Section 03: Bias-Variance Tradeoff and Train-Test Split

1. Bias-Variance Trade-off

Consider a simple statistical learning setting, in which we assume that there is some unknown function relating two random variables X and Y (e.g. $Y = 2X$). Let us denote this function by $Y = \eta(X)$; however, we don't know specifically what this function $\eta(\cdot)$ is. Our goal is as follows. Given X, we want to predict Y with the smallest possible error, in expectation. We formalize this notion below.

(a) Find the function η that minimizes the expected squared error $\mathbb{E}[(Y - \eta(X))^2]$. **Hint:** Observe that $\mathbb{E}[(Y - \eta(X))^2] = \mathbb{E}[\mathbb{E}[(Y - \eta(X))^2|X = x]]$ (The "Tower Rule").

(b) While ideally we want η to be what we computed above, in reality, however, we are restricted to our training data and a function class, the best we can do is $\hat{f}_D = \arg\min_{f \in F} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$, where $D = \{(x_i, y_i)\}$. Here, (x_i, y_i) is a sample from distribution P_{XY}. To account for the prediction error (i.e. quality of our estimator $\hat{f}_D$), we need to calculate

$$\mathbb{E}[\mathbb{E}_D[(Y - \hat{f}_D(x))^2]|X = x]]$$

We can break the expectation into

$$\mathbb{E}[\mathbb{E}[(Y - \eta(x))^2|X = x]] + \mathbb{E}_D[(\eta(x) - \hat{f}_D(x))^2]$$

$\mathbb{E}[\mathbb{E}[(Y - \eta(x))^2|X = x]]$ is called **irreducible error** — the error incurred even in ideal situation.

$\mathbb{E}_D[(\eta(x) - \hat{f}_D(x))^2]$ is called **learning error** — the error incurred by the learning setting (e.g. insufficient data, the chosen model class F is not expressive enough etc.)

Express the **learning error** in terms of

- bias — $(\eta(x) - \mathbb{E}_D[\hat{f}_D(x)])$
- and variance — $\mathbb{E}_D[(\mathbb{E}_D[\hat{f}_D(x)] - \hat{f}_D(x))^2]$

and explain why there is a trade-off.

Hint: $\eta(x) = \theta$, $\hat{f}_D(x) = \hat{\theta}$ and $\mathbb{E}[\hat{f}_D(x)] = \theta^*$

2. Generalized Least Squares Regression

We already saw linear regression in class and the ridge regression will be covered in week three. Here we consider a problem that generalizes both of these. As a reminder, in linear regression, we seek a model that captures a linear relationship between input data and output data. The general case we consider imposes additional structure on the model.

Consider an experiment in which you have n data points $x_i \in \mathbb{R}^d$ and corresponding n observations y_i. We wish to come up with a model $\omega \in \mathbb{R}^d$ that satisfies the following properties: first, the error $\sum_{i=1}^n (x_i^\top \omega - y_i)^2$ should be small; second, we don't want small changes in training data resulting in large changes in solution; third, we want to put different weights in controlling the magnitude of different coordinates of ω. We therefore define

$$\widehat{\omega}_{\text{general}} = \arg\min_{\omega} \sum_{i=1}^{n} (y_i - x_i^\top \omega)^2 + \lambda \sum_{i=1}^{d} D_{ii} \omega_i^2.$$

Here, D is a diagonal matrix, with positive entries on the diagonal. Observe that when D is the identity matrix, we recover ridge regression, and when $\lambda = 0$, we recover least squares regression. Different weights on D_{ii} cause the magnitudes of ω_i to be controlled differently.

2.1. Closed form in the general case

Deduce the closed form solution for $\widehat{\omega}_{\text{general}}$. You should be comfortable with proofs in the ”coordinate” form as well as the ”matrix” form.

2.2. Special cases: linear regression and ridge regression

(a) In the simple least squares case ($\lambda = 0$ above), what happens to the resulting $\widehat{\omega}$ if we double all the values of y_i?

(b) In the simple least squares case ($\lambda = 0$ above), what happens to the resulting $\widehat{\omega}$ if we double the data matrix $X \in \mathbb{R}^{n \times d}$?

(c) Suppose $D = I$ (that is, it is the identity matrix). That is, this is the *ridge* regression setting. Explain why $\lambda > 0$ ensures a ”well-conditioned” setting.

3. Biased Test Error

Is the test error unbiased for these programs? If not, how can we fix the code so it is?

3.1. Program 1

```
# Given dataset of 1000-by-50 feature
# matrix X, and 1000-by-1 labels vector

mu = np.mean(X, axis=0)
X = X - mu

idx = np.random.permutation(1000)
TRAIN = idx[0:900]
TEST = idx[900::]

ytrain = y[TRAIN]
Xtrain = X[TRAIN, :]

# solve for argmin_w ||Xtrain*w - ytrain||_2
w = np.linalg.solve(np.dot(Xtrain.T, Xtrain), np.dot(Xtrain.T, ytrain))

b = np.mean(ytrain)

ytest = y[TEST]
Xtest = X[TEST, :]

train_error = np.dot(np.dot(Xtrain, w)+b - ytrain,
              np.dot(Xtrain, w)+b - ytrain ) / len(TRAIN)
test_error = np.dot(np.dot(Xtest, w)+b - ytest,
              np.dot(Xtest, w)+b - ytest ) / len(TEST)

print('Train error = ', train_error)
print('Test error = ', test_error)
```

3.2. Program 2

```python
# Given dataset of 1000-by-50 feature
# matrix X, and 1000-by-1 labels vector

def fit(Xin, Yin, _lambda):
    mu = np.mean(Xin, axis=0)
    Xin = Xin - mu
    w = np.linalg.solve(np.dot(Xin.T, Xin) + _lambda * np.eye(Xin.shape[1]), np.dot(Xin.T, Yin))
    b = np.mean(Yin) - np.dot(w, mu)
    return w, b

def predict(w, b, Xin):
    return np.dot(Xin, w) + b

idx = np.random.permutation(1000)
TRAIN = idx[0:800]
VAL = idx[800:900]
TEST = idx[900::]

ytrain = y[TRAIN]
Xtrain = X[TRAIN, :]
yval = y[VAL]
Xval = X[VAL, :]

# use cross validation to pick the best hyper-parameter to use
lambdas = [10 ** -5, 10 ** -4, 10 ** -3, 10 ** -2]
err = np.zeros(len(lambdas))

for idx, _lambda in enumerate(lambdas):
    w, b = fit(Xtrain, ytrain, _lambda)
    yval_hat = predict(w, b, Xval)
    err[idx] = np.mean((yval_hat - yval)**2)

lambda_best = lambdas[np.argmin(err)]

Xtot = np.concatenate((Xtrain, Xval), axis=0)
ytot = np.concatenate((ytrain, yval), axis=0)

w, b = fit(Xtot, ytot, lambda_best)

ytest = y[TEST]
Xtest = X[TEST, :]

ytot_hat = predict(w, b, Xtot, lambda_best)
train_error = np.mean((ytot_hat - ytot) **2)
ytest_hat = predict(w, b, Xtest, lambda_best)
test_error = np.mean((ytest_hat - ytest) **2)

print('Train error = ', train_error)
print('Test error = ', test_error)
```

4. Extra: Stein's Paradox

In this problem, we'll use bias-variance tradeoff to find a non-obvious way of estimating the mean of unrelated distributions.

So far in class, we've always been trying to learn a function – given a bunch of features, understand how they predict the single-number output. In this problem, we're trying to do something a little different. We have n completely unrelated probability distributions. We're going to get one sample from each of the distributions, and attempt to predict each of their means. For some examples, our distributions might be: high temperature in Chicago on January 1st, low temperature in Seattle on December 1st, and your friend's score on the midterm.

More formally, let $\theta \in \mathbb{R}^n$ be the (unknown) true means of our n distributions. We will get a vector X where each $X_i \sim \mathcal{N}(\theta_i, \sigma^2)$. We're assuming that every distribution has the same variance, but our means could be very different. Our job is to report $\hat{\theta}$ to minimize our expected error: $\mathbb{E}\left[||\hat{\theta} - \theta||_2^2\right]$.

4.1. The Natural Estimator

The most natural estimator is to just guess X (i.e. set $\hat{\theta} = X$). It doesn't seem like we have any other viable strategy. We'll use bias-variance tradeoff to show that there's actually a better estimator.

(a) Split the error into bias^2 and variance. I.e. show

$$\mathbb{E}[||\hat{\theta} - \theta||_2^2] = ||\mathbb{E}[\hat{\theta}] - \theta||_2^2 + \mathbb{E}[||\hat{\theta} - \mathbb{E}[\hat{\theta}]||_2^2]$$

Hint: add and subtract $\mathbb{E}[\hat{\theta}]$.

(b) What is the variance of the estimator $\hat{\theta} = X$? Hint: Remember that for a random variable Z, Var$(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$

(c) What is the bias^2 of the estimator $\hat{\theta} = X$?

4.2. A Different Estimator

The Bias-Variance Tradeoff says that since our error is just the sum of the bias^2 and the variance, if we can find a way to "tradeoff" bias for variance, we can affect our error. With our previous estimator, the two sources of error are quite imbalanced. None of our error is from bias, it all comes from variance. Can we think of a way to reduce variance (even if it means increasing the bias)?

Normally, the way we would reduce variance would be to sample the random variables again and take the average of the samples. But we can't do that for this problem (it would take us a whole year to get another high temperature on January 1st). Another way to decrease the variance is to "scale down" the random variable. E.g. say we'll have $\frac{9}{10}$ of our estimator come from the random object, and the remaining $\frac{1}{10}$ come from somewhere else. What else can we use? Let's just use 0. This kind of estimator is sometimes called a "shrinkage estimator" because we're pulling the results toward 0.

Our estimator is going to be $\frac{9}{10}X$. We've certainly decreased the variance. But that should sound crazy – we're biasing ourselves. We're intentionally guessing something we **know** is a biased estimator. But our hope is that we will decrease the variance enough to more than cancel out the increase in bias. Let's see.

(a) We've changed the estimator, does the error still break down neatly into bias and variance? Or do we have to change some math from part a of the last question?

(b) What is the variance of the estimator $\hat{\theta} = \frac{9}{10}X$?

(c) What is the bias^2 of the estimator $\hat{\theta} = \frac{9}{10}X$?

(d) Suppose you know that the variance of our samples is quite a bit. Specifically assume $\sigma^2 > 1/10\theta_i^2$ for all i. (For the temperature examples we gave, this is pretty reasonable. At least if we use Celsius temperatures. The average Celsius high in Chicago is about 4 degrees, so a variance of about 1.6 degrees Celsius suffices) Have we improved the estimator?

4.3. Thinking More about the Estimators

The estimator we came up with in the last problem is unintuitive for more reasons than we've already seen. Suppose we scaled all our data points (i.e. instead of X we got $aX + b$), e.g. we were expecting to get our data in Celsius, but it came to us in Fahrenheit. We would expect that scaling our old estimate, i.e. now reporting $a\hat{\theta} + b$ would give us the same answer as if we did our estimate afresh knowing we were getting data in Fahrenheit.

(a) Is the "natural estimator" scale invariant?

(b) Is the "shrinkage estimator" $\frac{9}{10}X$ scale invariant?

It turns out we can do even better than estimating $\frac{9}{10}X$– our idea was to shrink X toward 0 by a constant ratio (i.e. multiply everything by $9/10$). If we instead shrink it in a way that depends on σ^2 and $||X||_2^2$, we'll be able to come up with an estimator that has less error than X, regardless of the relationship between σ^2 and θ.

The "James-Stein Estimator, $\hat{\theta} = \left(1 - \frac{(n-2)\sigma^2}{||X||_2^2}\right) X$, always has less error than X.

4.4. A Harder Calculation

Want more practice with bias-variance tradeoff? Here's another version of Stein's Paradox. Instead of shrinking toward 0, shrink toward $\overline{X}$, the mean of all of your data points, i.e. $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. If your data points are coming from similar sources (say each θ_i is a different baseball player's true batting average), you can think of this as reflecting a belief that all the means should be generally similar. (Though this shrinking still works even if the sources are very different, and the result is still counter-intuitive). Let **1** be the $n \times 1$ vector of all 1's, and let λ be a real number between 0 and 1. In this problem we'll find the way to choose λ to make $\hat{\theta} = (1-\lambda)X + \lambda\overline{X}\mathbf{1}$ as good of an estimator as possible.

(a) What is the variance of the estimator $\hat{\theta} = (1-\lambda)X + \lambda\overline{X}\mathbf{1}$?

(b) What is the bias^2 of the estimator?

(c) What value of λ will result in the best estimator?