School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 1, 2023)

Sample solutions: Week 7

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

   Model Bias:

   - 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 instances, 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.)
   - The notation of bias is slightly more natural 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. For example, "bias towards the majority class", when the model predicts too many instances as the majority class.

   Model variance is the tendency of a classifier to produce different classifications if it was trained on different training sets (randomly sampled from the same population). It is a measure of the inconsistency of the classifier between different training sets.

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

   In short, because it’s consistently wrong. Or, more specifically in the context of classification: the labels predicted by the classifier is consistently different to the true labels; this means that it must be making mistakes.

   (ii). 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. Between “Model Bias” and “Model Variance”, which one is more harmful for the

performance on test set than training set? Why?

The “model variance” is high when different randomly sampled training sets lead to very different predictions on the test set. The high variance indicates that the model overfits to the training set. In this case, the training error may decrease, but test error will increase. A model with high bias would show a bad performance on both train and test sets.

3. During the training process, your model shows significantly different performance across different training sets. (a) What can be the reason? (b) How can we solve the issue?

Based on the definition of “model variance” when a model performance changes significantly by small changes in the training set the model is “overfitted” or has “high variance”.

There are a few remedies to reduce variance of a model. Overfitting happens when your model is too complex it means that reducing the complexity of the model can improve the model variance. We can reduce the complexity of a model by reducing the number of features in the model or regularizing the features values. We can also reduce the overfitting problem by increasing the number of training instances, or the number of features. You may also aim to reduce the noise in our training set, although a certain “irreducible noise” is inherent inherent in all ML problems.

4. You are developing a model to detect an extremely contagious disease. Your data consists of 4000 patients, out of which 100 are diagnosed with this illness. You achieve 96% classification accuracy.

(i). Can you trust the outcome of your model? Explain why.

No, in this scenario, achieving 96% classification accuracy is not sufficient to trust the outcome of the model. This is because of the class imbalance in the data. Out of the 4000 patients, only 100 have the disease, which means that the majority class (patients without the disease) heavily outweighs the minority class (patients with the disease).

In such a scenario, a model that simply predicts the majority class for all instances will achieve an accuracy of 3900/4000=97.5%, even though it does not detect any cases of the disease. Therefore, your model performs worse than a majority class classifier. High accuracy can be deceiving in highly unbalanced data sets and is not enough to trust the model's performance.

(ii). What type of error is most important in this task?

In the task of detecting an extremely contagious disease, the most important type of error is the false negative (FN) error, also known as a type II error or a miss. This error occurs when the model predicts that a patient does not have the disease when they actually do.

False negatives are particularly important in this task because failing to detect a case of the disease can have serious consequences, including the spread of the disease to others and potentially fatal outcomes for the affected individual. Therefore, it is crucial to minimize false negatives in this task, even if it means increasing the false positive rate or decreasing overall accuracy.

(iii). Name at least one appropriate evaluation metric that you would choose to evaluate your model.

One appropriate evaluation metric that could be used to evaluate the performance of the model in detecting the contagious disease is recall (also known as sensitivity or true positive rate). Recall measures the proportion of actual positive cases (patients with the disease) that are correctly identified by the model as positive.

Recall = TP / TP + FN

Where TP=True Positive and FN=False Negative. We can see directly, that recall will decrease with increaseing false negative (FN) predictions,as the numerator will increase.

In this scenario, recall is a crucial metric because it directly measures the ability of the model to detect cases of the disease and minimize false negatives. A high recall indicates that the model is effective in detecting positive cases, while a low recall indicates that the model is missing positive cases and may need further refinement.

5. How does the choice of the `max_depth` hyperparameter for the stopping criterion affect the performance of decision trees?

The stopping criterion hyperparameters in a decision tree model determine when the algorithm should stop splitting the nodes and creating new branches. The choice of `max_depth` stopping criterion can have a significant impact on the performance and complexity of the model.

A high max_depth may lead to a larger tree, i.e., a more complex model. This generally results in higher performance on the training data. Especially on datasets with noise or outliers complex models tend to overfit the data (i.e., model the noise rather than general useful patterns). We will observe much lower performance on a test data set compared to the training data – a case of poor generalization. In contrast, a lower max_depth will lead to a simpler model with lower training accuracy but better generalization abilities. Selecting an appropriate stopping criterion is crucial in balancing the trade-off between model complexity and generalization performance.

6. How does the k value in k-NN algorithm affect the decision boundary between classes?

The k value in the k-NN (k-Nearest Neighbours) algorithm determines the number of nearest neighbours used to classify a data point. The decision boundary in k-NN is the boundary that separates the regions of different classes.

The decision boundary in k-NN is nonlinear and depends on the values of the k parameter and the distribution of the data. When k is small, the decision boundary tends to be more flexible and follows the contours of the training data closely, resulting in a high variance model that can overfit the training data. Conversely, when k is large, the decision boundary becomes smoother and less flexible, resulting in a higher bias model that may underfit the training data.

7. Explain the difference between “evaluation bias” and “model bias”.

Evaluation bias and model bias are related concepts, but they refer to different types of biases in the context of machine learning models.

Model bias refers to the systematic error in the modeling process that results in a model that is unable to capture the true underlying relationship between the input features and the target variable. Model bias can be caused by various factors, such as the choice of the model architecture, the selection of features, or the assumptions made by the model. A model with high bias is unable to fit the training data well and is likely to underfit, resulting in poor performance on both the training and test sets.

On the other hand, evaluation bias refers to the systematic error in the evaluation of a model that results in consistently overestimating or underestimating the true performance of the model. Evaluation bias can be caused by various factors, such as the choice of evaluation metric, the sampling bias in the data used for evaluation, or the inappropriate assumptions made by the evaluator. Evaluation bias can lead to a model that performs well on the training set but poorly on the test set, as the model has learned to optimize the biased evaluation metric rather than the true underlying task.

To summarize, model bias is a property of the model itself, while evaluation bias is a property of the evaluation process. Both biases can prevent us from developing generalizable models, but they require different strategies for correction. Model bias can be addressed by improving the modeling process, such as selecting more appropriate features or using a more complex model architecture. Evaluation bias can be addressed by selecting appropriate evaluation metrics and data sampling techniques that are less biased and more representative of the true performance of the model.