How to Train Your Intercept

A Comparison of Intercept Update Strategies in Coordinate Descent

Johan Larsson[1] **Frederik Fabricius Bjerre**[1]

[1]Department of Mathematical Sciences, University of Copenhagen

Date published: 2026-08-30 Last modified: 2026-08-30

Abstract

Coordinate descent solvers for regularized generalized linear models (GLMs) differ in how they update the intercept. This seemingly routine choice can substantially affect convergence when the response is imbalanced. For direct coordinate descent on the original GLM loss, we compare a conservative step based on worst-case curvature, a safeguarded Newton step based on current curvature, and repeated Newton steps that solve the intercept subproblem. The conservative update can leave much of the interaction between the intercept and the coefficients unresolved, producing a slowdown when that interaction controls convergence. A single Newton step avoids this attenuation and, in our experiments, follows nearly the same outer trajectory as exact minimization while requiring fewer inner steps. Simulated and real-data experiments, including comparisons across six production solvers and a controlled intervention within skglm, support this explanation. We also show how the same mechanism appears in quadratic-approximation methods and extends to multinomial models with rare classes. We recommend a safeguarded Newton update for direct coordinate descent and full intercept optimization within frozen local quadratic surrogates.

Keywords: intercepts, coordinate descent, optimization, regularization

submitted

Contents

1 Introduction **2**

2 Related work **5**

3 Theory **6**

3.1 Coordinate descent . . . 7

3.2 What sets the intercept apart . . . 8

3.3 Cross-influence of intercept and coefficients . . . 9

3.3.1 A normalized measure of intercept–coefficient coupling . . . 10

3.3.2 The profiled objective . . . 10

3.3.3 Strategies as Schur approximations . . . 11

3.3.4 Local-rate interpretation . . . 14

3.3.5 A safe Newton variant . . . 14

3.3.6 Extension to vector intercepts . . . 14

3.4 Connection to IRLS . . . 15

*Corresponding author: johan@jolars.co

4 Results **18**
4.1 Methodology and reproducibility . . . 18
4.2 Simulated data . . . 18
4.2.1 Synthetic trajectories and safeguards . . . 18
4.3 Real data . . . 20
4.4 Robustness and computational cost . . . 22
4.5 Imbalance and regularization . . . 22
4.6 Production solvers . . . 23
4.6.1 Additional solver diagnostics . . . 27

5 Discussion **27**

Supplementary material **29**
S1. Technical derivations and rate diagnostics . . . 29
S1.1 Rate heuristic and local-rate validation . . . 29
S2. Vector intercepts and multinomial models . . . 34
S2.1 Derivation and experiments . . . 34
S2.2 Full imbalance–amplitude sweep . . . 38
S3. Experimental methods and reproducibility . . . 38
S4. Robustness analyses . . . 41
S4.1 Cyclic versus permuted ordering . . . 41
S4.2 Warm-started path . . . 41
S4.3 Per-pass cost decomposition . . . 43
S4.4 Full warm-start grids . . . 45
S5. Production-solver diagnostics . . . 46
S5.1 Poisson family . . . 46
S5.2 Single-lambda cold start . . . 47

References **48**

1 Introduction

Popular solvers for regularized generalized linear models (GLMs) agree on coordinate descent but disagree about one coordinate: the intercept. GLMs usually include this coordinate to represent the baseline response and to absorb constant shifts in the predictors without changing the fitted slopes. It is unpenalized and never drops out of the fitted model, yet solvers update it in different ways. Research on coordinate descent has examined acceleration, parallelization, and convergence rates, but has largely treated the intercept update as a routine implementation choice. That choice can dominate convergence speed when the response is imbalanced. For direct coordinate descent, a single safeguarded Newton step per pass removes this slowdown in our experiments.

The disagreement follows a deeper split among implementations. Some solvers, including skglm[2] (Bertrand et al. 2022), run coordinate descent directly on the original GLM loss. Others repeatedly replace that loss with a quadratic approximation and solve the approximation by coordinate descent. A local quadratic already contains the current intercept curvature, so the update strategies coincide within it. A quadratic built from a worst-case curvature bound retains the same problem as the conservative direct update. The six production solvers in Section 4.6 instantiate these designs; we explain in Section 3 and Section 3.4 when the distinction matters.

[2]Among the surveyed solvers, only skglm's intercept update is version-dependent: version 0.5 and earlier take the gradient step described here, while a later patch replaces it with a Newton step (Section 4.6). We pin the version throughout wherever this behavior is at issue.

submitted

We study the optimization problem

$$\underset{\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p}{\text{minimize}} \Big(P(\beta_0, \beta) = F(\beta_0, \beta) + G_\lambda(\beta) \Big) \tag{1}$$

where P is the primal objective, F is the loss, and G_λ penalizes large coefficients while leaving the intercept unpenalized. Throughout the paper, we assume that G_λ is a proper, closed, convex, separable penalty and that F is the per-observation-averaged loss

$$F(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^{n} f(x_i^\top \beta + \beta_0; y_i),$$

where f is smooth, twice differentiable, and convex. Averaging by n keeps the global Lipschitz constant for the intercept, $L_0 = \sup f''$, independent of n. It also matches the objective scaling of the packaged solvers that we compare in Section 4.6: $(1/n) \cdot \text{loss} + \lambda \|\beta\|_1$. For brevity, we write x_i for the ith row of the design matrix X and x_j for its jth column, and we let $(\hat{\beta}_0, \hat{\beta})$ denote a solution to Equation 1. Under these assumptions, P is a convex composite objective. The assumptions cover every penalty and model in our analysis and experiments; we do not consider nonconvex penalties.

Within this problem, we compare three ways to update the intercept: a gradient step based on a global curvature bound, a Newton step based on the curvature at the current iterate, and exact minimization of the intercept subproblem. They can converge at substantially different rates when the response is imbalanced—for example, when positive cases are rare in logistic regression.

For the lasso experiments, $\lambda_{\max}$ is the smallest penalty for which all fitted slopes are zero. In the synthetic experiments, μ_0 denotes the baseline mean response at $x = 0$—a positive-class probability for binomial models and a rate for Poisson models—and s is the number of nonzero slopes in each generated coefficient vector. Once the slopes enter the linear predictor, μ_0 need not equal the empirical response mean.

Gradient Strategy Takes a single gradient step once per pass over the coordinates.

Newton Strategy Takes a single Newton step, wrapped in an Armijo backtracking line search, once per pass over the coordinates. The line search usually accepts the full step ($\alpha = 1$) and backtracks when the undamped step would overshoot.

Exact Strategy Iterates Armijo-safeguarded Newton steps on the intercept subproblem until it meets a tight numerical convergence criterion. It raises an error rather than return an iterate outside its documented residual bounds.

In the analysis, *exact* means setting the intercept to its conditional minimizer. In the experiments, the Exact Strategy computes this minimizer numerically. We document its tolerances, iteration cap, and scalar floating-point margin in Supplement S3. Every Exact Strategy result reported in our experiments therefore met the specified conditional-solve criterion; none is an uncontrolled capped approximation.

The three paths in Figure 1 trace the strategies across the level sets of Equation 1 for a toy problem with one feature and an intercept. The gradient strategy approaches the solution through many small steps, whereas the Newton and exact strategies reach it in only a few.

In Figure 2, we compare how quickly the three strategies converge on ℓ_1-regularized logistic regression on the w1a dataset (Platt 1998). The Newton and exact strategies bring the relative suboptimality bound below 10^{-8} after roughly 45 passes, while the gradient strategy takes more than seven times as many.

We make four contributions:

submitted

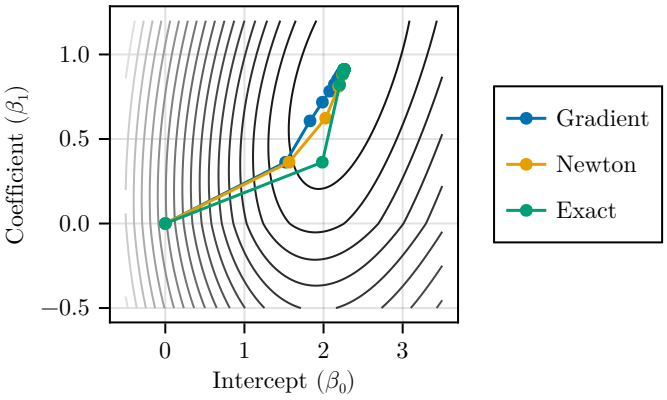


Figure 1: Intercept-update trajectories on a toy logistic problem. The paths show the iterates of the gradient, Newton, and exact strategies over the objective level sets for a one-feature model with an intercept.

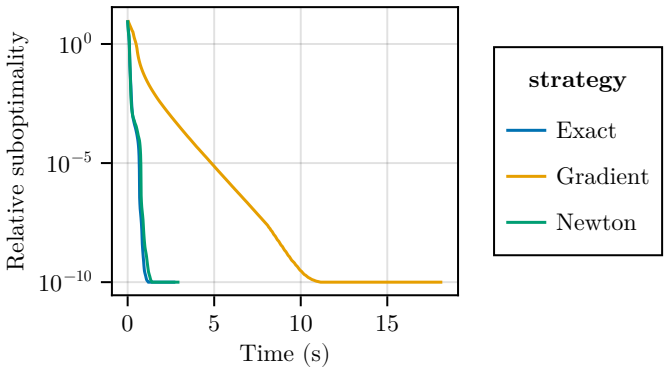


Figure 2: Overall convergence on w1a. The plot shows a relative suboptimality bound versus time for the three intercept strategies in ℓ_1-regularized logistic regression.

Submitted

- We formalize three strategies for updating an unpenalized intercept: gradient, Newton, and exact minimization.
- Using a Schur-complement analysis, we show that the mismatch between local intercept curvature and a global bound explains the gradient strategy's slowdown. In strongly coupled problems, the predicted gradient-to-Newton rate ratio approaches the ratio of global to local curvature.
- We classify CD-based GLM solvers by their treatment of curvature and the intercept, and we evaluate six production solvers using controlled comparisons where possible.
- We extend the analysis to multinomial logistic regression, where rare classes create low-curvature directions. Experiments on synthetic and real data show that Newton updates closely match exact minimization and can be much faster than gradient updates.

2 Related work

The question of how to update the intercept touches three strands of the literature, none of which has analyzed it as a separate problem. The first concerns how CD-based GLM solvers incorporate the intercept. Existing work generally treats this as a routine implementation choice. The glmnet literature (Friedman et al. 2010; Tay et al. 2023), for example, describes two weight schemes for the logistic IRLS inner solver: local weights and a global 1/4 upper bound. The software exposes them through `type.logistic = "Newton"` and `"modified.Newton"`, which the documentation presents as a speed trade-off rather than as a choice of intercept strategy.

LIBLINEAR's newGLMNET solver (Fan et al. 2008; Yuan et al. 2012) treats the intercept as the unregularized last coordinate of each prox-Newton subproblem. Coordinate descent then optimizes the intercept along with the coefficients. By the end of the inner solve, the intercept has reached its conditional minimum for the quadratic; the solver has no separate intercept loop on the original loss.[3]

BlitzL1 (Johnson and Guestrin 2015) does the same inside the prox-Newton inner CD, and additionally runs a capped one-dimensional Newton loop on the intercept on the original loss after each subproblem, to gradient tolerance.[4] celer (Massias et al. 2020) does not support intercepts for ℓ_1-logistic regression. Implementations therefore make varied choices, but previous work has not directly analyzed the response distributions for which those choices matter.

A second thread is the partialling-out/Frisch–Waugh–Lovell tradition. In linear regression, the intercept can be sidestepped: centering the features decouples it from the slopes and, when the intercept is left unpenalized, gives $\hat{\beta}_0 = \bar{y}$ (Lovell 1963). glmnet leaves the intercept unpenalized on translation-invariance grounds—penalizing β_0 would make the path depend on the arbitrary location of y—and this decoupling makes that choice essentially free for the Gaussian loss case. After solving for the slopes, the intercept is recovered as $\hat{\beta}_0 = \bar{y} - \bar{x}^\top \hat{\beta}$.

For non-Gaussian GLMs, the decoupling fails because curvature varies across observations. Ordinary feature centering no longer removes the interaction between the intercept and the slopes, and the closed form for $\hat{\beta}_0$ disappears. adelie (Yang and Hastie 2024) handles this dependence through its

[3] Source: https://github.com/cjlin1/liblinear/blob/491c9f1/linear.cpp#L1972-L1977. `solve_l1r_lr` (`linear.cpp:1796`) appends the intercept as a constant feature and skips it from regularization (`linear.cpp:1789-1790`); inside the inner CD QP solve the intercept update is the pure Newton step `z = -G/H` (`linear.cpp:1972-1977`), and the outer Newton loop closes without any separate intercept loop on the original loss.

[4] Source: https://github.com/tbjohns/BlitzL1/blob/aef8d02/src/solver.cpp#L27-L55. The loop is called from `run_prox_newton_iteration` (`solver.cpp:262`) after the working-set CD pass and step-size backtracking.

Submitted

proximal quasi-Newton approximation. At each linearization, it builds a weighted Gaussian surrogate and absorbs the intercept into that surrogate rather than treating it as a separate CD coordinate.[5]

For Gaussian data, the framework in Section 3 reduces to the familiar result: the loss has constant curvature, so the gradient and Newton strategies coincide on the intercept.

The third thread is majorization–minimization (MM). The constant $L_0 = 1/4$, which skglm uses as its global per-coordinate Lipschitz step, is the binary specialization of the quadratic Hessian majorant used by Böhning (1992) to build an MM algorithm for multinomial logistic regression. Hunter and Lange (2004) give the canonical exposition of MM. Krishnapuram et al. (2005) apply Böhning's bound to L1-penalized multinomial logistic regression, a direct precedent for our multinomial extension.

In regularized logistic regression, the same construction lives on as the `modified.Newton` mode of glmnet and the `MM` mode of biglasso (Zeng and Breheny 2021) (Table 2). Because the global majorant does not adapt to local curvature, it can produce a much smaller intercept correction than a Newton step when the response is imbalanced. In multinomial regression, the same mismatch appears along directions associated with rare classes. The MM literature treats this as loose majorization; we quantify the resulting slowdown for imbalanced binary and rare-class multinomial problems (Section 3.4, Figure 22).

3 Theory

The three strategies differ in how they respond to intercept curvature. The gradient strategy uses a global upper bound, Newton uses the curvature at the current iterate, and the exact strategy minimizes the intercept subproblem. On imbalanced data, the local curvature can be much smaller than the global bound. When the intercept—or a direction coupling it with a coefficient—controls convergence, this mismatch can make the gradient strategy slower by a factor proportional to the ratio between the two. Below, we distinguish the local result that we prove from a bound-based heuristic and the behavior that we observe experimentally.

Throughout this section, we work with the per-observation-averaged, canonical-link GLM loss

$$F(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^{n} f(x_i^\top \beta + \beta_0; y_i).$$

We first derive the scalar-intercept results for the Gaussian, binomial, and Poisson models. We later extend them to multinomial logistic regression, whose intercept is a vector. For observation i, let $\eta_i = x_i^\top \beta + \beta_0$ denote the linear predictor, and let g^{-1} denote the inverse link. The conditional mean is then $\mathrm{E}(y_i \mid \eta_i) = g^{-1}(\eta_i)$. For the canonical-link models considered here, the derivative of the negative log-likelihood simplifies to the *generalized residual*,

$$f'(\eta_i; y_i) = g^{-1}(\eta_i) - y_i = r_i.$$

All gradients below follow from this canonical-link identity, and all Hessians from its derivative. The identity does not hold for arbitrary noncanonical links. The loss, link, and inverse link for the four models we study—Gaussian, binomial, Poisson, and multinomial—appear in Table 1.

[5] Source: https://github.com/JamesYang007/adelie/blob/d6a1229/adelie/src/include/adelie_core/solver/solver_gaussian_pin_naive.hpp. The inner Gaussian solver uses IRLS-weighted column means in its gradient update (`solver_gaussian_pin_naive.hpp:87, 120`) and recovers the intercept at the end of the inner solve as the weighted mean of the working response plus the residual mean (`solver_gaussian_pin_naive.hpp:392`); the column means themselves are built with the IRLS weights in the outer GLM solver (`solver_glm_naive.hpp:366`).

submitted

Table 1: Loss, link, and inverse-link functions for the GLM families studied. The multinomial row uses the reference-class parameterization with class m as the reference ($\eta_m = 0$, $\mu_m = 1 - \sum_{j<m} \mu_j$), so η and μ in that row are $(m-1)$-vectors and log / exp are applied elementwise.

Model	$f(\eta; y)$	$g(\mu)$	$g^{-1}(\eta)$
Gaussian	$\frac{1}{2}(y-\eta)^2$	μ	η
Binomial	$\log(1+e^{\eta}) - \eta y$	$\log\left(\frac{\mu}{1-\mu}\right)$	$\frac{e^{\eta}}{1+e^{\eta}}$
Poisson	$e^{\eta} - \eta y$	$\log(\mu)$	e^{η}
Multinomial	$\log\left(1+\sum_{j=1}^{m-1} e^{\eta_j}\right) - \sum_{k=1}^{m-1} y_k \eta_k$	$\log\left(\frac{\mu}{1-\sum_{j=1}^{m-1} \mu_j}\right)$	$\frac{\exp(\eta)}{1+\sum_{j=1}^{m-1} e^{\eta_j}}$

3.1 Coordinate descent

Coordinate descent updates one coordinate at a time while holding the others fixed. For coefficient β_j, the chain rule gives

$$\frac{\partial}{\partial \beta_j} F(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^{n} x_{ij} f'(\eta_i; y_i).$$

The smooth part of a coefficient or intercept update takes a step in the negative partial-gradient direction,

$$\beta_j^{+} = \beta_j - \gamma_j \frac{\partial}{\partial \beta_j} F(\beta_0, \beta),$$

where γ_j is the step size. The usual CD choice is a Newton step, based on the second derivative

$$H_{jj} = \frac{\partial^2}{\partial \beta_j^2} F(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2 f''(\eta_i; y_i).$$

Letting $w_i = f''(\eta_i; y_i)$ be the *weights*, we can rewrite the Hessian diagonal as

$$H_{jj} = \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2 w_i.$$

This yields the step size

$$\gamma_j = H_{jj}^{-1} = \left(\frac{1}{n} \sum_{i=1}^{n} x_{ij}^2 w_i \right)^{-1}.$$

Applying the same logic to the intercept, define its local curvature as

$$H_{00} = \frac{\partial^2}{\partial \beta_0^2} F(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^{n} f''(\eta_i; y_i).$$

The corresponding Newton step size is $\gamma_0 = H_{00}^{-1}$. Production solvers, however, realize three different intercept updates:

submitted

1. A Newton step, setting γ_0 as above.
2. A gradient descent step, setting γ_0 to the reciprocal of the global Lipschitz constant with respect to β_0 or choosing it by a backtracking line search.
3. Repeated safeguarded Newton steps on the intercept subproblem, iterated until the scalar convergence contract described below is met—the *exact strategy*.

For a regular coefficient, this smooth step must also account for the penalty. By separability, write $G_\lambda(\beta) = \sum_{j=1}^{p} g_{\lambda,j}(\beta_j)$ and define

$$\operatorname{prox}_{\gamma g}(v) = \operatorname*{argmin}_{u \in \mathbb{R}} \left\{ g(u) + \frac{1}{2\gamma}(u - v)^2 \right\}.$$

The coordinate update is therefore the proximal-gradient step

$$\beta_j^+ = \operatorname{prox}_{\gamma_j g_{\lambda,j}} \left(\beta_j - \gamma_j \frac{\partial}{\partial \beta_j} F(\beta_0, \beta) \right).$$

For the lasso, $g_{\lambda,j}(u) = \lambda|u|$, this is soft-thresholding at $\gamma_j\lambda$. The intercept has no corresponding proximal step because it is unpenalized.

The resulting coordinate-descent algorithm appears in Algorithm 1.

In the implementation, every coefficient update in Algorithm 1 is wrapped in a Tseng–Yun Armijo backtrack on the proximal model decrease, so each pass is descent by construction. The pseudocode omits this guard for readability. The Newton branch likewise backtracks only when the full step overshoots, and the exact branch enforces the residual contract described above. Supplement S3 gives the constants, iteration caps, and numerical tolerances.

3.2 What sets the intercept apart

Four properties distinguish the intercept β_0 from a regular coefficient β_j:

1. The intercept's design column is $\mathbf{1}_n$. Treating β_0 as a coordinate is equivalent to augmenting X with a constant column of ones. Consequently, $H_{00} = (1/n)\sum_i w_i$ is a mean of weights rather than a weighted mean of squared features. The global Lipschitz constant $L_0 = \|\mathbf{1}\|^2 f''_{\max}/n = f''_{\max}$ does not shrink with the data; for binary logistic regression, $L_0 = 1/4$. Finally, the cross-Hessian $H_{0j} = (1/n)\sum_i w_i x_{ij}$ is the w-weighted column mean. The intercept therefore couples to every slope through a single scalar rather than through pairwise feature correlations.
2. The intercept is unpenalized. The regularizer G_λ acts only on β, so β_0 is the only coordinate whose update is smooth—no proximal operator, no soft-thresholding, no non-differentiable optimality condition. Hence, the strategies in Algorithm 1 can update β_0 directly.
3. The intercept is never inactive. State-of-the-art CD solvers use screening rules and working sets to limit the number of coordinates touched each pass. In high-dimensional settings, most β_j are touched in only a fraction of passes, and their average per-pass cost is correspondingly small. The intercept is in the active set of every pass by construction, so its per-pass treatment dominates the overall intercept-update cost, unlike that of any single slope.
4. In IRLS-CD, optimizing the intercept re-centers the quadratic surrogate by setting the weighted mean of its residuals to zero. No individual slope performs this role. adelie can therefore absorb the intercept into a weighted-centered parameterization of the inner Gaussian problem rather than update it as a separate coordinate. Section 3.4 derives this identity.

The first property explains why L_0/H_{00} is unbounded under response imbalance: L_0 does not adapt to the curvature at the current iterate, while H_{00} collapses with $(1/n)\sum_i w_i$ as the linear predictor

submitted

Algorithm 1 Cyclic coordinate descent.

Input: Data (X, y), $\lambda > 0$, initial intercept and coefficients (β_0, β)
while not converged **do**
  for j in $\{1, 2, ..., p\}$ **do**
    $\beta_j \leftarrow \mathrm{prox}_{\gamma_j g_{\lambda,j}}\left(\beta_j - \gamma_j \frac{\partial}{\partial \beta_j} F(\beta_0, \beta)\right)$
  end for
  if intercept strategy is "Newton" **then**
    $\delta \leftarrow -\left(\frac{\partial^2}{\partial \beta_0^2} F(\beta_0, \beta)\right)^{-1} \frac{\partial}{\partial \beta_0} F(\beta_0, \beta)$
    $\alpha \leftarrow 1$
    while $F(\beta_0 + \alpha\delta, \beta) > F(\beta_0, \beta) + c\alpha\delta \frac{\partial}{\partial \beta_0} F(\beta_0, \beta)$ **do**
      $\alpha \leftarrow \alpha/2$
    end while
    $\beta_0 \leftarrow \beta_0 + \alpha\delta$
  else if intercept strategy is "Gradient" **then**
    $\beta_0 \leftarrow \beta_0 - \frac{1}{L_0} \frac{\partial}{\partial \beta_0} F(\beta_0, \beta)$
  else if intercept strategy is "Exact" **then**
    $k \leftarrow 0$
    while $|\partial_0 F| > 10^{-10}$ and $k < 50$ **do**
      $\delta \leftarrow -\left(\frac{\partial^2}{\partial \beta_0^2} F(\beta_0, \beta)\right)^{-1} \frac{\partial}{\partial \beta_0} F(\beta_0, \beta)$
      choose α by Armijo backtracking as in the Newton branch
      $\beta_0 \leftarrow \beta_0 + \alpha\delta$
      $k \leftarrow k + 1$
    end while
    if $k = 50$ and $|\partial_0 F| > 5 \times 10^{-10}$ **then**
      raise a convergence error
    end if
  end if
end while
Output: (β_0, β)

saturates. The second and third explain why solvers can choose among several strategies and why the cost of that choice persists on every pass. The fourth explains why an IRLS solver can absorb the intercept into its quadratic surrogate.

3.3 Cross-influence of intercept and coefficients

The Hessian of F with respect to the intercept and the coefficients exposes how the intercept couples to each coefficient. Letting $w_i = f''(\eta_i; y_i)$, we have

$$H_{jk} = \frac{\partial^2}{\partial \beta_j \partial \beta_k} F(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^{n} x_{ij} x_{ik} w_i$$

for the coefficient–coefficient block and

$$H_{0j} = \frac{\partial^2}{\partial \beta_0 \partial \beta_j} F(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^{n} x_{ij} w_i$$

for the intercept–coefficient block. The coefficients couple to each other *pairwise* through the weighted Gram matrix of the features. Correlated features produce large $|H_{jk}|$, which is the classical

submitted

reason coordinate descent struggles on such designs. The intercept instead couples to each coefficient *globally* through the weighted mean of that feature, without an explicit pairwise-correlation term. This distinction is structural rather than probabilistic: outside squared loss, the weights depend on the full fitted predictor and therefore indirectly on the joint design.

For squared loss, $w_i = 1$ and $H_{0j} = (1/n)\sum_i x_{ij}$, so centering the features makes $H_{0j} = 0$ and the intercept decouples from the coefficients entirely. For other losses the weights depend on the linear predictor and the response, so H_{0j} is a w_i-weighted mean of x_{ij}, which is generally nonzero even after centering.

3.3.1 A normalized measure of intercept–coefficient coupling

To quantify how strongly the intercept and the jth coefficient are coupled, suppose that $H_{00} > 0$ and $H_{jj} > 0$ and define the normalized cross-curvature

$$\rho_{0j} = \frac{H_{0j}}{\sqrt{H_{00}H_{jj}}} \in [-1, 1], \tag{2}$$

which is invariant to rescaling of the features and the loss function, and which the Cauchy–Schwarz inequality bounds in $[-1, 1]$. A zero-curvature intercept direction does not admit a local Newton step, and the normalization is undefined when either diagonal curvature is zero; we exclude those directions when using ρ_{0j}. The two extremes are easy to interpret in the local quadratic model: $\rho_{0j} = 0$ means the coordinates are locally decoupled, while $|\rho_{0j}| \to 1$ means a change in β_0 is almost fully absorbed by an opposite change in β_j (or vice versa), making the two coordinates nearly degenerate along that direction.

3.3.2 The profiled objective

To see why ρ_{0j} controls the impact of the intercept update strategy, we consider the *profiled* objective, obtained by optimizing the intercept out:

$$\tilde{F}(\beta) = \min_{\beta_0} F(\beta_0, \beta).$$

Evaluate the Hessian blocks at $(\beta_0^\star(\beta), \beta)$. When $H_{00} > 0$, the Hessian of $\tilde{F}$ is the Schur complement

$$\tilde{H}_{jk} = H_{jk} - \frac{H_{0j}H_{0k}}{H_{00}}, \tag{3}$$

and in particular

$$\tilde{H}_{jj} = H_{jj} - \frac{H_{0j}^2}{H_{00}} = H_{jj}(1 - \rho_{0j}^2). \tag{4}$$

When $|\rho_{0j}|$ is close to 1, the profiled curvature $\tilde{H}_{jj}$ is much smaller than H_{jj} (Equation 4): most of the diagonal curvature seen in H is borrowed through the intercept and disappears once we profile the intercept out. At the first coefficient update after re-optimizing the intercept, coordinate descent sees the gradient of $\tilde{F}$, although its diagonal step still uses H_{jj}. If the intercept remains stale, the update instead sees the partial gradient of F. In the local quadratic models, the factor ρ_{0j}^2 measures the gap between the profiled curvature $\tilde{H}_{jj}$ and the unprofiled curvature H_{jj}.

submitted

3.3.3 Strategies as Schur approximations

We can now classify the three intercept-update strategies by how completely they remove the intercept–coefficient coupling in a local model. We track the *gradient response* of the coefficient subproblem to an intercept step. If we take the step $\beta_0 \mapsto \beta_0 - \alpha\partial_0 F$, twice continuous differentiability gives

$$\partial_j F(\beta_0 - \alpha\partial_0 F, \beta) = \partial_j F(\beta_0, \beta) - \alpha H_{j0}\partial_0 F(\beta_0, \beta) + r_j(\alpha), \tag{5}$$

where $r_j(\alpha) = o(|d_0|)$ as the intercept displacement $d_0 = -\alpha\partial_0 F$ tends to zero. If H_{j0} is locally Lipschitz, then $r_j(\alpha) = O(d_0^2)$. The remainder vanishes for a frozen quadratic.

When $H_{00} > 0$, the local Schur correction $\partial_j F - (H_{j0}/H_{00})\partial_0 F$ is the gradient that a Newton intercept step predicts for the profiled objective at the current β. It equals the profiled gradient in a frozen quadratic and approximates it locally for a nonlinear loss; conditional minimization gives the true profiled gradient. The three strategies differ in how closely the gradient response in Equation 5 approaches that target.

At the intercept's conditional minimizer, the next coefficient update uses the profiled gradient, and its diagonal curvature gives a conservative step relative to the profiled local quadratic model. More precisely:

Proposition 3.1. *(First-update profile equivalence.) Let F be twice continuously differentiable and strictly convex in β_0 for every β, and let $\tilde{F}(\beta) = \min_{\beta_0} F(\beta_0, \beta)$. Suppose the exact strategy has driven β_0 to the conditional minimizer $\beta_0^{(t)} = \arg\min_{\beta_0} F(\beta_0, \beta^{(t)})$, and suppose $H_{00} > 0$ and $\tilde{H}_{jj} > 0$ there. Then the* first *coefficient update of the next pass uses the gradient $\partial_j F(\beta_0^{(t)}, \beta^{(t)}) = \partial_j \tilde{F}(\beta^{(t)})$, so its update direction agrees with the corresponding coordinate direction for $\tilde{F}$ at $\beta^{(t)}$. Its diagonal step size remains $1/H_{jj}$. Because $\tilde{H}_{jj} \le H_{jj}$ by Equation 4, this step is no larger than the step $1/\tilde{H}_{jj}$ that minimizes $\tilde{F}$'s local quadratic model along e_j. This comparison concerns the two quadratic models at the current point; without a coordinate-curvature bound along the finite step, it does not imply descent of $\tilde{F}$ itself.*

Proof. Let $\beta_0^\star(\beta) = \arg\min_{\beta_0} F(\beta_0, \beta)$, which is well-defined and differentiable because F is strictly convex in β_0, so that $\tilde{F}(\beta) = F(\beta_0^\star(\beta), \beta)$. Differentiating through $\beta_0^\star$ and using the stationarity condition $\partial_0 F(\beta_0^\star(\beta), \beta) = 0$,

$$\partial_j \tilde{F}(\beta) = \partial_j F(\beta_0^\star(\beta), \beta) + \partial_0 F(\beta_0^\star(\beta), \beta)\partial_j \beta_0^\star(\beta) = \partial_j F(\beta_0^\star(\beta), \beta).$$

This is the envelope identity: stationarity sets the indirect term through $\beta_0^\star$ to zero. Evaluating at $\beta_0^{(t)} = \beta_0^\star(\beta^{(t)})$ gives the gradient equality. For the step size, $\tilde{H}_{jj} = H_{jj}(1-\rho_{0j}^2) \le H_{jj}$, so $1/H_{jj} \le 1/\tilde{H}_{jj}$ (Equation 4). The realized step is therefore at most the minimizing step $1/\tilde{H}_{jj}$ in $\tilde{F}$'s local quadratic model along e_j. The inequality alone supplies no finite-step descent guarantee for $\tilde{F}$. □

At the start of each pass, the exact strategy therefore gives the first coefficient update the gradient of $\tilde{F}$ and a step conservative relative to its local quadratic model. The gradient identity lasts for only one coefficient update. By the second update, $\beta_0^{(t)}$ is no longer the conditional minimizer, and the identity breaks.

To see how quickly it breaks, expand $\partial_0 F$ to first order in the β_j direction at the conditional minimizer. With $\partial_{0j} F = H_{0j}$ and $\partial_0 F(\beta_0^{(t)}, \beta^{(t)}) = 0$, a single coefficient update $\beta_j^{(t)} \mapsto \beta_j^{(t)} + \delta_j$ produces

$$\partial_0 F\big(\beta_0^{(t)}, \beta^{(t)} + \delta_j e_j\big) = H_{0j}\delta_j + O(\delta_j^2). \tag{6}$$

After one coordinate update of pass $t+1$, the intercept gradient is no longer zero, and the conditional-minimizer property that Proposition 3.1 depends on is lost.

submitted

This short-lived optimum limits the benefit of the exact strategy. Let k_0 denote the number of inner work units used at the end of a pass to drive $|\partial_0 F| < \varepsilon$, counting the mandatory residual check as one unit when no Newton direction is needed. The intercept subproblem $\beta_0 \mapsto F(\beta_0, \beta)$ is a smooth, strongly convex one-dimensional problem near any minimizer with $H_{00} > 0$. Once an inner iterate enters a neighborhood where the usual Newton conditions hold, its remaining iteration count scales as $O(\log\log(1/\varepsilon))$. This local result does not bound the steps needed to reach that neighborhood, nor does it show that every outer-pass subproblem begins there. Across the imbalance levels in our experiments, $k_0 \in [1, 5]$, and each step costs $O(n)$ work (Figure 29).

Only the first coefficient update receives the full benefit of exact profiling; by the second, the intercept gradient has drifted by $H_{0j}\delta_j$ (Equation 6). This suggests that the $k_0 - 1$ additional inner Newton steps will produce little improvement in outer convergence over a single Newton step. The experiments in Supplement S4.3 support this expectation on our test problems; we do not claim a global complexity bound for the exact strategy. The extra work should matter most when several $|H_{0j}|$ are large, because the next coordinate sweep then perturbs $\partial_0 F$ immediately.

Three regimes make the extra intercept iterations easier to justify.

First, along a warm-started λ path, each subproblem begins with $|\partial_0 F|$ already small. Then k_0 drops to one or two inner Newton steps, the $k_0 - 1$ penalty shrinks to zero, and the Newton and exact strategies produce nearly identical updates.

Second, in strongly coupled designs, the local envelope alignment of Proposition 3.1 suggests a first-update gain that can partially offset the per-pass overhead, though the cyclic drift of Equation 6 still applies.

Third, at a prox-Newton or IRLS linearization boundary, the solver recomputes the Hessian for the next subproblem, so the within-pass drift no longer matters. This case explains why LIBLINEAR fully resolves the intercept of its frozen quadratic and why BlitzL1 goes further, applying an exact-strategy loop to the original loss after the prox-Newton step (Section 3.4).

In the first two regimes, the exact strategy on the original loss is defensible. At a linearization boundary, fully resolving the quadratic surrogate is the analogous choice. Our recommendation against extra intercept iterations applies only to the cold-start, direct-CD setting outside these three regimes.

A single Newton step on the intercept matches the conditional minimizer at leading order. Write $g(t) = \partial_0 F(t, \beta)$, let $\beta_0^* = \beta_0^*(\beta)$ satisfy $g(\beta_0^*) = 0$, and set $H_{00} = g'(\beta_0)$. Taylor-expanding $g(\beta_0^*)$ around the current iterate β_0 gives

$$\partial_0 F(\beta_0, \beta) = H_{00}(\beta_0 - \beta_0^*) - \tfrac{1}{2}\partial_{000} F(\xi)(\beta_0 - \beta_0^*)^2, \tag{7}$$

for some ξ between β_0 and $\beta_0^*(\beta)$. Dividing by H_{00} and recognizing $\beta_0^{\text{Newton}} = \beta_0 - \partial_0 F / H_{00}$ on the left rewrites Equation 7 as the standard iterate-space Newton identity

$$|\beta_0^{\text{Newton}} - \beta_0^*(\beta)| = \frac{|\partial_{000} F(\xi)|}{2|H_{00}|}(\beta_0 - \beta_0^*)^2. \tag{8}$$

For a local bound, suppose that $0 < m \le g'(t) \le L$ and $|g''(t)| \le M$ between β_0 and β_0^*. The mean-value theorem gives $m|\beta_0 - \beta_0^*| \le |g(\beta_0)| \le L|\beta_0 - \beta_0^*|$, so Equation 8 yields

$$|\beta_0^{\text{Newton}} - \beta_0^*(\beta)| \le \frac{M}{2m^3}|\partial_0 F(\beta_0, \beta)|^2. \tag{9}$$

Here M may be taken as a bound on $|\partial_{000} F| = |(1/n)\sum_i f'''(\eta_i)|$, and hence as no larger than $\sup|f'''|$. For squared loss, $f''' \equiv 0$ and the Newton step is exact; for logistic regression $|f'''|$ is bounded by a

submitted

small constant. Thus, wherever the intercept curvature stays bounded away from zero, the residual shrinks quadratically as the current intercept gradient vanishes.

What matters next is whether the Newton residual harms the first coefficient update. Let $r_0 = \beta_0^{\text{Newton}} - \beta_0^*(\beta)$ be the Newton residual, with $|r_0|$ bounded by Equation 9. Taylor-expanding $\partial_j F$ in β_0 around the conditional minimizer $\beta_0^*(\beta)$, using $\partial_j F(\beta_0^*, \beta) = \partial_j \tilde{F}(\beta)$ and $H_{j0}^* = \partial_{0j} F(\beta_0^*, \beta)$, yields

$$\partial_j F\big(\beta_0^{\text{Newton}}, \beta\big) = \partial_j \tilde{F}(\beta) + H_{j0}^* r_0 + O(r_0^2), \tag{10}$$

i.e. the profiled gradient plus a bias of order $H_{j0}^* r_0$. The Newton strategy is therefore as good as the exact strategy on a coordinate j whenever

$$|H_{j0}^*| \cdot |r_0| \ll \big|\partial_j \tilde{F}(\beta)\big|. \tag{11}$$

Under the same local conditions as Equation 9, this bias is $O\big(|H_{j0}^*|\,|\partial_0 F|^2\big)$, so the coupling conclusion still holds with the explicit Newton-residual bound.

This criterion (Equation 11) behaves differently in two regimes. Near the optimum, $|r_0| = O((\partial_0 F)^2)$ shrinks quadratically while $\|\partial_j \tilde{F}\|$ shrinks only linearly with the iterate's distance from the optimum, so the bias becomes negligible. Far from the optimum, the inequality can fail when the third derivative is large, the intercept curvature approaches zero along the step, and $|\partial_0 F|$ is large at the same time. This is the cold-start regime, particularly for losses where f''' grows with η, such as Poisson ($f''' = e^\eta$). There the linear model implicit in a single Newton step becomes a poor approximation, and a safeguard such as backtracking the Newton step until the loss decreases recovers the leading-order accuracy of the coupling correction (Equation 10).

The error bound and coupling correction describe a single Newton step at the current iterate (Equation 9–Equation 10). The experiments in Section 4 show how those gains accumulate across an outer solve.

The gradient strategy is the Schur-coupling correction with step size $\alpha = 1/L_0$ rather than $1/H_{00}$. When $H_{00} > 0$, substituting $\alpha = 1/L_0$ into Equation 5 gives

$$\partial_j F^{\text{grad}} = \partial_j F - \frac{H_{00}}{L_0}\frac{H_{j0}}{H_{00}}\partial_0 F + r_j(1/L_0), \tag{12}$$

so, to first order, the gradient strategy removes a fraction $H_{00}/L_0 \in (0, 1]$ of the coupling component $(H_{j0}/H_{00})\partial_0 F$ that Newton targets at leading order. The exact strategy instead reaches the true profiled gradient by conditional minimization. In a frozen quadratic, $r_j(1/L_0) = 0$ and the correction fraction is exact. For a nonlinear loss, the leading-order residual relative to the local Schur-corrected target is $(1 - H_{00}/L_0)(H_{j0}/H_{00})\partial_0 F + r_j(1/L_0)$. When $L_0 = \infty$, the intercept step is zero, so no coupling is removed.

The leading-order correction is fully effective only when $H_{00} \approx L_0$, that is, when the Lipschitz bound is tight at the current iterate. Whenever H_{00} is appreciably below L_0—as happens for logistic regression once predictions become confident, or whenever the weights w_i are concentrated on a small subset of observations—the gradient strategy leaves a fraction $1 - H_{00}/L_0$ of the leading-order coupling correction unapplied. Coordinate descent then continues on a worse-conditioned local model than necessary.

The Poisson loss is the limiting case: its intercept curvature has no finite global upper bound, so $L_0 = \infty$ and $H_{00}/L_0 = 0$ at every finite iterate. The gradient strategy then leaves the intercept unchanged, so the coefficient subproblem sees the full, uncorrected gradient.

submitted

3.3.4 Local-rate interpretation

Equation 12 describes one coefficient update, not a convergence rate. A local quadratic calculation supplies the missing link. Freeze the Hessian at the optimum, restrict the coefficients to the active set A, and define the collective coupling

$$\kappa = \frac{H_{0A} H_{AA}^{-1} H_{A0}}{H_{00}} \in [0, 1). \tag{13}$$

If each pass first minimizes the active coefficient block and then updates the intercept, Newton contracts the intercept error by κ, whereas the gradient strategy contracts it by $1 - (H_{00}/L_0)(1 - \kappa)$. As $\kappa \to 1$, the ratio of the required pass counts approaches L_0/H_{00}. Thus, coupling determines when the intercept mode controls convergence, and the curvature ratio determines how much slower the gradient strategy becomes once it does.

This is a local explanation, not a global iteration-complexity theorem. The complete calculation, its frozen-random-product validation, and a deliberately non-sharp randomized-CD heuristic appear in Supplement S1.1. There we also show why the heuristic predicts the imbalance scaling but not the transition to the late-stage plateau.

3.3.5 A safe Newton variant

The Newton-coupling bias (Equation 10) identifies the cold-start regime—large $|\partial_0 F|$ combined with rapidly varying f''—as the setting where an unguarded Newton step on the intercept perturbs the next coefficient pass enough to matter. The fix is to wrap the Newton direction $-\partial_0 F / H_{00}$ in an Armijo backtracking line search: accept the full step ($\alpha = 1$) if it decreases F enough, otherwise halve α until it does. Our Newton strategy includes this safeguard. When it accepts the full step, the safeguard adds constant work; when the linear model is inadequate, the number of backtracking trials grows logarithmically. In the isolated-guard diagnostic in Figure 6 and Figure 7, we compare the guarded variant directly against a bare Newton step and show when the guard changes the result.

3.3.6 Extension to vector intercepts

The same mechanism extends to multinomial logistic regression. Under a reference-class parameterization, the intercept becomes a $(K-1)$-vector, and its Hessian is

$$H_{00} = \frac{1}{n} \sum_{i=1}^{n} \left\{ \operatorname{diag}(p_{i,1:K-1}) - p_{i,1:K-1} p_{i,1:K-1}^{\top} \right\}.$$

Rare free classes create low-curvature coordinate directions; a rare reference class creates a low-curvature common-shift direction. A global Böhning majorant therefore attenuates the intercept correction in precisely the modes associated with rare classes. A single Armijo-guarded Newton block step adapts to those modes, while the exact strategy adds inner iterations without a visible improvement in the outer trajectory. Supplement S2.1 derives the block update, identifies the relevant Rayleigh quotients, and tests this prediction on a grid of synthetic problems.

The pattern appears on two real datasets with opposite shapes (Figure 3). Yeoh2002 has $n = 248$, $p = 12\,625$, and six classes, while StatLog Shuttle has $n = 58\,000$, $p = 9$, and seven classes. On Yeoh2002, the gradient strategy needs roughly four times as many passes as Newton to bring the relative suboptimality bound below 10^{-4}. On Shuttle, whose rarest classes contain only ten and thirteen observations, it fails to reach 10^{-4} within 1,000 passes; Newton and exact reach that threshold in about 25 passes.

submitted

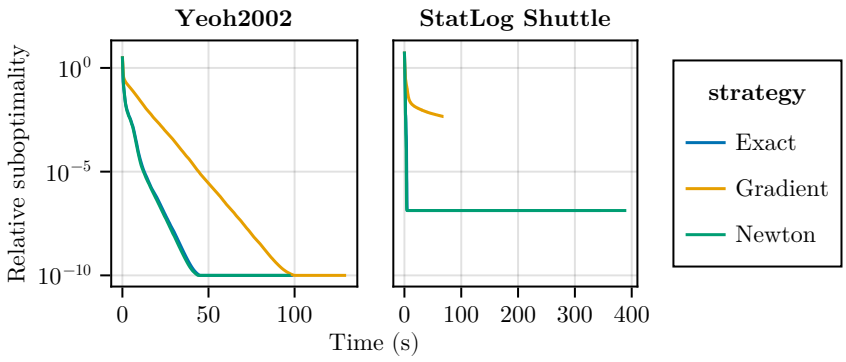


Figure 3: Multinomial logistic regression on two real datasets. The panels show a relative suboptimality bound versus time for Yeoh2002 ($p \gg n$) and StatLog Shuttle ($n \gg p$) at $\lambda = 0.05\lambda_{\max}$ under cyclic coordinate descent.

3.4 Connection to IRLS

So far, we have considered *direct* coordinate descent on F, which recomputes the weights $w_i = f''(\eta_i; y_i)$ after every coefficient update. A widely used alternative—and the one used by glmnet (Friedman et al. 2010)—is iteratively reweighted least squares (IRLS) (McCullagh and Nelder 1989). At the current iterate $(\beta_0^{(t)}, \beta^{(t)})$, IRLS forms the weighted least squares quadratic surrogate

$$\tilde{F}^{(t)}(\beta_0, \beta) = \frac{1}{2n} \sum_{i=1}^{n} w_i^{(t)} \left(z_i^{(t)} - \beta_0 - x_i^\top \beta\right)^2,$$

with working response $z_i^{(t)} = \eta_i^{(t)} - r_i^{(t)}/w_i^{(t)}$. It then solves $\tilde{F}^{(t)}$ by inner CD and re-linearizes. Two weight choices are used in practice: *local* weights $w_i^{(t)} = f''(\eta_i^{(t)}; y_i)$ (glmnet's `type.logistic = "Newton"`), and *MM-majorized* weights replacing $w_i^{(t)}$ by a global upper bound on f'' (e.g. $w_i \equiv 1/4$ for logistic; glmnet's `type.logistic = "modified.Newton"`, following the MM tradition of Hunter and Lange (2004)).

Within an inner solve with local weights, $\tilde{F}^{(t)}$ is quadratic in β_0 with curvature $(1/n)\sum_i w_i^{(t)} = H_{00}$, so the inner Lipschitz constant satisfies $L_0^{\text{inner}} = H_{00}$. The fraction $H_{00}/L_0^{\text{inner}} = 1$ in Equation 12, and Newton on a quadratic is exact, so the gradient, Newton, and exact strategies all coincide on the inner subproblem. Thus, the distinction that we analyze is specific to direct-CD solvers, such as skglm, and does not arise within a local-weight IRLS inner solve.

The strategies no longer coincide under MM majorization. Replacing $w_i^{(t)}$ by the constant majorant $1/4$ in the logistic case, the closed-form inner intercept update on $\tilde{F}^{(t)}$ is

$$\beta_0 \leftarrow \beta_0 + \frac{4}{n}\sum_{i=1}^{n}(y_i - p_i) = \beta_0 - \frac{1}{L_0}\partial_0 F, \qquad L_0 = 1/4,$$

which is the gradient strategy with a global Lipschitz step. The identity holds at the linearization point, where p_i is evaluated, so it describes the first inner intercept update after each re-linearization; once inner CD has moved β, subsequent inner updates drive the intercept to the conditional minimum of the frozen majorant quadratic (Table 2) rather than taking further $1/L_0$ steps on the original loss.

The correction transmitted at each re-linearization is nevertheless governed by the same attenuation. The Schur-coupling residual (Equation 12) thus applies directly to glmnet's `modified.Newton` mode:

submitted

under response imbalance, the leading-order correction fraction H_{00}/L_0 shrinks and the intercept update slows, approaching a true stall only as fitted probabilities reach the boundary.

MM majorization was introduced for numerical stability of the working response z_i when $w_i \to 0$—the same regime in which the resulting update converges slowly. For Poisson loss, no global upper bound on f'' exists, so this MM construction is unavailable. glmnet instead uses local-weight IRLS, where the three strategies coincide as described above.

Changing from local weights to the constant majorant also changes every coefficient's curvature and cross-curvature in the surrogate. A within-package mode comparison therefore isolates the IRLS weight scheme, not the intercept update alone. The intercept correction is the component of that scheme analyzed by Equation 12; Figure 15 supplies the intercept-only intervention.

Across IRLS linearizations the weights $w_i^{(t)}$ change with $\eta^{(t)}$, so even a fully resolved inner intercept becomes stale once the weights are recomputed. This outer-level drift is analogous to but distinct from the within-pass drift of Equation 6, which is direct-CD-specific.

The appropriate intercept strategy therefore depends on the solver family. The production solvers fall into the classes in Table 2. Of the CD-based packages that we surveyed, only skglm 0.5 and earlier apply the gradient strategy directly to the original loss. The remaining solvers—glmnet, biglasso, adelie, LIBLINEAR, BlitzL1, Lasso.jl, and ncvreg (Breheny and Huang 2011)—all linearize before doing CD, either via IRLS or via proximal-Newton (Lee et al. 2014) on a quadratic upper bound.[6] The mode switches within glmnet (`Newton` versus `modified.Newton`) and biglasso (`Newton` versus `MM`) isolate the effect of the weight scheme analyzed in Equation 12.

We exclude two related implementations from the table. Scikit-learn's L1-logistic path either delegates to LIBLINEAR or uses SAGA, a variance-reduced stochastic-gradient method outside the CD framework analyzed here. celer's L1-logistic estimator does not support an intercept in version 0.7.4, and its documentation redirects users with logistic needs to skglm.

Table 2: Classification of CD-based production solvers by algorithmic family and intercept treatment. IRLS and prox-Newton rows conditionally minimize a frozen quadratic surrogate, not the original GLM loss; BlitzL1 alone additionally applies a post-step one-dimensional Newton loop on the original loss.

Solver	Family	Intercept treatment
skglm 0.5	Constant-Lipschitz CD	Gradient on original loss ($L_0 = 1/4$)
glmnet (`type.logistic = "modified.Newton"`)	MM-IRLS	Conditional minimum of global-majorant quadratic
biglasso (`alg.logistic = "MM"`)	MM-IRLS	Conditional minimum of global-majorant quadratic
glmnet (`type.logistic = "Newton"`)	Local-IRLS	Conditional minimum of frozen local quadratic
biglasso (`alg.logistic = "Newton"`, default)	Local-IRLS	Conditional minimum of frozen local quadratic

[6] Sources for the two solvers not in Table 2: ncvreg recomputes local weights `w[i] = fmax2(mu * (1 - mu), 0.0001)` at each linearization and updates the intercept as the weighted-least-squares conditional minimum of the surrogate (`b0[l] = xwr / xwx + a0`, https://github.com/pbreheny/ncvreg/blob/77a3d83/src/glm.c#L198-L237); Lasso.jl fits GLMs by glmnet-style IRLS, an outer re-linearization loop with working weights and residuals around an inner CD solve (https://github.com/JuliaStats/Lasso.jl/blob/cffedae/src/coordinate_descent.jl#L696-L704).

submitted

Solver	Family	Intercept treatment
adelie	Local-IRLS	Conditional minimum of weighted, centered quadratic
LIBLINEAR (newGLMNET, L1-LR)	Prox-Newton	Conditional minimum of frozen quadratic; unregularized
BlitzL1	Prox-Newton	Post-step Newton convergence on original loss

LIBLINEAR (newGLMNET) and BlitzL1 treat the intercept differently at the boundary of each prox-Newton step. LIBLINEAR leaves the intercept unregularized inside inner CD, whose QP convergence carries it to the conditional minimum of the frozen quadratic—but not, in general, of the original loss. BlitzL1 additionally runs an explicit one-dimensional Newton loop to convergence on the original loss after each subproblem and therefore realizes the exact strategy at that boundary.

The within-pass drift of Equation 6 penalizes the exact strategy within direct CD because the inner Hessian drifts between iterations; that critique does not apply at the prox-Newton boundary, where the next subproblem's Hessian is recomputed anyway.

For direct CD, we therefore recommend a Newton update of the intercept. Within each IRLS or prox-Newton quadratic surrogate, we recommend optimizing the intercept fully, with an optional correction on the original loss at the linearization boundary. The surveyed production solvers follow the first two recommendations; BlitzL1 also follows the third.

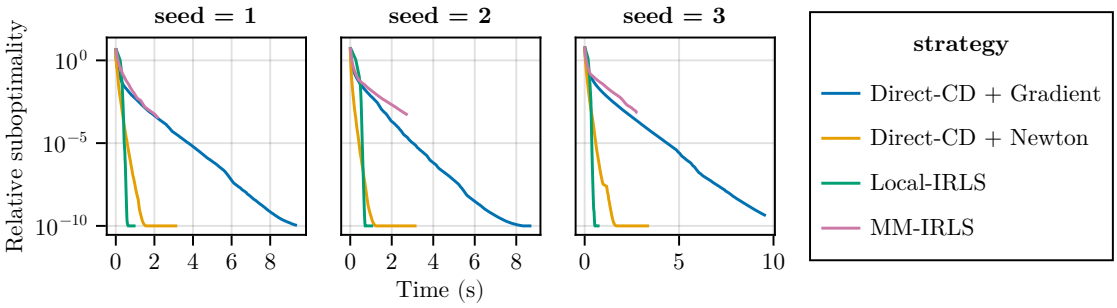


Figure 4: Direct CD and IRLS on an imbalanced logistic design. The panels show a relative suboptimality bound versus time for $\mu_0 = 0.99$, $n = 500$, $p = 1000$, $s = 10$, $\lambda = 0.05\lambda_{\max}$, and three random seeds. The Local-IRLS curve is shown once because the three intercept strategies coincide on the inner quadratic.

Local-IRLS converges fastest under $\mu_0 = 0.99$, crossing the 10^{-8} bound in about 25 passes across the three seeds (Figure 4). Direct-CD with Newton takes roughly twice as many passes, while Direct-CD with the gradient strategy takes several hundred. MM-IRLS remains above 10^{-4} after 200 passes. Within Local-IRLS, the three strategies coincide exactly: once $L_0^{\text{inner}} = H_{00}$, they share the single inner update $-\partial_0 \tilde{F}^{(t)}/H_{00}$ and produce identical iterates. The figure therefore plots only one curve; unit tests verify exact agreement of the intercept and coefficient paths.

Local-IRLS also outpaces Direct-CD with the Newton strategy on this problem. This difference concerns the two algorithmic families rather than their intercept strategies: IRLS-CD enjoys a constant inner Hessian and locally quadratic outer convergence, which favors it on canonical-link GLMs where the linearization is tight. Direct-CD has the edge when f''' is large (e.g. Poisson with extreme rates, where each linearization is loose far from the current iterate), when

working-response instability under saturation matters, or when screening and working-set methods (standard in modern ℓ_1 solvers) compose poorly with re-linearization (Bertrand et al. 2022). This family-level difference does not change the intercept recommendation: within either family, use local Newton curvature rather than a global Lipschitz bound.

4 Results

4.1 Methodology and reproducibility

We run the internal experiments using our Julia implementations of the solvers and report outer passes as the primary measure of work. Unless noted, synthetic designs use cyclic ordering, fixed seeds, standardized features, and an AR(1) correlation of $\rho = 0.6$. We score comparable trajectories against the strongest feasible dual value found for their shared problem instance, so the displayed suboptimality bound does not depend on which strategy produced the primal iterate. We use pinned datasets in the real-data experiments and follow the same shared-reference principle in the production comparison. We load cached outputs for computationally expensive experiments. The toy example in Figure 1 instead runs live when the notebook renders. We pin the Julia, R, and Python environments in the repository and give the complete construction, tolerances, budgets, data preparation, hardware, and reproduction protocol in Supplement S3.

4.2 Simulated data

We begin with the simplest consequence of Section 3. The gradient strategy's intercept step is the local Newton direction scaled by H_{00}/L_0. As imbalance pushes μ_0 toward one, H_{00} shrinks while $L_0 = 1/4$ remains fixed. We therefore expect the intercept update to slow.

4.2.1 Synthetic trajectories and safeguards

We test the strategy comparison on a standardized binary-logistic design with $n = 500$, $p = 1000$, $s = 10$, AR(1) correlation $\rho = 0.6$, cyclic CD, and $\lambda = 0.05\lambda_{\max}$.

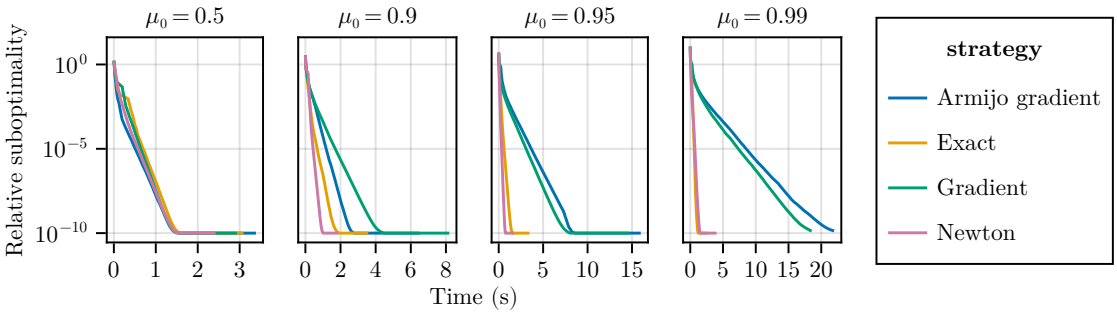


Figure 5: Imbalance sweep on the standardized logistic design. The panels show a relative suboptimality bound versus time for the intercept-update variants in the legend as μ_0 varies, with $n = 500$, $p = 1000$, $s = 10$, $\lambda = 0.05\lambda_{\max}$, and cyclic CD.

The Schur-coupling residual (Equation 12) leads to our first prediction. The gradient strategy scales the local Newton direction by H_{00}/L_0, which approaches zero as the response imbalance μ_0 approaches one. At $\mu_0 = 0.99$, the gradient strategy therefore needs about fifteen times as many passes as Newton to reach 10^{-8}, whereas the Newton and exact strategies remain close across the sweep (Figure 5). Their agreement supports the prediction that exact profiling adds little outer progress beyond one Newton step (Equation 6); Supplement S4.3 compares their inner work.

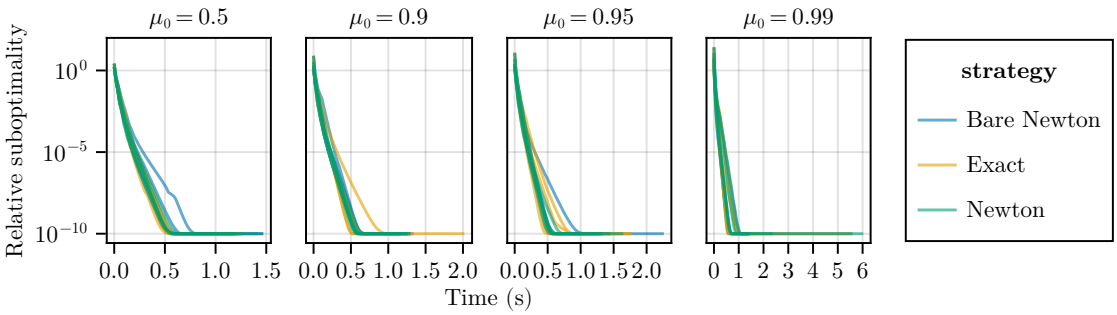


Figure 6: Logistic cold-start comparison. The panels show a relative suboptimality bound versus time for bare Newton, Armijo-guarded Newton, and the exact strategy across the μ_0 levels of Figure 5. Each panel shows five simulated replicates per strategy.

The Newton-coupling bias (Equation 10) leads to our second prediction. At a cold start, a large $|\partial_0 F|$ can make an unguarded Newton step on the intercept unstable. To test this prediction directly, we run bare Newton (without Armijo backtracking) alongside the guarded variant and the exact strategy in Figure 6.

For logistic regression, the three trajectories are indistinguishable, even in the most imbalanced case at $\mu_0 = 0.99$. The Armijo guard rarely fires: most runs are bit-identical to the bare Newton step, and the remaining pairs begin to differ only late in the solve, never at cold start. Every guarded and bare Newton pair crosses the shared 10^{-8} bound on the same pass; exact differs from them by only a handful of passes.

Thus, the Armijo safeguard does not improve the logistic cold starts tested here: the full Newton step is accepted at the initial iterate, and the few later differences do not systematically favor either variant. Logistic curvature is bounded, $f'' = \mu(1-\mu) \le 1/4$, but bounded curvature alone does not make an undamped Newton step globally descending. These experiments establish the narrower empirical result for the zero-initialized problem family above.

Losses with unbounded curvature behave differently. For Poisson regression with the canonical log link, $f''(\eta) = e^{\eta}$ and $f'''(\eta) = e^{\eta}$ both grow without bound, so the Newton linearization is loose far from the optimum and the unguarded step can overshoot by an arbitrary factor. We repeat the cold-start experiment on synthetic Poisson data with $n = 500$, $p = 1000$, $s = 10$, $\lambda = 0.1\lambda_{\max}$, and baseline rates $\mu_0 \in \{10, 30, 100, 300\}$.

At $\mu_0 = 300$, the first bare Newton step moves the cold-start intercept from zero to $\bar{y} - 1 \approx 300$, although the intercept-only optimum is only $\log(\bar{y}) \approx 6$. Because the log link exponentiates the intercept, this step implies a fitted rate near e^{300} and makes the loss astronomical at that intermediate iterate. The safeguard prevents this overshoot (Figure 7): the unguarded bound jumps several-fold after the first pass, while the guarded bound falls immediately.

The overshoot is nonetheless recoverable. The outer CD loop's own per-coordinate Armijo guard absorbs it over the next few passes, and the final primals agree to numerical precision. The three strategies return to nearly the same trajectory within a few passes.

The guard therefore keeps the iterates numerically reasonable but provides no visible improvement in the convergence rate here. The exact strategy is competitive in both experiments but pays the within-pass-drift cost identified in Equation 6. It applies the same Armijo safeguard to every inner Newton direction; every returned scalar update satisfies $|\partial_0 F| \le 10^{-10}$, and a cap hit would stop the experiment rather than enter the cache.

Submitted

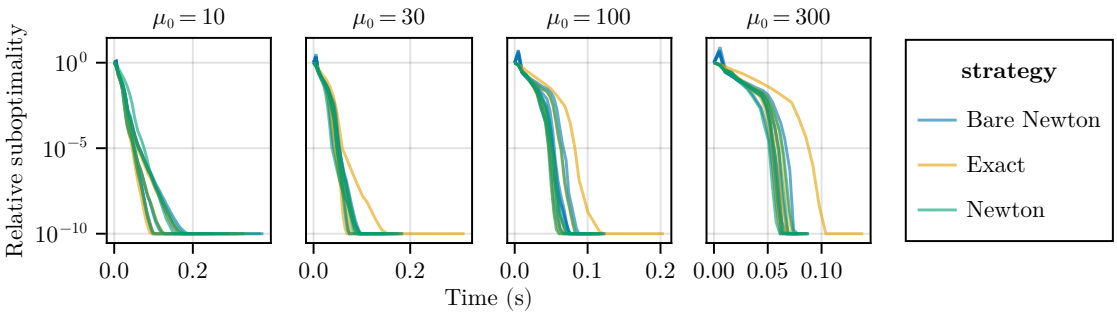


Figure 7: Poisson cold-start comparison. The panels show a relative suboptimality bound versus time for bare Newton, Armijo-guarded Newton, and the exact strategy at baseline rates $\mu_0 \in \{10, 30, 100, 300\}$, with $n = 500$, $p = 1000$, $s = 10$, and $\lambda = 0.1\lambda_{\max}$. Each panel shows five simulated replicates per strategy.

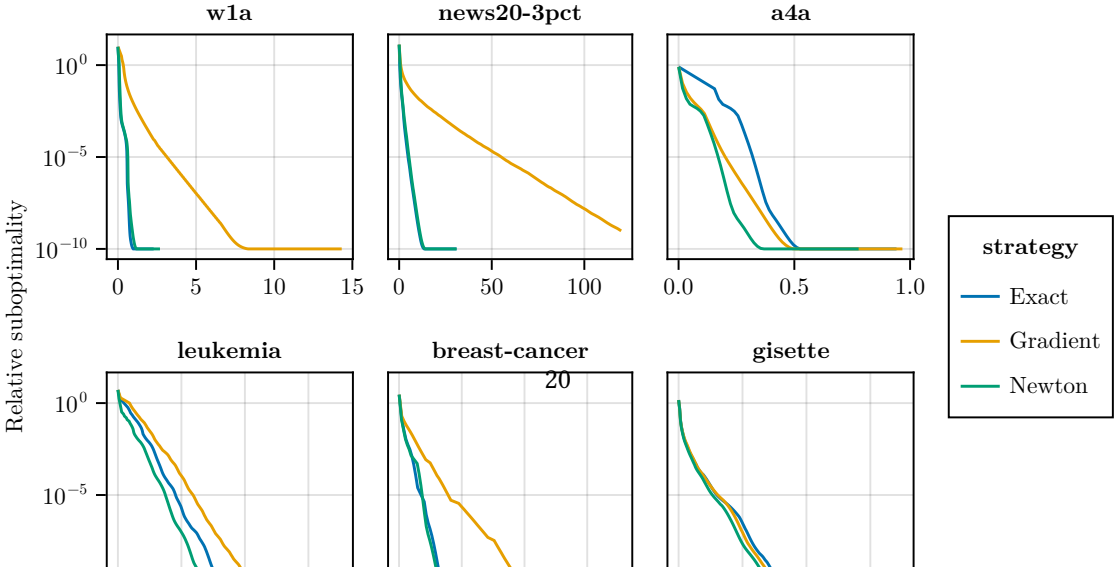
4.3 Real data

We next examine whether the same scaling appears on real problems. We run the three strategies on six LIBSVM benchmark problems at $\lambda = 0.05\lambda_{\max}$, listed in Table 3 and ordered by the intercept-only slowdown proxy $[4\pi_+(1-\pi_+)]^{-1}$. The table lists its reciprocal, the intercept-only curvature fraction $4\pi_+(1-\pi_+)$. The six problems span a ninefold range in predicted slowdown and include sparse and dense designs in both dimensional regimes.

The two most imbalanced problems, w1a and news20-3pct, are especially useful. They have nearly the same π_+ but very different sparse designs: w1a has 300 web-indicator features, while news20-3pct is drawn from a text corpus with an original vocabulary of roughly 1.3 million terms and retains 4,675 features after filtering. Agreement between them would argue against a peculiarity of w1a.

Table 3: LIBSVM problems used in Figure 8. π_+ is the positive-class marginal, and $4\pi_+(1-\pi_+)$ is the intercept-only curvature fraction, not the fitted H_{00}/L_0. news20-3pct is a deterministic 60/1,940 subsample of `news20.binary`; after subsampling, features with fewer than 20 nonzeros are dropped.

dataset	n	p	density	π_+	$4\pi_+(1-\pi_+)$
w1a	2,477	300	sparse	0.029	0.113
news20-3pct	2,000	4,675	sparse	0.030	0.116
a4a	4,781	123	sparse	0.248	0.747
leukemia	38	7,129	dense	0.711	0.823
breast-cancer	683	10	dense	0.350	0.910
gisette	6,000	5,000	dense	0.500	1.000



The two severely imbalanced datasets reproduce both predicted patterns on very different sparse designs. The gradient strategy takes about six times as many passes as Newton to reach 10^{-6} on w1a and about ten times as many to reach 10^{-4} on news20-3pct, before the latter reaches its wall-time cap. Both empirical ratios differ from the intercept-only slowdown proxy by similar constant factors; Figure 18 compares that proxy with an iterate-level alternative. Their agreement across designs is consistent with response imbalance driving the gap rather than a peculiarity of w1a.

For the moderately imbalanced a4a and breast-cancer data, the gradient/Newton gap is modest—less than threefold—consistent with their larger intercept-only curvature fractions. Leukemia ($n \ll p$, penalty-dominated) and gisette (dense and balanced) serve as negative controls: leukemia shows little separation, and the gisette pass counts are identical. The exact strategy tracks Newton on every panel, consistent with Equation 6 at this tolerance.

Poisson data offer a sharper version of the same test. PoissonLoss has no global upper bound on $f''(\eta) = e^{\eta}$, so $L_0 = \infty$ and the gradient-strategy update degenerates: the per-pass change is $-\sum_i(\hat{\mu}_i - y_i)/(L_0 n) = 0$, an exact no-op on the intercept. We test this on the congress109 phrase-count corpus of Taddy (2015).

The corpus contains $n = 529$ speakers from the 107th–109th US Congress and $p = 999$ political phrases (the 1,000-phrase vocabulary with the response phrase removed). The response y_i is speaker i's count of the phrase `american.people` ($\bar{y} = 11.83$, $y_{\max} = 396$), the highest-frequency phrase in the corpus; the features are the remaining 999 phrase counts after column standardization.

This setup follows the distributed-multinomial-regression construction of Taddy (2015), which approximates a multinomial model over the vocabulary by fitting an independent Poisson lasso for each term. The baseline rate $\bar{y} \approx 12$ places the problem near the beginning of the synthetic cold-start sweep in Figure 7 ($\mu_0 = 10$). At this rate, the first bare Newton step is $\bar{y} - 1 \approx 10.83$, while the intercept-only optimum is $\log(\bar{y}) \approx 2.47$. We set $\lambda = 0.05\lambda_{\max}$ and use cyclic CD with a 1 000-pass budget.

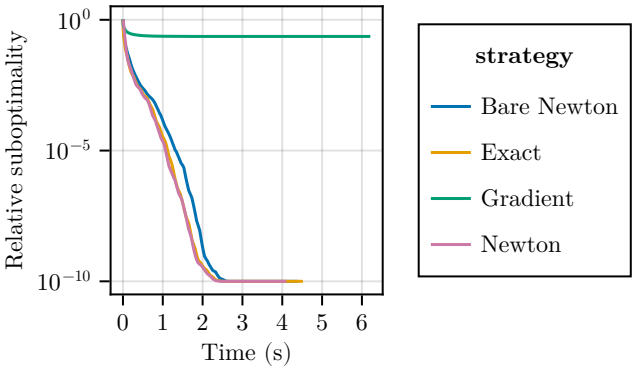


Figure 9: Poisson regression on congress109. The plot shows a relative suboptimality bound versus time for the four intercept strategies at $n = 529$, $p = 999$, $\bar{y} = 11.83$, and $\lambda = 0.05\lambda_{\max}$.

In Figure 9, the gradient strategy leaves the intercept unchanged because $L_0 = \infty$; this failure follows from the loss family rather than from tuning or a problem-specific imbalance level. Newton and exact agree on a single trajectory because the cdsolver's per-coordinate Armijo loop (Tseng–Yun sufficient decrease, Algorithm 1) absorbs the cold-start overshoot before the first intercept update fires. The synthetic suboptimality curves expose the unguarded first-pass excursion, but they rejoin the guarded curves within a few passes at every baseline rate we ran (Figure 7).

submitted

4.4 Robustness and computational cost

The strategy ranking does not depend on the default cyclic ordering. On w1a, permuting the coordinates changes pass counts by less than a factor of two; Newton and exact remain fast, and gradient remains slow (Supplement S4.1). Warm-starting a regularization path makes Newton and exact still harder to distinguish, but it does not rescue the gradient strategy on severely imbalanced w1a (Supplement S4.2).

Instrumenting the intercept solve explains the remaining difference between Newton and exact. Across $\mu_0 \in \{0.5, 0.9, 0.99\}$, exact uses roughly one-and-a-half to twice as many inner work units before reaching the shared 10^{-8} bound while finishing in essentially the same number of outer passes. Most of that overhead occurs in the first cold-start passes; warm starts largely avoid it (Supplement S4.3). The full ordering, path, and per-pass trajectories appear in Supplement S4.

4.5 Imbalance and regularization

The preceding single-λ experiments vary one quantity at a time. Both λ and μ_0 change $\hat{\eta}$ and hence H_{00}, while the active set also changes the collective coupling in Equation 13. We sweep $\mu_0 \in \{0.5, 0.7, 0.9, 0.95, 0.99\}$ against $\lambda/\lambda_{\max} \in \{0.5, 0.2, 0.1, 0.05, 0.02, 0.01\}$ on the standardized binary-logistic design of Figure 5 (3 seeds per cell), running each strategy until its local relative duality gap reaches 10^{-10} or a budget of 5000 outer passes. We report passes to the shared 10^{-8} suboptimality bound rather than wall-clock to decouple the per-coordinate Lipschitz-mismatch prediction from BLAS and memory effects.

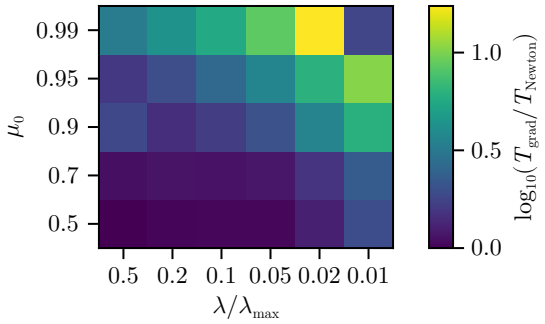


Figure 10: Gradient-to-Newton pass-count ratios over the $(\mu_0, \lambda/\lambda_{\max})$ grid. The heatmap shows $\log_{10}(T_{\text{gradient}}/T_{\text{Newton}})$ averaged over three seeds on the standardized logistic design; clipped cells mark runs that hit the 5000-pass budget.

Greater imbalance generally makes the gradient strategy slower relative to Newton (Figure 10), although the strongest-penalty column is not strictly monotone. The largest informative average ratio is roughly 17 at $\mu_0 = 0.99$ and $\lambda = 0.02\lambda_{\max}$. The $\mu_0 = 0.99$, $\lambda = 0.01\lambda_{\max}$ corner is the only cell in which any run saturates the 5000-pass budget. In one seed, all three strategies hit the cap without reaching the reporting threshold. The displayed average ratio in that cell is therefore uninformative rather than small; we return to it below.

At $\lambda = 0.05\lambda_{\max}$, the regime most relevant to the production path experiments, the empirical ratio grows from nearly one under balance to about eight at $\mu_0 = 0.99$. The corresponding iterate-level $L_0/H_{00}(\hat{\eta})$ proxy rises from about three to twelve. The theory therefore gets the ordering and the growth right, but it overstates the magnitude, with the discrepancy narrowing as imbalance increases.

The cold-start proxy $L_0/H_{00}(0) = 1/(4\mu_0(1-\mu_0))$ is looser still at high imbalance, reaching about 25 at $\mu_0 = 0.99$, because active features pull H_{00} above its intercept-only value $\mu_0(1-\mu_0)$ at this λ.

Under review

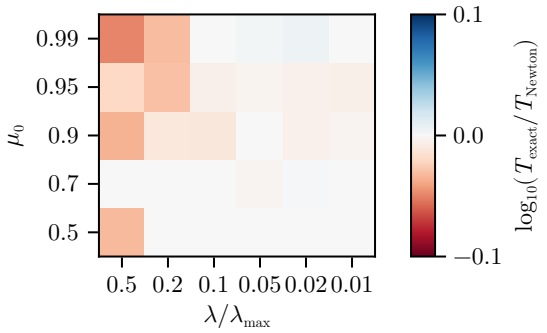


Figure 11: Exact-to-Newton pass-count ratios over the $(\mu_0, \lambda/\lambda_{\max})$ grid. The heatmap shows $\log_{10}(T_{\text{exact}}/T_{\text{Newton}})$ on the same grid and seed average as Figure 10, using a centered diverging colormap with range ±0.1.

As λ shrinks, the active set widens, H_{00} falls to or below $\mu_0(1-\mu_0)$, and the empirical ratio grows with it. At $\lambda = 0.01\lambda_{\max}$, for example, the ratio rises from about two under balance to about ten at $\mu_0 = 0.95$. We omit the $\mu_0 = 0.99$ corner because its seed-level cap makes the average uninformative.

By contrast, Figure 11 stays nearly flat at zero: the exact/Newton ratio is approximately one across the plane, and no cell departs from unity by more than about 10%. This agrees with Equation 6. When both methods converge, the exact strategy needs essentially the same number of outer passes as Newton. Its price is therefore a modest inner-solve overhead that grows with imbalance, not an outer-pass improvement, as Supplement S4.3 shows.

The two heatmaps show that the curvature ratio remains useful outside the tightly coupled plateau. It captures scaling and monotonicity well: the empirical ratio grows as the intercept update becomes more conservative ($H_{00}/L_0 \to 0$), and it generally rises with μ_0 and with $1/\lambda$. It does not predict the magnitude tightly: the iterate-level Equation 14 proxy usually lies above the empirical ratio, but the discrepancy varies across the grid. This is what one would expect from a comparison built from upper bounds and heuristic substitutions rather than from a sharp iteration-complexity formula.

The centering sweep of Figure 19 separates the first-update bound heuristic from the local rate. The heuristic gets the direction right but not the shape or scale. The collective coupling and frozen-pass calculation of Figure 20 recover the transition and the L_0/H_{00} plateau instead.

4.6 Production solvers

We now test the predictions for each strategy class on six production solvers and three imbalanced logistic problems: the shared synthetic design, w1a, and news20-3pct. Together they cover both directions of imbalance and three data regimes (a dense Gaussian design with $p > n$, sparse binary features with $n > p$, and sparse text features with $n < p$). Each problem fixes $\lambda = 0.05 \cdot \lambda_{\max}$. All solvers run in their recommended path mode (warm-start from $\lambda_{\max}$ down to λ in 50 geometric steps);[7] for each solver we sweep its convergence tolerance and plot a suboptimality bound against its available measure of work (CD passes, outer iterations, or wall-clock time).

For every panel, we evaluate the common objective $n^{-1}\sum_i \ell_i + \lambda\|\beta\|_1$ at the same final λ. For each problem, a separate high-accuracy run of our guarded-Newton CD solver defines the shared reference primal $F^\star$ and feasible dual lower bound $D^\star$. Their relative gap is below 10^{-8}; the cached references

[7] At the single-λ configuration reported in Supplement S5.2, biglasso `alg.logistic = "Newton"` returns inaccurate fits on both extreme-imbalance real datasets, while adelie returns an empty state on w1a. These results use loose solver tolerances and do not describe either package's default configuration.

submitted

and certificates are generated by `experiments/production-reference.jl`. We plot $P - D^\star$, an upper bound on each run's suboptimality certified by dual feasibility. This bound cannot claim more precision than the shared certificate, unlike the point estimate $P - F^\star$.

We divide these solvers into two groups according to the curvature used for the intercept correction (Table 2). The first group contains the gradient-strategy solvers—glmnet `modified.Newton`, biglasso `MM`, and skglm—while the second contains the local-IRLS solvers—glmnet `Newton`, biglasso `Newton`, and adelie. LIBLINEAR and BlitzL1 sit outside the split because they solve prox-Newton quadratics; only BlitzL1 additionally realizes the exact strategy on the original loss at the subproblem boundary.

Within-package switches in glmnet and biglasso isolate the weight scheme, skglm 0.5 supplies the direct-CD gradient case, and Figure 15 isolates skglm's upstream Newton fix. With LIBLINEAR, BlitzL1, and adelie, we test whether solvers that use local curvature converge quickly.

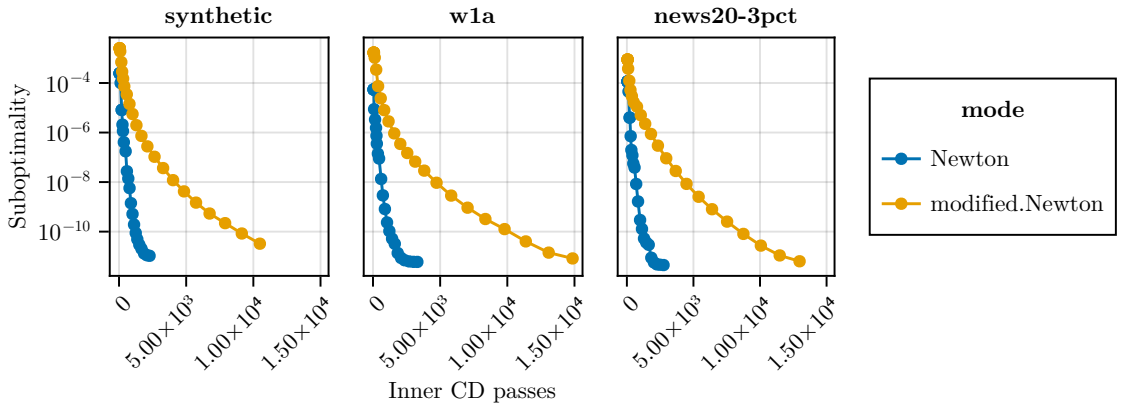


Figure 12: glmnet mode comparison on three imbalanced logistic problems. The panels show a suboptimality bound versus inner CD passes for `type.logistic = "Newton"` and `"modified.Newton"` in path mode on the shared synthetic problem, w1a, and news20-3pct. All modes use the common feasible dual reference $D^\star$ for each problem.

These three panels show the same division among the production solvers. In glmnet and biglasso, changing the weight scheme within the package produces the slowdown predicted for the constant-majorant update. In skglm 0.5—the only production solver in our survey whose intercept update is unambiguously the gradient strategy—the trajectory has the same slow convergence predicted by the Schur-coupling residual (Equation 12). That diagnosis motivated a pull request to skglm replacing the gradient update with a Newton step; the change was merged upstream as commit `8f9cbd77`.[8]

Across Figure 12–Figure 14, the fixed-problem comparisons are consistent with slow convergence under global curvature and fast convergence under local curvature. The glmnet and biglasso switches change the full IRLS weight scheme, however, and the cross-package panels also reflect implementation differences. To isolate the intercept update, we compare two skglm commits in Figure 15 that differ only in that update: `b03644fe`, the last pre-fix commit and behaviorally identical to the released 0.5, and `8f9cbd77`, the merged Newton fix. Running the same `AndersonCD` driver on both yields a controlled 30-cell comparison.

As in Figure 10, the ratio generally grows with μ_0 and with $1/\lambda$, although it is not monotone in every

[8] https://github.com/scikit-learn-contrib/skglm/pull/337, merged 2026-05-13. skglm 0.6 is unreleased at the time of writing; the controlled comparison below pins the two relevant commits directly.

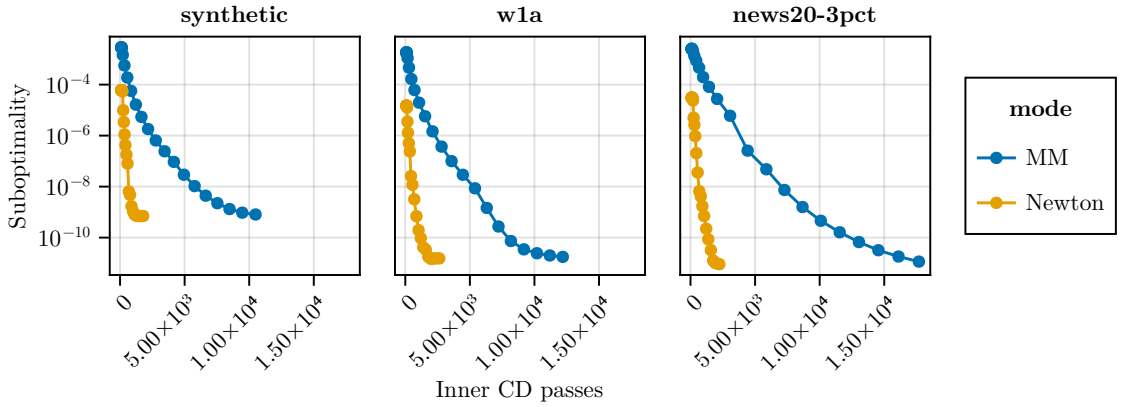


Figure 13: biglasso mode comparison on three imbalanced logistic problems. The panels show a suboptimality bound versus inner CD passes for `alg.logistic = "Newton"` and `"MM"` in path mode on the same three problems as Figure 12. All modes use the common feasible dual reference $D^\star$ for each problem.

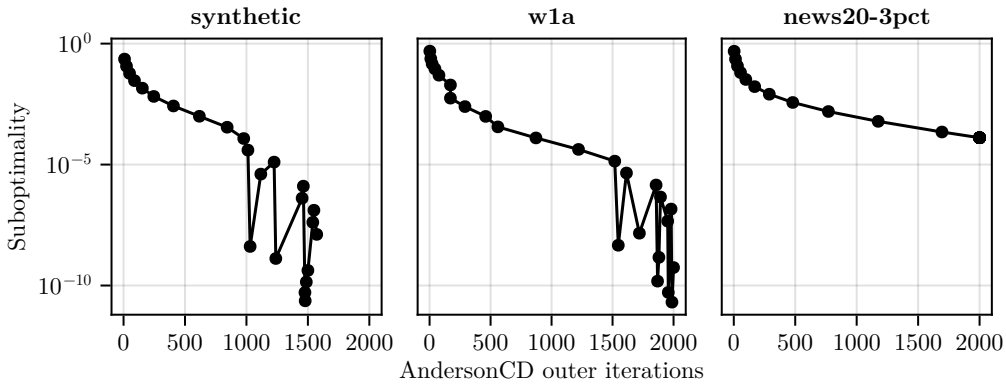


Figure 14: skglm 0.5 on three imbalanced logistic problems. The panels show a suboptimality bound versus `AndersonCD` outer iterations on the same three problems as Figure 12, using the common feasible dual reference $D^\star$ for each problem.

cell. The pre-fix commit needs about 30 times the iterations of the post-fix one even in the least separated cell, and more than 400 times as many in the most imbalanced, weakly regularized corner.

Even the smallest ratio is large because `AndersonCD` reaches its tolerance in fewer than 20 outer iterations once the intercept takes a Newton step, so any remaining slow movement of the intercept dominates the count. Many pre-fix runs hit the 5000-iteration cap, making the largest displayed ratios lower bounds rather than exact comparisons.

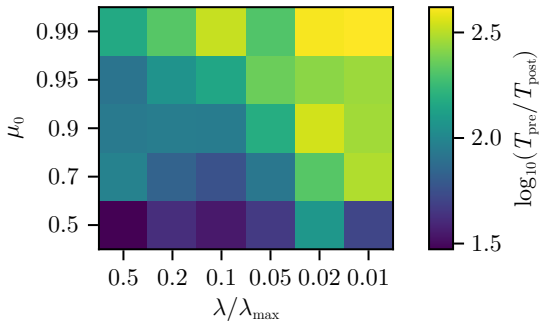


Figure 15: Within-skglm commit comparison. The heatmap shows $\log_{10}(T_{\texttt{b03644fe}}/T_{\texttt{8f9cbd77}})$ for `AndersonCD` across the same $(\mu_0, \lambda/\lambda_{\max})$ grid as Figure 10, averaged over three seeds; clipped cells mark runs that hit the 5000-iteration cap.

The prox-Newton solvers provide the complementary test. Resolving the intercept with local curvature inside each quadratic should avoid the global-majorant stall; an additional original-loss correction may improve the intercept at the subproblem boundary. LIBLINEAR (newGLMNET) implements the former, while BlitzL1 implements both. We compare their wall-clock cost to approach the shared certificate across the same three problems in Figure 16, complementing the inner-pass curves for skglm 0.5 (gradient strategy on the intercept) and glmnet's `modified.Newton` (the constant-majorant alternative).

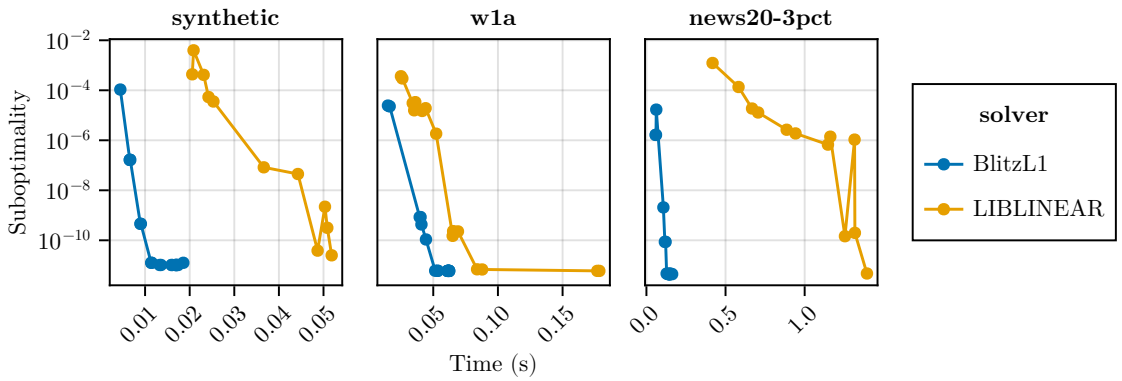


Figure 16: Prox-Newton solvers on three imbalanced logistic problems. The panels show a suboptimality bound versus wall-clock time for LIBLINEAR and BlitzL1 on the same three problems as Figure 12. LIBLINEAR runs that hit its internal `max_iter` cap without further progress are omitted. Both solvers use the common feasible dual reference $D^\star$ for each problem; timings are indicative single-run trajectories, not benchmark rankings.

We test the same prediction with a different implementation using adelie. It builds the same weighted

Gaussian surrogate as glmnet `Newton`, but it realizes the intercept implicitly: rather than updating β_0 as a coordinate inside the inner CD pass, it weighted-centers the features at the start of each surrogate and tracks the residual mean, reconstructing $\beta_0 = \bar{y} + \overline{\text{resid}}$ at the end of the inner solve. By construction, the intercept therefore sits at the surrogate optimum on every outer pass, with no separate update step.

That places adelie in the same class as glmnet `Newton` and biglasso `Newton`. In Figure 17, we sweep its convergence tolerance across the same three problems used above. The curves are flat above `tol` around 10^{-4}: path mode warm-starts each of the 50 lambdas from its predecessor, and those warm starts alone land within 10^{-5} of the optimum, so the tolerance has nothing left to do. Below that plateau, adelie's final objectives lie within the shared reference certificate: the suboptimality bounds are on the order of 10^{-11} and then become certificate-limited. Each fit completes well under a second, and adelie shows no stall or failed solve.

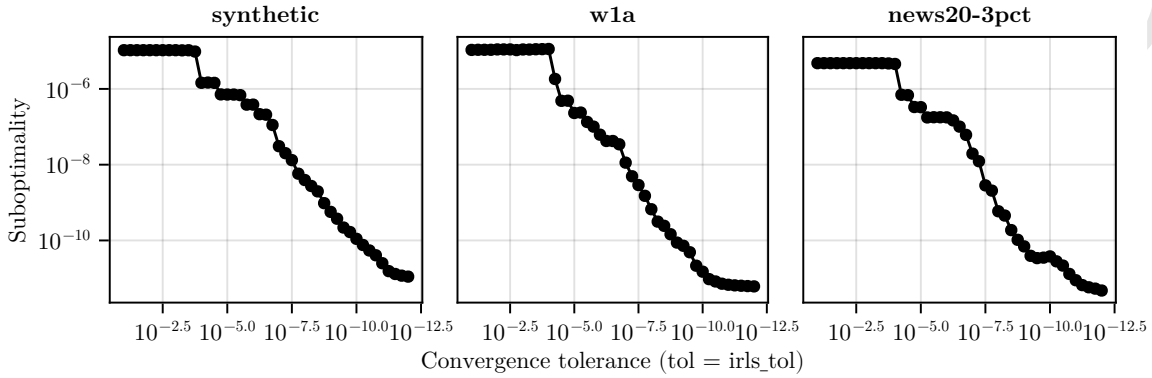


Figure 17: adelie on three imbalanced logistic problems. The panels show a suboptimality bound versus convergence tolerance on the same three problems as Figure 12. The horizontal axis is the coupled outer/inner tolerance (`tol = irls_tol`), and the bound uses the common feasible dual reference $D^\star$ for each problem.

4.6.1 Additional solver diagnostics

Poisson loss provides a structural check: because $f''(\eta) = e^\eta$ is unbounded, the constant-majorant modes have no Poisson analogue. The production solvers that support Poisson here use local IRLS or prox-Newton curvature and converge without the gradient-strategy stall (Supplement S5.1).

A separate single-λ diagnostic probes the opposite risk: local Newton curvature can be unstable at a cold start when solver tolerances are deliberately loose. The observed biglasso and adelie failures depend on the dataset and do not describe their default path-mode calls. We therefore treat them as evidence for safeguarding the Newton direction, not as a package ranking (Supplement S5.2).

5 Discussion

Our analysis yields a practical decision rule: direct coordinate descent on a smooth GLM loss should update the intercept with one Newton block step and an Armijo safeguard. Repeatedly optimizing the intercept adds inner work that is usually erased by the next coefficient sweep; warm-started paths are the main case in which that work becomes cheap enough to reconsider (Supplement S4.2). Solvers that freeze a local IRLS or proximal-Newton quadratic should instead resolve the intercept within that surrogate, where the quadratic curvature is already available.

Submitted

The curvature ratio H_{00}/L_0 explains this rule. On the original nonlinear loss, a global-majorant update removes that fraction of the local intercept–coefficient coupling to first order (Equation 12); in the frozen quadratic model, the fraction is exact. That model then explains how the one-step mismatch becomes the late-stage L_0/H_{00} rate plateau under strong collective coupling (Proposition 5.1, Figure 20). This is a local explanation, not a global iteration-complexity theorem.

The production comparisons test the rule outside our implementation. The glmnet and biglasso switches hold the package fixed but change the full IRLS weight scheme, while the before-and-after skglm comparison changes only the intercept update (Section 4.6). The skglm intervention supplies the direct causal evidence for the intercept update. The other comparisons support the broader prediction about global versus local curvature, but they do not isolate that update from the rest of the surrogate or implementation.

The Armijo guard completes the recommendation. It adds little work when the full Newton step is acceptable and prevents the Poisson cold-start overshoot in Figure 7. Using it by default avoids relying on a loss-specific global descent claim.

For multinomial problems, rare nonreference classes and rare reference classes create low-curvature intercept directions of different forms (Equation 19–Equation 20). Long-tailed label distributions therefore make the scalar binary mechanism a routine block phenomenon rather than an edge case.

Our analysis has two main limitations. First, our results for the Newton strategy are *local*. Equation 9 and Equation 10 are Taylor identities for a single Newton step, and Proposition 5.1 freezes the Hessian and solves the active coefficient block exactly. The frozen products of permuted coordinate passes reproduce the nonlinear rates, but we do not prove a general theorem for their Lyapunov exponents or a global convergence rate for CD with a Newton-updated intercept.

Second, the production-solver experiments in Section 4.6 include only three benchmark problems. The heatmap in Section 4.5 reproduces the binary direct-CD predictions across a 30-cell grid, but the cross-package comparison does not yet sweep the full (μ_0, λ) plane. Future work should extend the production-solver sweep along those axes and check the multinomial predictions against glmnet's symmetric-multinomial path and adelie's multi-response solver.

submitted

Supplementary material

The supplement collects the technical derivations, extended experiments, and implementation details that support the main article. It follows the order of the argument and keeps each figure beside the methods and interpretation that it supports.

S1. Technical derivations and rate diagnostics

S1.1 Rate heuristic and local-rate validation

The H_{00}/L_0 ratio also appears in standard randomized-CD rate bounds. Our algorithm does not satisfy all assumptions of those bounds: it updates the intercept on a fixed schedule, Newton uses local curvature, and the Schur reduction below is exact for only the first coefficient update of each pass. We therefore use the bound only as a scaling heuristic, not as an iteration-complexity result.

Standard sublinear bounds for randomized proximal CD (Wright 2015; Nesterov 2012; Richtárik and Takáč 2014) grow with a sum of terms $L_j R_j^2$, where L_j is a coordinate-wise Lipschitz constant and R_j is the cold-start distance to the optimum along coordinate j. The strategies enter this expression identically on coefficient coordinates but differently on the intercept: the gradient strategy uses the global $L_0 = \sup f''$, whereas the heuristic for Newton and exact substitutes the iterate-level H_{00}.

Remark. (Rate-gap heuristic, with a large-imbalance reduction.) Let $R_j^2 = (\beta_j^{(0)} - \beta_j^\star)^2$ denote the squared cold-start distance to the optimum along coordinate j, with the intercept as coordinate 0. Let B_{G} and B_{N} denote the resulting bound-based proxies for gradient and Newton. The coordinate decomposition gives

$$\frac{B_{\mathrm{G}}}{B_{\mathrm{N}}} = \frac{L_0 R_0^2 + \sum_{j\geq 1} H_{jj}\big(1 - (H_{00}/L_0)\rho_{0j}^2\big)R_j^2}{H_{00} R_0^2 + \sum_{j\geq 1} H_{jj}\big(1 - \rho_{0j}^2\big)R_j^2}. \tag{14}$$

Let S_{G} and S_{N} denote the coefficient sums in the numerator and denominator of Equation 14. In any sequence of problems for which

$$\frac{L_0 R_0^2}{S_{\mathrm{G}}} \to \infty, \qquad \frac{H_{00} R_0^2}{S_{\mathrm{N}}} \to \infty.$$

the intercept term dominates both sums, and Equation 14 collapses to

$$\frac{B_{\mathrm{G}}}{B_{\mathrm{N}}} \approx \frac{L_0}{H_{00}(\hat{\eta})}. \tag{15}$$

For logistic regression, $L_0 = 1/4$ stays fixed while $H_{00}(\hat{\eta}) \to 0$ as fitted probabilities approach the boundary, so the limit diverges whenever the two dominance conditions also hold. Response imbalance or a large $|\beta_0^\star|/\|\beta^\star\|$ ratio alone does not imply those conditions: the intercept score equation with active slopes generally does not give $\beta_0^\star = \mathrm{logit}(\mu_0)$. Outside the intercept-dominated regime, the full Equation 14 is needed.

The rate-gap heuristic starts from the standard sublinear bound for randomized proximal CD on a convex composite $F = f + \psi$ with f having coordinate-wise Lipschitz constants L_j (Wright 2015; Nesterov 2012; Richtárik and Takáč 2014). This bound grows with $\sum_j L_j R_j^2$ and leads to Equation 14. Three substitutions enter the derivation, each of which departs from the bound's assumptions; we retain only the predicted scaling:

1. *Schedule.* The strategies fix the intercept update at every pass rather than drawing it with the uniform coordinate probability the bound assumes. We treat the coordinate sum as a per-coordinate-cost surrogate.

Submitted

2. *Intercept curvature.* For the Newton and exact strategies, we plug in the iterate-level entry H_{00} rather than the fixed coordinate Lipschitz constant used by the CD bound. This substitution delivers the $H_{00}(\hat{\eta}) \to 0$ scaling that the figures track. The gradient strategy keeps the global $L_0 = \sup f''$, as the bound prescribes.

3. *Schur reduction on the coefficients.* The exact-strategy identity in Proposition 3.1, together with the Newton strategy's leading-order approximation in Equation 10, motivates using the profiled gradient at the *first* coefficient update of each pass. The corresponding local quadratic model has curvatures $\tilde{H}_{jj} = H_{jj}(1 - \rho_{0j}^2)$ by Equation 4, with ρ_{0j} the normalized cross-curvature of Equation 2. As a heuristic, we extend this one-update model to the full pass and treat the Schur curvature as the effective coefficient curvature. To leading order, the gradient strategy removes only a fraction H_{00}/L_0 of the coupling (Equation 12), giving the heuristic effective curvature $H_{jj}(1 - (H_{00}/L_0)\rho_{0j}^2)$.

Substituting the second and third approximations into the $L_j R_j^2$ sum gives Equation 14. Retaining only the R_0^2 term under the two curvature-weighted dominance conditions stated above gives Equation 15.

The heuristic proxy diverges as $H_{00}/L_0 \to 0$. In Figure 18, we test whether the empirical pass-count ratio follows this scaling; we do not test or imply an iteration-complexity theorem.

To summarize the coupling across slope coordinates, define the curvature-weighted mean

$$\bar{\rho}^2 = \frac{\sum_{j=1}^{p} H_{jj}\rho_{0j}^2}{\sum_{j=1}^{p} H_{jj}}.$$

This diagnostic is near zero when most slope coordinates are locally decoupled from the intercept and approaches one when most of the slope curvature lies in strongly coupled coordinates.

The approximation has two regimes. In strongly coupled designs ($\bar{\rho}^2 \to 1$, e.g. uncentered designs where the intercept absorbs the column means), the denominator's Schur term collapses and the ratio gains an additional factor of order $1/(1 - \bar{\rho}^2)$. In standardized designs ($\rho_{0j} \approx 0$), the Schur correction is inactive, but Equation 15 still diverges through the intercept's R_0^2 dominance. The per-coordinate Lipschitz mismatch on the intercept and the Schur coupling propagated to the coefficients are separate effects, although both become important as $H_{00}/L_0 \to 0$.

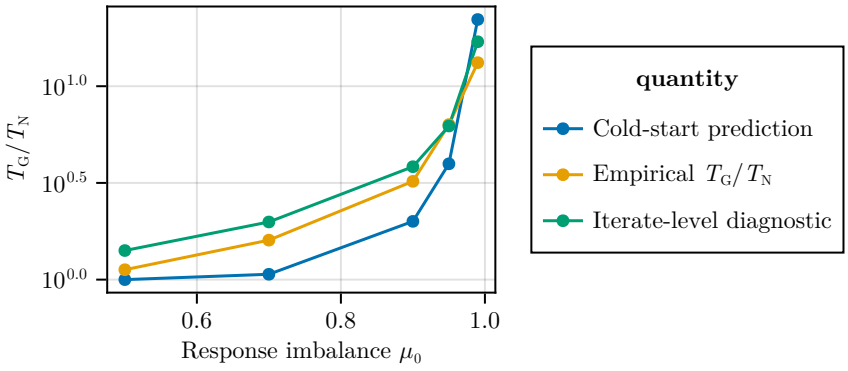


Figure 18: Rate-gap diagnostics on the standardized logistic design. The plot shows empirical T_G/T_N together with the cold-start and iterate-level evaluations of Equation 14, where T_G/T_N is the pass-count ratio to the relative suboptimality bound 10^{-6}, swept over μ_0 for $n = 500$, $p = 1000$, $s = 10$, and $\lambda = 0.05\lambda_{\max}$.

Unedited

In Figure 18, we compare both evaluations of Equation 14 with the empirical ratio. The cold-start proxy is evaluated at $\eta^{(0)} = \mathrm{logit}(\mu_0)$ and is therefore computable before solving. On the standardized design, its constant weights give $H_{0j} = \mu_0(1-\mu_0)\sum_i X_{ij}^{\mathrm{std}} = 0$ exactly, so the Schur correction vanishes. Consequently, Equation 14 reduces to

$$\omega_0 \frac{L_0}{H_{00}(\eta^{(0)})} + (1-\omega_0), \qquad \omega_0 = \frac{H_{00}(\eta^{(0)})R_0^2}{H_{00}(\eta^{(0)})R_0^2 + \sum_{j\geq 1} H_{jj}(\eta^{(0)})R_j^2}.$$

It is a convex combination of one and the intercept-only slowdown $L_0/H_{00}(\eta^{(0)}) = 1/(4\mu_0(1-\mu_0))$ from Section 4.5, with weight determined by the intercept's share of the curvature-weighted cold-start distance.

The iterate-level diagnostic substitutes $H_{00}(\hat{\eta})$, which is below $\mu_0(1-\mu_0)$ at moderate imbalance (active features pull predictions away from the symmetric center, decreasing weights) and above it at extreme imbalance (active features pull some predictions away from the majority-class corner, increasing weights).

Neither curve uniformly dominates. The cold-start proxy is tighter near balance, while the iterate-level proxy becomes tighter as imbalance grows. Together they bracket the empirical ratio over much of the sweep, but both overstate the slowdown at the most imbalanced endpoint as they approach the intercept-dominated limit L_0/H_{00} of Equation 15. We examine this discrepancy over a wider grid in Section 4.5.

The sweep in Figure 18 varies H_{00}/L_0 through μ_0 while keeping $\bar{\rho}^2$ close to zero in the standardized design. In a second experiment, we hold μ_0 fixed and vary $\bar{\rho}^2$ directly. This is a stronger test: rather than asking whether the heuristic tracks imbalance, it asks whether its first-update coupling correction predicts the long-run rate. On the same problem family with `means = :random` (so each column carries an $O(1)$ mean), a centering fraction α interpolates between the standardized and uncentered designs,

$$X(\alpha) = \big(X - (1-\alpha)\,\bar{x}^\top\big)/s, \qquad \alpha \in [0,1], \tag{16}$$

with the column scales s held fixed. Shifting a column by a constant is absorbed exactly by the unpenalized intercept, so $\beta^\star$, $\hat{\eta}$, H_{00}, and $\lambda_{\max}$ are all invariant along the sweep: H_{00}/L_0 remains about 0.30 when $\mu_0 = 0.5$ and 0.17 when $\mu_0 = 0.9$, even as $\bar{\rho}^2$ increases from nearly zero to roughly 0.5. Thus, the sweep changes the coupling without changing the curvature ratio.

The observed pass counts do not follow the shape suggested by the first-update bound heuristic (Figure 19). The empirical $T_{\mathrm{G}}/T_{\mathrm{N}}$ rises sharply at weak coupling and then settles near the asymptotic limit L_0/H_{00}. By contrast, the leading-order ratio $(1-(H_{00}/L_0)\bar{\rho}^2)/(1-\bar{\rho}^2)$ is nearly flat where the empirical curve is steep and remains well below it. The heuristic predicts the direction of the first-update effect but not the long-run shape or magnitude.

The sweep argues against intercept distance as the cause of this plateau. Across the high-coupling part of the sweep, uncentering changes $|\beta_0^\star|/\|\beta^\star\|$ non-monotonically by more than an order of magnitude while the pass-count ratio remains essentially unchanged. Instead, coupling determines which convergence mode is slowest, while H_{00}/L_0 determines the relative rate once the coupled mode dominates. This interpretation concerns late-stage local rates, not global iteration complexity from an arbitrary start.

This failure ends the role of Equation 14 as a rate model: it supplies a useful imbalance scaling, but not the long-run shape. The plateau instead follows from the local iteration. Freeze the Hessian at the optimum, restrict the coefficients to the active set A, and translate the optimum to zero. Once

submitted

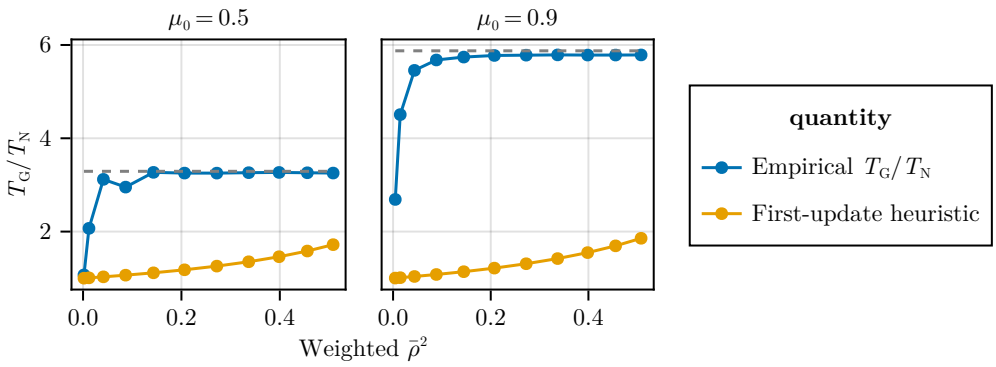


Figure 19: Centering sweep at fixed H_{00}/L_0. The panels compare empirical and first-update-heuristic T_G/T_N; the dashed line marks L_0/H_{00}. The centering fraction α in Equation 16 moves the weighted $\bar{\rho}^2$ from standardized to uncentered designs. Here $n = 500$, $p = 1000$, $s = 10$, $\lambda = 0.05\lambda_{\max}$, and the coordinate ordering is permuted.

the active signs are fixed, the penalty is locally affine and the quadratic error has Hessian

$$H = \begin{bmatrix} H_{00} & H_{0A} \\ H_{A0} & H_{AA} \end{bmatrix}.$$

The collective coupling between the intercept and the active coefficient block is κ, restating Equation 13:

$$\kappa = \frac{H_{0A} H_{AA}^{-1} H_{A0}}{H_{00}} \in [0, 1).$$

Unlike $\bar{\rho}^2$, which averages diagonal coordinate-wise correlations, κ allows many individually small correlations to combine through H_{AA}^{-1}.

Proposition 5.1. (Frozen two-block rate.) *Let $q = H_{00}/L_0$. On the frozen quadratic, suppose H is positive definite and each pass first minimizes over the active coefficient block and then updates the intercept. The Newton and gradient strategies contract the intercept error by*

$$r_N = \kappa, \qquad r_G = 1 - q(1 - \kappa),$$

respectively. When the intercept-error mode is the slowest mode of the pass—which requires κ close to one—their asymptotic pass-count ratio is

$$\frac{T_G}{T_N} \approx \frac{\log \kappa}{\log(1 - q(1-\kappa))} \longrightarrow \frac{1}{q} = \frac{L_0}{H_{00}} \quad \text{as } \kappa \to 1. \tag{17}$$

Outside that regime the ratio formula is not informative: as $\kappa \to 0$ it diverges, while the true pass-count ratio stays finite because Newton removes the intercept error in a single pass and a coefficient mode becomes the slowest.

To see this, let a denote the intercept error. Conditional minimization gives $\beta_A^+ = -H_{AA}^{-1} H_{A0} a$, after which the intercept gradient is $H_{00}(1 - \kappa)a$. Newton therefore leaves error κa, whereas the gradient strategy leaves $[1 - q(1 - \kappa)]a$. Taking logarithms gives Equation 17; expanding both logarithms at $\kappa = 1$ gives the limit. This route contains no cold-start distance R_0.

The proposition assumes exact minimization of the coefficient block, while the algorithm uses one permuted coordinate sweep. For that algorithm, let C_π be the frozen linear map for a coefficient

submitted

sweep in order π, and let

$$E_N = I - \frac{1}{H_{00}} e_0 e_0^\top H, \qquad E_G = I - \frac{1}{L_0} e_0 e_0^\top H = (1-q)I + qE_N.$$

One frozen pass is $P_{N,\pi} = E_N C_\pi$ or $P_{G,\pi} = E_G C_\pi$. Because we draw a new π each pass, the local rate is governed by products of these random matrices, not by the spectral radius of one cyclic pass.

We compare the top random-product rates with the late-stage slopes of the nonlinear solver in Figure 20. We fit each nonlinear log(relative suboptimality) curve between 10^{-4} and 10^{-8}, freeze H at the converged Newton solution, and propagate the corresponding random pass matrices for 20,000 passes. The twelve cells include the transition at $\rho = 0.6$ and a second $\rho = 0.2$ sweep that moves L_0/H_{00} from roughly 5 to 18.

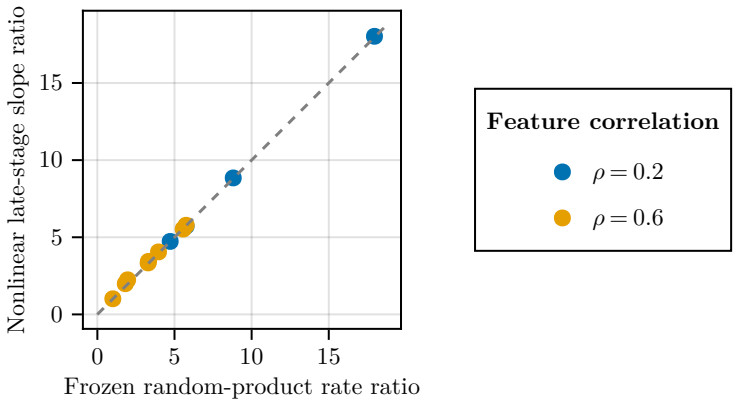


Figure 20: Frozen-pass validation of the centering mechanism. Each point compares the gradient/Newton ratio of the frozen random-product log rates with the ratio of fitted late-stage log-suboptimality slopes from the nonlinear solver, both under permuted coordinate ordering. The cells cover both the weak-to-strong coupling transition at $\rho = 0.6$ and a second $\rho = 0.2$ imbalance sweep. The dashed line is equality.

The frozen and nonlinear rates track each other across the transition. In the $\rho = 0.6$ sweep, κ rises from about 0.4 to almost one, while the frozen and nonlinear rate ratios both rise from about one to a little above three. In the $\rho = 0.2$ sweep, the two ratios again agree closely and reach roughly 18. Newton accepts the full intercept step on every late-stage pass in these cells, so its safeguard does not create the difference.

The rates change in two stages. Under weak coupling, an ordinary coefficient mode controls both strategies, and their rate ratio stays near one. As collective coupling approaches one, the coupled mode becomes slowest; relaxing its intercept correction by q then produces the $1/q = L_0/H_{00}$ plateau. We have proved the exact-block limit in Proposition 5.1 and verified the full permuted-sweep operator numerically. A general Lyapunov-exponent theorem for the random matrix products is beyond our present analysis.

The local-rate argument connects directly to Section 4.6. glmnet's `modified.Newton` mode and biglasso's `MM` mode both apply the gradient strategy at the global $L_0 = 1/4$ to the intercept inside their IRLS inner solves; L_0/H_{00} is therefore the strongly coupled limit suggested by the frozen local model. At $\mu_0 = 0.99$, this is consistent with the roughly fivefold and tenfold slowdowns reported in Figure 12 and Figure 13. skglm (version 0.5), the direct-CD solver, exhibits the same scaling on a continuous trajectory in Figure 14: the slow convergence from suboptimality $\approx 10^{-1}$ to $\approx 10^{-5}$ is the intercept's slow climb toward its fitted value at step size $1/L_0$.

submitted

An Armijo line search cannot rescue the gradient strategy when it starts from the global-Lipschitz step $-\partial_0 F/L_0$. It can only *shrink* the step, which is the wrong direction when $H_{00} \ll L_0$: the step is already too small, not too large. To recover the iterate-level curvature H_{00} a line search would need to *expand* the step beyond $1/L_0$, settling near $1/H_{00}$. But the step of size $1/H_{00}$ applied to the gradient direction is exactly the Newton step, so rescuing the gradient strategy turns it into Newton. A backtracking-Armijo variant of the gradient strategy therefore stalls identically to the bare-Lipschitz version: the Lipschitz step satisfies the sufficient-decrease condition on the first attempt, and the line search never fires (Figure 5).

At the time of these experiments, skglm (version 0.5) (Bertrand et al. 2022) used constant per-coordinate Lipschitz steps and, on its internal unaveraged loss, an intercept step of $4/n$. This is the gradient strategy defined above, with $L_0 = 1/4$ on the averaged loss, and it uses no line search. The other CD-based production packages we surveyed (glmnet, biglasso, LIBLINEAR, and BlitzL1) embed the intercept update inside an outer IRLS or proximal-Newton linearization (Section 3.4). None of the production CD solvers we surveyed use a growth-allowing line search on the intercept alone.

S2. Vector intercepts and multinomial models

S2.1 Derivation and experiments

We can extend the argument to multinomial logistic regression with K classes, where the scalar curvature ratio becomes a matrix operator. The intercept is a vector $\beta_0 \in \mathbb{R}^{K-1}$ under the reference-class parameterization $\eta_{iK} = 0$. Writing $\eta_{ik} = x_i^\top \beta_k + \beta_{0k}$ for $k = 1, \dots, K-1$ and $p_{ik} = e^{\eta_{ik}}/\big(1 + \sum_{l<K} e^{\eta_{il}}\big)$, the per-observation Hessian block has diagonal $p_{ik}(1-p_{ik})$ and off-diagonal $-p_{ik}p_{il}$. The intercept Hessian $H_{00} = (1/n)\sum_i \big(\mathrm{diag}(p_{i,1:K-1}) - p_{i,1:K-1}p_{i,1:K-1}^\top\big)$ is a $(K-1)\times(K-1)$ matrix.

The Newton strategy therefore becomes a small block solve $\delta = -H_{00}^{-1}\nabla_0 F$ followed by Armijo backtracking on the loss along that direction. The gradient identity in Proposition 3.1, the within-pass drift (Equation 6), and the Newton-coupling bias (Equation 10) all carry through with matrix-valued H_{00} in place of the scalar.

To derive the matrix analogue of Equation 12, let $g_0 = \nabla_0 F$, collect the coefficient gradients in g_β, and let $H_{\beta 0}$ be the coefficient–intercept cross block. For an intercept step $\beta_0 \mapsto \beta_0 - A g_0$, a Taylor expansion gives

$$g_\beta^+ = g_\beta - H_{\beta 0} A g_0 + O\big(\|A g_0\|^2\big).$$

Newton uses $A = H_{00}^{-1}$, while the gradient strategy uses $A = L_0^{-1} I$. Dropping the common second-order remainder gives

$$\begin{aligned} g_\beta^{\mathrm{grad}} &= g_\beta - H_{\beta 0} H_{00}^{-1} Q g_0, \qquad Q = \frac{H_{00}}{L_0}, \\ g_\beta^{\mathrm{grad}} - g_\beta^{\mathrm{prof}} &= H_{\beta 0} H_{00}^{-1}(I - Q) g_0, \quad g_\beta^{\mathrm{prof}} = g_\beta - H_{\beta 0} H_{00}^{-1} g_0. \end{aligned} \tag{18}$$

Thus Q, rather than the individual ratios $H_{00,kk}/L_0$, is the matrix analogue of the scalar correction fraction. Because the multinomial majorant satisfies $0 \preceq H_{00} \preceq L_0 I$, write $H_{00} = U\,\mathrm{diag}(\lambda_r)U^\top$. The gradient strategy then retains the fraction $\lambda_r/L_0 \in [0,1]$ of Newton's correction in eigenmode u_r. The cross block $H_{\beta 0}$ subsequently maps that attenuated correction into the coefficient gradients. A diagonal entry alone does not describe this coupled block solve.

Rare classes nevertheless force small eigenvalues. Put $q = K - 1$. For a nonreference class $k < K$, the Rayleigh quotient in its coordinate direction is

$$e_k^\top H_{00} e_k = \frac{1}{n}\sum_i p_{ik}(1-p_{ik}) \le \bar{p}_k(1-\bar{p}_k). \tag{19}$$

submitted

Here $\bar{p}_k = n^{-1}\sum_i p_{ik}$; at intercept stationarity, it equals the empirical class prevalence. For the reference class, the corresponding direction changes all free-class log odds together. With $v = \mathbf{1}_q/\sqrt{q}$,

$$v^\top H_{00} v = \frac{1}{qn}\sum_i p_{iK}(1-p_{iK}) \le \frac{\bar{p}_K(1-\bar{p}_K)}{q}. \tag{20}$$

In either case, the smallest eigenvalue is no larger than the displayed Rayleigh quotient and therefore vanishes with the rare-class prevalence. When the probabilities are constant across observations, this structure is especially explicit. If $D = \mathrm{diag}(\bar{p}_1, \dots, \bar{p}_q)$ and $\bar{p} = (\bar{p}_1, \dots, \bar{p}_q)^\top$, then

$$H_{00} = D - \bar{p}\bar{p}^\top, \qquad H_{00}^{-1} = D^{-1} + \frac{1}{\bar{p}_K}\mathbf{1}_q\mathbf{1}_q^\top. \tag{21}$$

The first term adapts to a rare free class, while the second adapts the common shift to a rare reference class. The global-Lipschitz gradient step does not adapt to either. With $K = 10$, even a uniform marginal is only 0.1, and typical text or image-classification label distributions skew further. Consequently, low-curvature intercept modes arise naturally in multinomial models and are not confined to extreme class imbalance.

We implement block coordinate descent over classes. For each feature j, we cycle through the $K-1$ class coefficients using elementwise ℓ_1 soft-thresholding, and we update the intercept vector once per pass. An alternative is to cycle through one class at a time while holding the others fixed, as in glmnet's symmetric-multinomial path. We return to that scheme after presenting the block-CD results.

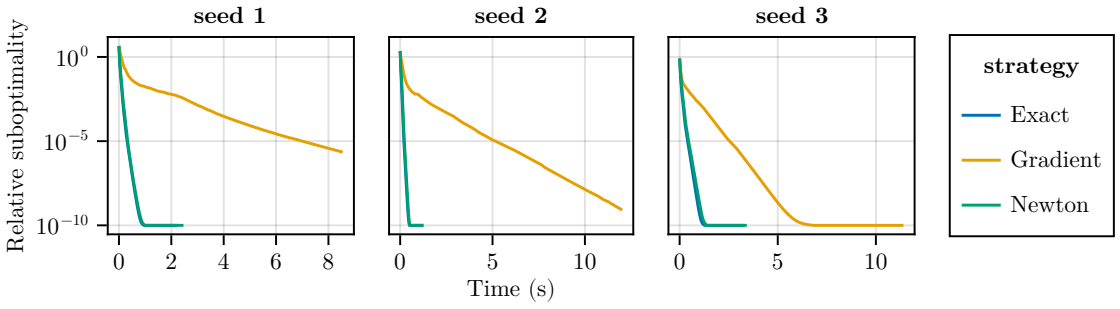


Figure 21: Imbalanced multinomial logistic regression. The panels show a relative suboptimality bound versus time for $K = 5$, class marginals $(0.7, 0.1, 0.1, 0.05, 0.05)$, $n = 500$, $p = 200$, $\lambda = 0.05\lambda_{\max}$, cyclic CD, and three random seeds.

We report a relative suboptimality bound (Figure 21). The multinomial CD solver constructs a feasible dual point by blending the centered residual with the intercept-only feasible point until every implied probability lies in the simplex, then rescaling onto the ℓ_∞ ball $\|X^\top\Theta\|_\infty \le \lambda$. Within each problem, we score every run against the largest dual value attained by any strategy. A stalling run is therefore assessed against a shared lower bound, and the curves cannot fall below the resolution of that certificate.

The gradient strategy slows sharply, as expected. Here class 4 is a free rare class, while class 5 is the rare reference class. Using the nominal prevalence 0.05, the two relevant Rayleigh quotients near intercept stationarity are bounded by $0.05\times0.95 = 0.0475$ in the class-4 coordinate and $0.05\times0.95/4 \approx 0.0119$ in the common-shift direction (Equation 19–Equation 20). Relative to the Böhning majorant $L_0 = 1/2$, these are at most 0.095 and 0.024. The gradient correction is therefore strongly attenuated

submitted

in at least one intercept eigenmode, although neither coordinate direction need be an eigenvector (Equation 18).

The strategy reaches the same optimum but takes roughly an order of magnitude longer. Across the three seeds, Newton and exact converge within roughly 50–100 passes, while the gradient strategy takes several hundred and, in the slowest run, exhausts the thousand-pass budget.

The Newton and exact strategies follow the same visible curve. The $(K-1)$-block Newton solve uses the full inverse in Equation 21, including the off-diagonal coupling $H_{00,kl} = -(1/n)\sum_i p_{ik}p_{il}$. It therefore rescales the low-curvature eigenmodes that the gradient strategy attenuates, while the Armijo guard absorbs any residual cold-start transient that the Newton-coupling bias (Equation 10) identifies in the scalar setting.

We extend the test to a two-dimensional grid by shrinking the rarest class probability $\bar{p}_K$ to 0.005 and increasing the feature amplitude to four times the baseline value. The four corners appear in Figure 22; across the full sweep (Figure 25), the Newton and exact trajectories overlap throughout. They reach tight certificates in all but the two most extreme cells, while the gradient strategy generally slows as $\bar{p}_K$ shrinks and amplitude grows. In those two extreme cells ($\bar{p}_K = 0.005$ and amplitude at least 1.5), no strategy obtains a useful median certificate within 1,000 passes. Those cells do not isolate the intercept update and therefore do not support the comparison.

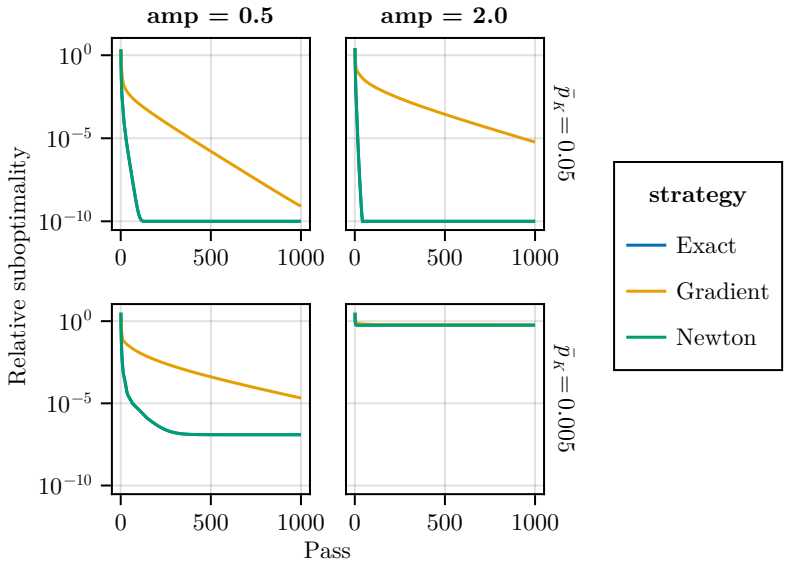


Figure 22: Multinomial imbalance-amplitude sweep. The panels show a relative suboptimality bound versus outer-pass index for the four corners of the rarest-class marginal $\bar{p}_K \in \{0.05, 0.005\}$ and feature-amplitude $\in \{0.5, 2.0\}$ grid on the same $K = 5$, $n = 500$, $p = 200$, $s = 5$, $\lambda = 0.05\lambda_{\max}$, $\rho = 0.3$ design as Figure 21; curves are medians over three seeds, with early-terminated trajectories carried forward at their terminal values.

For multinomial models, we therefore recommend a single Armijo-guarded Newton block step. The line search supplies the damping safeguard needed at a scalar cold start.

The same pattern holds on real high-dimensional multinomial data. The Yeoh2002 pediatric ALL gene-expression panel (Yeoh et al. 2002) is a $K = 6$ subtype problem on $n = 248$ samples and $p = 12\,625$ genes with class marginals BCR = 6%, MLL = 8%, E2A = 11%, T = 17%, Hyperdip = 26%, and TEL = 32%. The two rare classes BCR and MLL are nonreference classes, so they create low-

curvature directions in the intercept block (Equation 19). The gradient strategy should make little progress along those directions (Equation 18).

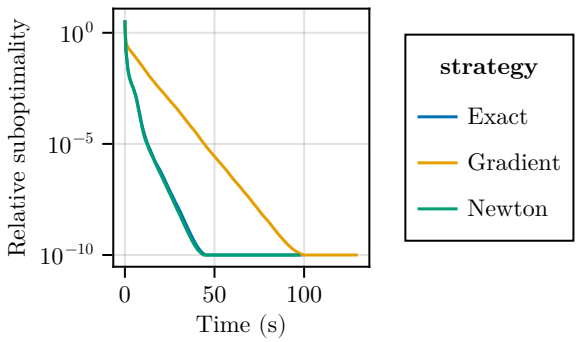


Figure 23: Multinomial logistic regression on Yeoh2002. The plot shows a relative suboptimality bound versus time at $\lambda = 0.05\lambda_{\max}$ under cyclic CD for $n = 248$, $p = 12\,625$, and $K = 6$. Suboptimality is bounded using the strongest shared feasible dual point.

On the $p > n$ Yeoh2002 gene-expression problem, the gradient strategy lags in the low-curvature modes associated with rare classes, while Newton and exact follow the same curve (Figure 23). This reproduces the synthetic pattern. The roughly fourfold slowdown at a relative suboptimality bound of 10^{-4} is milder than the order-of-magnitude slowdown of Figure 21. This is consistent with Yeoh2002's higher rare-class floor (6% rather than 5%) and may also reflect the penalty contribution when $n \ll p$.

The same separation appears when the dimensions are reversed. The StatLog Shuttle dataset (Michie et al. 1994) is a $K = 7$ sensor-data classification with $n = 58\,000$ observations and $p = 9$ features—the opposite shape to Yeoh2002—and class frequencies that span four orders of magnitude. The largest class contains nearly 80% of the observations, while the three rarest have only ten, thirteen, and fifty samples. Here the reference class is itself among the rarest, so the Rayleigh quotients fall much further below L_0 than in Yeoh2002 (Equation 19–Equation 20).

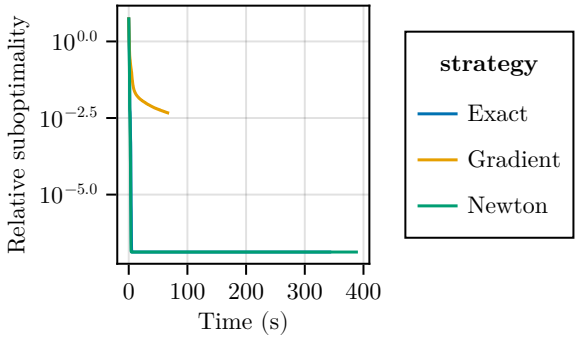


Figure 24: Multinomial logistic regression on StatLog Shuttle. The plot shows a relative suboptimality bound versus time at $\lambda = 0.05\lambda_{\max}$ under cyclic CD for $n = 58\,000$, $p = 9$, and $K = 7$. Suboptimality is bounded using the strongest shared feasible dual point.

On Shuttle, the Newton and exact strategies bring the relative suboptimality bound below 10^{-4} in about 25 passes, while the gradient strategy remains well above that threshold after the full thousand-pass budget. The slowdown is far more severe than the roughly fourfold gap on Yeoh because the

Submitted

rarest Shuttle classes are two orders of magnitude rarer. This is consistent with the smaller eigenmode correction factors in Equation 18; it is not a claim that a single Hessian diagonal controls the block solve. The rare-class slowdown therefore appears both when $p \gg n$ in the gene-expression data and when $n \gg p$ in the sensor data (Figure 24).

The class-by-class scheme of glmnet's symmetric-multinomial path leads to a parallel conclusion. Holding all η_{il} for $l \neq k$ fixed, the marginal subproblem in class k's linear predictors η_{ik} is, up to a per-observation offset $\log\left(1 + \sum_{l \neq k} e^{\eta_{il}}\right)$, a scalar logistic regression. The scalar Schur-coupling residual (Equation 12) therefore applies *per class*, and any rare class with extreme per-class response imbalance triggers exactly the gradient-strategy failure documented in the binary setting, now along the rare class's intercept coordinate. Multinomial models therefore expose two independent failure mechanisms: scalar imbalance under class-by-class CD and rare-class eigenmode attenuation under block CD. We recommend the Newton strategy for both schemes.

S2.2 Full imbalance–amplitude sweep

The four-corner view in Figure 22 shows the endpoints of the imbalance–amplitude experiment but hides the transition between them. Here we report all sixteen cells. We use the same $K = 5$ multinomial logistic design with $n = 500$, $p = 200$, $s = 5$, feature correlation $\rho = 0.3$, and $\lambda = 0.05\lambda_{\max}$. The rarest-class marginal ranges from 0.05 to 0.005, and the feature amplitude ranges from 0.5 to 2.0. Each curve is the median relative suboptimality bound over three seeds.

Across the grid, the gradient strategy generally stalls more severely as the rarest class becomes rarer and as feature amplitude grows; Newton and exact remain nearly indistinguishable. In the two cells with $\bar{p}_K = 0.005$ and amplitude at least 1.5, however, their shared median bound remains far from zero. At an expected rare-class count of only 2.5 and strong signal, near-separation and coefficient convergence dominate the intercept update. The grid therefore supports the curvature comparison over its certified region and marks the point at which this experiment ceases to isolate that mechanism. Where the gradient strategy reaches the 10^{-6} threshold, it generally needs at least an order of magnitude more passes than Newton. In most of the remaining certified cells, it misses that threshold within the 1 000-pass budget even though Newton reaches it.

S3. Experimental methods and reproducibility

We run all internal-solver experiments with the `Intercepts` Julia package on Julia 1.11 and the default OpenBLAS. Our primary measure of work is the number of outer passes. One pass consists of a sweep through all p coordinates and an intercept update. Unless noted, "passes to tolerance" means the number of completed passes needed to bring the shared relative suboptimality bound below 10^{-8}. Pass budgets (`maxit`) are 1000 for the single-λ figures and 5000 for the heatmap of Section 4.5; cells that miss tolerance are shown at the cap. We standardize the synthetic designs, use cyclic coordinate ordering unless noted, and set the default AR(1) feature correlation to $\rho = 0.6$. We fix the seeds per script. We average three seeds per cell in the heatmap of Section 4.5, show three seeds in Figure 4, and show five in the safeguard experiments. We use one fixed seed for the remaining single-trajectory figures.

Every coefficient update is accepted through a Tseng–Yun Armijo backtrack on the proximal model decrease, with slope constant 10^{-4}, step-halving, and at most 30 trials; in the common case the first trial is accepted. The `cdsolver` defaults match the paper's defaults (cyclic ordering, Newton intercept strategy), and every experiment script passes its configuration explicitly.

For scalar models, the exact strategy ordinarily returns when the normalized intercept gradient is at most 10^{-10}. If it consumes all 50 safeguarded Newton steps, a final check accepts a normalized gradient of at most 5×10^{-10}; larger residuals raise an error. This cap-only margin prevents saturated

submitted

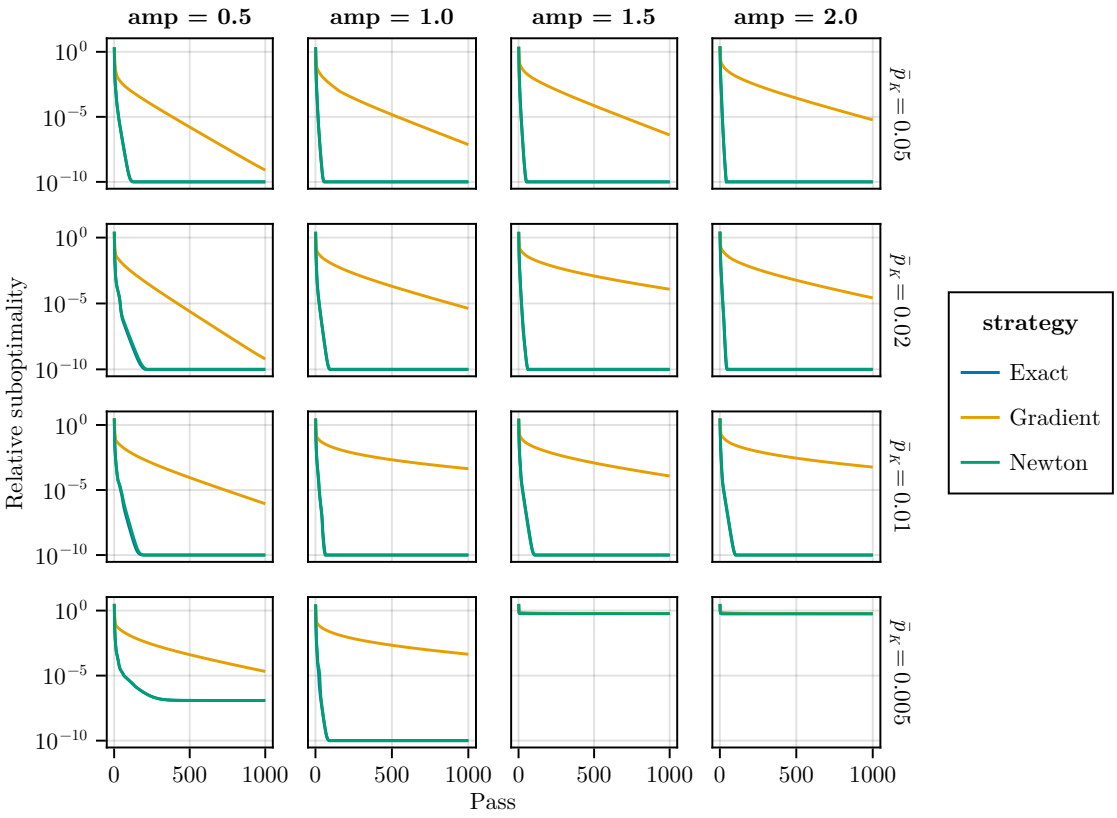


Figure 25: Full multinomial imbalance-amplitude sweep. The panels show a relative suboptimality bound versus outer-pass index over a 4×4 grid of rarest-class marginal $\bar{p}_K \in \{0.05, 0.02, 0.01, 0.005\}$ and feature amplitude $\in \{0.5, 1.0, 1.5, 2.0\}$. Curves are medians over three seeds for $K = 5$, $n = 500$, $p = 200$, $s = 5$, $\rho = 0.3$, and $\lambda = 0.05\lambda_{\max}$; early-terminated trajectories are carried forward at their terminal values.

floating-point predictors from stalling at the nominal boundary and remains twenty times smaller than the 10^{-8} reporting threshold. The multinomial implementation uses a strict 10^{-8} normalized block-gradient threshold and raises an error whenever it misses that target after 50 steps. Tests exercise both acceptance and rejection at the scalar cap.

At each diagnostic point, the raw residual lies in the loss-conjugate domain but need not satisfy dual stationarity for the unpenalized intercept. Simply subtracting its mean can leave that domain. We therefore blend the centered residual with the intercept-only feasible residual, taking the largest blend whose implied binomial probability, Poisson intensity, or multinomial probability vector remains in its conjugate domain. Both endpoints satisfy intercept stationarity. We then rescale the result toward zero to enforce $\|X^\top \Theta\|_\infty \leq \lambda$; this second step preserves the domain because it moves the implied mean toward the observed response. This feature-feasibility rescaling is the standard dual-scaling step used by Gap Safe rules (Fercoq et al. 2015; Ndiaye et al. 2017); the preceding anchor blend adds the conjugate-domain and unpenalized-intercept constraints needed here. The solver evaluates the conjugate only after checking domain membership and stops when its run-local relative duality gap reaches the stated tolerance.

For reporting, we put all strategies for the same problem instance on one reference. If $D_{s,t}$ is the feasible dual value from strategy s at diagnostic t, we set

$$\underline{D} = \max_{s,t} D_{s,t}, \qquad \overline{P} = \min_{s,t} P_{s,t}, \qquad B_{s,t} = \frac{P_{s,t} - \underline{D}}{|\overline{P}|}. \tag{22}$$

We use a floor of 10^{-15} for the denominator in code. Weak duality gives $P_{s,t} - P^\star \leq P_{s,t} - \underline{D}$, so $B_{s,t}$ is a relative suboptimality upper bound with a strategy-independent scale and reference. The certificate width $(\overline{P} - \underline{D})/|\overline{P}|$ is below 10^{-8} for every scalar trajectory figure in the paper. This is a retrospective comparison: a displayed curve may cross the reporting threshold before that run's own duality gap triggers termination. In the two longest rate-diagnostic sweeps, we compute the tight Newton reference first and stop a strategy as soon as its primal reaches the corresponding shared-bound threshold. This changes neither its iterates nor its reported crossing; it only avoids running beyond the point used in the figure.

For the real-data experiments, we use pinned datasets from Zenodo (DOI: 10.5281/zenodo.20315625) and standardize them before fitting. We include in the repository the code needed to retrieve these data and generate the shared standardized problem used in the production-solver comparison. In that comparison, we run glmnet 4.1-10 (R), biglasso 1.6-1 (R), adelie 1.0-8 (R), skglm 0.5 (Python), LIBLINEAR through scikit-learn's `solver="liblinear"` wrapper, and BlitzL1 at commit `aef8d02`. Given the cached inputs, the drivers are otherwise deterministic, so we run each solver once per tolerance setting while sweeping its native convergence parameter over a broad range. We document the exact commands, tolerance grids, and iterate caps with the repository materials.

We store all experimental outputs in the repository. We therefore render most figures from cached files, and regenerating a panel requires rerunning the corresponding experiment. By contrast, the illustrative Figure 1 runs at render time as a small reproducible example. We pin the Julia, R, and Python toolchains in the repository; they use their default OpenBLAS builds without thread pinning. We ran the experiments on a single x86_64 Linux workstation with an AMD Ryzen 9 7900 (12 cores, 24 threads) and 64 GB of RAM. The wall-clock measurements are therefore indicative rather than benchmark-grade.

submitted

S4. Robustness analyses

S4.1 Cyclic versus permuted ordering

The within-pass drift in Equation 6 does not depend on coordinate ordering, but strict cyclic order deterministically pairs the intercept solve with the same first coordinate on every pass, which could compound the drift.

On w1a, we find no such compounding under the per-coordinate Armijo backtracking used by our coordinate-descent solver, `cdsolver`, after every coordinate update (Supplement S3). All three strategies converge under both orderings, and the pass-count differences are less than a factor of two (Figure 26). Newton and exact remain below 100 passes under either ordering, whereas the gradient strategy takes about 340 under both. The qualitative ranking—Newton and exact fast, gradient slow—is unchanged, and we use cyclic CD as the default elsewhere in the paper.

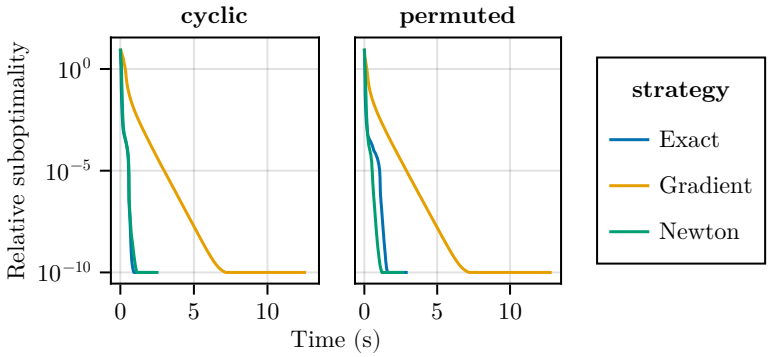


Figure 26: Ordering comparison on w1a. The panels show a relative suboptimality bound versus time under cyclic (left) and permuted (right) coordinate ordering for the same configuration as Figure 2.

S4.2 Warm-started path

In the single-λ experiments above, we use the most difficult regime: a cold start with a large $|\partial_0 F|$ at the zero iterate. The drift in Equation 6 and the small correction in Equation 12 have their largest effects there.

In practice, users rarely solve for a single λ in isolation. Instead, they sweep a grid $\lambda_1 > \lambda_2 > \cdots > \lambda_K$ and warm-start each subproblem from the previous solution. As the argument above suggests, warm-starting begins each subproblem with $|\partial_0 F|$ already small. A single Newton step is then essentially exact at the start, so $k_0 \in \{1, 2\}$ in the exact strategy and the $k_0 - 1$ per-pass overhead evaporates. The gradient-strategy slowdown should also lessen because the iterate is close to optimum and the H_{00}/L_0 shrinkage acts on an already-small residual (Equation 10).

We test this by solving an ℓ_1-logistic path on two problems—a simulated standardized design ($n = 500$, $p = 1000$, $s = 10$, $\mu_0 = 0.9$, $\rho = 0.6$) and the w1a benchmark used in Figure 2—over a geometrically spaced grid of $K = 20$ values from $\lambda_{\max}$ down to $5 \cdot 10^{-3}\lambda_{\max}$.

Each subproblem runs until its local relative duality gap reaches 10^{-8} or to a budget of 300 outer passes / 20 wall-seconds, whichever comes first. We then report the first pass at which the shared relative suboptimality bound reaches 10^{-8}. We compare the three strategies under both warm start (initialized from the previous-λ solution) and cold start (re-initialized at zero), and under both cyclic and permuted feature ordering. The main-text panels show cyclic CD; Figure 31 and Figure 32 give the full grid over both orderings.

submitted

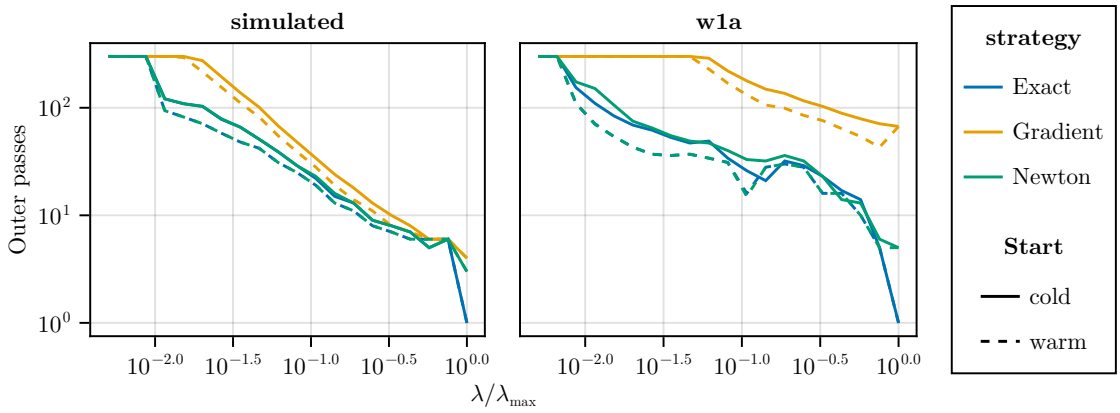


Figure 27: Warm-started and cold-started regularization paths. The panels show outer passes per subproblem along a logistic-regression λ path under cyclic CD; dashed lines use warm starts, solid lines cold starts, and curves capped at 300 hit the pass budget. The left panel shows the simulated imbalanced problem ($\mu_0 = 0.9$) and the right panel w1a.

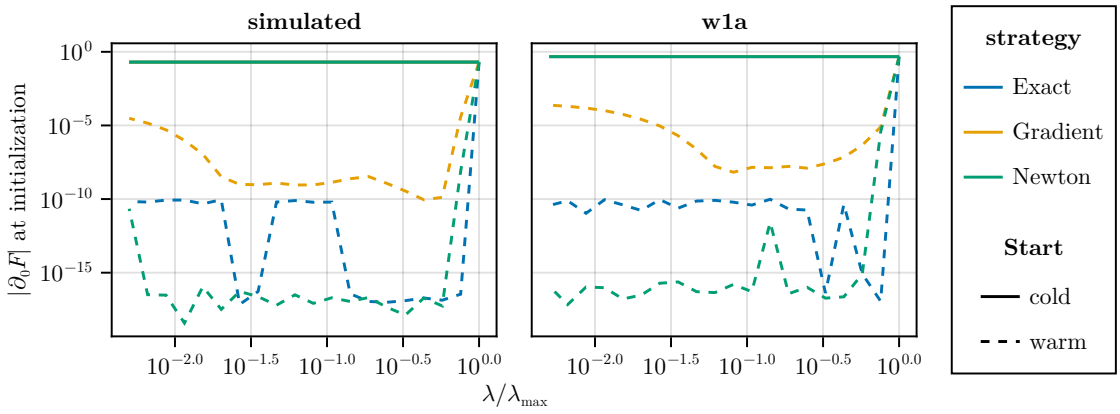


Figure 28: Initial intercept gradients along the regularization path. The panels show $|\partial_0 F|$ at the initial iterate of each subproblem for the same path and cyclic-CD setup as Figure 27; dashed lines use warm starts and solid lines cold starts.

In Figure 28, warm-starting drives $|\partial_0 F|$ toward zero as λ shrinks and the previous-λ solution moves closer to the current optimum. Cold start, by contrast, sets the initial iterate to zero at every λ, so $|\partial_0 F|$ is the marginal class imbalance—roughly 0.20 on the simulated problem and 0.47 on w1a, identical across strategies.

The pass-count reduction is modest and uneven (Figure 27). Averaged over the simulated path, warm-starting saves about 10% of the passes for all three strategies. On w1a, it again helps the gradient strategy only modestly, while Newton and exact often improve by roughly 20%–40%. The number of budget-capped subproblems does not fall uniformly, but Newton and exact remain close under every configuration. This is what Equation 6 suggests: once $|\partial_0 F|$ is already small, the $k_0 - 1$ per-pass penalty of the exact strategy largely disappears.

Warm-starting does not rescue the gradient strategy. At the smallest λ under cyclic ordering, its relative bound improves by roughly two orders of magnitude on the simulated problem and by more than one order on w1a. It still remains well above the corresponding Newton and exact bounds. Thus, a better initial point helps without compensating for the small H_{00}/L_0 correction.

These results leave the practical recommendation unchanged: use a safeguarded Newton step in direct CD and a fully resolved intercept on the frozen quadratic at a prox-Newton/IRLS boundary. The exact original-loss strategy is also defensible in a warm-started path solve, where its extra work is small.

S4.3 Per-pass cost decomposition

The pass-count and wall-clock figures above combine two quantities that the within-pass drift (Equation 6) separates: the *number of outer passes* needed to reach a tolerance and the *per-pass inner cost* of the intercept update. We test whether the exact strategy pays extra inner work per outer pass without a commensurate rate improvement. We count its Newton directions, but floor the count at one because even a zero-step solve must evaluate the conditional residual. Local quadratic convergence explains why few directions remain once an iterate is near the conditional minimizer, but it does not supply a global bound on the work.

We instrument the solver to record k_0 for every outer pass. We compare the resulting cost trajectories with the relative-suboptimality trajectories on the imbalanced simulated design of Figure 5 ($n = 500$, $p = 1000$, $s = 10$, cyclic CD, $\lambda/\lambda_{\max} = 0.05$), drawn with an independent seed. The solver runs to a local relative-duality-gap tolerance of 10^{-10}, and we report progress to the shared 10^{-8} bound (Equation 22), sweeping $\mu_0 \in \{0.5, 0.9, 0.99\}$ to vary the cold-start magnitude of $|\partial_0 F|$.

The two figures show the overhead directly. In Figure 29, the inner work peaks between two and five units on the cold-start pass, with larger peaks under greater imbalance, then decays to one as the iterate approaches the joint optimum. Summed through the shared 10^{-8} bound, exact uses roughly one-and-a-half to twice the work of Newton, again increasing with imbalance.

The extra work is hard to justify: the Newton and exact strategies trace almost the same relative-suboptimality curve and reach 10^{-8} in comparable pass counts that differ by at most one in these runs. The exact strategy spends more inner work as imbalance grows without improving outer-pass progress (Figure 30).

The penalty is modest in these runs: k_0 remains well below the inner-loop cap of 50, and most passes need at most one Newton direction. Every returned update satisfies the nominal 10^{-10} conditional-gradient tolerance because no update reaches the cap; the largest observed k_0 is five. A cap hit would invoke the final numerical check documented in Supplement S3, not automatically produce a cached result. This behavior is consistent with local quadratic convergence after the iterates approach their conditional minimizers, but the experiment, rather than a global Newton bound, establishes the

submitted

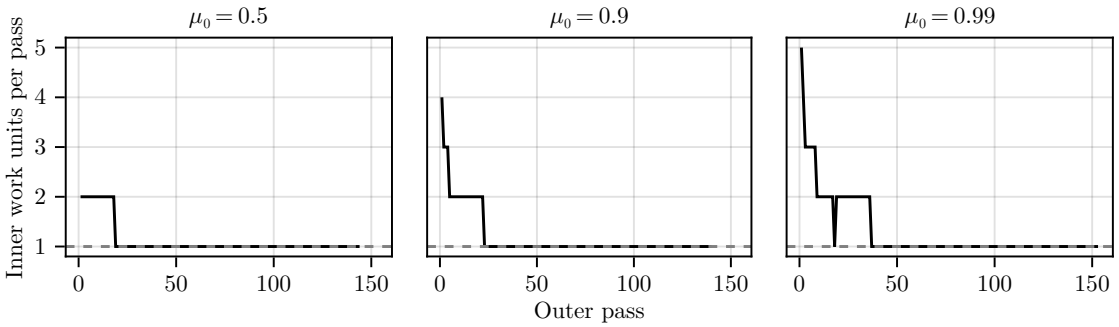


Figure 29: Per-pass inner work under the exact strategy. The panels show the Newton-direction count, floored at one for the mandatory residual check, versus outer-pass index for $\mu_0 \in \{0.5, 0.9, 0.99\}$ on an independent draw of the simulated design of Figure 5. The dashed line at $k_0 = 1$ marks the Newton strategy.

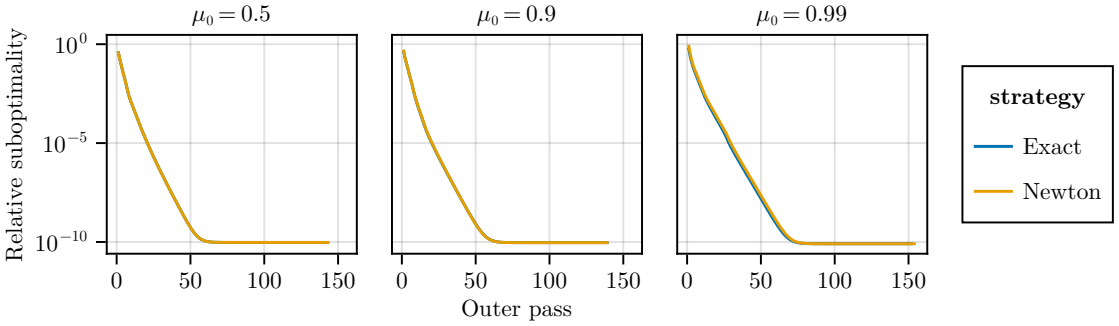


Figure 30: Per-pass progress on the same runs. The panels show a relative suboptimality bound after each outer pass for the Newton and exact strategies on the experiments of Figure 29.

reported cost. The exact strategy incurs most of its overhead in the first few cold-start passes, when $|\partial_0 F|$ is largest. The warm-start regime of Supplement S4.2 largely avoids that window.

S4.4 Full warm-start grids

The cyclic-CD panels in Supplement S4.2 show the ordering used throughout most of the article. Here we restore the permuted-CD runs to display the complete experiment. We solve a $K = 20$ regularization path from $\lambda_{\max}$ to $5 \cdot 10^{-3}\lambda_{\max}$ on two problems: the simulated design with $n = 500$, $p = 1000$, $s = 10$, $\mu_0 = 0.9$, and $\rho = 0.6$, and the w1a benchmark. Each subproblem starts either from zero or from the preceding solution and runs until its local relative duality gap reaches 10^{-8}, a budget of 300 outer passes, or a 20-second time limit. We report passes to the shared 10^{-8} suboptimality bound.

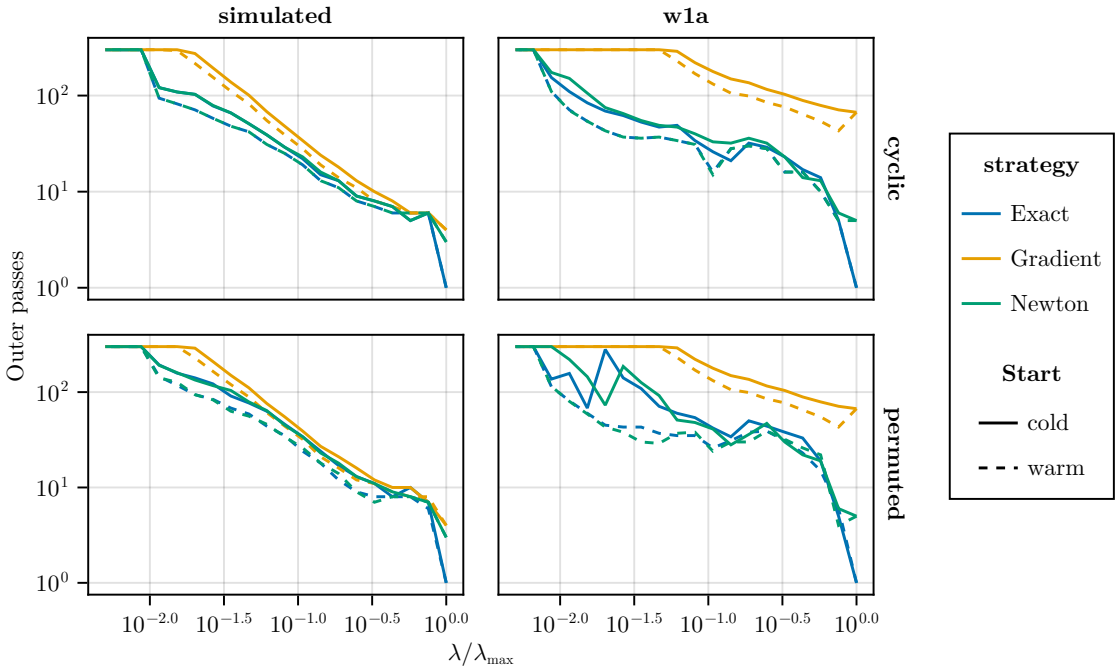


Figure 31: Full warm-started and cold-started regularization paths. The panels show outer passes per subproblem along the logistic-regression λ path on the simulated imbalanced design and w1a for cyclic CD (top) and permuted CD (bottom). Dashed lines use warm starts, solid lines use cold starts, and curves capped at 300 hit the pass budget.

The full pass-count grid confirms that feature ordering does not change the comparison among intercept strategies. Warm starts reduce average pass counts under every configuration, but they remove a budget cap in only two cases: the cyclic gradient run on the simulated path and the permuted Newton run on w1a. Newton and exact remain close under both orderings. The gradient strategy still slows most sharply at small λ, especially on w1a, where its conservative intercept correction controls the rate.

To connect these pass counts to the proposed mechanism, Figure 32 reports the intercept gradient before the first pass of every subproblem. This quantity measures how much intercept error the new subproblem inherits from its initialization.

Under both orderings, warm starting drives $|\partial_0 F|$ toward zero as the path proceeds, whereas cold starting resets it to the marginal class imbalance at every value of λ. The repeated pattern links the

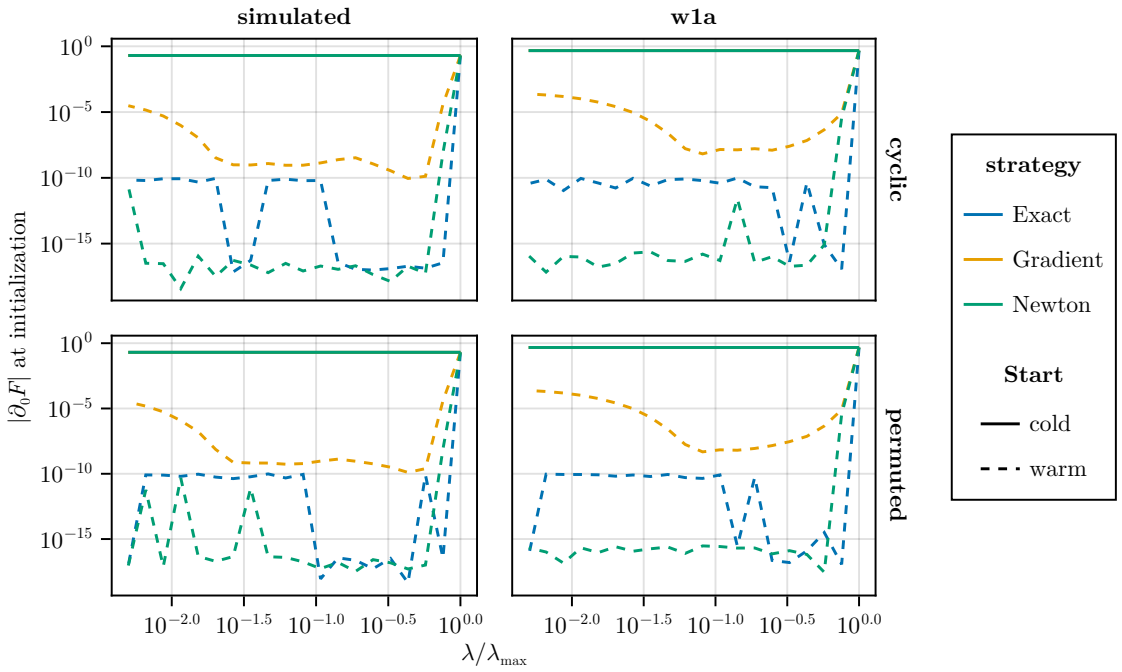


Figure 32: Full initial-gradient regularization paths. The panels show $|\partial_0 F|$ at the initial iterate of each subproblem along the $K = 20$ λ path on the simulated imbalanced design and w1a for cyclic CD (top) and permuted CD (bottom). Dashed lines use warm starts, and solid lines use cold starts.

Newton–exact agreement in Figure 31 to the initialization rather than to cyclic ordering: when the inherited intercept gradient is already small, resolving the conditional intercept problem beyond one guarded Newton step adds little.

S5. Production-solver diagnostics

S5.1 Poisson family

Poisson loss rules out several production solvers before any trajectory is recorded. Because $f''(\eta) = e^{\eta}$ is unbounded on $\mathbb{R}$, no global Lipschitz upper bound exists. Consequently, neither the $w \equiv 1/4$ MM majorant of glmnet `modified.Newton` and biglasso `MM` nor the constant-Lipschitz coordinate step of skglm `AndersonCD` has a Poisson analog. The status of each solver on this family is structural, not empirical:

- glmnet supports `family = "poisson"` with a single fitting mode (local-IRLS); no `type.poisson` switch exists.
- biglasso does not implement `family = "poisson"` at all; its signature is restricted to `gaussian`, `binomial`, `cox`, and `mgaussian`.
- skglm `AndersonCD` requires `datafit.get_lipschitz()`, which the `Poisson` datafit declines to implement (`AttributeError: Poisson is not compatible with solver AndersonCD`). The `AndersonCD` solver therefore cannot fit a Poisson model.
- skglm `ProxNewton` pairs with the `Poisson` datafit through local prox-Newton subproblems and represents skglm in the panel.
- adelie supports `glm.poisson(y)` through the same direct-Newton / weighted-centered-features parameterization as the logistic case.

submitted

Three solvers based on local Newton curvature therefore remain: glmnet, adelie, and skglm `ProxNewton`. We run each in path mode on congress109 with $\lambda = 0.05\lambda_{\max}$, sweeping the solver-specific tolerance (Figure 33).

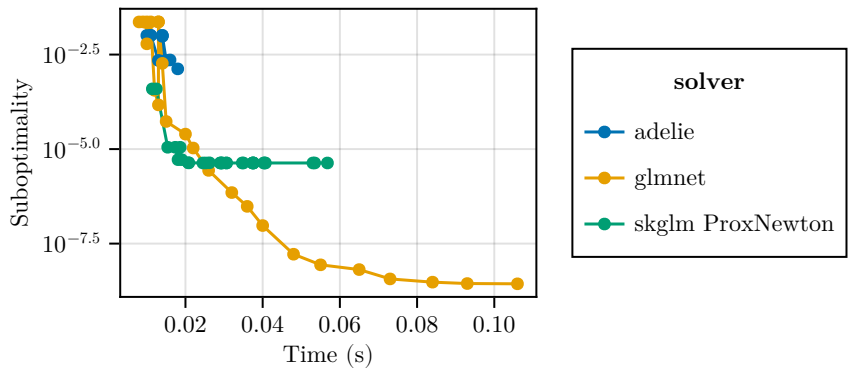


Figure 33: Production Poisson solvers on congress109. The plot shows a suboptimality bound versus wall-clock time for glmnet, adelie, and skglm `ProxNewton` in path mode at $n = 529$, $p = 999$, $\bar{y} = 11.83$, and $\lambda = 0.05\lambda_{\max}$. Every solver uses the common certified feasible dual reference $D^\star$.

The within-package weight-scheme switch used in Figure 12 and Figure 13 has no Poisson analog. The loss itself does not admit the $w \equiv 1/4$ surrogate used by the gradient-class solvers. Every working production Poisson solver in our survey therefore uses local Newton curvature, as the analysis recommends.

S5.2 Single-lambda cold start

The path-mode protocol used above lets inner CD avoid the transient at $\beta = 0$. When we instead solve a single λ from scratch, an unguarded Newton intercept update can become unstable (Equation 10). We examine this possibility in Table 4 at $\lambda = 0.05\lambda_{\max}$ with loose inner and outer solver tolerances of 10^{-1}, sweeping the iterate cap from 10^1 to 10^5; the path-mode reference primal is 0.0634 on w1a and 0.0497 on news20-3pct. These settings form a targeted diagnostic rather than a test of either package's default configuration, whose convergence tolerances are stricter.

Table 4: Single-λ cold-start diagnostic on w1a and news20-3pct. Each row reports the intercept estimate, primal objective, and iteration count at $\lambda = 0.05\lambda_{\max}$ with inner and outer solver tolerances of 10^{-1}; the iterate cap is swept from 10^1 to 10^5, and the path-mode reference primal is 0.0634 on w1a and 0.0497 on news20-3pct.

dataset	solver	β_0	primal	iterations
w1a	biglasso `alg.logistic = "Newton"`	-723 (range $[-1722, -291]$)	16.19	10^5 cap, oscillates
w1a	biglasso `alg.logistic = "MM"`	-3.51	0.088	2
w1a	adelie	—	—	empty state returned
news20-3pct	biglasso `alg.logistic = "Newton"`	-3.48	0.310	2 (premature stop)

submitted

dataset	solver	β_0	pri-mal	iterations
news20-3pct	biglasso `alg.logistic = "MM"`	−3.48	0.089	2
news20-3pct	adelie	−5.22	0.052	converges

In Table 4, the global majorant $L_0 = 1/4$ acts as an X-free safeguard on the intercept: two scalars (n and $\bar{y}$) determine the step, and that is enough to absorb the cold-start transient at trivial cost on the unpenalized coordinate. The analogous safeguard for a coefficient would require column-norm information.

Under this specific configuration, biglasso's default Newton mode returns inaccurate fits on both datasets: it reaches the iteration cap while oscillating on w1a and stops prematurely on news20-3pct. Adelie, whose only mode uses local Newton curvature, returns an empty state on w1a but converges on news20-3pct. The contrast between the datasets, despite their matched $\bar{y}$, is consistent with the failure depending on more than the cold-start gradient magnitude. A plausible explanation is that, on news20-3pct's wider sparse design, the ℓ_1 penalty pulls many entries to zero before the linear predictor reaches extreme values, damping the IRLS weight collapse. The diagnostic does not establish behavior at other values of λ, under other stopping tolerances, or for the packages' otherwise default calls.

References

Bertrand, Quentin, Quentin Klopfenstein, Pierre-Antoine Bannier, Gauthier Gidel, and Mathurin Massias. 2022. "Beyond L1: Faster and Better Sparse Models with Skglm." *Advances in Neural Information Processing Systems 35* (New Orleans, USA) 35 (November–December): 38950–65.https://proceedings.neurips.cc/paper_files/paper/2022/hash/fe5c31e525e9a26a1426ab0b589f42fe-Abstract-Conference.html .

Böhning, Dankmar. 1992. "Multinomial Logistic Regression Algorithm." *Annals of the Institute of Statistical Mathematics* 44 (1): 197–200. https://doi.org/10.1007/BF00048682.

Breheny, Patrick, and Jian Huang. 2011. "Coordinate Descent Algorithms for Nonconvex Penalized Regression, with Applications to Biological Feature Selection." *The Annals of Applied Statistics* 5 (1): 232–53. https://doi.org/10.1214/10-AOAS388.

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. "LIBLINEAR: A Library for Large Linear Classification." *Journal of Machine Learning Research* 9 (61): 1871–74. http://jmlr.org/papers/v9/fan08a.html.

Fercoq, Olivier, Alexandre Gramfort, and Joseph Salmon. 2015. "Mind the Duality Gap: Safer Rules for the Lasso." In *Proceedings of the 32nd International Conference on Machine Learning*, edited by Francis Bach and David Blei, vol. 37. Proceedings of Machine Learning Research. PMLR. https://proceedings.mlr.press/v37/fercoq15.html.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1–22. https://doi.org/10.18637/jss.v033.i01.

submitted

Hunter, David R., and Kenneth Lange. 2004. "A Tutorial on MM Algorithms." *The American Statistician* 58 (1): 30–37.

Johnson, Tyler B, and Carlos Guestrin. 2015. "Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization." *Proceedings of the 32nd International Conference on Machine Learning* (Lille, France) 37: 9. http://proceedings.mlr.press/v37/johnson15.pdf.

Krishnapuram, Balaji, Lawrence Carin, Mário A. T. Figueiredo, and Alexander J. Hartemink. 2005. "Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (6): 957–68. https://doi.org/10.1109/TPAMI.2005.127.

Lee, Jason D., Yuekai Sun, and Michael A. Saunders. 2014. "Proximal Newton-Type Methods for Minimizing Composite Functions." *SIAM Journal on Optimization* 24 (3): 1420–43. https://doi.org/10.1137/130921428.

Lovell, Michael C. 1963. "Seasonal Adjustment of Economic Time Series and Multiple Regression Analysis." *Journal of the American Statistical Association* 58 (304): 993–1010. https://doi.org/10.1080/01621459.1963.10480682.

Massias, Mathurin, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon. 2020. "Dual Extrapolation for Sparse Generalized Linear Models." *Journal of Machine Learning Research* 21 (234): 1–33. http://jmlr.org/papers/v21/19-587.html.

McCullagh, Peter, and John A. Nelder. 1989. *Generalized Linear Models.* 2nd ed. Chapman; Hall.

Michie, D., D. J. Spiegelhalter, and C. C. Taylor. 1994. *Machine Learning, Neural and Statistical Classification.* Edited by D. Michie, D. J. Spiegelhalter, and C. C. Taylor. Ellis Horwood.

Ndiaye, Eugene, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. 2017. "Gap Safe Screening Rules for Sparsity Enforcing Penalties." *Journal of Machine Learning Research* 18 (128): 1–33. https://jmlr.org/papers/v18/16-577.html.

Nesterov, Yu. 2012. "Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems." *SIAM Journal on Optimization* 22 (2): 341–62. https://doi.org/10.1137/100802001.

Platt, John C. 1998. "Fast Training of Support Vector Machines Using Sequential Minimal Optimization." In *Advances in Kernel Methods: Support Vector Learning*, 1st ed., edited by Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola. MIT Press. https://doi.org/10.7551/mitpress/1130.003.0016.

Richtárik, Peter, and Martin Takáč. 2014. "Iteration Complexity of Randomized Block-Coordinate Descent Methods for Minimizing a Composite Function." *Mathematical Programming* 144 (1–2): 1–38. https://doi.org/10.1007/s10107-012-0614-z.

Taddy, Matt. 2015. "Distributed Multinomial Regression." *The Annals of Applied Statistics* 9 (3): 1394–414. https://doi.org/10.1214/15-AOAS831.

Tay, J. Kenneth, Balasubramanian Narasimhan, and Trevor Hastie. 2023. "Elastic Net Regularization Paths for All Generalized Linear Models." *Journal of Statistical Software* 106 (1): 1–31. https:

submitted

//doi.org/10.18637/jss.v106.i01.

Wright, Stephen. 2015. "Coordinate Descent Algorithms." *Mathematical Programming: Series B* 151 (1): 3–34. https://doi.org/10.1007/s10107-015-0892-3.

Yang, James, and Trevor Hastie. 2024. *A Fast and Scalable Pathwise-Solver for Group Elastic Net.* https://arxiv.org/abs/2405.08631.

Yeoh, Eng-Juh, Mary E. Ross, Sheila A. Shurtleff, et al. 2002. "Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Acute Lymphoblastic Leukemia by Gene Expression Profiling." *Cancer Cell* 1 (2): 133–43. https://doi.org/10.1016/S1535-6108(02)00032-6.

Yuan, Guo-Xun, Chia-Hua Ho, and Chih-Jen Lin. 2012. "An Improved GLMNET for L1-Regularized Logistic Regression." *Journal of Machine Learning Research* 13 (64): 1999–2030. http://jmlr.org/papers/v13/yuan12a.html.

Zeng, Yaohui, and Patrick Breheny. 2021. "The Biglasso Package: A Memory- and Computation-Efficient Solver for Lasso Model Fitting with Big Data in R." *The R Journal* 12 (2): 6–19. https://doi.org/10.32614/RJ-2021-001.

submitted