CDA Spring 2026

Sneha Das

February 4, 2026

1 Week 1: Learning from targets (EPE, Bias, Variance, Model complexity)

The goal of this first set of exercises is to bridge the gap between theoretical definitions and empirical observations. We will explore why learning from finite data is fundamentally difficult, focusing on the concepts of instability, unobservable targets, and the bias-variance lens. We will combine controlled toy datasets with real-world data to study how bias, variance, and instability manifest in practice. Exercises 1.1, 1.2 use a simulated data, allowing us to calculate the actual Bias and Variance components of EPE. Exercise 1.3 uses real-world data (UCI wine dataset).

1.1 OLS and the 'Wobble' of Instability

Why can a theoretically 'unbiased' model like OLS fail so spectacularly on finite data?

1. **(Instability):** Use a simulation to demonstrate how small perturbations in training data lead to large changes in estimated coefficients ($\hat{\beta}$).
2. **(Identifiability):** Explore how feature correlation (collinearity) affects the variance of your estimates. Why does the matrix $(X^TX)^{-1}$ become problematic? You can also try different ρ values to further your understanding.
3. **(Unbiasedness vs. Error):** Discuss why a mean bias of zero does not guarantee a good prediction for a specific dataset. In what scenario does an 'unbiased' model have a high Expected Prediction Error (EPE)?

Files:

`ex_1_1_Q.py & ex1_1_sol.py`

1.2 Ridge Regression and Controlled Bias

Can we improve our predictions by deliberately 'breaking' our model? Use MSE (mean squared error) as the metric of performance.

- Derive (using pen and paper) the ridge regression solution by, as you would when minimizing any differentiable analytical function, differentiating $\| y - X\beta\| + \lambda\| \beta\|_2$ with respect to β, setting to zero and solving for β.
- (Optional) consider implementing the derived ridge regression solution from above for the following steps, instead of using the sklearn ridge package.
- **(Bias-Variance Tradeoff):** Quantify the relationship between the regularization parameter λ and the model's error components. As $\lambda \to \infty$, what happens to the variance and the squared bias?
- **(Regularization Path):** Observe how coefficients shrink as λ increases using a plot. Do all coefficients shrink at the same rate when features are correlated?
- **(Optimization):** Identify the 'sweet spot' in model complexity that minimizes total Expected Prediction Error (EPE).

Files:

`ex_1_2_Q.py & ex1_2_sol.py`

1.3 The M vs N (Wine Quality)

Overarching Question: How does the relationship between parameters (m) and data volume (n) dictate the 'regime' of learning?

- *In the code files* `ex_1_4_1_OLS_Q.py` *and* `ex_1_4_1_Q.py`*, run the OLS baseline and the Ridge complexity sweep using the original 11 features.*
- **(The Degrees of Freedom Paradox):** In the baseline test on Wine Quality ($n \approx 1600, m = 11$), the model did not show a clear U-shape error curve with respect to complexity. Why does the ratio m/n prevent overfitting in this 'data-rich' regime?
- *In the code files* `ex_1_4_2_OLS_Q.py` *and* `ex_1_4_2_Q.py`*, use* `PolynomialFeatures` *to expand the feature space and reduce the training set size.*
- **(Induced Overfitting):** By using *Polynomial Features* (degree 2), you increased m from 11 to 77. By reducing the training set to $n \approx 320$, the ratio m/n shifted significantly. Describe the behavior of the training and test error in this 'parameter-heavy' regime.
- **(The Relativity of Complexity):** Is a model with 100 parameters 'too complex' if you have 1,000,000 training samples? Discuss why complexity is a property of the *system* (model + data), not just the model.