James Simmons
Machine Learning
Homework 3
Fall 2021

Task 1:

Q1: What is the margin and support vectors?

The support vectors are the data points closest to the hyperplane that separates the classes; they can equivalently be described as the data points closest to the data points of a different class. The margin is the distance between the hyperplane in the support vectors.

Q2: How does SVM deal with non-separable data?

SVM deals with non-separable data by mapping the data onto a higher-dimentional space such that it becomes separable.

Q3: What is a kernel?

A kernel function is a function that defines a high-dimensional space and measures the correlation or distance between two data points in it. Regardless of the dimensionality of the space, it typically outputs a scalar value that can be very easily compared; this makes it very efficient.

Q4: How does a kernel relate to feaure vectors?

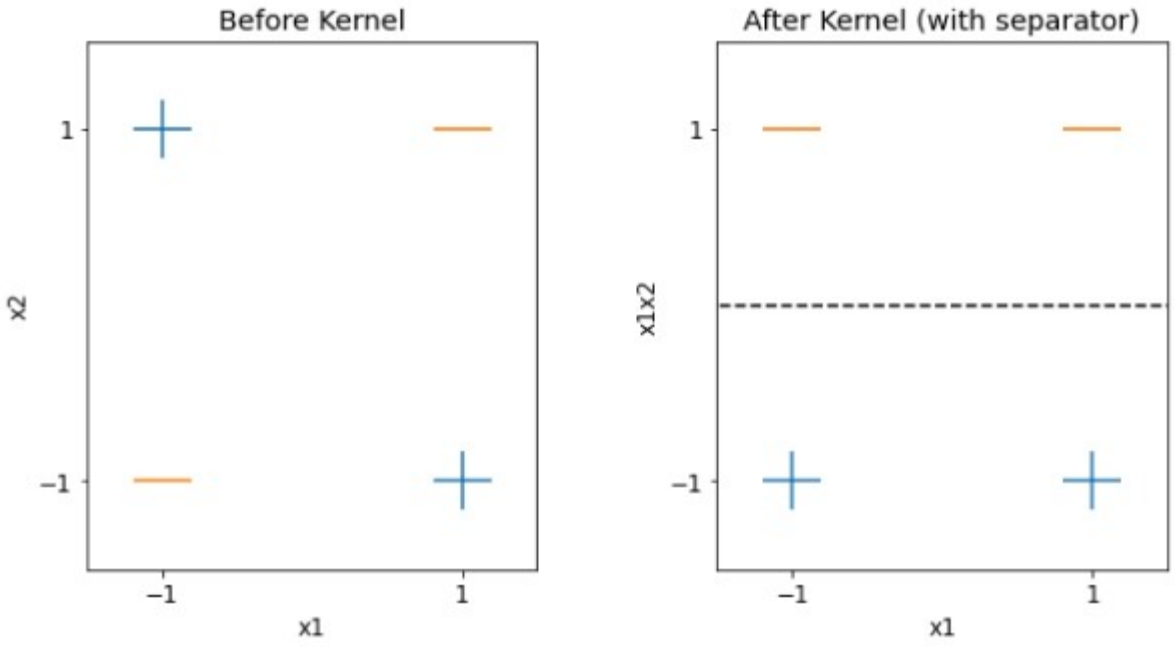
A feature vector is a vector in a higher-dimensional space known as the feature space. For instance, a vector (x1, x2) could be raised to a 3-dimensional space by mapping it to the feature vector (x1, x2, x1x2). If we wish to compare two vectors in this higher-dimensional space, one way is to perform these mappings and then perform a series of vector operations, but this is costly.

A kernel is a function that takes two vectors in a lower-dimensional space and returns a scalar value that represents a relation between the vectors that would exist had they been mapped into a higher-dimensional space, but without actually requiring that we do the full mapping process, increasing efficiency.

A kernel relates to feature vectors by allowing us to compare two feature vectors without actually calculating the vectors themselves.

Task 2:

Construct a support vector machine that computes the kernel function. Use four values of +1 and -1 for both inputs and outputs. Map the input [x1, x2] into a space consisting of x1 and x1x2. Draw the four input points in this space, and the maximal margin separator. What is the margin?



The size of the margin is 1, because the shortest distance between any point on the separator and any datapoint is 1.

Task 3:

Show that every circular region is linerarly separable from the rest of the 2D plane in the feature space (x_1, x_2, x_1^2, x_2^2).

A circular region R is defined by a, b, and r, such that a point (x1, x2) is in R if and only if:

$$(x_1-a)^2+(x_2-b)^2-r^2\leq 0$$
$$(x_1^2-2a\,x_1+a^2)+(x_2^2-2b\,x_2+b^2)-r^2\leq 0$$
$$-2a\,x_1-2b\,x_2+x_1^2+x_2^2+a^2+b^2+r^2\leq 0$$

Thus, given a vector V in the feature space (x_1, x_2, x_1^2, x_2^2), we conclude V is in R if and only if: V · (-2a, -2b, 1, 1) + a^2 + b^2 + r^2 ≤ 0.

Thus, we define the hyperplane H to be the set of all vectors V such that the above inequality is precicely equal to 0. H linearly separates R from the rest of the 2D plane in the feature space.

Task 4:

Show that an SVM using the polynomial kernel of degree 2, $K(u, v) = (1 + u \cdot v)^2$, is equivalent to a linear SVM in the feature space (1, x1, x2, $x1^2$, $x2^2$, x1x2) and hence that SVMs with this kernel can separate any elliptic region from the rest of the plane

Let R be an elliptic region in the 2D plane defined by a, b, c, and d such that a point (x1, x2) is in R if and only if:

$$c(x_1-a)^2+d(x_2-b)^2-1\leq 0$$
$$c(x_1^2-2a x_1+a^2)+d(x_2^2-2b x_2+b^2)-1\leq 0$$
$$-1+a^2 c+b^2 d-2ac\, x_1-2b d\, x_2+c x_1^2+d x_2^2\leq 0$$
$$\textit{filler}$$

Thus, given a vector V in the feature space (1, x1, x2, $x1^2$, $x2^2$, x1x2), we conclude V is in R if and only if:

$V \cdot (-1 + a^2c + b^2d, -2ac, -2bd, c, d, 0) \leq 0$.

Thus, we define the hyperplane H to be the set of all vectors V such that the above inequality is precicely equal to 0. H linearly separates R from the rest of the 2D plane in the feature space.

Let u = (u1, u2) and v = (v1, v2) be vectors on the 2D plane.

$$K(u,v)=(1+u\cdot v)^2$$
$$(1+u_1v_1+u_2v_2)^2$$
$$1+u_1v_1+u_2v_2+u_1v_1+u_1^2v_1^2+u_1v_1u_2v_2+u_2v_2+u_1v_1u_2v_2+u_2^2v_2^2$$
$$1+2u_1v_1+2u_2v_2+2u_1v_1u_2v_2+u_1^2v_1^2+u_2^2v_2^2$$
$$(1,\sqrt{2}u_1,\sqrt{2}u_2,u_1^2,u_2^2,\sqrt{2}u_1u_2)\cdot(1,\sqrt{2}v_1,\sqrt{2}v_2,v_1^2,v_2^2,\sqrt{2}v_1v_2)$$

Therefore, K is equivalent to the dot product of u and v after they have been raised into the feature space (1, x1, x2, $x1^2$, $x2^2$, x1x2). We know from the hyperplane H defined above that such a feature space can be used separate any such elliptic region R from the rest of the 2D plane.

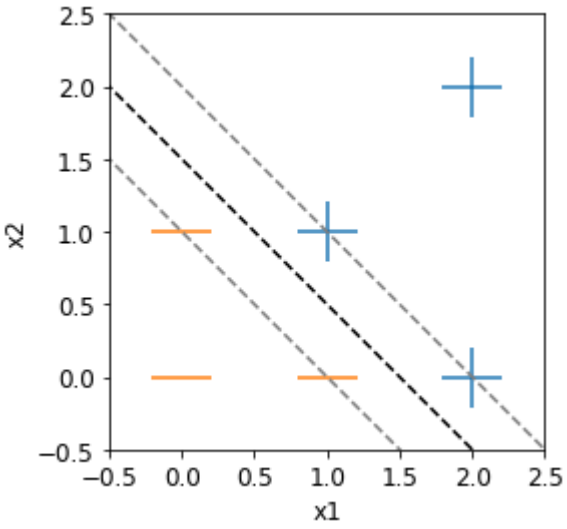
Task 5:

Consider the following training data

class	x_1	x_2
+	1	1
+	2	2
+	2	0
−	0	0
−	1	0
−	0	1

(a) Plot these six training points. Are the classes {+, -} linearly separable?

Yes, the classes are linearly separable.



Black: The center of the margin
Gray: The edges of the margin

(b) Construct the weight vector of the maximum margin hyperplane by inspection and identify the support vectors.

The support vectors are points:

(1, 1)
(2, 0)
(0, 1)
(0, 0)

The weight vector of the maximum margin hyperplane is (1, -1). The slope-intercept form of the hyperplane line is x2 = -x1 + 1.5.

Task 6:

Consider a dataset with 3 points in 1D

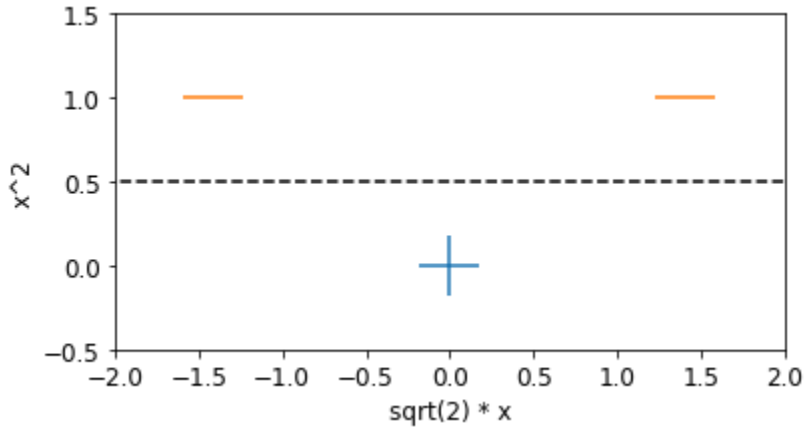
```
-1                  0                   1
|-----------------------------------------|
-                   +                   -
```

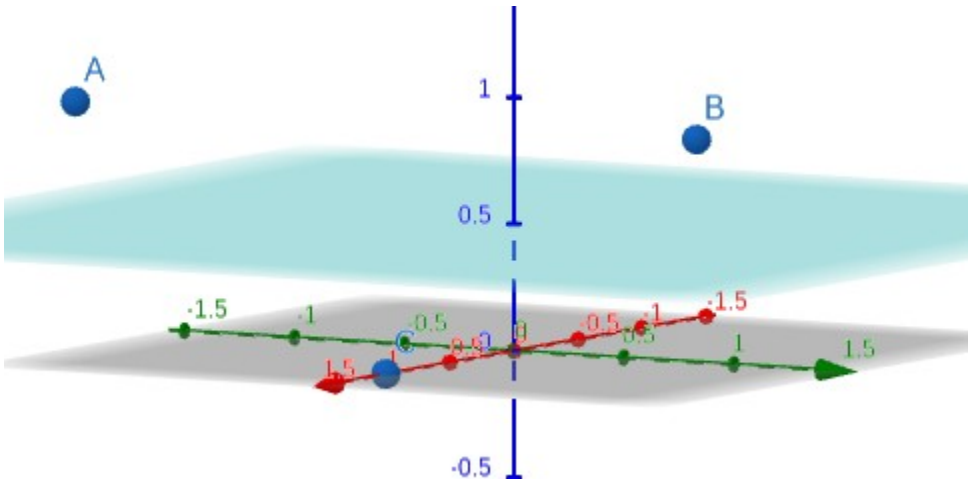
(a) Are the classes {+, -} linearly separable?

No.

(b) Consider mapping each point to 3D using new feature vectors φ(x) = [1, √2x, x^2]. Are the classes now linearly separable? If so, find a separating hyperplane.

Yes. Since all three data points will have the same value (1) in their first dimension, we can ignore that dimension for the purposes of finding a separating hyperplane. Below is a plot of the second and third dimensions ($\sqrt{2}x$, x^2), and a hyperplane that separates the classes ($x^2 = 0.5$). Also included is a screenshot from the 3D plot in geogebra.





Task 7:

Report the five-fold cross-validation classificaiton accruacies for linear, quadratic, and RBF kernels on the Titanic training set.

Similarly to in HW2, I selected which features to use by brute forcing every combination of features and finding the average cross-validation accuracy for each, since there were few enough for it to be practical.

The top 5 best combinations and their corresponding cross-validation accuracies for each of the kernels are listed below, along with the time it took to compute all 256 combinations. For the linear kernel, I excluded the Ticket field, since it made the system completely stall for some reason, so its time is only for the remaining 128 combinations (since none of the other kernels had Ticket in their top 5, it's unlikely it would be for linear, so I don't expect this to have made a meaningful difference). I also did it for a cubic filter (polynomial with degree 3) because I accidentially put in the wrong number for degree and figured I may as well keep the data.

Quadratic (11 seconds)

```
0.792: Sex, Embarked, SibSp
0.792: Sex, SibSp
0.791: Sex, Embarked, SibSp, Parch
0.790: Sex, SibSp, Parch
0.788: Sex, Embarked, Parch
```

Cubic (15 seconds)

```
0.793: Sex, SibSp
0.789: Sex, Embarked, SibSp
0.788: Sex, Parch
0.787: Sex, SibSp, Parch
0.787: Sex
```

RBF (17 seconds)

```
0.795: Pclass, Sex, Embarked, Parch
0.792: Sex, SibSp
0.791: Pclass, Sex, SibSp
0.790: Sex, SibSp, Parch
0.788: Pclass, Sex, Embarked
```

Linear (3 minutes, 41 seconds)

```
0.788: Sex, Embarked, Age, SibSp, Parch
0.788: Sex, SibSp, Parch
0.787: Sex, Embarked, SibSp, Parch
0.787: Sex, Fare
0.787: Sex, SibSp, Fare
```

Linear was the most consistent across the different feature selections, but also took significantly longer than the others for the worst results. The other three kernels executed in comparable times and had comparable accuracy distributions across their feature selections, with a general trend that the longer it took the better it did (in this case, time-to-compute is a pseudo stand-in for algorithmic complexity / sophistication), with RBF on top. For the most part, all kernels agreed on which feature selections were best, though the ordered varied.

Code

All code available at: https://github.com/Mesaj2000/Machine_Learning_5610/tree/master/hw3