6.036: Midterm, Spring 2018

Solutions

- This is a closed book exam. Calculators not permitted.
- The problems are not necessarily in any order of difficulty.
- Record all your answers in the places provided. If you run out of room for an answer, continue on a blank page and mark it clearly.
- If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.
- **Write your name on every page.**
- Come to the front to ask questions.

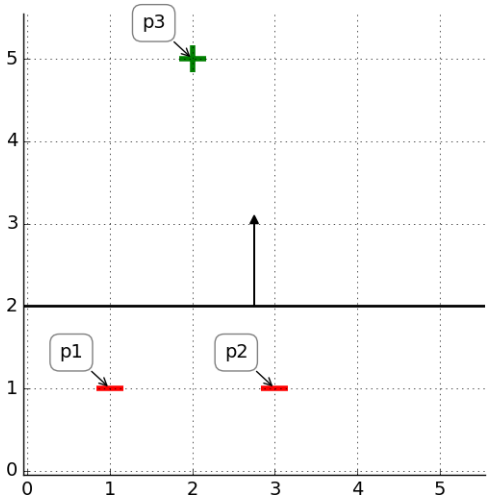
Name: _____ Athena ID: _____

Question	Points	Score
1	20	
2	10	
3	18	
4	28	
5	14	
6	10	
Total:	100	

Name: _____

Margin

1. (20 points) Here are some points and a hypothesis.



(a) Give parameters θ and θ_0 for the separator, such that $\|\theta\| = 1$.

i. θ: $[0, 1]^T$

ii. θ_0: -2

Solution: We can determine from the graph that $\theta = [0, 1]^T$ and $\theta_0 = -2$, since we are told $||\theta|| = 1$.

(b) What is the margin of each point with respect to that separator?

i. Point 1: **1**

ii. Point 2: **1**

iii. Point 3: **3**

Solution: We can determine from the graph that p1 and p2 are a perpendicular distance of 1 away from the separator, while p3 is a perpendicular distance of 3 away.

(c) What would be the next values of θ and θ_0 after one step of batch gradient descent where the objective is

$$J(\theta, \theta_0) = \frac{1}{3}\left(\sum_{i=1}^{3} L_{hinge}(y^{(i)}(\theta^T x^{(i)} + \theta_0))\right) + \lambda\|\theta\|^2,$$

with $\lambda = 0$ and step size $= 1$?

i. θ: $[0, 1]^T$

ii. θ_0: -2

Solution: There is no regularization, so the objective is solely the loss term. Recall that the derivative of hinge loss is 0 when $y(\theta^T x + \theta_0) \geq 1$. Since $y(\theta^T x + \theta_0) \geq 1$ for all three points no update occurs.

(d) What is the margin of the *whole data set* with respect to the new separator? Your answer should be a single number (or fraction).

1

Solution: The margin of the whole dataset is determined by the point or points closest to the separator. The smallest margin is 1, so the margin of the whole dataset is 1.

(e) What would be the next values of θ and θ_0 after one step of batch gradient descent where the objective is

$$J(\theta, \theta_0) = \frac{1}{3}\left(\sum_{i=1}^{3} L_{hinge}(y^{(i)}(\theta^T x^{(i)} + \theta_0))\right) + \lambda\|\theta\|^2,$$

with $\lambda = .01$ and step size = 1 ?

i. θ: $[0, .98]^T$

ii. θ_0: -2

Solution: Regularization is introduced. Now, derivative of the whole objective is the derivative of the regularization term, since the hinge loss derivative is still 0. We have

$$\nabla_\theta J = 2\lambda\theta$$
$$\nabla_{\theta_0} J = 0$$

With $\lambda = 0.01$ and a step size of 1, θ becomes $[0, 0.98]^T$ and θ_0 remains -2.

(f) What is the *margin of the whole data set* with respect to the new separator? Your answer should be a single number (or fraction).

1.04

Solution: We calculate the margin for each point p by calculating the perpendicular distance to the separator

$$\frac{\theta^T p + \theta_0}{||\theta||}$$

This gives a perpendicular distance of 1.04 for p1 and p2, and 2.96 for p3. The smallest margin is 1.04, so this is the margin of the dataset.

Note that although it works out to the same answer in this case, we can't just use $\frac{1}{||\theta||}$ here because $|y(\theta^T x + \theta_0)| \neq 1$.

Now consider the situation when $\theta = [0, 1]^T, \theta_0 = -1/2$.

(g) What would be the next values of θ and θ_0 after one step of batch gradient descent where the objective is

$$J(\theta, \theta_0) = \frac{1}{3}\left(\sum_{i=1}^{3} L_{hinge}(y^{(i)}(\theta^T x^{(i)} + \theta_0))\right) + \lambda\|\theta\|^2,$$

with $\lambda = 0$ and step size = 1 ?

i. θ: $[-4/3, 1/3]^T$

ii. θ_0: $-7/6$

Solution: Now there is no regularization, but the new separator misclassifies p1 and p2. The gradient contribution for each misclassified point is $-y^{(i)}x^{(i)}$.

$$\begin{aligned}
\nabla_\theta J &= -1/3(-[1, 1] - [3, 1]) = [4/3, 2/3] \\
\nabla_{\theta_0} J &= -1/3(-1 - 1) = 2/3 \\
\text{New } \theta &= [-4/3, 1/3] \\
\text{New } \theta_0 &= -7/6
\end{aligned}$$

Name: _____

Sources of error

2. (10 points) Recall that *structural* error arises when the hypothesis class cannot represent a hypothesis that performs well on the test data and *estimation* error arises when the parameters of a hypotheses cannot be estimated well based on the training data.

   Following is a collection of potential cures for a situation in which your learning algorithm generates a hypothesis with high test error.

   For each one, indicate whether it **can reduce** structural error, estimation error, neither, or both.

   (a) Penalize $\|\theta\|^2$ during training
   ◯ structural error √ **estimation error** ◯ both ◯ neither

   (b) Penalize $\|\theta\|^2$ during testing
   ◯ structural error ◯ estimation error ◯ both √ **neither**

   (c) Increase the amount of training data
   ◯ structural error √ **estimation error** ◯ both ◯ neither

   (d) Increase the order of a fixed polynomial basis
   √ **structural error** ◯ estimation error ◯ both ◯ neither

   (e) Decrease the order of a fixed polynomial basis
   ◯ structural error √ **estimation error** ◯ both ◯ neither

   (f) Add more layers with linear activation functions to your neural network
   ◯ structural error ◯ estimation error ◯ both √ **neither**

   (g) Add more layers with non-linear activation functions to your neural network
   √ **structural error** ◯ estimation error ◯ both ◯ neither

   (h) Stop training before training error reaches 0
   ◯ structural error √ **estimation error** ◯ both ◯ neither

   Solution: Recall that both structural and estimation error are defined in terms of error on the test data. Structural error is inherent to the hypothesis class being insufficiently rich to represent the data, while estimation error occurs when training does not achieve good generalizable parameters.

   a) Penalizing $||\theta||^2$ during training can reduce estimation error because it can prevent overfitting on training data. Therefore, it can improve generalization to test data.

   b) Penalizing $||\theta||^2$ during testing can have no effect because θ is not updated during testing. The additional penalty from the regularization term of the objective would be the same for every test point.

   c) Increasing the amount of training data can reduce estimation error because more updates to the parameters of the hypothesis class during training can result in better predictions in testing, in the case of no overfitting.

   d) Increasing the order of a fixed polynomial basis is an example of increasing the complexity of the hypothesis class. If we begin with an overly simple model (high structural

error) on data generated from a complicated distribution, making the model more complex could achieve better performance on test data.

e) Decreasing the order of a fixed polynomial basis is an example of decreasing the complexity of the hypothesis class. We can potentially combat overfitting by reducing model complexity, so decreasing polynomial order could reduce estimation error.

f) Adding layers with linear activations will not affect model complexity.

g) Adding layers with non-linear activations is another example of increasing the complexity of the hypothesis class, which can result in better predictions in testing.

h) Stopping training early, before training error reaches zero, is one way to prevent overfitting. Therefore it can reduce estimation error.

For each of the following situations, indicate whether the **poor performance is due to** high structural error, high estimation error, neither, or both.

(i) Neural network has very low training error but high testing error.
◯ structural error √ **estimation error** ◯ both ◯ neither

(j) Neural network training error is persistently high, as is test error.
√ **structural error** ◯ estimation error ◯ both ◯ neither

Solution: i) A neural network with low training error but high testing error has overfit, so it has high estimation error. We could potentially combat this by regularizing, choosing a simpler hypothesis class, or by stopping training early.

j) If neural network training error and test error are both persistently high, we are probably using the wrong type of model. Therefore, structural error is high and we should consider other hypothesis classes.

Name: _____

Formulation

3. (18 points) We want to design a neural network to solve each of these problems. For each one, indicate a good choice of:

   - representation of **target output values** y
   - activation function on the output layer
   - loss function

   Note: Write a mathematical expression for the loss function, not just the type of loss in words. You can assume "*guess*" and "*actual*" are scalars or vectors depending on context; use subscripts on these variables to index the output if it is a vector.

   (a) Predict when a train will arrive based on the day, time, and weather, in minutes relative to the scheduled arrival time. If your prediction is *after* the train actually arrives, it has loss 100. If before, then the loss is the number of minutes early you predict.

   i. Representation of target output value:
   ◯ integer ✓ **real number** ◯ one-hot vector ◯ vector of values in $\{0, 1\}$

   Solution: We need continuous values to represent time, so real numbers are the right choice.

   ii. Output activation function: **linear**

   Solution: The loss is either a constant or the number of minutes early, implying a linear activation.

   iii. Loss function (provide full equation):
   $Loss(guess, actual) =$

   Solution:
   $$\begin{cases} 100 & \text{if } guess > actual \\ actual - guess & \text{otherwise} \end{cases}$$
   The linear activation is piecewise based on the constraints. If $guess > actual$, the prediction is late and the loss is 100. If $guess \leq actual$, the prediction is early and the loss is the number of minutes early.

   (b) Predict which items—out of 10,000 possible items sold by Things 'R' Us—a shopper will purchase during one shopping trip, based on their previous shopping history. You care only about whether or not an item is bought (*not* the quantity purchased), and any given customer can order multiple items.

   i. Representation of target output value:
   ◯ integer ◯ real number ◯ one-hot vector ✓ **vector of values in** $\{0, 1\}$

   Solution: At a first glance, the only reasonable choices are one-hot and a vector of $\{0, 1\}$ values because there is no linear relationship that would be implied by integers or real numbers. Since customers can purchase multiple items, we can eliminate one-hot as an option. We therefore require a vector of $\{0, 1\}$ values.

ii. Output activation function: **sigmoid**

Solution: We use sigmoid because [0, 1] is the correct range of output values for each of the 10000 item categories. Furthermore, we potentially want predictions for a given person's different purchases to be independent of each other, which sigmoid allows.

iii. Loss function (provide full equation):
$Loss(guess, actual) =$

Solution:

$$-\sum_{i=1}^{10000} (actual_i \log guess_i + (1 - actual_i) \log(1 - guess_i))$$

The target output is a vector of {0, 1} values and the training data is comprised of vectors of {0, 1} values. We can use the NLL loss for binary variables over 10000 indices, which we get by taking the log of

$$\prod_{i=1}^{10000} {guess_i}^{actual_i} (1 - guess_i)^{1-actual_i}$$

Monotonicity allows us to use the log form instead. We use this because derivatives of a summation are much easier to handle than derivatives of a product.

(c) Predict the single nationality of a person (out of 100 possible values) based on their walking speed and clothing.

i. Representation of target output value:
◯ integer ◯ real number √ **one-hot vector** ◯ vector of values in $\{0, 1\}$

Solution: A one-hot vector makes sense assuming each person has exactly one nationality. Nationalities do not have any linear relationship with each other.

ii. Output activation function: **softmax**

Solution: We can use softmax to predict the single nationality of a person. This may feel counterintuitive because softmax produces a vector of probabilities. However, remember that we train on the vector representing the true nationality of a person, which contains only a single 1.

Sigmoid is not a great choice because it can have independent outputs. We want to predict a single nationality, so we want dependent outputs where the probabilities of the other classes decrease as the probability of one class increases.

iii. Loss function (provide full equation):
$Loss(guess, actual) =$

Solution:

$$-\sum_{i=1}^{100} actual_i \log guess_i$$

Name: _____

The target output is a one-hot vector and the training data is comprised of one-hot vectors. We can use the general form of NLL multiclass (NLLM) loss for 100 classes, which we get by taking the log of

$$\prod_{i=1}^{100} guess_i{}^{actual_i}$$

Again, we want the log form for ease of taking derivatives in backpropagation.

Name: _____

Radial basis features

4. (28 points) We will consider a systematic way of creating a new feature space called *radial basis functions*. To define the new features, we need a set of example points $E = (E_1, \ldots, E_k)$ where each example $E_i \in \mathbb{R}^d$ where d is the dimension of the original input space. Our feature transformation is:

$$\phi(x) = \begin{bmatrix} \exp(-\beta \|E_1 - x\|^2) \\ \exp(-\beta \|E_2 - x\|^2) \\ \cdots \\ \exp(-\beta \|E_k - x\|^2) \end{bmatrix}$$

for some $\beta > 0$.

(a) What is the dimension of $\phi(x)$? __k **or** $k \times 1$__

Solution: Dimension of the column vector is k or k by 1, from the transformed feature vector given.

(b) Consider the following concrete example:
$d = 1$, $E = [[1], [2]]$, $\beta = 1$
Original data set $X = [[0], [1], [2]]$, $Y = [[+1], [-1], [+1]]$

i. Is the data set X, Y linearly separable in the original space?
◯ Yes √ **No**

If yes, provide parameters that describe a separator, otherwise write 'None.'

ii. θ: __**None**__

iii. θ_0: __**None**__

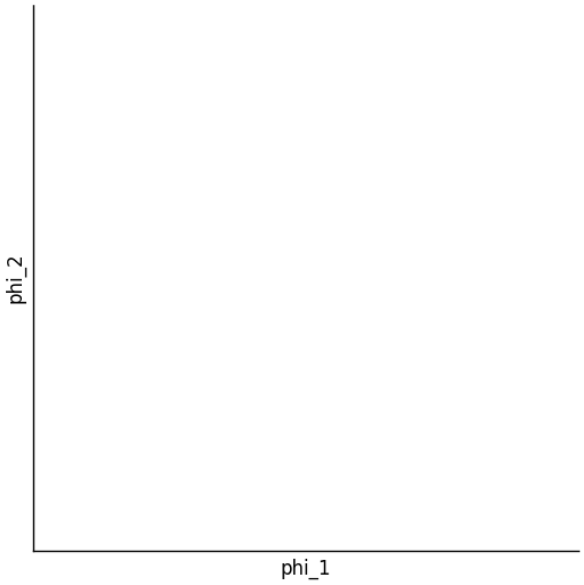
Solution: i) In the original space, these three examples are collinear with the negative example in between the positive examples, so they're not linearly separable.

ii) None by i)

iii) None by i)

(c) On the axes below, plot the points $\phi([0]), \phi([1]), \phi([2])$. Label them clearly. This table may be useful:

v	$\exp(-v)$
0	1.0
1	0.37
2	0.13
4	0.02
8	0.0003



Solution: The calculations are

$$\phi([0]) = [exp(-1), exp(-4)] = [0.37, 0.02]$$

$$\phi([1]) = [exp(0), exp(-1)] = [1.0, 0.37]$$

$$\phi([2]) = [exp(-1), exp(0)] = [0.37, 1.0]$$

(d) Is the data set $\phi(X), Y$ linearly separable?

√ **Yes** ◯ No

Solution: From the plot, the transformed features are linearly separable with a vertical separator.

(e) If so, provide parameters that describe a separator.

i. θ: $[-1, 0]$

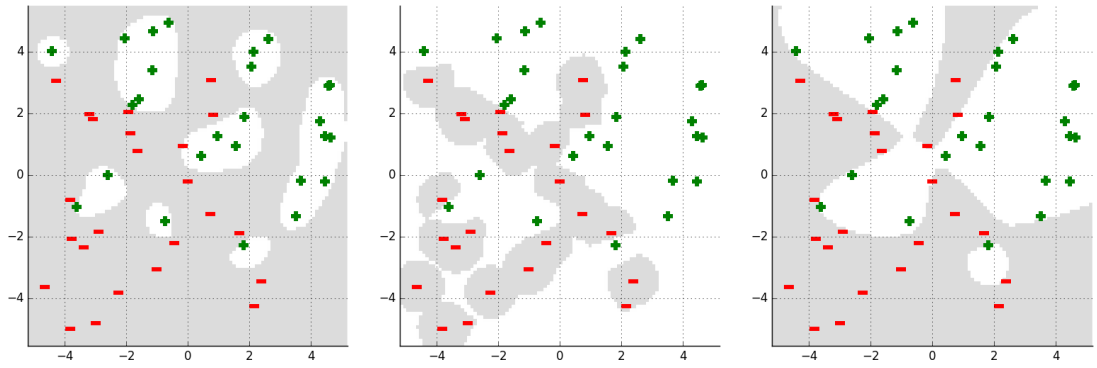
ii. θ_0: 0.5

Solution: From the plot, a good separator is $\theta = [-1, 0]$ and $\theta_0 = 0.5$. The separator should be vertical and θ_0 should fall between 0.37 and 1.0.

One common strategy is to use the x values in the training set as E. We will do that in the following questions.

(f) Now, in a 2D domain, the following plots show separators obtained for values of $\beta \in \{0.1, 1.0, 1000\}$ using the perceptron algorithm. The shaded area corresponds to regions of space that will be classified as negative.

Match the plot to the value.



Beta: **1** Beta: **1000** Beta: **0.1**

Solution: β determines the range or influence each point has on the boundary, since it acts as a multiplicative factor on the squared distance term inside the exponential. A small value of β results in each point having wider influence, which would produce smoother contours as in the third graph. The middle graph has contours that are localized closer to each individual point, suggesting a large value of β. The first graph is somewhere between the other two.

(g) In the data set above, the perceptron algorithm made 344, 42, and 162930 mistakes on three of the runs, but we forgot which β values they corresponded to. Match the number of mistakes to the $\beta \in \{0.1, 1.0, 1000\}$.

i. Mistakes: 42 Beta: **1000**

ii. Mistakes: 344 Beta: **1.0**

iii. Mistakes: 162930 Beta: **0.1**

Solution: As β goes to infinity, we approach a one-hot representation with contours localized around each point. This can be solved with one pass through the data with perceptron. Therefore, higher β values – more localization – correspond to fewer mistakes.

(h) In the limit as β approaches ∞, what familiar form does this feature representation take?

one hot encoding!

Solution: Since higher β gives more localization around each point, β approaches a one-hot representation as it increases.

Now consider the case where we fix the number, k, of example points, but allow the coordinates of the points to be adjusted by the learning algorithm. We can think of this as a kind of neural network, parameterized by $E_1, \ldots, E_k$ as well as a k-dimensional weight vector W and offset W_0, so that the output of the network is

$$\hat{y} = W^T \phi(x) + W_0$$

(i) If our loss function on a single data point is $Loss(\hat{y}, y) = (\hat{y} - y)^2$, what is $\nabla_W Loss(\hat{y}, y)$?

Solution:

$$\begin{aligned}
\nabla_W Loss(\hat{y}, y) &= \nabla_W (\hat{y} - y)^2 \\
&= 2(\hat{y} - y) \nabla_W \hat{y} \\
&= 2(\hat{y} - y)\phi(x)
\end{aligned}$$

(j) What is $\nabla_{E_1} Loss(\hat{y}, y)$? (Remember that $\frac{d}{dt}\exp(t) = \exp(t)$.)

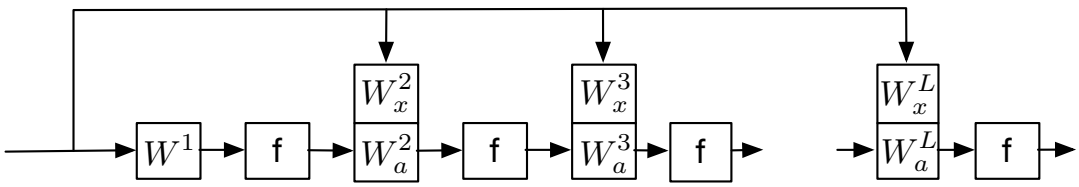
Solution:

$$\begin{aligned}
\nabla_{E_1} Loss(\hat{y}, y) &= \nabla_{E_1} (\hat{y} - y)^2 \\
&= 2(\hat{y} - y) \nabla_{E_1} \hat{y} \\
&= 2(\hat{y} - y) \nabla_{E_1} W^T \phi(x) \\
&= 2(\hat{y} - y) W_1 \nabla_{E_1} \phi_1(x) \\
&= 2(\hat{y} - y) W_1 \nabla_{E_1} \exp(-\beta \|E_1 - x\|^2) \\
&= -2\beta(\hat{y} - y) W_1 \exp(-\beta \|E_1 - x\|^2) \nabla_{E_1} \|E_1 - x\|^2 \\
&= -4\beta(\hat{y} - y) W_1 \exp(-\beta \|E_1 - x\|^2)(E_1 - x)
\end{aligned}$$

Name: _____

Shortcut connections

5. (14 points) In some neural-network models, it has proved useful to pass the input value, unchanged, into each of the subsequent layers, as shown in the figure below.



The forward pass is governed by the following equations:

$$
\begin{aligned}
a^0 &= x \\
a^1 &= f({W^1}^T a^0 + W_0^1) \\
z^l &= {W_a^l}^T a^{l-1} + {W_x^l}^T x + W_0^l \\
a^l &= f(z^l)
\end{aligned}
$$

Note that the second line above defines how to compute the output of the *first* layer, and the third and fourth lines define how to compute the output of all subsequent layers for $l = 2 \ldots L$.

(a) Let $m^l = n^{l-1}$ be the number of inputs entering layer l and n^l be the number of outputs. So, m^0 is the dimension of the input vector.

What is the dimension of W_a^l for $l > 1$? $\underline{m^l \times n^l \textbf{ or } n^{l-1}} \times n^l$

Solution: W_a^l for $l > 1$ needs the number of rows to equal the number of inputs and the number of columns to equal the number of outputs. So there are m^l or n^{l-1} rows and n^l columns.

(b) What is the dimension of W_x^l for $l > 1$? $\underline{\quad m^0 \times n^l \quad}$

Solution: W_x^l also requires n^l columns, but the rows have to match the dimension of x for the multiplication. Therefore there are m^0 rows. Note that it is the transpose of W_x^l which multiplies x so the number of columns of ${W_x^l}^T$ (i.e. the number of rows of W_x^l) must match the number of rows of x.

(c) Now we will think of the backward pass of back-propagation for the "linear" modules in this network. Given $\partial \text{Loss}/\partial z^l$, a^{l-1}, W_a^l and x

i. Write an expression for $\partial \text{Loss}/\partial a^{l-1}$.

Solution: $W_a^l \cdot \partial \text{Loss}/\partial z^l$

ii. Write an expression for $\partial \text{Loss}/\partial W_a^l$.

Solution: $a^{l-1}(\partial \text{Loss}/\partial z^l)^T$

iii. Write an expression for $\partial \text{Loss}/\partial W_x^l$.

Solution: $x(\partial \text{Loss}/\partial z^l)^T$

Name: _____

Passive-aggressive algorithm

6. (10 points) The perceptron algorithm, through the origin, iterates through its data set, considering each point $(x^{(i)}, y^{(i)})$, where $x^{(i)} \in R^d$ and $y^{(i)} \in \{+1, -1\}$, and making changes to its parameters θ based on that point. If the point is classified correctly, it makes no change. If the point is not classified correctly, the algorithm performs the update:

$$\theta = \theta + y^{(i)} x^{(i)}$$

After this update, the data point $x^{(i)}, y^{(i)}$ may still not be classified correctly.

(a) In two dimensions (when $d = 2$), provide values for θ, $x^{(i)}$ and $y^{(i)}$ for which this is the case (there are *many* possible answers—any one will do).

i. θ : **0, 100**

ii. $x^{(i)}$: **(0, -1)**

iii. $y^{(i)}$: **+1**

(b) The *passive-aggressive* algorithm is a variation of the perceptron algorithm which performs an update for any point satisfying $y^{(i)}\theta^T x^{(i)} < 1$. The update has the form

$$\theta = \theta + cy^{(i)}x^{(i)}$$

where c may be a function of $x^{(i)}$, $y^{(i)}$, and θ. We are interested in finding a minimal value of $c > 0$ for which the data point will satisfy

$$y^{(i)}\theta_{new}^T x^{(i)} \geq 1 \ .$$

after the update. It turns out that the solution has the form:

$$c = \alpha(1 - y^{(i)}\theta^T x^{(i)})$$

Give an expression for α that represents the smallest magnitude update that will cause $y^{(i)}\theta_{new}^T x^{(i)} \geq 1$ to be true. You may use θ, $x^{(i)}$, and/or $y^{(i)}$ in your expression.

Solution:

$$\alpha = \frac{1}{\|x^{(i)}\|^2}$$

.

We start with $y\theta_{new}^\top x \geq 1$. We can then substitute and expand θ_{new} by considering the update rule

$$\begin{aligned} 1 &\leq y(\theta + cyx)^\top x \\ &\leq y\theta^\top x + cy^2 x^\top x \end{aligned}$$

Moving terms around, we get

$$\begin{aligned} 1 - y\theta^\top x &\leq cy^2 x^\top x \\ \frac{1}{x^\top x}(1 - y\theta^\top x) &\leq cy^2 \\ \frac{1}{||x||_2^2}(1 - y\theta^\top x) &\leq c \end{aligned}$$

where the last step holds from $y^2 = 1$. Since $y\theta^\top x < 1$ (from the assumption in the problem), the left hand side is non negative and we can conclude that the smallest c is > 0 and is at equality of the bound we found. Therefore,

$$\frac{1}{||x||_2^2}(1 - y\theta^\top x) = c \implies \alpha = \frac{1}{||x||_2^2}.$$

(Another explanation.)
You are essentially just solving $y^{(i)}\theta_{new}^T x^{(i)} = 1$ for c and matching that solution to the specified form. We know that the minimal $c > 0$ has to match the lower bound of

our inequality because as the amount of the "correction" increases, $y^{(i)}\theta_{new}^T x^{(i)}$ must be greater, i.e. "more correct". Substituting in the update, we have

$$\begin{aligned} y^{(i)}(\theta + cy^{(i)}x^{(i)})^T x^{(i)} &= 1 \\ y^{(i)}\theta^T x + c(y^{(i)})^2 (x^{(i)})^T x^{(i)} &= 1 \\ y^{(i)}\theta^T x^{(i)} + c\|x^{(i)}\|^2 &= 1 \\ c &= \frac{1}{\|x^{(i)}\|^2}(1 - y^{(i)}\theta^T x^{(i)}) \end{aligned}$$

Therefore, $\alpha = \frac{1}{\|x^{(i)}\|^2}$.

Name: _____

Work space