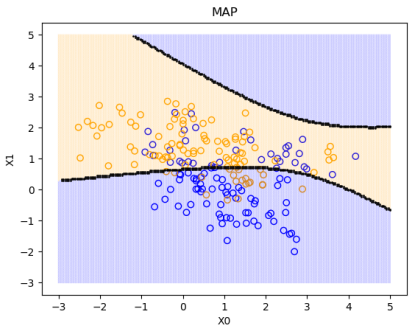
CSCI 737 Patter recognition A1

Satwik Mishra, sm5036

February 13, 2020

1. Results of MAP: Here we classify to the most probable class based on the conditional probability distribution density of the classes where the posterior probaility is given by $P(w_k/z) = \frac{p(z/w_k)P(w_k)}{p(z)}$, where $P(w_k)$ is the prior probability, that can be pre computed based based upon training sample data we have. If we don't know the true distribution of the data, then we have to make certain assumption. Depending on this assumption, we can compute the posterior probability. So it is similar to a linear classifier in the sense that we are making an assumption in terms of the distribution of the feature space. On For faster compution and as(monotonicity is maintained since log is also monotonically increasing).

(a) Visualization of Training Data



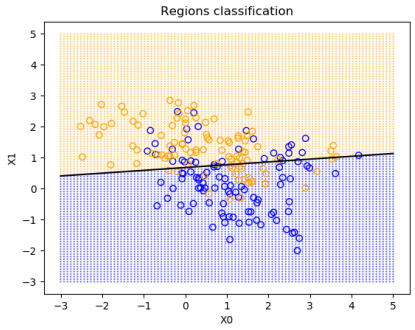
(b) Decision boundary for MAP

```
Accuracy: 71.5
          Decision
Truth     [64 36]
          [21 79]
```

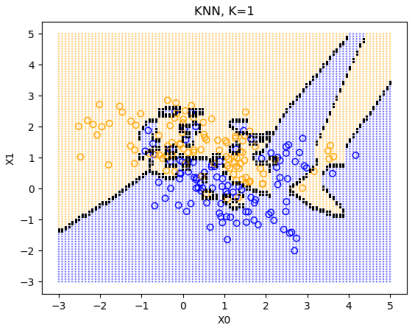
(c) Confusion matrix for MAP

Figure 1: Results for MAP Classifier

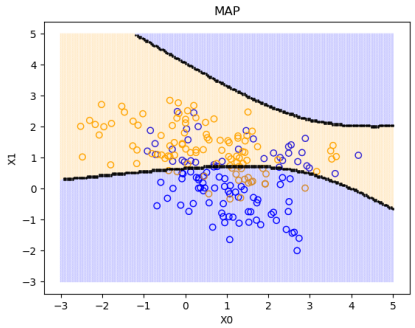
(a) **KNN vs Linear classifier (Least Squares) vs MAP**



(a) Decision boundary for Linear classifier



(b) Decision boundary for KNN classifier, K=1



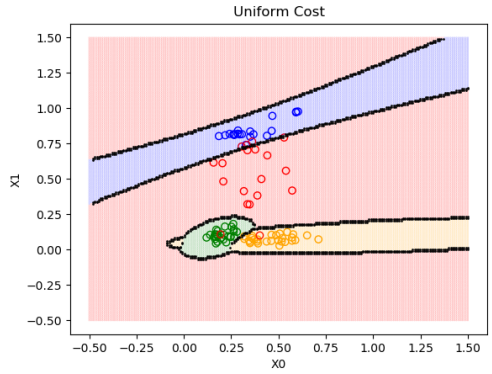
(c) Decision boundary for MAP

Figure 2: KNN vs Linear (Squared sum error) vs MAP Classifier

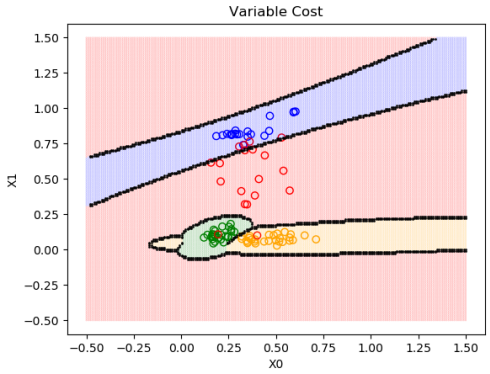
i. KNN has low bias, as we don't have any prior assumption about the type of the data. However it has high variance because a small change in the training data in a sub region can greatly impact the decision boundary hence it has high variance. Hence as we can see in the graphs, for k=1 the decision curve is not smooth and is affected by few pints in a sub region.Similarity with bayes is that KNN, somewhat approximates the same solution as that of a bayes classifier, but it does so by relaxing the conditional probability in the local sub cluster of K-neighbors.

ii. Linear Least Squares classifier: Here we do have a prior assumption that the data is linearly separable and there fore we try to fit a hyperplane as a decision boundary. This gives us a smoother decision curve compared to that of a KNN. As it is not getting affected by small changes in the training input space hence has low variance but high bias as we are heavily dependent on the underlying assumption of linearly separable feature space.

2. Bayesian classifier



(a) Bayesian classifier with 0/1 Cost function



(b) Bayesian classifier with Variable Cost function

```
        Decision
Truth   [19  0  0  1]
        [ 0 28  0  0]
        [ 0  0 27  0]
        [ 1  1  1 16]

Rate of correct classification 95.74468085106383
```

(c) Confusion matrix with 0/1 Cost function

```
        Decision
Truth   [20  0  0  0]
        [ 0 28  0  0]
        [ 0  0 27  0]
        [ 3  1  1 14]

Rate of correct classification 94.68085106382979
```

(d) Confusion matrix with Variable Cost function

Figure 3: Classification of bolt, nut, ring and scrap (confusion matrix follows the same order from left to right and top to bottom)

(a) As the data we are working with, the covairance matrix for each class tells us that the features are not statistically independent, hence the values of the features (x1, x2) where P(x) = f(x1, x2), such the contours of P(x) will be elipses or an elipsoid in the multidimensional space. The decison boundary obtained is somewhat parabolic which reflects the fact that our underlying assumption regarding the distribution of the data is that it is gaussian, given by , $p(x/w_i) = \frac{e^{(-\frac{1}{2}*(x-\mu_i)^T(C_i)^{-1}(x-\mu_i))}}{(2\pi|C_i|)^{\frac{1}{2}}}$ and the exponent term is quadratic.

3. On increasing the prior probability of the scrap class, as shown in the figure , without adjusting the cost matrix, points which were previously classified as Bolts are now getting classified as scraps. Now in order to maintain the behaviour of the algorithm when the prior probability was not doubled and to get the same boundaries I increased the cost associated with assigning a point to scraps when the actual state of the point is bolts from 0.03 to 0.7. And on doing so, I got the similar looking plots for the decision boundary and the same confusion matrix as Figure 3(d) and Figure 4(b).

```
        Decision
Truth   [19  0  0  1]
        [ 0 28  0  0]
        [ 0  0 27  0]
        [ 0  1  1 17]

Rate of correct classification 96.80851063829788
```

(a) Prior probability doubled for Scraps, keeping the variable cost same as before

```
        Decision
Truth   [20  0  0  0]
        [ 0 28  0  0]
        [ 0  0 27  0]
        [ 3  1  1 14]

Rate of correct classification 94.68085106382979
```

(b) Prior probability doubled for Scraps, increasing the cost for assigning to scraps when the true state is nuts

Figure 4: Confusion matrix when the prior of Scraps is doubled)