# Question 8 (20 pts) -- COMPLETE

Create examples to explain the property and the importance of the following kernels:

* Graph kernel
* String kernel
* Sigmoid kernel

# Graph Kernel

Explanation:

Graphs are natural data structures to help study relationships between structured objects. Where its nodes are objects and edges are the relationships between them.

Graph kernels allow the ability to measure the similarity between two graphs. These graphs can take the shape of proteins and enzymes, social networks, or web data mining.

There are multiple proposals for kernels between graphs, our focus will be on the basic implementation of a random walk kernel. This implementation performs random walks on 2 graphs and compares the number of same paths found to measure their similarity.

Say for example we have two graphs. Both are in the structure of a triangle.

G1

```

1-----2

- -

3

```

and G2

```

4-----5

- -

6

```

The similarity between them using random walk can be visualized as



The similarity between G1 and G2 can be computed as :

Where is defined as

Ref : <https://jmlr.csail.mit.edu/papers/volume11/vishwanathan10a/vishwanathan10a.pdf>

<https://en.wikipedia.org/wiki/Graph_kernel>

# String Kernel

Explanation:

String kernels allow the ability to measure the similarity between two Strings. Strings can happen in the form of large paragraphs, small ones, single sentence structures, and can even be composed of written out enzymes or proteins.

The String kernel works in extension with support vector machines to handle sequence data by mapping data into a vector space and then performing the kernel function on the space to find the overall similarity between the data points. For example with the string kernel we do not care if the sequences contain contiguous substrings. If we say we have the substring “c-a-r” and additional substrings “card”, and “cardiovascular” the length of the sequences who have our initial substring is 3 and 3 respectively.

We can then measure the distance of similarity between our sequences using a decay factor lambda belonging to (0,1) and multiplying.

For example taking our previous comments on “c-a-r” to find the similarity between “car” and “cat”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | c-a | c-t | a-t | c-r | a-r |
| car | lambda^2 | 0 | 0 | lamda^3 | lamda^2 |
| cat | lamda^2 | lamda^3 | lamda^2 | 0 | 0 |

K(car,car) is the same as taking lambda^2 + lambda^3 two times which is equivalent to 2lamda^4 + lambda^6. Additionally we can find the similarity between car and cat by K(cat, car) which is taking lambda^2 + lambda^3 and lambda^2 + lambda ^3 since the lamda ^3 is coming from two separate places in our matrix we cannot consider them to be similar. This leaves us with just lambda^4 in similarity between our two substrings.

Ref:

<https://www.slideshare.net/DibyenduNath/text-classification-using-text-kernels>

<https://en.wikipedia.org/wiki/String_kernel>

# Sigmoid Kernel

Explanation:

The sigmoid kernel works in unison with support vector machines(SVM). Having a sigmoid kernel activated during svm training allows any point that is in the relative relation to say X, be positively correlated with an attribute. Bellow is a basic sigmoid function to emphasize this point.



The similarity between two points can be defined as where x, y are the input vectors γ is known as slope, and c\_0 is known as intercept. Training an SVM with a kernel function requires that the model store the data and compute the (in this case sigmoid) function on new data during testing.

Using different kernel functions allows us to better classify nonlinear data using a linear model. Depending on your data you can find a kernel that is best suited to the type of classification that you are wanting to implement. This is best done with hyper parameter tuning. As a basic example let's consider the following points:



There isn't currently any real relationship between. But let's consider everything in the green circle its own classification and everything outside of it not. Using a basic linear model we might get a line of best fit even with SVM to be:



Where the redline shows the separation between data classifications. This line might be okay, but notice there is some error going on in basic classification. Using a sigmoid kernel with the sigmoid activation function might bring up something like this:



Where the data is still being classified with less error over our dataset. This can be useful in improving classification models and finding more similarities between data points for specific attributes/features in SVM.

Ref:

<https://scikit-learn.org/stable/modules/metrics.html#sigmoid-kernel>

<https://www.r-bloggers.com/2019/10/support-vector-machines-with-the-mlr-package/>