CM146, Fall 2017
Problem Set 3: SVM and Kernels
Due Nov 22, 2017

Author: Shivraj Gill

1 Problem 1

(a) Problem 1a

Solution: Solution to problem 1a

To show that $K(\boldsymbol{x}, \boldsymbol{z})$ is a kernel function, we will define a 2x2 matrix K such that

$$K = \begin{pmatrix} k(x,x) = n_x & k(x,z) = n_{x,z} \\ k(z,x) = n_{z,x} & k(z,z) = n_z \end{pmatrix}$$

Note that n_x, n_z represents the number of unique words in document x and document z respectively and $n_{x,z} = n_{z,x}$ represent the number of unique words in the intersection of both x and z. Also, n_x, $n_z \geq n_{x,z}$

From the Mercer Theorem, we know that $K(\boldsymbol{x}, \boldsymbol{z})$ is a kernel function if and only if it's matrix K is PSD. This is equivalent to showing all of K's eigenvalues $\lambda \geq 0$.

$$det(K - \lambda I) = (n_x - \lambda)(n_z - \lambda) - n_{x,z}^2 = 0$$

Solving for λ gives us,

$$\lambda_1 = \frac{n_x + n_z + \sqrt{(n_x - n_z)^2 + 4n_{x,z}^2}}{2}$$
$$\lambda_2 = \frac{n_x + n_z - \sqrt{(n_x - n_z)^2 + 4n_{x,z}^2}}{2}$$

We know that $\lambda_1 \geq 0$ because the square root is always positive and n_x, n_z are nonnegative. $\lambda_2 \geq 0$ since

$$\begin{aligned}
&\lambda_2 = \frac{n_x + n_z - \sqrt{(n_x - n_z)^2 + 4n_{x,z}^2}}{2} \geq 0 \\
&\Rightarrow (n_x + n_z)^2 \geq (n_x - n_z)^2 + 4n_{x,z}^2 \\
&\Rightarrow 4n_x n_y \geq 4n_{x,z}^2 \\
&\Rightarrow n_x n_y \geq n_{x,z}^2
\end{aligned}$$

This is true since n_x, $n_z \geq n_{x,z}$. We have shown that both $\lambda_1, \lambda_2 \geq 0$ so K is PSD matrix and thus, $K(\boldsymbol{x}, \boldsymbol{z})$ is a kernel function.

(b) Problem 1b **Solution:** Solution to problem 1b

We define $k_1(x, z) = x \cdot z$, since that is given in the problem, and define f(x) = $\frac{1}{\|x\|}$ and f(z) = $\frac{1}{\|z\|}$.
From the scaling rule, we can say $k_2(x, z) = \frac{xz}{\|x\|\|z\|}$ is also a kernel.
Next, we define $k_3(x, z) = 1$, which is also a kernel since it satisfies the property $k_3(x, z) = k_3(z, x) = 1$ and $k_3(x, z) = \phi(x)^T \phi(z) = 1 \cdot 1 = 1$
From the sum rule we can define,
$k_4(x, z) = 1 + \frac{xz}{\|x\|\|z\|}$.
Finally from the product rule, we can define
$k_5(x, z) = k_4(x, z) \cdot k_4(x, z) \cdot k_4(x, z) = k_4(x, z)^3 = (1 + (\frac{xz}{\|x\|\|z\|}))^3$.
Thus, we have shown $(1 + (\frac{xz}{\|x\|\|z\|}))^3$ is a kernel function.

(c) Problem 1c

Solution: Solution to problem 1c We will get $K_\beta(\boldsymbol{x}, \boldsymbol{z})$ to be expressed as inner products by plugging in the components and expanding the exponent.

$$\begin{aligned}
K_\beta(\boldsymbol{x}, \boldsymbol{z}) &= (1 + \beta x \cdot z)^3 \text{ where x,z} \in \mathbb{R}^2 \\
&= (1 + \beta(x_1 z_1 + x_2 z_2)) \\
&= \beta^3 x_2^3 z_2^3 + 3\beta^3 x_1 x_2^2 z_1 z_2^2 + 3\beta^3 x_1^2 x_2 z_1^2 z_2 + \beta^3 x_1^3 z_1^3 + 3\beta^2 x_2^2 z_2^2 \\
&+ 6\beta^2 x_1 x_2 z_1 z_2 + 3\beta^2 x_1^2 z_1^2 + 3\beta x_2 z_2 + 3\beta x_1 z_1 + 1
\end{aligned}$$

By examination, this gives us

$$\phi(x) = \begin{bmatrix} \beta^{\frac{3}{2}} x_1^3 \\ \beta^{\frac{3}{2}} \sqrt{3} x_1^2 x_2 \\ \beta^{\frac{3}{2}} \sqrt{3} x_1 x_2^2 \\ \beta^{\frac{3}{2}} x_2^3 \\ \beta \sqrt{3} x_1^2 \\ \beta \sqrt{6} x_1 x_2 \\ \beta \sqrt{3} x_2^2 \\ \sqrt{\beta} \sqrt{3} x_1 \\ \sqrt{\beta} \sqrt{3} x_2 \\ 1 \end{bmatrix} \tag{1}$$

$K_\beta(x, z)$ has a similar feature transformation to K(x,z), where the only difference is that $\phi(x)$ for K(x,z) is not multiplied by a β term. So all the components in $\phi(x)$ for K(x,z) are exactly the same outside of the β term.
β acts as a parameter that scales $\phi(x)$. Note how that that for components of $\phi(x)$ with smaller powers of the features such as the component $x_1 x_2$, the scaling is just done by a factor of β, but for components with larger powers such as $x_1^2 x_2$, the scaling is done by $\beta^{\frac{3}{2}}$. Thus, if $0 < \beta < 1$ (which is usually the case),then components of $\phi(x)$ with higher level features, will be scaled down by $\beta^{\frac{3}{2}}$, and hence preventing them from gaining really large values thereby reducing their overall affect on training.

2 Problem 2

(a) Problem 2a **Solution:** Solution to problem 2a

For a single training vector, x = $(a, e)^T$ with label y = -1, we have the following optimization problem.

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \frac{1}{2}\|\theta\|^2 \\ \text{subject to} \quad & -\theta(a,e)^T \geq 1 \end{aligned}$$

We define our Lagrangian function to be

$$L(\theta, \alpha) = \frac{1}{2}\|\theta\|^2 + \alpha(\theta(a,e)^T + 1)$$

Now we define our primal to be

$$\min_{\theta} \max_{\alpha} \quad L(\theta, \alpha)$$

We can solve this problem using duality. In other words, we define

$$\begin{aligned} & \max_{\alpha} \quad g(\alpha) \\ \text{where } g(\alpha) = & \min_{\theta} \quad L(\theta, \alpha) \end{aligned}$$

Fist we find $g(\alpha)$ by differentiating L w.r.t θ which gives us

$$\begin{aligned} \frac{\partial L}{\theta} &= \theta + \alpha(a,e)^T = 0 \\ \Rightarrow \hat{\theta} &= -\alpha(a,e)^T \end{aligned}$$

This gives us:

$$\begin{aligned} g(\alpha) &= \frac{1}{2}\|-\alpha(a,e)^T\|^2 + \alpha(-\alpha(a,e)^T(a,e)^T + 1) \\ &= \frac{1}{2}(\alpha^2a^2 + \alpha^2e^2) + \alpha(-\alpha a^2 - \alpha e^2 + 1) \\ &= \frac{-\alpha^2(a^2+e^2)}{2} + \alpha \end{aligned}$$

Taking the derivative of g(α) w.r.t α and setting it to 0 give us:

$$\begin{aligned} \frac{\partial g(\alpha)}{\partial \alpha} &= 0 \\ \Rightarrow -\alpha(a^2+e^2)+1 &= 0 \\ \Rightarrow \hat{\alpha} &= \frac{1}{a^2+e^2} \end{aligned}$$

Substituting $\hat{\alpha}$ into $\hat{\theta}$ gives us

$$\theta^* = -\frac{1}{a^2+e^2}(a,e)^T$$

θ^* is a unit vector in the direction of x.

(b) Problem 2b **Solution:** Solution to problem 2b

We have the following optimization problem.

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \frac{1}{2}\|\theta\|^2 \\ \text{subject to} \quad & \theta_1+\theta_2 \geq 1 \\ \text{and} \quad & -\theta_1 \leq 1 \end{aligned}$$

We define our Lagrangian function to be

$$L(\theta,\alpha) = \frac{1}{2}(\theta_1^2+\theta_2^2)+\alpha_1(1-\theta_1-\theta_2)+\alpha_2(1+\theta_1)$$

Now we define our primal to be

$$\min_{\theta}\max_{\alpha} \quad L(\theta,\alpha)$$

We can solve this problem using duality. In other words, we define

$$\begin{aligned} & \max_{\alpha} \quad g(\alpha) \\ \text{where } g(\alpha) = & \min_{\theta} \quad L(\theta,\alpha) \end{aligned}$$

Fist we find $g(\alpha)$ by differentiating L w.r.t θ which gives us

$$\begin{aligned} \frac{\partial L}{\theta_1} &= \theta_1-\alpha_1+\alpha_2 = 0 \\ \Rightarrow \theta_1 &= \alpha_1-\alpha_2 \\ \frac{\partial L}{\theta_2} &= \theta_2-\alpha_1 = 0 \\ \Rightarrow \theta_2 &= \alpha_1 \end{aligned}$$

This gives us:

$$g(\alpha) = \frac{1}{2}(\alpha_1^2 - 2\alpha_1\alpha_2 + \alpha_2^2 + \alpha_1^2) + \alpha_1(1 - 2\alpha_1 + \alpha_2) + \alpha_2(\alpha_1 - \alpha_2 + 1)$$

Taking the derivative of g(α) w.r.t α and setting it to 0 give us:

$$\begin{aligned} \frac{g(\alpha)}{\partial\alpha_1} &= 0 \\ \Rightarrow -2\alpha_1 + \alpha_2 + 1 &= 0 \\ \Rightarrow \alpha_1 &= \frac{\alpha_2 + 1}{2} \\ \frac{g(\alpha)}{\partial\alpha_2} &= 0 \\ \Rightarrow -\alpha_2 + \alpha_1 + 1 &= 0 \\ \Rightarrow \alpha_2 &= \alpha_1 + 1 \\ \Rightarrow \alpha_1 = 2, \alpha_2 &= 3 \end{aligned}$$

Substituting α into θ gives us

$$\theta^* = (-1, 2)^T$$

Thus, the margin is $\gamma = \frac{1}{\|\theta\|} = \frac{1}{\sqrt{5}}$

(c) Problem 2c **Solution:** Solution to problem 2c

We have the following optimization problem.

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \frac{1}{2}\|\theta\|^2 \\ \text{subject to} \quad & \theta_1 + \theta_2 + b \geq 1 \\ \text{and} \quad & -(\theta_1 + b) \leq 1 \end{aligned}$$

We define our Lagrangian function to be

$$L(\theta, b, \alpha) = \frac{1}{2}(\theta_1^2 + \theta_2^2) + \alpha_1(1 - b - \theta_1 - \theta_2) + \alpha_2(1 + b + \theta_1)$$

Now we define our primal to be

$$\min_{\theta} \max_{\alpha} \quad L(\theta, \alpha)$$

We can solve this problem using duality. In other words, we define

$$\begin{aligned} &\max_{\alpha} \quad g(\alpha) \\ \text{where } g(\alpha) = &\min_{\theta} \quad L(\theta, \alpha) \end{aligned}$$

Fist we find $g(\alpha)$ by differentiating L w.r.t θ which gives us

$$\begin{aligned} \frac{\partial L}{\theta_1} &= \theta_1 - \alpha_1 + \alpha_2 = 0 \\ \Rightarrow &\theta_1 = \alpha_1 - \alpha_2 \\ \frac{\partial L}{\theta_2} &= \theta_2 - \alpha_1 = 0 \\ \Rightarrow &\theta_2 = \alpha_1 \end{aligned}$$

This gives us:

$$g(\alpha) = -\alpha_1^2 + \alpha_1\alpha_2 - \frac{1}{2}\alpha_2^2 + \alpha_1(1-b) + \alpha_2(1+b)$$

Taking the derivative of g(α) w.r.t α and b and setting it to 0 give us:

$$\begin{aligned} \frac{g(\alpha)}{\partial \alpha_1} &= 0 \\ \Rightarrow -2\alpha_1 + \alpha_2 + 1 - b &= 0 \\ \Rightarrow \alpha_1 &= \frac{\alpha_2 + 1 - b}{2} \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{g(\alpha)}{\partial \alpha_2} &= 0 \\ \Rightarrow -\alpha_2 + \alpha_1 + 1 + b &= 0 \\ \Rightarrow \alpha_2 &= \alpha_1 + 1 + b \end{aligned} \tag{3}$$

$$\frac{g(\alpha)}{\partial b} = b(\alpha_2 - \alpha_1) = 0 \tag{4}$$

Now if b = 0, then we get the same result as part b). If $(\alpha_2 - \alpha_1) = 0$, then we get:

$$\begin{aligned} \alpha_2 &= \alpha_1 \\ \alpha_1 &= \frac{\alpha_1 + 1 - b}{2} \text{ From (1)} \\ \Rightarrow \alpha_1 &= 1 - b \end{aligned}$$

$$\alpha_2 = \alpha_1 + 1 + b$$
$$\Rightarrow \alpha_1 = \alpha_1 + 1 + b$$
$$\Rightarrow b = -1, \alpha_1 = 2, \alpha_2 = 2$$

Substituting α into θ gives us

$$\theta^* = (0, 2)^T$$

Thus, overall with an offset, we get the following results:

$$\theta^* = (0, 2)^T, b^* = -1, \gamma = \frac{1}{2}$$

Without an offset, we had the following results:

$$\theta^* = (1, -2)^T, \gamma = \frac{1}{\sqrt{5}}$$

This shows that an offset terms reduces the margin, γ, for our decision boundary.

3 Problem 3

(a) Problem 3.2b **Solution:** Solution to problem 3.2b

It would be beneficial to maintain class proportions across folds, because we want our classifier to learn the decision boundary that separates the classes the best, which can only happen if the model is trained on enough positive and negative examples. In other words, our classifier would not be biased towards a specific class if each fold represents the classes with equal proportions. Moreover, if a fold has a high proportion of one class, our decision boundary could overfit towards that class in order to minimize the training error. On the other hand, if a fold has a low proportion of one class, then that class would have less influence on the model and would make it more difficult for the decision boundary to separate the classes well.

(b) Problem 3.2d **Solution:** Solution to problem 3.2d

C	Accuracy	F1	AUROC	Precision	Sensitivity	Specificity
0.001	0.7089	0.8297	0.5000	0.7089	1.0000	0.0000
0.01	0.7107	0.8306	0.5031	0.7102	1.0000	0.0100
0.1	0.806	0.8755	0.7188	0.8357	0.9294	0.0062
1.0	0.8146	0.8749	0.7531	0.8562	0.9017	0.5081
10.0	0.8182	0.8766	0.7592	0.8595	0.9017	0.6045
100.0	0.8182	0.8766	0.7592	0.8595	0.9017	0.6167
Best C	10.0	10.0	10.0	10.0	0.001	10.0

We see that for every metric besides sensitivity, our performance score increases as C increases and plateaus at the optimal value, $C = 10$. For sensitivity, the performance score decreases as C increases, and plateaus at $C = 1$. The optimal value for sensitivity is at $C = 0.001$. Another important observation is that specificity takes up a long range of values from $C = 0.001$ to $C = 100$. This shows that as C increases, the number of true negatives increases, and hence the specificity increases. Conversely, as C increases, the number of true positives decreases, and hence the sensitivity decreases. This indicates that our hyperplane is shifting in a direction to include more correctly classified negative examples while losing some correctly classified positive examples.

(c) Problem 3.3a **Solution:** Solution to problem 3.3a

γ determines the variance and bias of the RBF-kernel, and hence the impact of support vectors. When γ is large, the RBF-kernel will have a small variance and high bias, so support vectors would have little influence on the label of a training example. If γ is large enough,

then we can have overfitting(low training error and high test error) issues because none of the support vectors would influence the training examples and so our model would end up memorizing the training examples. Conversely, when γ is small, the RBF-kernel will have a large variance and small bias , so the support vectors influence many points around it, and potentially influencing those that are far away. If γ is small enough, then we can have underfitting(high training error and high test error) because each support vector has a huge influence on surrounding points so our model would be unable to capture any patterns in the training examples.

(d) Problem 3.3b **Solution:** Solution to problem 3.3b

My grid was the pairwise combinations of C = [0.001, 0.01, 0.1, 1, 10, 100] and γ = [0.001, 0.01, 0.1, 1, 10, 100]. I used this grid because it considers every single hyperparameter combination of C and γ. It allows me to see how the model performs when C is large and γ is small, or vice versa. The grid contains a wide array of different hyperparameter values, allowing us to test out many different combinations, and finding the combination that gives us the optimal model performance.

(e) Problem 3.3c **Solution:** Solution to problem 3.3c

Metric	Score	C	g
Accuracy	0.8165	100	0.01
F1	0.8763	100	0.01
AUROC	0.7545	100	0.01
Precision	0.8583	100	0.01
Sensitivity	1.0000	0.01	0.001
Specificity	0.6047	100	0.01

Similar to the linear kernel, we see that every metric besides sensitivity agree on the optimal parameter value, that is C = 100, γ = 0.001. Outside of sensitivity, the model performance improves when C $\geq$ 1 and $\gamma \leq 0.1$ for every metric. Further, when C $<$ 1, the values of γ do not affect the model performance, and the model has low performance scores. Conversely, for sensitivity, when C $>$ 1, and $\gamma > 0.01$, the model performance score improves. Sensitivity

(f) Problem 3.4a **Solution:** Solution to problem 3.4a

For the linear kernel SVM, I used C = 10 since this was the most common optimal C from all of the performance metrics. For the RBF kernel SVM , I used C = 100, $\gamma = 0.01$ since this was the most common optimal pair from all of the performance metrics.

Metric	Linear Kernel SVM	RBF-Kernel SVM
Accuracy	0.7429	0.7571
F1	0.4709	0.4516
AUROC	0.6395	0.6361
Precision	0.6154	0.7000
Sensitivity	0.3810	0.3333
Specificity	0.8980	0.9388

(g) Problem 3.4c **Solution:** Solution to problem 3.4c

The table tells us that both kernels perform very similarly on each performance metric. Moreover, we see that both kernels have overfitting issues with the sensitivity metric, and hence perform very poorly on test data. The two kernels differ most when it comes to the precision metric, where the difference in score is about 0.085. Lastly, the RBF-Kernel with the specificity metric achieves the best overall performance score, 0.9388, which is by far the highest performance score.