Greedy Walks on Neighbour-Hyperplane Graphs in p-adic Regression

(Author details omitted)

February 25, 2026

Abstract

In ultrametric regression problems (including p-adic regression), the objective is typically locally constant and "gradient-like" training rules are uninformative. One natural alternative is a discrete greedy descent on a finite set of candidate models. This note records a simple computational experiment motivated by the following phenomenon: when the loss is ultrametric, a best-fit hyperplane can be chosen to pass through $d{+}1$ data points (for d features), so there are finitely many candidates. If we define two hyperplanes to be *neighbours* when they differ by exactly one of their defining points, we obtain a natural graph on candidate hyperplanes. A naive optimiser then repeatedly moves to the neighbouring hyperplane with the largest loss decrease, stopping at a local minimum.

We implement this neighbour-hyperplane graph for a toy p-adic linear regression model with synthetic p-adic noise, enumerate all hyperplanes through $d{+}1$ points, *deduplicate identical coefficient vectors*, and measure how often several *improvement-only descent policies* reach a *global* minimum from a random start. Besides deterministic steepest descent, we consider randomised policies that choose among improving moves uniformly, proportional to improvement size, or via a softmax with temperature. Across the tested regimes, improvement-only greedy descent frequently gets trapped: in the p-adic hyperplane graphs we observe many local minima and only moderate "global basin" fractions.

1 Neighbour hyperplanes and greedy descent

Fix a dataset of n points (x_i, y_i) with $x_i \in \mathbb{Z}^d$ and $y_i \in \mathbb{Z}$. A (affine) hyperplane is a parameter vector

$$\beta = (\beta_0, \beta_1, \ldots, \beta_d)$$

representing the prediction $\hat{y} = \beta_0 + \sum_{j=1}^{d} \beta_j x_j$.

Definition 1 (Ultrametric residual loss)**.** *Fix a prime p and define the p-adic absolute value on rationals by $|q|_p = p^{-v_p(q)}$ with $|0|_p = 0$. For a parameter vector β define the residuals*

$$r_i(\beta) = y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{i,j},$$

and the ℓ_1 ultrametric loss

$$L(\beta) = \sum_{i=1}^{n} |r_i(\beta)|_p.$$

The key property motivating a finite search is that, under ultrametric losses, best fits can be chosen to interpolate points. We take that as an organising principle: enumerate candidate hyperplanes by selecting $d{+}1$ points and solving the corresponding linear system over $\mathbb{Q}$.

Algorithm 1 Steepest-descent walk on neighbour hyperplanes

Require: Candidate set $\{\beta(S)\}$ indexed by $(d{+}1)$-subsets S, loss $L(\cdot)$

1: Choose an initial subset S
2: **while** there exists a neighbour S' of S with $L(\beta(S')) < L(\beta(S))$ **do**
3: Set $S \leftarrow \arg\min_{S' \sim S} L(\beta(S'))$ (break ties deterministically)
4: **end while**
5: **return** local minimum S

Definition 2 (Neighbour hyperplanes)**.** *Let $S \subset \{1, \ldots, n\}$ be a subset of indices of size $d{+}1$. Let $\beta(S)$ be the hyperplane interpolating the points in S (if the system is nonsingular). Two such subsets S and S' are* neighbours *if $|S \cap S'| = d$ (i.e. they differ by exactly one point).*

Different $(d{+}1)$-subsets can define the *same* hyperplane (for example, if the dataset is exactly linear and all points lie on one hyperplane). In our implementation we therefore *deduplicate* by coefficient vector and treat the graph nodes as *unique* hyperplanes; an edge between two hyperplanes exists if it is induced by at least one single-point swap between defining subsets.

Any *improvement-only* descent policy repeatedly moves to a neighbouring node with strictly lower loss, stopping at a local minimum.

2 Synthetic p-adic regression experiments

2.1 Noise models

There is no single canonical analogue of "Gaussian noise" on $\mathbb{Q}_p$. A standard baseline on $\mathbb{Z}_p$ is Haar-uniform noise. We use two simple integer-valued approximations:

- **Haar (truncated)**: sample U uniformly modulo p^M and return $\varepsilon = p^{k_0}U$ (with a symmetric representative for U).
- **Valuation+unit**: sample a valuation $K \geq k_0$ with a geometric tail, choose a small coefficient a not divisible by p, and return $\varepsilon = a\,p^K$.

Both reflect the idea that $v_p(\varepsilon)$ is typically k_0 but occasionally larger.

2.2 Experimental design

For each trial we sample an integer "ground-truth" hyperplane and integer covariates x_i, then set

$$y_i = \beta_0^\star + \sum_{j=1}^{d} \beta_j^\star x_{i,j} + \varepsilon_i.$$

In the runs reported here, the generating coefficients are sampled as p-powers (i.e. $\beta_j^\star \in \{\pm p^k\}$ for a small exponent range).

We then enumerate all hyperplanes through $d{+}1$ points, deduplicate identical coefficient vectors, compute $L(\beta)$ for each unique hyperplane, and analyse several improvement-only descent policies on the induced neighbour graph:

- **Global-hit probability**: for a uniformly random start node, the probability that the descent process ends at a global minimum.
- **Local minima count**: the number of greedy sinks.
- **True-is-global**: whether the generating $\beta^\star$ is itself globally optimal under L.

p	model	k_0	steepest	uniform	proportional	softmax (T=1)
3	haar	0	0.346	0.329	0.360	0.349
3	haar	1	0.346	0.331	0.361	0.339
3	haar	2	0.346	0.329	0.360	0.332
3	haar	3	0.346	0.329	0.360	0.330
3	valuation	0	0.360	0.341	0.371	0.360
3	valuation	1	0.360	0.341	0.371	0.349
3	valuation	2	0.360	0.341	0.371	0.344
3	valuation	3	0.360	0.341	0.371	0.342
11	haar	0	0.221	0.189	0.210	0.213
11	haar	1	0.221	0.189	0.210	0.191
11	haar	2	0.221	0.189	0.210	0.189
11	haar	3	0.221	0.189	0.210	0.189
11	valuation	0	0.227	0.192	0.211	0.214
11	valuation	1	0.227	0.192	0.211	0.194
11	valuation	2	0.227	0.192	0.211	0.192
11	valuation	3	0.227	0.192	0.211	0.192

Table 1: Global-hit probabilities for improvement-only descent policies on the neighbour-hyperplane graph ($n = 20, d = 3$). Each entry is the mean global-hit probability over trials for that configuration. (Standard errors and additional diagnostics are recorded in `outputs/2026-02-20/analysis/experiment_report.md`.)

The descent policies we test are:

- **Steepest**: choose a neighbour with maximal loss decrease (deterministic tie-break).
- **Uniform**: choose uniformly among all improving neighbours.
- **Proportional**: choose with probability proportional to improvement size.
- **Softmax**: choose among improving neighbours with probability proportional to $\exp(\Delta/T)$ where Δ is the loss decrease and $T > 0$ is a temperature (we use $T = 1$).

In all runs we used $n = 20$ points and $d = 3$ features (so each candidate hyperplane is defined by 4 points), coefficient and covariate bounds 5, loss type ℓ_1, and noise precision parameter $M = 6$.

2.3 Results

Table 1 summarises global-hit probabilities for each descent policy. Two qualitative findings stand out:

1. With noise, the neighbour-hyperplane landscape has *many* bad local minima; improvement-only descent reaches a global minimum only a fraction of the time (typically ≈ 0.15–0.40 here), and the exact probability depends modestly on the policy.
2. In the noisy settings tested here, the generating hyperplane $\beta^\star$ is never globally optimal. Intuitively, an interpolating hyperplane can "spend" degrees of freedom to make d+1 residuals exactly zero while keeping the remaining residual valuations high, so the optimum tends to overfit rather than recover $\beta^\star$.

Figure 1 highlights the modest policy dependence and the difference between two baseline noisy models.

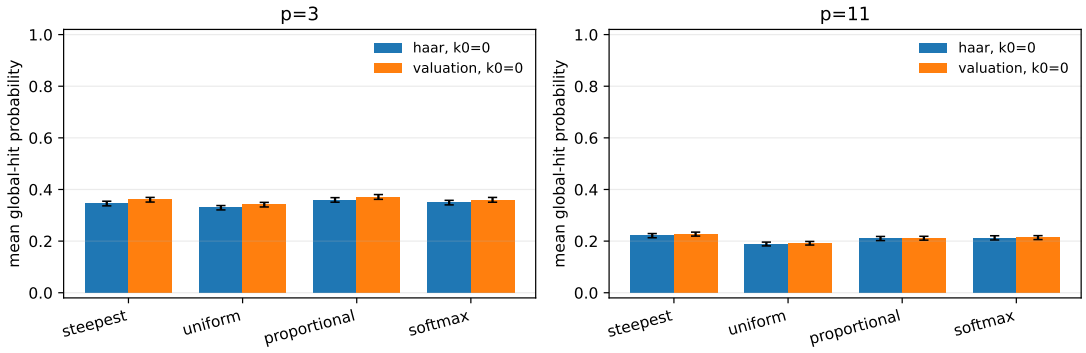

Figure 1: Baseline noisy global-hit probabilities by policy, comparing haar($k_0 = 0$) vs valuation($k_0 = 0$). Error bars are standard errors across trials.

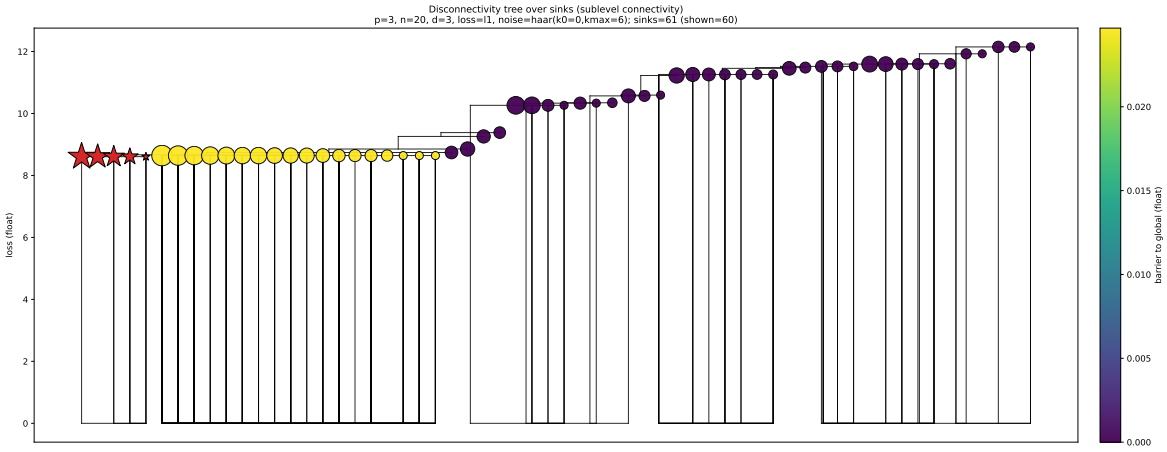

Figure 2: Disconnectivity tree over sinks in one representative landscape snapshot (seed 0). Leaves are sinks; merge heights are sublevel connectivity thresholds. Marker size $\propto$ basin size; colour indicates barrier-to-global; global minima are red stars.

3 Landscape visualisations

Beyond global-hit probabilities, it is useful to visualise the *structure* of the loss landscape on the unique-hyperplane neighbour graph. Naively drawing the full neighbour graph produces a high-dimensional "hairball". Two compressed views are particularly interpretable in this ultrametric setting.

3.1 Disconnectivity tree (merge tree)

Figure 2 shows a *disconnectivity graph* (also called a merge tree) constructed by growing the sublevel set $\{ v : L(v) \leq \tau \}$ as τ increases: leaves are local minima (greedy sinks), and two branches merge at the loss level where their sublevel components first become connected. The tree encodes *barrier heights* separating basins. In this plot, leaf marker size is proportional to basin size under steepest descent; global minima are marked by red stars; and colour indicates the *barrier-to-global* (the minimised maximum loss along a path to a global basin, minus the sink loss).

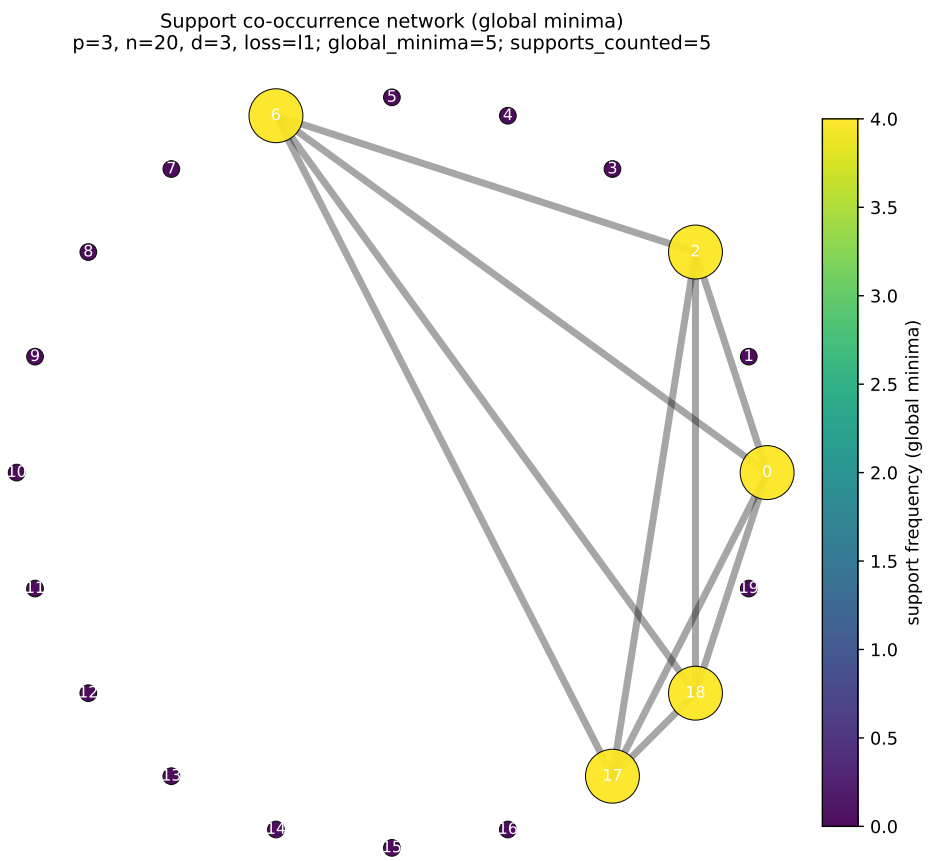

Figure 3: Support co-occurrence network for global minima in the same snapshot (seed 0). Nodes are data points; node colour/size reflect how often each point appears in a global-minimum support; edge width reflects pair co-occurrence count.

3.2 Support co-occurrence network (data points as nodes)

Ultrametric interpolation encourages overfitting by allowing the optimiser to "spend" degrees of freedom to make a small set of residuals exactly zero. A simple way to make this visible is to build a co-occurrence network on the data points (Figure 3): vertices are points $i \in \{1, \ldots, n\}$, and the edge weight between i and j is the number of times they co-occur in defining $(d{+}1)$-supports of *global* minima. Dense subgraphs highlight "kingmaker" cliques that repeatedly anchor optimal interpolating hyperplanes.

4 Implementation notes

All experiments are implemented as small, dependency-free Python programs:

- `code/padic_linear_regression.py` generates data, enumerates hyperplanes, builds the neighbour relation, and computes greedy basins.
- `code/make_experiment_report.py` aggregates CSVs and produces the plots included here.

5 Discussion

The neighbour-hyperplane graph provides a concrete finite combinatorial object attached to an ultrametric regression instance, and greedy descent on this graph is an appealing "first baseline" heuristic. These experiments suggest two general cautions:

1. **Local minima can be abundant.** Even when each node has only $O(nd)$ neighbours, the greedy-walk landscape can have many sinks.

2. **Overfitting is structurally favoured.** When the candidate class contains interpolating models, ultrametric losses may strongly prefer exact interpolation of a subset of points unless an explicit regulariser counterbalances it.