ML CASE STUDY

Presentation by Team 5

Made By-
Suhani Singh, IIT Roorkie
Yatika Jena, IIT Guwahati

PART 1

Understanding the error surface

Approach Taken

- Generated synthetic data of height vs. weight relationship and added Gaussian noise.
- Task: Estimate the relationship between height and weight using a linear model
- Used a line of the form $y = w_0 + w_1x$ to fit noisy data
- Main focus: Estimate the best values for w_0 and w_1 that minimize error

Understanding the error surface

Modeling Techniques

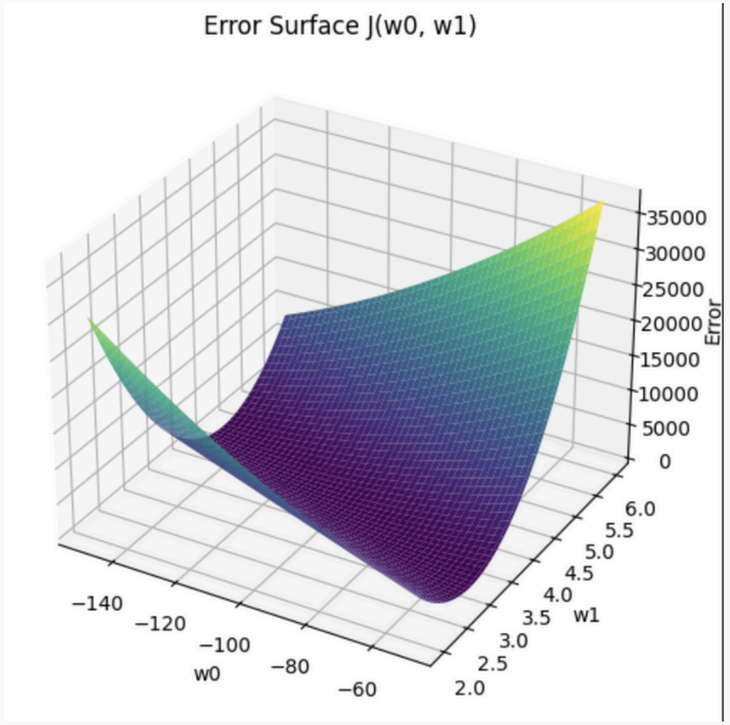
Used two methods:

1. Error Surface Method

→ Computed error values (MSE) across a grid of w_0 and w_1 values

→ Visualized the error as a 3D surface

→ Minimum point gives the best empirical estimate of weights



2. Least Squares Method

→ Applied the closed-form formula

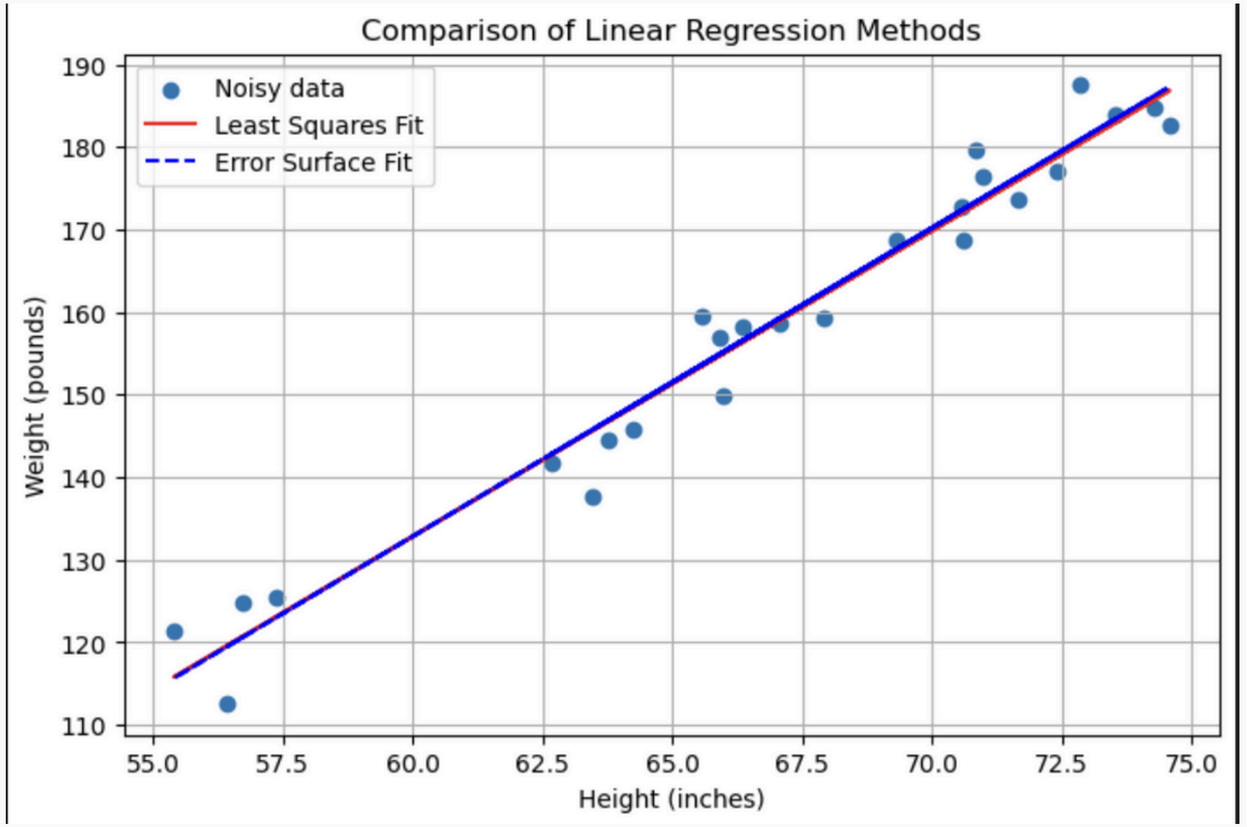
→ Directly calculated optimal w_0 and w_1

$$w_{opt} = (X^T X)^{-1} X^T t$$

Understanding the error surface

Conclusion

- Best-fit lines from both approaches almost overlapped on the plot
- Closed-form least squares is recommended for efficiency and accuracy.
- Visual exploration (error surface) is useful for understanding model behavior.



PART 2

Understanding model order and overfitting

Approach Taken

- Started with N = 20 points, random train/test split (10 each).
- Evaluated performance across polynomial degrees M = 0–9 using RMSE.
- Introduced Ridge regularization ($\lambda \in \{0, 1e\text{-}7, 1e\text{-}4, 1e\text{-}2, 1, 10\}$).
- Later scaled up to N = 100 training points to test data sufficiency.
- Added experiments varying bias-term regularization.

Understanding model order and overfitting

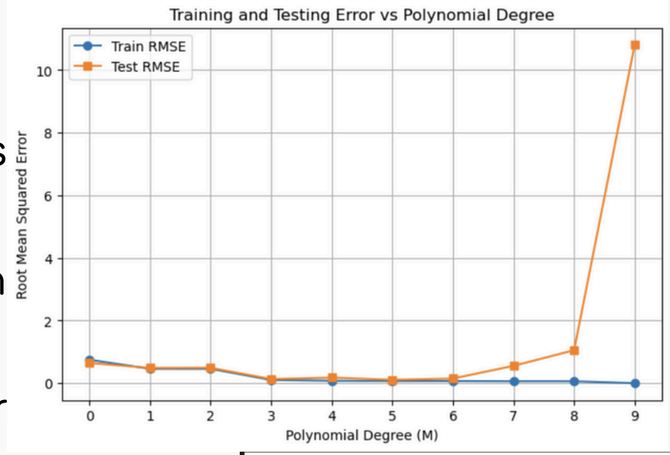
Modeling Techniques

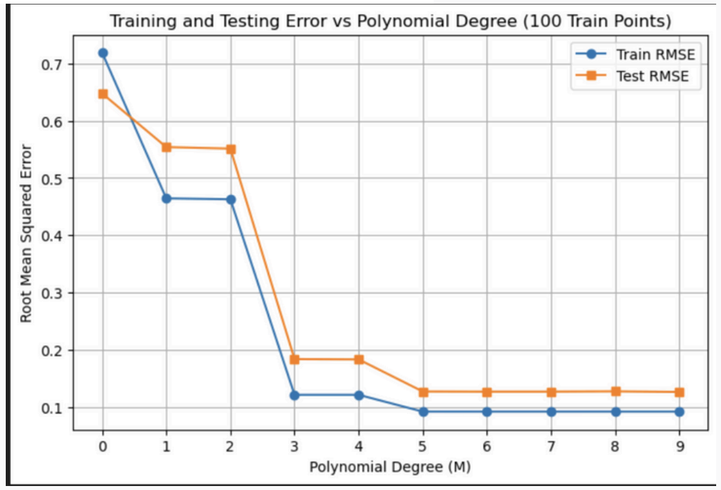
- Polynomial Expansion: Input data was transformed using polynomial basis functions up to degree M
- Least Squares Estimation: We used the closed-form solution of the least squares method to estimate model parameters, which minimizes the residual sum of squares.
- Regularization (Ridge Regression): A penalty term (λ) was added to control overfitting by shrinking weights, especially in high-degree models.
- Bias-Term Regularization Toggle: We introduced a flag to optionally exclude the bias (intercept) term from regularization, allowing us to study its influence separately.

Understanding model order and overfitting

Key Findings/Results

- Low-degree models underfit; high-degree models overfit (low train error, high test error).
- Increasing training data improves generalization and flattens error curves.
- Regularization reduces test error, especially for large M.
- Excluding bias term from regularization improved performance when data had constant offset





PART 3

Understanding the choice of kernel

Approach Taken

- Started with synthetic data resembling a) sinusoidal and b) piecewise functions with added noise.
- attempted kernel regression using:
  1.Polynomial basis
  2.Gaussian Kernel
  3.Sigmoid Kernel
- Evaluated model order M=1 to 10 to study underfitting vs overfitting.
- Used RMSE to compare training and test errors for different kernels and targets.

Understanding the choice of kernel

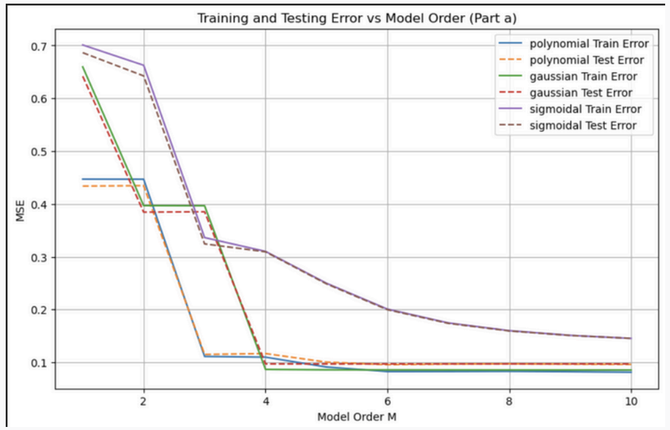
Modeling Techniques

- Used Kernelized Ridge Regression with three kernel types to model noisy synthetic data.
- Polynomial Basis: Global monomial expansion up to degree M; prone to overfitting for complex or non-smooth targets.
- Gaussian Kernel: Localized features using RBFs; best suited for capturing fine patterns in both sinusoidal and piecewise data.
- Sigmoid Kernel: $\tanh(\alpha xz+c)$; performance varied, sensitive to α and c.
- Regularization applied to ensure numerical stability and prevent overfitting.

Understanding the choice of kernel

Findings & Conclusion

- **Sinusoidal Target:**
  - Polynomial kernel achieved lowest test error at moderate M; ideal for smooth, global trends.
  - Gaussian kernel underperformed due to its localized nature.
  - Sigmoid kernel gave mixed results.
- **Piecewise Target:**
  - Gaussian kernel excelled by adapting to sharp transitions and local variations.
  - Polynomial basis struggled with discontinuities.
  - Sigmoid kernel showed moderate success but lacked consistency across M.
- **General Insights:**
  - Optimal model complexity lies around M=5-7; higher M leads to overfitting across kernels.
  - Kernel regression offers flexibility, but effectiveness is highly task dependent.



PART 4

Understanding training parameters

Approach Taken-

- Explored SGD as an alternative to closed-form solution.
- Systematically varied hyperparameters (step size & batch size).
- Compared models using MSE and weight differences.

Modeling Techniques-

- Kernel Regression with:
  1. Polynomial kernel
  2. Gaussian kernel
  3. Sigmoidal kernel
- Used SGD as iterative optimizer.
- Closed-form used for baseline comparison.

```
    # Compute prediction and error
    y_pred = np.dot(phi_i, weights)
    error = y_pred - y_i

    # Compute gradient: ∇w = 2 * error * phi_i
    gradient = 2 * error * phi_i
    gradient_sum += gradient

# Update weights using average gradient
avg_gradient = gradient_sum / batch_size_actual
weights -= stepSize * avg_gradient
```

Key Findings/Results-

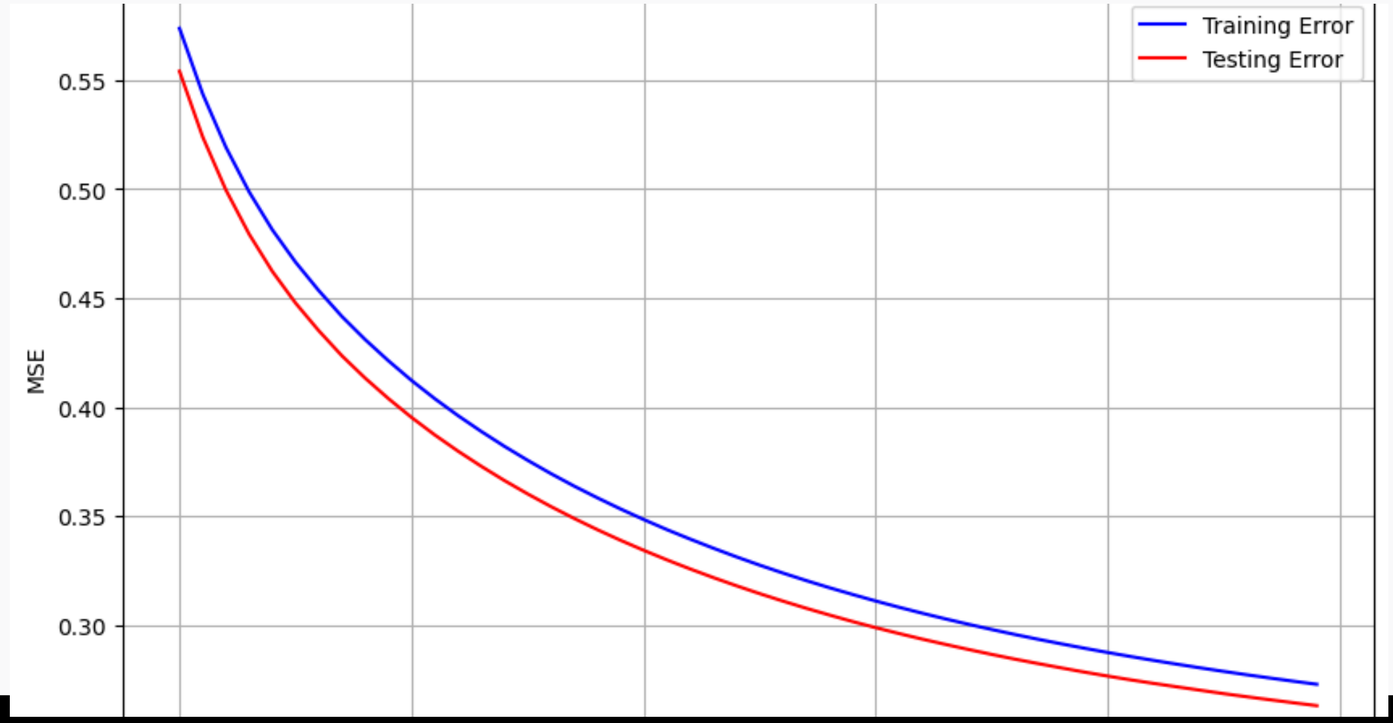
- Best performance for η = 0.01 and mini-batch sizes.
- SGD approximates closed-form weights well if tuned properly.
- Kernel choice affects performance; polynomial kernel showed good results with low M.

Challenges Faced-

- Selecting right η — too small or too large can ruin learning.
- Ensuring stability across different kernels and batch sizes.
- Handling noisy updates in pure SGD.

Conclusion

- Proper tuning of step size and batch size is critical.
- SGD is flexible and scalable but requires careful parameter control.
- Closed-form solutions are faster for small datasets, but SGD generalizes better for larger, online settings.



PART 5

Understanding bias-variance trade-off

Approach Taken-

- We aimed to study how bias and variance behave under different levels of regularization in a high-capacity model.
- Step 1: Generated 100 datasets of noisy sinusoidal data (N = 25).
- Step 2: Used a 25th-order linear regression model (24 Gaussian basis functions + 1 bias term).
- Step 3: Applied regularized least squares (ridge regression) to fit the model.
- Tried simpler polynomial models initially, but they couldn't capture localized sinusoidal patterns well.
- Gaussian basis functions offered better local representation and flexibility.

Modeling Techniques-

- Model Used: Linear Regression with Gaussian Basis Functions
- Design Matrix: Each feature vector consisted of:
- 24 Gaussian basis functions centered between 0 and 1
- 1 constant bias term
- Regularization Technique: Ridge Regression
- Tested three different regularization coefficients:
- Low $\lambda = 1e\text{-}5$
- Medium $\lambda = 1$
- High $\lambda = 1000$

Key Findings / Results

1.Low λ (1e-5)

- 100 Estimated Curves: High variability; most curves overfit the noise
- Mean Prediction vs True Function: Mean aligns well with true sine curve
- Conclusion:
- Low Bias, High Variance

2.Medium λ (1)

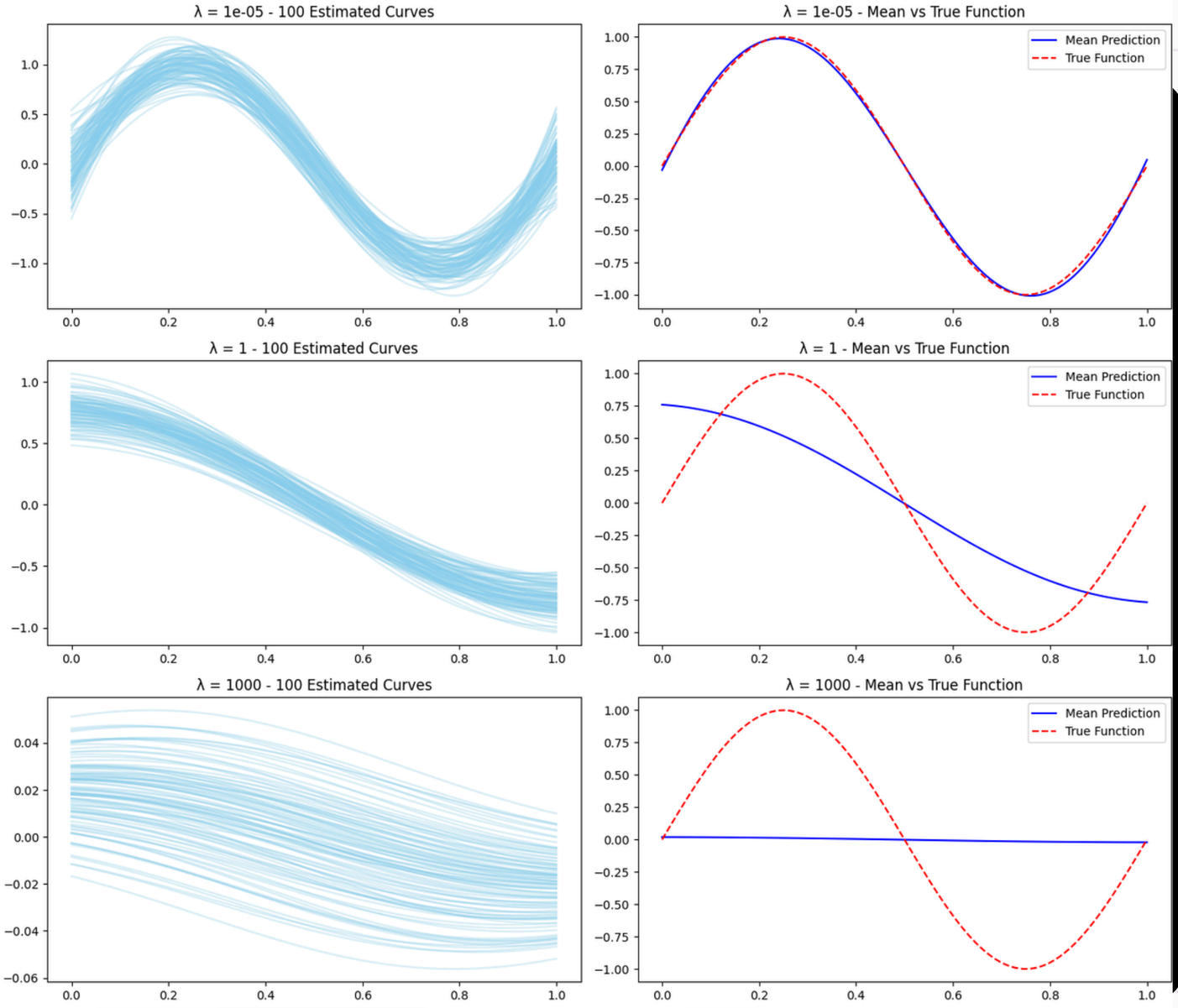
- 100 Estimated Curves: Less variation, smoother curves
- Mean Prediction: Close to true function with slight smoothing
- Conclusion:
- Balanced Bias-Variance; Good Generalization

3. High λ (1000)

- 100 Estimated Curves: Almost identical, flat predictions
- Mean Prediction: Fails to match true sine function
- Conclusion:
    - High Bias, Low Variance

Key Insight:

- Low regularization overfits (low bias, high variance)
- High regularization underfits (high bias, low variance)
- Moderate regularization balances both
- Recommendation: Always tune regularization hyperparameters carefully for optimal generalization.



PART 6

Understanding bias-variance trade-off

Approach Taken-

- We aimed to explore MAP estimation by performing Bayesian sequential updates on the weights of a 20th-order model.
- Started with a standard normal prior on parameters.
- Performed Bayesian updates with each new data point to get the posterior distribution.
- Compared sampled curve fits from the posterior against the true sinusoidal function.
- Also analyzed how predictive uncertainty evolves with increasing training data (10 to 100 points).
- Initially tried using fixed design matrices without sequential updates, but that approach failed to model growing confidence with more data — hence MAP + Bayesian updates worked best.

Modeling Techniques-

- Model Type: Linear regression with 20 Gaussian basis functions and a bias term.
- Prior Assumption:
- Weight prior: N(0,α−1*I), where α=1.0
- Likelihood Assumption:
- Noise variance: σ^2=0.01 and β=100
- Bayesian Learning :
  Performed sequential updates to posterior.

- Predictive Distribution:
-

Mean: $\phi(x_*)^T m_N$

Variance: $\phi(x_*)^T S_N \phi(x_*)$

MAP estimation is based on Bayes' theorem, which states:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

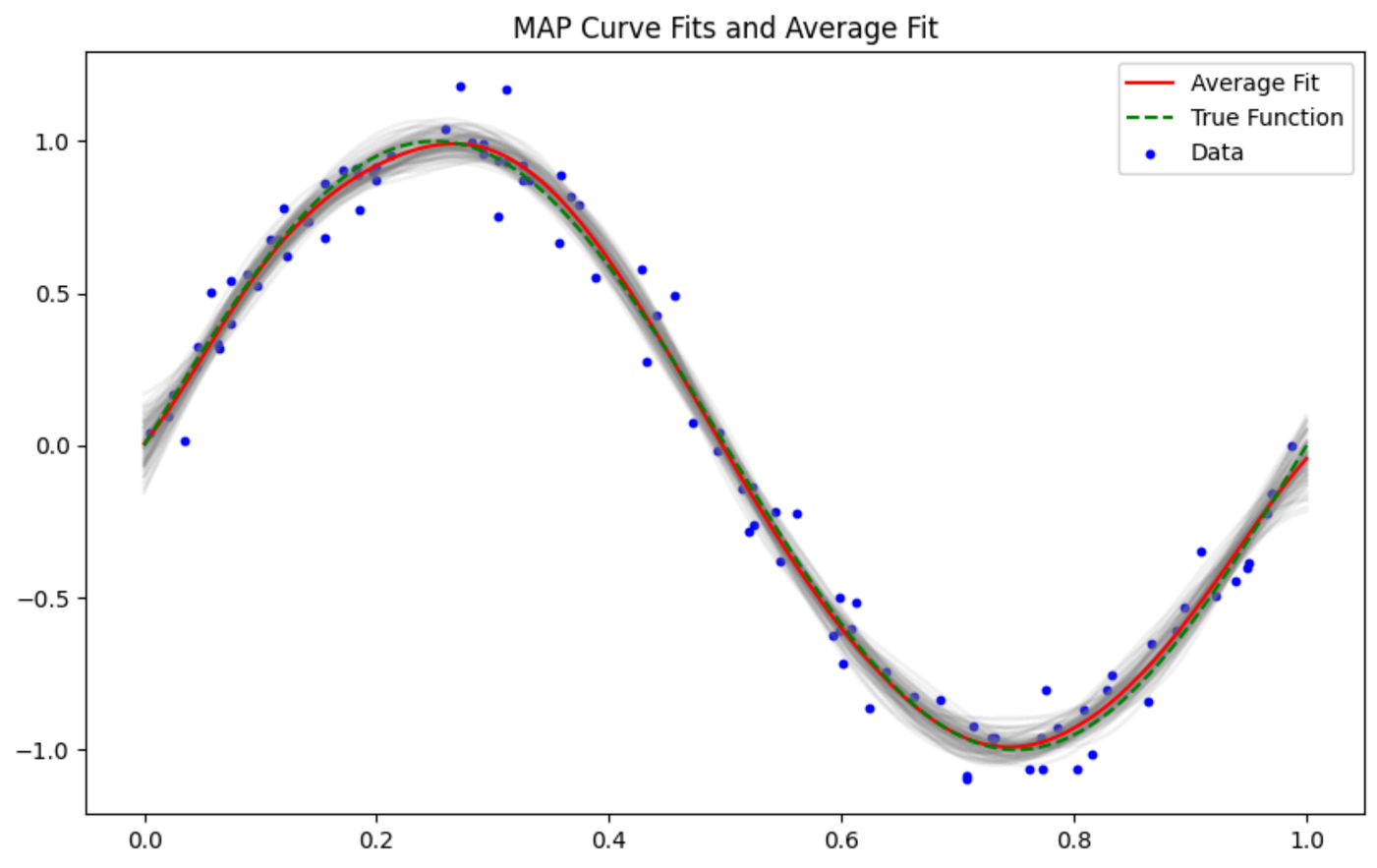
where:

- $P(\theta|D)$ is the posterior probability.
- $P(D|\theta)$ is the likelihood.
- $P(\theta)$ is the prior probability.
- $P(D)$ is the marginal likelihood (evidence).

$$\mathbf{S}_N^{-1} = \alpha I + \beta \sum \phi_n \phi_n^T, \quad \mathbf{m}_N = \mathbf{S}_N \left(\beta \sum t_n \phi_n \right)$$

Key Findings / Results-

- MAP Curve Fitting
- Sampled 100 weight vectors from the final posterior distribution.
- Generated predicted curves and averaged them.
- Observation:
- Individual sampled fits vary due to posterior uncertainty.
- The average fit closely matches the true sine function.
- Demonstrates strong regularization and generalization.



- As more data is used:
- Posterior becomes sharper.
- Predictive uncertainty decreases.
- Mean prediction becomes more confident.

Challenges Faced-

- Numerical Stability: Inverting large matrices required care; ensured α and β were scaled appropriately.
- Posterior Sampling: Needed enough samples to see reliable curve behavior.
- Computational Time: Repeated matrix operations for each data subset were intensive but necessary for visualizing learning dynamics.

Conclusion-

MAP estimation enables robust modeling under uncertainty by incorporating prior beliefs and updating them with observed data.

Key Insight:

Early models are uncertain and flexible.

As data grows, predictions become accurate and confident.

Predictive distributions offer a principled way to estimate not just predictions, but their confidence.

Recommendation: Use Bayesian methods like MAP when data is noisy or limited — they provide both predictions and uncertainty quantification.

THANK YOU

For your attention.