An MM Algorithm for Fixed Effects Multinomial Logit Models

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

September 7, 2026

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

2026 IESR-Warwick-SUFE-SDU Workshop

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

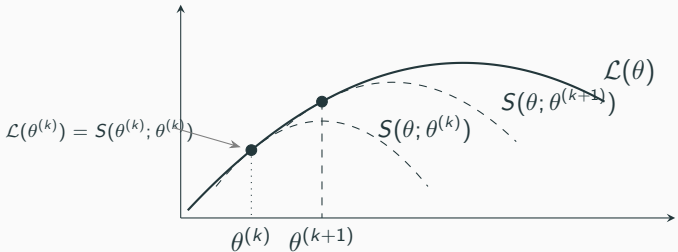
1. Use the Minorization–Maximization (MM) algorithm — a cousin of the EM algorithm — to replace the hard logit likelihood with a sequence of easy problems.
   - **EM**, the familiar recipe for latent-variable models (mixtures, random effects), guesses the missing data, averages over it (E-step), then 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 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

Choice model: utility $u_{ijt} = \psi_{ijt}(\theta) + \varepsilon_{ijt}$, linear index $\psi_{ijt}(\theta) = x_{it}\beta_j + w_{jt}\gamma_i + \alpha_{ij}$; the (concave, but hard to maximize) log-likelihood is

$$\mathcal{L}(\theta) \;=\; \sum_{i,t}\sum_{j} y_{ijt}\log p_{ijt}(\theta), \qquad p_{ijt}(\theta) \;=\; \frac{\exp\{\psi_{ijt}(\theta)\}}{\sum_m \exp\{\psi_{imt}(\theta)\}}.$$

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.

Why a regression is the right hard problem to have

Steps 5–6 of that loop are just a *linear* regression — so fifty years of linear panel machinery applies:

- **Demeaning (Frisch–Waugh–Lovell):** absorb the fixed effects instead of estimating a dummy for each — the regression keeps a handful of coefficients no matter how many fixed effects the model has.
- **Multi-way fixed effects:** person×alternative *and* alternative×time effects, using standard tools (reghdfe-style alternating projections).
- **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 happens a single time; each iteration is essentially one matrix multiply.

An alternative flavor updates parameter blocks one at a time (coefficients, then fixed effects) — even simpler per step, same guarantee, sometimes faster.

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 in the paper).
- 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.

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 (the popular machine-learning gradient optimizer we'll race against shortly), there is no learning rate to tune. (If a jump overshoots, fall back to a plain MM step to stay safe.)

- **Bias correction:** with fixed effects, short panels create the usual incidental-parameters bias; the split-panel jackknife (estimate on each half, recombine) removes the leading term — and MM's speed makes the extra estimations cheap.

Does It Work?

Horse race 1: a million fixed effects

Setup: 3 alternatives, 20 periods, up to 500,000 people ⇒ up to 1,000,000 fixed effects; 250 simulation runs per size.

Contestants:

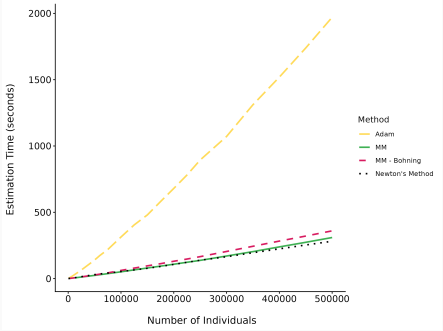
- Our MM and Böhning's MM, each with and without acceleration.
- **Newton's method** — not the off-the-shelf kind: a bespoke implementation exploiting the Hessian's sparse structure. (Off-the-shelf Stata/R: one iteration at 10,000 fixed effects can take longer than our *entire* estimation at a million.)
- **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.

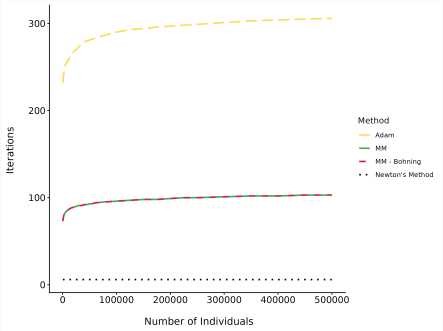
All methods land on the same estimates (differences $< 10^{-6}$) — so this is purely a race about speed and effort.

Race 1, without acceleration



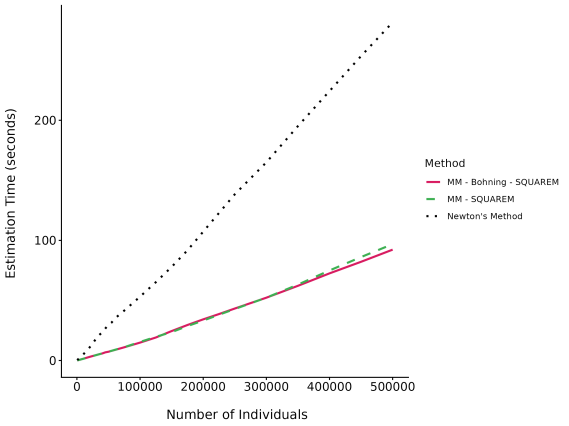
Newton wins the unaccelerated race (≈4 min at 1M fixed effects; Adam ≈30 min, over 7× slower); the MM methods sit right behind, only slightly slower than Newton. Ours and Böhning's take mathematically identical steps here — the tiny runtime gap is just the cost of Böhning's whitening step.

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 $\sim$100 iterations, confirming their mathematical equivalence; Adam needs $3\times$ as many. Cheap steps, more of them: that trade is what lets Newton win on time despite doing far more per iteration.

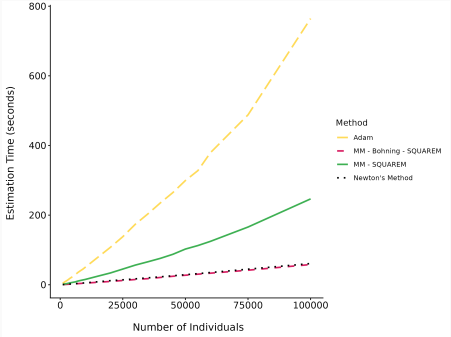
Race 1, with acceleration



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.

Horse race 2: everyone gets their own coefficient

Setup: replace the fixed effects with a person-specific slope on an alternative-level variable (think: everyone their own price sensitivity); up to 100,000 people.



This is where the algorithms differ: our simple bound ignores cross-alternative links, so plain MM slows down; accelerated MM stays competitive (beats Adam), and accelerated Böhning matches Newton. Even in its worst case, MM remains a reasonable, easy option.

Application: Hospital Choice

Hospital choice and merger review

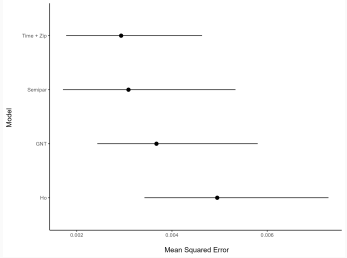
- Antitrust analysis of hospital mergers hinges on diversion ratios: if hospital A disappears, where do its patients go? Estimated from patient-level choice models.
- Influential current practice (Raval–Rosenbaum–Tenn 2017, used in litigation): split patients into many small groups, estimate a tiny logit 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 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 included.
- For fixed-effects models, our simple algorithm takes the same steps as Böhning's classic method — at lower cost; with acceleration it outruns even heavily optimized Newton ($\sim$90 seconds for a million fixed effects).
- 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!