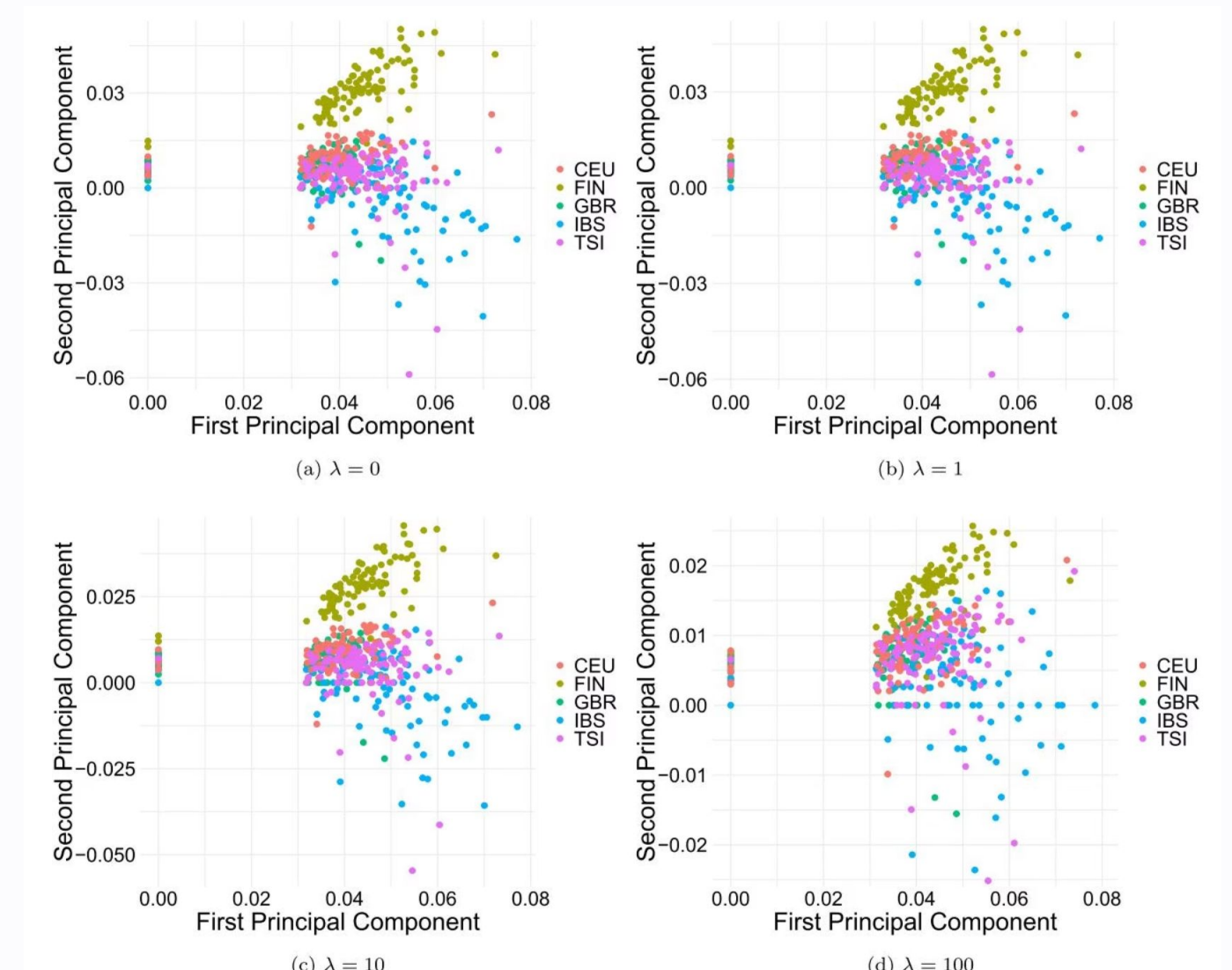
Results: Population Stratification

Unsmoothed PEP



Shows good stratification for $\lambda=0, 1$, and 10 , but discernibility decreases significantly at $\lambda=100$.

Smoothed PEP



Maintains good stratification across all penalty values, with clearer clustering even at $\lambda=100$, demonstrating enhanced performance.

Results: Population Stratification

Within sum of squares

Model	λ			
	0	1	10	100
Unsmoothed PEP	1.6654	1.6702	1.7244	1.8839
Smoothed PEP	1.6621	1.6626	1.6665	1.6910

Between sum of squares

Model	λ			
	0	1	10	100
Unsmoothed PEP	0.1278	0.1231	0.0912	0.0770
Smoothed PEP	0.1334	0.1313	0.1149	0.0531

Average silhouette score

Model	λ			
	0	1	10	100
Unsmoothed PEP	−0.0459	−0.0475	−0.0610	−0.2066
Smoothed PEP	−0.0175	−0.0183	−0.0257	−0.0828

Experimental Studies!

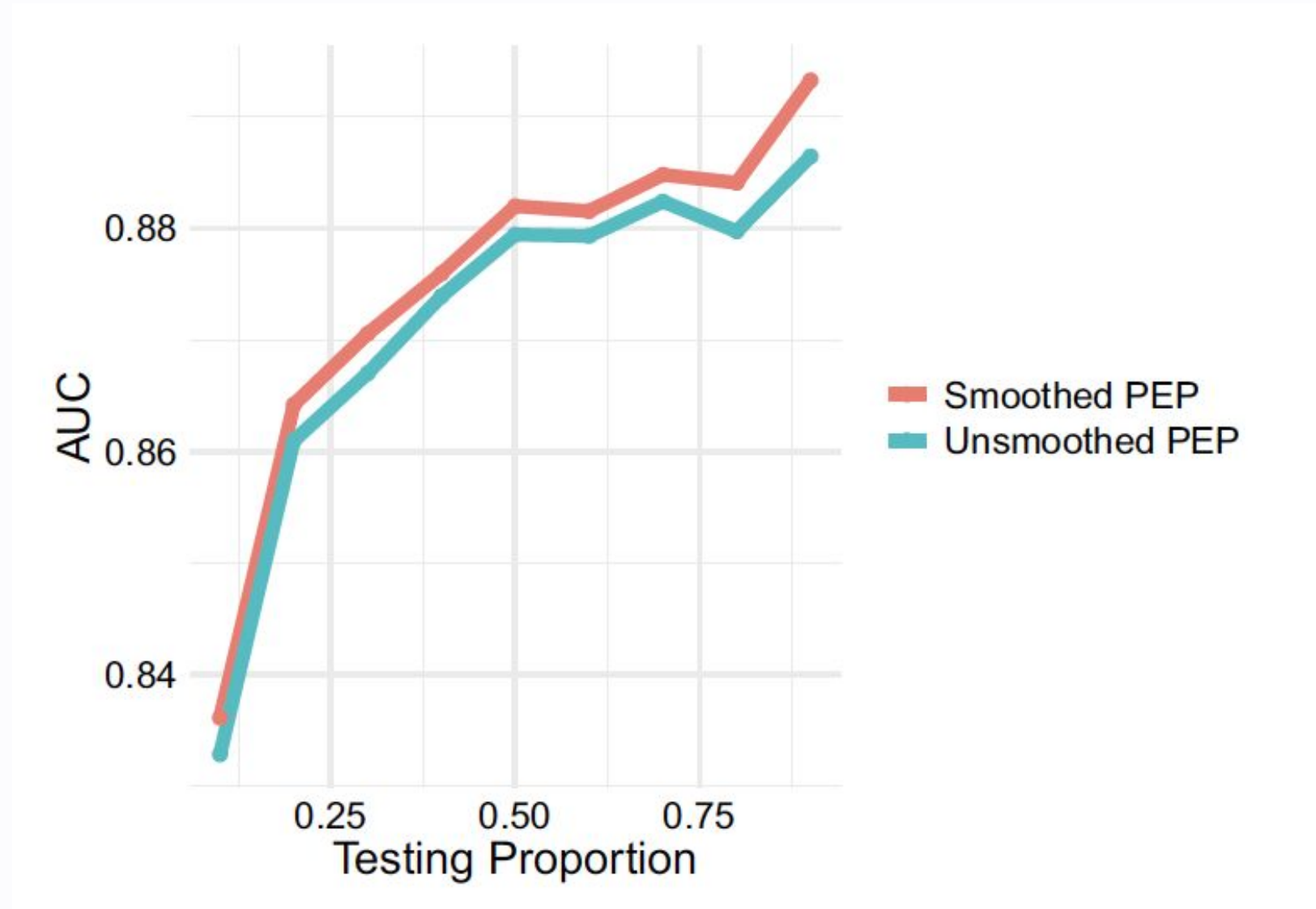
1. Population Stratification on 1000 Genomes Project Data

2. Polygenic Risk Score Application (SARS-CoV-2 Mortality)

3. Clustering in Iris Benchmark Dataset

4. Comparison with State-of-the-Art Sparse PCA Algorithms

Results: Polygenic Risk Scores for Covid Mortality



Experimental Studies

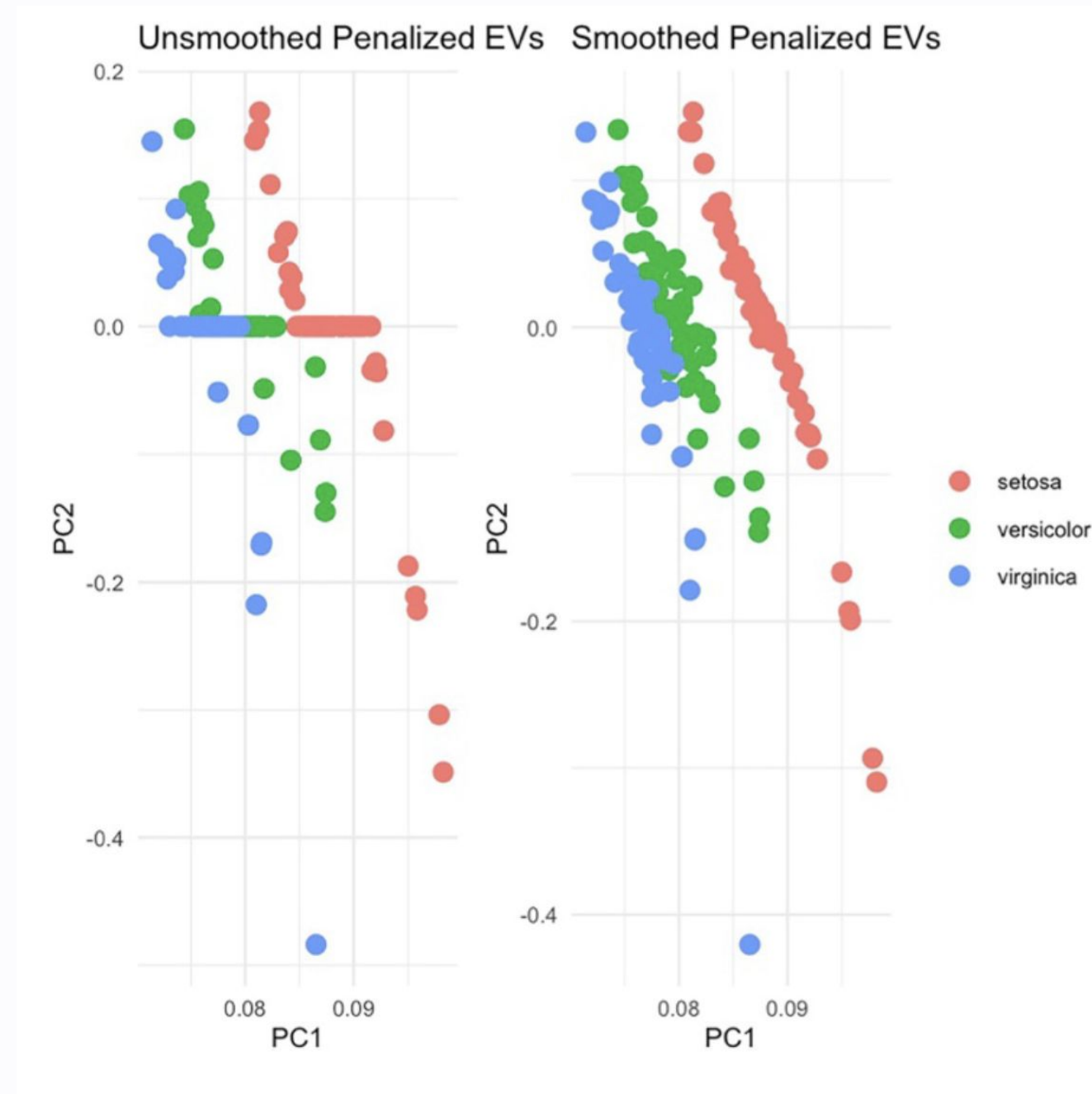
1. Population Stratification on 1000 Genomes Project Data

2. Polygenic Risk Score Application (SARS-CoV-2 Mortality)

3. Clustering in Iris Benchmark Dataset

4. Comparison with State-of-the-Art Sparse PCA Algorithms

Results: Iris Benchmark Dataset Clustering



Experimental Studies

1. Population Stratification on 1000 Genomes Project Data

2. Polygenic Risk Score Application (SARS-CoV-2 Mortality)

3. Clustering in Iris Benchmark Dataset

4. Comparison with State-of-the-Art Sparse PCA Algorithms

Results: Comparison with Sparse PCA Algorithms

Table 4 Cosine similarity between the computed leading eigenvector from each sparse PCA algorithm and the true (planted) leading eigenvector as a function of the matrix dimension n and the sparsity level ρ (proportion of zeros)

n	Sparsity ρ	Cosine similarity							
		Journee6	Aspremont2	Jung2	Gaynanova1	Song4	Zhang1	Shen1	Smoothed PEP
10	0.1	0.9037999	0.5275607	0.11229495	0.9914799	0.92853305	0.32581428	0.9759526	0.9914892
	0.2	0.9317490	0.8869066	0.58735246	0.9965534	0.91725763	0.63849359	0.9970452	0.9964332
	0.3	0.9089373	0.3214823	0.19747244	0.9973516	0.90620679	0.13018827	0.9894365	0.9971395
	0.4	0.8738096	0.8606155	0.04051950	0.9952860	0.73089078	0.19253621	0.9916363	0.9949463
	0.5	0.8927113	0.9034062	0.08860906	0.9945179	0.91792699	0.27524794	0.9971003	0.9942426
20	0.1	0.9039006	0.4619756	0.14390288	0.9965611	0.88988985	0.24930905	0.9758550	0.9962799
	0.2	0.6297668	0.4087665	0.17633557	0.9965252	0.90907549	0.09943160	0.9712495	0.9963236
	0.3	0.7863718	0.6305140	0.00825861	0.9937772	0.76705991	0.06413241	0.9655523	0.9934047
	0.4	0.7391398	0.8164070	0.23050779	0.9954877	0.90933260	0.27164501	0.9815923	0.9952056
	0.5	0.9278953	0.6929346	0.15936494	0.9964539	0.52389778	0.13210300	0.9731384	0.9958513
50	0.1	0.8838522	0.1805368	0.12775405	0.9921327	0.84785138	0.10742227	0.8802893	0.9920830
	0.2	0.8770866	0.2299920	0.22687887	0.9935021	0.84988699	0.15669446	0.8508515	0.9932049
	0.3	0.8785249	0.1048715	0.14177232	0.9943568	0.85364046	0.03533143	0.9335731	0.9936751
	0.4	0.2689499	0.5967501	0.11209041	0.9927075	0.06249241	0.16759339	0.9361840	0.9920095
	0.5	0.5923197	0.6376264	0.19823132	0.9946354	0.69934528	0.15969902	0.9290508	0.9936740
100	0.1	0.8904958	0.2116873	0.10061459	0.9933355	0.88229786	0.17661137	0.7553414	0.9934581
	0.2	0.1882329	0.2977159	0.11584467	0.9927417	0.88426056	0.06979850	0.8078601	0.9924776
	0.3	0.3219067	0.1805002	0.06364077	0.9943353	0.86676384	0.05520514	0.8453197	0.9936151
	0.4	0.1661451	0.4221019	0.02058195	0.9941975	0.81987146	0.08392663	0.8391396	0.9928401
	0.5	0.8969307	0.2601693	0.00060988	0.9943816	0.90599720	0.24571967	0.8685434	0.9930091

Table 6 False Positive Rate measuring the proportion of falsely identified nonzero elements for each sparse PCA method as a function of the matrix dimension n and the sparsity level ρ (proportion of zeros)

n	Sparsity ρ	Support recovery (False Positive Rate)							
		Journee6	Aspremont2	Jung2	Gaynanova1	Song4	Zhang1	Shen1	Smoothed PEP
10	0.1	0.000	0.000	0.000	0.000	1.000	0.000	1.000	0.000
	0.2	0.000	0.000	0.000	0.000	1.000	0.000	1.000	0.000
	0.3	0.000	0.000	0.000	0.000	1.000	0.000	1.000	0.000
	0.4	0.250	0.250	0.000	0.000	1.000	0.000	1.000	0.000
	0.5	0.000	0.400	0.000	0.200	1.000	0.000	1.000	0.000
20	0.1	0.000	0.000	0.000	0.000	1.000	0.000	1.000	0.000
	0.2	0.250	0.000	0.000	0.250	1.000	0.000	1.000	0.000
	0.3	0.167	0.000	0.000	0.000	0.833	0.000	1.000	0.000
	0.4	0.125	0.000	0.000	0.125	1.000	0.000	1.000	0.000
	0.5	0.000	0.000	0.000	0.200	0.200	0.000	1.000	0.000
50	0.1	0.000	0.000	0.000	0.000	1.000	0.000	1.000	0.000
	0.2	0.000	0.000	0.000	0.400	0.900	0.000	1.000	0.000
	0.3	0.000	0.000	0.000	0.267	1.000	0.000	1.000	0.000
	0.4	0.050	0.000	0.000	0.300	0.650	0.000	1.000	0.000
	0.5	0.040	0.000	0.000	0.160	0.760	0.000	1.000	0.000
100	0.1	0.000	0.000	0.000	0.500	1.000	0.000	1.000	0.000
	0.2	0.000	0.000	0.000	0.250	0.950	0.000	1.000	0.000
	0.3	0.033	0.000	0.000	0.367	1.000	0.000	1.000	0.000
	0.4	0.025	0.000	0.000	0.200	0.850	0.000	1.000	0.000
	0.5	0.000	0.000	0.000	0.300	1.000	0.000	1.000	0.000

Table 5 True Positive Rate measuring the proportion of correctly identified nonzero elements in the estimated eigenvector for each sparse PCA method as a function of the matrix dimension n and the sparsity level ρ (proportion of zeros)

n	Sparsity ρ	Support recovery (True Positive Rate)							
		Journee6	Aspremont2	Jung2	Gaynanova1	Song4	Zhang1	Shen1	Smoothed PEP
10	0.1	0.889	1.000	1.000	1.000	0.889	1.000	0.778	1.000
	0.2	0.875	1.000	1.000	1.000	0.875	1.000	1.000	1.000
	0.3	0.857	1.000	1.000	1.000	0.857	1.000	0.714	1.000
	0.4	1.000	0.667	1.000	1.000	0.667	1.000	0.500	1.000
	0.5	0.800	0.600	1.000	1.000	1.000	1.000	1.000	1.000
20	0.1	0.944	1.000	1.000	0.944	0.889	1.000	0.722	1.000
	0.2	1.000	1.000	1.000	1.000	0.750	1.000	0.563	1.000
	0.3	1.000	0.857	1.000	1.000	0.786	1.000	0.500	1.000
	0.4	1.000	0.833	1.000	1.000	0.917	1.000	0.833	1.000
	0.5	0.900	0.800	1.000	1.000	0.900	1.000	0.500	1.000
50	0.1	0.978	1.000	1.000	0.978	0.778	1.000	0.289	1.000
	0.2	0.975	1.000	1.000	1.000	0.850	1.000	0.275	1.000
	0.3	0.971	1.000	1.000	0.971	0.800	1.000	0.400	1.000
	0.4	1.000	0.933	1.000	1.000	0.233	1.000	0.500	1.000
	0.5	1.000	0.960	1.000	1.000	0.720	1.000	0.280	1.000
100	0.1	0.989	1.000	1.000	0.967	0.767	1.000	0.167	1.000
	0.2	0.988	1.000	1.000	0.913	0.663	1.000	0.200	1.000
	0.3	1.000	1.000	1.000	1.000	0.700	1.000	0.229	1.000
	0.4	1.000	0.967	0.983	0.983	0.717	1.000	0.250	1.000
	0.5	0.980	0.960	1.000	1.000	0.760	1.000	0.340	1.000

Table 7 Elapsed compute times (in seconds) for each sparse PCA algorithm as a function of the matrix dimension n and the sparsity level ρ (proportion of zeros)

n	Sparsity ρ	Elapsed time (seconds)							
		Journee6	Aspremont2	Jung2	Gaynanova1	Song4	Zhang1	Shen1	Smoothed PEP
10	0.1	0.028	0.015	0.003	0.000	0.002	0.016	0.000	0.001
	0.2	0.028	0.047	0.003	0.000	0.002	0.015	0.000	0.001
	0.3	0.027	0.015	0.003	0.000	0.053	0.014	0.001	0.001
	0.4	0.027	0.233	0.003	0.001	0.003	0.015	0.000	0.002
	0.5	0.027	0.450	0.003	0.000	0.002	0.015	0.000	0.001
20	0.1	0.106	0.052	0.005	0.000	0.003	0.027	0.000	0.003
	0.2	0.108	0.051	0.005	0.000	0.003	0.023	0.000	0.005
	0.3	0.108	0.349	0.004	0.000	0.004	0.023	0.000	0.005
	0.4	0.109	0.841	0.004	0.000	0.004	0.022	0.000	0.003
	0.5	0.104	0.780	0.004	0.000	0.003	0.023	0.001	0.002
50	0.1	0.740	0.810	0.018	0.000	0.007	0.084	0.001	0.018
	0.2	0.753	0.720	0.017	0.000	0.007	0.093	0.001	0.015
	0.3	0.736	0.731	0.018	0.000	0.007	0.089	0.001	0.018
	0.4	0.754	1.901	0.018	0.001	0.008	0.096	0.001	0.016
	0.5	0.738	4.272	0.018	0.000	0.007	0.088	0.001	0.020
100	0.1	3.585	14.089	0.102	0.001	0.017	0.353	0.005	0.106
	0.2	3.837	13.657	0.102	0.000	0.016	0.263	0.005	0.105
	0.3	3.715	13.425	0.102	0.000	0.017	0.262	0.004	0.113
	0.4	3.705	23.994	0.102	0.000	0.016	0.264	0.005	0.124
	0.5	3.964	46.213	0.102	0.000	0.016	0.341	0.005	0.118

Conclusions

- Our goal was to address the numerical difficulties arising from the non-differentiability of the L1 penalty in PEP by applying smoothing
- Our proposed smoothed PEP retains clear population stratification, increases numerical stability, and obtains meaningful eigenvectors
- It also increases prediction accuracy in polygenic risk scores and enhances discernibility of clusterings
- In comparison studies, the smoothed PEP consistently demonstrates high accuracy, state-of-the-art support recovery, and fast runtime!

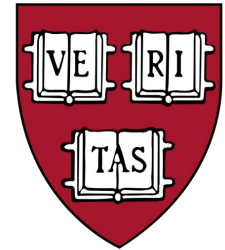
Future Directions

- R package SPEV!
- Developing a rigorous theoretical proof demonstrating the increased numerical stability of smoothed PEP
- Conducting further experiments on genomic data, potentially identifying new associations not detectable with ordinary principal components

Acknowledgments



SF Bay Area ASA
Student Travel Award for JSM 2025



HARVARD
UNIVERSITY

Dr. Georg Hahn
Department of Biostatistics
Harvard ERC Award



Stanford
University

Department of Biomedical Data Science
NLM T15 Training Grant

Thank You!