STA 561 HW4 (SVM & Kernels)

Daniel Truver

2/28/2018

(1) Constructing Kernels

Let K_1 be a kernel.

(a)

$aK_1(x,z)$ is a kernel only if $a > 0$.

$$aK_1(x,z) = a\langle \Phi(x), \Phi(z)\rangle = \langle \sqrt{a}\Phi(x), \sqrt{a}\Phi(z)\rangle$$

If $a < 0$, we will get

$$\int_{X\times X} aK_1(x,z)f(x)f(z)\ dxdz = a\int_{X\times X} K_1(x,z)f(x)f(z)\ dxdz < 0,$$

which is a problem.

(b)

The following is not a kernel.

$$K(x,z) = \langle x,z\rangle^3 + (\langle x,z\rangle - 1)^2 = \langle x,z\rangle^3 + \langle x,z\rangle^2 - 2\langle x,z\rangle + 1$$

We can show that it violates the Cauchy-Schwarz Inequality. Take $x = 0$, and z such that

$$||z||_2^2 \in (0,1).$$

Then

$$\begin{aligned} K(x,z) &= 1 \\ K(x,x) &= 1 \quad \text{(already a problem)} \\ K(z,z) &= (||z||_2^2)^3 + (||z||_2^2)^2 - 2||z||_2^2 + 1 \\ &= r < 1 \qquad \text{(since } ||z||^2 \in (0,1)) \end{aligned}$$

Now we have

$$\sqrt{K(x,x)K(z,z)} = \sqrt{r} < 1 = K(x,z),$$

which is a violation of the Cauchy-Schwarz Inequality. Therefore, K is not a valid kernel.

(c)

We first note that

$$k(x,z) = \langle x,z\rangle^2 = \langle x,z\rangle\langle x,z\rangle$$

is a valid kernel since it is the product of two valid kernels. We know this fact as a result of Mercer's Theorem.

$$\begin{aligned} k_1(x,z) &= \sum_{j=1}^{\infty} \psi_j(x)\psi_j(z) \quad \text{(the $\psi's$ have eaten the $\lambda's$)} \\ k_2(x,z) &= \sum_{i=1}^{\infty} \Psi_i(x)\Psi_i(z) \\ k_1(x,z)k_2(x,z) &= \sum_{j=1}^{\infty}\sum_{i=1}^{\infty} \psi_j(x)\Psi_i(x)\psi_j(z)\Psi_i(z) \end{aligned}$$

Take

$$\phi_{(i,j)}(x) = \psi_j(x)\Psi_i(x).$$

We can construct a feature map $\Phi(x)$ with features $\phi_{(i,j)}(x)$.

Then, we have our defined feature map, so

$$k_1(x,z)k_2(x,z) = \sum_{(i,j)\in\mathbb{N}\times\mathbb{N}} \phi_{(i,j)}(x)\phi_{(i,j)}(z) = \langle \Phi(x), \Phi(z)\rangle$$

is a kernel. Similarly for the finite case.

We now play the same game with

$$\exp(-||x||^2)\exp(-||z||^2).$$

If we take $g(x) = \exp(-||x||^2)$, then the above is the product $g(x)g(z)$ which is a kernel with feature map $\Phi = g$.

So,

$$K(x,z) = \langle x,z\rangle^2 + \exp(-||x||^2)\exp(-||z||^2)$$

is a valid kernel

(2) RKHS

We first note that $X = [0,1] \subset \mathbb{R}$ is compact because it is closed and bounded. We next need to show that $\mathcal{F}$ is a Hilbert space. That is, we need it to be a complete innner product space. We'll go with the standard L_2 inner product. To show that it is complete, consider a Cauchy sequence of functions $\{f_n(x)\} = \{a_n x\}$ from this space.

$$\forall\epsilon > 0,\ \exists N : \forall m > n \geq N$$
$$\left(\int_0^1 |f_m(x) - f_n(x)|^2 dx\right)^{1/2} < \frac{\epsilon}{\sqrt{3}}$$

The $\sqrt{3}$ will come up later.

We want to show that $\lim\{f_n\} \in \mathcal{F}$.

$$\begin{aligned} \left(\int_0^1 |f_m(x) - f_n(x)|^2 dx\right)^{1/2} &= \left(\int_0^1 |a_m x - a_n x|^2 dx\right)^{1/2} \\ &= \left(\int_0^1 |a_m - a_n|^2 x^2 dx\right)^{1/2} \\ &= \frac{1}{\sqrt{3}}|a_m - a_n| < \frac{\epsilon}{\sqrt{3}} \\ \implies |a_m - a_n| < \epsilon \end{aligned}$$

This gives us that $\{a_n\}$ is a Cauchy sequence of real numbers, which we know converges to some $a \in \mathbb{R}$. So we are suspicious that $a_n x \to ax$.

We can verify this. Given $\epsilon > 0$, $\exists N$ s.t.

$$\forall n \geq N : |a - a_n| < \epsilon,$$

since we know a_n is Cauchy. Therefore

$$\begin{aligned}\left(\int_0^1 |f_n(x) - f(x)|^2 dx\right)^{1/2} &= \left(\int_0^1 |a_n - a|^2 x^2 dx\right)^{1/2} \\ &= \frac{1}{\sqrt{3}}|a_n - a| < \epsilon,\end{aligned}$$

and we can say that

$$a_n x \to ax : f_n \to f \in \mathcal{F}.$$

We conclude that $\mathcal{F}$ is complete and a bonafide Hilbert space. Now, onward to the reproducing property.

First, we will show that $\forall f \in \mathcal{F}$, we have

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) = \sum_{i=1}^{m} \alpha_i (\cdot x_i)$$

This seems fairly simple. Take $\alpha_1 = 1, x_i = a \in \mathbb{R}$, and obtain $f(x) = k(x, a) = ax$.

We now need $\langle k(\cdot, x), f\rangle = f(x)\ \forall f \in \mathcal{F}$ Again, this seems alright.

$$\langle k(\cdot, x), f\rangle = \langle k(\cdot, x), k(\cdot, a)\rangle = k(x, a) = ax = f(x)$$

(3) Convexity and KKT Conditions

(a) Lagrangian and Dual Form

We first rewrite the constraints so that that appears as $g_i(w, \eta, \eta^*) < 0$

$$\min_{w,\eta,\eta^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\eta_i + \eta_i^*)$$

subject to

$$\begin{aligned} y_i - w^T x_i - \epsilon - \eta_i &\leq 0 \\ w^T x_i - y_i - \epsilon - \eta_i^* &\leq 0 \\ -\eta_i &\leq 0 \\ -\eta_i^* &\leq 0 \\ i &= 1, \ldots, n \end{aligned}$$

In total, we have $m = 4n$ constraints of the form $g_j(w, \eta, \eta^*) \leq 0$. In the following lagrangian, consider α to be a vector in $4n$ dimensions, with the first n corresponding to the first constraint, the second n correspoding to the second constraint, and so on. We write the lagrangian as

$$\begin{aligned}\mathcal{L}(w, \eta, \eta^*, \alpha) &= \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\eta_i + \eta_i^*) + \sum_{j=1}^{m}\alpha_j g_j(w, \eta, \eta^*) \\ &= \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\eta_i + \eta_i^*) \\ &+ \sum_{i=1}^{n}\alpha_i[y_i - w^T x_i - \epsilon - \eta_i] + \alpha_{i+n}[w^T x_i - y_i - \epsilon - \eta_i^*] + \alpha_{i+2n}[-\eta_i] + \alpha_{i+3n}[-\eta_i^*]\end{aligned}$$

We now have this painful Lagrangian from which to derive the dual form. Let's begin with the implications of Lagrangian stationarity.

$$\begin{aligned}
\nabla_w \mathcal{L}(w, \eta, \eta^*, \alpha) &= 0 \\
w + \sum_{i=1}^{n} [-\alpha_i x_i + \alpha_{i+n} x_i] &= 0 \implies w = \sum_{i=1}^{n} [\alpha_i x_i - \alpha_{i+n} x_i] \\
\nabla_\eta \mathcal{L}(w, \eta, \eta^*, \alpha) &= 0 \implies C - \alpha_i - \alpha_{i+2n} = 0 \\
\nabla_{\eta^*} \mathcal{L}(w, \eta, \eta^*, \alpha) &= 0 \implies C - \alpha_{i+n} - \alpha_{i+3n} = 0
\end{aligned}$$

From the other KKT conditions, we get

$$\alpha_j \geq 0 \ \forall j \alpha_j g_j(w, \eta, \eta^*) = 0 \ \forall j g_j(w, \eta, \eta^*) \leq 0$$

Now let's expand that Lagrangian and hope for the best.

$$\begin{aligned}
\mathcal{L}(w, \eta, \eta^*, \alpha) &= \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} (\eta_i + \eta_i^*) \\
&+ \sum_{i=1}^{n} \alpha_i [y_i - w^T x_i - \epsilon - \eta_i] + \alpha_{i+n}[w^T x_i - y_i - \epsilon - \eta_i^*] + \alpha_{i+2n}[-\eta_i] + \alpha_{i+3n}[-\eta_i^*] \\
&= \frac{1}{2}||w||^2 + \sum_{i=1}^{n} C\eta_i + \sum_{i=1}^{n} C\eta^* \\
&+ \sum_{i=1}^{n} \alpha_i y_i - w^T \sum_{i=1}^{n} \alpha_i x_i - \sum_{i=1}^{n} \alpha_i \epsilon - \sum_{i=1}^{n} \alpha_i \eta_i \\
&+ w^T \sum_{i=1}^{n} \alpha_{i+n} x_i - \sum_{i=1}^{n} \alpha_{i+n} y_i - \sum_{i=1}^{n} \alpha_{i+n} \epsilon - \sum_{i=1}^{n} \alpha_{i+n} \eta^* \\
&- \sum_{i=1}^{n} \alpha_{i+2n} \eta_i - \sum_{i=1}^{n} \alpha_{i+3n} \eta_i^* \\
&= \frac{1}{2}||w||^2 + \cancel{\sum_{i=1}^{n} (C - \alpha_i - \alpha_{i+2n})\eta_i} + \cancel{\sum_{i=1}^{n} (C - \alpha_{i+n} - \alpha_{i+3n})\eta_i^*} \\
&+ \sum_{i=1}^{n} \alpha_i y_i - w^T \sum_{i=1}^{n} \alpha_i x_i - \sum_{i=1}^{n} \alpha_i \epsilon \\
&+ w^T \sum_{i=1}^{n} \alpha_{i+n} x_i - \sum_{i=1}^{n} \alpha_{i+n} y_i - \sum_{i=1}^{n} \alpha_{i+n} \epsilon \\
&= \frac{1}{2} w^T w - w^T \sum_{i=1}^{n} \alpha_i x_i + w^T \sum_{i=1}^{n} \alpha_{i+n} x_i \\
&+ \sum_{i=1}^{n} y_i (\alpha_i - \alpha_{i+n}) - \sum_{i=1}^{n} \epsilon(\alpha_i + \alpha_{i+n}) \\
&= -\frac{1}{2} w^T w + \sum_{i=1}^{n} y_i (\alpha_i - \alpha_{i+n}) - \sum_{i=1}^{n} \epsilon(\alpha_i + \alpha_{i+n}) \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} (\alpha_i - \alpha_{i+n})(\alpha_k - \alpha_{k+n}) x_i^T x_k + \sum_{i=1}^{n} y_i (\alpha_i - \alpha_{i+n}) - \sum_{i=1}^{n} \epsilon(\alpha_i + \alpha_{i+n}) \\
&= \mathcal{L}(\alpha)
\end{aligned}$$

To my surprise, it seems this only depends on the first $2n$ of the α_j's correspoding to the first two constraints. Perhaps we could have ignored the non-negativity constraints on the η's, but that feels morally wrong, and I don't have a clear enough understanding of what the η's are doing to ignore them.

So, we get the dual form:

$$\max_{\alpha} \mathcal{L}(\alpha)$$

(b) Support Vectors

Let's take a look at our complementary slackness conditions.

$$\alpha_i^*[y_i - w^T x_i - \epsilon - \eta_i] = 0 \implies \begin{cases} \alpha_i^* > 0 & \implies y_i - w^T x_i = \epsilon + \eta_i \\ y_i - w^T x_i - \epsilon - \eta_i < 0 & \implies \alpha_i^* = 0 \end{cases}$$

$$\alpha_i^*[w^T x_i - y_i - \epsilon - \eta_i^*] = 0 \implies \begin{cases} \alpha_i^* > 0 & \implies w^T x_i - y_i = \epsilon + \eta_i^* \\ w^T x_i - y_i - \epsilon - \eta_i^* < 0 & \implies \alpha_i^* = 0 \end{cases}$$

Putting these together, and examining the $\alpha_i^* > 0$ case, we suspect that the support vectors are

$$\{x_i : |y_i - w^T x_i| = \epsilon + \eta\}.$$

I admit some uncertainty about the role of the η's. Since, as far I can tell, they are positive numbers constrained only by C in the objective functions, we conclude that:

$$\{\text{support vectors}\} \subseteq \{x_i : |y_i - w^T x_i| \geq \epsilon\}$$

(c) Influence of Epsilon

We suspect that increasing ϵ will make our model less likely to overfit. We are willing to accept more errors in the training set. On the other hand, forcing ϵ to zero will force us to fit more closely to the data. If our suspicion about the identities of the support vectors from (b) is true, we will have more support vectors as ϵ approaches 0. That is, more and more points from our dataset will influence the decision boundary, which makes us more likely to overfit.

At this point, I finally get why it's called the epsilon-insensitive loss.

(d) Influence of C

I am still uncertain about the role of the η's. The only place I can clearly see their effects is in the constraints. As C gets larger, it seems the η's will get smaller. When we look at the original constraints of the problem,

$$y_i - w^T x_i - \epsilon \leq \eta_i$$
$$w^T x_i - y_i - \epsilon \leq \eta_i^*$$

we see the sending the η's to 0 will force the difference between y and f closer to ϵ. That is, we will have to use a function f that fits more closely to the data, making it more likely that we overfit.

(e) Test Points

From the KKT condition of Lagrangian Stationarity, we have an expression for w.

$$w = \sum_{i=1}^{n} (\alpha_i - \alpha_{i+n}) x_i$$

To calculate the value of f at a new point, x_{new} we use

$$w^T x_{new} = \sum_{i=1}^{n} (\alpha_i - \alpha_{i+n}) x_i^T x_{new}.$$

Were we to employ the kernel trick, we would replace the dot product with $k(x_i, x_{new})$.

(4) SVM Implementation

Oh dear, here we go.

Special thanks to https://gist.github.com/rwalk/64f1365f7a2470c498a4 for explaining how `quadprog` and `kernlab` work.

(a) Hard SVM

We have the credit card data here just to see if the function works.

```r
library(quadprog)
train = function(X, y, eps = 1e-5){
  # preparing for QP solver
  Q = sapply(1:nrow(X), function(i){ y[i] * t(X)[,i] })
  D = t(Q) %*% Q
  d = matrix(1, nrow = nrow(X))
  b0 = rbind( matrix(0, nrow = 1, ncol = 1) , matrix(0, nrow = nrow(X), ncol = 1) )
  A = t(rbind(matrix(y, nrow = 1, ncol=nrow(X)), diag(nrow = nrow(X))))
  # QP solver
  sol = solve.QP(D + eps*diag(nrow(X)), d, A, b0, meq = 1)
  alpha = sol$solution
  comp = (alpha * y[,1]) * X # computes summand of lambda formula
  lambda = apply(comp, 2, sum) # apply sum over the rows to get lambda
  return(lambda)
}
predictHard = function(X_new, lambda){ #call this `predictHard` or it masks default `predict`
  return(sign(X_new %*% lambda))
}
```

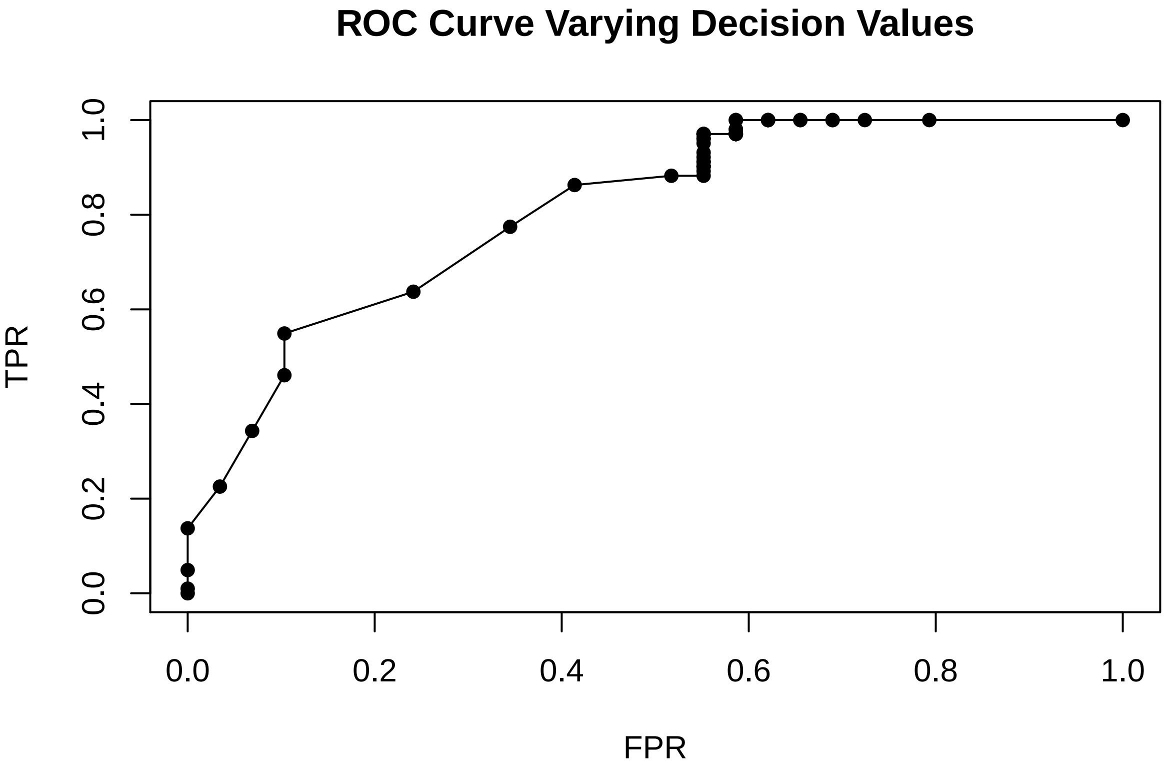
(b) Credit Card Data

```r
creditCard = read.csv("creditCard.csv") %>%
  mutate(Class = factor(ifelse(Class == 1, 1, -1)))
n_train = ceiling(0.9*nrow(creditCard))
set.seed(2018)
trainIndex = sample(1:nrow(creditCard), n_train)
train.df = creditCard[trainIndex,]
test.df = creditCard[-trainIndex,]
X_train = as.matrix(train.df %>% select(-Class))
y_train = as.matrix(train.df$Class)
X_test = as.matrix(test.df %>% select(-Class))
y_test = as.matrix(test.df$Class)
```

We will construct ROC curves by changing the cost parameter within the model.

```r
library(e1071)
firstSVM = svm(Class~., data = train.df, kernel = "linear", decision.values = TRUE,
               probability = TRUE)
firstPred = predict(firstSVM, newdata = test.df, decision.values = TRUE, probability = TRUE)
truth = test.df$Class
accuracy = mean(firstPred == truth)
numpoints = 50
TPR = rep(NA, numpoints)
FPR = rep(NA, numpoints)
i = 1
```

```
for (c in seq(0,1, length.out = numpoints)){
  decision.values = attributes(firstPred)$probabilities
  newDecision = rep(0, length(firstPred))
  newDecision[decision.values[,1] > c] = 1
  predictions = sign(newDecision - .5)
  tpr = sum(predictions == 1 & predictions == truth)/sum(truth == 1)
  fpr = sum(predictions == 1 & predictions != truth)/sum(truth == -1)
  TPR[i] = tpr
  FPR[i] = fpr
  i = i + 1
}
{
  plot(FPR, TPR, pch = 16, xlim = c(0,1), main = "ROC Curve Varying Decision Values")
  lines(FPR, TPR)
}
```



```
AUCapprox = 0
for (i in 2:numpoints){
  AUCapprox = AUCapprox + (TPR[i-1] + TPR[i])/2 * (-FPR[i]+FPR[i-1])
}
```

Table 1: Some Interesting Values of our SVM

Accuracy	AUC
0.855	0.809

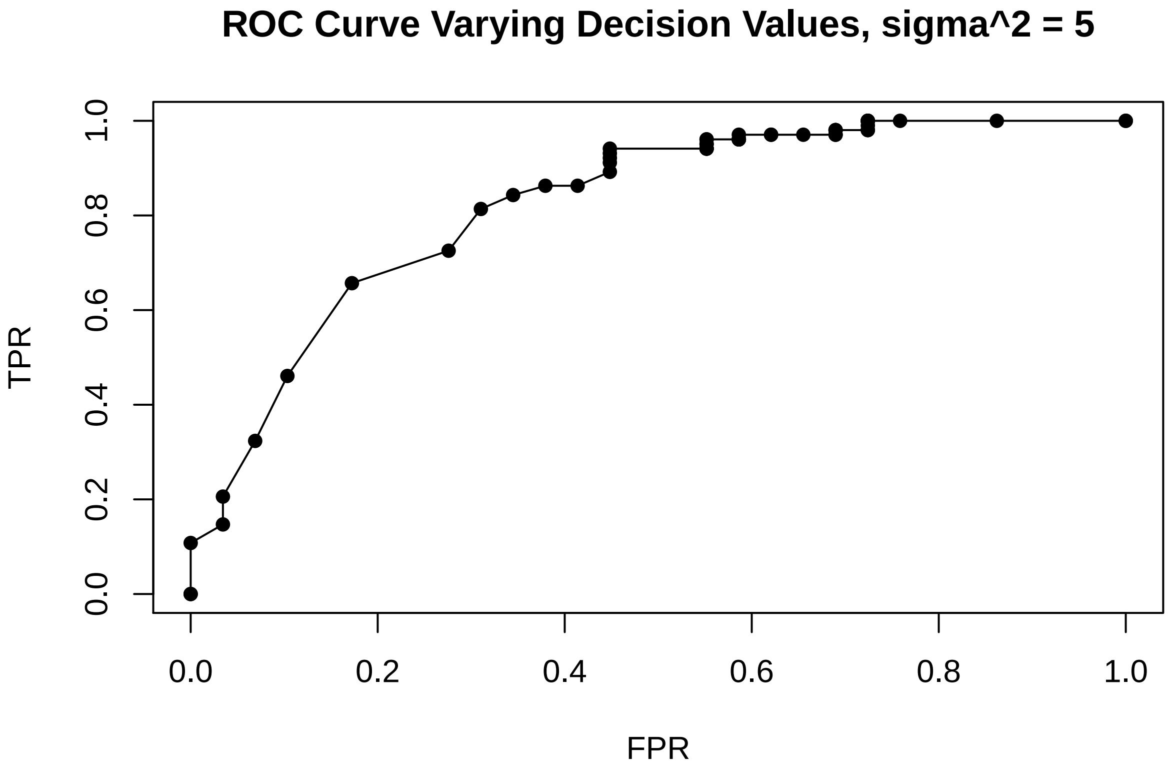
(c) SVM with Radial Basis Kernel

```r
radSVM.5 = svm(Class ~., data = train.df, kernel = "radial", gamma = 1/5, probability = TRUE)
radPred.5 = predict(radSVM.5, newdata = test.df, probability =  TRUE)
accuracy.5 = mean(radPred.5 == truth)
radSVM.25 = svm(Class ~., data = train.df, kernel = "radial", gamma = 1/25, probability = TRUE)
radPred.25 = predict(radSVM.25, newdata = test.df, probability = TRUE)
accuracy.25 = mean(radPred.25 == truth)
```

Table 2: Accuracy Comparison of Kernels

Linear Kernel	Radial Basis Sigma^2 = 5	Radial Basis Sigma^2 = 25
0.855	0.847	0.855

Well, that’s disappointing. We now construct the ROC curves as done above.



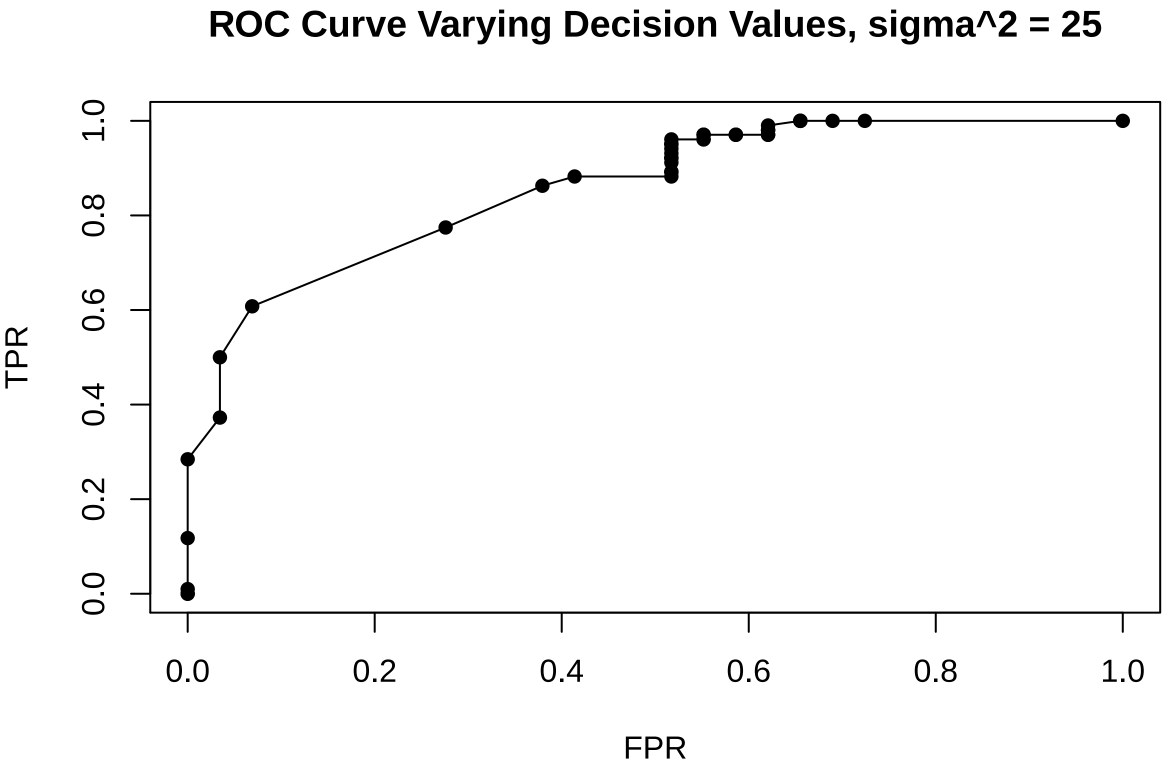


Table 3: Comparison of AUC between Kernels

AUC Linear Kernel	AUC Radial Kernel, sigmaˆ2 = 5	AUC Radial Kernel, sigmaˆ2 = 25
0.809	0.822	0.859