Solution to Examination: Programming for Data Science (ID2214)

Henrik Boström

Jan. 7, 2019

Part I (Theory, 10 points)

1a. Methodology, 2 points

The estimated accuracy of a model will vary depending on what set of instances is used to evaluate its performance. If we consider such a set of instances to be a random sample drawn from some (unknown) target distribution, the estimated accuracy of the model will sometimes be lower and sometimes higher than the true accuracy (the accuracy of the model wrt. the target distribution). If the compared models have similar true accuracies, then the observed differences in the estimated accuracies will mainly be due to the sample we have drawn, and hence the extreme values of the observed estimated accuracies will be biased; the lowest value will be overly pessimistic and the highest value overly optimistic, and the difference between the extreme values and the true accuracies will increase with the number of compared models. Hence, if we select the highest of these values as an estimate for the true accuracy of the best model, we will systematically be overestimating the performance. However, if the true accuracy of one of the models is much higher than for the others (and hence would almost always be outperforming the others independently of the sample used), then this bias will be smaller. Since we typically do not beforehand know what the true accuracies are, there is hence a high risk that the estimated accuracy is indeed too high.

1b. Data preparation, 2 points

One-hot-encoding results in that there will be one new (binary) feature for each possible value of a categorical feature, and a categorical value is represented by assigning all the values for the new features to zero, except for the one that corresponds to the categorical value, which is assigned to one.

There is no need handle missing values prior to employing one-hot-encoding, i.e., imputation or removal of missing values are not necessary beforehand. When creating the new features, missing values may either be ignored, i.e., no features are generated that correspond to the missing values, or represented by binary features, i.e., indicating whether the categorical value is missing or not. If the first option is employed, then a missing categorical value may be represented by setting all values for the new features to zero, while for the

second option, all values are set to zero except for the feature that represents missingness.

1c. Naïve Bayes, 2 points

When calculating $P(c1|f1 = 0\&f2 = 1)$ using naïve Bayes, the class prior $P(c1)$ is multiplied with $P(f1 = 0|c1)$ and $P(f2 = 1|c1)$, and divided by $P(f1 = 0\&f2 = 1)$. Since $P(f1 = 0|c1) = 0$, it means that according to naïve Bayes, $P(c1|f1 = 0\&f2 = 1) = 0$.

When calculating $P(c2|f1 = 0\&f2 = 1)$ using naïve Bayes, the class prior $P(c2)$ is multiplied with $P(f1 = 0|c2)$ and $P(f2 = 1|c2)$, and divided by $P(f1 = 0\&f2 = 1)$. Since $P(f2 = 1|c2) = 1-P(f2 = 0|c2) = 0$, it means that according to naïve Bayes, $P(c2|f1 = 0\&f2 = 1) = 0$.

In other words, naïve Bayes would assign a probability of zero for both class labels, and hence both labels would maximize the expression.

1d. Performance metrics, 2 points

Assume a classification task with the class labels + and -, and that we have two positive (+) instances p1, p2 and two negative (-) instances n1, n2.

If M predicts the following probabilities for the class +:

p1: 0.4, p2: 0.3, n1: 0.2, n2: 0.1

then the accuracy is 2/4 = 0.5 (assuming that the label with the highest probability is predicted), while the AUC is 1/2*(2/2)+1/2*(2/2) = 1.0

1e. Clustering, 2 points

Relative strengths of agglomerative clustering include:

- the output is more informative, corresponding to varying numbers of non-overlapping clusters
- the output is derived deterministically (no need for repeated runs)
- the number of clusters does not need to be specified

Relative strengths of k-means clustering include:

- typically faster execution
- will produce the desired number of clusters (if known)

Part II (Programming, 20 points)

2a. Data preparation, 10 points

```
import numpy as np

def random_projection(word_lists,rand_proj,dim):
   p = len(word_lists)
   result = np.empty((p,dim))
   for i in range(p):
      paragraph = word_lists[i]
      vec = np.zeros(dim)
      for word in paragraph:
         vec += rand_proj[word]
      result[i] = vec
   return result
```

2b. Combining models, 10 points

```
import numpy as np

def decision_forest(df,min_leaf,no_of_trees,no_of_features):
   forest = np.empty(no_of_trees,dtype="object")
   features = list(df.columns)
   if "CLASS" in features:
      target = "CLASS"
   elif "REGRESSION" in features:
      target = "REGRESSION"
   features.remove(target)
   orig_index = df.index
   no_instances = df.shape[0]
   for i in range(no_of_trees):
      bootstrap_index = np.random.choice(orig_index,no_instances,replace=True)
      feature_sample = np.random.choice(features,no_of_features,replace=False)
      mod_df = df.loc[bootstrap_index,np.append(feature_sample,target)]
      forest[i] = decision_tree(mod_df,min_leaf)
   return forest
```