

Robust covariance estimation for partially observed functional data

2021-08-05

Covariance estimator

- Kernel function으로 weight을 주고 local version으로 covariance를 estimate해보려 했으나, weight을 어떻게 줘야할 지 계속 헤매고 있었고, 막상 식으로 적어보니 결국 loss를 Huber loss로 바꾼 Nadaraya-Watson estimator (Local constant smoothing)와 동일하여 local 방법은 중단했습니다...
- 2-step estimation으로 아래 논문에 나온 bivariate sandwich smoothing을 이용하여 2-step으로 covariance를 estimate하는 방법을 찾아보았습니다. (드라이브에 ref_paper에도 넣어두었습니다.)

Xiao, L., Li, Y., & Ruppert, D. (2013). Fast bivariate P-splines: the sandwich smoother. Journal of the Royal Statistical Society: SERIES B: Statistical Methodology, 577-599.

- 논문의 6장에 application으로 functional data에서 covariance를 estimate하는 부분이 있는데, 여기서는 sample covariance에 sandwich smoother를 적용하여 smoothed covariance를 estimate하는 부분이 언급되어 있고 이를 bivariate local linear smoother와 비교한 부분이 있습니다. (이 방법을 계속 하고 있었는데 여기서는 GCV를 사용하여 smoothing parameter를 결정하기 때문에 따로 CV를 해줄 필요가 없습니다.)
- `refund::fbps()` 함수로 구현되어 있습니다.

Simulation results

- 우선 outlier 2인 경우에 50번 반복한 결과만 확인하였으며, 에러나는 seed가 조금 바뀌면서 기존 결과와 약간 다르지만 거의 비슷합니다.
- 파란 부분이 sandwich smoother를 사용한 2-step 방법으로 한 결과입니다. (penalized spline 기반이라 knots를 결정해주긴 해야하지만, 이는 우선 default 값을 사용했고 smoothing parameter는 GCV rule에 의해 결정되었습니다.)

1. Delaigle et al.(2020) setting

Method	PVE	Reconstruction	Completion	Eigenfunction
Yao	0.94	0.65 (0.52)	1.11 (0.68)	1.23 (0.60)
Huber	0.82	0.10 (0.04)	0.37 (0.17)	0.08 (0.11)
Kraus			1.88 (0.90)	
Kraus-M			1.03 (0.41)	
Kraus-M(sm)			0.32 (0.27)	
Boente	1.00	0.21 (0.55)	0.57 (1.42)	0.14 (0.05)
M-est	0.76	0.20 (0.40)	0.66 (1.42)	0.19 (0.12)
M-est-noise	0.81	0.07 (0.02)	0.23 (0.09)	0.19 (0.12)
M-est(smooth)	0.94	0.13 (0.20)	0.54 (0.97)	0.16 (0.11)
M-est(smooth)-noise	0.96	0.05 (0.02)	0.18 (0.08)	0.16 (0.11)

2. Kraus(2015) setting

Method	PVE	Reconstruction	Completion	Eigenfunction
Yao	0.97	0.30 (0.16)	0.38 (0.19)	1.64 (0.24)
Huber	0.85	0.10 (0.06)	0.27 (0.10)	1.43 (0.10)
Kraus			0.46 (0.27)	
Kraus-M			0.16 (0.06)	
Kraus-M(sm)			0.06 (0.02)	
Boente	1.00	0.07 (0.05)	0.23 (0.20)	0.79 (0.11)
M-est	0.74	0.04 (0.02)	0.07 (0.04)	0.65 (0.12)
M-est-noise	0.80	0.03 (0.00)	0.05 (0.01)	0.65 (0.12)
M-est(smooth)	0.93	0.03 (0.01)	0.09 (0.04)	0.34 (0.15)
M-est(smooth)-noise	0.97	0.01 (0.00)	0.04 (0.01)	0.34 (0.15)

문제점...?

- noise variance에서 trimming이 아닌 quantile 값으로 대체하는 방법(winsorization)을 고려해보려 하였으나, A_0 에서 winsorize한 cut-off하여 다시 A_1 과 B 에 적용하는 것이 불가능한 것 같습니다.
- Trimming으로는 A_0 term에서 값이 큰 curve들을 제외해버리고, 이 curve index들을 A_1 과 B 계산에서도 제외함으로서 계산이 가능하였는데, winsorization으로는 해당 index에 해당하는 A_1 , B 의 값을 quantile로 지정해주는 것도 불가능하기 때문에, 이를 지정해주는 것이 의미가 없어 보입니다...
- 추가로 Lin & Wang (2020) 논문에서 noise variance의 bandwidth는 snippet 경우의 convergence rate을 통해 empirical bandwidth를 제안하고 있고, 코드도 현재는 그렇게 되어있습니다. 이 부분도 수정이 필요해 보이는데 좀 더 확인해보겠습니다...