Schnell - aber valide?

Partitionierte Inferenz bei großen On-Farm-Experimenten

Damon Raeis-Dana

BASF Digital Farming / xarvio

Das große Versprechen

Georeferenzierte Ertragskarten bedeuten:

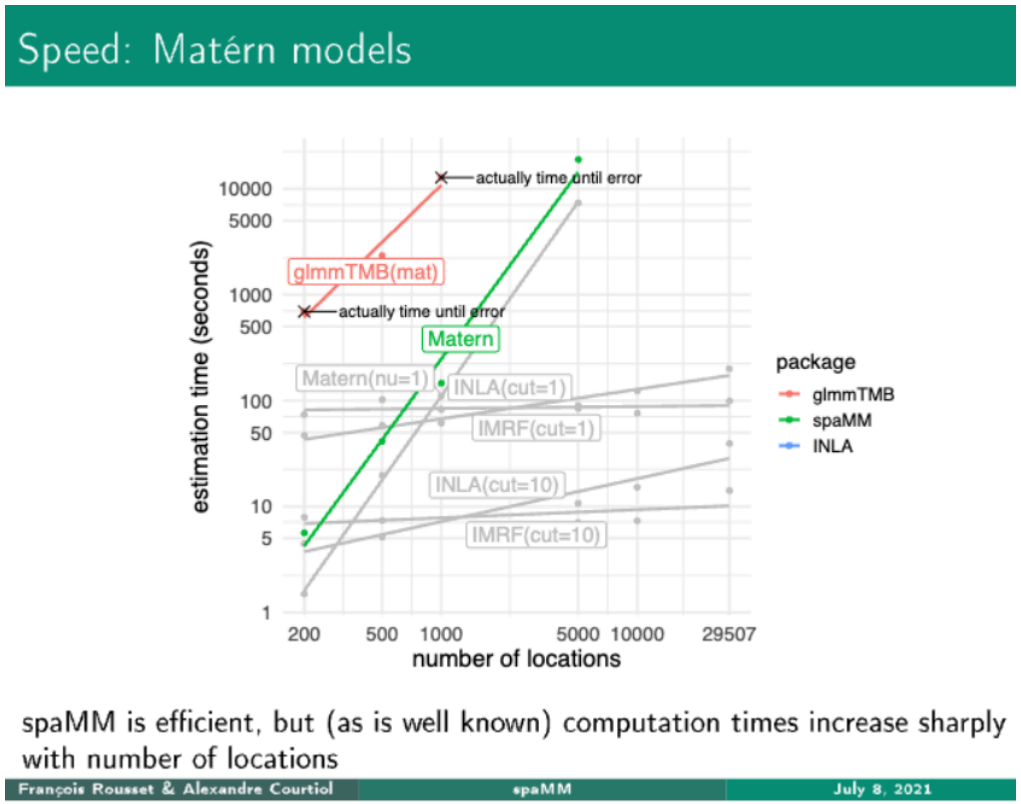
- hohe räumliche Auflösung
- viele Daten = (vermeintlich) viel Information
- digital = schnelle automatisierte Auswertung
- bessere Grenzdifferenzen – einfach(er) zum Ziel **Signifikanz**?

Die Ernüchterung?

- viele *räumlich autokorrelierte* Daten = hohe Rechenzeit
- konkurrierende Modelltypen zum “Ausprobieren”
  - Variogramme, Anisotropie, Blockstruktur, Kovariaten, …
- bereits ab $n > 2.000$ wird Modellauswahl zeitlich aufwendig

Effektive Rechenzeit

- 5.000 Beobachtungen *können* bereits > 2h rechnen



Rechenzeit als Praxisproblem I

- Feldversuch mit 100.000 Datenpunkten – was tun?
- Bei klassischer Modellierung muss Kovarianz-Matrix invertiert werden
- Rechenzeit wächst kubisch
  - Doppelte Datenmenge: 8-fache Rechenzeit ⚠️
  - 10-fache Datenmenge: 1000-fache Rechenzeit 🔥

Rechenzeit als Praxisproblem II

- Spezifische Zerlegungen der Kovarianzmatrix können Rechenzeit reduzieren, bleiben aber essentiell kubisch
- Das kann teils durch Parallelisierung kompensiert werden
- Aber: auch RAM limitiert
  - n = 100.000 führt zu 80 GB RAM-Bedarf
  - 20-30k typisches Limit für Consumer-Geräte

Was kann helfen?

- Rechencluster
- Parallelisierung
- Variogramme, bei denen $\gamma_{h>\text{Range}} = 0$ erzeugen *sparsity*
- Approximationen der Kovarianzmatrix (NNGP, Sparse/Low-Rank, INLA)

Referenzmodell

- Geostatistisches Modell mit räumlicher Korrelation

$$\mathrm{y} = X\beta + Zb + \epsilon, b \sim \mathcal{N}(0, G), \varepsilon \sim \mathcal{N}(0, \Sigma(\theta))$$

- $X\beta$ Treatment und Kovariaten
- Zb Blockstruktur (Random Effects)
- $\Sigma(\theta)$ räumliche Korrelation
- Konzpetionell attraktiv, häufig mit `nlme::lme` umgesetzt
  - Schätzung von $\Sigma(\theta)$ teuer - wächst kubisch $\mathcal{O}(n^3)$

1 spmodel

Idee

> **Intuition**
>
> - Schätzung auf Subsets der Daten sollte *im Prinzip* die selben Effekte liefern wie der volle Datensatz.

- Teile das Feld in disjunkte Sub-Regionen
- Berechne die Effekte in jeder Sub-Region und aggregiere die Ergebnisse.
- Aggregation ist kein simples Mitteln, sondern bedient sich der *composite likelihood*
- Methode beschrieben in
  - **Indexing and partitioning the spatial linear model for large data sets** (Ver Hoef, 2023)

Funktionsweise

- Beobachtungen werden in **räumlich kompakte lokale Gruppen** eingeteilt (z.B. per *kmeans, Hexagone, Raster*)
- Annahme: Beobachtungen aus verschiedenen Gruppen sind **unkorreliert**.
- **Innerhalb** jeder Gruppe bleibt räumliche Kovarianzstruktur (z.B. Matérn) erhalten.
- Vor allem lokale Dependenzstruktur bleibt erhalten

Folge für die Kovarianzmatrix

- Block-diagonale, dünn besetzte Kovarianzmatrix
- viele kleine Matrixoperationen statt einer großen n x n Matrix
- **Wichtig**: geschätzt wird ein einziges globales Modell

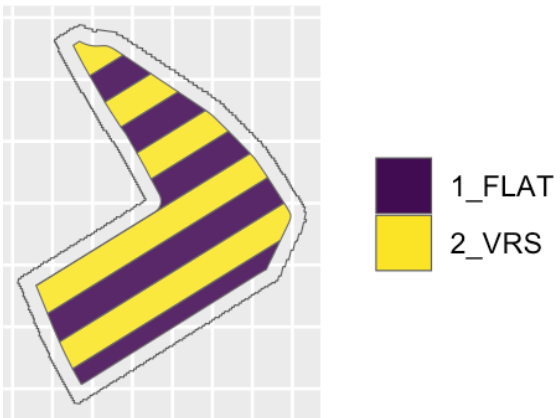
Nachteile

- Inferenz für fixe Effekte ist exakt *unter der block-diagonalen Kovarianzstruktur*
  - relativ zum vollen, dichten Modell ist sie eine Approximation
- Annahme der Unkorreliertheit zwischen Subsets ist nicht immer valide

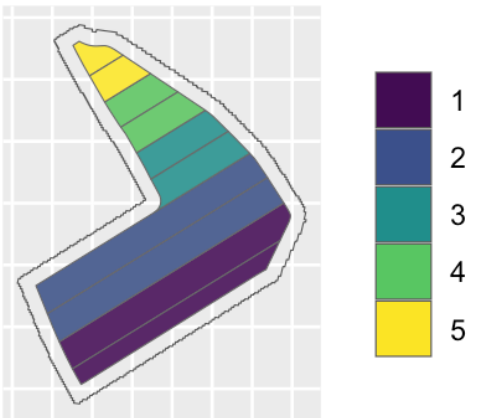
2 Versuch

Versuch - visuell

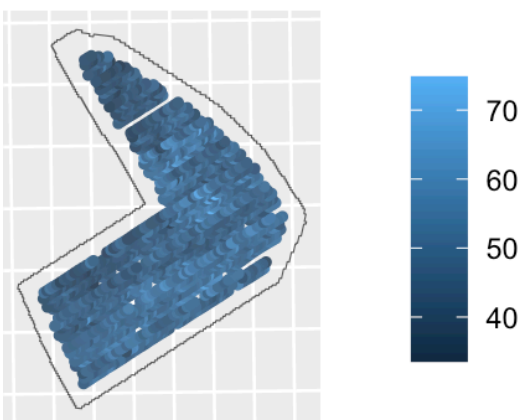
Trial - Treatment No.



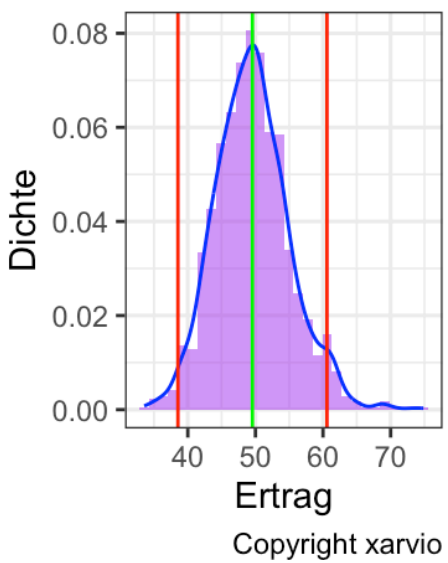
Trial - Replicate No.



Ertrag - bereinigt



Ertragsverteilung - Bereinigt



Zentraler Vergleich

- Beispiel-Datensatz hat ca. $n = 1.600$
- `spmodel` nutzt in der Standardeinstellung $m = 100$ als Blockgröße

```
local = list(method = "kmeans", size = 100, var_adjust = "theoretical", parallel = FALSE)
```

- Aus einem *1600 x 1600* Problem werden sechzehn *100 x 100* Probleme
- Rechenleistung von vollem Modell zu einem einzelnen Block: $\frac{1600^3}{100^3} = 4096$ x höher
- Da aber 16 Blöcke berechnet werden müssen, ergibt sich 4096/16 = 256-fache hypothetische Rechenzeitersparnis
- Echte Schätzung beinhaltet *Partitionierung*, *Overhead der Optimisierung*, *wiederholte Likelihood Evaluationen*.

Schätzung

Model	Estimate	SE	Statistic	Statistic_type	AIC	time_sec
linear_regression	2.00	0.28	7.22	t	9543.30	0.00
linear_mixed_model	1.23	0.26	4.78	t	9262.57	0.52
gls_exp	2.90	0.62	4.66	t	8922.07	38.27
spmodel_exp_100	3.12	0.61	5.15	z	8955.18	0.49
spmodel_exp_256	3.07	0.60	5.12	z	8936.08	0.65
spmodel_exp_512	2.82	0.56	5.00	z	8932.43	2.59
spmodel_exp_aniso	2.28	0.95	2.41	z	8874.41	7.39

- Lineare Regression und LMM mit deutlich schlechterer Modellanpassung
- Niedriger AIC des anisotropischen Modells kann auf Fehlspezifikation hindeuten
- Zentrales Ergebnis davon aber unberührt – Schätzung und Standardfehler durch `spmodel` in deutlich kürzerer Zeit (Faktor 80)

3 Fazit

Fazit

- Große Ertragskartendatensätze sind **inferenziell nicht einfach**.
- Das geostatistische Referenzmodell ist **konzeptionell attraktiv**, aber rechnerisch ab $n > 10.000$ oft **unpraktisch**.
- `spmodel` ist eine plausible und gut dokumentierte Approximation für große Datensätze.
- Die Inferenz ist **exakt unter dem blockweisen Modell**, aber bleibt eine Approximation relativ zum dichten Vollmodell.

Empfehlungen

- **Partitionen räumlich kompakt wählen**, nicht schachbrettartig/inter-leaved.
  - Publizierte Evidenz spricht für kompakte Partitionen
- Für **Reproduzierbarkeit** `set.seed` verwenden (*kmeans* stochastisch)
- Bei **Instabilität** Partitionen vergrößern
- **Anisotropie** vorher prüfen: erhöht Rechenaufwand signifikant
- Modellwahl mit ML, finale Schätzung mit REML

Effizienz

Total (n)	Subset (m)	Gruppen (B=n/m)	Local cost (Bx)	Full cost (B^3x)	Full / local time ratio (B^2)
1,000	100	10	(10x)	(1000x)	100
5,000	100	50	(50x)	(125000x)	2500
10,000	100	100	(100x)	(1000000x)	10000
50,000	100	500	(500x)	(125000000x)	250000
100,000	100	1000	(1000x)	(1000000000x)	1000000

Bei 50.000 Datenpunkten in 100er Subsets rechnet `spmodel` **theoretisch** etwa 250.000 mal schneller – kein Laufzeitversprechen oder Benchmarking.

Theoretisch reduziert sich der dominante Kovarianz-Algebra-Aufwand um einen Faktor von etwa B^2.

Echte Schätzung beinhaltet *Partitionierung*, *Overhead der Optimisierung*, *wiederholte Likelihood Evaluationen*.

Alternativen:

- `spaMM`
- `BRISC`
- `GPVecchia`

4 Backup

Modellvergleich - “Klassisch”

- Linear Mixed Model

$$\text{Yield} = \text{Treatment} + (1|\text{Block}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

- Linear Mixed Model + Kovariate

$$\text{Yield} = \text{Treatment} + \text{Productivity} + (1|\text{Block}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

Modellvergleich - Geostatistisch

- Geostatistisches Modell (Matern, isotropisch)

$$\text{Yield} = \text{Treatment} + \varepsilon(s), \quad \varepsilon(s) \sim \mathcal{GP}(0, C(h)), \quad C(h) = \text{Matern}(h; \nu, \rho)$$

- Geostatistisches Modell (Matern, isotropisch) + Block-Struktur

$$\text{Yield} = \text{Treatment} + (1|\text{Block}) + \varepsilon(s), \quad \varepsilon(s) \sim \mathcal{GP}(0, C(h)), \quad C(h) = \text{Matern}(h; \nu, \rho)$$

- Geostatistisches Modell (Matern, anisotropisch)

$$\text{Yield} = \text{Treatment} + \varepsilon(s), \quad \varepsilon(s) \sim \mathcal{GP}(0, C(h, \theta)), \quad C(h, \theta) = \text{Matern}(h, \theta; \nu, \rho)$$

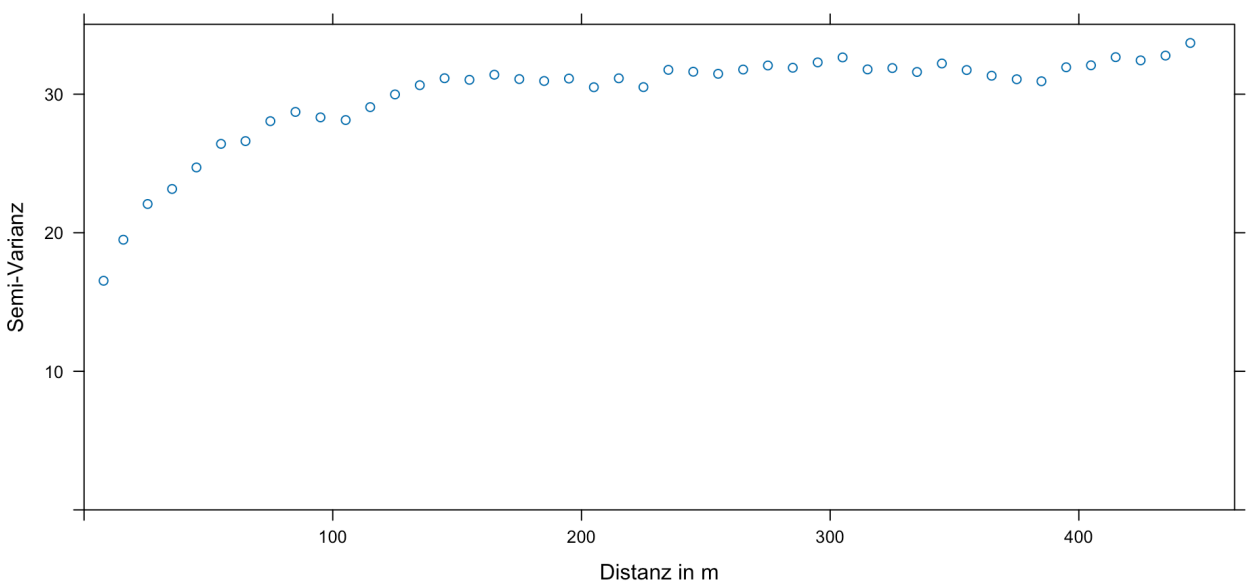
- Geostatistisches Modell + Interaktion mit Kovariate (Matern, isotropisch)

$$\text{Yield} = \text{Treatment x Productivity} + \varepsilon(s), \quad \varepsilon(s) \sim \mathcal{GP}(0, C(h)), \quad C(h) = \text{Matern}(h; \nu, \rho$$

Variogramm

Intuition

- Ertragspunkte sind **nicht unabhängig**:
  - $n_{\text{effektiv}} << n$
- Punkte, die nah beieinander liegen, sind sich ähnlicher als weit entfernte Punkte.
- Modelliere die **Korrelation zwischen Punkten** als **Funktion der Distanz**



Anisotropie?

- Anisotropie beschreibt die Winkelabhängigkeit der Distanzen
  - Korrelation zwischen Punkten als **Funktion der Distanz *und des Winkels***
- Im landwirt. Kontext diktieren z.B. Fahrspuren und Gelände diese Abhängigkeit