APTS: Statistical Machine Learning Suggested Assessment

You can download these questions in PDF if preferred. This assessment is light-touch and intended to reinforce understanding of a few select concepts from the course. You should speak to your supervisor about appropriate questions given the goals of your learning and your research interests.

NOTE: any undefined notation in these solutions is notation that was standardised on in the course notes, so please reference those for any clarifications required.

Q1. Uncertain prediction

There may be situations where it is helpful to say *"I'm unsure"* rather than choosing a particular response label in classification problems. Consider a binary classification problem (so $\mathcal{Y} = \{0, 1\}$), but where we now permit predicting one of three outcomes, $\{0, 1, u\}$, with u representing "unsure". We extend the 0-1 loss with a fixed loss, c, for failing to make a decision:

$$\mathcal{L}(y, \hat{y}) := \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{if } y \neq \hat{y} \text{ and } \hat{y} \neq u \\ c & \text{if } \hat{y} = u \end{cases}$$

Determine the Bayes predictor and the reasonable range of values which make sense for c.

Q2. Cross-validation

Consider the toy binary classification problem where Y is independent of any X, with $\mathbb{P}(Y = 1) = \frac{1}{2}$. The empirical approximation of the Bayes predictor under 0-1 loss is to predict the majority class in the training set.

a. Prove that in this particular setting, when the training set size n is even, leave-one-out (LOO) cross validation is an unbiased estimator of the Bayes error, $\mathcal{E}^\star = \frac{1}{2}$. *Hint:* in this instance we are taking an expectation over training sets, $\mathbb{E}_{D_n}\left[\widehat{\text{Err}}_{\text{loo}}\right]$.

b. Of course, even though LOO is unbiased in this case, particular realisations of training set may result in quite different error estimates. What is the LOO estimate of the 0-1 loss when the training set contains exactly half $y_i = 0$ and half $y_i = 1$?

c. Different types of cross-validation can exhibit quite different behaviour. Perform an experiment in R for this toy setting which produces Monte Carlo simulations of training sets of size $n = 100$. For each Monte Carlo simulated training set produce LOO and 2-fold cross validation estimates of the 0-1 loss using each training set, as well as producing a hold-out estimate of the 0-1 loss using a test set (ie simulate an additional iid test set of size n).

   Examine the mean loss and standard deviation across simulations. Is LOO or 2-fold a lower variance estimator? Finally, produce box plots using the individual Monte Carlo simulated 0-1 loss estimates to observe the substantial difference in behaviour between LOO and 2-fold even in this simple case.

Q3. Simulation problem

Consider a binary classification problem, $\mathcal{Y} = \{0, 1\}$, with feature space $\mathcal{X} = \mathbb{R}^3_+$. Assume that π_{XY} is such that π_X is independently and identically Exponentially distributed in each dimension (rate $\lambda = 1$), whilst

$\pi_{Y \mid X}$ is Bernoulli distributed:

$$\mathbb{P}(Y = 1 \mid X = \mathbf{x}) = \begin{cases} 1 & \text{if } x_1 + x_2 + x_3 < 4 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

That is, given we know all components of $\mathbf{x}$ there is no uncertainty in y. Then clearly, under 0-1 loss,

$$g^\star(\mathbf{x}) = \begin{cases} 1 & \text{if } x_1 + x_2 + x_3 < 4 \\ 0 & \text{otherwise} \end{cases}$$

and $\mathcal{E}^\star = 0$.

a. We know $X_i \sim \text{Exp}(\lambda = 1), i \in \{1, 2, 3\}$, but it transpires that we cannot observe X_3. Therefore, we only observe (X_1, X_2) and must still predict Y. Prove that in this situation the Bayes predictor is

$$g^\star(\mathbf{x}) = \begin{cases} 1 & \text{if } x_1 + x_2 < 4 - \log 2 \\ 0 & \text{otherwise} \end{cases}$$

What is the Bayes error under 0-1 loss in this setting? *Hint:* You will find it useful to recall that the sum of two $\text{Exp}(1)$ random variables is a $\text{Gamma}(2, 1)$ random variable.

b. Similarly, compute the Bayes predictor and corresponding Bayes error under 0-1 loss when:

   i. only X_1 is observable. *Hint:* you may need to solve the condition for the Bayes predictor numerically in this case;

   ii. no predictors are observable.

c. In reality, we would not know (3.1) or that $X_i \sim \text{Exp}(\lambda = 1), i \in \{1, 2, 3\}$, so we would actually fit a model (eg k-nearest neighbour, trees etc) to predict the response.

   i. Simulate the learning curves for a logistic regression, 1-nearest neighbour and 100 tree random forest model by repeatedly simulating training and testing data sets from the above model. Do so in the case where the models (a) see all predictors; and separately where (b) they see only (x_1, x_2). To ensure some smoothness in the learning curves repeat the simulation at least 10 times at each training sample size. *Hint:* use a logarithmic spacing for training set sizes from $n = 50$ to $n = 20000$, for example, like the following in R:

```
ceiling(exp(seq(log(50), log(20000), length.out = 20)))
```

```
##  [1]    50    69    94   129   177   242   332   455   624   855  1171  1605
## [13]  2200  3016  4134  5666  7766 10645 14591 20000
```

   You should find that logistic regression performs best and achieves the Bayes error in both cases (a)+(b). You should find the ordering of knn and random forests depends on how many predictors are available.

   ii. Logistic regression does so well above because the true model belongs to the class we are fitting. Repeat the experiment above, with the simple modification of adding a cubic term in X_2 to the true model:

$$\mathbb{P}(Y = 1 \mid X = \mathbf{x}) = \begin{cases} 1 & \text{if } x_1 + x_2 + x_3 + x_2^3 < 4 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

   You should find random forests now match the performance of logistic regression with 2 predictors, and that with a full set of predictors there is a particularly interesting dependence on training sample size as to which model dominates.

Q4. Applied problem

If your PhD involves a problem for which you can identify a prediction task, try utilising the methods of the course to evaluate the predictive performance you can achieve in a supervised learning setting. Ensure that you think deeply about the problem and create a robust pipeline. Also use the opportunity to compare approaches in the course. For example,

- What loss functions are appropriate to the task?
- Do local methods and tree methods have very different performance?
- How do in-sample and out of sample error estimation procedures differ?
- In classification tasks, do different models have different calibration properties and can you successfully correct any that are miscalibrated?
- Does a super learner out-perform any constituent model?
- Are your results reproducible? Have you used `tidymodels`/`mlr3`?