Synthetic Differences in Differences Simulations

Joshua Derenski

Simulations

Parameters

- Number of Ls: 1
- Draws per L: 100
- Number of Units: 100
- Number of Control Units: 98
- Number of Times: 200
- Number of pre-treatment Times: 198
- Rank of L: 10
- Autocorrelation Parameter: 0.6
- True Effect Size: 10
- Error Type: gaussian
- Error Variance (if Gaussian error): 1
- Degrees of freedom (if t-error): 5
- Exchangable: FALSE
- Penalized: TRUE
- Rank Estimation Method: threshold
- Scaling for L: 1

Simulation Description

In this simulation we sample 1 signal matricies $L \in \mathbb{R}^{100\times 200}$ of rank 10, and for each L we sample 100 Ys such that for each Y $E(Y_{ij}) = L_{ij} + \tau W_{ij}$ where W is known, and $\tau = 10$ (in the case of the t-distribution, we ensure the median of each cell is $L_{ij} + \tau W_{ij}$). We generate L so that the rows and columns are not exchangable, and we estimate the weights in SDID using penalized regressions. For estimating the rank of L we are using a threshold method. From each corresponding pair of (L, Y) an estimate of τ is generated via method i, which we call $\hat{\tau}_{(L,Y),i}$. To evaluate the performance of method i we calculate the following RMSE:

$$RMSE_i = \sqrt{\frac{1}{100}\sum_{(L,Y)} (\hat{\tau}_{(L,Y),i} - \tau)^2}$$

Our Method vs Competitors, Fixed Parameters

Results

Signal to Noise Ratio

```
## [1] 17.44909
```

Rmse for DID

```
## [1] 6.832237
```

Se for Rmse for DID

```
## [1] 0
```

Rmse for SC

```
## [1] 2.877996
```

Se for Rmse for SC

```
## [1] 0
```

Rmse for our Method (Explicit Tau)

```
## [1] 0.7457445
```

Se for Rmse for our Method (Explicit Tau)

```
## [1] 0
```

Rmse for SDID

```
## [1] 1.554131
```

Se for Rmse for SDID

```
## [1] 0
```

RMSE For Our Method (Not Explicit Tau)

```
## [1] 0.8741724
```

Se for Rmse for Our Method (Not Explicit Tau)

```
## [1] 0
```

RMSE For Oracle (Perfect L)

```
## [1] 0.5660541
```

RMSE For Oracle (Perfect L)

```
## [1] 0
```

Matrix Bias vs Reduction in Variance due to Averaging

For more general designs of W (like the block design scheme considered here) we allow a block in the bottom right hand corner of W to be non-zero. When implementing our method, we have two competing effects on estimation:

- The bias that's introduced by making more of the Y_{ij}s zero.
- The help we get with estimating τ by being able to average over cells (because we assume *tau*) is the same for all units and times.

It would appear that accurracy increases for estimating τ to a point, and then decreases when the bias introduced by replacement of cells with 0 in Y becomes too great.

Influence of N_0 on Performance

Influence of ρ on Performance

Influence of τ on Performance

Influence of Rank Error on Performance