creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

이 학 석 사 학 위 논 문

Convex hull peeling을 이용한 eigen vector 추정

2018년 2월

서울대학교 대학원

통계학과

최 지 수

Contents

1 서론 **3**

2 데이터 분류로써의 Convex hull peeling과 특징 **5**

2.1. Convex hull peeling을 이용한 고유벡터 추정 7

2.1.1. 가중 평균이 필요한 이유 9

2.1.2. Convex hull의 선택 11

3 모의 실험 **13**

3.1. 실험 설계 13

3.1.1. Simulation 1 : 잡음이 없는 경우 17

3.1.2. 잡음이 있는 경우 : Simulation 2,3,4 18

4 다변량에서의 적용 **21**

5 결론 **27**

References **28**

List of Tables

List of Figures

2.1 전체 데이터와 hull별 데이터 6

3.1 4가지 모의 실험 15
3.2 Simulation 1 17
3.3 잡음이 있는 경우 18
3.4 전체적인 잡음이 있는 경우 hull별 잡음점 20

4.1 다차원 확장 26

국문초록

다차원 데이터를 분류하는 방법중 하나로써 Convex hull peeling이 분포의 고유벡터를 추정하는데 가진 특징을 관찰한다. 이를 통해 Convex hull peeling을 이용한 고유벡터 추정 방법을 만들어 본다. 잡음과 이상점이 있는 모의 실험을 통해 제안된 방법이 가진 효과를 살펴보고 마지막으로 다차원으로 확장하는 방법을 생각한다. 모의 실험 결과 잡음과 이상점이 있는 경우에는 기존의 고유벡터 추정방법보다 효과적인 추정값을 제시함을 보였다. 또한 8차원까지 확장한 결과에 대해서 확장하였을 때에도 효과적으로 추정한다는 결과가 나왔다. 하지만 다차원 확장을 하는 과정에서 방법이 유일하지 않고 개선 할 수 있고 추후 연구가 가능하다.

Keyword : *Convex hull peeling, robust,eigen vector*

Student Number : 2016-20278

Chapter 1

서론

분산행렬은 변수의 산포와 다른 변수의 관계를 알아볼 수 있는 통계량이다. 특히나 다차원 변수를 쓰는 분야에서는 대부분 분산행렬을 이용하여 분석하는 방법이 많다. 분산행렬을 이용하여 다차원 데이터를 분석하는 방법의 하나로써 고윳값분해(Eugen decomposition)가 있다. 고윳값분해는 분산행렬을 이해하는 대표적인 방법의 하나다. 고윳값 분해를 이용하는 방법들은 대표적으로 먼저 표본 분산행렬을 구하고 그것을 고윳값분해하여 고윳값과 고유벡터를 구하게 된다. 고윳값 분해는 행렬을 고유벡터와 고윳값으로 분해함으로써 변수들의 성질을 보기 쉽게 파악할 수 있게 한다. 실제로 다차원 데이터를 분석하는 방법 중 고윳값분해를 이용하는 방법은 인자분석(Factor analysis)과 주성분 분석(PCA) 등이 있다.

대표적인 다차원 데이터의 분석에 사용되는 고윳값 분해방법은 표본 분산행렬을 분해하는 방법이다. 이때 사용하는 표본 분산행렬은 적률 이용 추정량이다. 적률 추정량은 데이터의 이상치나 잡음에 영향을 쉽게 받는다. 따라서 이상치나 잡음의 문제가 있으면 표본 분산행렬의 추정값이 정확하지 않게 된다. 데이터의 분산 행렬의 추정값이 원래의 값과 괴리가 크다면 이를 통해

구한 고유 벡터와 고윳값 또한 참값과 멀어지게 된다. 이처럼 데이터에 이상값과 잡음이 있다고 생각될 경우 고윳값 분해는 정확성을 잃고 고윳값 분해가 중간 단계로 사용되는 PCA와 Factor analysis의 통계 분석 결과는 신뢰하기 힘들어진다.

다차원 분석에서 분산행렬은 변수 간의 관계를 표현하고 분석의 대상이 되는 통계량이다. 그러나 앞서 설명했듯, 이상점이나 잡음에 영향받는 표본 분산행렬이 가진 문제점은 이후 다차원 분석 자체를 어렵게 하므로 이를 해결하기 위한 여러 가지 방법들이 소개되어 왔다.

예를 들면 데이터에서 이 상점을 제거하려 하거나 [1.1] 혹은 M estimator를 이용하여 [1.2] 강건한 공분산 행렬을 만들려고 하거나 처음부터 강건한 고유벡터를 만드는 방법[1.3] 등이 개발되어 왔었다. 위와 같은 방법들은 분포가정과 복잡한 수식이 사용된다. 그리고 강건한 분산행렬을 만들기 위해 분산 행렬을 만드는 경우 조정된 분산행렬이 양 정치행렬이 되는지 보여야 했고 그렇지 않으면 양 정치행렬로 만들기 위해 인위적인 조정이 필요하다.

본 논문에서는 분포가정이 필요 없고 간단한 방법인 Convex hull peeling을 이용하여 이상치나 잡음에 영향을 덜 받는 강건한 고유벡터 추정방법을 제안하고자 한다.

[1.1] Multivariate Outlier Detection and Robust Covariance Matrix Estimation

[1.2] Campbell (1980)

[1.3] Application of Robust Eigenvectors to the Compression of Infrared Spectral Libraries

Chapter 2

데이터 분류로써의 Convex hull peeling과 특징

과거에서부터 통계학자들은 데이터을 분류하기 위해 데이터의 순위(깊이)라는 개념을 생각해 왔다. 다차원 데이터의 순위(깊이)는 전체 데이터를 정해진 기준에 따라 같은 깊이의 그룹으로 나누어 질 수 있으며 구체적으로 Depth function 라는 함수를 이용해 깊이를 산출해낸다. Depth function의 종류는 Location Depth, Mahalanobis Depth, Oja depth, Convex hull peeling 등 다양한 함수가 있을 수 있다. Convex hull peeling은 Depth function의 한 종류로써 다차원에서도 적용가능하며 빠르게 데이터를 분류할 수 있다.

Convex hull peeling은 데이터 전체를 포함하는 convex hull을 만들고 만든 convex hull 제외하고 다시 convex hull을 만드는 과정을 반복한다. 이를 반복하면 모든 데이터는 어떠한 hull에 속하게 되고 데이터를 분류하는 방법이다. 일반적으로 hull이 바깥으로 갈수록 낮은 깊이를 부여하지만 본문에서는 안쪽부터 깊이의 순위를 부여했다.

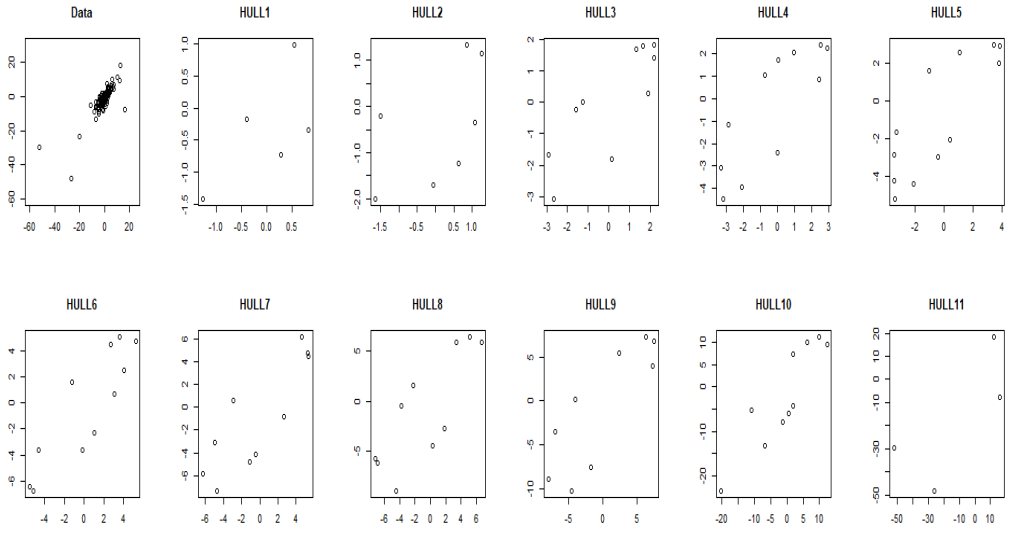

Figure 2.1: 전체 데이터와 hull별 데이터

Figure2.1은 2차원에서 분산 $\begin{pmatrix} 10 & 8 \\ 8 & 10 \end{pmatrix}$을 가지고 자유도가 3인 T-분포로부터 생성한 100개의 데이터를 를 Convex hull peeling 결과이다. Figure2.1 살펴볼 수 있는 Convex hull peeling의 첫 번째 특징은 데이터를 분류하는데 별다른 추정과 가정이 필요 없다는 점이다. Location Depth과 Mahalanobis Depth는 Depth function를 만드는데 평균점과 분산 행렬이 필요하고 주어진 데이터를 이용하여 추정한다. 그러나 Convex hull peeling은 별도의 추정과 가정 없이 convex hull이라는 직관적인 방법으로 바깥에서 중심 쪽으로 데이터를 분류한다. 또한 Convex hull peeling은 데이터를 그룹으로 묶어서 깊이를 부여한다. 같은 hull에 속한 데이터는 같은 깊이가 주어지며 다른 Depth function과 다르게 개별 데이터 별로 깊이를 세밀하게 부여하지 못하며 같은 깊이를 가진 점들이 여러 개 존재한다. 그리고 Convex hull peeling으로 나누어진 hull들은 깊이가 깊은 hull이 포함을 알 수 있으며 점점 커지는 모양임을 알 수 있다. 임

의의 점들의 집합에 대해서 Convex hull이 가지는 특징은 Efron(1965)[2.1]에서 찾아 볼 수 있다.

위 Figure2.1을 관찰하면 분포에서 주어지는 점을 Convex hull peeling으로 분류하면 중간 깊이의 hull부터 hull에 속한 점들의 분포모양이 전체 데이터 분포방향과 비슷함을 관찰 할 수 있다. 이러한 특징은 Convex hull peeling 방법상의 특징에 기인한 것이다. 각 hull의 분포 방향이 전체 분포와 비슷 함으로 각 hull별 고유벡터는 전체 데이터의 고유 벡터와 비슷하다. 결과적으로 임의의 점이 아닌 분포에서 생성된 점들을 Convex hull peeling으로 나누면 단순히 데이터를 나누는 역할을 할 뿐만이 아니라 전체 고유벡터를 보존하는 방향으로 데이터를 분할 하고 있음을 알 수 있다. 또한 이상점(Outlier)의 경우 Convex hull peeling을 수행할 시 높은 번호의 hull에 속한다. 이를 이용하면 Convex hull peeling으로 이상점도 찾을수 있다.

2.1. Convex hull peeling을 이용한 고유벡터 추정

d차원 벡터 $x_i \in R^d$ 으로 이루어진 전체 데이터 집합을 $D = \{x_1, \cdots, x_N\}$라 하고 데이터의 분산행렬(Σ) 는 $d \times d$ 행렬이 된다. 또한 분산행렬 Σ를 고윳값 분해하여 얻어지는 d차원 고유 벡터 $v_1, \cdots, v_d$들과 d개의 $e_1, \cdots, e_d$ 고윳값이라 하자. 실제로 정확한 분산행렬 Σ을 정확하게 알지 못함으로 표본 분산행렬($\hat{\Sigma}$)을 고윳값 분해 하여 고유벡터를 추정하는 게 일반적인 방법이다. 이런 기존의 방법을 방법0이라고 하자.

방법0의 절차를 구체적으로 쓰면 다음과 같다.

step.1 모든 데이터($x_i \in D$)를 이용하여 표본 분산 행렬 $\hat{\Sigma}$을 구한다.

step.2 표본 분산 행렬($\hat{\Sigma}$)에서 고유 벡터 $\hat{v}_1^0, \ldots, \hat{v}_d^0$를 구한다.

[2.1] The Convex Hull of a Random Set of Points, Bradley Efron

Convex hull peeling을 이용하여 데이터를 분류하면 모든 데이터는 Convex hull peeling에 의해 n개의 hull로 분류된다. 그 중에서 i번째 hull에 속하는 데이터의 집합을 D_i이라고 표기하면 Convex hull peeling의 방법의 특징상 분류된 데이터는 다음과 같은 성질을 가진다.

$$D = \bigcup_{i=1}^{n} D_i \ , \ D_i \cap D_j = \emptyset$$

다음으로 D_i에 속한 데이터의 표본 분산 행렬을 $\hat{\Sigma}_i$로 표기하고 i번째 hull에 속한 데이터 D_i로부터 표본분산행렬을 $\hat{\Sigma}_i$을 추정한 후, $\hat{\Sigma}_i$에서 부터 고유벡터를 추정한다면 그 추정 값을 전체 데이터의 고유벡터의 추정 값이다. 위 문단인에서 살펴 보았듯이 Convex hull peeling의 방법으로 나누어진 hull들은 전체 분포의 산포모양과 비슷하기 때문에 $\hat{\Sigma}_i$의 고유벡터가 Σ의 고유벡터의 추정 값과 비슷할 것으로 생각 할 수 있기 때문이다. 때문에 고유벡터를 추정할 때 Convex hull peeling으로 분류된 하나의 hull을 가지고 추정해도 되지만 데이터를 일부를 사용하는 것보다 가능한 많은 데이터를 사용하는 것이 정확한 추정을 가능하게 함으로 각 hull에서 구한 $\hat{\Sigma}_i$들을 적절하게 합쳐서 Σ의 고유벡터를 구한다. Convex hull peeling을 이용하여 고유벡터를 구하는 구체적인 방법은 다음과 같다.

step.1 Convex hull peeling을 이용하여 전체 데이터(D)를 n개의 hull D_i로 분류한다.

step.2 각 hull D_i 에서 추정분산행렬 $\hat{\Sigma}_i$과 각 hull의 면적 s_i을 구한다.

step.3 적절한 가중치 w_i를 설정한 후, 2차원일 경우 각 hull에서 구한 면적을 이용하여 $w_i = {s_i}^{-1}$로 설정 $w_1, w_2, \ldots, w_n$을 이용하여 가중평균 분산 행렬($\hat{\Sigma}_w$)을 다음과 같이 만든다.

$$\hat{\Sigma}_w = \frac{\sum_{i=1}^{n} w_i \hat{\Sigma}_i}{\sum_{i=1}^{n} w_i}.$$

step.4 $\hat{\Sigma}_w$로부터 고유벡터 $\hat{v}_1^1, \ldots, \hat{v}_d^1$을 구하고 이를 고유벡터의 추정 값으로 한다.

방법1은 전체 데이터를 Convex hull peeling을 이용해 hull을 기준으로 나누고 각 hull에서 분산행렬을 계산하고 계산된 분산행렬을 가중 평균하여 새로운 평균분산행렬을 만들었다. 방법1을 사용하여 만든 가중평균 분산 행렬 $\hat{\Sigma}_w$는 각 hull에서 만든 양 정치행렬들의 선형결합임으로 양 정치행렬이 된다, 정칙행렬 역시 같은 이유로 성립한다. Convex hull peeling을 통해 만든 가중평균행렬은 양정치 정칙행렬이 되고 이는 분산행렬의 기본적인 성질을 만족한다.

2.1.1. 가중 평균이 필요한 이유

Figure2.1에서 볼 수 있듯이 Convex hull peeling으로 나누어진 hull들은 전체 분포의 모양과 비슷하고 hull의 크기가 밖으로 갈수록 커진다. 데이터의 분산행렬을 Σ이라 하고 i번째 hull의 데이터를 D_i이라 할 때 D_i의 분산행렬 Σ_i은 적절한 상수 c_i에 의해 $\Sigma_i = c_i\Sigma$로 표현된다. 마찬가지로 D_{i+1} 역시 적절한 상수 c_{i+1}를 사용하여 $\Sigma_{i+1} = c_{i+1}\Sigma$로 표현될 수 있고 Convex hull peeling의 방법상 D_{i+1}는 D_i를 완전히 감싸기 때문에 해당하는 상수가 $c_i < c_{i+1}$인 관계를 가진다는 것을 알 수 있다. Convex hull peeling으로 나누어진 데이터들 (D_i)을 이용해 구한 $\hat{\Sigma}_i$들은 전체 분산행렬(Σ)에 어떤 상수들에 비례하고 각 상수 열들은 $c_1 < c_2 < \cdots < c_n$의 관계가 성립한다.

$$\Sigma_i = c_i\Sigma \quad \text{where } c_1 < c_2 < \cdots < c_n$$

분류된 hull의 분산행렬 $(\hat{\Sigma}_i)$에서 전체 분산행렬(Σ)을 효과적으로 추정 하기 위해 i번째 분산행렬 Σ_i에 비례상수 c_i를 알아야 하고 이를 이용하여 가중치w_i를 $w_i = c_i^{-1}$로 설정했다. 가중치가 곱해진 hull의 분산행렬은 $w_i\Sigma_i = \Sigma$ 가

된다. 비례상수 c_i들은 단조증가하며 hull에 속한 데이터의 산포의 크기에 비례함을 알 수 있다. 또한 전체 분산 행렬에 비례하는 정확한 상수는 알 수 없지만 c_i들과 같은 비율로 단조 증가하는 상수열을 알고 있어도 분산행렬 Σ를 추정 할 수 있다.

방법1에서는 가중치 값으로 hull의 면적 s_i^{-1}를 대신 사용했다. 2차원에서 hull의 표면적은 hull의 분산행렬인 $c_i\Sigma$의 분산 크기에 비례하기 때문에 산포의 크기를 반영하는데 적절하다. 또한 비례상수 c_i가 단조 증가하는 성질을 가지고 있듯 Convex hull peeling의 방법적인 특징으로 인해 hull들이 그전 hull을 포함하고 있기 때문에 표면적(s_i) 또한 단조 증가하는 성질을 가지고 있다.

비례상수가 $c_i \sim \kappa_i s_i$인 관계에 있으면 hull의 표면적(s_i)를 이용한 가중평균분산 행렬은 다음과 같이 된다.

$$\begin{aligned}
\Sigma_w = \frac{\sum_{i=1}^n w_i \Sigma_i}{\sum_{i=1}^n w_i} &= \frac{\sum_{i=1}^n s_i^{-1}\Sigma_i}{\sum_{i=1}^n s_i^{-1}} \\
&= \frac{\sum_{i=1}^n \kappa_i^{-1} c_i^{-1}\Sigma_i}{\sum_{i=1}^n c_i^{-1}\kappa_i^{-1}} = \frac{\sum_{i=1}^n \kappa_i^{-1}\Sigma}{\sum_{i=1}^n c_i^{-1}\kappa_i^{-1}} \\
&= \kappa\Sigma \quad \text{where } \kappa = \frac{\sum_{i=1}^n \kappa_i^{-1}}{\sum_{i=1}^n c_i^{-1}\kappa_i^{-1}}
\end{aligned}$$

따라서 비례상수(c_i)와 비례하는 새로운 상수열을 가지고 가중평균을 하여 가중평균 분산행렬을 만들게 되더라도 원래 분산행렬(Σ)에 비례하게 된다. 가중평균 분산행렬이 실제 분산행렬에 상수가 곱해진 행렬의 고유벡터 추정하는 것이 된다. 상수 배 차이가 나는 행렬은 고유벡터 값이 같기 때문에 상수 배(κ)를 정확히 모르더라도 고유벡터 값은 같음으로 가중평균 분산 행렬(Σ_w)은 전체 분산행렬을(Σ)의 고유벡터를 추정하는데 적절하다. 또한 가중치로 면적의 역수를 가중치로 생각하면 이상점이나 데이터 변동의 영향력을 크게 줄일 수 있다. 데이터 분포의 외곽의 hull에는 변동성이 큰 데이터 혹은 이상점이 있다. 외곽의 hull은 면적이 크기 때문에 가장 적은 가중치를 가지고 있기 때문에 가중평균 분산행렬에서 차지하는 역할이 적다.

고유벡터뿐만 아니라 Convex hull peeling방법을 통해 고윳값을 계산할 수 있다. 가중평균행렬을 통해 고유벡터를 구하고 구한 고유벡터를 가지고 고윳값을 구할 수 있다. 구체적으로, 가중평균행렬로 구한 고유벡터를 u라고 할 때 그에 해당하는 고윳값(v_u)이 다음식이 성립해야 함을 이용한다

$$v_u = u^t \Sigma u$$

이렇게 고유벡터와 고윳값을 구하는 방법은 전체분산행렬의 고윳값과 고유벡터를 동시에 구하지 않고 Convex hull peeling을 이용하여 고유벡터를 먼저 구하고 고윳값을 구한다. 고유벡터와 고윳값은 따로 떨어져서 생각할 수 없는 것이지만 고유벡터의 정확성만을 높일 수 있다면 얽혀 있는 성질을 이용하여 고윳값 또한 구할 수 있다.

3차원이상에서는 면적의 개념이 명확해지지 않기 때문에 가중치로 면적을 이용하는 것보다는 각 hull의 분산행렬(Σ_i)에서 고윳값을 이용하는 방법도 있다. 전체 분산행렬의 고윳값을(e_k , $k = 1, \ldots, d$) 이라고 하면 Convex hull peeling으로 나누어진 데이터 의 분산행렬 Σ_i의 고윳값(e_k^i , $k = 1, \ldots, d$)은 다음과 같이 쓸 수 있다.

$$e_k = c_i e_k^i$$

따라서 같은 순번의 고유 벡터끼리는 c_i만큼 비례하고 이를 통해 비례상수와 비슷한 상수열을 만들 수 있다. 각 hull별로 계산한 분산행렬을 가지고 가중평균 분산행렬을 만드는 방법 이외에도 각hull별로 고유벡터를 구해 이를 합치는 방법도 있을 수 있다.

2.1.2. Convex hull의 선택

이상점과 잡음이 있을 경우 고유벡터 추정 시 n개의 hull 전체를 사용하는 대신 hull의 선택이 필요하다. Figure2.1을 관찰한 결과로 convex hull peeling

을 통해 알 수 있듯이 모든 hull들이 전체 데이터 분포 모양을 따르지는 않는다. Figure2.1의 경우 첫 번째 hull은 점이 2개의 점으로 이루어져 있다. 많은 경우에 안쪽으로 갈수록 hull을 이루는 점의 개수는 줄어들고 가장 안쪽에 이르러서는 2개나 3개의 점이 hull을 구성하고 있게 된다. hull을 구성하는 점이 적으면 hull의 분포가 전체적인 분포 모양과 다르게 우연적인 효과에 영향을 받게 된다. 또한 Figure 2.1에서 hull에 포함되는 데이터 수는 안쪽에서 바깥쪽으로 갈수록 늘어나고 가장 바깥에 있는 hull 또한 hull에 포함되는 데이터 수도 적기 때문에 전체 분포 모양을 반영하지 못한다. 따라서 가장 안쪽과 끝부분에 속한 hull들은 데이터수가 2개나 3개로 적기 때문에 어떤 분포를 형성한다기보다는 우연적인 영향으로 모양이 정해진다. 따라서 안쪽과 바깥쪽 hull에 한해서 조정 해야 한다는 것을 알 수 있다. 또한 앞선 문단에서 언급했듯이 Convex hull peeling의 특징으로 인해 깊이가 깊은 외곽의 hull일 수록 hull에 이상점과 변동성이 큰 데이터가 있기 때문에 이를 반영해야 한다.

이와 같은 hull의 문제를 해결하기 위해서 hull을 제거하거나 몇 개의 hull을 합치는 것을 고려할 수 있다. 안쪽에 존재하는 hull은 단지 적은 수의 데이터가 있음으로 인해 문제가 발생하는 것임으로 두 세 개의 hull을 합쳐 하나의 데이터 집합으로 놓음으로 부족한 데이터 수를 채우고 바깥쪽의 hull의 경우 이상점이 존재하거나 변동성이 큰 데이터가 포함될 가능성이 있음으로 부분 제거하였다. 이렇게 hull을 조정할 경우 몇 개의 hull을 합치고 몇 개의 hull을 제거 해야 하는지는 신중하게 생각을 해봐야 한다. 안쪽 hull의 합침은 데이터 손실이 없지만 바깥쪽 hull의 제거는 데이터에서 얻을 수 있는 정보를 잃는 것이기 때문에 많은 수의 hull의 제거는 추정을 오히려 힘들게 한다는 것을 고려하며 정해야 한다. 결국 hull의 선택 정도는 데이터에 이상점이나 잡음의 존재 여부에 따라 달라지며 데이터의 분포를 관찰하고 나서 이상점이나 잡음이 존재한다고 판단될 경우 몇 개의 외곽 hull을 제거하는 것이 좋다.

Chapter 3

모의 실험

잡음과 이상점이 있을 경우 Convex hull peeling을 이용한 방법이 고유벡터를 정확하게 추정하는지 기존의 방법과 비교하기 위해 모의실험을 설계했다. 현실 데이터에서는 정확한 고유벡터를 알 수 없는 경우가 많음으로 R 프로그램을 사용하여 주어진 고유벡터를 가진 데이터를 생성하고 생성된 데이터를 이용하여 고유벡터 추정을 했다.

3.1. 실험 설계

분산행렬 Σ의 고유벡터들을 $v_1, \cdots, v_d$라고 하면 한번의 모의실험에 주어진 Σ를 따르는 데이터를 n개 생성한다. 그리고 잡음을 표현하는 데이터 n'개를 넣어 Σ를 따르는 분포와 합친다. 결과적으로 $n+n'$개의 데이터를 가지고 위에 설명한 2가지 방법(방법0,방법1)들이 얼마나 정확하게 고유벡터 값을 $v_1, \cdots, v_d$ 를 추정했는지 비교한다. 추정 값의 비교기준으로 실제 고유벡터와 추정벡터의 차이를 두 벡터열의 유클리드 Norm의 합을 비교기준으로 놓았다. 방법j 의 에러 값은 방법j 를 사용하여 구한 $\hat{v_1^j}, \cdots, \hat{v_d^j}$과 참값인 고유 벡터 $v_1, \cdots, v_d$의

유클리드 Norm의 합을 계산한 값이고 SE_j 로 놓았다.

$$SE_j = \sum_{i=1}^{d} \hat{v_i^j} - v_i$$

한번의 모의실험에서 새로운 분산행렬로 데이터를 생성하고 고유벡터를 두 가지 다른 방법(방법0와 방법1)으로 추정하기 때문에 SE_0, SE_1 두 가지 값이 나오게 된다. 이런 반복적인 모의 실험을 m번 반복적으로 할 것이기 때문에 이를 구별하기 위해 i번째 모의실험에서 나온 에러 값을 SE_{i0}, SE_{i1} 라고 하고 매 반복 시 마다 두 오차의 차이 $SE_{i1} - SE_{i0}$ 를 저장하고 상자그림을 그려 각 방법이 참값인 고유벡터 v를 추정하는데 어느 정도 정확도로 추정하였는지 살펴보았다. 상자그림의 평균과 대부분의 값이 음수에 속해 있는다면 반복적인 모의실험에서 방법1이 매번 데이터를 새로 생성할 때마다 더 나은 추정 량을 제공했다는 것을 의미 한다.

또한 모의 실험을 여러 번 시행하기 때문에 일일이 데이터를 살펴 볼 수 없음으로 Convex hull의 선택에 관한 사항들(Convex hull을 얼마나 제거할 것인지와 어떤 가중 값을 사용할 것인지)을 통일하였다. 본래는 데이터를 관찰하고 어느 정도의 hull을 제거하고 합칠 것인지를 고민해야 더 좋은 추정을 얻을 수 있지만 많은 수의 반복적인 실험을 할 때 마다 일일이 정해 줄 수 없음으로 모두 같은 방법을 사용하였다. 이상점과 잡음을 넣은 상태의 모의 실험에서는 반복적으로 500번의 모의 실험 데이터를 생성하고 분석함으로 매번 분포를 관찰하여 hull의 수준을 정할 수 없다. 따라서 일괄적으로 안쪽(3개) 부분은 합치고 바깥 부분(3개) 은 제거하는 방법을 썼다. 각각 3개의 hull을 제거하고 합치는 방법이 제일 좋은 방법인지는 알 수 없다. 몇 개의 hull을 조정해야 하는지는 데이터의 분포에 따라 다르기 때문이다.

Convex hull의 선택에 관한 자세한 방법은 아래와 같이 두었다.

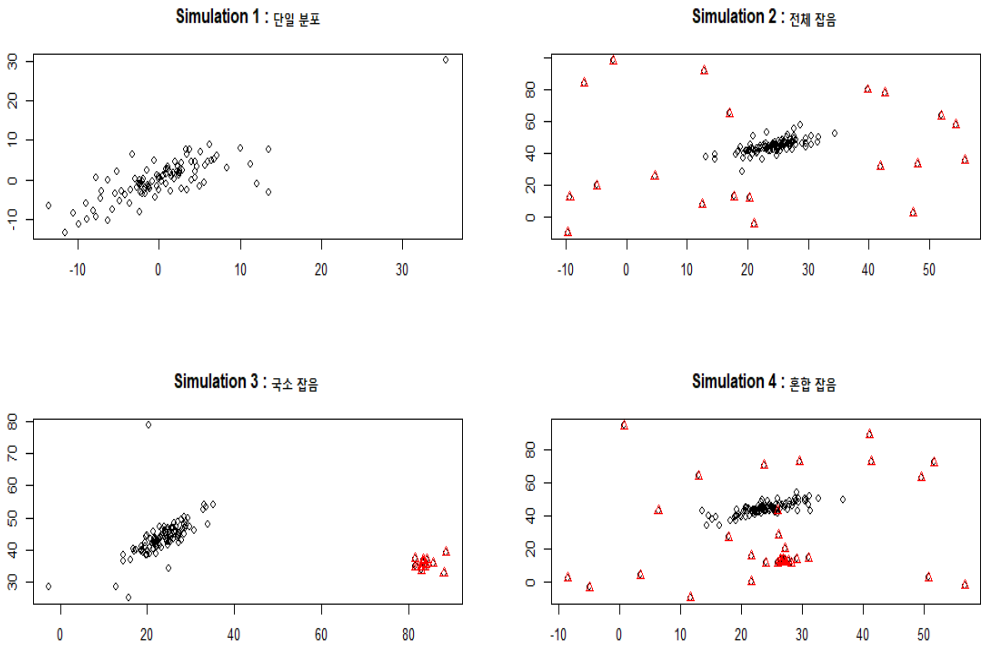

Figure 3.1: 4가지 모의 실험

a way0의 방법은 전체 데이터를 가지고 표본 분산 행렬을 만든 다음 고유 벡터를 구하였다.

b (가중 값 선택) way1의 가중 값의 선택으로 각 hull의 넓이의 역수를 가중치로 생각하였다.

c (Convex hull선택) 가장 안쪽에 있는 hull에서 3번째까지의 hull은 하나의 hull로 합치고 바깥의 3개의 hull은 제거하였다.

모의실험은 다양한 잡음이 있는 상황을 구현하기 위해 4가지 상황으로 나누었다. 4가지 모의실험 상황을 Simulation1 , Simulation2 , Simulation3 ,

Simulation4로 표기 하였다. Figure3.1에서 파란색으로 표시된 점은 잡음과 이상점을 나타 내고 검은색으로 표시된 데이터들은 정상적으로 Σ를 따라 생성된 데이터이다. 4가지 모의 실험 모두 Σ는 $\begin{pmatrix} 10 & 8 \\ 8 & 10 \end{pmatrix}$이고 평균은 원점이다. 분포 또한 최대한 이상점이 있는 상황을 만들기 위해 T분포 자유도 3로 설정하였다. 모의 실험들은 모두 동일하게 500번 반복해서 오차 값의 제곱 합을 계산하였다. 구체적인 모의실험 상황은 다음과 같다.

1. Simulation1은 잡음 없는 상황일 때 에러 값을 비교하기 위해 구현된 상황이다. T분포의 자유도를 3으로 줌으로써 잡음이 없더라도 이상점이 있을 때의 상황을 구현하였다.

2. Simulation2에서는 잡음이 균일하게 퍼져 있을 때를 상정한 경우로 T분포 자유도3에서 100개의 데이터를 생성하였다. 그 후 100개의 좌표의 (x, y) 성분값중 절대값이 가장 큰 값에 +10을 한 범위만큼을 창으로 하여 uniform한 잡음을 20개가 추가하였다.

3. Simulation3은 T분포 자유도3 에서 생성된 100개의 데이터에 국지적 잡음 15개가 추가되었을 경우이다. 잡음은 4개의 점 $(\pm 11, \pm 23)$중 임의 점을 중심점으로 T분포 자유도 2를 따른다.

4. Simulation4은 Simulation3와 Simulation2의 경우를 합한 경우로 T분포 자유도3 에서 생성된 100개의 데이터에 국소적인 잡음15개와 전체적인 잡음20개 총 35개의 잡음이 동시에 있을 경우를 생각해보았다.

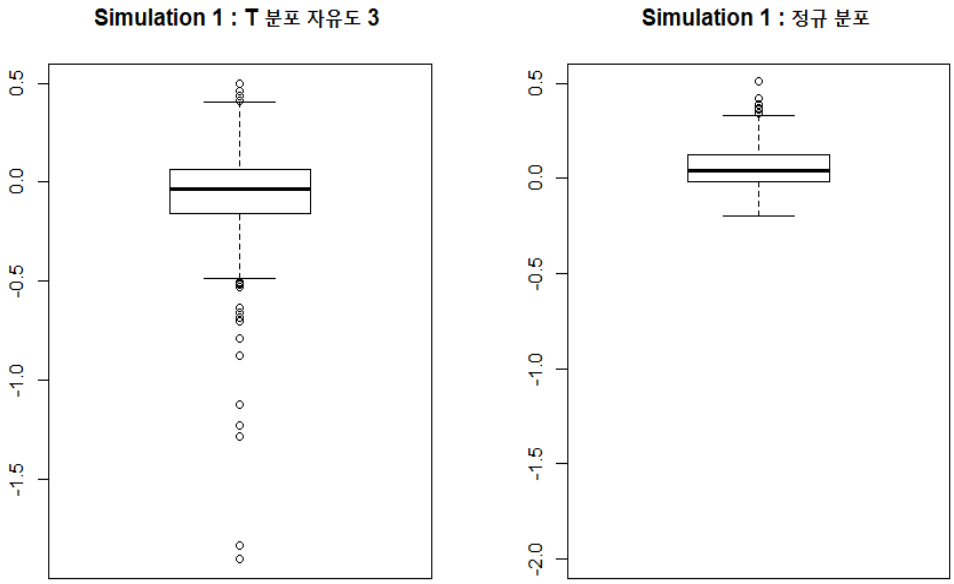

Figure 3.2: Simulation 1

3.1.1. Simulation 1 : 잡음이 없는 경우

Figure3.2은 모의실험을 1번 했을 때 같은 분산행렬을 가지고 있는 두 개의 분포 Normal 분포와 T분포(자유도 3)에서 각각 100개의 표본을 생성하고 앞서 말한 3가지 방법으로 참값과의 오차제곱을 계산하는 모의실험을 500번 반복한 상자그림이다. 상자그림의 x축의 way0은 방법0으로 구했을 때 나오는 고유벡터 추정 값과 참값의 오차제곱합의 분포이며 way1은 방법1에 대응된다. Simulation 1 처럼 잡음이 없는 경우, Figure3.2 왼쪽에서 볼 수 있듯이 상자그림의 결과 정규 분포가 모 분포 일 때는 적률추정법을 기반으로 한 방법0가 오차의 제곱 합이 더 작은 경향이 있음을 알 수 있다. 그렇지만 T분포 자유도3에서는 오차 제곱 합이 서로 비슷하다. 이는 T분포 자유도3에서는 이상점이

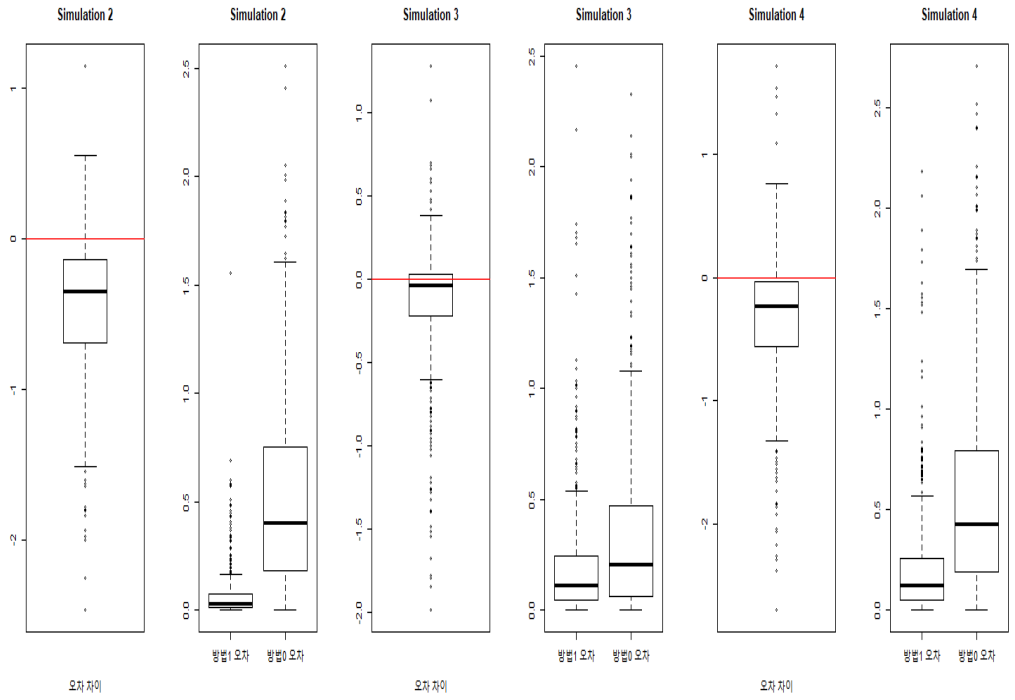

Figure 3.3: 잡음이 있는 경우

생기기 쉽기 때문에 이상점이 있을 경우, 방법0 에서 적률추정의 한계점과 방법1에서의 Convex hull peeling방법의 추정을 실제 값에 더 가깝게 한다고 볼 수 있다.

3.1.2. 잡음이 있는 경우 : Simulation 2,3,4

잡음과 이상점이 분명하게 존재하는 Simulation 2,3,4의 경우에 모의실험을 반복하였다. Simulation 2,3,4에서 데이터는 모두 T분포 자유도3에서 100개의 표본에 잡음의 상황이 각기 다른 경우이다. Convex hull peeling 을 이용한 방법1이 방법0보다 오차제곱의 합이 적은 결과를 얻어 냄을 관찰 할 수 있다. 먼저, Convex hull peeling 이용한 방법1은 일괄적으로 가장 바깥에 있는 3개의

hull을 제거하고 , 그 외의 hull의 데이터를 가지고 가중평균행렬을 만들고 고유벡터를 추정 하였다. 위 과정 중 hull을 제거하는 과정에서 가장 바깥의 hull을 제거하면서 이상점과 잡음이 제거 될 수 있다. 가중 평균의 경우, 가중 값으로 방법1의 경우 hull의 면적 넓이의 역수로 하기 때문에 바깥의 큰 hull의 경우 더 적은 가중 값을 가지게 되고 이는 추정하는 방법에서 더 적은 역할을 함을 알 수 있다. 따라서 변동성이 큰 데이터가 더 적은 가중 값으로 추정에 기여하게 되고 변동성이 큰 데이터로 인한 영향을 줄여줄 수 있다. 또한 Convex hull peeling 방법을 살펴 보면 hull 중에 잡음이 섞여 있더라도 고유벡터의 추정을 잘 할 수 있게 함을 알 수 있다. 이상점같이 전체 분포에서 완전히 벗어난 데이터의 경우 hull의 바깥쪽에 있을 가능성이 크다. 이 경우 hull의 제거나 적은 가중 값으로 영향을 줄일 수 있지만 바깥이 아닌 hull 안쪽에서 생성된 잡음은 제거 될 수가 없다. 그러나 이런 경우에도 Convex hull peeling을 이용한 방법1이 고유벡터를 추정하는데 효과적임을 Figure3.4 살펴 볼 수 있다.

Figure3.4은 Simulation 2에서 T 분포 80개의 점과 잡음 20개인 100개의 데이터를 Convex hull peeling을 하고 hull 별로 그린 그림이다. T분포에서 생성된 점은 검은색으로 잡음은 빨간색으로 표시하였다. 먼저 앞에서 언급했듯이 hull의 번호가 커지고 바깥쪽으로 감에 따라 hull을 구성하는 점은 잡음의 비율이 더 많아 진다. 그리고 중간 깊이에 해당하는 hull에서는 8번, 10번, 11번 hull에서는 잡음이 섞여 있지만 잡음을 포함한 hull의 모양이 전체 분포 모양을 형성하는데 오히려 도움이 되고 있다. 빨간색 표시가 없을 경우 어느 것이 잡음이고 데이터인지 알 수 없을 정도이다. 소수의 잡음이 전체적인 분포의 모양을 유지하는데 도움이 되는 현상은 hull의 번호가 커지면서 hull을 구성하는 점 중에 잡음의 비율이 많아지면서 사라진다. Convex hull peeling은 주 분포의 분포 범위를 넘어선 이상점에 대해서는 바깥의 hull에 속하게 해서 적은 가중 값을 갖거나 아예 제거되는 식으로 조정되고 주 분포 범위 안의

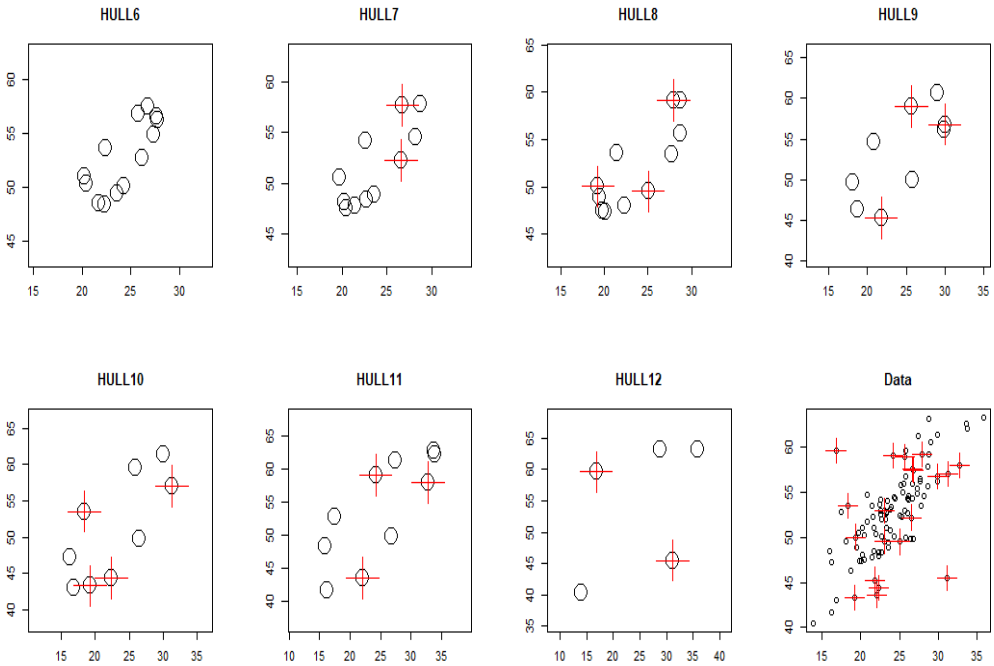

Figure 3.4: 전체적인 잡음이 있는 경우 hull별 잡음점

hull에서는 주 분포의 분포 방향을 형성하는데 도움이 되는 방향으로 hull이 형성되고 있었다.

Chapter 4

다변량에서의 적용

다차원 자료는 데이터들의 차원이 적어도 3이상일때를 가르킨다. 차원이 올라갈수록 분석이 쉽지 않아 진다. 다차원에서는 상상하지 못한 여러 성질들이 성립하지 않게 되기 때문이다. Convex hull peeling을 이용한 방법1의 경우, 2차원에 한에 hull별 분산행렬을 만들고 면적을 이용하여 가중 값을 설정하였다. 하지만 다차원으로 차원을 확장하게 되면 문제가 발생하게 된다. 어떤 차원이 되든지 간에 hull별 분산행렬을 만드는 데는 문제가 없지만 가중 값을 지정하는 데에 문제가 있게 된다. 먼저 다차원에서는 면적의 개념이 성립하지 않게 되기 때문이다. 3차원에서는 면적을 부피로 생각하면 되지만 그 이상의 차원에서는 적절한 면적을 고르기 쉽지 않기 때문이다. 이에 대한 해결 방법으로 앞서 소개 했듯이 hull별 부피 대신 고윳값을 이용하는 방법을 생각 할 수 있다. 그렇지만 이 경우 비례상수열의 단조증가 성이 보장되지 않는다. 또한 2,3차원 데이터를 Convex hull peeling을 했을 때 가장 바깥에 있는 점들은 첫 번째 성분과 두 번째 성분의 관점에서 모두 이상점에 속함을 알 수 있다. 하지만 다차원으로 확장하게 되면 가장 바깥에 있는 점은 모든 성분의 관점에서 이상점이라는 보장을 할 수 없기 때문이다. 이렇듯 낮은 2,3차원에서는 여러

방면에서 차원의 성질이 단순하기 때문에 데이터의 특성을 파악하기 쉽지만 다차원으로 확장하게 되면서 단번에 파악하기 어려워진다. 실제로 고유벡터를 추정하기 위해 다차원 분산행렬에 Convex hull peeling을 이용하게 된다면 오차의 차이가 크게 차이 나지 않게 된다.다차원 자료를 알기 위해 다차원 분산행렬의 성질을 살펴보면, 만약 4차원 자료의 분산행렬은 다음과 같이 나타낼 수 있다.

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{pmatrix}$$

σ_{ij}는 i번째와 j번째의 공분산을 나타내고 σ_{ii}는 i번째의 분산을 나타낸다

위 분산행렬 Σ를 따르는 4차원 데이터 N개가 있고 그 중 k번째 데이터 x_k를 다음과 같이 쓸 수 있다

$$x_k = (x_{1k}, x_{2k}, x_{3k}, x_{4k}) \quad k = 1 \ldots k$$

N개의 데이터를 이용하여 구한 표본 분산행렬($\hat{\Sigma}$)는 다음과 같다.

$$\hat{\Sigma} = \begin{pmatrix} \hat{\sigma}_{11} & \hat{\sigma}_{12} & \hat{\sigma}_{13} & \hat{\sigma}_{14} \\ \hat{\sigma}_{21} & \hat{\sigma}_{22} & \hat{\sigma}_{23} & \hat{\sigma}_{24} \\ \hat{\sigma}_{31} & \hat{\sigma}_{32} & \hat{\sigma}_{33} & \hat{\sigma}_{34} \\ \hat{\sigma}_{41} & \hat{\sigma}_{42} & \hat{\sigma}_{43} & \hat{\sigma}_{44} \end{pmatrix}$$

이때 $\hat{\sigma}_{ij} = \frac{1}{N}\sum_{k=1}^{N}\left(x_{ki} - \bar{x}_i\right)\left(x_{kj} - \bar{x}_j\right)$으로 주어진다. $\hat{\sigma}_{ij}$ 처럼 i번째 성분과 j번째 성분의 공분산을 추정할 때는 (x_{ki}, x_{kj}) 만 이용 , 벡터의 i, j성분인 2차원 데이터만 쓰인다는 것을 알 수 있다.표본 분산행렬의 모든 성분의 추정에 2차원 데이터만 쓴다는 것은 4차원 보다 큰 차원에서 분산행렬을 만들 때도

마찬가지 이다. 4차원 데이터에서 $1,3$성분을 Convex hull peeling을 이용하여 추정한 가중평균 분산행렬을 $\hat{\Sigma}_{(1,3)}$이라고 하면 위에서 언급한 대로 원래 분산행렬의 상수배가 됨을 기대 할 수 있다.

$$\tilde{\Sigma}_{(1,3)} = \begin{pmatrix} \tilde{\sigma}_{11} & 0 & \tilde{\sigma}_{13} & 0 \\ 0 & 0 & 0 & 0 \\ \tilde{\sigma}_{31} & 0 & \tilde{\sigma}_{33} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \simeq k_{(1,3)}\Sigma_{(1,3)} = k_{(1,3)} \begin{pmatrix} \sigma_{11} & 0 & \sigma_{13} & 0 \\ 0 & 0 & 0 & 0 \\ \sigma_{31} & 0 & \sigma_{33} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

이렇게 다차원 변수를 한 쌍씩 짝짓게 되면 총 ${}_4C_2$개, 6개의 쌍이 나오게 되며 $\tilde{\Sigma}_{(1,2)}, \tilde{\Sigma}_{(2,3)}, \tilde{\Sigma}_{(3,4)}, \tilde{\Sigma}_{(1,4)}, \tilde{\Sigma}_{(2,4)}, \tilde{\Sigma}_{(1,3)}$ 로 표현 할 수 있으며 분산행렬 Σ의 어떤 상수 배에 가까움을 기대 할 수 있다. 다차원자료를 2개씩 묶어서 부분적인 분산행렬들을 추정하여 모든 성분의 값을 추정 할 수 있지만 알 수 없는 상수 배만큼의 차이 난다. 6개의 부분 분산행렬에서 $(1,3)$성분과 $(1,2)$성분의 분산행렬은 다음과 같이 표현 된다

$$\tilde{\Sigma}_{(1,3)} = \begin{pmatrix} \tilde{\sigma}_{11} & 0 & \tilde{\sigma}_{13} & 0 \\ 0 & 0 & 0 & 0 \\ \tilde{\sigma}_{31} & 0 & \tilde{\sigma}_{33} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \simeq k_{(1,3)} \begin{pmatrix} \sigma_{11} & 0 & \sigma_{13} & 0 \\ 0 & 0 & 0 & 0 \\ \sigma_{31} & 0 & \sigma_{33} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\tilde{\Sigma}_{(1,2)} = \begin{pmatrix} \tilde{\eta}_{11} & \tilde{\eta}_{12} & 0 & 0 \\ \tilde{\eta}_{21} & \tilde{\eta}_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \simeq k_{(1,2)} \begin{pmatrix} \sigma_{11} & \sigma_{12} & 0 & 0 \\ \sigma_{21} & \sigma_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$\tilde{\Sigma}_{(1,3)}$의 1성분과 $\tilde{\Sigma}_{(1,2)}$의 1성분은 서로 겹치게 된다 그러나 실제적으로 각각 다른 Convex hull peeling의 hull을 사용함으로 서로 다르다. 구체적으로 두

값이 차이가 나는 이유는 크게 두 가지로 첫 번째 추정을 할 때 오차로 인해 정확한 값을 구할 수 없는 것과 두 번째로 비례 상수 $k_{(1,3)}$과 $k_{(1,2)}$가 다르기 때문이다. 따라서 대각성분 값이 다르기 때문에 Convex hull peeling으로 두 성분씩 본 가중평균 분산행렬들을 단순히 합하면 안 된다. 그러나 각 분산행렬의 대각성분에 있는 분산 값들을 이용하여 최대한 비슷하게 만들어 줄 수 있다. $\tilde{\Sigma}_{(1,3)}$과 $\tilde{\Sigma}_{(1,2)}$의 대각 성분 중 겹치는 1성분을 적당한 상수 $c_{(1,2)}$과 $c_{(1,3)}$을 곱해서 같게 만들어 같게 만들 수 있다. 그러면 두 부분 행렬의 첫 번째 대각성분값은 서로 같게 되어 하나의 행렬로 합칠 수 있음으로 4차원 데이터에서 6개의 부분행렬을 대각성분들을 비교하여 합칠 수 있다.

먼저 4차원 데이터에서 $\tilde{\Sigma}_{(1,2)}, \tilde{\Sigma}_{(1,3)}, \tilde{\Sigma}_{(1,4)}$ 3개의 부분행렬들은 1성분의 분산 값을 상수 배 하여 서로 같게 만들 수 있고 서로 맞추어 줬다고 하면 3개의 부분행렬은 다음과 같이 나온다.

$$\tilde{\Sigma}_{(1,2)} = \begin{pmatrix} \tilde{\sigma}_{11} & \tilde{\sigma}_{12} & 0 & 0 \\ \tilde{\sigma}_{21} & \tilde{\sigma}_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tilde{\Sigma}_{(1,3)} = \begin{pmatrix} \tilde{\sigma}_{11} & 0 & \tilde{\sigma}_{13} & 0 \\ 0 & 0 & 0 & 0 \\ \tilde{\sigma}_{31} & 0 & \tilde{\sigma}_{33} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tilde{\Sigma}_{(1,4)} = \begin{pmatrix} \tilde{\sigma}_{11} & 0 & 0 & \tilde{\sigma}_{14} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \tilde{\sigma}_{41} & 0 & 0 & \tilde{\sigma}_{44} \end{pmatrix},$$

$$\tilde{\Sigma}_{(1,2)} + \tilde{\Sigma}_{(1,3)} + \tilde{\Sigma}_{(1,4)} = \begin{pmatrix} \tilde{\sigma}_{11} & \tilde{\sigma}_{12} & \tilde{\sigma}_{13} & \tilde{\sigma}_{14} \\ \tilde{\sigma}_{21} & \tilde{\sigma}_{22} & 0 & 0 \\ \tilde{\sigma}_{31} & 0 & \tilde{\sigma}_{33} & 0 \\ \tilde{\sigma}_{41} & 0 & 0 & \tilde{\sigma}_{44} \end{pmatrix}$$

먼저 상수 배로 첫 번째 분산 값을 맞춘 세 행렬을 더하면 1성분과 관련된 분산행렬의 성분 값들이 모두 채워지게 된다. 다음으로 2성분과 관련된 부분행렬 $\tilde{\Sigma}_{(2,3)}, \tilde{\Sigma}_{(2,4)}$들의 2번째 분산 값을 상수배하여 $\tilde{\sigma}_{22}$ 와 맞춰 주고 더하면 다음과 같이 된다

$$\tilde{\Sigma}_{(1,2)} + \tilde{\Sigma}_{(1,3)} + \tilde{\Sigma}_{(1,4)} + \tilde{\Sigma}_{(2,3)} + \tilde{\Sigma}_{(2,4)} = \begin{pmatrix} \tilde{\sigma}_{11} & \tilde{\sigma}_{12} & \tilde{\sigma}_{13} & \tilde{\sigma}_{14} \\ \tilde{\sigma}_{21} & \tilde{\sigma}_{22} & \tilde{\sigma}_{23} & \tilde{\sigma}_{24} \\ \tilde{\sigma}_{31} & \tilde{\sigma}_{32} & \tilde{\sigma}_{33} & 0 \\ \tilde{\sigma}_{41} & \tilde{\sigma}_{42} & 0 & \tilde{\sigma}_{44} \end{pmatrix}$$

다음으로 3성분과 관련된 부분행렬 $\tilde{\Sigma}_{(3,4)}$를 이용하여 위와 같이 상수배를 $\tilde{\sigma}_{(3,3)}$에 맞추고 더하면 모든 성분을 구할 수 있다.

$$\tilde{\Sigma}_{(1,2)} + \tilde{\Sigma}_{(1,3)} + \tilde{\Sigma}_{(1,4)} + \tilde{\Sigma}_{(2,3)} + \tilde{\Sigma}_{(2,4)} + \tilde{\Sigma}_{(3,3)} = \begin{pmatrix} \tilde{\sigma}_{11} & \tilde{\sigma}_{12} & \tilde{\sigma}_{13} & \tilde{\sigma}_{14} \\ \tilde{\sigma}_{21} & \tilde{\sigma}_{22} & \tilde{\sigma}_{23} & \tilde{\sigma}_{24} \\ \tilde{\sigma}_{31} & \tilde{\sigma}_{32} & \tilde{\sigma}_{33} & \tilde{\sigma}_{34} \\ \tilde{\sigma}_{41} & \tilde{\sigma}_{42} & \tilde{\sigma}_{43} & \tilde{\sigma}_{44} \end{pmatrix}$$

위와 같은 과정은 4차원보다 큰 차원에서도 항상 할 수 있다. 그러나 다차원으로 확장 할 때 위와 같은 방법은 하나의 예일 뿐이다. Convex hull peeling이 잘 적용되는 2차원에서 적용하기 위해 두 성분씩 묶어서 사용하였고 다차원으로 확장하기 위해 대각성분을 강제적으로 맞추었다. 만약 대각성분의 추정이 잘못되었을 경우 맞춘 결과가 전체적으로 잘 못 될 가능성이 있고 합치는 과정에서 다음과 같은 식을 사용하기 때문에 $\tilde{\Sigma}_{(1,2)} + \tilde{\Sigma}_{(1,3)} + \tilde{\Sigma}_{(1,4)}$이 애초에 잘못되었다면 연쇄적으로 추정이 잘못될 가능성이 있다. 또한 마지막 대각 성분은 생각하지 않은 문제가 있다. 때문에 위 같은 다차원 확장방법은 완전하지 않고 개선될 여지가 있다.

소개한 위의 방법을 이용하여 3차원 이상에서 부터 T분포 자유도 3 을 따르는 주 분포 80개와 전체 잡음 20개를 넣어 100개의 데이터를 생성하는 200번의 반복실험을 수행하였다.

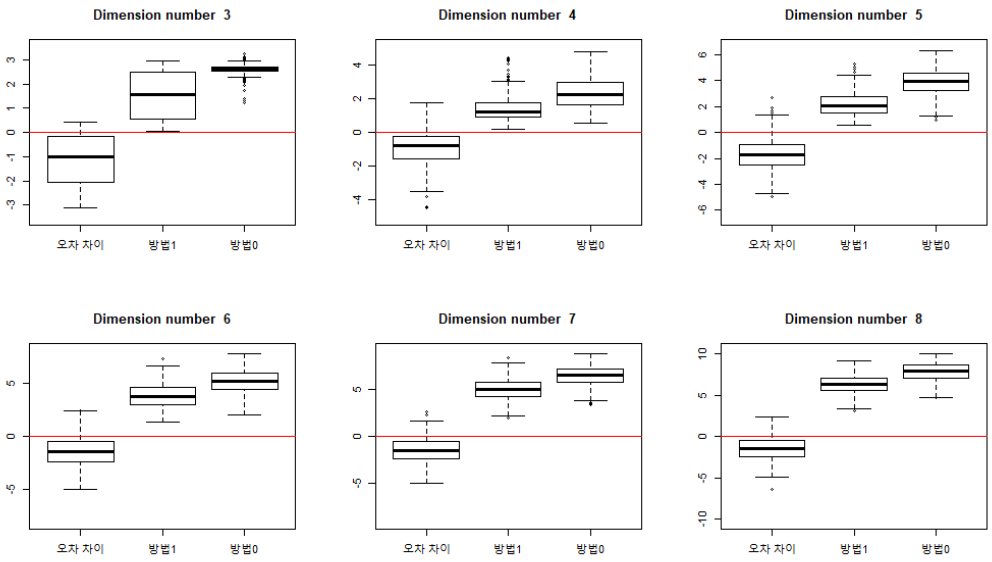

Figure 4.1: 다차원 확장

다차원 데이터를 2개씩 묶고 Convex hull peeling을 이용하는 것은 다음과 같은 특징이 있다 만약에 데이터의 점을 이상점이라고 생각하고 10% 를 제거 했을 경우 10%의 점은 데이터에서 사라지고 추정에 사용되지 못하게 된다. 그러나 2개씩 묶어서 Convex hull peeling을 사용 했을 경우 1,2 성분에서을 이상점으로 판단되어 제거 되었던 점이 2,3성분이나 3,4성분등 다른 성분을 중심으로 묶었을 때에는 사용 될 수 있다. 따라서 Convex hull peeling에서 사용되는 데이터들은 90%보다 높다고 생각 할 수 있으며 이는 최대한 많은 데이터를 사용하는 관점에서 장점이 될 수 있다.

Chapter 5

결론

Convex hull peeling은 데이터의 분류 뿐만 아니라 다른 성질을 살펴 보았다. 먼저 Convex hull peeling이 가지고 있는 분포 형태를 보존하는 특징을 살펴 보고 이용하여 고유벡터를 추정하는 두 가지 방법을 만들어 보았다. 제시된 방법들이 어느 정도로 추정을 하는 지 살펴 보기 위해 특정 분포를 가정한 상태에서 여러 가지 잡음의 상황을 제시하고 반복 실험을 통해 오차제곱 합을 관찰하였다. 그 결과 Convex hull peeling을 이용한 각각의 방법들은 잡음이 있을 경우 고유벡터를 추정하는 데에 있어 적률추정법보다 좋은 결과를 얻을 수 있음을 알 수 있었다. 그 이유는 Convex hull들이 바깥으로 갈수록 잡음의 수가 많아지는 점과 Convex hull peeling이 분포 형태를 유지하려는 쪽으로 데이터들을 분류하는 특징 때문이라고 생각 할 수 있었다. 다차원 확장에서 여러 차원을 두 개씩 묶어서 다차원 확장이 가능했고 모의실험결과 잡음에서도 좋은 결과를 내었지만 개선할 여지가 있는 방법이었다.

References

[1] Daniel Peña & Francisco J Prieto. *Multivariate Outlier Detection and Robust Covariance Matrix Estimation.* Technometrics, 43(3):286-310,2001

[2] N. A. Campbell. *Robust Procedures in Multivariate Analysis I: Robust Covariance Estimation.* Journal of the Royal Statistical Society. Series C, 1980,pp. 231-237

[3] P. B. Harrington and T. L. Isenhour. *Application of robust eigenvectors to the compression of infrared spectral libraries.* Anal. Chem., 1988, 60 (24), pp 2687–2692

[4] Bradley Efron. *The Convex Hull of a Random Set of Points.* Biometrika. Vol. 52, Dec., 1965, pp. 331-343

Abstract

Jisoo Choi
The Department of Statistics
The Graduate School
Seoul National University

Convex hull peeling is one of the methods for classifying multidimensional data. In this paper, we propose an eigenvector estimation method using Convex hull peeling. Through simulations with noise and anomalies, we examine the effects of the proposed method and finally consider how to extend it to the multidimensional. Simulation results show that the proposed method provides more effective estimates than the conventional eigenvector estimation method. In addition, it has been found out that even when extended to the 8-dimensional result, it is estimated effectively. However, in the course of multidimensional expansion, the method is not unique and can be improved and further studies are possible.

Keyword : *Convex hull peeling, roburst,eigen vector*
Student Number : 2017-20278