Learner's Notes: Non-Linear Optimization

Zehaan Naik

December 2025

1 Introduction

In statistics, optimization refers to the set of problems that involve computing the values of a group of parameters such that this selection ends up minimizing or maximizing some *objective function*. One such problem that is often hyped up by ML influencers on YouTube is linear regression. To frame it simply, linear regression tries to fit some given data as the following model:

$$Y = X\beta + \varepsilon, \tag{1}$$

where X is our data matrix (each row is an observation and each column is a feature), Y is the vector of responses we hope to explain, and ε is the usual "everything we don't know how to model" term. The star of the show is β, the coefficient vector we want to estimate.

If you have ever wondered what linear regression *actually* does beneath all the fancy scikit-learn wrappers, it simply asks:

> "Which value of β makes the predicted values $X\beta$ as close as possible to the actual observations Y?"

And, the way we formalize "as close as possible" is by solving an optimization problem that minimizes the sum of squared residuals. This is a smooth, convex, delightfully well-behaved problem that every optimization textbook loves because nothing goes wrong. I will not go into great depth on how this works (you can find nearly 50 zillion people doing the same everywhere online). Rather, what I am interested in here is breaking this simple, sweet structure and figuring out what more we can explain about the world using this simple idea of finding the "best" possible parameters that relate a function to some given data.

Before we wade any deeper into the swamp of non-linear optimization, it is worth clarifying a very common misconception: *a model is not called non-linear just because the data appear in some curvy, wiggly, or quadratic form.* Expressions like

$$Y = aX^2 + \varepsilon \qquad \text{or} \qquad Y = b\sin(X) + \varepsilon$$

might look non-linear at first glance (after all, there is a square and a sine floating around), but they are still linear *models* in the strict statistical sense. Why? Because the parameters a and b enter the model linearly. If you treat X^2 or $\sin(X)$ as just another "feature," then the model is no different from ordinary linear regression with a transformed design matrix.

So what actually counts as a *non-linear* model? Precisely those situations where the parameters themselves appear in a non-linear fashion. For example:

$$Y = \beta_1 e^{\beta_2 X} + \varepsilon, \qquad Y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} + \varepsilon, \qquad Y = \frac{\beta_1 X}{\beta_2 + X} + \varepsilon.$$

In each of these, the parameters $(\beta_0, \beta_1, \beta_2, \ldots)$ are tangled up inside exponentials, reciprocals, or other non-linear transformations. This is where the comfortable convex world of linear regression breaks down, and optimization becomes far more interesting (read: painful).

Understanding this distinction is crucial: *non-linearity is a property of how the parameters enter the model, not of how funky the data look*. Once this is clear, we can properly appreciate why non-linear optimization requires fundamentally different tools and why the tidy closed-form solutions of linear regression vanish the moment we step outside its linear-in-parameters cocoon.

Note. Please excuse the shorthand notation, such as $\sin(X)$ or $1/\beta_2 X$, as I know it does not make any sense for matrices. I use the notation here heuristically and will provide the details of the math later. (Go with the vibes for now!)

To close this introduction, let me briefly outline what this note will (and will not) try to do. The overarching goal is to peel back the layers of non-linear optimization and understand how statisticians and data scientists actually estimate parameters when the model refuses to behave nicely. We will start by looking at non-linear models themselves, before diving into the numerical machinery required to fit them. Along the way, we will also step outside the comfortable Gaussian world and examine what happens when the error distribution changes. This naturally leads us to estimators such as the Least Absolute Deviations (LAD) estimator, Poisson regression–style likelihoods, and other models where the objective function looks nothing like the familiar sum of squares.

A large portion of these notes will focus on the optimization algorithms that make all of this possible. We will discuss workhorses such as Newton-Raphson, Gauss-Newton, gradient-based methods, and a few related techniques that show up so often in practice that one eventually forgets they are numerical approximations. The goal is to understand not just how to implement these algorithms, but why they work, when they fail, and what kinds of theoretical guarantees we can (reasonably) expect from them.

Finally, while this is not intended to be a full course on statistical inference, I will include enough of the theoretical backbone to keep us intellectually honest. This includes basic consistency results, convergence heuristics, and some lightweight discussion of hypothesis testing in non-linear settings. Think of this section as the minimum amount of asymptotic theory one needs to avoid accidentally committing statistical crimes. In short, these notes are about understanding both the *what* and the *how* of non-linear optimization.

2 Non-Gaussian errors, LAD and count models

So far we have (implicitly) assumed the comforting Gaussian error model when we wrote

$$Y = X\beta + \varepsilon,$$

and then minimized the sum of squared residuals. One neat fact about that choice is that it *is* the maximum-likelihood estimator (MLE) when ε is Gaussian. But if the noise is not Gaussian, the MLE (and hence the natural objective function) changes — and with it the optimization problem we must solve. In this section we quickly connect a few common error models to the corresponding estimation problems you will actually optimize.

2.1 Laplace (double-exponential) errors and LAD

Assume the residuals ε_i are iid Laplace with density

$$f(\varepsilon_i) = \frac{1}{2b} \exp\big(- |\varepsilon_i|/b \big),$$

where $b > 0$ is a scale parameter. Under the model $Y_i = x_i^\top \beta + \varepsilon_i$, the log-likelihood (up to constants) is

$$\ell(\beta) \;=\; -\frac{1}{b}\sum_{i=1}^{n} |Y_i - x_i^\top \beta|.$$

Maximizing $\ell(\beta)$ (or equivalently minimizing the negative log-likelihood) gives the *least absolute deviations* (LAD) estimator

$$\hat{\beta}_{\mathrm{LAD}} \;=\; \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} |Y_i - x_i^\top \beta|. \tag{2}$$

Several useful remarks:

- LAD is robust to outliers in Y: the objective grows only linearly (not quadratically) with large residuals.
- The objective in (2) is convex but non-smooth. The subgradient condition is

  $$0 \in -\sum_{i=1}^{n} \operatorname{sign}(Y_i - x_i^\top \hat{\beta})\, x_i,$$

  where $\operatorname{sign}(t) \in [-1, 1]$ at $t = 0$.
- Computational approaches include linear programming formulations, specialized interior-point methods, or subgradient/proximal algorithms (Huber and Ronchetti, 2009; Koenker, 2005; Koenker and d'Orey, 1987; Barrodale and Roberts, 1974). One can also use *iteratively reweighted least squares* (IRLS) variants that approximate the L_1 loss by weighted least squares in each iteration.

2.2 Count data and the Poisson model

For non-negative integer responses, a standard choice is the Poisson model. The canonical GLM formulation is

$$Y_i \sim \mathrm{Poisson}(\mu_i), \qquad \mu_i = \exp(x_i^\top \beta),$$

so that the log-likelihood is

$$\ell(\beta) \;=\; \sum_{i=1}^{n} \big\{ Y_i x_i^\top \beta - \exp(x_i^\top \beta) - \log(Y_i!) \big\}.$$

Dropping constants and forming the negative log-likelihood (the objective to minimize) yields

$$L(\beta) \;=\; \sum_{i=1}^{n} \big\{ \exp(x_i^\top \beta) - Y_i x_i^\top \beta \big\}. \tag{3}$$

This objective is smooth and convex in β. Its gradient and Hessian are

$$\nabla L(\beta) \;=\; \sum_{i=1}^{n} \big\{ \exp(x_i^\top \beta) - Y_i \big\} x_i \qquad \text{and} \qquad \nabla^2 L(\beta) \;=\; \sum_{i=1}^{n} \exp(x_i^\top \beta)\, x_i x_i^\top,$$

so classical Newton–Raphson or Fisher scoring / IRLS methods are natural choices for optimization. Note the Hessian is positive semidefinite, which explains why convex optimization routines work well here.

2.3 Connecting loss functions and likelihoods

A useful mental rule is:

$$\textbf{Negative log-likelihood} \longleftrightarrow \textbf{Objective (loss) function}.$$

Thus, non-Gaussian error models generally produce different loss geometries:

- Gaussian errors $\Rightarrow$ quadratic loss $\sum(Y_i - x_i^\top \beta)^2$ (smooth, strongly convex if $X^\top X$ is full rank).
- Laplace errors $\Rightarrow$ L_1 loss $\sum |Y_i - x_i^\top \beta|$ (convex, non-smooth).
- Poisson model $\Rightarrow$ convex exponential-family loss $\sum\{\exp(x_i^\top \beta) - Y_i x_i^\top \beta\}$ (smooth).

2.4 Practical and theoretical notes

1. **Optimization differences.** Smooth convex problems (Gaussian, Poisson) admit fast second-order methods (Newton, quasi-Newton, IRLS). Non-smooth convex problems (LAD) require subgradient, proximal methods, or reformulation as linear programs.
2. **Statistical differences.** The MLE under the correct noise model is asymptotically normal under mild regularity conditions, but the variance formulas and efficiency depend on the actual error distribution. For robust methods like LAD, the influence of outliers is reduced compared to least squares.
3. **Modeling choices matter.** Transforming Y or using link functions (as in GLMs) is often preferable to naively applying least squares to count or binary data.
4. **Regularization.** In practice, we often add penalties (ridge, lasso, elastic-net) to these objectives. The optimization landscape then becomes a penalized M-estimation problem; many of the same algorithmic ideas (proximal operators, coordinate descent, IRLS+penalty) carry over.

In the following sections we will:

- derive and implement numerical algorithms (Newton–Raphson, Gauss–Newton, IRLS, subgradient and proximal methods) for these objectives,
- study basic consistency and asymptotic normality results for the corresponding estimators (under regularity conditions),
- and highlight practical pitfalls when the objective is non-smooth, non-convex, or poorly scaled.

References

Barrodale, I. and Roberts, F. D. K. (1974). Solution of an overdetermined system of equations in the l_1 norm. *Communications of the ACM*, 17(6):319–320. Algorithm 478.

Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics*. Wiley.

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press, Cambridge.

Koenker, R. W. and d'Orey, V. (1987). Computing regression quantiles. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(3):383–393.