Swiss Institute
of Banking and Finance

University of St.Gallen

Spatial Regression Models: When Small Data Sets Lead to Big Data Problems

Zeno Adams

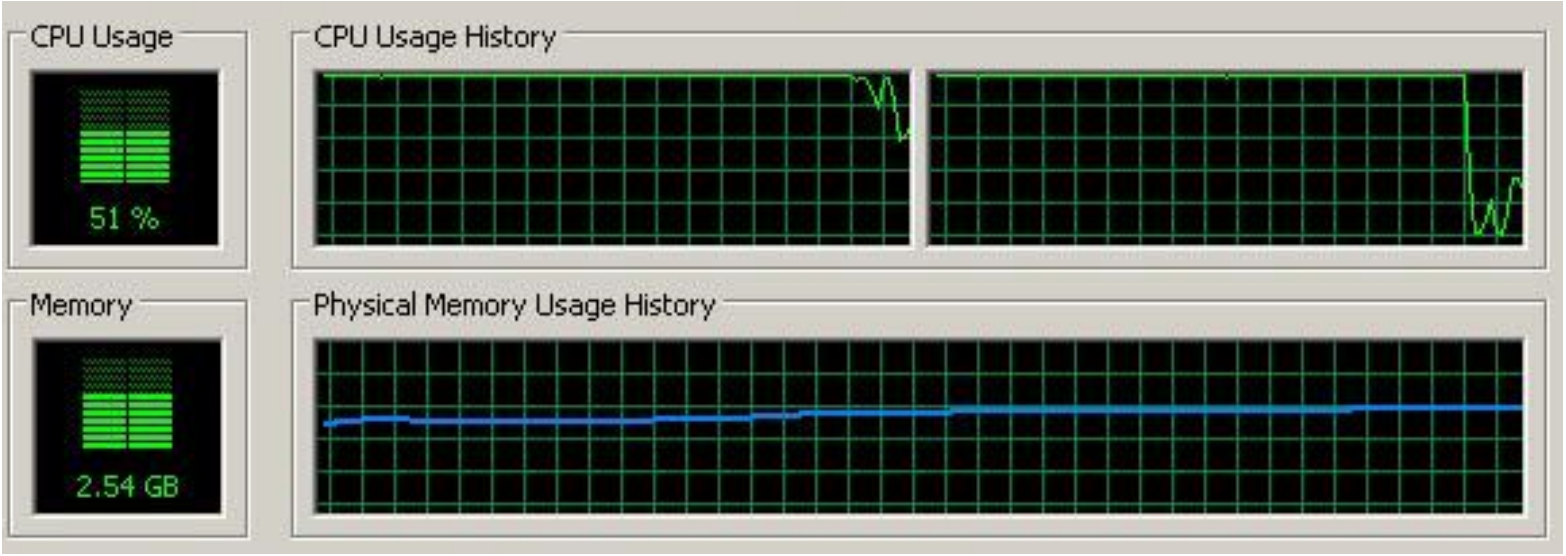
September 27, 2016

Summary of Findings

1. Use sparse matrices
2. Reduce multivariate optimization problem to univariate problem
3. Use precomputed terms that do not update in iterations
4. Use taylor series expansion to reduce matrix size

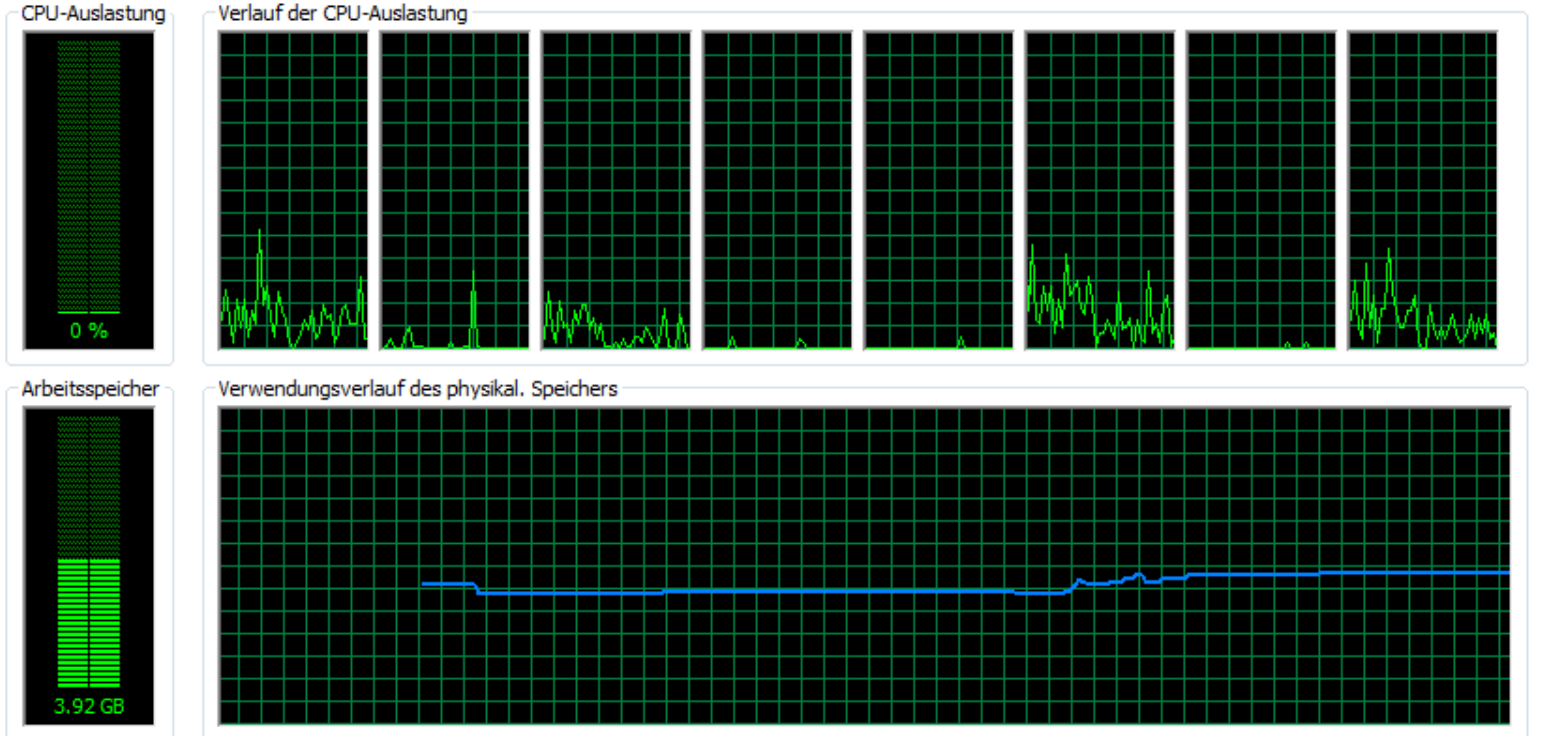
Spatial Data and Data Handling

```
Error: cannot allocate vector of size 21.3 Gb
In addition: Warning messages:
1: In diag(N * T) :
  Reached total allocation of 4089Mb: see help(memory.size)
```



Spatial Data and Data Handling

```
Error: cannot allocate vector of size 18.6 Gb
In addition: Warning messages:
1: In diag(N * T) :
  Reached total allocation of 8103Mb: see help(memory.size)
```



Spatial Data and Data Handling

106 Swiss MS Regions

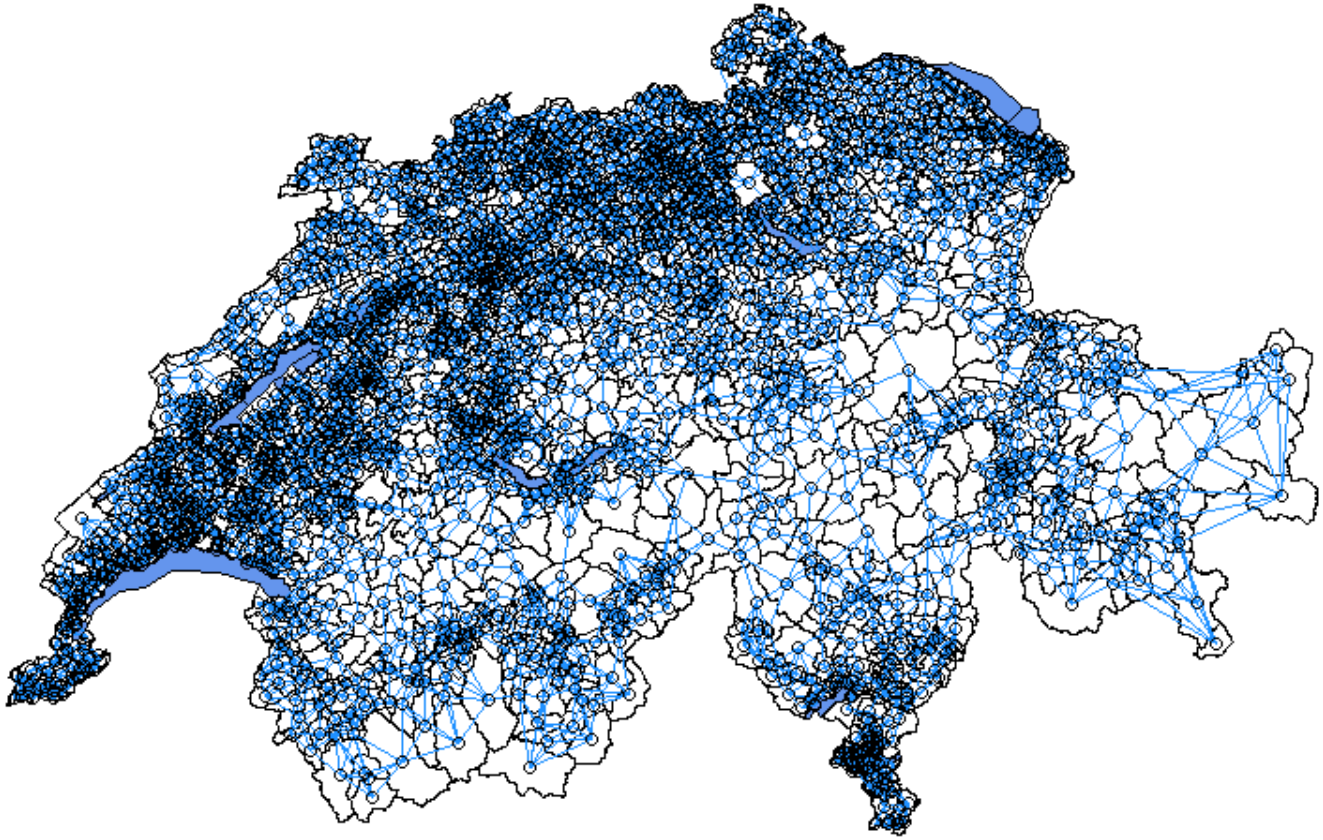
Spatial Data and Data Handling

An example of a Weight Matrix

	Zü	*Gl*	*Ft*	*Lt*	*Wi*
Zü	0	1/6	1/6	1/6	0
Gl	1/6	0	1/6	0	1/6
Ft	1/4	1/4	0	1/4	0
Lt	1/3	0	1/3	0	0
Wi	0	1/4	0	0	0

Spatial Data and Data Handling

2428 Swiss Municipalities



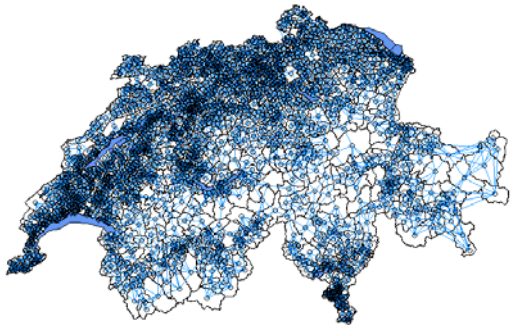
Spatial Data and Data Handling

106 x 106 Weight Matrix:

- 11,236 entries
- 0.1 MB memory

2428 x 2428 Weight Matrix:

- 5,895,184 entries
- 45.1 MB memory



Spatial Data and Data Handling

Panel version of the 2428 x 2428 Swiss Municipality Weight Matrix:

- N = 2428 municipalities
- T = 20 years
- NT x NT weight matrix $(W_N \otimes I_T)$. 2,358,073,600 entries
-

```
> W <- kronecker(W,I.T)
Error: cannot allocate vector of size 17.6 Gb
```

	Zü	Gl	Ft	Lt	Wi
Zü	0	1/6	1/6	1/6	0
Gl	1/6	0	1/6	0	1/6
Ft	1/4	1/4	0	1/4	0
Lt	1/3	0	1/3	0	0
Wi	0	1/4	0	0	0

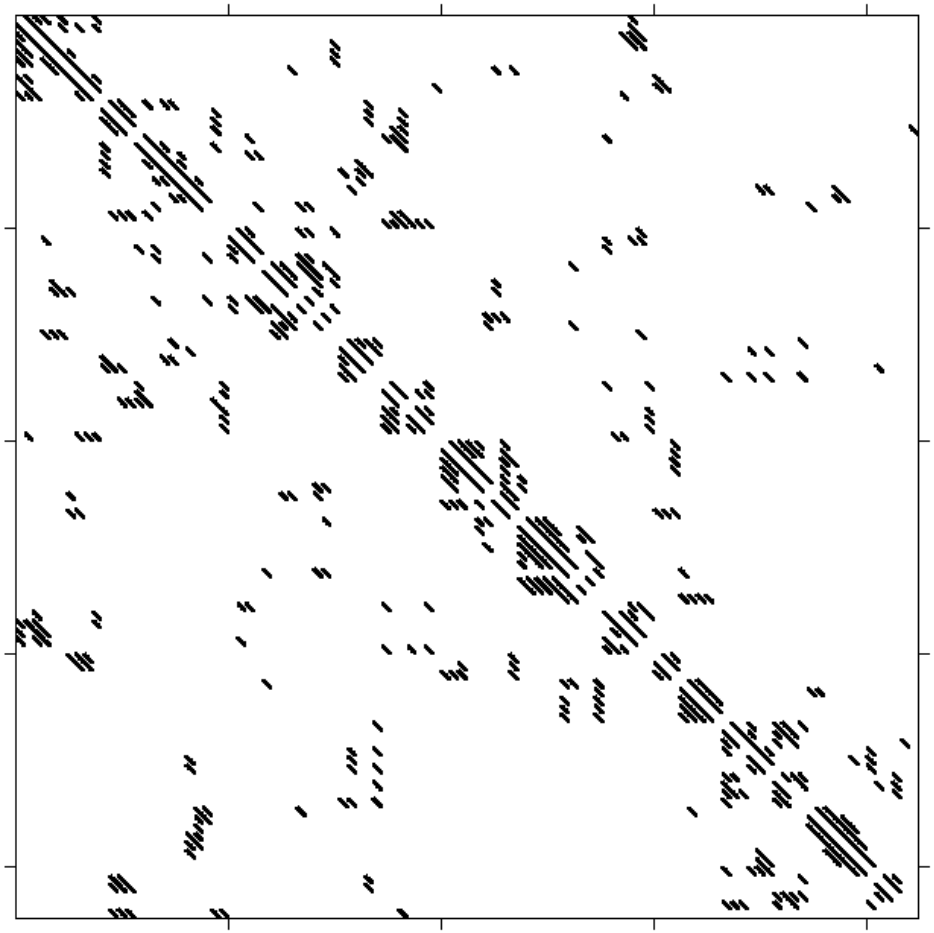
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \vdots \\ 0 & 0 & 0 & \cdots & \ddots \end{bmatrix}$$

Sparse Matrix Tools in R

Use Sparse Matrix representation

- reduces memory burden for sparse matrices

$$\left(W_N \otimes I_T\right) =$$



Sparse Matrix Tools in R

Sparse Matrix representation:

- Example of $\left(W_N \otimes I_T\right)$ in sparse matrix format for 106 MS regions

```
> library(Matrix)
> T <- 20 ; N <- 106
> I.T = diag(T)
> A <- kronecker(W,I.T)
> format(object.size(A), units = "Mb")
[1] "34.3 Mb"
> A2 <- as(A, "sparseMatrix")
> format(object.size(A2), units = "Mb")
[1] "0.2 Mb"
```

$\left(W_N \otimes I_T\right)$

```
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
0.1666667 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.1666667 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.1666667 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.1666667 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000 0.1666667
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
```

$sparse\left(W_N \otimes I_T\right)$

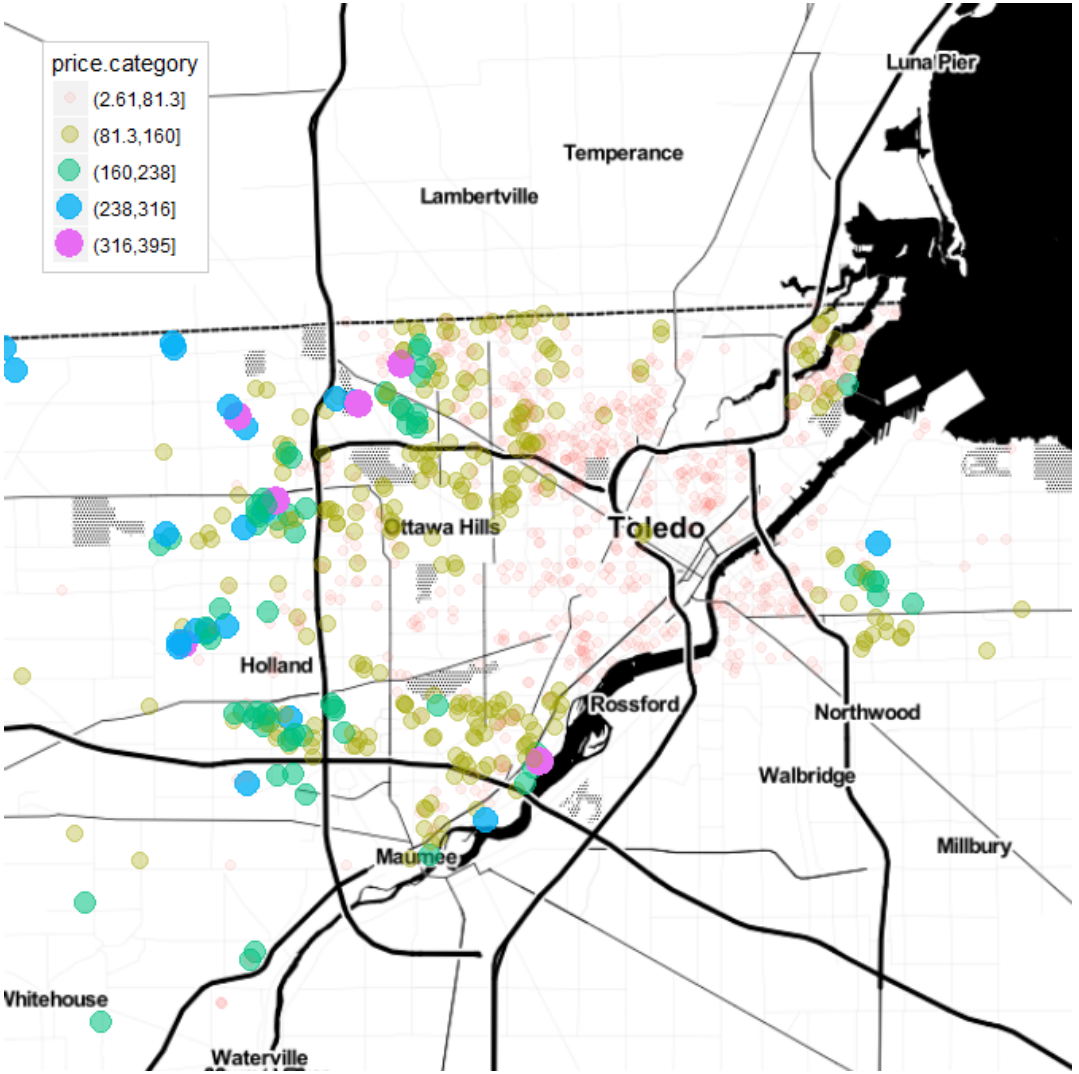
```
.         .         .         .         .
.         .         .         .         .
.         .         .         .         .
.         .         .         .         .
0.1666667 .         .         .         .
.         0.1666667 .         .         .
.         .         0.1666667 .         .
.         .         .         0.1666667 .
.         .         .         .         0.1666667
.         .         .         .         .
.         .         .         .         .
```

Example: Hedonic House Price Model

Lucas County, Oh, Data Set, 1998

Simple OLS regression

Dependent Variable: House Prices	
Age	-756***
# Bedrooms	-2,402
# Bathrooms	12,194***
Total Living Area	480***
Obs.	1,000
Adj.R-squared	0.66



Spatial Autoregressive Model

SAR Model:

$$Y = \rho WY + X\beta + e$$

Likelihood Function:

$$e = Y - \rho WY - X\beta$$

$$\ln \mathcal{L} = \ln \left| I_n - \rho W \right| - \frac{n}{2} \ln \left(2\pi\sigma^2 \right) - \frac{e'e}{2\sigma^2}$$

```
Y <- as.matrix(dat2$price/1000)
colnames(Y) <- "HP"
X <- cbind(1,dat2$age, dat2$beds, dat2$baths, dat2$tla)
colnames(X) <- c("intercept","age","beds","baths","tla")

normal.lik1 <- function(theta,y,X) {
beta <- theta[1:5]
sigma <- theta[6]
rho <- theta[7]
n <- nrow(X)
In <- diag(n)
# log likelihood:
e <- y-rho*W%*%y-X%*%beta
logl <- log(det(In - rho*W)) - (n/2)*log(2*pi*sigma^2) - crossprod(e)/(2*sigma^2)
return(-logl)
}

fit2 <- optim(c(1,1,1,1,1,1,0.5),normal.lik1,y = Y, X = X, method = "BFGS",
control = list(maxit = 1000, trace = TRUE))
```

Spatial Autoregressive Model

Problems with the full maximum likelihood model:

- Variables need to be of similar scale `Y <- as.matrix(dat2$price/1000)`
- Starting values `fit2 <- optim(c(1,1,1,1,1,1,0.5)`
- Rather slow:

```
> system.time(optim(c(1,1,1,1,1,1,0.5),normal.lik1,y = Y, X = X, method = "BFGS",
+ control = list(maxit = 1000, trace = TRUE)))
initial  value 13491179.595561
iter  10 value 15959.519480
iter  20 value 8208.581732
iter  30 value 4900.658879
iter  40 value 4868.730939
final  value 4868.465631
converged
       User      System verstrichen
      99.92        3.40      103.35
```

Adressing Drawbacks of the SAR Model

Change 1

Reduce multivariate optimization problem to univariate problem

$$Y = \rho WY + X\beta + e$$

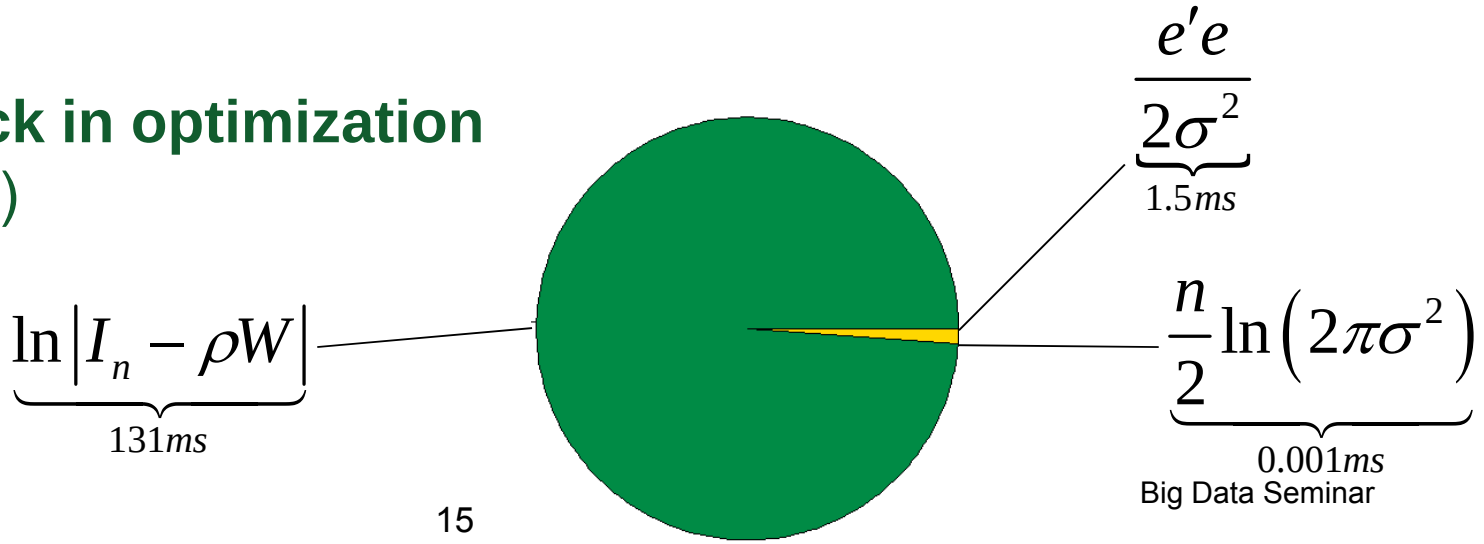
$$\underbrace{\left(I_n - \rho W\right)Y}_{Y^*} = X\beta + e$$

➡ Estimate $Y^* = X\beta + e$ by OLS

Change 2

Remove Bottleneck in optimization

(Core i7-4770 CPU)



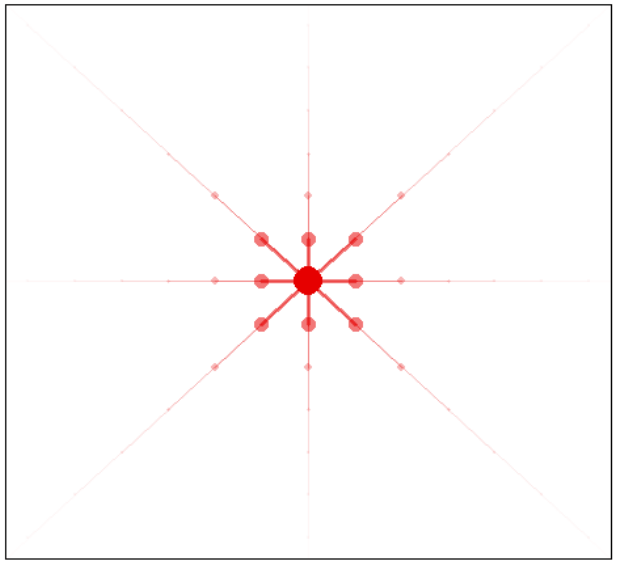
The MESS Model

The MESS specification:

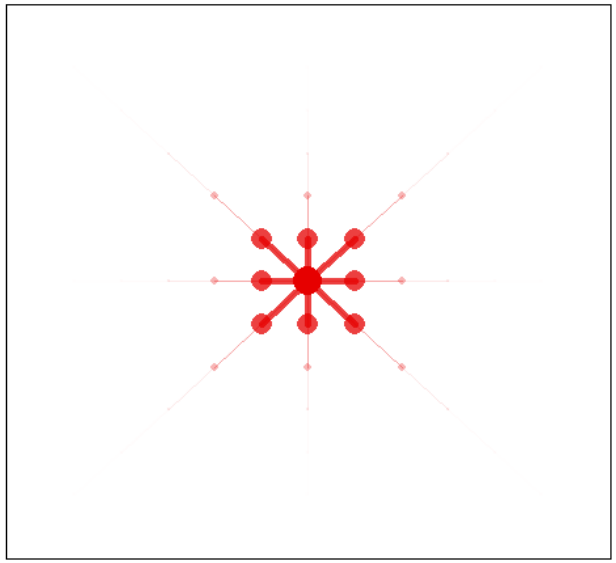
Instead of $\left(I_n - \rho W\right)Y = X\beta + e$

Now $SY = X\beta + e$ $\qquad S = e^{\alpha W} = \sum_{i=0}^{\infty} \frac{\alpha^i W^i}{i!}$

Diffusion of SAR Modell, ρ= 0.5276



Diffusion of MESS Modell, α= -0.75



The MESS Model

Example: Taylor Series Expansion

$$\exp(x) \approx \sum_{i=0}^{10} \frac{x^i}{i!}$$

x	exp(x)	Taylor series
0	1.00	1.00
1	2.72	2.72
2	7.39	7.39
3	20.09	20.08
4	54.60	54.44
5	148.41	146.38

$$W = \begin{bmatrix} 2 & 1 \\ 3 & -1 \end{bmatrix}$$

$$e^W \approx \sum_{i=0}^{10} \frac{W^i}{i!} = \begin{bmatrix} 13.51 & 3.52 \\ 10.56 & 2.95 \end{bmatrix}$$

The MESS Model

Concentrated Likelihood Function:

$$\ln \mathcal{L} = \kappa + \ln |S| - \left(n/2\right) \ln \left(e'e\right)$$

Trick: the log determinant disappears

$$\left|S\right| = \left|e^{\alpha W}\right| = e^{tr\left(\alpha W\right)} = e^{0} = 1$$

Estimating $e'e$ using the residual maker matrix M:

$$MY = \left(I_n - X\left(X'X\right)^{-1} X'\right) Y = Y - X\beta = e$$

$$e'e = \left(MSY\right)' \left(MSY\right) = Y'S'M'MSY = Y'S'MSY$$

The MESS Model

Problem: multiplication of large matrices

$$\ln \mathcal{L} = -\frac{n}{2} \ln \left(\underbrace{Y'}_{1xn} \underbrace{S'}_{nxn} \underbrace{M}_{nxn} \underbrace{S}_{nxn} \underbrace{Y}_{nx1} \right)$$

Speedup 1: decompose estimation to get precomputed values

- Precomputed matrix Q
- Vector with updated values $v = f(\alpha)$

$$\ln \mathcal{L} = -\frac{n}{2} \ln \left(v' Q v \right)$$

The MESS Model

Speedup 2: Reduce large nxn matrices to small 12x12 matrices

$$v' = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{11} \end{bmatrix}$$

$$SY = e^{\alpha W} Y \approx \sum_{i=0}^{11} \frac{\alpha^i W^i Y}{i!}$$

$$\tilde{Y} = \begin{bmatrix} W^0 Y & W^1 Y & W^2 Y & \cdots & W^{11} Y \end{bmatrix}$$

$$G = \begin{bmatrix} 1/0! & & & & \\ 0 & 1/1! & 0 & \cdots & 0 \\ 0 & 0 & 1/2! & 0 & 0 \\ \vdots & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1/11! \end{bmatrix}$$

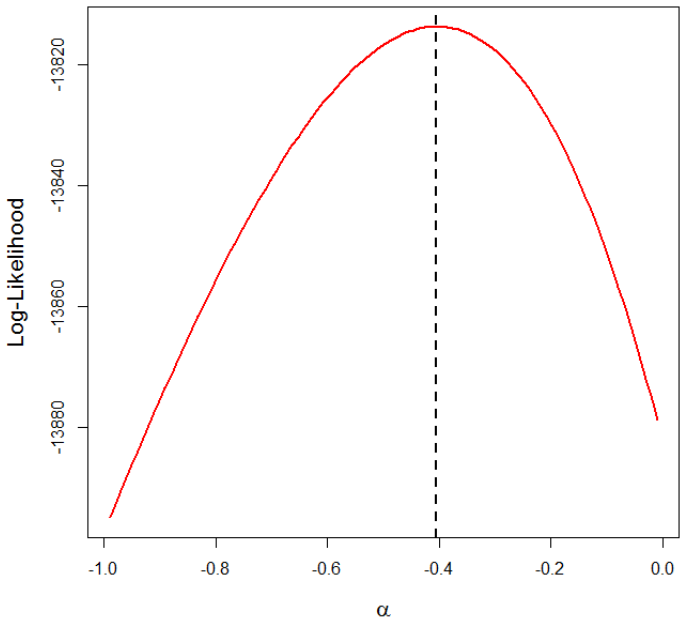
$$\underbrace{Q}_{12x12} = \underbrace{G}_{12x12} \underbrace{\tilde{Y}'}_{12xn} \underbrace{M}_{nxn} \underbrace{\tilde{Y}}_{nx12} \underbrace{G}_{12x12}$$

The MESS Model

Estimation time in MESS: 3.8 milliseconds

```
# estimate the spatial dependence parameter alpha using concentrated ML:
# Procedure described in Ch.9 of LeSage and Pace (2009)
library(expm)
y <- dat2$price
q <- 12
y.tilde <-  sapply(0:(q-1), function(x) (W%^%x)%*%y) # (9.5)
G <- matrix(0,q,q)
diag(G) <- sapply(0:(q-1), function(x) 1/factorial(x)) # (9.6)
X <- as.matrix(cbind(1, dat2$age, dat2$beds, dat2$baths, dat2$tla))
colnames(X) <- c("intercept","age","beds","baths","tla")
In <- diag(n)
M <- In - X%*%solve(t(X)%*%X)%*%t(X) # Residual Maker Matrix
Q <- G%*%(t(y.tilde)%*%M%*%y.tilde)%*%G

iter <- 100
alpha <- seq(-0.99,-0.01, length.out = iter)
logl <- numeric(iter)
for (i in 1:iter) {
v <- sapply(0:(q-1), function(x) alpha[i]^x)
ee <- t(v)%*%Q%*%v
logl[i] <- -n/2*log(ee)
}
```



Model Comparison

	OLS	SAR	MESS
ρ	-	0.36	-
α	-	-	-0.41
Age	-756***	-473***	-495***
# Bedrooms	-2,402	-314	-433
# Bathrooms	12,194***	9,987***	10,249***
Total Living Area	480***	373***	380***
Adj.R-squared	0.66	0.79	0.80
Computing Time	2.14 ms	103350 ms	3.8 ms

Application Example

$$X_1 = \left[\underbrace{1}_{Intercept} \quad \underbrace{10}_{Age} \quad \underbrace{3}_{Beds} \quad \underbrace{2}_{Baths} \quad \underbrace{130}_{TLA} \right]'$$

$$Loc_1 = \left[\underbrace{-83.73025}_{Longitude} \quad \underbrace{41.66623}_{Latitude} \right]'$$

$$X_2 = \left[\underbrace{1}_{Intercept} \quad \underbrace{10}_{Age} \quad \underbrace{3}_{Beds} \quad \underbrace{2}_{Baths} \quad \underbrace{130}_{TLA} \right]'$$

$$Loc_2 = \left[\underbrace{-83.49643}_{Longitude} \quad \underbrace{41.63656}_{Latitude} \right]'$$

Model Results	House 1	House 2
Estimated Value OLS	$113,643	$113,643
Estimated Value MESS	$137,583	$82,828