Linear and Quadratic Models
least square fitting
and the subjects that interest me a lot in this course

Salar Ghaderi

Abstract

This report presents a comprehensive analysis of the weighted least-squares fitting applied to a data set extracted from *Instrumentation and Methods for Astrophysics* [1]. The study involves three key exercises: fitting a linear model to a subset of data, examining the effects of incorporating high-uncertainty data points on the model's reliability, and extending the method to fit a quadratic model. We provide detailed derivations, discuss the implications of including data with large uncertainties, and analyze the paradoxical reduction in slope variance despite the addition of less reliable data.

The Python codes used in this report can be found at the following link: GitHub Repository.

Introduction

least-squares fitting is a statistical method used when data points have varying uncertainties. The technique minimizes the weighted sum of squared residuals, ensuring that points with smaller uncertainties exert greater influence on the fit. For a linear model $y = mx + b$, the parameters m and b are obtained by solving:

$$\min_{m,b} \sum_{i=1}^{N} \frac{\left[y_i - (mx_i + b)\right]^2}{\sigma_{y_i}^2}.$$

In matrix form, this is expressed as:

$$\chi^2 = (\mathbf{Y} - A\mathbf{X})^\mathsf{T} C^{-1} (\mathbf{Y} - A\mathbf{X}),$$

where $\mathbf{X}$ contains the model parameters, A is the design matrix, $\mathbf{Y}$ is the vector of observations, and C is the covariance matrix of the measurements.

This report follows the methodology outlined in [1], using data from Table 1 of the referenced article.

Exercise 1: Linear Fit for ID=5..20

Setup and Derivation

We consider the linear model:

$$y = mx + b,$$

with parameters $\mathbf{X} = (b, m)$. The design matrix A and covariance matrix C are defined as:

$$A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \quad C = \text{diag}(\sigma_{y_1}^2, \ldots, \sigma_{y_N}^2).$$

The weighted least-squares solution is given by:

$$\mathbf{X} = (A^\mathsf{T} C^{-1} A)^{-1} A^\mathsf{T} C^{-1} \mathbf{Y}.$$

We apply this to data points ID=5..20.

0.1 Result and Figure

The solution yielded:

$$b \approx 34.05 \pm 18.25, \quad m \approx 2.24 \pm 0.11, \quad \text{var}(m) = 0.01.$$

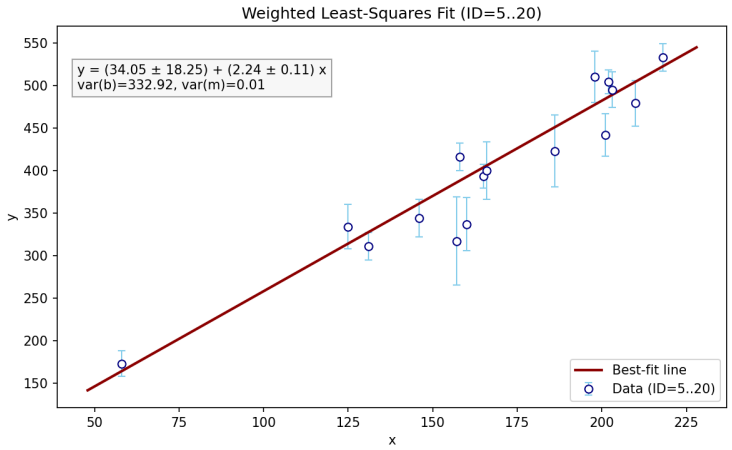

Figure 1: Linear fit for ID=5..20, showing data with error bars and the best-fit line.

Exercise 2: Linear Fit with ID=1..20 and the Paradox

Motivation and Setup

In this exercise, we include all 20 data points (ID=1..20). Some of the initial points (ID=1..4) have significantly larger uncertainties σ_y. Intuitively, one might expect that adding high-uncertainty points would degrade the precision of the slope m. However, the results indicate a paradoxical decrease in the slope variance.

Result and Discussion

The fit results after including all points are:

$$b \approx 213.27 \pm 14.39, \quad m \approx 1.08 \pm 0.08, \quad \text{var}(m) \approx 0.0060.$$

Despite adding points with larger uncertainties, σ_m^2 decreased from 0.01 to 0.0060, suggesting an increase in confidence. This appears contradictory.

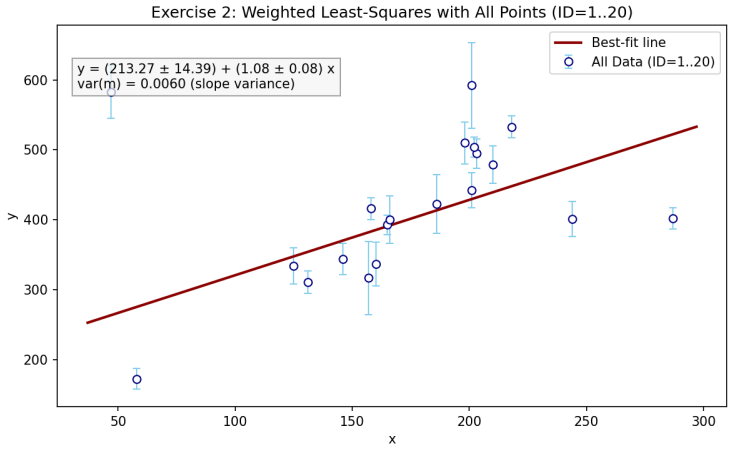

Figure 2: Linear fit using all 20 points. The slope's variance decreased, which seems contradictory given the high-uncertainty data added.

Mathematical Explanation of the Paradox

The decrease in σ_m^2 can be attributed to the mathematical structure of the weighted least-squares solution:

$$\mathrm{Cov}(\mathbf{X}) = (A^\mathsf{T} C^{-1} A)^{-1}.$$

Adding more data points increases the size of A, which generally improves the determination of m. Even if the new points have large σ_y, their inclusion increases the number of constraints, leading to a formal reduction in σ_m^2. This suggests a false sense of confidence in m, especially if the added data are from a different regime or have unaccounted systematic errors.

Exercise 3: Quadratic Fit for ID=5..20

Generalization to a Second-Order Polynomial

We extend the model to a quadratic form:

$$g(x) = b + mx + qx^2.$$

The parameter vector is $\mathbf{X} = (b, m, q)$, and the design matrix A becomes:

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{pmatrix}.$$

The solution and covariance are obtained via:

$$\mathbf{X} = (A^\mathsf{T} C^{-1} A)^{-1} A^\mathsf{T} C^{-1} \mathbf{Y}, \quad \mathrm{Cov}(\mathbf{X}) = (A^\mathsf{T} C^{-1} A)^{-1}.$$

Results

For ID=5..20, the quadratic fit yields:

$b \approx 72.89 \pm 38.91$ (var $= 1514.11$), $m \approx 1.60 \pm 0.58$ (var $= 0.34$), $q \approx 0.0023 \pm 0.0020$ (var ≈ 0.0000).

The quadratic coefficient q is small and consistent with zero, suggesting the data may not strongly support a quadratic term.

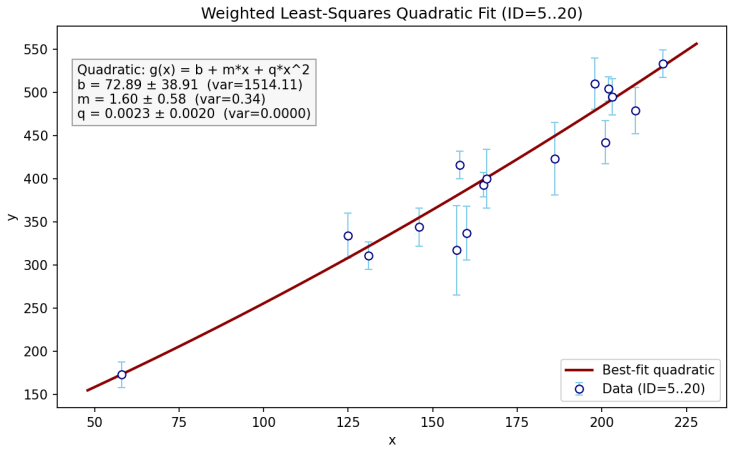

Figure 3: Quadratic fit $g(x) = b + mx + qx^2$ for ID=5..20. The quadratic term is small and not strongly constrained.

Brief Notes on the Code

The implementation extends the linear model by adding a column for x^2:

```
A = np.column_stack((np.ones(N), x_vals, x_vals**2))

tmp = A.T @ C_inv @ A
cov_params = np.linalg.inv(tmp)
X_best = cov_params @ (A.T @ C_inv @ y_vals)

b_best, m_best, q_best = X_best
var_b, var_m, var_q = np.diag(cov_params)
```

The variances $\sigma_b^2, \sigma_m^2, \sigma_q^2$ are extracted from the diagonal of the covariance matrix.

Some words on the subjects covered in this course that particularly interest me (currently i am not engaged in research (I am a first-year student))

Today's AI is incredibly good at spotting patterns in large datasets. Deep learning and statistical models have achieved amazing results in image recognition, language processing, and prediction. But when it comes to generating real, innovative scientific ideas—ideas that break new ground and explain why things happen—the current statistical approach starts to show its limits. This essay explores some of the main topics and techniques that can help us move beyond the current statistical machine paradigm and toward an AI that can truly generate science.

The Statistical Machine Paradigm: What It Is and Where It Falls Short

What Is the Statistical Machine Paradigm?

Most of our AI systems today are built on the idea of learning from data. They are designed to find patterns, optimize a cost function (often using methods like stochastic gradient descent), and make predictions. This includes everything from simple linear regressions to complex deep neural networks. Techniques such as kernel regression, Gaussian processes, and expectation-maximization are all part of this toolkit. Whether we take a frequentist approach (focusing on long-run frequencies and hypothesis testing) or a Bayesian approach (using prior information to update our beliefs), these methods essentially boil down to crunching numbers and fitting models to data.

Where Does It Fall Short?

- **Data-Fitting vs. Real Understanding:** Although these models excel at interpolation—fitting data they have seen—they often lack true understanding. They miss the deeper, causal relationships that explain why things work the way they do. This is a major hurdle if we want AI to generate novel scientific theories rather than simply repeat established patterns.
- **Uncertainty and Overconfidence:** Techniques such as Markov chain Monte Carlo (MCMC), bootstrap, and jackknife help estimate uncertainty, but many statistical models still tend to be overconfident. They may not fully capture the uncertainties inherent in complex scientific systems.
- **The Black Box Issue:** Many models function as "black boxes." They provide answers without offering much insight into the reasoning behind them. In science, understanding the process behind a result is as important as the result itself.
- **Causation vs. Correlation:** Statistical models are excellent at identifying correlations, but scientific insight demands an understanding of causation. Without grasping cause and effect, an AI might generate ideas that sound plausible but lack genuine explanatory power.

Main Topics and Techniques to Enhance a Science-Generating AI

To move toward an AI that can generate new science, we need to explore several advanced topics and techniques that address the shortcomings of the current statistical approach.

Uncertainty and Estimation Methods

- **Frequentist vs. Bayesian Estimation:** Combining both approaches can improve uncertainty quantification. Bayesian methods allow the integration of prior knowledge and dynamic updating of beliefs as new data arrives, while frequentist methods provide robust hypothesis tests and confidence intervals.
- **Nonlinear Least Squares, Profiling, and Marginalization:** These techniques assist in fitting complex models where not all parameters are of direct interest (i.e., handling nuisance parameters). By focusing on key components and integrating out the rest, more robust models can be built.
- **Posterior Quantities and MCMC:** Techniques like MCMC and reparameterization allow exploration of the full range of model outcomes, ensuring the AI properly understands the uncertainty in its predictions.
- **Bootstrap and Jackknife:** Resampling methods offer insights into the precision of our estimates, which is crucial for verifying that the AI's scientific ideas are not mere statistical flukes.

Optimization, Signal Processing, and Model Evaluation

- **Signal-to-Noise, Cost, and Optimization:** Balancing signal and noise is essential. Advanced optimization techniques ensure that the AI captures meaningful patterns without overfitting the data.
- **Visualization and Residual Analysis:** Clear visualization tools and residual analysis help assess model performance. By examining what remains unexplained, we can identify missing factors vital for scientific discovery.
- **Decision Theory and Model Selection:** Applying information criteria (such as AIC or BIC) and cross-validation allows the AI to compare models and select the best one, akin to how scientists choose theories based on their predictive power and simplicity.

Advanced Regression, Filtering, and Hierarchical Models

- **Big Linear Models, Interpolation, and Double Descent:** These concepts help manage the trade-off between model complexity and performance. Sometimes, adding complexity (up to a point) can improve predictions, a phenomenon known as double descent.
- **Filters, Kernel Regression, and Gaussian Processes:** Non-parametric methods like these enable the AI to capture nonlinear relationships with built-in uncertainty estimates. They provide a mathematically robust approach to modeling complex data.
- **Hierarchical and Latent-Variable Models:** Many scientific problems have layers of complexity. Hierarchical models and methods like principal components analysis (PCA) help uncover hidden structures and reduce noise in the data.

Causal Inference and Beyond-Statistical Methods

- **Causal Structure and Causal Inference:** Moving beyond correlations, causal inference techniques help the AI understand what causes what. This is key for generating insights that are truly explanatory rather than merely descriptive.
- **Simulation-Based and Likelihood-Free Inference:** When traditional likelihoods are too difficult to compute, simulation-based approaches allow the AI to compare simulated data with real observations, opening the door to studying complex systems where direct calculations fall short.

Integrating Deep Learning with Rigorous Statistics

- **Neural Networks and Deep Learning:** While deep learning is excellent at pattern recognition, integrating it with robust statistical methods (e.g., Bayesian neural networks) can improve uncertainty quantification. Autoencoders with variational inference, for example, help capture latent structures in the data.
- **Neuro-Symbolic Integration:** Blending symbolic reasoning (logic, rules) with neural networks could enable higher-level reasoning, filling a critical gap in current statistical models.

The Importance of Good Data Management

- **Data Handling and Preprocessing:** Effective data management is vital. Using databases and SQL, handling missing data via inpainting, and employing sufficient statistics are all key to maintaining a clean data pipeline.
- **Posterior Predictive Checks and Noise Resampling:** These methods help ensure that the AI's models not only fit the data well but are also robust and realistic in their predictions.

References

[1] David W. Hogg, Jo Bovy, and Dustin Lang. *Data Analysis Recipes: Fitting a Model to Data.* Instrumentation and Methods for Astrophysics (astro-ph.IM); Data Analysis, Statistics and Probability (physics.data-an), 2010. https://doi.org/10.48550/arXiv.1008.4686