Homework 1: ML

Solutions

Q1. Recall the definition of the expected risk $R(f)$ of a predictor f. While this cannot be computed in general note that here we defined $P_{\mathcal{X}\times\mathcal{Y}}$. Which function f^* is an obvious Bayes predictor? Make sure to explain why the risk $R(f^*)$ is minimum at f^*.

$f^*(x) = g(x)$

Bayes Predictor, $f^* = \arg\min_f R(f) = \arg\min_f \mathbb{E}_{(x,y)\sim P_{\mathcal{X}\times\mathcal{Y}}}[l(\hat{y}, y)]$
$= \arg\min_f \mathbb{E}_{(x,y)\sim P_{\mathcal{X}\times\mathcal{Y}}}[\frac{1}{2}(\hat{y} - y)^2]$

This is non-negative, so the function $g(x)$ which has zero risk is the Bayes Predictor.

Q2. Using $\mathcal{H}_2$ as your hypothesis class, which function $f_{\mathcal{H}}$ is a risk minimizer in $\mathcal{H}_2$? Recall the definition of the approximation error. What is the approximation error achieved by $f^*_{\mathcal{H}_2}$?

$f^*_{\mathcal{H}_2}(x) = g(x)$, this is the Bayes predictor and is in the hypothesis space $\mathcal{H}_2$ so it is the risk minimizer within $\mathcal{H}_2$

Q3 Considering now $\mathcal{H}_d$, with $d > 2$. Justify an inequality between $R(f^*_{\mathcal{H}_d})$ and $R(f^*_{\mathcal{H}_2})$. Which function $f^*_{\mathcal{H}_d}$ is a risk minimizer in $\mathcal{H}_d$? What is the approximation error achieved by $f^*_{\mathcal{H}_d}$?

In a general case, the inequality is: $R(f^*_{\mathcal{H}_d}) \leq R(f^*_{\mathcal{H}_2})$

For any subspace $\mathcal{H}_d$, with $d > 2$, all functions from $\mathcal{H}_2$ are included in $\mathcal{H}_d$ because you can set the coefficients $b_k, k \in \{3...d\}$ to zero and set b_0, b_1, b_2 accordingly to match the required function from $\mathcal{H}_2$. So the risk minimizer function from the hypothesis space $\mathcal{H}_2$, $f^*_{\mathcal{H}_2}$, is included in $\mathcal{H}_d$, $d > 2$. So the minimzer from $\mathcal{H}_d$, $d > 2$ i.e $f^*_{\mathcal{H}_d}$ has to either further minimze the risk or at worst it matches that from $f^*_{\mathcal{H}_2}$.

The minimizer $f^*_{\mathcal{H}_d}$ is the same as $g(x)$ but it is defined as $b_0 = a_0, b_1 = a_1, b_2 = a_2, b_{3...d} = 0$. And as a result the approximation error is zero again because this is the Bayes predictor.

Q4. For this question we assume $a_0 = 0$. Considering $H = \{f : x \to b_1x; b_1 \in \mathbb{R}\}$, which function $f^*_{\mathcal{H}}$ is a risk minimizer in $\mathcal{H}$? What is the approximation error achieved by $f_{\mathcal{H}}$? In particular what is the approximation error achieved if furthermore $a_2 = 0$ in the definition of true underlying relation g(x) above?

We are given that $a_0 = 0$. Therefore $y = a_1x + a_2x^2$.

Also prediction $\hat{y} = f(x) = b_1x$. So the minimizer $f^*_{\mathcal{H}} = \arg\min_{f\in\mathcal{H}} \frac{1}{2}\mathbb{E}[(f(x) - y)^2]$.

Solving the expectation alone (note we are told that $x \sim Unif[0,1]$ so in $[0,1]$, probability density function $f_X(x) = 1$):
$\mathbb{E}[(f(x) - y)^2] = \int_0^1 [(f(x) - y)^2]\,dx$
$= \int_0^1 [(b_1x - a_1x - a_2x^2)^2]\,dx$
$= \int_0^1 x^2 \cdot [(b_1 - a_1 - a_2x)^2]\,dx$
$= \int_0^1 x^2 \cdot [b_1^2 + a_1^2 + a_2^2x^2 - 2a_1b_1 - 2a_2b_1x + 2a_1a_2x]\,dx$
$= \int_0^1 b_1^2x^2 + a_1^2x^2 + a_2^2x^4 - 2a_1b_1x^2 - 2a_2b_1x^3 + 2a_1a_2x^3]\,dx$
$= \frac{b_1^2x^3}{3} + \frac{a_1^2x^3}{3} + \frac{a_2^2x^5}{5} - \frac{2a_1b_1x^3}{3} - \frac{2a_2b_1x^4}{4} + \frac{2a_1a_2x^4}{4}\Big|_0^1$
$= \frac{b_1^2}{3} + \frac{a_1^2}{3} + \frac{a_2^2}{5} - \frac{2a_1b_1}{3} - \frac{2a_2b_1}{4} + \frac{2a_1a_2}{4}$
We now need to find the function $f \in \mathcal{H}$ that minimizes this i.e. the value of b_1 that minimizes the above expression. To do this, we can differentiate by b_1 as:
$\frac{d\mathbb{E}[(f(x)-y)^2]}{db_1} = \frac{2b_1}{3} - \frac{2a_1}{3} - \frac{2a_2}{4} = 0$
Or $b_1 = a_1 + \frac{3}{4}a_2$

We can verify that the double derivative is positive ($\frac{d^2\mathbb{E}[(f(x)-y)^2]}{db_1^2} = \frac{2}{3}$) and hence the value found for b_1 is a minima.

The associated approximation error, $R(f_{\mathcal{H}}) - R(f^*) = R(f_{\mathcal{H}}) - 0$
$= \frac{1}{2}\mathbb{E}[(f(x) - y)^2] = \frac{1}{2}\int_0^1 (b_1x - a_1x - a_2x^2)^2 dx$
$= \frac{1}{2}\int_0^1 (\frac{3}{4}a_2x - a_2x^2)^2 dx = \frac{1}{2}\int_0^1 \frac{a_2^2}{16}(3x - 4x^2)^2 dx$
$= \frac{a_2^2}{32}\int_0^1 9x^2 + 16x^4 - 24x^3 dx$
$= \frac{a_2^2}{32}\left[\frac{9}{3} + \frac{16}{5} - \frac{24}{4}\right] = \frac{a_2^2}{160}$
Furthermore if we apply the constraint that $a_2 = 0$, then $f^*_{\mathcal{H}}(x) = a_1x$. And the associated approximation error is zero.

Q5. Show that the empirical risk minimizer (ERM) $\hat{b}$ is given by the following minimization $\hat{b} = \arg\min_b ||Xb - y||_2^2$.

As long as you show that opening out the matrix expression gives you the same result as the ERM, you get the points.

ERM: $R(\hat{f}_n) = \min_{f\in\mathcal{H}_d} \frac{1}{n}\sum_{i=1}^{n} l(f(x_i, y_i) = \min_{f\in\mathcal{H}_d}\frac{1}{2n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 = \min_{f\in\mathcal{H}_d}\sum_{i=1}^{n}(f(x_i) - y_i)^2$

$||Xb - y||_2^2 = \left(\sqrt{\sum_{i=1}^{n}(f(x_i) - y_i)^2}\right)^2 = \sum_{i=1}^{n}(f(x_i) - y_i)^2 = \sum_{i=1}^{n} 2 \times l(f(x_i), y_i)$

The 2 in the expression can be dropped when we look to minimize.

Therefore: $\arg\min_{b} ||Xb - y||_2^2 = \min_{f\in\mathcal{H}_d}\frac{1}{2n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$

Q6. If $N > d$ and X is full rank, show that $\hat{b} = (X^TX)^{-1}X^Ty$. (Hint: you should take the gradients of the loss above with respect to b). Why do we need to use the conditions $N > d$ and X full rank?

From Q5, we showed that ERM, $\hat{b} = \arg\min_{b} ||Xb - y||_2^2$

$||Xb - y||_2^2 = (Xb - y)^T(Xb - y)$

$= (b^TX^T - y^T)(Xb - y)$

$= b^TX^TXb - y^TXb - b^TX^Ty + y^Ty$

$= b^TX^TXb - 2b^TX^Ty + y^Ty$ (because y^TX^Tb is a scalar and we can add the two terms)

To minimize this, we need to take the derivative w.r.t. b as follows:

$\frac{d||Xb-y||_2^2}{db} = \frac{d}{db}\left(b^TX^TXb - 2b^TX^Ty + y^Ty\right) = 2X^TXb - 2X^Ty = 0$

Or $2X^TXb = 2X^Ty$

Or $b = (X^TX)^{-1}X^Ty$

The second derivative $(2X^TX)$ is non negative so the obtained b is a minima.
Therefore $\hat{b} = \arg\min_{b} ||Xb - y||_2^2 = (X^TX)^{-1}X^Ty$

For X^TX to be invertible, we need the rank of X^TX to be d because X^TX is a $d \times d$ matrix. To satisfy this we need both of X to be full rank as well as $N > d$. If the X is full rank, that means that $rank(X) = min(N, d)$ so if $N < d$ then the condition on X^TX would not be satisfied. (Note we use that $rank(X) = rank(X^TX)$)

Q7. Write a function called least_square_estimator taking as input a design matrix X and the corresponding vector y returning b. Your function should handle any value of N and d, and in particular return an error if $N \leq d$. (Drawing x at random from the uniform distribution makes it almost certain that any design matrix X with $d \geq 1$ we generate is full rank).

```
def least_squares_estimator(X, y):
    if X.shape[0] <= X.shape[1] - 1:
        print("Error. Number of features > Number of examples.")
        return
    b = np.linalg.pinv(X.T @ X) @ X.T @ y
    return b
```

Q8. Recall the definition of the empirical risk $\hat{R}(\hat{f})$ on a sample $\{xi, yi\}_{i=1}^{N}$ for a prediction function $\hat{f}$. Write a function empirical risk to compute the empirical risk of f_b taking as input a design matrix X, a vector y and the vector b parametrizing the predictor.

```
def empirical_risk(X, y, b):
    preds = X @ b
    n = len(y)
    emp_risk = np.sum((preds - y)**2) / (2*n)
    return emp_risk
```

Q9. Use your code to estimate $\hat{b}$ from x_train, y_train using $d = 5$. Compare $\hat{b}$ and a. Make a single plot (Plot 1) of the plan (x, y) displaying the points in the training set, values of the true underlying function g(x) in [0, 1] and values of the estimated function f in [0, 1]. Make sure to include a legend to your plot. The code to estimate this is attached at the end of the assignment

Comparing the found $\hat{b}$ with true a:
Vector $\hat{b}$: $[8.820, 2.001, 4.894, -2.882e^{-06}, 1.93e^{-06}, -4.610e^{-07}]$
Vector a : $[8.820, 2.001, 4.894]$
We can see that there's pretty much a perfect match with the first three coefficients matching upto 3 decimal places and subsquent coefficients in $\hat{b}$ taking very small values.
The required figure is Fig. 1. This might look slightly different depending on a, but you should show a perfect fit.

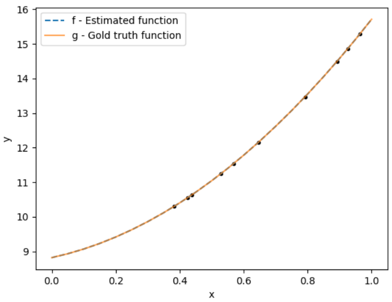


Figure 1: Scatter plot of train points along with true underlying function g and estimated function f (d-value 5) from x = 0 to x=1. We can see a perfect fit between f and g.

Q10. Now you can adjust d. What is the minimum value for which we get a "perfect fit"? How does this result relates with your conclusions on the approximation error above?

For d=2 and above there is a perfect fit. This agrees with our conclusions from Q2 and Q3 where the approximation error for $\mathcal{H}_2$ is zero and approximation error for $\mathcal{H}_d(d > 2) \leq \mathcal{H}_2$ (and hence is zero for all of those as well). Basically the true distribution function g belongs to all hypothesis spaces from d=2 onwards. A plot to confirm this programmatically would look something like Figure 2.

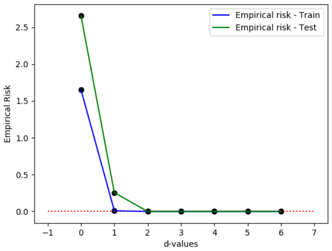
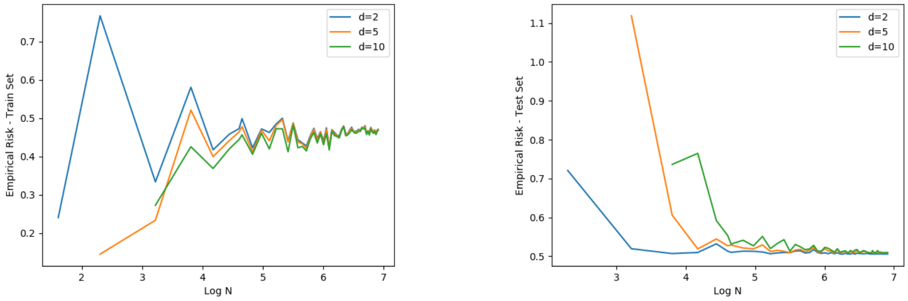


Figure 2: Test for minimum d-value needed for perfect fit. We can see that there is a perfect fit i.e. zero empirical risk, for $d \geq 2$

Q11. Plot e_t and e_g as a function of N for $d < N < 1000$ for $d = 2, d = 5$ and $d = 10$ (Plot 2). You may want to use a logarithmic scale in the plot. Include also plots similar to Plot 1 for 2 or 3 different values of N for each value of d.

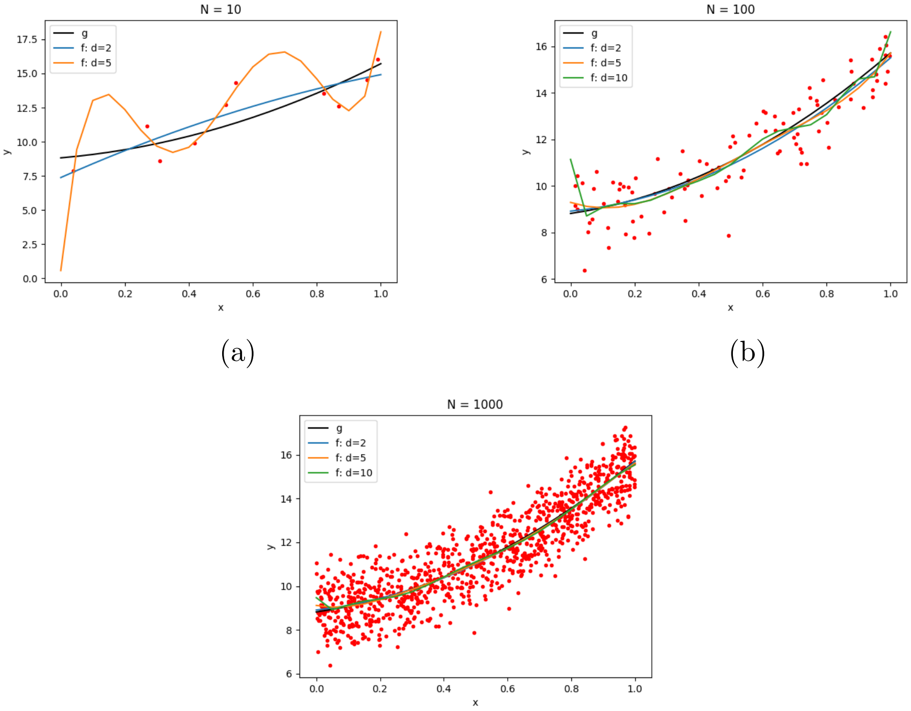


(a)

(b)

Figure 3: (a) Training error e_t as a function of log N for d=2, 5, 10 where N is the size of the train set. Note that the x-axis is in the log scale. (b) Generalization error e_g as a function of log N for d=2, 5, 10 where N is the size of the train set. Note that the x-axis is in the log scale and the size of the test set is always 1000.

Fig. 3a and Fig. 3b are the required plots of training error and generalization error. The plot indicates that the training error tends to even out with increase in N. For generalization we observe a trend of decreasing test error as we increase N. Also a higher d-value seems to have lower train error and higher generalization error for the same value of N. This trend is more apparent at smaller values of N with there being negligible difference as N becomes very large. The model tends to overfit to the train data at small N and large d.



(a)

(b)

(c)

Figure 4: Scatter plot of (x, y) train points for N=10 (a), N=100 (b), N=1000 (c) along with true underlying function g and estimated function f from x = 0 to x=1 while varying d

Fig. 4a, Fig. 4b and Fig. 4c are the 3 required figures for N=10, 100 and 1000 respectively. Each plot has the train points scattered with the function lines marked. You can see that with increase in d the function tends to overfit with this effect less visible as we increase N. The exact values of N and d in your plots can be different, as long as you're able to see the expected relationships.

Q12. Recall the definition of the estimation error. Using the test set, (which we intentionally chose large so as to take advantage of the law of large numbers) give an empirical estimator of the estimation error. For the same values of N and d above plot the estimation error as a function of N

Estimation Error, $R(\hat{f}_n) - R(\hat{f_{\mathcal{H}}})$ where the hypothesis space is $\mathcal{H}$.

$$R(f_{\mathcal{H}}) = \underset{f\in\mathcal{H}}{\arg\min}\, \mathbb{E}[l(f(x), y)]$$

$$R(\hat{f}_n) = \min_{f\in\mathcal{H}_d} \frac{1}{n}\sum_{i=1}^{n} l(f(x_i, y_i)$$

The empirical risk $(R(\hat{f}_n))$ can be obtained from the predictions and the data. To find the

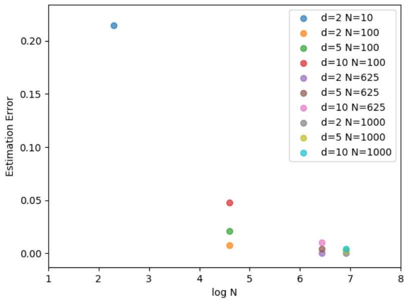


Figure 5: Plot of estimation errors varying w.r.t. log N where N is the training set size. We try out 4 different train set sizes and up to 3 d-values for each N.

risk minimizer within the hypothesis space, we use an estimate. We know that $f_{\mathcal{H}}$ is supposed to be the function which minimizes expected loss within the hypothesis space. But since we know the underlying function of the data ($g(x)$), we can use this as the minimizer. Additionally, we are given a large test set. So by the law of large numbers, the mean is an unbiased estimator of the expectation. So:

$$R(f_{\mathcal{H}_d}) = \underset{f \in \mathcal{H}_d}{\arg\min}\ \mathbb{E}[l(f(x), y)]$$

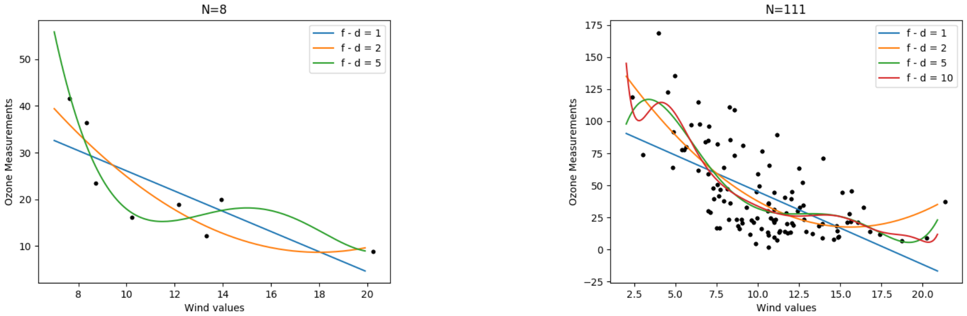
$$= \underset{f \in \mathcal{H}_d}{\arg\min} \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} l(f(x_i), y_i) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} l(a_0 + a_1 x_i + a_2 x_i^2, y_i).$$

$$\text{Estimation Error} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} l(\hat{f}_n(x_i), y_i) - \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} l(a_0 + a_1 x_i + a_2 x_i^2, y_i)$$

We try out 4 different N values and up to 3 d-values. The plot can be seen in Fig. 5. The first trend we observe are that with an increase in N, there is generally a decrease in estimation error. This is not unexpected because we would expect that as the sample gets bigger the estimate is closer to the true minimizer. The other trend we observe is that within the same N, when the d-value is larger there is more of an estimation error on the test set so it is likely that the more expressive functions fit with higher d-values have overfit to the train set and are unable to generalize in the test setting. Also this difference gets smaller with higher N as we would expect less overfitting with a larger sample.

Q13. The generalization error gives in practice an information related to the estimation error. Comment on the results of (Plot 2 and 3). What is the effect of increasing N? What is the effect of increasing d?

The general trend indicates that the training error tends to even out with increase in N. For generalization we observe a general trend of decreasing in test error as we increase N. This is not too unexpected as we expect the model to generalize better when the parameters are set based on a larger set of points. The main point to note with d is that there is more overfitting (especially visible in 4a) when d is higher which results in smaller training errors (Fig. 3a) and larger generalization errors (Fig. 3b). The trends in d are more pronounced for smaller values of N. The same trends for N, d are also seen in estimation error as shown



(a)

(b)

Figure 6: Scatter plot of data points (each subplot varies N) and estimating functions f to predict ozone value as a function of wind value while varying d

in Fig. 5 for Q12. By in large, an increase in N reduces estimation error and for a particular value of N a higher d value has a higher estimation error.

Q14. Besides from the approximation and estimation there is a last source of error we have not discussed here. Can you comment on the optimization error of the algorithm we are implementing?

We would expect zero optimization error because our analytical approach finds the exact solution for the least squares optimization. Given the data observed X, we cannot optimize any further than this closed form solution.

Q15. Reporting plots, discuss the again in this context the results when varying N (subsampling the training data) and d.

The answer is open ended. As long as you provide plots to view the trends in N and d you should get the points. The general trend (Figure 6) is a decrease in ozone value as wind value increases where a balanced fit is seen with $d = 2$. Within each subsample, as we increase d, we observe more overfitting particularly as d approaches N. From Figure 7, for the same N, a higher d-value usually corresponds to a lower risk value since the more expressive function is better able to represent the train sets.

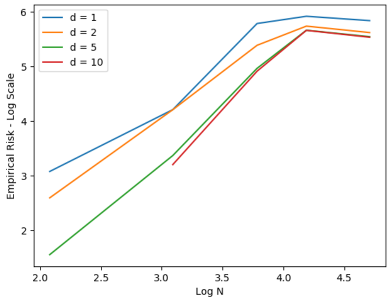


Figure 7: Empirical risk e_t (in the log scale) as a function of log N for d=1, 2, 5, 10