An MM Algorithm for Fixed Effects Multinomial Logit Models

Mingli Chen[1] Ian Meeker[2] Marc Rysman[3] Shuang Wang[2]

September 9, 2026

[1]University of Warwick [2]Charles River Associates [3]Boston University

Motivation

- Fixed effects are everywhere in modern empirical work — and in *linear* models they are easy: demeaning absorbs millions of them.
- In discrete choice models like the multinomial logit, there is no such shortcut. Standard practice: estimate a dummy-variable coefficient for every fixed effect, by Newton-type methods.
- That gets slow, memory-hungry, and eventually *infeasible* as the number of fixed effects grows — think one intercept per household, per hospital, per store...
- So in practice researchers compromise: split the sample, discretize variables, or drop the fixed effects altogether.

This paper

A way to estimate multinomial logit models with millions of fixed effects — by turning the problem into a sequence of ordinary linear regressions.

Why estimating the fixed effects matters

- Classic workaround: Chamberlain's conditional logit *avoids* the fixed effects by conditioning them out. Clever — but then you never get them back.
  - Extends beyond binary choice in principle, but the conditioning event's combinatorics blow up fast in periods and alternatives — tractable only for very short panels.
  - Breaks with any other individual-specific heterogeneity, e.g. a person-specific coefficient.
- Without the fixed effects in hand you cannot predict choice probabilities, compute elasticities or welfare, or run counterfactuals. That has kept many applied researchers away from fixed-effects logit models.
- Our approach estimates every fixed effect and coefficient directly — so the model's outputs are actually usable (e.g. Dubois–Griffith–O'Connell 2020 need individual price coefficients to compute compensating variation).
- Already in use: school choice in India (Sahai 2023); payment choice with 1.4 million fixed effects (Rysman–Wang–Wozniak 2025).

What we do

1. Use the Minorization–Maximization (MM) algorithm — a cousin of the EM algorithm — to replace the hard logit likelihood with a sequence of easy problems.
   - **EM**, the familiar recipe for latent-variable models (mixtures, random effects), guesses the missing data, averages over it (E-step), then re-maximizes (M-step).
   - **Why it doesn't work here:** EM would replace the missing utility with its conditional expectation, but *the likelihood of the expectation is not the expectation of the likelihood* — that substitution is biased for the multinomial logit. MM instead builds a surrogate for the likelihood directly, with no data-augmentation required.
2. Each easy problem turns out to be an OLS regression $\Rightarrow$ all the linear panel tricks (demeaning, multi-way fixed effects, heterogeneous slopes) suddenly apply to a nonlinear model.

What we do (continued)

3. Connect our method to a classic: Böhning's (1992) MM algorithm for the logit — which never considered fixed effects. Once recast as a regression, it takes exactly the same steps as ours in the fixed-effects case: his needs a whitened (GLS) regression, ours is plain OLS, so we get his accuracy for free, with simpler code.
4. Add an off-the-shelf accelerator: the result estimates 1,000,000 fixed effects in about 90 seconds, beating a heavily optimized Newton's method with a fraction of the programming effort.
5. Apply it to hospital choice: current antitrust practice can't jointly estimate that many fixed effects, so it settles for cruder models — ours doesn't have to.

The Idea

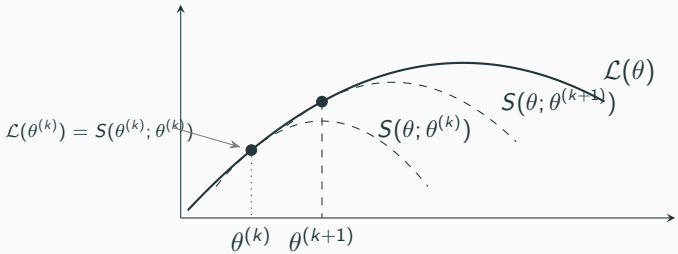
The model, formally

- Individual i chooses among J alternatives at each of T occasions. Utility:

$$u_{ijt} = \underbrace{x_{it}\beta_j}_{\text{alt-specific coef.}} + \underbrace{w_{jt}\gamma_i}_{\text{indiv-specific coef.}} + \underbrace{\alpha_{ij}}_{\text{indiv}\times\text{alt FE}} + \varepsilon_{ijt}, \qquad \varepsilon_{ijt} \overset{iid}{\sim} \text{Type I EV.}$$

- $y_{ijt} = 1\{i \text{ chooses } j \text{ at } t\}$; the linear index is $\psi_{ijt}(\theta) = x_{it}\beta_j + w_{jt}\gamma_i + \alpha_{ij}$, and choice probabilities are the usual softmax $p_{ijt}(\theta) = \exp\{\psi_{ijt}(\theta)\} / \sum_m \exp\{\psi_{imt}(\theta)\}$.
- $\theta = \{\beta_j, \gamma_i, \alpha_{ij}\}$ can run into the millions of parameters once α_{ij} has an entry per individual–alternative pair.
- Log-likelihood $\mathcal{L}(\theta) = \sum_{i,t} \sum_j y_{ijt} \log p_{ijt}(\theta)$ is globally concave — so there's a unique maximum — but with millions of parameters, brute-force Newton-type maximization is what gets slow or infeasible.

The MM idea: make a hard problem easy, repeatedly



$$\theta^{(k+1)} = \arg\max_{\theta} S(\theta;\theta^{(k)}), \qquad S(\theta;\theta^{(k)}) \le \mathcal{L}(\theta), \quad S(\theta^{(k)};\theta^{(k)}) = \mathcal{L}(\theta^{(k)}).$$

- Build a simple surrogate $S(\cdot;\theta^{(k)})$ that sits *below* the true likelihood $\mathcal{L}$ everywhere and touches it at the current guess $\theta^{(k)}$; maximize S to get $\theta^{(k+1)}$; repeat.
- Because $S \le \mathcal{L}$ with equality at $\theta^{(k)}$, every step is guaranteed to improve the likelihood — no step sizes to tune, no divergence. (EM is the special case where the surrogate is a conditional expectation.)

Our surrogate: the logit becomes a regression

Recall the log-likelihood $\mathcal{L}(\theta) = \sum_{i,t} \sum_j y_{ijt} \log p_{ijt}(\theta)$ — concave, but hard to maximize directly.

How we find S: Taylor-expand $\mathcal{L}$ around $\theta^{(k)}$, replace its messy cross-alternative Hessian with a simple scalar curvature bound, and complete the square — since $\psi_{ijt}(\theta)$ is linear in θ, this gives an exact least-squares ("quasi-linear") problem that reproduces $\mathcal{L}(\theta^{(k)})$ at $\theta = \theta^{(k)}$:

$$S(\theta; \theta^{(k)}) \;=\; \underbrace{\mathcal{L}(\theta^{(k)}) + \textstyle\sum_{i,j,t} (y_{ijt} - p_{ijt}^{(k)})^2}_{\text{const, does not depend on } \theta} \;-\; \tfrac{1}{4} \sum_{i,j,t} \big(v_{ijt}^{(k)} - \psi_{ijt}(\theta)\big)^2,$$

$$v_{ijt}^{(k)} \;=\; \psi_{ijt}(\theta^{(k)}) \;+\; 2\big(y_{ijt} - p_{ijt}^{(k)}\big).$$

Maximizing S is exactly OLS of v on the covariates and fixed-effect dummies: $\theta^{(k+1)} = (Z'Z)^{-1} Z' v^{(k)}$. Repeat until nothing changes.

Putting it together: the algorithm

1: Initialize $\theta^{(0)}$ (e.g. all zeros)
2: **repeat**
3: Compute choice probabilities $p_{ijt}^{(k)}$ from current $\theta^{(k)}$
4: Form the pseudo-outcome $v_{ijt}^{(k)} = x_{it}\beta_j^{(k)} + w_{jt}\gamma_i^{(k)} + \alpha_{ij}^{(k)} + 2\big(y_{ijt} - p_{ijt}^{(k)}\big)$
5: Demean $v^{(k)}$ and the regressors to absorb the fixed effects
6: Solve $\theta^{(k+1)} = (Z'Z)^{-1}Z'v^{(k)}$ by OLS
7: **until** $\|\theta^{(k+1)} - \theta^{(k)}\|$ is below tolerance
8: **return** $\hat{\theta} = \theta^{(k+1)}$

- That's the whole algorithm — every iteration is: update probabilities, build a pseudo-outcome, run OLS.
- No step size, no line search, no matrix inversion bigger than the shared coefficients — fixed effects, and any individual-specific slopes, are recovered afterward without ever inverting anything of that size.

Absorbing fixed effects: dummy variables vs. demeaning

Brute force: treat each α_{ij} as its own regression coefficient — stack x_{it}, w_{jt}, and $I \times J$ dummies into Z, run OLS. Exact, but $Z'Z$ has a dimension for every fixed effect: fine for thousands, impossible for millions.

Demeaning (Frisch–Waugh–Lovell): the surrogate is linear at each iteration, so FWL applies exactly as in a linear model.

1. Demean the pseudo-outcome $v^{(k)}$ and the regressors within each fixed-effect group $\Rightarrow$ recover the shared coefficients $\eta = (\beta, \gamma)$ from a *small* demeaned regression, however many fixed effects there are:

$$\eta^{(k+1)} = (\tilde{\mathcal{Z}}'\tilde{\mathcal{Z}})^{-1}\tilde{\mathcal{Z}}'\tilde{v}^{(k)}.$$

2. Recover the fixed effects afterward, as a residual:

$$\alpha^{(k+1)} = v^{(k)} - \mathcal{Z}\eta^{(k+1)} - \left(\tilde{v}^{(k)} - \tilde{\mathcal{Z}}\eta^{(k+1)}\right).$$

Multi-way fixed effects (person×alternative *and* alternative×time) need iterative demeaning (alternating projections) instead of simple averaging — off-the-shelf tools (`reghdfe`, `PyHDFE`, `lfe`) already do this.

Skipping demeaning: conditional maximization (MCM)

Instead of jointly maximizing the surrogate S over all parameters, maximize it one block at a time, holding the rest fixed:

$$\eta^{(k+1)} = \arg\max_{\eta} S(\eta, \alpha^{(k)}; \theta^{(k)}), \qquad \alpha^{(k+1)} = \arg\max_{\alpha} S(\eta^{(k+1)}, \alpha; \theta^{(k)}).$$

Each block's first-order condition has a simple closed form — no demeaning needed, just OLS with the other block's current value plugged in as a constant:

$$\eta^{(k+1)} \leftarrow (\mathcal{Z}'\mathcal{Z})^{-1}\mathcal{Z}'(v^{(k)} - \alpha^{(k)}), \qquad \alpha_{ij}^{(k+1)} \leftarrow \tfrac{1}{T}\textstyle\sum_t \left(v_{ijt}^{(k)} - z_{ijt}\eta^{(k+1)}\right).$$

Still guaranteed to climb: since each step only improves (not fully maximizes) S,

$$\mathcal{L}(\theta^{(k)}) = S(\theta^{(k)}; \theta^{(k)}) \le S(\eta^{(k+1)}, \alpha^{(k)}; \theta^{(k)}) \le S(\eta^{(k+1)}, \alpha^{(k+1)}; \theta^{(k)}) \le \mathcal{L}(\theta^{(k+1)}).$$

We call this Minorization–Conditional-Maximization (MCM) — it generalizes the ECM idea (Meng–Rubin 1993) to our setting. Trade-off: cheaper steps, but ignoring cross-block terms can mean more of them — best when blocks are close to independent.

Beyond one-way fixed effects

Once the maximization step is a regression, the rest of the linear panel toolkit comes along for free:

- **Multi-way fixed effects:** person×alternative *and* alternative×time effects together, via alternating-projection demeaning.
- **Heterogeneous coefficients:** person-specific slopes (e.g. everyone their own price sensitivity), via generalized demeaning.
- **Precompute once:** the regressors never change across iterations, so the expensive matrix work (the pseudo-inverse, or the demeaning transform) happens a single time; each iteration afterward is essentially one matrix multiply.
- **Or skip demeaning altogether:** conditional maximization (MCM) trades exactness in each step for simplicity — useful when demeaning itself is the expensive part, as we'll see in the horse races.

Connections & Extensions

A classic connection: Böhning (1992)

- Thirty years ago, Böhning proposed an MM algorithm for the multinomial logit with a *smarter* curvature bound — it knows the alternatives are related, so it takes better-aimed steps. The price: each step is a *weighted* (GLS) regression. Böhning's paper never mentions fixed effects — it's built for a single shared coefficient vector.
- First, we recast his update as a regression too — so the same fixed-effects machinery (demeaning, etc.) can be layered onto *his* algorithm as well.
- The surprise: for models where every coefficient is alternative-specific — which includes our headline fixed-effects case — the two algorithms take exactly the same steps, iteration for iteration (proof next).
- So our simple OLS version gets Böhning's better-aimed path *for free*, with none of his reweighting machinery. Where the methods genuinely differ is with person-specific coefficients — we'll see that in the second simulation.

Why the two algorithms coincide, exactly

For models where *every* coefficient is alternative-specific (our headline fixed-effects case), the two updates are algebraically the same:

- Böhning normalizes against a base alternative and takes a GLS step with bound $\boldsymbol{B} = \frac{1}{2}(\boldsymbol{I}_{J-1} - \frac{1}{J}\mathbf{1}\mathbf{1}')$. Because the same regressors enter every alternative, his design matrix factors as a Kronecker product, $\dot{\boldsymbol{Z}} = \boldsymbol{X} \otimes \boldsymbol{I}_{J-1}$, and the whole GLS step collapses — using $\boldsymbol{B}^{-1} = 2(\boldsymbol{I}_{J-1} + \mathbf{1}\mathbf{1}')$ and the fact that logit probabilities sum to one — to

$$\Delta\theta_{B,j} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\big[2(y_j - p_j^{(k)}) - 2(y_1 - p_1^{(k)})\big].$$

- Our plain OLS step, differenced against the same base alternative, gives

$$\Delta\theta_{MM,j} - \Delta\theta_{MM,1} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\big[2(y_j - p_j^{(k)}) - 2(y_1 - p_1^{(k)})\big].$$

Identical, term for term — not just at the optimum: every relative parameter step, predicted probability, and log-likelihood value matches, at every iteration.

One framework: preconditioned gradient ascent

Every method we compare updates parameters as

$$\theta^{(k+1)} = \theta^{(k)} + \sigma_k P_k \nabla \mathcal{L}(\theta^{(k)})$$

— they differ only in the preconditioner P_k and the step size σ_k:

- **Gradient ascent:** $P_k = I$, constant σ — simple, but highly sensitive to the step size.
- **Adam:** a dynamic, diagonal P_k built from momentum in the gradient and squared gradient — cheap per step, but ignores cross-parameter structure and needs its hyperparameters tuned.
- **Newton:** $P_k = -\nabla^2 \mathcal{L}(\theta^{(k)})^{-1}$, the exact inverse Hessian — fastest convergence, most expensive step.
- **Our MM:** constant $P_k = (Z'Z)^{-1}$, derived from a strict minorizing bound — guarantees ascent with a *fixed* step size, no tuning at all.
- **Böhning's MM:** a different constant metric that captures cross-alternative correlation — sharper steps, at the cost of a GLS regression.

All the algorithms in this talk are the same update, with a different guess at curvature.

The practical toolkit: acceleration

SQUAREM: view the MM update as a fixed-point map $\theta^{(k+1)} = F(\theta^{(k)})$. Take two ordinary steps, $u_k = F(\theta^{(k)}) - \theta^{(k)}$ and $w_k = F(F(\theta^{(k)})) - F(\theta^{(k)})$, and jump straight to

$$\theta^{(k+1)} = \theta^{(k)} - 2s_k u_k + s_k^2(w_k - u_k), \qquad s_k = -\frac{\|u_k\|}{\|w_k - u_k\|}.$$

The step size s_k is set automatically each round — unlike Adam, there is no learning rate to tune. (If a jump overshoots, fall back to a plain MM step to stay safe.)

Applies to any MM variant — ours, Böhning's, or the conditional-maximization (MCM) version — with no change to the underlying algorithm.

Bias correction: the split-panel jackknife

- With fixed effects and a finite number of periods T, MLE has the usual incidental parameters bias, of order $1/T$.
- Split-panel jackknife (Dhaene–Jochmans 2015): estimate on the full panel $\hat{\theta}$, and separately on each half, $\hat{\theta}_1$ and $\hat{\theta}_2$; correct as

$$\tilde{\theta} = 2\hat{\theta} - \tfrac{1}{2}(\hat{\theta}_1 + \hat{\theta}_2).$$

  This removes the leading bias term (and most of the rest).
- MM's speed is what makes this practical: three full estimations instead of one is a minor cost when each one takes seconds, not hours.
- If the data are stable over time, the correction doesn't change the asymptotic variance — just report the standard errors from the full sample alongside $\tilde{\theta}$; if it isn't, an influence-function correction (next slide) handles that too.

Standard errors: why they're hard

Inference needs $\mathrm{Var}(\hat{\beta})$ — the top-left block of $-\mathcal{H}^{-1}$, where $\mathcal{H}$ is partitioned into the shared coefficients β and the high-dimensional nuisance parameters $\xi = (\alpha, \gamma)$. By the Schur complement,

$$\mathrm{Var}(\hat{\beta}) = -\left(H_{\beta\beta} - H_{\beta\xi} H_{\xi\xi}^{-1} H_{\beta\xi}'\right)^{-1}.$$

- Naively, this means inverting $H_{\xi\xi}$ — the Hessian block over *every* fixed effect. Exactly the massive matrix our whole algorithm was built to avoid.
- **The fix:** replace our constant curvature bound with the *true* logit weights $\Omega_{it} = \mathrm{diag}(p_{it}) - p_{it} p_{it}'$. The surrogate then becomes the *exact* second-order expansion of $\mathcal{L}$, and maximizing it is again a (reweighted) regression:

$$\theta^{(k+1)} = \theta^{(k)} + (Z'\Omega^{(k)} Z)^{-1} Z' \left(y - p^{(k)}\right).$$

- This is Newton–Raphson, just written as Iteratively Reweighted Least Squares (IRLS) — the same "augment + solve" structure as our MM algorithm, at the true curvature instead of a constant bound.

Standard errors: the payoff

Run that one IRLS step *once*, at the already-converged MLE, instead of using it to estimate (where it would be $\sim 7\times$ slower than our MM algorithm, per Race 1).

- By Frisch–Waugh–Lovell, the Schur complement above *is* algebraically the weighted cross-product of the covariates, demeaned using those true weights $\hat{\Omega}$. So one weighted regression at convergence gives the variance directly — no giant matrix ever inverted:

$$\mathrm{Var}(\hat{\beta}) = \left(\tilde{X}'\hat{\Omega}\tilde{X}\right)^{-1}.$$

- The same residualized covariates give cluster-robust standard errors (the usual sandwich estimator) essentially for free.
- **Delta method** for transformations of β alone; **bootstrap** for anything involving the fixed effects themselves (e.g. marginal effects) — affordable here specifically because MM estimation is fast enough to re-run hundreds of times.

Does It Work?

Horse race 1: a million fixed effects

Setup: $u_{ijt} = x_{it}\beta_j + \alpha_{ij} + \varepsilon_{ijt}$ ($x_{it}, \alpha_{ij} \sim \mathcal{N}(0,1)$, true $\beta = (0, 0.5, 1)$); 3 alternatives, 20 periods, up to 500,000 people $\Rightarrow$ up to 1,000,000 fixed effects; 250 simulation runs per size.

Contestants:

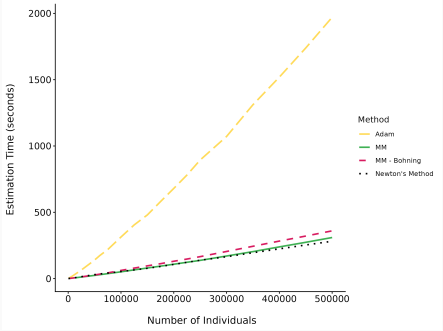
- Our MM and Böhning's MM, each with and without acceleration.
- Our **MCM** variant (conditional maximization, no demeaning) — same estimates, different plumbing; results in a moment.
- **Newton's method** — not the off-the-shelf kind: a bespoke implementation exploiting the Hessian's sparse structure. (Off-the-shelf Stata/R: one iteration at 10,000 fixed effects can take longer than our *entire* estimation at a million.)
- **Adam** — the machine-learning workhorse, recommended for large-scale logit models: takes the *same* log-likelihood gradient $\nabla\mathcal{L}(\theta^{(k)})$ our surrogate is built from, and rescales each coordinate by its running average $\hat{m}^{(k)}$ and squared running average $\hat{v}^{(k)}$,

$$\theta^{(k+1)} = \theta^{(k)} + \sigma \frac{\hat{m}^{(k)}}{\sqrt{\hat{v}^{(k)}} + \epsilon},$$

  but σ must be hand-tuned — too small and it stalls, too large and it fails to converge at all.

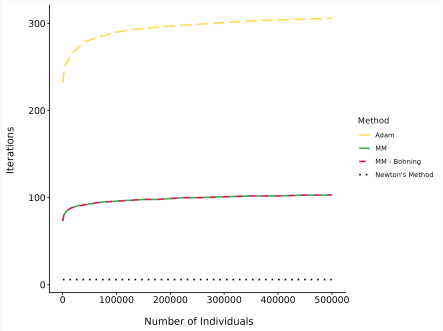
All methods land on the same estimates (differences $< 10^{-6}$) — so this is purely a race about speed and effort.

Race 1, without acceleration



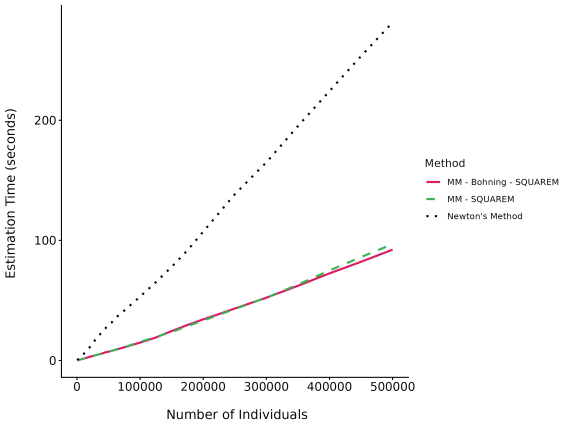
Newton wins the unaccelerated race (≈4 min at 1M fixed effects; Adam ≈30 min, over 7× slower); the MM methods sit right behind, only slightly slower than Newton. Ours and Böhning's take mathematically identical steps here — the tiny runtime gap is just the cost of Böhning's whitening step.

Race 1, iteration counts



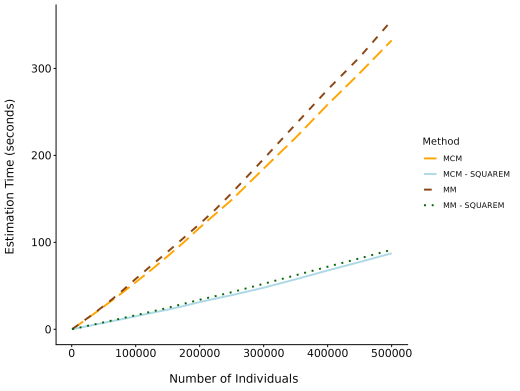
Newton always converges in 6 iterations — it just makes each one expensive. Ours and Böhning's MM take the exact same ∼100 iterations, confirming their mathematical equivalence; Adam needs 3× as many. Cheap steps, more of them: that trade is what lets Newton win on time despite doing far more per iteration.

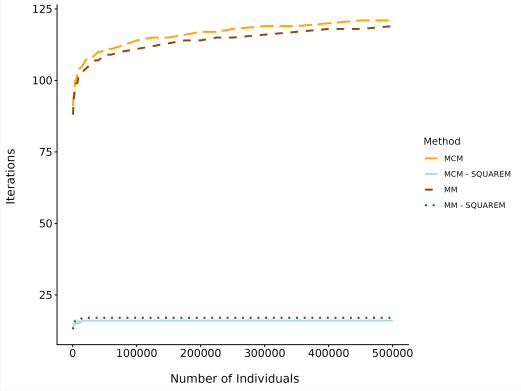
Race 1, with acceleration



With acceleration, MM beats the bespoke Newton by more than 2×: a million fixed effects in ∼1.5 minutes — from code that is a small fraction of the effort. Newton only wins races when someone spends weeks tuning it.

Race 1: does conditional maximization help?





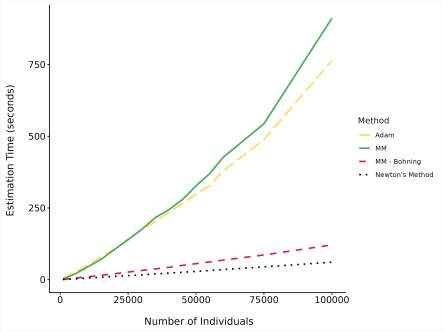
MCM needs a few more iterations than demeaned MM ($\sim$120 vs. $\sim$100) — skipping demeaning isn't free. But with one-way fixed effects, demeaning is already cheap, so the two land at nearly the same runtime; once accelerated, they're indistinguishable.

Horse race 2: everyone gets their own coefficient

Setup: $u_{ijt} = x_{it}\beta_j + w_{jt}\gamma_i + \varepsilon_{ijt}$ — replace the fixed effects with a person-specific slope γ_i on an alternative-level variable (think: everyone their own price sensitivity), $\gamma_i \sim \mathcal{N}(0, 0.5^2)$.

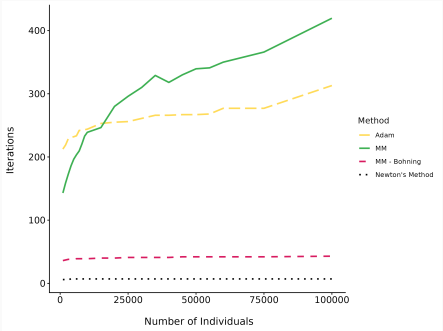
- $T = 50$ periods now, up from 20 — longer panels avoid the convergence problems caused by consumers who pick the same alternative every time.
- Up to 100,000 people this time; same contestants as Race 1 (ours and Böhning's MM, Newton, Adam, each with and without acceleration).

Race 2, without acceleration



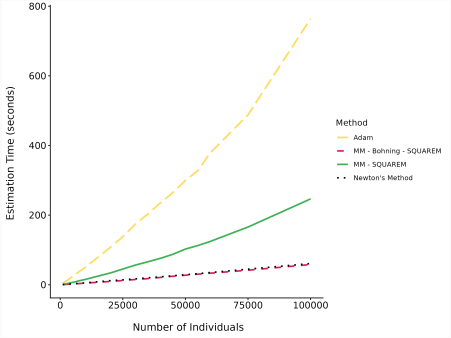
A different story here: our plain MM is now the slowest algorithm. Its scalar-identity bound ignores the dependence across alternatives that a person-specific coefficient creates, so it takes small, cautious steps to avoid overshooting. Böhning's MM, which *does* capture that dependence, stays only slightly slower than Newton.

Race 2, iteration counts



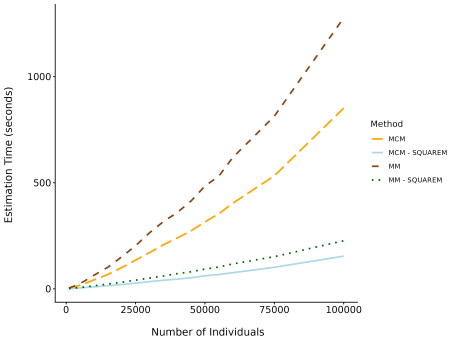
Without acceleration, plain MM's iteration count climbs with N (up to $\sim$420) — it ignores the cross-alternative correlation that matters once coefficients are person-specific. Böhning's correlation-aware bound stays flat around 30–40 iterations; Newton needs just 6. This is exactly the case where the methods genuinely differ.

Race 2, with acceleration



Acceleration rescues plain MM: it now beats Adam and stays competitive, while accelerated Böhning matches Newton. Even in its worst case, our simple MM remains a reasonable, easy-to-code option — and where dependencies across alternatives matter this much, Böhning's extra machinery earns its keep.

Race 2: does conditional maximization help more here?



MCM and MM take the same number of iterations here (the parameter blocks are independent by construction) — but MCM's simple subtraction update is cheaper than the generalized demeaning that individual-specific coefficients require, so MCM wins on time, both with and without acceleration. Conditional maximization helps most exactly when demeaning itself is the expensive step.

Application: Hospital Choice

Hospital choice and merger review

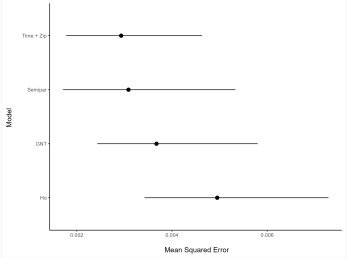
- Antitrust analysis of hospital mergers hinges on diversion ratios: if hospital A disappears, where do its patients go? Estimated from patient-level choice models.
- Influential current practice (Raval–Rosenbaum–Tenn 2017, used in litigation): split patients into many small groups, estimate a tiny logit in each — effectively 800+ hospital-group fixed effects.
- The workaround exists *because* estimating that many fixed effects jointly was infeasible. Cost: no continuous variables (travel time must be binned!), no coefficients shared across groups, no overlapping groupings.
- Our method removes the constraint: estimate the whole population jointly, with continuous travel time, many fixed effects, and anything else you like.

A natural experiment to keep us honest: the 2007 Americus tornado closed Sumter Regional Hospital (half of local admissions) for a year — so we *observe* the true diversions and can score every model's predictions against them.

Predicting where patients actually went

Four models, scored by mean squared prediction error against the observed diversions (Georgia discharge data; bootstrap confidence intervals):

- *Semipar* (Raval–Rosenbaum–Tenn 2017): hospital intercepts only, a separate logit per zip/acuity/age/diagnosis cell — 816 indicators, no continuous variables.
- *GNT* and *Ho*: travel time + hospital fixed effects + patient×hospital characteristic interactions.
- *Time+ZIP* (ours): travel time + hospital fixed effects + 3 ZIP×hospital indicators — 32 indicators total.



Time+ZIP predicts best, edging out *Semipar*: a few dozen well-chosen effects plus a continuous variable can beat hundreds of bins.

Takeaways

- Multinomial logit with millions of fixed effects is now practical: iterate simple OLS (or block-updated MCM) regressions, let linear panel tools do the heavy lifting.
- Every step provably improves the likelihood; no tuning parameters; converges to the exact MLE — with bias correction and standard errors both derivable from the same regression machinery.
- For fixed-effects models, our simple algorithm takes exactly the same steps as Böhning's classic method, provably — at lower cost; with acceleration it outruns even heavily optimized Newton ($\sim$90 seconds for a million fixed effects).
- Every method compared here — ours, Böhning's, Adam, Newton — is a preconditioned gradient step; they differ only in how much curvature they use and what that costs.
- Richer, better-predicting choice models where practice currently compromises — demonstrated for hospital merger analysis.
- The recipe likely extends to other nonlinear models with linear indices.

Code available: `imeeker.com/research` Thank you!