UNIVERSITÄT BIELEFELD
Faculty of Business Administration and Economics

Advances in the initialization of probit model estimation

and the {ino} R package

Lennart Oelschläger Dietmar Bauer Marius Ötting

Bielefeld University, Econometrics Group

18 November 2022

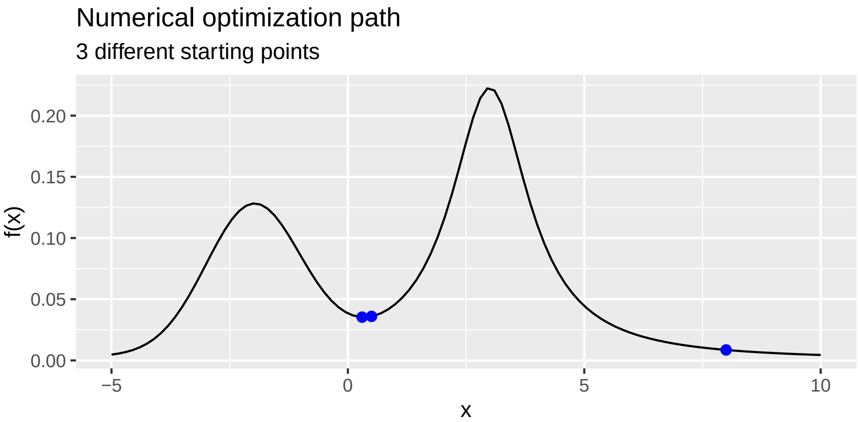
What is this talk about?

1. The initialization problem
2. The probit model
3. New initialization idea
4. The {ino} R package

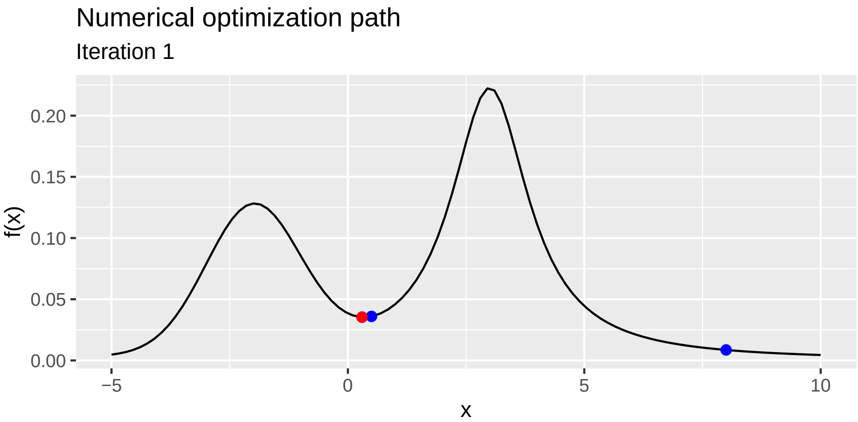
What is this talk about?

1. The initialization problem
2. The probit model
3. New initialization idea
4. The {`ino`} R package

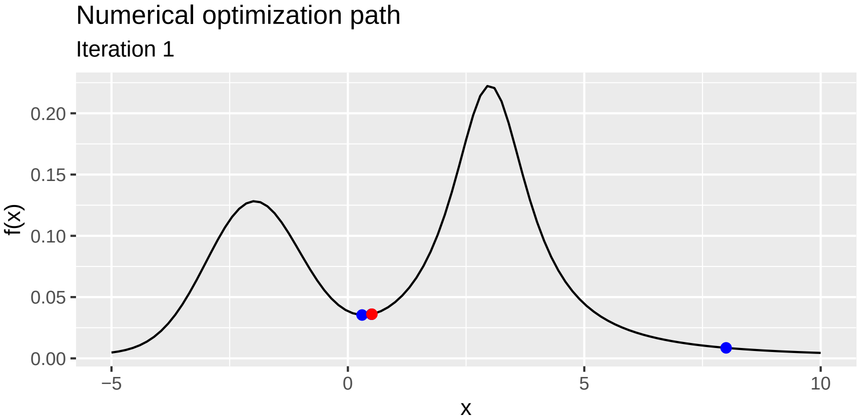
Why do we have a problem?



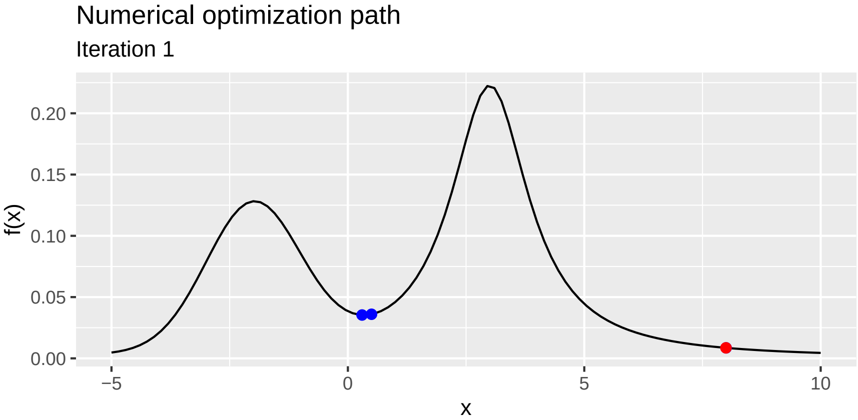
We find a local optimum



We find the global optimum



Convergence takes long



What is this talk about?

1. The initialization problem
2. The probit model
3. New initialization idea
4. The {ino} R package

Model definition

Given choice data

- Discrete choice of decider n: $y_n \in \{1, \dots, J\}$
- Matrix of (alternative- or decider-specific) covariates of n: $X_n \in \mathbb{R}^{J \times P}$

Model definition

Given choice data

- Discrete choice of decider n: $y_n \in \{1, \dots, J\}$
- Matrix of (alternative- or decider-specific) covariates of n: $X_n \in \mathbb{R}^{J \times P}$

Probit model

$$
\begin{aligned}
U_n &= X_n\beta + \epsilon_n \in \mathbb{R}^J && \text{(Latent utilities)} \\
\epsilon_n &\sim \mathcal{N}_J(0, \Sigma) && \text{(Error term)} \\
y_n &= \arg\max U_n && \text{(Choice link)}
\end{aligned}
$$

What we want? Estimates $\hat{\beta}$ (mean sensitivities) and $\hat{\Sigma}$ (error characterization).

Model normalization

The probit model (like any utility model) must be normalized:

Scale normalization

- $U > U' \;\Leftrightarrow\; c \cdot U > c \cdot U' \quad \forall\, c \in \mathbb{R}_+$
- For identification, fix, e.g., one entry of β to 1 (determines c)

Model normalization

The probit model (like any utility model) must be normalized:

Scale normalization

- $U > U' \;\Leftrightarrow\; c \cdot U > c \cdot U' \quad \forall\, c \in \mathbb{R}_+$
- For identification, fix, e.g., one entry of β to 1 (determines c)

Level normalization

- $U > U' \;\Leftrightarrow\; U + k > U' + k \quad \forall\, k \in \mathbb{R}$
- Consider utility differences: $U > U' \Leftrightarrow (U + k) - (U' + k) > 0$ (cancels k)
- Note: we loose one dimension ($J \rightsquigarrow J - 1$)

Utility differences

Difference utility vector $U_n \in \mathbb{R}^J$ with respect to some reference alternative i:

$$\Delta_i U_n \in \mathbb{R}^{J-1}$$

Utility differences

Difference utility vector $U_n \in \mathbb{R}^J$ with respect to some reference alternative i:

$$\Delta_i U_n \in \mathbb{R}^{J-1}$$

Note that (choosing $i = y_n$):

$$\Delta_{y_n} U_n < 0.$$

Utility differences

Difference utility vector $U_n \in \mathbb{R}^J$ with respect to some reference alternative i:

$$\Delta_i U_n \in \mathbb{R}^{J-1}$$

Note that (choosing $i = y_n$):

$$\Delta_{y_n} U_n < 0.$$

The difference operator looks like this:

$$\Delta_i = \begin{array}{c c} & i \\ \begin{array}{c} \\ \\ i-1 \\ i+1 \\ \\ \\ \end{array} & \begin{pmatrix} 1 & & & -1 & & & \\ & \ddots & & -1 & & 0 & \\ & & 1 & -1 & & & \\ & & & -1 & 1 & & \\ & 0 & & -1 & & \ddots & \\ & & & -1 & & & 1 \end{pmatrix} \end{array} \in \{-1, 0, 1\}^{(J-1)\times J}$$

Optimization problem

Probability for choosing alternative i:

$$P_{ni}(\beta, \Sigma) = \mathsf{Prob}(\Delta_i U_n < 0) = \underbrace{\Phi_{J-1}(-\Delta_i X_n \beta \mid 0, \Delta_i \Sigma \Delta_i')}_{\text{Computation expensive}}$$

Optimization problem

Probability for choosing alternative i:

$$P_{ni}(\beta, \Sigma) = \mathsf{Prob}(\Delta_i U_n < 0) = \underbrace{\Phi_{J-1}(-\Delta_i X_n \beta \mid 0, \Delta_i \Sigma \Delta_i')}_{\text{Computation expensive}}$$

Log-likelihood function:

$$\log \mathcal{L}(\beta, \Sigma) = \sum_n \log P_{ny_n}(\beta, \Sigma)$$

Optimization problem

Probability for choosing alternative i:

$$P_{ni}(\beta, \Sigma) = \text{Prob}(\Delta_i U_n < 0) = \underbrace{\Phi_{J-1}(-\Delta_i X_n \beta \mid 0, \Delta_i \Sigma \Delta_i')}_{\text{Computation expensive}}$$

Log-likelihood function:

$$\log \mathcal{L}(\beta, \Sigma) = \sum_n \log P_{ny_n}(\beta, \Sigma)$$

Find MLE:

$$(\hat{\beta}, \hat{\Sigma}) = \arg\max \log \mathcal{L}(\beta, \Sigma)$$

Optimization problem

Probability for choosing alternative i:

$$P_{ni}(\beta, \Sigma) = \text{Prob}(\Delta_i U_n < 0) = \underbrace{\Phi_{J-1}(-\Delta_i X_n \beta \mid 0, \Delta_i \Sigma \Delta_i')}_{\text{Computation expensive}}$$

Log-likelihood function:

$$\log \mathcal{L}(\beta, \Sigma) = \sum_n \log P_{ny_n}(\beta, \Sigma)$$

Find MLE:

$$(\hat{\beta}, \hat{\Sigma}) = \arg\max \log \mathcal{L}(\beta, \Sigma)$$

Note: instead of Σ, optimize over L with $\Sigma = LL'$, where L is the lower-triangular Cholesky root with positive diagonal entries (for uniqueness)

What is this talk about?

1 The initialization problem

2 The probit model

3 New initialization idea

4 The {ino} R package

Using regression

Let's initialize β.

Using regression

Let's initialize β. Idea:

1. Assume that Σ is known (if unknown, set $\Sigma = 1^{J \times J}$)
2. Consider first-order Taylor approximation of $P_{n:}$ around 0:

$$P_{n:}(-\Delta_{:} X_n \beta \mid \Sigma) = P_{n:}(0 \mid \Sigma) + \nabla P_{n:}(0 \mid \Sigma) \cdot (-\Delta_{:} X_n \beta) + R$$

3. Since $\mathbb{E}(y_n \mid \Sigma) = P_{n:}(-\Delta_{:} X_n \beta \mid \Sigma)$:

$$y_n = P_{n:}(0 \mid \Sigma) + \underbrace{\nabla P_{n:}(0 \mid \Sigma) \cdot (-\Delta_{:} X_n}_{\tilde{X}_n} \beta) + e_n \quad \text{(not a catch-22!)}$$

4. Compute OLS estimator $\hat{\beta}_{OLS}$ (very fast, just matrix product and inverting)

Using MCMC

And what about Σ?

Using MCMC

And what about Σ?

Trigger warning

Bayes people, please cover your eyes. Abuse of Bayes idea incoming.

Using MCMC

And what about Σ?

Trigger warning

Bayes people, please cover your eyes. Abuse of Bayes idea incoming.

Idea:

1. Assume that β is known (if unknown, set $\beta = \hat{\beta}_{OLS}$)
2. Consider posterior of model parameters, including augmented $(U_n)_n$:

$$\text{Prob}(\beta, \Sigma, U \mid y) \propto \text{Prob}(\beta, \Sigma) \cdot \text{Prob}(U \mid \beta, \Sigma) \cdot 1\{y_n = \arg\max U_n\}$$

3. Assume conjugate prior and draw from posterior using Gibbs sampling (fairly fast)
4. Find $\hat{\Sigma}_{MCMC}$ as marginal posterior mode

Putting it together

Algorithm:

1. Initialize $\Sigma = 1^{J \times J}$
2. Estimate $\hat{\beta}_{OLS}$ using OLS
3. Estimate $\hat{\Sigma}_{MCMC}$ via MCMC
4. Initialize MLE with $(\hat{\beta}_{OLS}, \hat{\Sigma}_{MCMC})$

Hope:

- with Step 2 and 3, we initialize MLE close at the global optimum
- so that 4 is faster and more likely converges

Simulation results

Settings: $N = 200$, $J = 4$, $P = 4$, $X_n \overset{iid}{\sim} \mathcal{N}(0,1)^{J \times P}$

True parameter: $\beta \sim \mathcal{N}(0,1)^P$, $\beta_1 = 1$, $\Sigma = LL' \sim \mathcal{W}^{-1}$

Compare: Random initialization versus strategy in terms of computation time (`sec`) and deviation of MLE from true parameter (`ndev`)

Simulation results

Settings: $N = 200$, $J = 4$, $P = 4$, $X_n \overset{iid}{\sim} \mathcal{N}(0,1)^{J \times P}$

True parameter: $\beta \sim \mathcal{N}(0,1)^P$, $\beta_1 = 1$, $\Sigma = LL' \sim \mathcal{W}^{-1}$

Compare: Random initialization versus strategy in terms of computation time (`sec`) and deviation of MLE from true parameter (`ndev`)

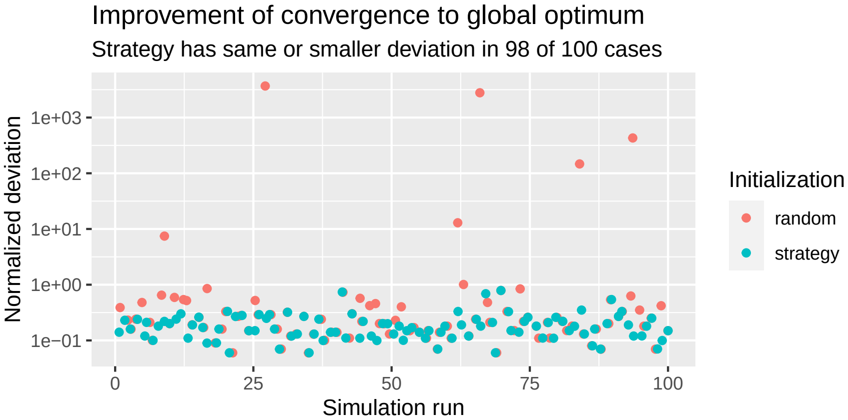
Table 1: One example run.

	b_1	b_2	b_3	b_4	l_1	l_2	l_3	l_4	l_5	l_6	sec	ndev
true_par	1	0.57	-1.10	1.17	2.46	-0.04	-0.13	2.38	0.10	1.13	0.00	0.00
init_random	1	-0.10	1.15	-0.91	0.23	-0.17	-0.91	0.78	-0.27	0.95	0.00	0.43
est_random	1	0.69	-1.14	1.36	2.10	-1.25	-0.32	1.48	0.38	0.92	487.14	0.16
init_strategy	1	0.87	-1.28	1.25	1.99	-1.22	-0.36	0.72	-0.52	0.66	1.20	0.23
est_strategy	1	0.69	-1.14	1.36	2.10	-1.25	-0.32	1.48	0.38	0.92	101.37	0.16

Simulation results

UNIVERSITÄT BIELEFELD
Faculty of Business Administration and Economics

Simulation results



What is this talk about?

1 The initialization problem

2 The probit model

3 New initialization idea

4 The {ino} R package

Purpose

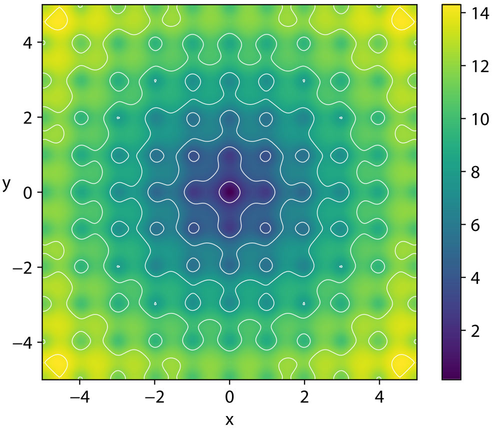
- Joint work with Marius
- Implements strategies for the **i**nitialization of **n**umerical **o**ptimization:
  - effect of random initialization versus fixed initialization
  - effect of standardizing covariates
  - effect of subsetting covariates
  - effect of alternating optimization
  - comparing optimizer
  - number of identified optima
- Available on CRAN

```
> library("ino")
```

Setup

```
> x <- setup_ino(
+   f = f_ackley,
+   npar = 2,
+   global = c(0,0),
+   opt = set_optimizer_nlm()
+ )
```

```
## Function to be optimized
## f: f_ackley
## npar: 2
##
## Numerical optimizer
## 'stats::nlm': <optimizer 'stats::nlm'>
##
## Optimization runs
## Records: 0
```



Random initialization

```
> random_initialization(x) %>% get_vars(vars = ".estimate")
```

```
## [1]  2.82139e-07 -1.75042e-07
```

Random initialization

```
> random_initialization(x) %>% get_vars(vars = ".estimate")
```

```
## [1]  2.82139e-07 -1.75042e-07
```

```
> x <- random_initialization(
+   x, runs = 100, ncores = 3,
+   sampler = function() stats::rnorm(npar(x))
+ )
```

Random initialization

```
> random_initialization(x) %>% get_vars(vars = ".estimate")

## [1]  2.82139e-07 -1.75042e-07

> x <- random_initialization(
+   x, runs = 100, ncores = 3,
+   sampler = function() stats::rnorm(npar(x))
+ )

> overview_optima(x, digits = 2)

##   optimum frequency
## 1       0        44
## 2    2.58        36
## 3    3.57        12
## 4    5.38         6
## 5    6.56         1
## 6    7.96         1
```

Thanks for listening!

Key message:

- MLE for probit model is sensitive to initial values
- Regression + MCMC reduce computation time
- {`ino`} provides universal initialization strategies

Open questions:

- Consistency of strategy?
- How to initialize parameters of mixing distribution?

Please let me know:

- How is initialization an issue for you?
- Thoughts on {ino}?
- Other ideas for initialization?

loelschlaeger.de/talks

loelschlaeger@uni-bielefeld.de