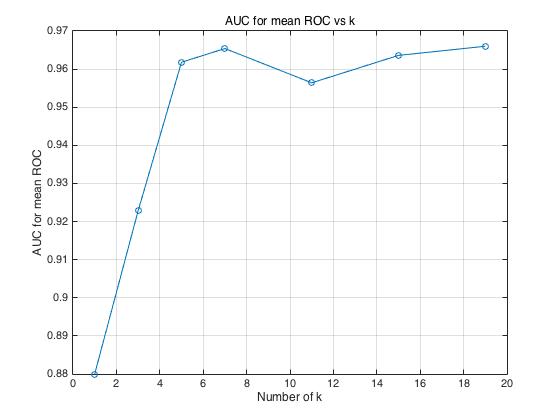
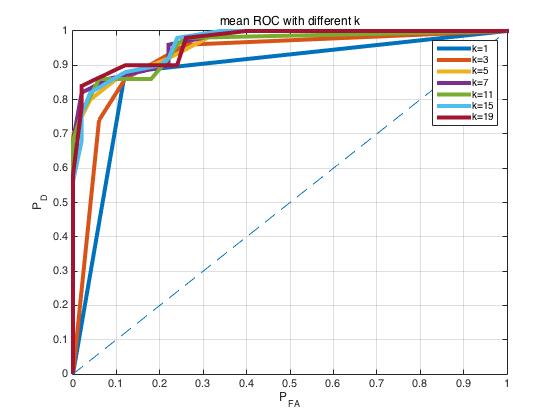
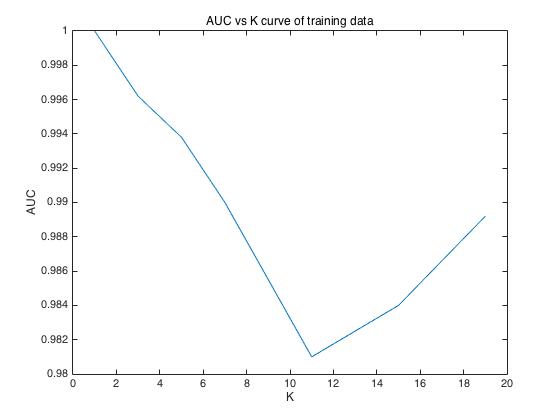
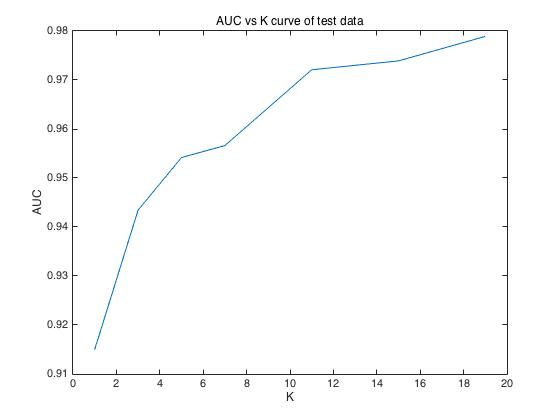
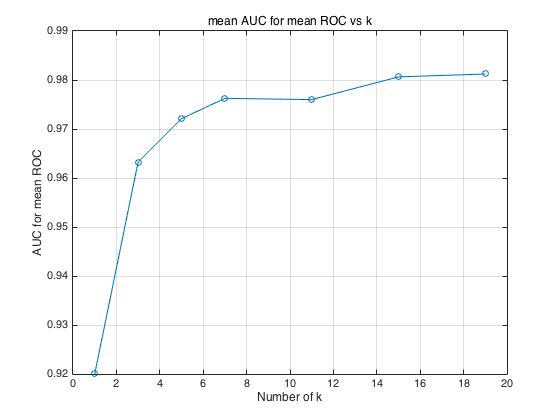
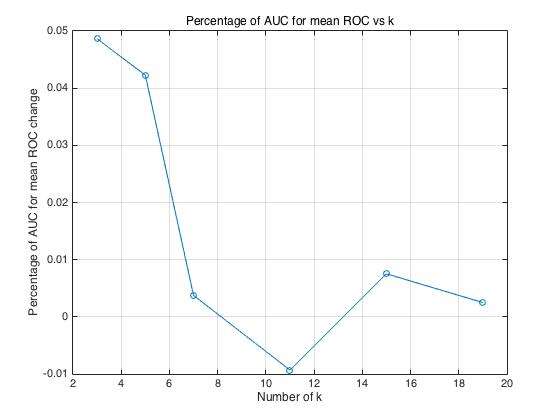
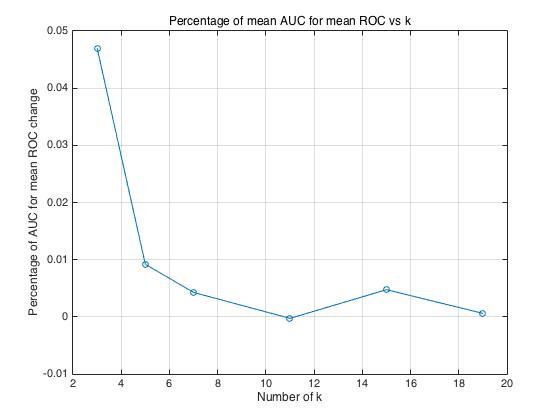
**ECE681 HW1 Report**

SHENGXIN QIAN

**Cross-Validation**

Cross-validated ROC curves and AUC for the mean ROCs vs k curves are shown at the top. As we can see, when k=1, the performance of KNN is much lower than the other k choice because it is overfitting as expected. When choosing other k, the performance is similar.

Comparing cross-validated AUC curve with training data AUC curve, we can see they have opposite trend and the performance of latter is higher than the former. The cross-validated AUC increases a lot at first with increasing k and has a plateau. The training data AUC generally decreases with increasing k at first and increases after k=11. The difference is because the classifier highly tuned to the training data which lead to the high performance with low k. With increasing k, the overtraining decreases before k=11. After k=11, the trend is more consistent with cross-validated AUC curve.

 Comparing cross-validated AUC curve with testing data AUC curve, we can see the trend is similar but still has some difference. The plateau of cross-validated AUC curve is flatter than testing data AUC curve. The increasing speed of testing data AUC is lower in the range before plateau. This is because, without cross-validation, the classifier highly tuned to the training data and the overfitting problem with low k make it worse. When we compare the value of mean cross-validated AUC with the value of testing data AUC, we can see the performance of former is a little bit higher than the latter with each k. This is also because the classifier may highly tune to training data and cross-validation would mitigate the effect. The cross validation result would be more close to the ideal result with a typical data set.

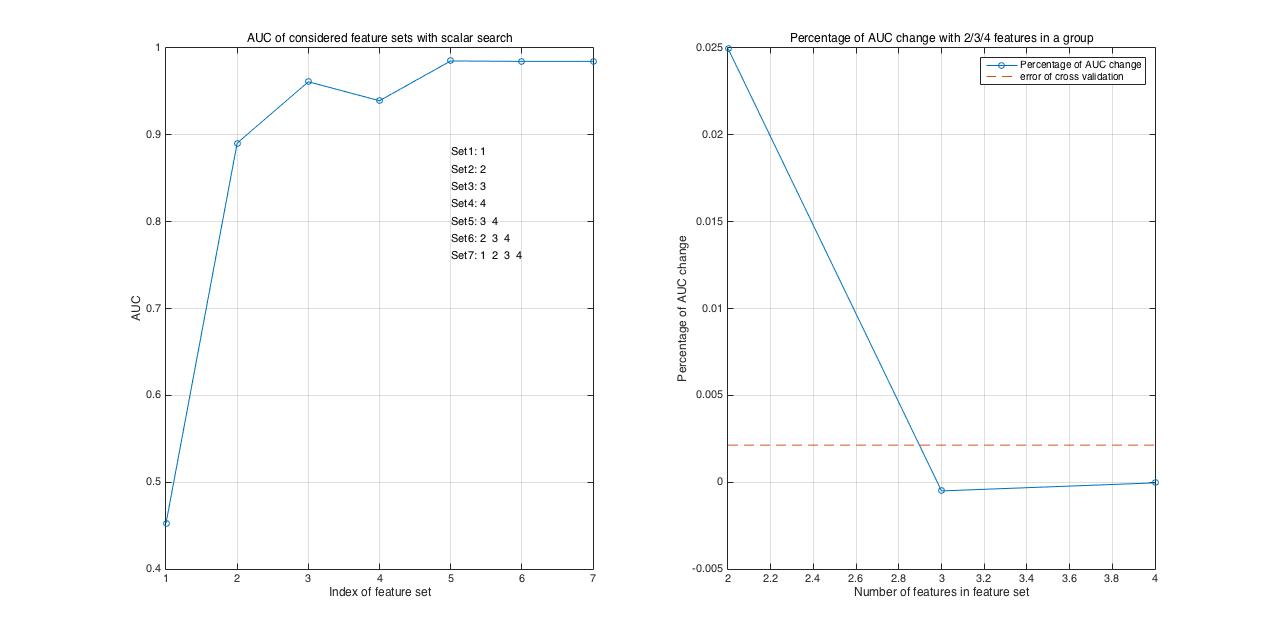
If we want to choose k depending on cross validation result, we could look at two pictures above. They are both the percentage of AUC change with each k. The left is the curve of one realization. The right shows the mean AUC change. (20 realization result). As we can see in the picture on the left, when we choose k=3 and k=5, we could get more than 4% performance increment. We could only get less than 0.6% performance increment with k>=7. So, if we want to choose the proper k depending on the result of one realization, we should choose k=5. When k>=7, the performance increment is minimum but the computational complexity is much higher. When k<5, the predicted performance is lower than k=5.

If we consider about k depending on the mean AUC, the k=5 or k=3 are both good choice because their performance different is 1%. If we need better-predicted performance, we would choose 5. If we need less computational complexity, we would choose 3. Same reason as above why we would not choose k>=7 and k<=3. I would like to choose k=5 which would bring better AUC.

**Feature Selection**

We have 4 features in our training data set.

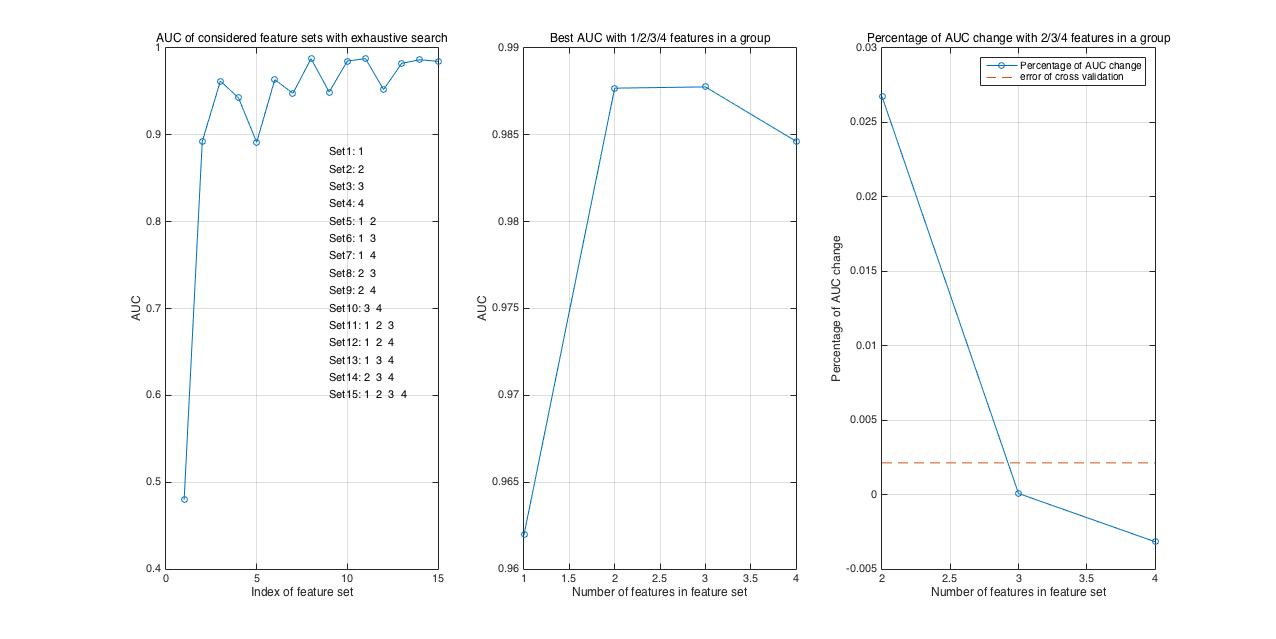
* 1 feature :
* 2 feature :
* 3 feature :
* 4 feature :



1. **Scalar forward search**

Two pictures above are the curve of the considered feature sets vs feature set index and the curve of the percentage of AUC change with 2/3/4 features in a set. If we choose the best feature set depending on the absolute performance, we would choose feature set {1,2,3,4} which has highest AUC for mean ROC. However, the cross validation itself also has error because we randomly choose 10 folders in a data set. The red line of dashes shows the relative error of cross validation. The AUC change less than the relative error of cross validation is meaningless which means when we choose 2 features in a set, the AUC increment is meaningful. More than that, the computational cost of 3/4 features in a set is much higher. I would choose 2 features in a set. The best 2 feature set with scalar search is {3,4}.

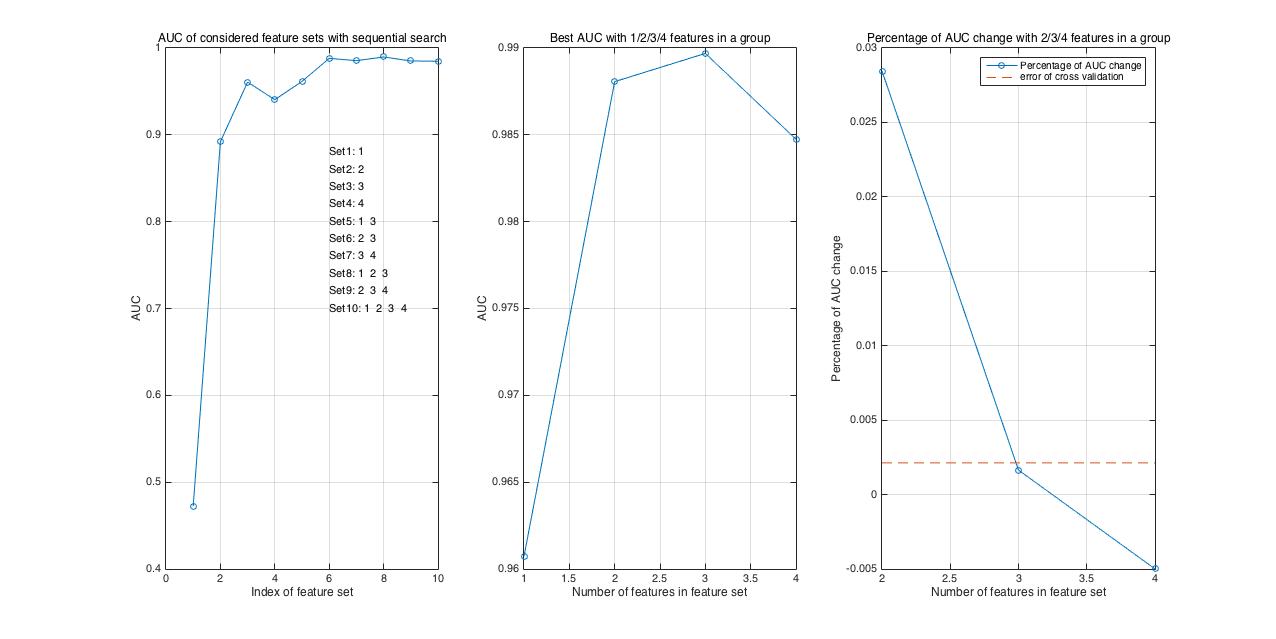
This result makes sense because feature 3 has largest mean value difference and second lowest variance. Even though feature 4 has the largest variance but it also has largest mean value difference. AUC of feature 2 is lower than feature 3 because its mean value difference is lower than feature 3. The feature 1 has lowest AUC because the distribution is same in two classes and there is no discriminability between two classes with feature 1. Larger mean value difference represents higher center difference and lower variance represents the distribution is closer to center. That is why scalar search would only search {1,2,3,4,34,234,1234} these 7 feature sets.



1. **Exhaustive search**

As we can see, the left picture is the AUC for the feature sets considered including all 15 possible sets. The middle picture is the best AUC with 1/2/3/4 features in a set. The right picture is the relative AUC change of the middle picture. The single feature AUCs are the same as the result of the scalar search.

As we can see in the middle picture, best 2 feature set has the same AUC for mean ROC as the best 3 feature set. Because the 3 feature set has higher computational complexity, so the best one in 2 feature set is the best choice. The best 2 feature set is {2,3}. The result is different from that in scalar search because we would miss this set in scalar search.



1. **Sequential search**

As we can see, when we do the sequential search, even the number considered feature sets is lower than exhaustive search, it still covers the best 2/3 feature set. However, its complexity is lower than exhaustive search. The result makes sense. When we choose between 2/3 feature set (they both has highest AUC for mean ROC), as we can see, the relative increment between best 2/3 feature set is less than the error of cross validation which means the difference is meaningless. So I would choose the feature set {2,3} with lower computational complexity.

1. **Blind test**

For the blind test, I would choose feature set {2,3}. The reason was talked in exhaustive search analysis and sequential search analysis.