

# CS229 Homework 2

Haiyuan Mei

TOTAL POINTS

**64.25 / 74**

## QUESTION 1

### Logistic Regression: Training stability 15 pts

#### 1.1 a 2 / 2

- ✓ + 2 pts Correct
- + 0 pts Unanswered
- + 0 pts Incorrect
- + 1 pts Mentioned B took longer, but either claimed it converged or did not specify it did not converge.
- + 0 pts Ran dataset B for a very very long time (multiple days!?)
- + 0 pts Mistagged

#### 1.2 b 5 / 5

- ✓ + 5 pts Correct: linearly separable and explanation
- + 3 pts Almost: linearly separable but insufficient explanation
- + 1 pts Attempt
- + 0 pts Blank
- + 0 pts Failed to tag properly :(

#### 1.3 C 4.5 / 5

- 0 pts Correct
- 1 pts Part i incorrect
- 1 pts Part ii incorrect
- 0.5 pts Part ii insufficient justification (e.g. only addresses convergence as satisfying the stopping condition)
- 1 pts Part iii incorrect
- 1 pts Part iv incorrect
- 1 pts Part v incorrect
- ✓ - 0.5 pts Part v insufficient justification (e.g. assumes Gaussian noise makes data linearly inseparable)
- 5 pts Blank or incomplete

#### 1.4 d 3 / 3

#### ✓ + 3 pts Correct

- + 1 pts Incorrect answer, SVM is not vulnerable
- + 2.5 pts Insufficient explanation, not mentioning maximize geometric margin (hint)
- + 2.5 pts The geometric margin should be maximized instead of minimized
- + 0 pts No Answer
- + 1 pts No explanation

## QUESTION 2

### Model Calibration 10 pts

#### 2.1 a 5 / 5

- ✓ + 5 pts Correct
- + 4 pts Minor error
- + 4 pts Derivation not complete
- + 3 pts Major error
- + 2 pts Barely attempted
- + 0 pts Not attempted

#### 2.2 b 2 / 3

- + 0 pts Not attempted
- + 3 pts Correct
- + 1 pts Both direction are not correct
- ✓ + 2 pts One direction is correct
- + 1 pts Conclusion without explanation
- + 2.5 pts Minor errors
- + 1.5 pts One direction wrong, one direction with minor error
- + 1 pts Both directions are wrong

#### 2.3 C 1 / 2

- + 2 pts Correct
- + 1 pts Incorrect explanation of effect of sign of bias term
- ✓ + 1 pts Incomplete/Incorrect explanation
- + 0 pts Not attempted

### QUESTION 3

## Bayesian Interpretation of Regularization 20 pts

### 3.1 a 3 / 3

- ✓ + 3 pts Correct
- + 2 pts Partially correct with minor mistakes
- + 0 pts Incorrect or not attempted or incorrectly tagged

### 3.2 b 4.75 / 5

- ✓ - 0 pts Correct
- ✓ - 0.25 pts Wrong multivariate Gaussian expression (constant of normalization)
  - 1 pts Minor mistake in the proof
  - 1 pts Too many steps skipped
  - 2 pts Correct approach but wrong result
  - 3 pts Correct approach but wrong proof
  - 4 pts Wrong proof and result
  - 5 pts No or wrong submission
  - 1 pts Hard to read

### 3.3 c 6 / 7

- + 7 pts Correct
- ✓ + 6 pts Minor Error
  - + 5 pts Error
  - + 4 pts Major error
  - + 3 pts Did not solve for theta map
  - + 2 pts Incomplete solution
  - + 1 pts Attempted
  - + 0 pts Missing
- ☹ There should be no m in the final expression

### 3.4 d 3 / 5

- 0 pts Correct
- 2 pts Missing  $\sigma^2$  in expression for gamma
- ✓ - 2 pts Wrong value for gamma (or not reported)
  - 5 pts Unattempted
  - 2 pts Does not match  $J(\theta)$  expression (argmax vs argmin)
  - 1 pts Only shown for 1 example. Does not match  $J(\theta)$
  - 1 pts Minor error

### QUESTION 4

## Constructing kernels 18 pts

### 4.1 a 1 / 1

- ✓ + 1 pts correct with proof
- + 0 pts Incorrect or missing/incorrect proof
- + 0 pts Tag the pages!

### 4.2 b 1 / 1

- ✓ + 1 pts Correct with proof
- + 0 pts Incorrect or missing/incorrect proof
- + 0 pts Incorrect Tags/Illegible scans

### 4.3 c 1 / 1

- ✓ + 1 pts Correct with proof.
- + 0 pts Incorrect or missing/incorrect proof
- + 0 pts Incorrect Tags/Illegible scans

### 4.4 d 1 / 1

- ✓ + 1 pts Correct with proof
- + 0 pts Incorrect or missing/incorrect proof
- + 0 pts Incorrect Tags/Illegible scans

### 4.5 e 2 / 5

- + 5 pts Correct with proof
- ✓ + 3 pts incomplete or partially correct proof
  - + 1 pts correct answer, but missing/incorrect proof
  - + 0 pts missing/incorrect
  - + 2 pts  $x^T x$  is not a invertible matrix
- 1 Point adjustment
  - ☹  $x x^T$  is not identity.

### 4.6 f 3 / 3

- ✓ + 3 pts Correct with proof
- + 1 pts Correct answer, missing/incorrect proof
- + 0 pts Incorrect/missing

### 4.7 g 3 / 3

- ✓ + 3 pts Correct with proof
- + 1 pts correct answer, missing/incorrect proof
- + 0 pts Incorrect/missing
- + 0 pts Click here to replace this description.

### 4.8 h 3 / 3

- ✓ - 0 pts Correct
- 3 pts Incorrect/ No submission
- 1 pts Minor mistake

- **2 pts** Incorrect proof
- **0 pts** Please tag your pages

QUESTION 5

## Kernelizing the Perceptron 11 pts

### 5.1 a 8 / 9

- + **3 pts** i) Correct
- ✓ + **2 pts** i) partial soln, missing initial value or linear representation
- + **1 pts** i) partial soln
- + **0 pts** i) Incorrect
- ✓ + **3 pts** ii) correct
- + **2 pts** ii) missing kernel representation/ partial soln
- + **1 pts** ii) partial solution
- ✓ + **3 pts** iii) correct
- + **2 pts** iii) Partial soln / missing full reasoning
- + **1 pts** iii) partial solution
- + **0 pts** iii) Blank/Incorrect
- + **0 pts** Wrong solution/ unattempted
- + **0 pts** ii) blank/incorrect

### 5.2 C 2 / 2

- ✓ + **2 pts** Correct
- + **1 pts** Only correctly mentioned that dot product perform poorly
- + **0 pts** Blank
- + **0 pts** Incorrect

# CS 229, Fall 2018

## Problem Set #2 Solutions: Supervised Learning II

Haiyuan Mei ([hmei0411@stanford.edu](mailto:hmei0411@stanford.edu))

### 1. Logistic Regression: Training stability

- a) dataset A converges very quickly, but dataset B doesn't seem to be able to converge ever.
- b) The short answer is that Logistic Regression will not converge when data is perfectly separable. I will try to explain this mathematically as following.

- First part of the proof below will show that if the data is perfectly separable, the gradient of loss function w.r.t.  $\theta$  can never be 0, which means the log-likelihood can never reach to a maximum. The probability of a point is:

$$p(x^{(i)}) = \frac{1}{1 + e^{y^{(i)}\theta^T x^{(i)}}}$$

For simplicity, assume  $x^{(i)}$  only has one feature,  $\theta$  is just a scalar. A correctly classified point should satisfy  $y^{(i)}x^{(i)}\theta < 0$  s. t.  $p(x^{(i)}) > 0.5$ ; Otherwise the sample is misclassified.

The MLE is the solution of  $\ell'(\theta) = 0$ . To further simplify the notation let's assume  $c^{(i)} = y^{(i)}x^{(i)}$ , the log-likelihood loss can be written as:

$$\ell(\theta) = \frac{1}{m} \sum_{i=1}^m \log(h(x^{(i)})) = -\frac{1}{m} \sum_{i=1}^m \log(1 + e^{y^{(i)}x^{(i)}\theta}) = -\frac{1}{m} \sum_{i=1}^m \log(1 + e^{c^{(i)}\theta})$$

Take the derivative w.r.t.  $\theta$ :

$$\ell'(\theta) = -\frac{1}{m} \sum_{i=1}^m \frac{c^{(i)} e^{c^{(i)}\theta}}{1 + e^{c^{(i)}\theta}} = \frac{1}{m} \sum_{i=1}^m \frac{-c^{(i)}}{1 + e^{-c^{(i)}\theta}}$$

Since  $c^{(i)}\theta < 0$ ,  $1 + e^{-c^{(i)}\theta} > 0$ , we have  $\ell'(\theta) \neq 0$  for all  $\theta$ . This explains why there is no convergence. The plot of  $\ell'(\theta)$  should look something like the following. Note that when  $\theta < 0$ , we should have  $c^{(i)} > 0$  to make  $c^{(i)}\theta < 0$ , which is the lower part of Figure 1; when  $\theta > 0$ , we should have  $c^{(i)} < 0$  to make  $c^{(i)}\theta < 0$ , which is the upper part of Figure 1.

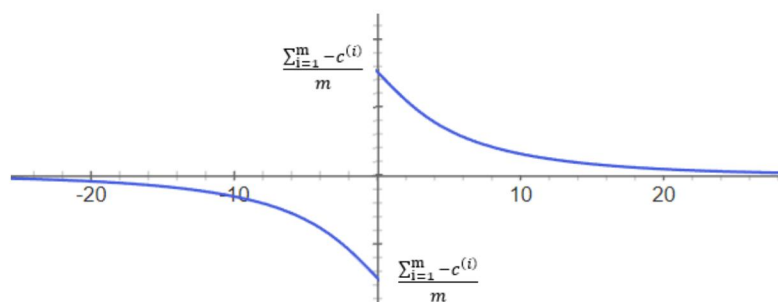


Figure 1, Gradient w.r.t.  $\theta$

- Now consider there is a misclassified point. Just one misclassified point will make  $\ell'(\theta) = 0$  solvable, hence the algorithm can converge. A misclassified point means in the assumption

1.1 a 2 / 2

✓ + 2 pts Correct

+ 0 pts Unanswered

+ 0 pts Incorrect

+ 1 pts Mentioned B took longer, but either claimed it converged or did not specify it did not converge.

+ 0 pts Ran dataset B for a very very long time (multiple days!?)

+ 0 pts Mistagged

# CS 229, Fall 2018

## Problem Set #2 Solutions: Supervised Learning II

Haiyuan Mei ([hmei0411@stanford.edu](mailto:hmei0411@stanford.edu))

### 1. Logistic Regression: Training stability

- a) dataset A converges very quickly, but dataset B doesn't seem to be able to converge ever.
- b) The short answer is that Logistic Regression will not converge when data is perfectly separable. I will try to explain this mathematically as following.

- First part of the proof below will show that if the data is perfectly separable, the gradient of loss function w.r.t.  $\theta$  can never be 0, which means the log-likelihood can never reach to a maximum. The probability of a point is:

$$p(x^{(i)}) = \frac{1}{1 + e^{y^{(i)}\theta^T x^{(i)}}}$$

For simplicity, assume  $x^{(i)}$  only has one feature,  $\theta$  is just a scalar. A correctly classified point should satisfy  $y^{(i)}x^{(i)}\theta < 0$  s. t.  $p(x^{(i)}) > 0.5$ ; Otherwise the sample is misclassified.

The MLE is the solution of  $\ell'(\theta) = 0$ . To further simplify the notation let's assume  $c^{(i)} = y^{(i)}x^{(i)}$ , the log-likelihood loss can be written as:

$$\ell(\theta) = \frac{1}{m} \sum_{i=1}^m \log(h(x^{(i)})) = -\frac{1}{m} \sum_{i=1}^m \log(1 + e^{y^{(i)}x^{(i)}\theta}) = -\frac{1}{m} \sum_{i=1}^m \log(1 + e^{c^{(i)}\theta})$$

Take the derivative w.r.t.  $\theta$ :

$$\ell'(\theta) = -\frac{1}{m} \sum_{i=1}^m \frac{c^{(i)} e^{c^{(i)}\theta}}{1 + e^{c^{(i)}\theta}} = \frac{1}{m} \sum_{i=1}^m \frac{-c^{(i)}}{1 + e^{-c^{(i)}\theta}}$$

Since  $c^{(i)}\theta < 0$ ,  $1 + e^{-c^{(i)}\theta} > 0$ , we have  $\ell'(\theta) \neq 0$  for all  $\theta$ . This explains why there is no convergence. The plot of  $\ell'(\theta)$  should look something like the following. Note that when  $\theta < 0$ , we should have  $c^{(i)} > 0$  to make  $c^{(i)}\theta < 0$ , which is the lower part of Figure 1; when  $\theta > 0$ , we should have  $c^{(i)} < 0$  to make  $c^{(i)}\theta < 0$ , which is the upper part of Figure 1.

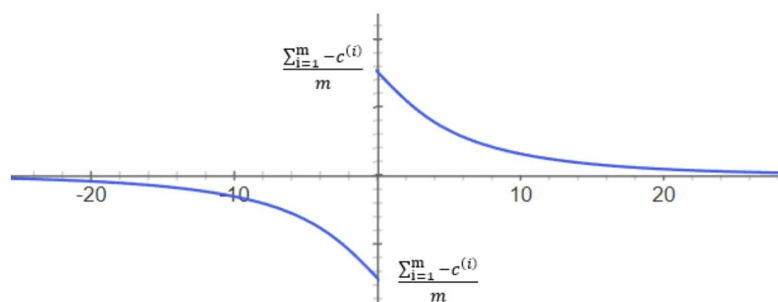


Figure 1, Gradient w.r.t.  $\theta$

- Now consider there is a misclassified point. Just one misclassified point will make  $\ell'(\theta) = 0$  solvable, hence the algorithm can converge. A misclassified point means in the assumption

1.2 b 5 / 5

- ✓ + **5 pts** Correct: linearly separable and explanation
- + **3 pts** Almost: linearly separable but insufficient explanation
- + **1 pts** Attempt
- + **0 pts** Blank
- + **0 pts** Failed to tag properly :(

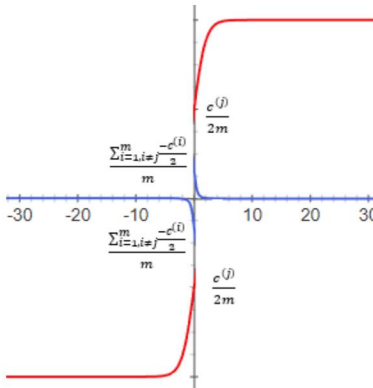


Figure 3, Not possible

It turns out that this is impossible so long as the data points are generated IID from the same distribution; there is always a solution for  $\ell'(\theta) = 0$  in this case.

- c) For each of these possible modifications, state whether it would lead to Convergence.
- Using a different constant learning rate.  
Answer:  
**No.** The problem is with the parameter being scaled infinitely to get a larger probability to get a maximum log-likelihood. Change of learning rate would not solve this problem.
  - Decreasing the learning rate over time.  
Answer:  
**No.** the answer is same as above.
  - Linear scaling of the input features.  
Answer:  
**No.** It has the same effect as to scale the parameters which doesn't help.
  - Adding a regularization term  $\|\theta\|_2^2$  to the loss function.  
Answer:  
**Yes.** L2 regularization will penalize large  $\theta$  hence it will prevent the parameters to scale infinitely when it tries to maximize the log-likelihood.
  - Adding zero-mean Gaussian noise to the training data or labels: will have misclassified points.  
Answer:  
**Yes.** It will add misclassified samples to the dataset which will make logistic regression converge.
- d) In SVM,  $\theta$  is going to be normalized and cannot be infinitely increased when trying to maximize the geometric margin; strict separation issue would no longer be a problem.



### 1.3 C 4.5 / 5

- **0 pts** Correct
- **1 pts** Part i incorrect
- **1 pts** Part ii incorrect
- **0.5 pts** Part ii insufficient justification (e.g. only addresses convergence as satisfying the stopping condition)
- **1 pts** Part iii incorrect
- **1 pts** Part iv incorrect
- **1 pts** Part v incorrect
- ✓ - **0.5 pts** Part v insufficient justification (e.g. assumes Gaussian noise makes data linearly inseparable)
- **5 pts** Blank or incomplete

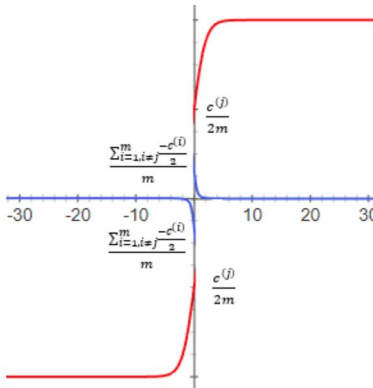


Figure 3, Not possible

It turns out that this is impossible so long as the data points are generated IID from the same distribution; there is always a solution for  $\ell'(\theta) = 0$  in this case.

- c) For each of these possible modifications, state whether it would lead to Convergence.
- Using a different constant learning rate.  
Answer:  
**No.** The problem is with the parameter being scaled infinitely to get a larger probability to get a maximum log-likelihood. Change of learning rate would not solve this problem.
  - Decreasing the learning rate over time.  
Answer:  
**No.** the answer is same as above.
  - Linear scaling of the input features.  
Answer:  
**No.** It has the same effect as to scale the parameters which doesn't help.
  - Adding a regularization term  $\|\theta\|_2^2$  to the loss function.  
Answer:  
**Yes.** L2 regularization will penalize large  $\theta$  hence it will prevent the parameters to scale infinitely when it tries to maximize the log-likelihood.
  - Adding zero-mean Gaussian noise to the training data or labels: will have misclassified points.  
Answer:  
**Yes.** It will add misclassified samples to the dataset which will make logistic regression converge.
- d) In SVM,  $\theta$  is going to be normalized and cannot be infinitely increased when trying to maximize the geometric margin; strict separation issue would no longer be a problem.

1.4 d 3 / 3

✓ + 3 pts Correct

+ 1 pts Incorrect answer, SVM is not vulnerable

+ 2.5 pts Insufficient explanation, not mentioning maximize geometric margin (hint)

+ 2.5 pts The geometric margin should be maximized instead of minimized

+ 0 pts No Answer

+ 1 pts No explanation

## 2. Model Calibration

- a) Prove calibration condition holds true for Logistic regression over range  $(a, b) = (0, 1)$ . Firstly, the log-likelihood of all points is given by

$$\ell(\theta) = \sum_1^m y^{(i)} \log p^{(i)} + (1 - y^{(i)}) \log(1 - p^{(i)})$$

The maximum likelihood parameter  $\theta$  is just the solution given by  $\nabla_{\theta} \ell(\theta) = \vec{0}$ . Note that  $p^{(i)} = g(z^{(i)})$ ,  $\nabla_{\theta} p^{(i)'} = g(z^{(i)}) (1 - g(z^{(i)})) x^{(i)}$

$$\begin{aligned} \nabla_{\theta} \ell(\theta) &= \sum_1^m (y^{(i)} p^{(i)} (1 - p^{(i)}) / p^{(i)} + (1 - y^{(i)}) (-p^{(i)}) (1 - p^{(i)}) / (1 - p^{(i)})) x^{(i)} \\ \nabla_{\theta} \ell(\theta) &= \sum_1^m (y^{(i)} (1 - p^{(i)}) + (1 - y^{(i)}) (-p^{(i)})) x^{(i)} = \vec{0} \Rightarrow \\ \sum_1^m y^{(i)} x^{(i)} &= \sum_1^m p^{(i)} x^{(i)} \end{aligned}$$

with  $y = [y^{(0)}, y^{(1)}, \dots, y^{(m)}]$ ,  $p = [p^{(0)}, p^{(1)}, \dots, p^{(m)}]$ . Use the fact that we include a bias term:  $x_0^{(i)} = 1$ , by observing the first component of  $\sum_1^m y^{(i)} x^{(i)}$  and  $\sum_1^m p^{(i)} x^{(i)}$ , we have:

$$\begin{aligned} \sum_1^m y^{(i)} &= \sum_1^m p^{(i)} \Rightarrow \\ \frac{\sum_{i \in I_{0,1}} 1\{y^{(i)} = 1\}}{|i \in I_{0,1}|} &= \frac{\sum_{i \in I_{0,1}} P(y^{(i)} = 1 | x^{(i)}; \theta)}{|i \in I_{0,1}|} \end{aligned}$$

- b) Perfect calibration doesn't mean perfect accuracy. If for any  $(a, b) \subset [0, 1]$  the property in the question holds true, consider two train examples with index  $i$  and  $j$ , they have different probability range  $h_{\theta}(x^{(i)}) \in (a^{(i)}, b^{(i)})$  and  $h_{\theta}(x^{(j)}) \in (a^{(j)}, b^{(j)})$ . By switching two samples' probability ranges, the perfect calibration still holds true, but the two training examples will have different probability ranges separately.

Conversely if the model achieves perfect accuracy, it is perfectly calibrated. This can be explained by clapping  $h_{\theta}(x^{(i)}) \in (a^{(i)}, b^{(i)})$  to the probability of every single training sample, the perfect calibration condition holds true always.

- c) What effect including L2 regularization in the logistic regression objective has on model calibration.

L2 regularization penalizes large value of  $\|\theta\|_2^2$ , which makes  $h_{\theta}(x) = \frac{1}{1 + \exp(-\theta^T x)}$  not able to grow infinitely to get a larger probability, which means the calibration range is smaller than  $(0, 1)$ .

2.1 a 5 / 5

✓ + 5 pts Correct

+ 4 pts Minor error

+ 4 pts Derivation not complete

+ 3 pts Major error

+ 2 pts Barely attempted

+ 0 pts Not attempted

## 2. Model Calibration

- a) Prove calibration condition holds true for Logistic regression over range  $(a, b) = (0, 1)$ . Firstly, the log-likelihood of all points is given by

$$\ell(\theta) = \sum_1^m y^{(i)} \log p^{(i)} + (1 - y^{(i)}) \log(1 - p^{(i)})$$

The maximum likelihood parameter  $\theta$  is just the solution given by  $\nabla_{\theta} \ell(\theta) = \vec{0}$ . Note that  $p^{(i)} = g(z^{(i)})$ ,  $\nabla_{\theta} p^{(i)'} = g(z^{(i)}) (1 - g(z^{(i)})) x^{(i)}$

$$\begin{aligned} \nabla_{\theta} \ell(\theta) &= \sum_1^m (y^{(i)} p^{(i)} (1 - p^{(i)}) / p^{(i)} + (1 - y^{(i)}) (-p^{(i)}) (1 - p^{(i)}) / (1 - p^{(i)})) x^{(i)} \\ \nabla_{\theta} \ell(\theta) &= \sum_1^m (y^{(i)} (1 - p^{(i)}) + (1 - y^{(i)}) (-p^{(i)})) x^{(i)} = \vec{0} \Rightarrow \\ \sum_1^m y^{(i)} x^{(i)} &= \sum_1^m p^{(i)} x^{(i)} \end{aligned}$$

with  $y = [y^{(0)}, y^{(1)}, \dots, y^{(m)}]$ ,  $p = [p^{(0)}, p^{(1)}, \dots, p^{(m)}]$ . Use the fact that we include a bias term:  $x_0^{(i)} = 1$ , by observing the first component of  $\sum_1^m y^{(i)} x^{(i)}$  and  $\sum_1^m p^{(i)} x^{(i)}$ , we have:

$$\begin{aligned} \sum_1^m y^{(i)} &= \sum_1^m p^{(i)} \Rightarrow \\ \frac{\sum_{i \in I_{0,1}} 1\{y^{(i)} = 1\}}{|i \in I_{0,1}|} &= \frac{\sum_{i \in I_{0,1}} P(y^{(i)} = 1 | x^{(i)}; \theta)}{|i \in I_{0,1}|} \end{aligned}$$

- b) Perfect calibration doesn't mean perfect accuracy. If for any  $(a, b) \subset [0, 1]$  the property in the question holds true, consider two train examples with index  $i$  and  $j$ , they have different probability range  $h_{\theta}(x^{(i)}) \in (a^{(i)}, b^{(i)})$  and  $h_{\theta}(x^{(j)}) \in (a^{(j)}, b^{(j)})$ . By switching two samples' probability ranges, the perfect calibration still holds true, but the two training examples will have different probability ranges separately.

Conversely if the model achieves perfect accuracy, it is perfectly calibrated. This can be explained by clapping  $h_{\theta}(x^{(i)}) \in (a^{(i)}, b^{(i)})$  to the probability of every single training sample, the perfect calibration condition holds true always.

- c) What effect including L2 regularization in the logistic regression objective has on model calibration.

L2 regularization penalizes large value of  $\|\theta\|_2^2$ , which makes  $h_{\theta}(x) = \frac{1}{1 + \exp(-\theta^T x)}$  not able to grow infinitely to get a larger probability, which means the calibration range is smaller than  $(0, 1)$ .

## 2.2 b 2 / 3

- + **0 pts** Not attempted
- + **3 pts** Correct
- + **1 pts** Both direction are not correct
- ✓ + **2 pts** One direction is correct
- + **1 pts** Conclusion without explanation
- + **2.5 pts** Minor errors
- + **1.5 pts** One direction wrong, one direction with minor error
- + **1 pts** Both directions are wrong

## 2. Model Calibration

- a) Prove calibration condition holds true for Logistic regression over range  $(a, b) = (0, 1)$ . Firstly, the log-likelihood of all points is given by

$$\ell(\theta) = \sum_1^m y^{(i)} \log p^{(i)} + (1 - y^{(i)}) \log(1 - p^{(i)})$$

The maximum likelihood parameter  $\theta$  is just the solution given by  $\nabla_{\theta} \ell(\theta) = \vec{0}$ . Note that  $p^{(i)} = g(z^{(i)})$ ,  $\nabla_{\theta} p^{(i)'} = g(z^{(i)}) (1 - g(z^{(i)})) x^{(i)}$

$$\begin{aligned} \nabla_{\theta} \ell(\theta) &= \sum_1^m (y^{(i)} p^{(i)} (1 - p^{(i)}) / p^{(i)} + (1 - y^{(i)}) (-p^{(i)}) (1 - p^{(i)}) / (1 - p^{(i)})) x^{(i)} \\ \nabla_{\theta} \ell(\theta) &= \sum_1^m (y^{(i)} (1 - p^{(i)}) + (1 - y^{(i)}) (-p^{(i)})) x^{(i)} = \vec{0} \Rightarrow \\ \sum_1^m y^{(i)} x^{(i)} &= \sum_1^m p^{(i)} x^{(i)} \end{aligned}$$

with  $y = [y^{(0)}, y^{(1)}, \dots, y^{(m)}]$ ,  $p = [p^{(0)}, p^{(1)}, \dots, p^{(m)}]$ . Use the fact that we include a bias term:  $x_0^{(i)} = 1$ , by observing the first component of  $\sum_1^m y^{(i)} x^{(i)}$  and  $\sum_1^m p^{(i)} x^{(i)}$ , we have:

$$\begin{aligned} \sum_1^m y^{(i)} &= \sum_1^m p^{(i)} \Rightarrow \\ \frac{\sum_{i \in I_{0,1}} 1\{y^{(i)} = 1\}}{|i \in I_{0,1}|} &= \frac{\sum_{i \in I_{0,1}} P(y^{(i)} = 1 | x^{(i)}; \theta)}{|i \in I_{0,1}|} \end{aligned}$$

- b) Perfect calibration doesn't mean perfect accuracy. If for any  $(a, b) \subset [0, 1]$  the property in the question holds true, consider two train examples with index  $i$  and  $j$ , they have different probability range  $h_{\theta}(x^{(i)}) \in (a^{(i)}, b^{(i)})$  and  $h_{\theta}(x^{(j)}) \in (a^{(j)}, b^{(j)})$ . By switching two samples' probability ranges, the perfect calibration still holds true, but the two training examples will have different probability ranges separately.

Conversely if the model achieves perfect accuracy, it is perfectly calibrated. This can be explained by clapping  $h_{\theta}(x^{(i)}) \in (a^{(i)}, b^{(i)})$  to the probability of every single training sample, the perfect calibration condition holds true always.

- c) What effect including L2 regularization in the logistic regression objective has on model calibration.

L2 regularization penalizes large value of  $\|\theta\|_2^2$ , which makes  $h_{\theta}(x) = \frac{1}{1 + \exp(-\theta^T x)}$  not able to grow infinitely to get a larger probability, which means the calibration range is smaller than  $(0, 1)$ .



**2.3 C 1 / 2**

- + **2 pts** Correct
- + **1 pts** Incorrect explanation of effect of sign of bias term
- ✓ + **1 pts** Incomplete/Incorrect explanation
- + **0 pts** Not attempted

### 3. Bayesian Interpretation of Regularization

- a) Show that  $\theta_{MAP} = \arg \max_{\theta} p(y|x, \theta)p(\theta)$  if we assume that  $p(\theta) = p(\theta|x)$ .

**PROOF:**

The posterior distribution of parameter  $\theta$  is:

$$\begin{aligned} p(\theta|x, y) &= \frac{p(y|x, \theta)p(\theta|x)}{p(y|x)} \\ &= \frac{p(y|x, \theta)p(\theta|x)}{\int p(y|x, \theta)p(\theta|x)d\theta} \end{aligned}$$

The first equality is based on Bayesian rule with a condition, the second equality is the marginal probability over  $\theta$  conditioned on  $x$ . The denominator of the second is independent on the parameter, so we have:

$$\begin{aligned} \theta_{MAP} &= \arg \max_{\theta} p(\theta|x, y) \\ &= \arg \max_{\theta} \frac{p(y|x, \theta)p(\theta|x)}{\int p(y|x, \theta)p(\theta|x)d\theta} \\ &= \arg \max_{\theta} p(y|x, \theta)p(\theta|x) \\ &= \arg \max_{\theta} p(y|x, \theta)p(\theta) \end{aligned}$$

Which is exactly what we need to prove.

- b) Show that MAP estimation with zero-mean Gaussian prior is equivalent to applying L2 regularization with MLE.

**PROOF:**

From above we have the MAP estimation given by:

$$\theta_{MAP} = \arg \max_{\theta} p(y|x, \theta)p(\theta)$$

Change the max log-likelihood estimation to min negative log-likelihood estimation, and use the fact the  $\theta$  is a zero mean Gaussian:

$$\begin{aligned} \theta_{MAP} &= \arg \min_{\theta} -\log(p(y|x, \theta)p(\theta)) \\ &= \arg \min_{\theta} -\log p(y|x, \theta) - \log \left( \frac{1}{\sqrt{2\pi\eta}} \exp \left( -\frac{\theta^T I \theta}{2\eta^2} \right) \right) \\ &= \arg \min_{\theta} -\log p(y|x, \theta) + \frac{\theta^T I \theta}{2\eta^2} \\ &= \arg \min_{\theta} -\log p(y|x, \theta) + \frac{\|\theta\|_2^2}{2\eta^2} \end{aligned}$$

From the last equality we have:

$$\lambda = \frac{1}{2\eta^2}$$

3.1 a 3 / 3

✓ + 3 pts Correct

+ 2 pts Partially correct with minor mistakes

+ 0 pts Incorrect or not attempted or incorrectly tagged

### 3. Bayesian Interpretation of Regularization

- a) Show that  $\theta_{MAP} = \arg \max_{\theta} p(y|x, \theta)p(\theta)$  if we assume that  $p(\theta) = p(\theta|x)$ .

**PROOF:**

The posterior distribution of parameter  $\theta$  is:

$$\begin{aligned} p(\theta|x, y) &= \frac{p(y|x, \theta)p(\theta|x)}{p(y|x)} \\ &= \frac{p(y|x, \theta)p(\theta|x)}{\int p(y|x, \theta)p(\theta|x)d\theta} \end{aligned}$$

The first equality is based on Bayesian rule with a condition, the second equality is the marginal probability over  $\theta$  conditioned on  $x$ . The denominator of the second is independent on the parameter, so we have:

$$\begin{aligned} \theta_{MAP} &= \arg \max_{\theta} p(\theta|x, y) \\ &= \arg \max_{\theta} \frac{p(y|x, \theta)p(\theta|x)}{\int p(y|x, \theta)p(\theta|x)d\theta} \\ &= \arg \max_{\theta} p(y|x, \theta)p(\theta|x) \\ &= \arg \max_{\theta} p(y|x, \theta)p(\theta) \end{aligned}$$

Which is exactly what we need to prove.

- b) Show that MAP estimation with zero-mean Gaussian prior is equivalent to applying L2 regularization with MLE.

**PROOF:**

From above we have the MAP estimation given by:

$$\theta_{MAP} = \arg \max_{\theta} p(y|x, \theta)p(\theta)$$

Change the max log-likelihood estimation to min negative log-likelihood estimation, and use the fact the  $\theta$  is a zero mean Gaussian:

$$\begin{aligned} \theta_{MAP} &= \arg \min_{\theta} -\log(p(y|x, \theta)p(\theta)) \\ &= \arg \min_{\theta} -\log p(y|x, \theta) - \log \left( \frac{1}{\sqrt{2\pi\eta}} \exp \left( -\frac{\theta^T I \theta}{2\eta^2} \right) \right) \\ &= \arg \min_{\theta} -\log p(y|x, \theta) + \frac{\theta^T I \theta}{2\eta^2} \\ &= \arg \min_{\theta} -\log p(y|x, \theta) + \frac{\|\theta\|_2^2}{2\eta^2} \end{aligned}$$

From the last equality we have:

$$\lambda = \frac{1}{2\eta^2}$$

3.2 b 4.75 / 5

✓ - 0 pts Correct

✓ - 0.25 pts Wrong multivariate Gaussian expression (constant of normalization)

- 1 pts Minor mistake in the proof
- 1 pts Too many steps skipped
- 2 pts Correct approach but wrong result
- 3 pts Correct approach but wrong proof
- 4 pts Wrong proof and result
- 5 pts No or wrong submission
- 1 pts Hard to read

- c) Closed form expression for  $\theta_{\text{MAP}}$  with Gaussian prior and Gaussian distributed dataset
- i. The Likelihood function is:

$$L(\theta) = \prod_{i=1}^m p(y^{(i)} | x^{(i)}, \theta) p(\theta) = \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2}\right) \frac{1}{\sqrt{2\pi}\eta} \exp\left(-\frac{\theta^T I \theta}{2\eta^2}\right)$$

- ii. The negative log-likelihood loss can then be derived from above:

$$\ell(\theta) = \sum_{i=1}^m \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2} + \frac{m}{2\eta^2} \theta^T I \theta - m \log(\sqrt{2\pi}\delta) - m \log(\sqrt{2\pi}\eta)$$

The MAP estimation is just the solution at the stationary point. Take derivative w.r.t.  $\theta$  and set it to  $\vec{0}$ :

$$\sum_{i=1}^m -\frac{(y - \theta^T x)x}{\delta^2} + \frac{m}{\eta^2} \theta = \vec{0}$$

Vectorize the above sum,  $X \in \mathbb{R}^{m \times n}$ ,  $\theta \in \mathbb{R}^n$ ,  $\vec{y} \in \mathbb{R}^m$ :

$$\begin{aligned} \frac{X^T(\vec{y} - X\theta)}{\delta^2} &= \frac{mI}{\eta^2} \theta \Rightarrow \frac{X^T \vec{y}}{\delta^2} = \left( \frac{X^T X}{\delta^2} + \frac{m}{\eta^2} I \right) \theta \Rightarrow \\ \theta &= \left( X^T X + \frac{\delta^2 m}{\eta^2} I \right)^{-1} X^T \vec{y} \end{aligned}$$

So the closed form  $\theta_{\text{MAP}} = \left( X^T X + \frac{\delta^2 m}{\eta^2} I \right)^{-1} X^T \vec{y}$ .

- d) Show that  $\theta_{\text{MAP}}$  in the case of Laplace prior is equivalent to the solution of a linear regression with L1 regularization.

Like c), the Likelihood function is given by

$$L(\theta) = \prod_{i=1}^m p(y^{(i)} | x^{(i)}, \theta) p(\theta) = \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2}\right) \frac{1}{2b} \exp\left(-\frac{|\theta|}{b}\right)$$

The negative log-likelihood loss is then:

$$J(\theta) = \sum_{i=1}^m \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2} + \frac{m}{b} |\theta|$$

Scale the loss doesn't change anything to the problem, so the loss can just be written as:

$$\begin{aligned} J(\theta) &= \sum_{i=1}^m (y^{(i)} - \theta^T x^{(i)})^2 + \frac{2\delta^2 m}{b} \|\theta\|_1 \\ &= (X\theta - \vec{y})^T (X\theta - \vec{y}) + \frac{2\delta^2 m}{b} \|\theta\|_1 \\ &= \|X\theta - \vec{y}\|_2^2 + \frac{2\delta^2 m}{b} \|\theta\|_1 \end{aligned}$$

Where  $\lambda = \frac{2\delta^2 m}{b}$

### 3.3 C 6 / 7

+ 7 pts Correct

✓ + 6 pts Minor Error

+ 5 pts Error

+ 4 pts Major error

+ 3 pts Did not solve for theta map

+ 2 pts Incomplete solution

+ 1 pts Attempted

+ 0 pts Missing

💬 There should be no m in the final expression

- c) Closed form expression for  $\theta_{\text{MAP}}$  with Gaussian prior and Gaussian distributed dataset
- i. The Likelihood function is:

$$L(\theta) = \prod_{i=1}^m p(y^{(i)} | x^{(i)}, \theta) p(\theta) = \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2}\right) \frac{1}{\sqrt{2\pi}\eta} \exp\left(-\frac{\theta^T I \theta}{2\eta^2}\right)$$

- ii. The negative log-likelihood loss can then be derived from above:

$$\ell(\theta) = \sum_{i=1}^m \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2} + \frac{m}{2\eta^2} \theta^T I \theta - m \log(\sqrt{2\pi}\delta) - m \log(\sqrt{2\pi}\eta)$$

The MAP estimation is just the solution at the stationary point. Take derivative w.r.t.  $\theta$  and set it to  $\vec{0}$ :

$$\sum_{i=1}^m -\frac{(y - \theta^T x)x}{\delta^2} + \frac{m}{\eta^2} \theta = \vec{0}$$

Vectorize the above sum,  $X \in \mathbb{R}^{m \times n}$ ,  $\theta \in \mathbb{R}^n$ ,  $\vec{y} \in \mathbb{R}^m$ :

$$\begin{aligned} \frac{X^T(\vec{y} - X\theta)}{\delta^2} &= \frac{mI}{\eta^2} \theta \Rightarrow \frac{X^T \vec{y}}{\delta^2} = \left( \frac{X^T X}{\delta^2} + \frac{m}{\eta^2} I \right) \theta \Rightarrow \\ \theta &= \left( X^T X + \frac{\delta^2 m}{\eta^2} I \right)^{-1} X^T \vec{y} \end{aligned}$$

So the closed form  $\theta_{\text{MAP}} = \left( X^T X + \frac{\delta^2 m}{\eta^2} I \right)^{-1} X^T \vec{y}$ .

- d) Show that  $\theta_{\text{MAP}}$  in the case of Laplace prior is equivalent to the solution of a linear regression with L1 regularization.

Like c), the Likelihood function is given by

$$L(\theta) = \prod_{i=1}^m p(y^{(i)} | x^{(i)}, \theta) p(\theta) = \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2}\right) \frac{1}{2b} \exp\left(-\frac{|\theta|}{b}\right)$$

The negative log-likelihood loss is then:

$$J(\theta) = \sum_{i=1}^m \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\delta^2} + \frac{m}{b} |\theta|$$

Scale the loss doesn't change anything to the problem, so the loss can just be written as:

$$\begin{aligned} J(\theta) &= \sum_{i=1}^m (y^{(i)} - \theta^T x^{(i)})^2 + \frac{2\delta^2 m}{b} \|\theta\|_1 \\ &= (X\theta - \vec{y})^T (X\theta - \vec{y}) + \frac{2\delta^2 m}{b} \|\theta\|_1 \\ &= \|X\theta - \vec{y}\|_2^2 + \frac{2\delta^2 m}{b} \|\theta\|_1 \end{aligned}$$

Where  $\lambda = \frac{2\delta^2 m}{b}$



### 3.4 d 3 / 5

- 0 pts Correct
- 2 pts Missing  $\sigma^2$  in expression for  $\gamma$
- ✓ - 2 pts **Wrong value for  $\gamma$  (or not reported)**
- 5 pts Unattempted
- 2 pts Does not match  $J(\theta)$  expression (argmax vs argmin)
- 1 pts Only shown for 1 example. Does not match  $J(\theta)$
- 1 pts Minor error

#### 4. Constructing kernels.

Apply Mercer's theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite. (For the ease of readability,  $K_1$  And  $K_1$  are the same thing in this answer.)

a)  $K(x, z) = K_1(x, z) + K_2(x, z)$ :

It's a kernel, it is obvious that if  $K_1$  and  $K_2$  are both symmetric and positive semi-definitive.

Frist,  $K$  is symmetric because, from:

$$\begin{aligned} K_{1ij} &= K_{1ji} \\ K_{2ij} &= K_{2ji} \\ \Rightarrow \\ K_{1ij} + K_{2ij} &= K_{1ji} + K_{2ji} \Rightarrow \\ K_{ij} &= K_{ji} \end{aligned}$$

Second,  $K$  is positive semi-definitive, because both  $K_1$  and  $K_2$  are PSD,  $K=K_1+K_2$  is also a PSD matrix:

$$z^T K z = z^T (K_1 + K_2) z = z^T K_1 z + z^T K_2 z \geq 0$$

b)  $K(x, z) = K_1(x, z) - K_2(x, z)$ :

Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.

For example,  $K_1 = I$ ,  $K_2 = 2I$  ( $I$  is identity matrix), they both PSDs, but  $K_1 - K_2 = -I$  is negative definite.

c)  $K(x, z) = a K_1(x, z)$ :

It is a kernel only when  $a \geq 0$ . If  $a < 0$ ,  $K_1 = I$ , then  $aK_1(x, z)$  is negative definitive.

d)  $K(x, z) = -a K_1(x, z)$ :

Similar to c), It is a kernel only when  $a \leq 0$ . For example,  $a = 1$ ,  $K_1 = I$ , we have  $z^T K_1 z = -az^T z \leq 0$

e)  $K(x, z) = K_1(x, z)K_2(x, z)$ :

It is a kernel since because,

$$\begin{aligned} K_1(x, z)K_2(x, z) &= K_2^T(x, z)K_1^T(x, z) = K_2(x, z)K_1(x, z) \\ K_{ij} &= \sum_k K_{1ik}K_{2kj} \\ K_{ji} &= \sum_k K_{1jk}K_{2ki} = \sum_k K_{1kj}K_{2ik} = \sum_k K_{2ik}K_{1kj} \end{aligned}$$

Summarize the above 3 equations,  $K_{ij} = K_{ji}$  which means  $K$  is symmetric;

since both  $K_1$  and  $K_2$  are PSD, for any given vector  $x$ , we have  $x^T K_1 x \geq 0$ ,  $x^T K_2 x \geq 0$ , it implies that:

$$\begin{aligned} x^T K_1 x x^T K_2 x &\geq 0 \\ \Rightarrow x^T K_1 K_2 x &\geq 0 \end{aligned}$$

f)  $K(x, z) = f(x)f(z)$ :

It is a kernel. From class we know that for a feature mapping  $\phi(x)$ ,  $K(x, z) = \phi(x)^T \phi(z)$  is a kernel. In this case we simply can set  $\phi(x) = f(x)$ , it's just a 1d mapping.

g)  $K(x, z) = K_3(\phi(x), \phi(z))$ :

It is a kernel. Since  $K_3$  is a Kernel,  $K(x, z) = K_3(\phi(x), \phi(z)) = \psi(\phi(x))^T \psi(\phi(z)) = (\psi \circ \phi(x))^T (\psi \circ \phi(z))$  for a feature mapping  $\psi \circ \phi$ .

4.1 a 1 / 1

✓ + 1 pts correct with proof

+ 0 pts Incorrect or missing/incorrect proof

+ 0 pts Tag the pages!

#### 4. Constructing kernels.

Apply Mercer's theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite. (For the ease of readability,  $K_1$  And  $K_2$  are the same thing in this answer.)

a)  $K(x, z) = K_1(x, z) + K_2(x, z)$ :

It's a kernel, it is obvious that if  $K_1$  and  $K_2$  are both symmetric and positive semi-definitive.

Frist,  $K$  is symmetric because, from:

$$\begin{aligned} K_{1ij} &= K_{1ji} \\ K_{2ij} &= K_{2ji} \\ \Rightarrow \\ K_{1ij} + K_{2ij} &= K_{1ji} + K_{2ji} \Rightarrow \\ K_{ij} &= K_{ji} \end{aligned}$$

Second,  $K$  is positive semi-definitive, because both  $K_1$  and  $K_2$  are PSD,  $K=K_1+K_2$  is also a PSD matrix:

$$z^T K z = z^T (K_1 + K_2) z = z^T K_1 z + z^T K_2 z \geq 0$$

b)  $K(x, z) = K_1(x, z) - K_2(x, z)$ :

Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.

For example,  $K_1 = I$ ,  $K_2 = 2I$  ( $I$  is identity matrix), they both PSDs, but  $K_1 - K_2 = -I$  is negative definite.

c)  $K(x, z) = a K_1(x, z)$ :

It is a kernel only when  $a \geq 0$ . If  $a < 0$ ,  $K_1 = I$ , then  $aK_1(x, z)$  is negative definitive.

d)  $K(x, z) = -a K_1(x, z)$ :

Similar to c), It is a kernel only when  $a \leq 0$ . For example,  $a = 1$ ,  $K_1 = I$ , we have  $z^T K_1 z = -az^T z \leq 0$

e)  $K(x, z) = K_1(x, z)K_2(x, z)$ :

It is a kernel since because,

$$\begin{aligned} K_1(x, z)K_2(x, z) &= K_2^T(x, z)K_1^T(x, z) = K_2(x, z)K_1(x, z) \\ K_{ij} &= \sum_k K_{1ik}K_{2kj} \\ K_{ji} &= \sum_k K_{1jk}K_{2ki} = \sum_k K_{1kj}K_{2ik} = \sum_k K_{2ik}K_{1kj} \end{aligned}$$

Summarize the above 3 equations,  $K_{ij} = K_{ji}$  which means  $K$  is symmetric;

since both  $K_1$  and  $K_2$  are PSD, for any given vector  $x$ , we have  $x^T K_1 x \geq 0$ ,  $x^T K_2 x \geq 0$ , it implies that:

$$\begin{aligned} x^T K_1 x x^T K_2 x &\geq 0 \\ \Rightarrow x^T K_1 K_2 x &\geq 0 \end{aligned}$$

f)  $K(x, z) = f(x)f(z)$ :

It is a kernel. From class we know that for a feature mapping  $\phi(x)$ ,  $K(x, z) = \phi(x)^T \phi(z)$  is a kernel. In this case we simply can set  $\phi(x) = f(x)$ , it's just a 1d mapping.

g)  $K(x, z) = K_3(\phi(x), \phi(z))$ :

It is a kernel. Since  $K_3$  is a Kernel,  $K(x, z) = K_3(\phi(x), \phi(z)) = \psi(\phi(x))^T \psi(\phi(z)) = (\psi \circ \phi(x))^T (\psi \circ \phi(z))$  for a feature mapping  $\psi \circ \phi$ .

4.2 b 1 / 1

✓ + 1 pts Correct with proof

+ 0 pts Incorrect or missing/incorrect proof

+ 0 pts Incorrect Tags/Illegible scans

#### 4. Constructing kernels.

Apply Mercer's theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite. (For the ease of readability,  $K_1$  And  $K_2$  are the same thing in this answer.)

a)  $K(x, z) = K_1(x, z) + K_2(x, z)$ :

It's a kernel, it is obvious that if  $K_1$  and  $K_2$  are both symmetric and positive semi-definitive.

Frist,  $K$  is symmetric because, from:

$$\begin{aligned} K_{1ij} &= K_{1ji} \\ K_{2ij} &= K_{2ji} \\ \Rightarrow \\ K_{1ij} + K_{2ij} &= K_{1ji} + K_{2ji} \Rightarrow \\ K_{ij} &= K_{ji} \end{aligned}$$

Second,  $K$  is positive semi-definitive, because both  $K_1$  and  $K_2$  are PSD,  $K=K_1+K_2$  is also a PSD matrix:

$$z^T K z = z^T (K_1 + K_2) z = z^T K_1 z + z^T K_2 z \geq 0$$

b)  $K(x, z) = K_1(x, z) - K_2(x, z)$ :

Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.

For example,  $K_1 = I$ ,  $K_2 = 2I$  ( $I$  is identity matrix), they both PSDs, but  $K_1 - K_2 = -I$  is negative definite.

c)  $K(x, z) = a K_1(x, z)$ :

It is a kernel only when  $a \geq 0$ . If  $a < 0$ ,  $K_1 = I$ , then  $aK_1(x, z)$  is negative definite.

d)  $K(x, z) = -a K_1(x, z)$ :

Similar to c), It is a kernel only when  $a \leq 0$ . For example,  $a = 1$ ,  $K_1 = I$ , we have  $z^T K_1 z = -az^T z \leq 0$

e)  $K(x, z) = K_1(x, z)K_2(x, z)$ :

It is a kernel since because,

$$\begin{aligned} K_1(x, z)K_2(x, z) &= K_2^T(x, z)K_1^T(x, z) = K_2(x, z)K_1(x, z) \\ K_{ij} &= \sum_k K_{1ik}K_{2kj} \\ K_{ji} &= \sum_k K_{1jk}K_{2ki} = \sum_k K_{1kj}K_{2ik} = \sum_k K_{2ik}K_{1kj} \end{aligned}$$

Summarize the above 3 equations,  $K_{ij} = K_{ji}$  which means  $K$  is symmetric;

since both  $K_1$  and  $K_2$  are PSD, for any given vector  $x$ , we have  $x^T K_1 x \geq 0$ ,  $x^T K_2 x \geq 0$ , it implies that:

$$\begin{aligned} x^T K_1 x x^T K_2 x &\geq 0 \\ \Rightarrow x^T K_1 K_2 x &\geq 0 \end{aligned}$$

f)  $K(x, z) = f(x)f(z)$ :

It is a kernel. From class we know that for a feature mapping  $\phi(x)$ ,  $K(x, z) = \phi(x)^T \phi(z)$  is a kernel. In this case we simply can set  $\phi(x) = f(x)$ , it's just a 1d mapping.

g)  $K(x, z) = K_3(\phi(x), \phi(z))$ :

It is a kernel. Since  $K_3$  is a Kernel,  $K(x, z) = K_3(\phi(x), \phi(z)) = \psi(\phi(x))^T \psi(\phi(z)) = (\psi \circ \phi(x))^T (\psi \circ \phi(z))$  for a feature mapping  $\psi \circ \phi$ .

4.3 C 1 / 1

✓ + 1 pts Correct with proof.

+ 0 pts Incorrect or missing/incorrect proof

+ 0 pts Incorrect Tags/Illegible scans

#### 4. Constructing kernels.

Apply Mercer's theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite. (For the ease of readability,  $K_1$  And  $K_1$  are the same thing in this answer.)

a)  $K(x, z) = K_1(x, z) + K_2(x, z)$ :

It's a kernel, it is obvious that if  $K_1$  and  $K_2$  are both symmetric and positive semi-definitive.

Frist,  $K$  is symmetric because, from:

$$\begin{aligned} K_{1ij} &= K_{1ji} \\ K_{2ij} &= K_{2ji} \\ \Rightarrow \\ K_{1ij} + K_{2ij} &= K_{1ji} + K_{2ji} \Rightarrow \\ K_{ij} &= K_{ji} \end{aligned}$$

Second,  $K$  is positive semi-definitive, because both  $K_1$  and  $K_2$  are PSD,  $K=K_1+K_2$  is also a PSD matrix:

$$z^T K z = z^T (K_1 + K_2) z = z^T K_1 z + z^T K_2 z \geq 0$$

b)  $K(x, z) = K_1(x, z) - K_2(x, z)$ :

Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.

For example,  $K_1 = I$ ,  $K_2 = 2I$  ( $I$  is identity matrix), they both PSDs, but  $K_1 - K_2 = -I$  is negative definite.

c)  $K(x, z) = a K_1(x, z)$ :

It is a kernel only when  $a \geq 0$ . If  $a < 0$ ,  $K_1 = I$ , then  $aK_1(x, z)$  is negative definite.

d)  $K(x, z) = -a K_1(x, z)$ :

Similar to c), It is a kernel only when  $a \leq 0$ . For example,  $a = 1$ ,  $K_1 = I$ , we have  $z^T K_1 z = -az^T z \leq 0$

e)  $K(x, z) = K_1(x, z)K_2(x, z)$ :

It is a kernel since because,

$$\begin{aligned} K_1(x, z)K_2(x, z) &= K_2^T(x, z)K_1^T(x, z) = K_2(x, z)K_1(x, z) \\ K_{ij} &= \sum_k K_{1ik}K_{2kj} \\ K_{ji} &= \sum_k K_{1jk}K_{2ki} = \sum_k K_{1kj}K_{2ik} = \sum_k K_{2ik}K_{1kj} \end{aligned}$$

Summarize the above 3 equations,  $K_{ij} = K_{ji}$  which means  $K$  is symmetric;

since both  $K_1$  and  $K_2$  are PSD, for any given vector  $x$ , we have  $x^T K_1 x \geq 0$ ,  $x^T K_2 x \geq 0$ , it implies that:

$$\begin{aligned} x^T K_1 x x^T K_2 x &\geq 0 \\ \Rightarrow x^T K_1 K_2 x &\geq 0 \end{aligned}$$

f)  $K(x, z) = f(x)f(z)$ :

It is a kernel. From class we know that for a feature mapping  $\phi(x)$ ,  $K(x, z) = \phi(x)^T \phi(z)$  is a kernel. In this case we simply can set  $\phi(x) = f(x)$ , it's just a 1d mapping.

g)  $K(x, z) = K_3(\phi(x), \phi(z))$ :

It is a kernel. Since  $K_3$  is a Kernel,  $K(x, z) = K_3(\phi(x), \phi(z)) = \psi(\phi(x))^T \psi(\phi(z)) = (\psi \circ \phi(x))^T (\psi \circ \phi(z))$  for a feature mapping  $\psi \circ \phi$ .



4.4 d 1 / 1

✓ + 1 pts Correct with proof

+ 0 pts Incorrect or missing/incorrect proof

+ 0 pts Incorrect Tags/Illegible scans

#### 4. Constructing kernels.

Apply Mercer's theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite. (For the ease of readability,  $K_1$  And  $K_1$  are the same thing in this answer.)

a)  $K(x, z) = K_1(x, z) + K_2(x, z)$ :

It's a kernel, it is obvious that if  $K_1$  and  $K_2$  are both symmetric and positive semi-definitive.

Frist,  $K$  is symmetric because, from:

$$\begin{aligned} K_{1ij} &= K_{1ji} \\ K_{2ij} &= K_{2ji} \\ \Rightarrow \\ K_{1ij} + K_{2ij} &= K_{1ji} + K_{2ji} \Rightarrow \\ K_{ij} &= K_{ji} \end{aligned}$$

Second,  $K$  is positive semi-definitive, because both  $K_1$  and  $K_2$  are PSD,  $K=K_1+K_2$  is also a PSD matrix:

$$z^T K z = z^T (K_1 + K_2) z = z^T K_1 z + z^T K_2 z \geq 0$$

b)  $K(x, z) = K_1(x, z) - K_2(x, z)$ :

Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.

For example,  $K_1 = I$ ,  $K_2 = 2I$  ( $I$  is identity matrix), they both PSDs, but  $K_1 - K_2 = -I$  is negative definite.

c)  $K(x, z) = a K_1(x, z)$ :

It is a kernel only when  $a \geq 0$ . If  $a < 0$ ,  $K_1 = I$ , then  $aK_1(x, z)$  is negative definitive.

d)  $K(x, z) = -a K_1(x, z)$ :

Similar to c), It is a kernel only when  $a \leq 0$ . For example,  $a = 1$ ,  $K_1 = I$ , we have  $z^T K_1 z = -az^T z \leq 0$

e)  $K(x, z) = K_1(x, z)K_2(x, z)$ :

It is a kernel since because,

$$\begin{aligned} K_1(x, z)K_2(x, z) &= K_2^T(x, z)K_1^T(x, z) = K_2(x, z)K_1(x, z) \\ K_{ij} &= \sum_k K_{1ik}K_{2kj} \\ K_{ji} &= \sum_k K_{1jk}K_{2ki} = \sum_k K_{1kj}K_{2ik} = \sum_k K_{2ik}K_{1kj} \end{aligned}$$

Summarize the above 3 equations,  $K_{ij} = K_{ji}$  which means  $K$  is symmetric;

since both  $K_1$  and  $K_2$  are PSD, for any given vector  $x$ , we have  $x^T K_1 x \geq 0$ ,  $x^T K_2 x \geq 0$ , it implies that:

$$\begin{aligned} x^T K_1 x x^T K_2 x &\geq 0 \\ \Rightarrow x^T K_1 K_2 x &\geq 0 \end{aligned}$$

f)  $K(x, z) = f(x)f(z)$ :

It is a kernel. From class we know that for a feature mapping  $\phi(x)$ ,  $K(x, z) = \phi(x)^T \phi(z)$  is a kernel. In this case we simply can set  $\phi(x) = f(x)$ , it's just a 1d mapping.

g)  $K(x, z) = K_3(\phi(x), \phi(z))$ :

It is a kernel. Since  $K_3$  is a Kernel,  $K(x, z) = K_3(\phi(x), \phi(z)) = \psi(\phi(x))^T \psi(\phi(z)) = (\psi \circ \phi(x))^T (\psi \circ \phi(z))$  for a feature mapping  $\psi \circ \phi$ .

4.5 e 2 / 5

+ 5 pts Correct with proof

✓ + 3 pts incomplete or partially correct proof

+ 1 pts correct answer, but missing/incorrect proof

+ 0 pts missing/incorrect

+ 2 pts  $x^T x$  is not a invertible matrix

- 1 Point adjustment

💬  $x x^T$  is not identity.

#### 4. Constructing kernels.

Apply Mercer's theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite. (For the ease of readability,  $K_1$  And  $K_1$  are the same thing in this answer.)

a)  $K(x, z) = K_1(x, z) + K_2(x, z)$ :

It's a kernel, it is obvious that if  $K_1$  and  $K_2$  are both symmetric and positive semi-definitive.

Frist,  $K$  is symmetric because, from:

$$\begin{aligned} K_{1ij} &= K_{1ji} \\ K_{2ij} &= K_{2ji} \\ \Rightarrow \\ K_{1ij} + K_{2ij} &= K_{1ji} + K_{2ji} \Rightarrow \\ K_{ij} &= K_{ji} \end{aligned}$$

Second,  $K$  is positive semi-definitive, because both  $K_1$  and  $K_2$  are PSD,  $K=K_1+K_2$  is also a PSD matrix:

$$z^T K z = z^T (K_1 + K_2) z = z^T K_1 z + z^T K_2 z \geq 0$$

b)  $K(x, z) = K_1(x, z) - K_2(x, z)$ :

Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.

For example,  $K_1 = I$ ,  $K_2 = 2I$  ( $I$  is identity matrix), they both PSDs, but  $K_1 - K_2 = -I$  is negative definite.

c)  $K(x, z) = a K_1(x, z)$ :

It is a kernel only when  $a \geq 0$ . If  $a < 0$ ,  $K_1 = I$ , then  $aK_1(x, z)$  is negative definitive.

d)  $K(x, z) = -a K_1(x, z)$ :

Similar to c), It is a kernel only when  $a \leq 0$ . For example,  $a = 1$ ,  $K_1 = I$ , we have  $z^T K_1 z = -az^T z \leq 0$

e)  $K(x, z) = K_1(x, z)K_2(x, z)$ :

It is a kernel since because,

$$\begin{aligned} K_1(x, z)K_2(x, z) &= K_2^T(x, z)K_1^T(x, z) = K_2(x, z)K_1(x, z) \\ K_{ij} &= \sum_k K_{1ik}K_{2kj} \\ K_{ji} &= \sum_k K_{1jk}K_{2ki} = \sum_k K_{1kj}K_{2ik} = \sum_k K_{2ik}K_{1kj} \end{aligned}$$

Summarize the above 3 equations,  $K_{ij} = K_{ji}$  which means  $K$  is symmetric;

since both  $K_1$  and  $K_2$  are PSD, for any given vector  $x$ , we have  $x^T K_1 x \geq 0$ ,  $x^T K_2 x \geq 0$ , it implies that:

$$\begin{aligned} x^T K_1 x x^T K_2 x &\geq 0 \\ \Rightarrow x^T K_1 K_2 x &\geq 0 \end{aligned}$$

f)  $K(x, z) = f(x)f(z)$ :

It is a kernel. From class we know that for a feature mapping  $\phi(x)$ ,  $K(x, z) = \phi(x)^T \phi(z)$  is a kernel. In this case we simply can set  $\phi(x) = f(x)$ , it's just a 1d mapping.

g)  $K(x, z) = K_3(\phi(x), \phi(z))$ :

It is a kernel. Since  $K_3$  is a Kernel,  $K(x, z) = K_3(\phi(x), \phi(z)) = \psi(\phi(x))^T \psi(\phi(z)) = (\psi \circ \phi(x))^T (\psi \circ \phi(z))$  for a feature mapping  $\psi \circ \phi$ .

4.6 f 3 / 3

✓ + 3 pts Correct with proof

+ 1 pts Correct answer, missing/incorrect proof

+ 0 pts Incorrect/missing

#### 4. Constructing kernels.

Apply Mercer's theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite. (For the ease of readability,  $K_1$  And  $K_1$  are the same thing in this answer.)

a)  $K(x, z) = K_1(x, z) + K_2(x, z)$ :

It's a kernel, it is obvious that if  $K_1$  and  $K_2$  are both symmetric and positive semi-definitive.

Frist,  $K$  is symmetric because, from:

$$\begin{aligned} K_{1ij} &= K_{1ji} \\ K_{2ij} &= K_{2ji} \\ \Rightarrow \\ K_{1ij} + K_{2ij} &= K_{1ji} + K_{2ji} \Rightarrow \\ K_{ij} &= K_{ji} \end{aligned}$$

Second,  $K$  is positive semi-definitive, because both  $K_1$  and  $K_2$  are PSD,  $K=K_1+K_2$  is also a PSD matrix:

$$z^T K z = z^T (K_1 + K_2) z = z^T K_1 z + z^T K_2 z \geq 0$$

b)  $K(x, z) = K_1(x, z) - K_2(x, z)$ :

Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.

For example,  $K_1 = I$ ,  $K_2 = 2I$  ( $I$  is identity matrix), they both PSDs, but  $K_1 - K_2 = -I$  is negative definite.

c)  $K(x, z) = a K_1(x, z)$ :

It is a kernel only when  $a \geq 0$ . If  $a < 0$ ,  $K_1 = I$ , then  $aK_1(x, z)$  is negative definitive.

d)  $K(x, z) = -a K_1(x, z)$ :

Similar to c), It is a kernel only when  $a \leq 0$ . For example,  $a = 1$ ,  $K_1 = I$ , we have  $z^T K_1 z = -az^T z \leq 0$

e)  $K(x, z) = K_1(x, z)K_2(x, z)$ :

It is a kernel since because,

$$\begin{aligned} K_1(x, z)K_2(x, z) &= K_2^T(x, z)K_1^T(x, z) = K_2(x, z)K_1(x, z) \\ K_{ij} &= \sum_k K_{1ik}K_{2kj} \\ K_{ji} &= \sum_k K_{1jk}K_{2ki} = \sum_k K_{1kj}K_{2ik} = \sum_k K_{2ik}K_{1kj} \end{aligned}$$

Summarize the above 3 equations,  $K_{ij} = K_{ji}$  which means  $K$  is symmetric;

since both  $K_1$  and  $K_2$  are PSD, for any given vector  $x$ , we have  $x^T K_1 x \geq 0$ ,  $x^T K_2 x \geq 0$ , it implies that:

$$\begin{aligned} x^T K_1 x x^T K_2 x &\geq 0 \\ \Rightarrow x^T K_1 K_2 x &\geq 0 \end{aligned}$$

f)  $K(x, z) = f(x)f(z)$ :

It is a kernel. From class we know that for a feature mapping  $\phi(x)$ ,  $K(x, z) = \phi(x)^T \phi(z)$  is a kernel. In this case we simply can set  $\phi(x) = f(x)$ , it's just a 1d mapping.

g)  $K(x, z) = K_3(\phi(x), \phi(z))$ :

It is a kernel. Since  $K_3$  is a Kernel,  $K(x, z) = K_3(\phi(x), \phi(z)) = \psi(\phi(x))^T \psi(\phi(z)) = (\psi \circ \phi(x))^T (\psi \circ \phi(z))$  for a feature mapping  $\psi \circ \phi$ .

4.7 g 3 / 3

✓ + 3 pts Correct with proof

+ 1 pts correct answer, missing/incorrect proof

+ 0 pts Incorrect/missing

+ 0 pts [Click here to replace this description.](#)

h)  $K(x, z) = p(K_1(x, z))$ :

It is a kernel. First,  $K_1^n(x, z)$  is still a kernel;  $aK_1(x, z)$  is still a kernel when  $a \geq 0$ ,  $K_1(x, z) + c$  is a kernel if  $c \geq 0$ ; combine them together we have that  $p(K_1(x, z))$  is still a kernel.



4.8 h 3 / 3

✓ - **0 pts** Correct

- **3 pts** Incorrect/ No submission

- **1 pts** Minor mistake

- **2 pts** Incorrect proof

- **0 pts** Please tag your pages

## 5. Kernelizing the Perceptron

a) Kernel trick questions.

i. How you would apply the “kernel trick” to the perceptron

Answer:

Observe the update rule, the series of  $\theta^{(i)}$  is given by:

$$\theta^{(i+1)} = \theta^{(i)} + \alpha \left( y^{(i+1)} - g \left( \theta^{(i)} \phi(x^{(i+1)}) \right) \right) \phi(x^{(i+1)})$$

Since  $\theta$  is infinite dimensional and cannot be represented by computer, the relationship between two successive  $\theta^{(i)}$  and  $\theta^{(i+1)}$  however can be calculated iteratively, the values are just scalars which are denoted by  $\beta^{(i)}$  given by:

$$\beta^{(1)} = \alpha \left( y^{(1)} - g \left( \theta^{(0)} \phi(x^{(1)}) \right) \right), \theta^{(0)} \phi(x^{(1)}) = (\theta^{(0)}) \phi(x^{(1)}) = 0$$

$$\beta^{(2)} = \alpha \left( y^{(2)} - g \left( \theta^{(1)} \phi(x^{(2)}) \right) \right), \theta^{(1)} \phi(x^{(2)}) = \left( \theta^{(0)} + \beta^{(1)} \phi(x^{(1)}) \right) \phi(x^{(2)})$$

$$\beta^{(3)} = \alpha \left( y^{(3)} - g \left( \theta^{(2)} \phi(x^{(3)}) \right) \right), \theta^{(2)} \phi(x^{(3)}) = \left( \theta^{(0)} + \beta^{(1)} \phi(x^{(1)}) + \beta^{(2)} \phi(x^{(2)}) \right) \phi(x^{(3)})$$

...

$$\begin{aligned} \beta^{(i+1)} &= \alpha \left( y^{(i+1)} - g \left( \theta^{(i)} \phi(x^{(i+1)}) \right) \right), \theta^{(i)} \phi(x^{(i+1)}) \\ &= \left( \theta^{(0)} + \beta^{(1)} \phi(x^{(1)}) + \beta^{(2)} \phi(x^{(2)}) + \dots + \beta^{(i)} \phi(x^{(i)}) \right) \phi(x^{(i+1)}) \end{aligned}$$

We can see that  $\beta^{(i+1)} = \sum_{k=1}^i \beta^{(k)} \phi(x^{(k)}) \phi(x^{(i+1)}) = \sum_{k=1}^i \beta^{(k)} \text{Kernel}(x^{(k)}, x^{(i+1)})$

We do not need to store any  $\theta^{(i)}$  value, including  $\theta^{(0)}$ , only need to store the array of  $\beta^{(i)}$ s. Whenever there is a need for  $\theta^{(i)}$ , it is just used to calculate the margin  $\theta^{(i)} \phi(x)$  which can just be written as  $\sum_{k=1}^i \beta^{(k)} \text{Kernel}(x^{(k)}, x)$ . For  $\theta^{(0)}$  the margin is just  $\theta^{(0)} \phi(x) = 0$ .

ii. How you will efficiently make a prediction on a new input  $x^{(i+1)}$ .

Answer:

As described in i), we only need to figure out the margin value  $\theta^{(i)} \phi(x^{(i+1)})$  which is just given by:

$$z = \theta^{(i)} \phi(x^{(i+1)}) = \sum_{k=1}^i \beta^{(k)} \text{Kernel}(x^{(k)}, x^{(i+1)})$$

And use  $g(z) = \text{sign}(z)$  to make a prediction.

iii. How you will modify the update rule given above to perform an update to  $\theta$

Answer:

As described in i),  $\theta$  is never explicitly represented by computer, only the linear coefficients  $\beta^{(i)}$ s are stored; The learning update will only update the coefficients. Whenever the calculation of margins needed, the coefficients are used to calculate margins:

$$z = \theta^{(i)} \phi(x) = \sum_{k=1}^i \beta^{(k)} \text{Kernel}(x^{(k)}, x)$$

5.1 a 8 / 9

+ 3 pts i) Correct

✓ + 2 pts i) partial soln, missing initial value or linear representation

+ 1 pts i) partial soln

+ 0 pts i) Incorrect

✓ + 3 pts ii) correct

+ 2 pts ii) missing kernel representation/ partial soln

+ 1 pts ii) partial solution

✓ + 3 pts iii) correct

+ 2 pts iii) Partial soln / missing full reasoning

+ 1 pts iii) partial solution

+ 0 pts iii) Blank/Incorrect

+ 0 pts Wrong solution/ unattempted

+ 0 pts ii) blank/incorrect

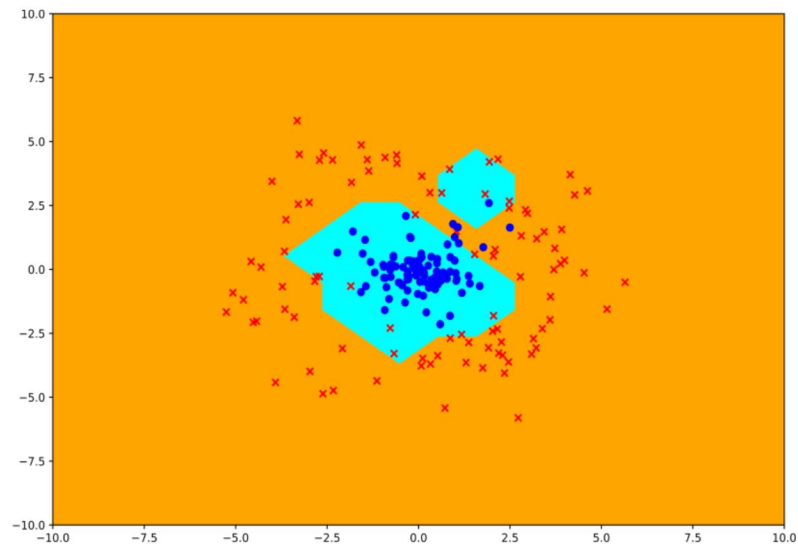
b) Implement kernelized perceptron.

Answer:

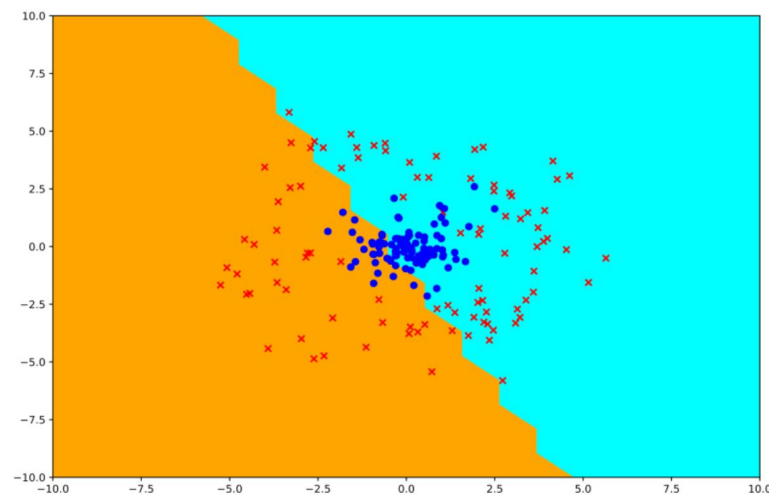
c) Which kernel performs badly and why.

Answer:

From the plots we can see that RBT kernels makes pretty decent predictions and indeed maps to higher dimensional features;



However the dot product kernel cannot make good prediction; It divides the sample space into two halves with a wavy line.



The reason for this is because the feature map doesn't contain multiple degree monomials, which means it cannot regress complex decision boundary.

5.2 C 2 / 2

✓ + 2 pts Correct

+ 1 pts Only correctly mentioned that dot product perform poorly

+ 0 pts Blank

+ 0 pts Incorrect