Sharif University of Technology
department of computer engineering
ML for bioinformatics, March 2022

Hadis Ahmadian
400211524
Assignment #2

Question One

Part #1

$$p(k=1|a=1 \wedge b=1 \wedge c=0) = \frac{p(a=1 \wedge b=1 \wedge c=0|k=1)p(k=1)}{p(x)}$$

Because p(x) is constant for all classes, we can simply ignore it in calculating relative probability of classes to each other:

$$p(k=1|a=1 \wedge b=1 \wedge c=0) = p(a=1 \wedge b=1 \wedge c=0|k=1)p(k=1)$$

$$p(k=1) = \frac{4}{8} = \frac{1}{2}$$

Since we are using naive Bayes, we assume conditional independence exists $\implies$

$$p(a=1 \wedge b=1 \wedge c=0|k=1) = p(a=1|k=1) * p(b=1|k=1) * p(c=0|k=1) = \frac{2}{4} * \frac{1}{4} * \frac{2}{4} = \frac{1}{16}$$

$$\implies p(k=1|a=1 \wedge b=1 \wedge c=0) = \frac{1}{32}$$

Part #2

2.1)

let L be the loss function,R be the risk function, f be the classifier and f^* be the optimal classifier:

$$L(y,f) = \begin{cases} \alpha & f(x)=1,\ y=0 \\ \beta & f(x)=0,\ y=1 \end{cases}$$

$$f^* = argmin\ R(f) = argmin\ E(L(y,f)) =$$

$$argmin\ p(y=0|x) * \alpha * l_{f(x)=1} + p(y=1|x) * \beta * l_{f(x)=0} =$$

$$argmin\ p(y=0|x) * \alpha * l_{f(x)=1} + p(y=1|x) * \beta * (1 - l_{f(x)=1}) =$$

$$argmin\ p(y=0|x) * \alpha * l_{f(x)=1} + p(y=1|x) * \beta - p(y=1|x) * \beta * l_{f(x)=1} =$$

$$argmin\ (p(y=0|x) * \alpha - p(y=1|x) * \beta) * l_{f(x)=1} + p(y=1|x) * \beta \implies$$

$$f^* = \begin{cases} C1 & if \quad p(y=0|x) * \alpha < p(y=1|x) * \beta \\ C0 & if \quad otherwise \end{cases}$$

* * * * *

2.2)

$$Priors: \quad p(C_1) = 1/2\ ,\ p(C_2) = 1/2$$

$$likelihoods: \quad p(x=0|C_1) = 1-p\ ,\ p(x=1|C_1) = p$$

$$p(x=0|C_0) = 1-q\ ,\ p(x=1|C_0) = q$$

$$I: p(C_1|x=1) = p(x=1|C_1) * p(C_1) = p * \frac{1}{2}$$

$$II: p(C_0|x=1) = p(x=1|C_0) * p(C_0) = q * \frac{1}{2}$$

$$p > q \implies I > II$$

$$I : p(C_1|x = 0) = p(x = 0|C_1) * p(C_1) = (1 - p) * \frac{1}{2}$$

$$II : p(C_0|x = 0) = p(x = 0|C_0) * p(C_0) = (1 - q) * \frac{1}{2}$$

$$p > q \implies II > I$$

$$\implies C(x) = \begin{cases} 0(C_0) & if \quad x = 0 \\ 1(C_1) & if \quad x = 1 \end{cases}$$

$$Risk = p(C(x) \neq Y) = p((C(x) = 0 \wedge Y = 1) \vee (C(x) = 1 \wedge Y = 0)) =$$

$$p(x = 0 \wedge Y = 1) + (x = 1 \wedge Y = 0) = \frac{1}{2}[(1 - p) + (q)]$$

$$* * * * *$$

2.3)
we can see in figure c(ج), x2 tends to increase with x1 and decrease as it decreases. This means that they are dependent on each other, meaning they have a noticeable covariance (high correlation). However in figure b(ب), this condition does not hold and x1 and x2 are not correlated. so in figure b we can use naive bayse because of conditional independence but in figure c, conditional independence does not hold.

.....

Part #3

SVM is not usually used as an online method of learning because in usual forms of the algorithm, it needs all of the data to operate. However if we want to use some variation of it, there will be two cases. In the case of hard margin, because SVM does not allow misclassifications, it won't be good with outliers. In the case of soft margin misclassifications are allowed so it will be better than hard margin but still because the margin has a limit and there will be a cost for misclassifications, it won't be very good with extreme outliers.

LOGISTIC REGRESSION is also not very good with outliers. The reason is that LR is trying to find a dependency between features so the slope of the regression line is affected by data with unusual features (outlier data). As a result LR won't be a good option for data with outlires and extreme noise, specially in higher dimensions.

NAIVE BAYSE, unlike previous methods, performs good with outliers. Naive bayes averages out the outlier data when it estimates the conditional probability and the cases that happen more often will have much more weight, also another reasons is that there is an assumption of independence of features in naive bayes.

■

Questions Two .....

Part #1:

Figures 4 and 3 are both linear so one of them is for classifier number 1 and the other, for classifier number 2. The difference between these two classifiers is their cost (parameter C). Higher C means more cost for misclassification, so less support vectors. Classifier 2 has higher cost, so less support vectors (and smaller margin) so figure 3 is for classifier 2 and figure 4 is for classifier 1.

Figure 1 and 6 both have RBF kernel so one of them is for classifier number 4 and the other, for classifier number 5. The coefficients($\frac{-1}{4}$ and -4) of classifiers were calculated as $\frac{1}{-2\sigma^2}$ so σ for classifier 4 is $\sqrt{8}$ and σ for classifier 5 is $\sqrt{\frac{1}{8}}$. The bigger the σ, the less support vectors we'll have, so classifier 5 is for figure 6 and classifier 4 is for figure 1.

decision boundary of a quadratic kernel would be either a hyperbolic or a ellipse. so classifier 3 is for figure 5.

this leaves us with figure 2 to be for none of the classifiers.

.....

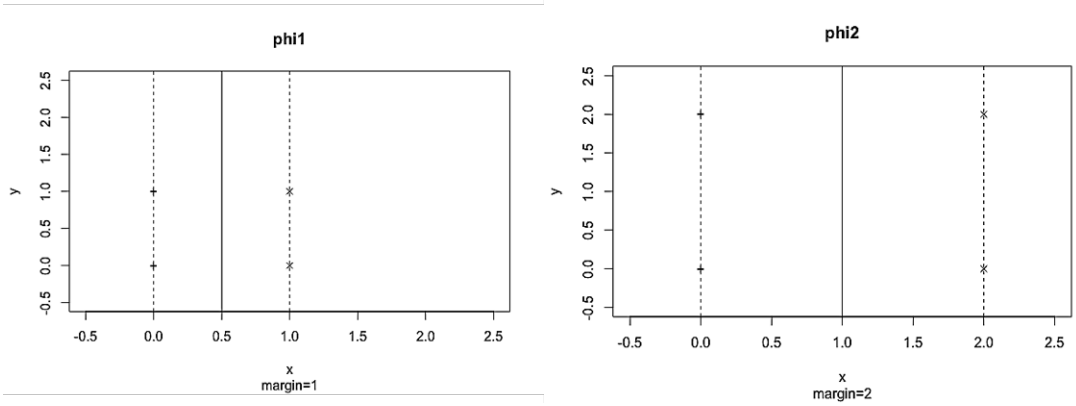
Part #2:

2.1) False: output of SVM is a hyper-plane (in a possibly higher dimension than features, depended on kernel) that is the decision boundary in that dimension.

2.2) False: different kernels, map data to different dimensions for classification, so the hyperplane and so the support vectors will be different. Also different kernels have different parameters and even changing parameters of a fixed kernel will change the support vectors, so for different kernels, the support vectors won't be the same.

2.3) True : in example below, we can see even though two kernels do the same thing (and will have same function on data), have different margins (because ϕ_2 multiplies each coordinate by 2 relative to ϕ_1) :

$$\phi_1(x, y) = (x, y) \; and \; \phi_2(x) = (2x, 2y)$$



(example from: https://stats.stackexchange.com/questions/144272/two-margin-comparison-and-one-conclusion)

■

Questions Three .....

The main problem of a simple decision tree is over-fitting to data and as a result, less generalizability. When trees in a random forest have less correlation, it means they have more randomness and that trees focus on different aspects of the data set and that the model is not just memorizing the training set. This decreases the over-fitting and so the accuracy on test set will increase.

■
