CS129: Applied Machine Learning
Discussion Section 5

1 Adding a Predictor in Logistic Regression

Suppose we have a classification problem with a binary response Y and a p-dimensional predictor vector $X = (X_1, \ldots, X_p)$. Logistic regression is fit to a set of n samples. Then, logistic regression is fit again to the same observations, but now we include one additional predictor, so that

$$X = (X_1, \ldots, X_p, X_{p+1}).$$

Explain how the **training error**, **test error**, and **coefficients** change in each of the following cases:

(a) $X_{p+1} = X_1 + 2X_p$.

Since the new predictor is exactly collinear with 2 of the old predictors, the coefficients β_1, β_p, and β_{p+1} are unidentifiable, as logistic regression maximizes a likelihood which only depends on a linear combination of the predictors. The prediction remains unchanged, and therefore so do the training and test errors.

(b) X_{p+1} is a random variable independent of Y.

Since the number of samples is finite, logistic regression may assign a positive coefficient to X_{p+1} even though it is independent of the response; this will likely affect other coefficients as well. The training error can only decrease, whereas the test error will increase because the bias remains the same while variance increases.

2 Multi-Task Learning vs. Separate Models

You are given a dataset of patients who visited UPMC Hospital in 2011. For each patient, a set of features (e.g., temperature, height, lab results) has been extracted.

Your goal is to predict whether a new patient has any of the following conditions:

- Diabetes
- Heart disease
- Alzheimer's disease

A patient may have zero, one, or multiple conditions.

You are considering two modeling approaches:

Approach A: Train three separate neural networks — one for each disease.

Approach B: Train a single neural network with three output neurons (one per disease) that share a common hidden representation.

Discussion Questions

Which method do you prefer? Briefly justify your answer.

Neural networks with a shared hidden layer (**Approach B**), can capture dependencies between diseases. It can be shown that in some cases, when there is a dependency between the output nodes, having a shared node in the hidden layer can improve the accuracy.

If there is no dependency between diseases (output neurons), then we would prefer to have a separate neural network for each disease as in **Approach A**.

3 When Validation Performance Misleads You

You split your dataset into three subsets: training, validation (dev), and test.

You train 10 different models on the training set and select the model that achieves the lowest error on the validation set.

The selected model achieves strong performance on the validation set. However, when evaluated on the test set, its error is significantly worse.

(a) Give two possible explanations for this behavior.

Two possible explanations:

(a) **Overfitting to the validation set.** Since we selected the model with the lowest validation error among 10 candidates, we may have indirectly overfit to the validation set.

(b) **Distribution mismatch (selection bias).** The training and validation data may not be representative of the test data distribution.

(b) For each explanation, describe briefly what you would check or change to diagnose the issue.

For overfitting to validation:

- Use a larger validation set.
- Use cross-validation.
- Reduce the number of model comparisons.

For distribution mismatch:

- Check whether training/validation data differ systematically from test data.
- Ensure all splits are drawn from the same underlying distribution.

Example: You classify images. All your train and validation examples are daytime images, but your test images are nighttime images.

4 Estimating μ Under Noise

Assume we want to estimate an unknown quantity μ. We observe i.i.d. samples

$$Y_1, Y_2, \ldots, Y_n,$$

where each $Y_i \sim \mathcal{N}(\mu, \sigma^2)$.

(a) How would you estimate μ from the samples?

A standard estimator is the sample mean:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} Y_i.$$

(Equivalently, this is the maximum-likelihood estimator for a Normal distribution with known σ^2.)

(b) The estimator in (a) can be sensitive when σ is large. What can you do to reduce the impact of noise?

Collect more samples (increase n). The variance of the sample mean decreases as

$$\mathrm{Var}(\hat{\mu}) = \frac{\sigma^2}{n}.$$

So increasing n reduces variance even if σ is large.

(c) If sampling is expensive, suggest a different estimator for μ. How does its bias/variance compare to the sample mean?

Use the sample median. It is more robust to outliers / heavy-tailed noise (and can have lower sensitivity to extreme values).

Compared to the mean, the median can have different bias/variance behavior: it may be biased in some settings and can trade off bias and variance depending on the noise distribution. In practice, it is often preferred when data contain outliers or are not well-modeled by a Normal distribution.

5 Bias–Variance Tradeoff (Conceptual Questions)

(a) Describe a setting in which **bias** is worse than variance. Briefly justify your answer.

Example: Consider a casino that runs a very large number of independent games each day. Because outcomes are averaged over many plays, variance in individual game results has little long-term effect. However, even a small systematic bias (e.g., a negative expected return per game) will accumulate over time and lead to consistent losses. Thus, in this setting, bias is more critical than variance.

(b) Describe a setting in which **variance** is worse than bias. Briefly justify your answer.

Example: Consider waiting for a bus. Passengers typically tolerate a small systematic delay (bias) if arrival times are predictable. However, large variability in arrival times (high variance) creates uncertainty and frustration. Moreover, in queuing systems, the expected waiting time depends not only on the mean arrival rate but also on its variance. Hence, variance can be more problematic than bias.

6 Bias, Variance, and Model Flexibility

In the context of bias-variance tradeoff, flexibility is the capacity of a model class to fit a wide variety of functions. More formally:

- A more flexible model can represent more complex patterns in the data.
- A less flexible model is more constrained in the shapes of functions it can represent.

(i) Suppose you have regression data generated by a polynomial of degree 3. Characterize the bias-variance of the estimates of the following models on the data with respect to the true model by circling the appropriate entry.

Model	Bias	Variance
Linear regression	low / high	low / high
Polynomial regression with degree 3	low / high	low / high
Polynomial regression with degree 10	low / high	low / high

Model	Bias	Variance
Linear regression	**high**	**low**
Polynomial regression with degree 3	**low**	**low**
Polynomial regression with degree 10	**low**	**high**

(ii) For each part below indicate whether we would generally expect the performance of a **flexible** statistical learning method to be **better** or **worse** than an **inflexible** method. Justify your answer.

(i) The sample size n is extremely large, and the number of predictors p is small.

A flexible method is better because we are less at risk of overfitting if we have lots of data and only a few relevant predictors. Note you could make an argument that an inflexible method is better since we don't have a lot of predictors. This is not incorrect, but if we have lots of data, use it! It doesn't get as much use in inflexible models.

(ii) The number of predictors p is extremely large, and the number of observations n is small.

Use an inflexible method. Since p is very large, it is easy to overfit or to incorporate predictors into the model that are not actually helpful in predicting the output (e.g. your model may capture noise/variance). Also for small n you are much more likely to see spurious relationships that aren't actually present in the population.

(iii) The relationship between the predictors and response is highly non-linear.

Flexible models. Flexible models will allow you to capture different (non-linear) relationships. Unless you know exactly what the relationship between X and Y is and you choose a very inflexible model that happens to capture a specific non-linear relationship that is true in the world, a flexible hypothesis space is more likely to work better.

(iv) The variance of the error terms, i.e. $\sigma^2 = \text{Var}(\varepsilon)$, is extremely high.

Inflexible. You are very likely to find relationships that are just due to noise. Flexible models will also try to find and fit patterns in the irreducible noise which will cause high variance in the final model.

7 Bias–Variance from Learning Curves

The following graphs plots the error distribution of three models. Comment on each model in terms of bias-variance.

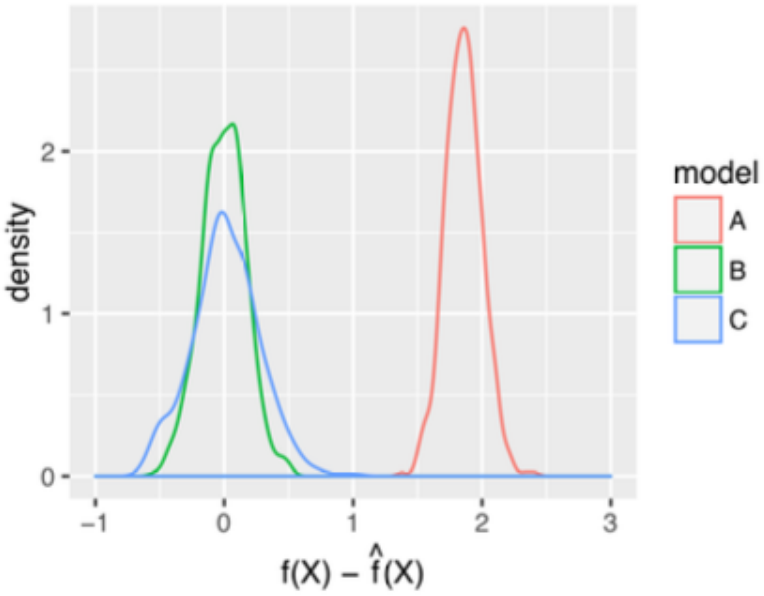

Figure 1: Learning curves for three different models.

The three figures above show the difference between bias-variance in terms of prediction.

Model 1: High bias, low variance
Model 2: Low bias, low variance
Model 3: Low bias, high variance

8 Diagnosing Gradient Descent Behavior

The following plots show the cost function $J(\theta)$ versus the number of iterations for three different learning rates.

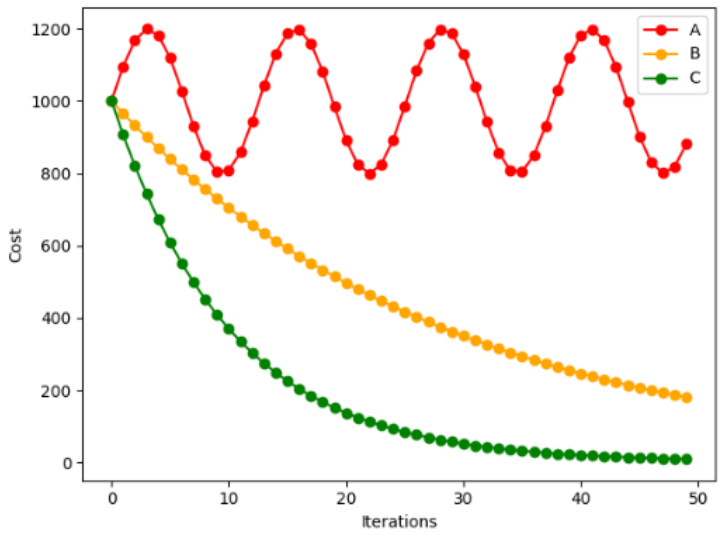

Figure 2: Cost vs. iterations for three different learning rates.

(a) Which plot corresponds to a learning rate that is too high? Justify your answer.

Plot A corresponds to a learning rate that is too high. The oscillations indicate that gradient descent is overshooting the minimum and failing to converge.

(b) Which plot corresponds to a learning rate that is too low? What would you adjust?

Plot B corresponds to a learning rate that is too low. Convergence is very slow. Increasing the learning rate slightly would improve convergence speed.

(c) In Plot C, the cost flattens out after some iterations. What does this indicate?

Plot C shows proper convergence. The cost stabilizes because gradient descent has reached (or is very close to) the minimum.

(d) If the cost never meaningfully decreases despite many iterations, what might be wrong?

Possible issues include:

- The learning rate is too high.
- Features are not properly scaled.
- The model is misspecified (e.g., the data is not well modeled by a linear relationship).
- There are significant data quality issues.