Advance Statistical Learning. Sheet 5

Consider again the data set dat_class.csv from Sheet 4 Exercise 2 and the k-nearest neighbours' method. This time we want to estimate the generalization error on this data set using cross validation.

a) **First, we want to investigate how the number c of cross validation folds influences the results. Perform c-fold cross validation 30 times for each combination of k ∈ {1, 5, 10, 20, 50, 100, 200} neighbours and c ∈ {2, 5, 10, 20, 50, 100, 200} cross validation (CV) folds. Use the misclassification rate as the error measure again. Calculate the cross-validation error as the average misclassification rate over the c folds in each of the 30 repetitions.**

We can see the boxplots display the distribution of CV errors across 30 repetitions for each combination of $k \in \{1,5,10,20,50,100,200\}$ and $c \in \{1,5,10,20,50,100,200\}$ in Figure 1, Figure 2 and Figure 3.

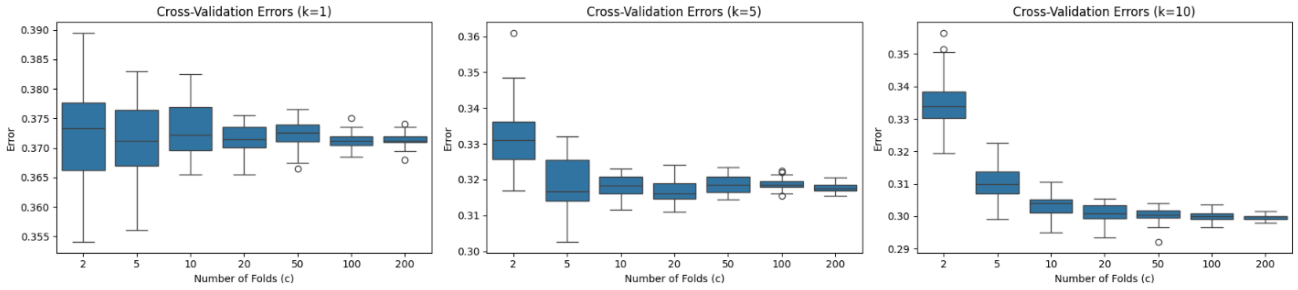


Figure 1: CV Errors for k = {1, 5, 10}

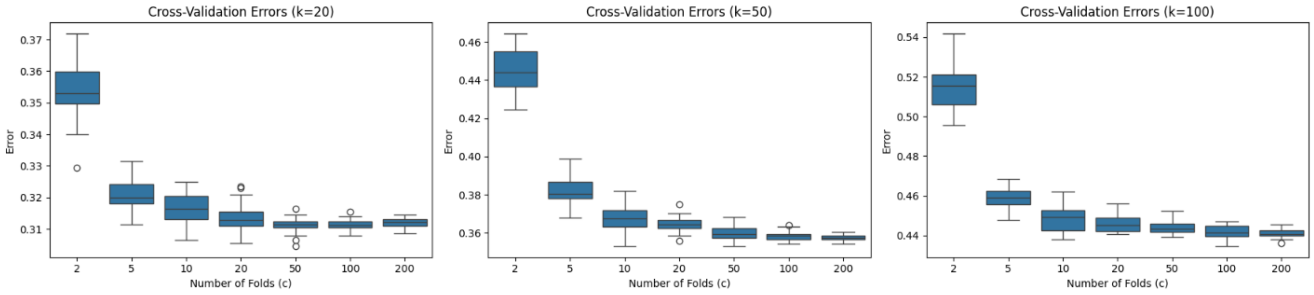


Figure 2: CV Errors for k = {20, 50, 100}

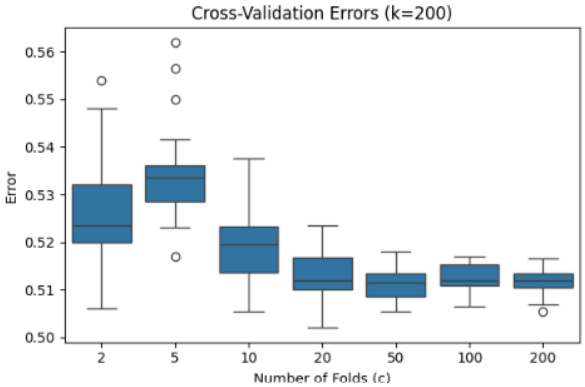


Figure 3: CV Errors for k = 200

We observe that there is a higher variance when the c value is small.

- For small values of c (like 2 or 5, Figure 1), the variation in CV error across the 30 repetitions is high. This happens because fewer folds mean larger validation sets and, therefore, fewer training points, making results more variable.

In the other hand, the estimation gets more stable and accurate at larger c values:

- As c increases ($c > 50, c = \{50,100,200\}$, Figure 2 and Figure 3), the spread of errors decreases. This happens because with more folds, each training set is mostly the full data set, making the error estimate more reliable.

Also, there's a convergence of the median error:

- Across all k values, the median CV error stabilizes as c increases. Thus, we could say that using a large c gives us more stable and accurate estimates of generalization error.

We can say that even though larger c values give us more stable error estimates, they also require more computation. Therefore, we think that 10-fold CV gives a good compromise between accuracy and efficiency.

b) What is the cross validation called if we use c = 2000? Describe in one sentence how a boxplot of the cross validation errors of 30 replications with this c would look like.

If we set $c = 2000$, which is the number of observations in the dataset (this means $c = 2000 = n$), then, this is called **Leave-One-Out Cross-Validation (LOOCV)**. The name comes directly from how the method works. In each iteration it:

1. **Leaves out one** single data point as the validation set.
2. Trains the model on the remaining $n - 1 = 1999$ data points.
3. **Evaluates** the model on the **left-out** point.

The look of the boxplot will be defined by the spread or variance of the 30 repetitions being very narrow, nearly a flat line. We know this because each fold leaves out only one sample, so all training sets are almost identical. Therefore, each CV error estimate is very similar across repetitions.

So, we can remark that LOOCV provides very low-variance but potentially high-variance estimates per fold although it is computationally expensive. The boxplot will appear as a nearly flat horizontal bar.

c) Use 10-fold cross validation to calculate the cross validation error once for every k ∈ {1, 2, …, 50} and visualize the errors as a function of k. Briefly comment on your result.

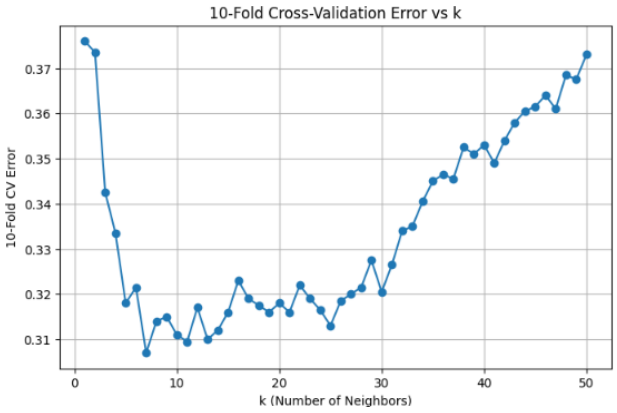


Figure 4: 10 - Fold CV Error vs k Plot

The plot in Figure 4 shows how the 10 - fold (CV) error varies with the number of neighbours $k \in \{1,2,\ldots,50\}$ in the KNN classifier.

We observe that there's a **high error at very small k** ($k = \{1,2,3\}$). This is because the model **overfits** by memorizing the training data, so it reacts strongly to noise. As a result, the error is high when generalized to unseen data.

The minimum CV error occurs around $\boldsymbol{k = (9 - 11)}$. We will refer this as the **optimal region** where the model achieves a good trade-off between bias and variance. The model generalizes well without being too simple or too complex.

As the **k gets larger**, the error increases gradually. The model is becoming too smooth and starts **underfitting**. It averages over many distant neighbours and loses the ability of capturing local patterns.

To summarize, we would recommend using a $k = (9 - 11)$ because it performs the smallest CV error. Therefore, this range should be chosen as the **final model parameter**.

Q2> Consider the hypothesis space H which contains all $h : R \to \{0, 1\}$ of the form

(
1 if $x \in [z_1, z_2] \cup [z_3, z_4]$,
$h(x) =$
0 else
where $z_1, z_2, z_3, z_4 \in R$ and $z_1 \leq z_2 \leq z_3 \leq z_4$.
Show that the VC dimension of H is 4.
Hint: If you use the same idea more than once in your proof, a short analog to . . . reference after the rst time is sucient.

A> Hypothesis space is defined as:

$$h(x) = \begin{cases} 1 & , \text{if } x \in [z_1, z_2] \cup [z_3, z_4] \\ 0 & , \text{otherwise} \end{cases}$$

where, $z_1 \leq z_2 \leq z_3 \leq z_4$.

Clearly hypothesis plane is 1D as it depends only on x.

Possible labellings:

$$2^n = 2^4 = 16$$

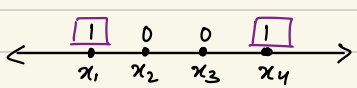
To proove V.C. dimension is 4, it should shatter all sets / combinations of them, considering following points →

1001 for point x_1 & x_4.
1100 for point x_1 & x_2.
0110 for point x_2 & x_3.
0101 for point x_2 & x_4.
⋮

and so on.

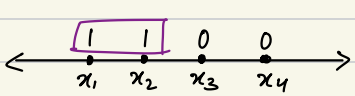
We can shatter them as shown below:

1001 →



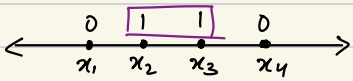
We can choose the interval $[x_1, x_1] \cup [x_4, x_4]$

1100 →



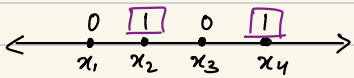
We can choose the interval $[x_1, x_2]$.

0 1 1 0 →



We can choose the interval $[x_2, x_3]$.

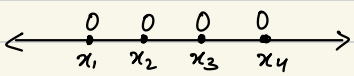
0 1 0 1 →



We can choose the interval $[x_2, x_2] \cup [x_4, x_4]$.

Whereas, for $x_1 = x_2 = x_3 = x_4 = 0$

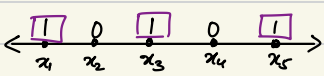
0 0 0 0 →



We can choose the interval $[x_1, x_4]$.

If we consider the same for 5 points instead:

1 0 1 0 1 →



We would require atleast 3 disjoint sets.
Therefore, the VC dimension of the given hypothesis is 4.

Q3) In the lecture, the pinball loss for $\alpha \in (0,1)$ and the Huber loss were introduced. Now we want to
construct a new loss that is a combination of the two. Its general form with $\delta, \epsilon > 0$ is shown below.
The loss should have the following properties:
quadratic in $[-\delta, \epsilon]$, like the Huber loss
linear with slope $-(1-\alpha)$ left of $-\delta$, like the pinball loss
linear with slope α right of ϵ, like the pinball loss
continuous and dierentiable everywhere
Derive the formula of the new loss depending on α.

A> when, $r \in [-\delta, \epsilon]$

it is quadratic due to Huber loss:

$$L_\alpha(r) = \frac{1}{2} r^2$$

when, $r < -\delta$

it should be linear with slope $-(1-\alpha)$, like the pinball loss:

at $r = -\delta$: $\quad L_\alpha(-\delta) = \frac{1}{2}\delta^2$

$$\Rightarrow L_\alpha(r) = -(1-\alpha)(r+\delta) + \frac{1}{2}\delta^2 \; ; \; r < -\delta$$

when, $r > \epsilon$

it should be linear with slope α:

$$L_\alpha(\epsilon) = \frac{1}{2}\epsilon^2$$

$$\Rightarrow L_\alpha(r) = \alpha(r-\epsilon) + \frac{1}{2}\epsilon^2 \; ; \; r > \epsilon$$

Hence, combined function:

$$L_\alpha(r) = \begin{cases} -(1-\alpha)(r+\delta) + \frac{1}{2}\delta^2 & ; \; r < -\delta \\ \frac{1}{2} r^2 & ; \; -\delta \leq r \leq \epsilon \\ \alpha(r-\epsilon) + \frac{1}{2}\epsilon^2 & ; \; r > \epsilon \end{cases}$$