(CIS 581)

Computational Learning

Project 1

Student Name:

- Muhammad Mustafa

Introduction:

In this project, I applied polynomial curve-fitting for regression learning to model U.S. COVID-19 cases based on weekly data. The performance of different polynomial models was evaluated using root-mean-squared error (RMSE). To ensure optimal model selection, I employed 12-fold cross-validation (CV) to determine the best polynomial degree (d*) for fitting the training data.

During CV, I trained and tested polynomial models of degrees 0 to 28, computing RMSE for each hypothesis class on each fold. I then recorded and averaged the RMSE values across all folds, selecting the polynomial degree that minimized the CV test RMSE.

After identifying d*, I further optimized a 28-degree polynomial by tuning the regularization parameter (λ*) using CV. Finally, I trained the best-selected models on all available training data and evaluated their performance on a separate test set. The results include RMSE values, learned coefficient weights, and polynomial curves visualizing the model's fit to the data.

Discussion and Observations

After conducting the polynomial regression analysis on the COVID-19 dataset using 12-fold cross-validation, I arrived at the following key findings:

Selection of Optimal Polynomial Degree:

- The cross-validation results showed that the optimal polynomial degree for fitting the data without regularization was $d = 17$.
- Lower degree polynomial ($d \leq 5$) exhibited underfitting, as they failed to capture the complex trends in the dataset.

- Higher-degree polynomials ($d > 17$) led to overfitting, where the model performed well on training data but had a much higher error on the test data.

Selection of Optimal Regularization Parameter:

- When training a 28-degree polynomial, I applied different values of the regularization parameter (λ) to prevent overfitting.
- The best regularization parameter was found to be $\lambda^* = 0.000$, meaning that no regularization was needed for this dataset. This suggests that the complexity of the model was manageable without additional penalization of the coefficient magnitudes.
- Zero regularization ($\lambda = 0$) led to overfitting, especially with high-degree polynomials ($d=28$).
- Excessively large λ values underfit the data by forcing the coefficients to be too small and limiting the model's ability to capture the underlying trend.

The performance comparison of the two final models:

Model	Training RMSE	Test RMSE
Polynomial ($d^*=17$, $\lambda=0$)	0.1430	0.2747
Regularized ($d= 28$, $\lambda^* = 0.000$)	0.0882	0.2992

Conclusion:

- The best model for predicting COVID-19 cases was the polynomial regression model with $d^*=17$. This degree provided a good balance, effectively capturing the overall trend in the data without overfitting or underfitting.
- When using a 28-degree polynomial, no regularization ($\lambda = 0$) was required to achieve optimal performance. A regularization parameter of $\lambda = 0.000$ resulted in a model that performed well on the training data but did not generalize effectively to the test set.
- The polynomial model with $d^*=17$ had a relatively low test RMSE (0.2747), indicating a better ability to generalize than the overfitted regularized model.
- The $d^*=17$ polynomial fit the overall trend in COVID-19 case data accurately, without excessive fluctuations but the regularized $d = 28$ model produced a smoother curve, which reduced unnecessary fluctuations but didn't capture important details.

References:

Scikit-learn developers. (n.d.). sklearn.preprocessing.StandardScaler. Scikit-learn 0.24 documentation. Retrieved from https://scikit-learn.org/0.24/modules/generated/sklearn.preprocessing.StandardScaler.html

Scikit-learn developers. (n.d.). sklearn.linear_model.Ridge. Scikit-learn 0.24 documentation. Retrieved from https://scikit-learn.org/0.24/modules/generated/sklearn.linear_model.Ridge.html

Scikit-learn developers. (n.d.). Cross-validation on diabetes dataset exercise. Scikit-learn 0.24 documentation. Retrieved from https://scikit-learn.org/0.24/auto_examples/exercises/plot_cv_diabetes.html

NumPy developers. (n.d.). NumPy 2.2 Reference Documentation. Retrieved from https://numpy.org/doc/2.2/reference/index.html

Pandas developers. (n.d.). I/O API Reference — Pandas Documentation. Retrieved from https://pandas.pydata.org/docs/reference/io.html

Muhammad Mustafa. (n.d.). Project-1 [GitHub repository]. Retrieved from https://github.com/muhammadmustafa17/Project-1