School of Computing and Information Systems
The University of Melbourne
COMP30027 MACHINE LEARNING (Semester 1, 2019)

Tutorial sample solutions: Week 10

1. What is the difference between "model bias" and "model variance"?

   - In the context of evaluation:
     - Model bias is the propensity of a classifier to systematically produce the same errors; if it doesn't produce errors, it is unbiased; if it produces different kinds of errors on different training sets, it is also unbiased. (An example of the latter: the instance is truly of class A, but sometimes the system calls it B and sometimes the system calls it C.)
     - Note that this makes more sense in a **regression** context, where we can sensibly measure the difference between the prediction and the true value. In a classification context, these can only be "same" or "different".
     - Consequently, a typical interpretation of bias in a classification context is whether the classifier labels the test data in such a way that the **distribution** of predicted classes systematically doesn't match the distribution of actual classes. You might like to this about why this will necessarily mean that the system produces the same errors systematically.
   - Systematic variance is the propensity of a classifier to systematically produce different classifications. It is a measure of the inconsistency of the classifier, from training set to training set.

   (a) Why is a high bias, low variance classifier undesirable?

   - In short, because it's consistently wrong. (Or, using the other interpretation: the distribution of labels predicted by the classifier is consistently different to the distribution of the true labels; this means that it must be making mistakes.) The fact that it's consistent is little consolation here, unless we have some way of leveraging that information (for example, in some ensemble methods).

   (b) Why is a low bias, high variance classifier (usually) undesirable?

   - This is less obvious – it's low bias, so that it must be making a bunch of correct decisions. The fact that it's high variance means that not all of the predictions can possibly be correct (or it would be low-variance!) — and the correct predictions will change, perhaps drastically, as we change the training data.
   - One obvious problem here is that it's difficult to be certain about the performance of the classifier at all: we might estimate its error rate to be low on one set of data, and high on another set of data.
   - The real issue becomes more obvious when we consider the alternative formulation: the low bias means that the distribution of predictions matches the distribution of true labels; however, the high variance means that *which* instances are getting assigned to which label must be changing every time.
   - This suggests the real problem — namely, that what we have is the second kind of unbiased classifier: one that makes different kinds of errors on different training sets, but always errors; and not the first kind: one that is usually correct.

2. Describe how validation/development set, and cross-validation can help reduce overfitting?

   Machine learning models usually have one or more (hyper)parameters that control for model complexity, the ability of model to fit noise in the training set. In a practical application, we need to determine the values of such parameters, and the principal objective in doing so is usually to achieve the best predictive performance on new data. Furthermore, as well as finding the appropriate values for complexity parameters within a given model, we may wish to consider a range of different types of model in order to find the best one for our particular application. We

know that the performance on `training data` is not a good indicator of perdictive performance on unseen data because of overfitting.

If data is plentiful, then one approach is simply to use some of the available data to train a range of models, or a given model with a range of values for its complexity parameters, and then to compare them on independent data, sometimes called a `validation set`, and select the one having the best predictive performance. If the model design is iterated many times using a lim- ited size data set, then some overfitting to the validation data can occur and so it may be necessary to keep aside a third `test set` on which the performance of the selected model is finally evaluated.

In many applications, however, the supply of data for training and testing will be limited, and in order to build good models, we wish to use as much of the available data as possible for training. However, if the validation set is small, it will give a relatively noisy estimate of predictive performance. One solution to this dilemma is to use `cross-validation`.

3. Why `ensembling` reduces model variance?

   We know from statistics that `averaging` reduces variance. If $Z_1, \ldots, Z_n$ are i.i.d random variables:

   $$Var(\frac{1}{N}\sum_i Z_i) = \frac{1}{N}Var(Z_i)$$

   So the idea is that if several models are averaged, the model variance decreases without having an effect on bias. The problem is that there is only one training set, so how do we get multiple models? The answer to this problem is ensembling. Ensembling creates multiple models by creating multiple training set from one training set (e.g. bagging, random forrests), or by training multiple learning algorithms (e.g. stacking). The predictions of the individual are then combined (averaged) to reduce final model variance.