Problem Set 3

Shalin Shah

March 1, 2020

1 Kernel

1.a Is the function a kernel?

Iff for *any N and any* $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, $\mathbf{K}$ is positive semidefinite, then our function is a kernel function. However, from a less concrete and more theoretical standpoint, the function k is actually looking for differences between the two documents and not for similarities and kernel functions are meant to find the similarities between two data points. So, the function given isn't a kernel.

1.b Show the following is a kernel: $(1 + (\frac{\mathbf{x}}{||\mathbf{x}||}) * (\frac{\mathbf{z}}{||\mathbf{z}||}))^3$

If we multiply this out, we get:
$(1 + (\frac{\mathbf{x}}{||\mathbf{x}||}) * (\frac{\mathbf{z}}{||\mathbf{z}||}))^3 = 1 + 3(\frac{\mathbf{x}}{||\mathbf{x}||})(\frac{\mathbf{z}}{||\mathbf{z}||}) + 3(\frac{\mathbf{x}}{||\mathbf{x}||})^2(\frac{\mathbf{z}}{||\mathbf{z}||})^2 + (\frac{\mathbf{x}}{||\mathbf{x}||})^3(\frac{\mathbf{z}}{||\mathbf{z}||})^3$
$= 1 + (\frac{3}{||\mathbf{x}||||\mathbf{z}||})\mathbf{x} \cdot \mathbf{z} + (\frac{3}{||\mathbf{x}||^2||\mathbf{z}||^2})[\mathbf{x} \cdot \mathbf{z}][\mathbf{x} \cdot \mathbf{z}] + (\frac{1}{||\mathbf{x}||^3||\mathbf{z}||^3})[\mathbf{x} \cdot \mathbf{z}][\mathbf{x} \cdot \mathbf{z}][\mathbf{x} \cdot \mathbf{z}]$

But, we can format this to fit the construction rules given in the problem. All the constants in front of the multiplication fit rule 1 ($\frac{3}{||\mathbf{x}||||\mathbf{z}||}$ can be the psuedo "$f(\mathbf{x})$" and 1 can be the psuedo "$f(\mathbf{z})$" for the second term in the addition). We already know that $\mathbf{x} \cdot \mathbf{z}$ is a kernel, so the second term in the addition is already fine. From rule 3, we know that $\mathbf{x} \cdot \mathbf{z}^N$ is also a kernel since $\mathbf{x} \cdot \mathbf{z}$ is a kernel, so individually all the terms in the sum are a kernel. Finally, from rule 2, we know that the addition of two kernels is a kernel as well, and since all four terms in the addition are kernels, the entire expression is also a kernel.

1.c What are the similarities/differences from the kernel given and kernel β?

First off, $(a + b)^3 = a^3 + 3a^2b + 3b^2a + b^3$, and here $a = 1$ and $b = \beta\mathbf{x} \cdot \mathbf{z}$.
$\beta\mathbf{x} \cdot \mathbf{z} = \beta(x_1z_1 + x_2z_2)$.

All together, we get the following:
$1 + 3\beta(x_1z_1 + x_2z_2) + 3\beta^2(x_1z_1 + x_2z_2)^2 + \beta^3(x_1z_1 + x_2z_2)^3$

$= 1+3\beta x_1z_1+3\beta x_2z_2+3\beta^2x_1^2z_1^2+6\beta^2x_1z_1x_2z_2+3\beta^2x_2^2z_2^2+\beta^3x_1^3z_1^3+3\beta^3x_1^2z_1^2x_2z_2+$
$3\beta^3x_1z_1x_2^2z_2^2 + \beta^3x_2^3z_2^3$.

This is our equation. We see that if $\beta = 1$ the equation generalizes to the regular case where the kernel equals $(1 + \mathbf{x} \cdot \mathbf{z})^3$. The similarities are that the two relay on the same exact input output pairs: x_1, z_1, x_2, and z_2. The differences are that the first one is multiplied by an increasing order for every few terms. This likely makes it so that some features have a larger impact than others, something that we don't see in the case where all are equal and have the same effect on the final outcome as the other. The parameter is likely a hyperparameter that needs to be tuned before being run on a model.

And so to generalize this, we try to find $\phi(\cdot)$ which we'll find for both $\mathbf{x}$ and $\mathbf{z}$. We realize that the equation we get above can be represented by a dot product of an $\mathbf{x}$ and $\mathbf{z}$ (as also listed in Piazza post 175). And so, we look to calculate what makes the dot product.

For $\mathbf{x}$:
$\phi(\mathbf{x}) = (1, \sqrt[2]{3\beta}x_1, \sqrt[2]{3\beta}x_2, \sqrt[2]{3}\beta x_1^2, \sqrt[2]{6}\beta x_1 x_2, \sqrt[2]{3}\beta x_2^2, \sqrt[2]{\beta}\beta x_1^3, \sqrt[2]{3\beta}\beta x_1^2 x_2, \sqrt[2]{3\beta}\beta x_1 x_2^2, \sqrt[2]{3\beta}\beta x_2^3)$
For $\mathbf{z}$:
$\phi(\mathbf{z}) = (1, \sqrt[2]{3\beta}z_1, \sqrt[2]{3\beta}z_2, \sqrt[2]{3}\beta z_1^2, \sqrt[2]{6}\beta z_1 z_2, \sqrt[2]{3}\beta z_2^2, \sqrt[2]{\beta}\beta z_1^3, \sqrt[2]{3\beta}\beta z_1^2 z_2, \sqrt[2]{3\beta}\beta z_1 z_2^2, \sqrt[2]{3\beta}\beta z_2^3)$
It can then be verified that $k_\beta(\mathbf{x}, \mathbf{z}) = \phi^T(\mathbf{x}) \cdot \phi(\mathbf{z})$, and so the solution is complete.

2 SVM

2.a What is the θ^* that satisfies the constrained minimization?

What we have to do is find a vector θ^* that will be a solution. We have to factor in the Lagrange factors and so using the Lagrange multiplier approach, we get that the constraint term is multiplied by a factor α. $L(\mathbf{x}, \alpha) = \frac{1}{2}||\theta||^2 + \alpha y_n \theta^T \mathbf{x}_n$ (definition of a Lagrangian, Slide 25 of February 24^{th} lecture)
Now, what we're actually doing is finding $\mathbf{x}$ and α that allow us to find the same solution as the primal problem, and this can by done by finding:

$$min_{\mathbf{x}} max_{\alpha} L(\mathbf{x}, \alpha)$$

We convert this into a dual problem, where we can get a similar solution:

$$d^* = max_{\alpha} min_{\mathbf{x}} L(\mathbf{x}, \alpha)$$

The partial of the norm squared, $\partial(\frac{1}{2}||\theta||^2)$, is equal to θ and the partial of the other term is simply $\alpha y_n \mathbf{x}_n$. Plugging in these values, we get that:

$$\theta + (-1)\alpha(a, e)^T = \theta - \alpha(a, e)^T = 0$$

From here, we solve for theta and get that

$$\theta = \alpha(a, e)^T$$

We've now found the $min_{\mathbf{x}}$ value, and so now we find the max_{α} value.
For any vector $\mathbf{v}$, $||\mathbf{v}||^2 = v_1^2 + v_2^2 + \cdots + v_n^2$, and so for our norm we get that $|| - \alpha[a, e]||^2 = (\alpha a)^2 + (\alpha e)^2$. The entire expression to maximize thus becomes: $\frac{\alpha^2}{2}(a^2 + e^2) + \alpha y_n \theta^T \mathbf{x}_n - 1$, but $y_n = -1$ and $\theta^T \mathbf{x}_n = -\alpha[a, e]^T[a, e]$. The second term thus becomes $-\alpha[a^2 + e^2] - 1$. max_{α} is found by setting $\frac{\partial[\frac{\alpha^2}{2}(a^2+e^2) - \alpha(\alpha[a^2+e^2]-1)]}{\partial \alpha} = \frac{\alpha(a^2+e^2)}{2} - \alpha(a^2 + e^2) + 1 = 0$. This becomes:

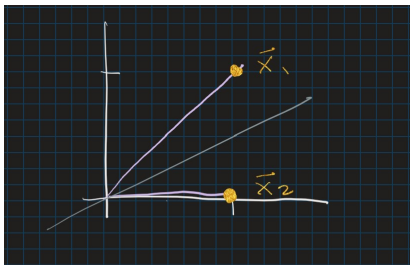
$$\alpha = \frac{1}{a^2 + e^2}$$

and so the vector θ^* that satisfies this will be

$$\theta^* = \alpha[a, e]^T = \frac{1}{a^2 + e^2}[a, e]^T$$

2.b Find θ^* and margin γ

Intuitively, we know that we need to find a line that splits the two data points down the middle as such:



We currently want to find $min\frac{1}{2}||\theta||^2$. For us to do that, we can use a system of equations to find the equation for a line that goes in between the two. The process is done using the following logic:

We know that $y_n\theta^T\mathbf{x}_n \geq 1$ and so we can use this equation twice with the data we have. $\theta^T = [\theta_1, \theta_2]^T$.
For training example $\mathbf{x}_1$:

$$1 \cdot \theta^T \cdot [1, 1] \geq -1$$
$$\theta_1 + \theta_2 \geq -1$$

For training example $\mathbf{x}_2$:

$$-1 \cdot \theta^T \cdot [1, 0] \geq -1$$
$$-\theta_1 \geq -1$$

Solving this system of equations, we get that $\theta_1 \leq -1$ and $\theta_2 \geq 2$. From here, we get that

$$\theta^* = [-1, 2]^T$$

The margin

$$\gamma = \frac{1}{||\theta||} = \frac{1}{\sqrt[2]{5}}$$

2.c If the parameter b were allowed to be non-zero, how does this change the classifier and the margin from the previous question?

Now, instead of only having to worry about θ, we have to worry about b. Since the two points are right on top of each other as shown in the picture from the

previous question, we don't necessarily see any dependence on x_1 for either of the data points. Thus, we'd want to have a straight line go in the middle of the two points with a slope of 0, more specifically meaning that x_1 has no effect on the vector but x_2 does. The vector that would do this for us is $\theta^* = [0, 1]^T$. We would also have $b^* = \frac{1}{2}$. The margin is $\frac{1}{2}$.

3 Twitter analysis using SVMs

3.2 b We should maintain class proportions across folds so that the model can actually learn based on the true data. Otherwise, if we use mostly positive values, the model will have a lot of informaiton about the positive ones and little to none about the negative ones, and thus won't learn as much as we would like it to.

3.2 d

C	accuracy	F1-score	AUROC	precision	sensitivity	specificity
10^{-3}	.7089	.8297	.5	.7089	1	0
10^{-2}	.7107	.8306	.5031	.7102	1	.0063
10^{-1}	.8060	.8755	.7187	.8357	.9294	.5081
10^{0}	.8146	.8749	.7531	.8562	.9017	.6045
10^{1}	.8182	.8766	.7592	.8595	.9017	.6167
10^{2}	.8182	.8766	.7592	.8595	.9017	.6167
best C	100	100	100	100	0.001	100

It seems that for most of the performance metrics, that greater C yields higher accuracy. This is true for all of the metrics besides sensitivity, in which the lowest two C values yielded higher accuracy.

3.3 a The additional hyperparameter is related inversely to the variance of the kernel and so larger values of it mean that values that are actually close to one another will be closer, but this can lead to overfitting. On the other hand, smaller values can lead to underfitting.

3.3 b and c For this one, I used the same C values from the last part but also included changing γ values and got the same results. We see that as γ decreases that in general the performance increases in terms of our accuracy.