SUTD 2021 50.007 Midterm Examination

James Raphael Tiovalen 1004555

Problem 1: True or False

1. True. A training set S can have different geometric margins depending on how the hyperplane is defined. The optimal hyperplane for a given S is the one that gives maximum geometric margin over all possible hyperplanes.
2. True. A complex learning model can reduce the training error as well as the test error.
3. True. In clustering, the choice of which distance metric to use is important as it will determine the type of clusters you will find.
4. True. To use stochastic gradient descent algorithm for minimization, you must have a convex function. Otherwise, it will never let us reach the global optimal value.
5. True. For a small set of training examples, your model trained on that data will have high variance.
6. True. In k-medoids clustering algorithm, we can use different distance measures other than the squared Euclidean distance.
7. True. The hinge loss function allows us to obtain the minimum empirical risk for classification even when the training examples are not linearly separable.

Problem 2: Multiple Choice Questions

1. The following statements are true:
   - Logistic Regression is a model that is used for classification, not for regression.
   - Suppose that you have trained a logistic regression classifier, and it outputs on a new example a prediction of 0.67. This means that our estimate for $P(y = 0|x; \theta)$ is 0.33.
2. Even after we have found a good value for the regularization term λ using cross-validation, since we have acquired more data/a larger dataset (more training examples), the λ value would need to be decreased.
3. Only the following statement is true:
   - The radial basis function kernel is special in many ways. For example, running the SVM with such a kernel function will always be able to return you a separable solution provided the training examples are all distinct.
4. In SVM, maximum margin linear separators can be found by solving the primal quadratic programming problem:

$$\min \frac{1}{2}\|w\|^2 \text{ subject to } d_i(w^T x_i + b) \geq 1$$

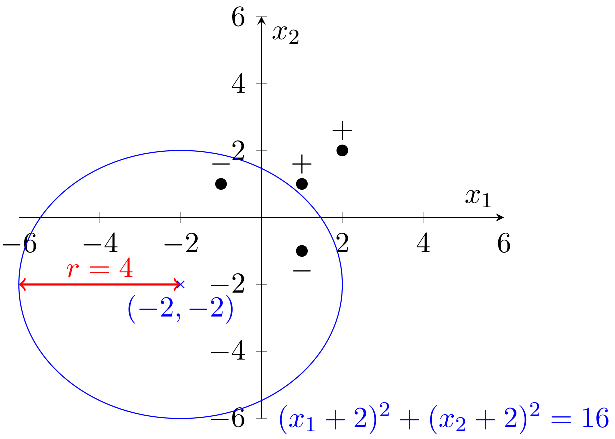
5. The problems that can be solved using the K-medoids algorithm include:

- Classification of dogs and cats.
- Categorizing news articles into sports, technology, finance and politics categories.

6. When the $\mathcal{H}$ set is too large (no constraints), the classifier will **overfit** on the training data and when the $\mathcal{H}$ set is too small, it will **underfit** the training data.

Problem 3: Classification

(a) We can plot a visual, graphical representation of the given data points, as well as one possible classifier:



The classifier specified in the diagram has the parameters $a = -2, b = -2, r = 4$. More formally, such a classifier can be defined as

$$h(x) = \begin{cases} +1, & \text{if } (x_1 + 2)^2 + (x_2 + 2)^2 \geq 16; \\ -1, & \text{if } (x_1 + 2)^2 + (x_2 + 2)^2 < 16 \end{cases}$$

(b) When a training data point $x^{(t)}$ is misclassified, the sign of $(\theta^{(k)} \cdot x^{(t)})$ would disagree with $y^{(t)}$; thus, the product $y^{(t)}(\theta^{(k)} \cdot x^{(t)})$ is non-positive. Given that the updated parameters are given by $\theta^{(k+1)} = \theta^{(k)} + y^{(t)}x^{(t)}$, then when we consider classifying the sample training example $x^{(t)}$ after the update, using the new parameters $\theta^{(k+1)}$, then we would obtain:

$$\begin{aligned} y^{(t)}(\theta^{(k+1)} \cdot x^{(t)}) &= y^{(t)}(\theta^{(k)} + y^{(t)}x^{(t)}) \cdot x^{(t)} \\ &= y^{(t)}(\theta^{(k)} \cdot x^{(t)}) + (y^{(t)})^2(x^{(t)} \cdot x^{(t)}) \\ &= y^{(t)}(\theta^{(k)} \cdot x^{(t)}) + \left\| x^{(t)} \right\|^2 \end{aligned}$$

Since $\left\| x^{(t)} \right\|^2 > 0$, the value of $y^{(t)}(\theta \cdot x^{(t)})$ is guaranteed to increase as a result of the update (i.e., becomes more positive). As we consider the same training example repeatedly, we will necessarily change the parameters such that the example will be eventually classified correctly (i.e., the value of $y^{(t)}(\theta \cdot x^{(t)})$ becomes positive).

Problem 4: Regression

Let the features x_1, x_2, x_3, x_4 be the production price, selling price, retail price of competitors, and total number of candies in the market, respectively.

1. For linear regression, we simply want to find the real values of the weights $\theta = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4\end{bmatrix}^T$ and θ_0 that minimizes the least square loss and optimizes the linear model $f(x;\theta,\theta_0) = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_0$. A possible drawback might be that this model is sensitive to outliers, and we need to assume that each feature is independent from each other, which might not be the case in real life. The model might also be more susceptible to over-fitting. However, this model has a lower computational time complexity.

2. For polynomial regression, we want to find the real values of the weights θ, θ_0, and N that minimizes the least square loss and optimizes a polynomial model of $f(x;\theta,\theta_0)$. This polynomial model will include all powers of each feature up to N (e.g., x_1^2, x_2^2, etc.), which is the maximum power that we will consider, as well as feature interactions between each feature (e.g., x_1x_2, x_2x_3, etc.). While a polynomial regression might provide a much better approximation of the relationship between the features and the amount of candy that they will be able to sell, a possible drawback might be that this model is also sensitive to outliers, and it might have a much higher computational time complexity.

3. For ridge regression, our prediction process is similar to the linear regression, except now, we will also add a regularization parameter $\lambda \geq 0$ to the model. This way, we can exclude potentially irrelevant features that might not affect the amount of candy that they will be able to sell at all. However, since the regularization parameter will trade variance for bias, the final estimate using a ridge regression model might be quite biased.

Problem 5: K-Means

a) For one iteration, we have:

Data		Distance to		Cluster
i	z_i	$C_1 = (1,0)$	$C_2 = (3,2)$	Assignment
1	(1,0)	0	$\sqrt{8}$	C_1
2	(3,2)	$\sqrt{8}$	0	C_2
3	(2,4)	$\sqrt{17}$	$\sqrt{5}$	C_2
4	(8,7)	$\sqrt{98}$	$\sqrt{50}$	C_2
5	(9,11)	$\sqrt{185}$	$\sqrt{117}$	C_2
6	(10,10)	$\sqrt{181}$	$\sqrt{113}$	C_2

b) We need at least **two** iterations to converge to the final cluster assignments (**three**, if we include the additional final check, at which there will be no more changes to the cluster assignments). The final cluster assignments would be: $C_1 = \{z_1, z_2, z_3\}$ and $C_2 = \{z_4, z_5, z_6\}$.

c) Euclidean distance of z_7 to C_1 is $\sqrt{(11-2)^2 + (8-2)^2} = \sqrt{117}$, while Euclidean distance of z_7 to C_2 is $\sqrt{(11-9)^2 + (\frac{28}{3} - 8)^2} = \sqrt{\frac{52}{9}}$. Hence, since $\sqrt{\frac{52}{9}} < \sqrt{117}$, z_7 will be assigned to cluster C_2.

Problem 6: SVM

1. This is because from one of the KKT conditions, we have $C = \alpha_i + \beta_i$. Since $\alpha_i = C - \beta_i$, $\alpha_i \geq 0$, and $\beta_i \geq 0$, we can combine said constraints to form the constraint $0 \leq \alpha_i \leq C$.

2. The kernel trick is to find $K(x, x') = \varphi(x)\varphi(x') = \exp(-\frac{\|x-x'\|}{2})$ instead of finding the explicit form of $\varphi(x)$ or $\varphi(x')$, which is much more difficult. It is useful when the data is not linearly separable, and when we want to transform the data into a higher dimension space to make it linearly separable. For example, by using a radial basis function kernel, we can run SVM and always obtain a separable solution (provided that the training examples are distinct).

3. Yes. A polynomial with degree 0 is a valid kernel, since it is symmetric and positive semi-definite.

4. Yes, since it is symmetric and positive semi-definite.

5. No, since while it is symmetric, it is not positive semi-definite (due to the -8).

6. No, since it is not symmetric.

Problem 7: Logistic Regression

(a) We might need logistic regression when we need to classify whether a tumor is malignant or benign. The data will be labelled. The input will be labelled tumor images, while the outputs will be the probability that it is malignant. We might prefer using logistic regression over SVM since the data might not be linearly separable, and that we want to know the probability of the tumor being malignant.

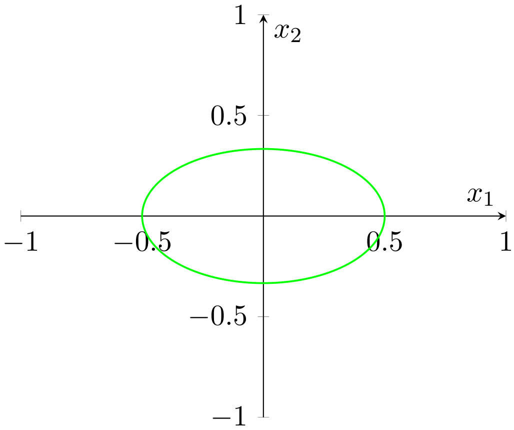
(b) Given that $\theta = \begin{bmatrix} -1 & 0 & 0 & 4 & 9 \end{bmatrix}^T$, we have:

$$y = \begin{cases} 1, & \text{if } -1 + 4x_1^2 + 9x_2^2 \geq 0; \\ 0, & \text{if } -1 + 4x_1^2 + 9x_2^2 < 0 \end{cases}$$

We can rearrange the inequalities to obtain our decision boundary as:

$$y = \begin{cases} 1, & \text{if } 4x_1^2 + 9x_2^2 \geq 1; \\ 0, & \text{if } 4x_1^2 + 9x_2^2 < 1 \end{cases}$$

Graphically, we can visualize the decision boundary as an ellipse centered at $(0, 0)$, with major axis of length 1 and minor axis of length $\frac{2}{3}$. We can plot the diagram as such:



(c) No, it will not change. This is because the new data will be test data, instead of being considered as training data.

Problem 8: Neural Networks

Output of h_1 is 0 since $g(x) = (-1) \cdot 2 + 2 \cdot 1 + (-1) \cdot 2 = -2 + 2 - 2 = -2 < 0$, while output of h_2 is 1 since $g(x) = 0 \cdot 2 + (-1) \cdot 1 + 1 \cdot 2 = 0 - 1 + 2 = 1 \geq 0$. Hence, final output is 2, since $g(x) = 0 \cdot 1 + 1 \cdot 2 = 0 + 2 = 2 \geq 0$.