[5주차] A projection pursuit framework for supervised dimension reduction of high dimensional small sample datasets 요약

2022.02.17 예지혜

Introduction

PP는 인덱스에 따라 다양한 패턴 인식이 가능하고, 새로운 데이터도 매핑할 수 있다는 장점이 있다. 클러스터링, 분류, 회귀, 분포 추정 등 다양한 목적을 위한 PP 인덱스가 연구되어 왔다.

하지만 유전자 microarray 데이터와 같이 n<<p인 데이터의 계산 비용이 크다는 것이 문제점이다. 전통적인 PP 최적화 방법인 그래디언트 방법이나 Newton 방법은 매우 예민해서 poor local을 찾아버리는 단점이 있다. 이후 GA, SA, RSSA, PSO 등 다양한 방법이 등장했지만 여전히 microarray 데이터를 다룰 정도의 수준은 도달하지 못했다.

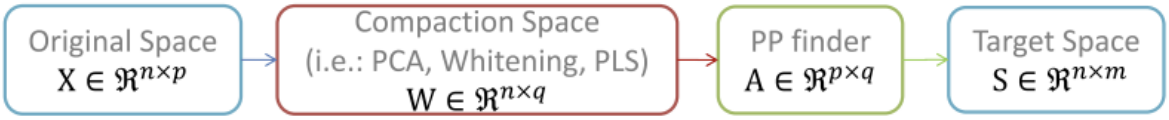


Fig. 1. Framework WSPP.

이러한 문제를 해결하기 위해 이 논문에서는 2단계의 프레임워크를 제안한다. Step 1은 compaction stage로 PCA, Whitening, Partial Least Squares 등의 방법을 통해 데이터를 압축한다. Step 2는 PP 적용 단계로 앞서 압축시킨 데이터에서 GA optimizer와 결합한 SPP 방법을 통해 투영을 실행한다. 이 논문에서는 microarray 데이터를 이용해 다양한 압축 방법, 다양한 PP index, 여러 차원으로 구성된 여러 개의 세팅에 대해 앞서 소개한 프레임워크를 실험해본다.

2. Projection pursuit

2.1. Sequential projection pursuit (SPP)

m개의 방향을 찾기 위해 SPP는 m번의 연쇄적인 최적화를 진행한다. 첫 번째 방향 a_1을 찾으면 이와 다른 방향을 찾기 위해 첫번째 방향으로 데이터를 "Gaussianize"한다. 이러한 구조 제거를 통해 찾은 두 번째 방향 a_2는 a_1과 직교한다.

2.2. PP optimization

PP에서는 최적화 방법도 중요한데 이 프레임워크에서는 PPGA를 사용한다.

얻어진 결과들이 데이터의 서로 다른 정보를 의미한다는 것이 보장되어야 하는데, 이를 위해 "structure removal" 방식이 사용되어져 왔다. 하지만 이러한 연쇄적인 적용 방법은 데이터의 왜곡을 일으킬 수 있다. 즉, 제거하고 남은 데이터가 더 이상 원데이터의 정보를 가지고 있지 않을 수도 있다는 것이다.

이를 보완하기 위해 orthogonal complement space 개념을 사용하는 방법이 제안되었다. 현재 데이터를 찾아낸 투영 벡터의 orthogonal complement 방향으로 투영시켜 다음 과정을 진행하는 방법이다. 그러면 각 투영 공간 간의 직교성을 유지하면서 데이터 왜곡을 막을 수 있다.

2.3. Projection pursuit indices

- Index Bhattacharya (Bat): 그룹 간 Bhattacharya 거리

$$\Im_{Bat} = \min_{i,j \in C} \left\{ \frac{1}{4} \frac{(\mu_i - \mu_j)^2}{\sigma_i + \sigma_j} + \frac{1}{2} \log \left(\frac{\sigma_i + \sigma_j}{2\sqrt{\sigma_i \sigma_j}} \right) \right\},$$

- Index quality projected clusters (qpc): compact pure 클러스터를 찾는 인덱스. 알파는 같은 그룹일 때 양수, 다른 그룹이면 음수이다.

$$\Im_{qpc} = \sum_{i,j=1}^{n} \alpha_{i,j} G((x_i - x_j)\mathbf{a}),$$

- Index Fisher linear discriminant analysis (lda): LDA 방법이 적용된 인덱스로, 그룹 간 거리를 크게 하고 그룹 내 거리를 작게 하는 투영을 찾는다. B는 그룹 간 행렬, W는 그룹 내 행렬이다.

$$\Im_{lda} = 1 - \frac{|\mathbf{A}^T \mathbf{W} \mathbf{A}|}{|\mathbf{A}^T (\mathbf{W} + \mathbf{B}) \mathbf{A}|},$$

- Index neighborhood components analysis (nca): nca 방법의 비용함수로, 올바르게 분류되었을 확률을 의미한다.

$$\Im_{nca} = \sum_{i}^{n} \sum_{j \in \Omega_i} p_{ij}, \quad p_{ii} = 0 \text{ and } p_{ij} = \exp(-\|x_i\mathbf{A} - x_j\mathbf{A}\|^2) / \sum_{k \neq i} \exp(-\|x_i\mathbf{A} - x_k\mathbf{A}\|^2)$$

- Index locality preserving (Lp): Lpp 방법을 기반으로 하는 비지도 학습 인덱스이다. 이웃한 데이터들을 모으는 투영을 찾아내며, Lpp 기준의 역수 형태이다. L은 k근접 그래프의 Laplacian 행렬이다.

$$\Im_{Lp} = 1/(\mathbf{A}^T \mathbf{X} \mathbf{L} \mathbf{A}),$$

3. A PP framework for supervised dimension reduction of large p small n data

3.1. Compaction stage

p차원의 X 공간을 q차원의 W 공간으로 압축시키는 것이 목적으로, 인지도, 구현 가능성, 계산 용이성을 기준으로 여러 방법들을 테스트하였다.

- The whitening transform (Whiten)
Sphering이라고도 불리는 이 방법은 상관성이 없고 정규화 된 데이터로 변환한다. 변환 후 각 데이터는 (n-1) 차원의 simplex의 꼭짓점에 배치되기 때문에 거리 기반 방법은 사용할 수 없다. 하지만 변형된 데이터의 무관한 특성들을 제거하다 보면 informative 데이터를 만들 수 있다고 한다. Whitening은 centered matrix의 SVD를 통해 얻어진다. $X = UDV'$에서 U의 첫번째 q개의 열이 whitened data W가 된다. 또, $W = XR$로 두면 이때의 $R = \tilde{V}\tilde{D}^{-1}$는 transformation matrix에 해당한다. ($\tilde{V}$는 V의 첫 q개 열)

- PCA
마찬가지로 SVD를 사용하며, $W = XR$로 두면 이때 $R = \tilde{V}$이다.

Whitening과 PCA 모두 SVD를 사용하며, 직교성을 보장하기 위해 제약조건 $R^T R = I$를 사용한다.

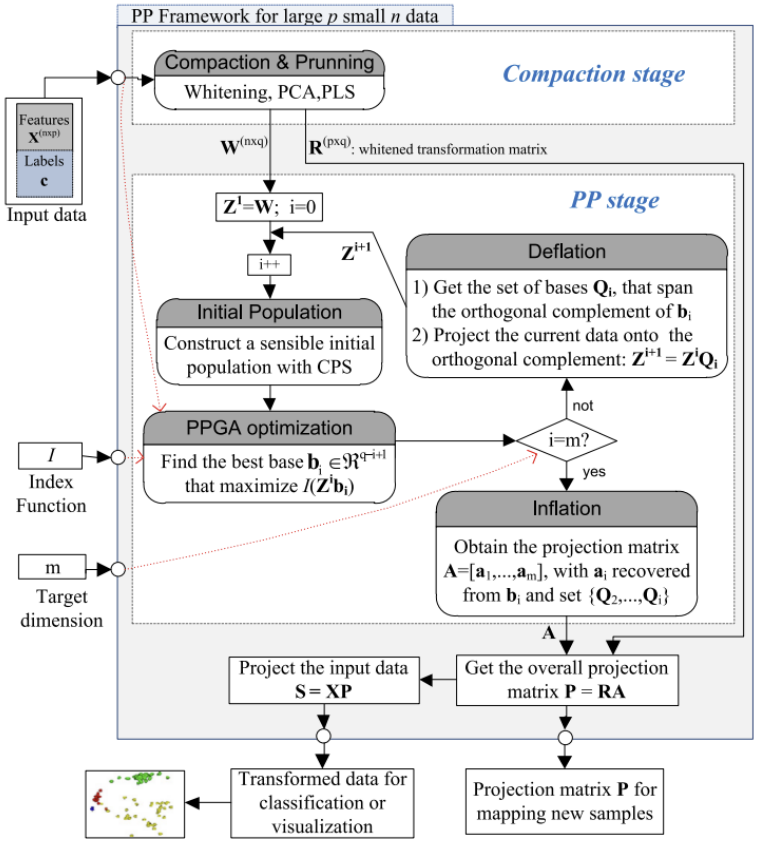
- Partial least squares (Pls)
Pls는 잠재변수 간의 관계를 모델링 하는 지도 학습 방법으로, 이를 이용해 Y와 X간의 공분산을 최대화하는 가중치 벡터 r을 최적화한다. 다중 회귀와 PCA의 중간 정도로 여겨지며 Pls 결과는 무상관이고 크기 순으로 정렬된다.

3.2. PP stage

압축된 데이터에서 투영을 찾는 단계에서는 SPP 방식을 약간 수정하여 사용한다.

- Initial population: 수렴 시간을 줄이고 이후의 PP 최적화를 적절히 진행하기 위한 요소로, 클래스의 경계 정보를 이용하여 후보 projection bases를 얻는다. (CPS 방법)

- Projection pursuit genetic algorithm (PPGA): GA를 약간 수정한 최적화 방법으로, GA는 바이너리 인코딩과 canonical 연산자를 사용하지만, PPGA는 실수 인코딩과 crossover 연산자를 사용한다. 크로스오버 연산자는 높은 무작위성을 띠고, mutation은 오히려 수렴 속도를 낮춰서 사용하지 않는다.

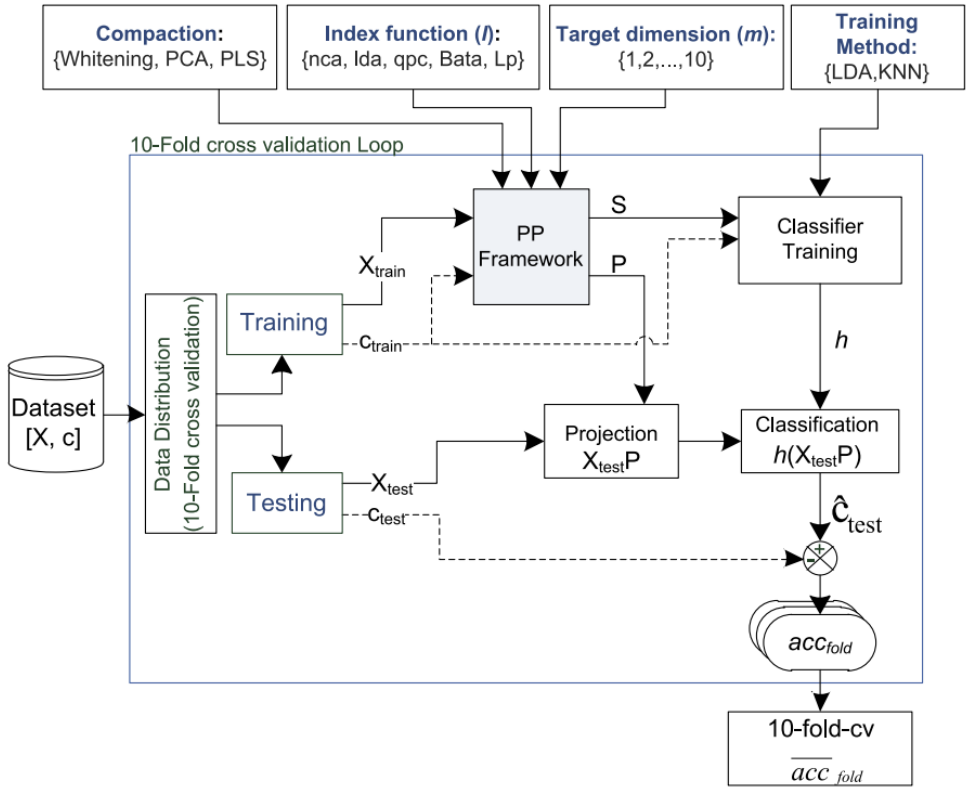


- Deflation/Inflation: 이 과정은 반복을 통해 얻어진 투영들이 서로 직교함을 보장하기 위한 방법이다. 디플레이션 과정은 residual data Z^i로부터 basis b_i를 찾고, 이의 직교 보수 basis Q_i를 계산해 다음 residual data를 구한다. ($Z^{i+1} = Z^i Q_i$) 이 방법의 장점은 단계를 반복할수록 차원이 낮아져 base를 얻는 게 쉬워진다는 것이다.
인플레이션 과정은 모든 m개의 베이스를 모두 얻은 후 다시 투영행렬 A를 계산하는 과정이다.

4. Experimental evaluation

4.1. Experimental setup

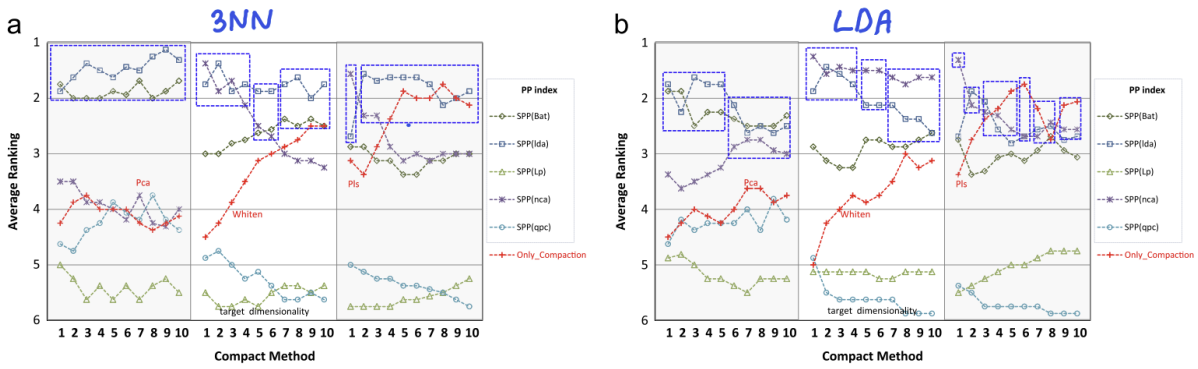
compaction 방법(Whitening, PCA, PLS), PP index, 타겟 차원, 분류기 방법(LDA, K-NN) 등 다양한 구성으로 다음의 실험이 독립적으로 진행된다. 성능 비교는 10-fold cv error로 평가되며, 데이터는 10개의 폴드로 분해되어 train과 test data로 나뉜다. 이 중 train data만 PP 프레임워크를 거쳐 분류기를 학습시키는 데에 사용된다. 이를 통해 얻어진 투영행렬과 분류 모델을 test data에 적용해 성능을 확인한다.



4.2. Results and discussion

compaction의 성능을 비교하기 위해 PP를 거치지 않은 결과(Only Compaction)도 함께 비교하였다. 대체로 지도학습인 PLS가 Whitening과 PCA보다 이 프레임워크를 진행했을 때 성능 개선이 크지 않다. 타겟 정보를 이용하기 때문에 애초에 성능이 좋기 때문이다. (그렇다고 해서 PP search가 더 좋아지는 것은 아니다.)

lda index는 낮은 차원에서도 정보를 잘 찾기 때문에 다른 인덱스보다 성능이 좋다.



의미를 더 정확하게 분석하기 위해, 각 PP 인덱스의 정확도를 분류기와 차원에 따라 평균을 냈다. 다양한 데이터에서 여러 번 반복된 실험 결과를 평균 낸 것이다. 통계적 차이를 검정하기 위해서는 Friedman의 검정 통계량과 Dunn's technique을 사용하였다. 통계적으로 차이가 유의하지 않은 것들을 점선 박스로 나타냈다.

그 결과를 정리해보자면, pls의 경우 1, 2차원에서만 프레임워크가 더 잘 작동했고, 고차원에서는 별 차이가 없었다. qpc와 Lp 인덱스는 대체로 가장 좋지 않은 성능을 보였다.

Dimension Reduction Method	Classifier	3NN					LDA					Total runtime of experiments (s)
	Dimension	2	4	6	8	10	2	4	6	8	10	
	Pca+SPP(Bat)	5.1	4.8	3.3	3.1	2.1	5.7	6.0	6.4	6.4	6.6	175103
Best →	Pca+SPP(lda)	4.6	**2.6**	**2.0**	**1.6**	**1.7**	6.9	5.6	5.6	6.5	6.6	47287
	Pca+SPP(Lp)	18.4	17.2	15.8	15.1	14.1	16.3	16.0	15.9	15.4	14.9	74413
	Pca+SPP(nca)	11.8	10.0	10.3	10.4	8.1	11.8	9.6	8.0	7.7	8.3	252075
	Pca+SPP(qpc)	15.0	12.9	10.6	9.3	9.4	14.1	12.5	11.5	11.1	10.8	254061
	Pls+SPP(Bat)	8.7	8.1	9.3	10.5	10.6	9.9	9.5	9.8	9.5	10.0	119542
	Pls+SPP(lda)	4.4	4.5	4.6	6.1	7.2	4.3	7.9	8.6	8.4	8.1	51268
	Pls+SPP(Lp)	21.3	22.3	21.8	21.4	20.9	20.6	20.8	21.0	20.6	20.5	50106
	Pls+SPP(nca)	7.1	8.3	11.4	12.4	13.1	5.6	5.7	6.4	6.9	7.2	172127
	Pls+SPP(qpc)	17.6	19.9	20.1	20.9	21.4	19.9	20.9	21.1	21.8	21.8	192253
	Whiten+SPP(Bat)	8.8	9.0	10.6	10.8	12.3	8.9	8.6	8.1	8.3	8.4	141286
	Whiten+SPP(lda)	**2.6**	5.6	7.4	7.9	9.4	**4.1**	4.5	6.1	7.3	7.6	45673
	Whiten+SPP(Lp)	21.4	21.4	21.3	21.4	21.3	19.8	19.9	19.9	19.8	19.5	55262
	Whiten+SPP(nca)	4.4	7.1	11.3	12.9	14.3	4.3	**4.3**	**4.3**	**5.6**	5.4	212116
	Whiten+SPP(qpc)	17.5	20.0	20.6	21.5	22.0	20.3	21.1	21.0	21.4	21.4	222746
	Pca	13.8	11.8	10.6	10.1	8.8	15.1	12.3	11.3	9.8	9.6	565
	Whiten	13.6	12.4	10.9	11.6	12.0	15.1	12.3	11.3	9.8	9.6	571
	Pls	9.0	6.3	5.3	6.0	8.5	6.8	6.4	5.4	7.1	**4.9**	3268
	LLE	15.9	16.4	15.9	15.5	15.9	16.0	16.3	14.8	13.9	14.4	1914
	NCA	15.1	14.3	12.9	11.3	9.5	14.5	13.6	13.3	12.7	13.5	228902
	SIR	10.6	10.9	11.9	11.4	10.9	9.4	13.0	15.3	15.5	16.9	288
	ReliefF	14.9	14.9	14.1	12.6	10.9	14.0	14.7	15.8	15.3	15.1	15887
	Ttest	14.6	15.5	14.0	12.4	11.9	12.9	14.7	15.6	15.6	14.9	12709

각 프레임워크 조합과 기존의 차원축소 방법론 8가지를 포함해 총 23가지의 방법을 비교한 결과이다. 대체로 프레임워크의 성능이 좋았고, 그 중에서도 PCA + LDA 인덱스의 조합이 좋은 성능을 보였다. 기존 방법 중에서는 Pls의 성능이 가장 좋았다. 속도 면에서는 프레임워크가 대체로 더 느리지만, 실제로 데이터에 적용할 때는 몇 초 밖에 차이 나지 않는다. 속도 면에서도 PCA + LDA 조합이 가장 경제적이다.

5. Conclusion

대체로 프레임워크의 성능이 낮은 차원으로의 차원축소에서 더 뛰어난 것을 확인할 수 있었다.

데이터의 특성에 따라 잘 작동하는 프레임워크의 구성이 달라지므로, 이를 찾기 위해서는 메타학습 접근법을 사용하는 것이 좋다.