COMS 4771 Machine Learning (Spring 2018)
Problem Set #1

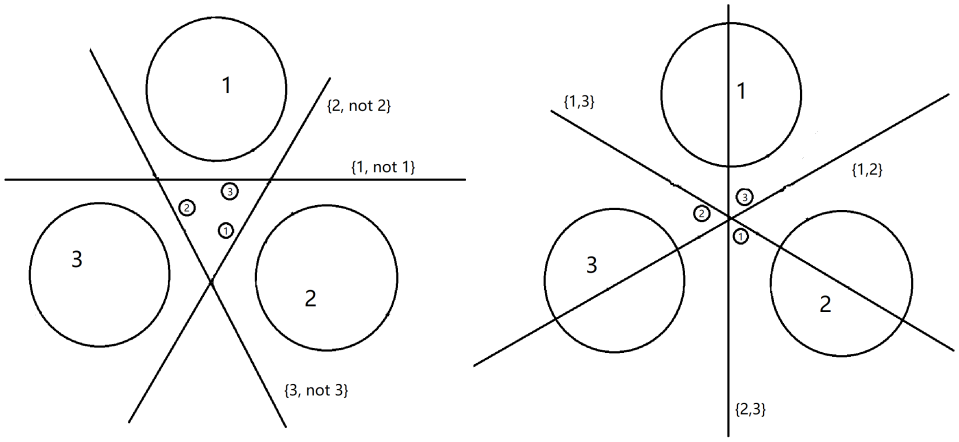
Jianfu Yang - jy2863 Wenda Xu - wx2195 Fan Yang - fy2232
- wx2195@columbia.edu

March 3, 2018

Problem 1

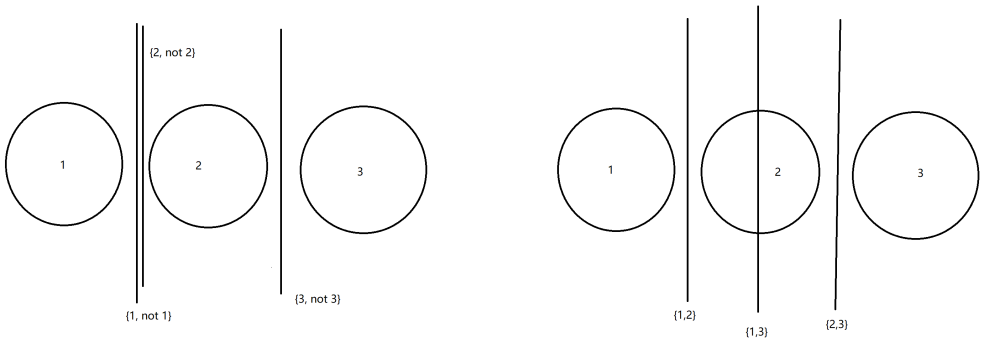
(i) OvR: when a new data comes, applying all f_i to test it. If the set of results only contains one 1(assuming it's for f_i) and the others are all 0, then the prediction would be class i. Otherwise, if there are several 1s returned, among these classes, choose the class with the smallest label (in this case, it may make mistake). Finallly, if no 1 is returned, choose class 1 as the output. In this algorithm, we need call classifier k times totally, while each f_i will be called once.
OvO: firstly gives each class a count of 0. Then when a new data comes, applying all f_{ij} to test it. For each f_{ij}, if the classifier classify the test data as class i, then the count of class i increases by one; symmetrically for j. After calling all f_{ij}, choose the class with the largest count as the prediction. If there are several classes all have the same largest count, then choose the class with the smallest label. In this algotirhm, we need call classifier $\frac{k(k-1)}{2}$ times totally, while each f_{ij} will be called once.

(ii) (1) We give the dataset as shown in the picture:



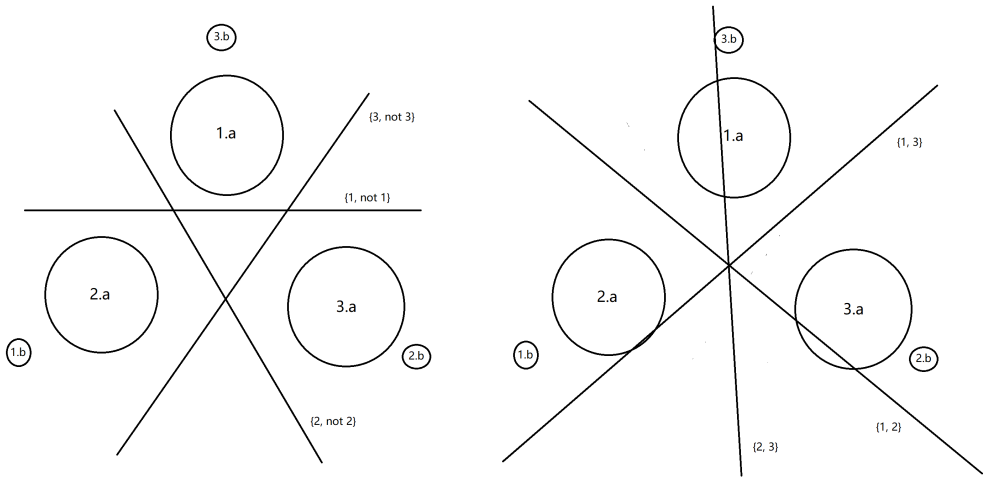
First one is OvR while the second one is OvO. Each large circle contains several points of one class; while each small circle represents only one point. The linear classifier is obtained by a soft-margin algorithm.
OvR: large circle 1 will be labeled (1, *not* 2, *not* 3), hence it's correct; similar with large circle 2 and 3. Small circles will all be labeled (*not* 1, *not* 2, *not* 3), hence by the OvR algorithm in part(i), they will get label as 1. Thus one of these three points is correct.
OvO: large cirle 1 will get two votes for 1 from $\{1, 2\}$ and $\{1, 3\}$, hence it's correct; similar with large circle 2 and 3. Small circle 1 will get two votes for 2 from $\{1, 2\}$ and $\{2, 3\}$, hence it's incorrect; similar for small circle 2 and 3. Hence, all of these three points are labeled incorrectly.
Thus, OvR has higher accuracy than OvO in this case.

(2) We give the dataset as shown in the picture:



First one is OvR while second one is OvO. Each circle contains several points of one class.
OvR: circle 1 will be labeled as (1, *not* 2, *not* 3), hence it's labeled correctly. Circle 2 will labeled as (*not* $1, 2$, *not* 3), hence it's labeled correctly. Circle 3 will get (*not* $1, 2, 3$), thus it will be labeled 2, which is incorrect.
OvO: circle 1 will get two votes for 1 from $\{1, 2\}$ and $\{1, 3\}$; circle 2 will get two votes for 2 from $\{1, 2\}$ and $\{2, 3\}$; circle 3 will get two votes for 3 from $\{1, 3\}$ and $\{2, 3\}$. Hence all points are labeled correctly.
Thus, OvO has higher accuracy than OvR in this case.

(3) We give the dataset as shown in the picture:



First one is OvR while second one is OvO.Each circle denotes several points. $1.a$ and $1.b$ belongs to class 1; $2.a$ and $2.b$ belongs to class 2; $3.a$ and $3.b$ belongs to class 3. The lines are the classifiers. Assume the numbers of points in $1.a$, $2.a$ and $3.a$ are ϵn respectively, while $1.b$, $2.b$ and $3.b$ has $(1-\epsilon)n$ points respectively. OvR: $1.a$ will be labeled as $(1,\ not\ 2,\ not\ 3)$, hence $1.a$ is labeled correctly; similar for $2.a$ and $3.a$. $1.b$ will be labeled as $(not\ 1, 2,\ not\ 3)$, hence $1.b$ is labeled incorrectly; similar for $2.b$ and $3.b$. The accuracy would be at most:

$$accuracy = \frac{3\epsilon n}{3((1-\epsilon)n + \epsilon n)} = \epsilon$$

OvO: $1.a$ could get two votes for class 1 from $\{1,3\}$ and $\{1,2\}$, hence $1.a$ is labeled correctly; similar for $2.a$ and $3.a$. $1.b$ will get two votes for class 2 from $\{1,2\}$ and $\{2,3\}$, hence $1.b$ is labeled incorrectly; similar for $2.b$ and $3.b$. The accuracy would be:

$$accuracy = \frac{3\epsilon n}{3((1-\epsilon)n + \epsilon n)} = \epsilon$$

Hence, both algorithm will get accuracy of at most ϵ.

(4) The dataset is similar with above. This time small circles contain ϵn points respectively, while large circles contain $(1-\epsilon)n$ points respectively. Hence the accuracy will be complementary of last case. It will be at least $1-\epsilon$.

(iii) The algorithm would be divide&conquer.
We first divide the label space into two parts: $[1, ..., \lfloor \frac{k}{2} \rfloor]$ and $[\lfloor \frac{k}{2} \rfloor + 1, ..., k]$. One can implement a classifier as $f : \mathbb{R}^d \to \{first\ part, second\ part\}$, i.e., assigning the examples from the first part label 1, the second part label 0.
Then provide this division and method of establishing classifier to both of the two parts recursively until the part only contains one class.
We could treat this as a decision tree. Each node (contains the root) is a classifier, 1

is the left subtree while 0 is the right. Each class is a leaf of this tree. Since we always gurantee that two subtrees rooted at the same node will have the nunmber of elements with difference no more than 1, the tree is well balanced. Hence, the height of the tree should be $\lceil \log_2 k \rceil + 1$, i.e., when we go from the root to a leaf, the maximum number of classifier calling should be $c = \lceil \log_2 k \rceil$.
If one implementation only need to call classifiers $\lceil \log_2 k \rceil - 1$ times, then this decision tree will have at most $2^{\lceil \log_2 k \rceil - 1}$ leaves. This will archieve the maximum of $k - 1$ while $k = 2^n + 1$. Hence, there will be at least two classes haven't been compared (i.e., there isn't classifier which could classify them). Then the algorithm will classify test data of these two classes randomly, causing a lower accuracy.

Problem 2

$$f(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{x}_a||^2$$
$$\frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2} = \frac{\partial\ 2(\boldsymbol{x} - \boldsymbol{x}_a)}{\partial \boldsymbol{x}} = 2\boldsymbol{I}_{d\times d}$$

d is the dimension of $\boldsymbol{x}$. For any $\boldsymbol{z}$ in $\mathbb{R}^d$:

$$\boldsymbol{z}^T \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2} \boldsymbol{z} = 2\boldsymbol{z}^T I_{d\times d}\boldsymbol{z} = 2\boldsymbol{z}^T\boldsymbol{z} = 2||\boldsymbol{z}||^2 \geqslant 0$$

Thus $f(\boldsymbol{x})$ is convex. Besides, the constraint $g(\boldsymbol{x}) = 0$ is a hyperplane, hence the line of any two points in this set would also locates in this set, i.e., the constraint is an affine set. Thus, the problem is a convex optimal problem. Establish the Lagrangian function:

$$\mathcal{L}(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}) \quad (\lambda \geqslant 0)$$

To minimize $\mathcal{L}(\boldsymbol{x}, \lambda)$, the derivatives should be 0:

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \lambda)}{\partial \boldsymbol{x}} = 2(\boldsymbol{x} - \boldsymbol{x}_a) + \lambda\boldsymbol{\omega} = 0$$
$$\boldsymbol{x} = \boldsymbol{x}_a - \frac{\lambda}{2}\boldsymbol{\omega}$$
$$\boldsymbol{\omega}\cdot\boldsymbol{x} + \omega_0 = \boldsymbol{\omega}\cdot\boldsymbol{x}_a + \omega_0 - \frac{\lambda}{2}||\boldsymbol{\omega}||^2$$
$$g(\boldsymbol{x}) = 0 = g(\boldsymbol{x}_a) - \frac{\lambda}{2}||\boldsymbol{\omega}||^2$$
$$\frac{|g(\boldsymbol{x}_a)|}{||\boldsymbol{\omega}||} = \frac{\lambda}{2}||\boldsymbol{\omega}|| = ||\boldsymbol{x} - \boldsymbol{x}_a|| = distance$$

The distance is minimized with $f(\boldsymbol{x})$. Thus, the minium distance from $\boldsymbol{x}_a$ to the hyperplane is $\frac{|g(\boldsymbol{x}_a)|}{||\boldsymbol{\omega}||}$.

Problem 3

(i) For example, think about a dataset consists of $[\boldsymbol{x}_1 = (1, 1), y_1 = 1]$ and $[\boldsymbol{x}_2 = (1, -1 - \epsilon), y_2 = -1]$ for some small $\epsilon > 0$. Each time, the algorithm will first test $\boldsymbol{x}_1$. Now, after one iteration, $\boldsymbol{\omega}_1 = (1, 1)$.

$$sign(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_1) = sign(2) = 1 = y_1$$

$$sign(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_1) = sign(-\epsilon) = -1 = y_2$$

Hence, the algorithm terminates. The γ' now is:

$$\gamma' = \min_{\boldsymbol{x}} \frac{|\boldsymbol{\omega}_1 \cdot \boldsymbol{x}|}{||\boldsymbol{\omega}_1||} = \min(\sqrt{2}, \frac{\sqrt{2}}{2}\epsilon) = \frac{\sqrt{2}}{2}\epsilon$$

Consider $\boldsymbol{\omega} = (0, 1)$, we can find the nearly largest γ is 1 (since ϵ is small enough), which could be significantly larger than γ'.

(ii) (a) Since the algorithm terminates, $2y_i(\boldsymbol{\omega}_T \cdot \boldsymbol{x}_i) > \gamma||\boldsymbol{\omega}_T||$ must apply for all i. Recognize that y_i must have the same sign with $\boldsymbol{\omega}_T \cdot \boldsymbol{x}_i$ now for i. Hence:

$$\min_{\boldsymbol{x}}(2|\boldsymbol{\omega}_T \cdot \boldsymbol{x}_i|) > \gamma||\boldsymbol{\omega}_T||$$

Distance from $\boldsymbol{x}_i$ to the hyperplane $\boldsymbol{\omega}_T \cdot \boldsymbol{x} = 0$ is $\frac{|\boldsymbol{\omega}_T \cdot \boldsymbol{x}_i|}{||\boldsymbol{\omega}_T||}$ as derived in Problem2.

$$\gamma' = \min_{\boldsymbol{x}} \frac{|\boldsymbol{\omega}_T \cdot \boldsymbol{x}_i|}{||\boldsymbol{\omega}_T||} > \frac{\gamma}{2}$$

Here γ' means the margin we get. Hence, we prove that the lower bound of γ' is $\frac{\gamma}{2}$.

(b) (i) Assuming that $\boldsymbol{\omega}^*$ is the optimal hyperplane with margin γ.

$$\begin{aligned}
\boldsymbol{\omega}_t \cdot \boldsymbol{\omega}^* &= (\boldsymbol{\omega}_{t-1} + y\boldsymbol{x}) \cdot \boldsymbol{\omega}^* \\
&= \boldsymbol{\omega}_{t-1} \cdot \boldsymbol{\omega}^* + y\boldsymbol{x} \cdot \boldsymbol{\omega}^* \\
&\geqslant \boldsymbol{\omega}_{t-1} \cdot \boldsymbol{\omega}^* + \gamma||\boldsymbol{\omega}^*|| \\
||\boldsymbol{\omega}_T||\ ||\boldsymbol{\omega}^*|| &\geqslant \boldsymbol{\omega}_T \cdot \boldsymbol{\omega}^* \\
&\geqslant \boldsymbol{\omega}_0 \cdot \boldsymbol{\omega}^* + T\gamma||\boldsymbol{\omega}^*|| \\
&= T\gamma||\boldsymbol{\omega}^*|| \\
||\boldsymbol{\omega}_T|| &\geqslant T\gamma
\end{aligned}$$

Proven.

(ii)

$$\begin{aligned}
||\boldsymbol{\omega}_t||^2 &= ||\boldsymbol{\omega}_{t-1} + y\boldsymbol{x}||^2 \\
&= ||\boldsymbol{\omega}_{t-1}||^2 + 2y\boldsymbol{\omega}_{t-1} \cdot \boldsymbol{x} + ||y\boldsymbol{x}||^2 \\
&\leqslant ||\boldsymbol{\omega}_{t-1}||^2 + \gamma||\boldsymbol{\omega}_{t-1}|| + ||y\boldsymbol{x}||^2 \\
&\leqslant (||\boldsymbol{\omega}_{t-1}||^2 + \gamma||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma^2}{4}) + R^2 \\
&= (||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2})^2 + R^2
\end{aligned}$$

Proven.

(iii)

$$\begin{aligned}
||\boldsymbol{\omega}_t||^2 &\leqslant (||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2})^2 + R^2 \\
R^2 &\geqslant ||\boldsymbol{\omega}_t||^2 - (||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2})^2 \\
&= (||\boldsymbol{\omega}_t|| - (||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2}))(||\boldsymbol{\omega}_t|| + (||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2})) \\
\frac{R^2}{||\boldsymbol{\omega}_t|| + ||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2}} &\geqslant ||\boldsymbol{\omega}_t|| - (||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2}) \\
||\boldsymbol{\omega}_t|| &\leqslant ||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2} + \frac{R^2}{||\boldsymbol{\omega}_t|| + ||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2}}
\end{aligned}$$

Proven.

(iv)

$$\begin{aligned}
||\boldsymbol{\omega}_t|| + ||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2} &\geqslant \frac{4R^2}{\gamma} \\
\frac{R^2}{||\boldsymbol{\omega}_t|| + ||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2}} &\leqslant \frac{\gamma}{4} \\
||\boldsymbol{\omega}_t|| &\leqslant ||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2} + \frac{R^2}{||\boldsymbol{\omega}_t|| + ||\boldsymbol{\omega}_{t-1}|| + \frac{\gamma}{2}} \\
&\leqslant ||\boldsymbol{\omega}_{t-1}|| + \frac{3\gamma}{4}
\end{aligned}$$

Proven.

(v)

$$||\boldsymbol{\omega}_0|| = 0 \leqslant R$$

Since R is the largest distance among the points and γ is the margin,

$$\gamma \leqslant R$$

$$R\gamma \leqslant R^2 \leqslant 4R^2$$

$$R \leqslant \frac{4R^2}{\gamma}$$

$$||\boldsymbol{\omega}_0|| \leqslant R \leqslant \frac{4R^2}{\gamma}$$

Hence, $||\boldsymbol{\omega}||$ will grow from 0 and achieve $||\boldsymbol{\omega}_T|| \geqslant \frac{4R^2}{\gamma}$, there must be intermediate t_0 satisfies the inequalities.
Proven.

(vi) Beginning from $t = t_0$, the inequality $||\boldsymbol{\omega}_t|| > \frac{4R^2}{\gamma}$ is satisfied. Hence,

$$||\boldsymbol{\omega}_T|| \leqslant ||\boldsymbol{\omega}_{t_0-1}|| + \frac{3}{4}(T - (t_0 - 1))\gamma \leqslant ||\boldsymbol{\omega}_{t_0-1}|| + \frac{3}{4}T\gamma$$

Combine (i) and (iv):

$$T\gamma \leqslant ||\boldsymbol{\omega}_T|| \leqslant ||\boldsymbol{\omega}_{t_0-1}|| + \frac{3}{4}T\gamma \leqslant \frac{4R^2}{\gamma} + \frac{3}{4}T\gamma$$

$$\frac{1}{4}T\gamma \leqslant \frac{4R^2}{\gamma}$$

$$T \leqslant (\frac{4R}{\gamma})^2$$

Thus, the mistake bound is $(\frac{4R}{\gamma})^2$.

Problem 4

(i) Since $x_1 \dots x_n$ are distinct, we can always find a small enough σ which ensures that there is no overlap between the range $[x_i - \sigma, x_i + \sigma]$ and $[x_j - \sigma, x_j + \sigma]$ for all i and j ($1 \leqslant i, j \leqslant n$ and $i \neq j$). For each x_i, $\Phi_\sigma(x_i)$ will be 0 while α satisfies $|\alpha - x_i| \geqslant \sigma$. Now let $\omega(\alpha) = \sum_{i=1}^{n} y_i \mathbf{1}[|\alpha - x_i| < \sigma]$. Since there is no overlap between the ranges, while α is in the range $[x_i - \sigma, x_i + \sigma]$, $\omega(\alpha) = y_i$.
For each x_i,

$$\omega(\alpha) \cdot \Phi_\sigma(x_i) = \begin{cases} 0 & |\alpha - x_i| \geqslant \sigma \\ y_i e^{-\frac{1}{1-(\frac{|\alpha - x_i|}{\sigma})^2}} & |\alpha - x_i| < \sigma \end{cases}$$

$$\begin{aligned} sign(\int_{-\infty}^{+\infty} \omega(\alpha) \cdot \Phi_\sigma(x_i) d\alpha) &= sign(\int_{x_i - \sigma}^{x_i + \sigma} y_i e^{-\frac{1}{1-(\frac{|\alpha - x_i|}{\sigma})^2}} d\alpha) \\ &= sign(y_i \int_{-\sigma}^{+\sigma} e^{-\frac{1}{1-(\frac{|\alpha|}{\sigma})^2}} d\alpha) \quad (Note: e^{-\frac{1}{1-(\frac{|\alpha|}{\sigma})^2}} > 0) \\ &= sign(y_i) \end{aligned}$$

Hence, applying $\omega(\alpha)$ as the weight, any binary labeling of the n points could be linearly separated.

(ii) From above, while $x \neq x'$, their range won't have overlap, causing $\Phi_\sigma(x) \cdot \Phi_\sigma(x') = 0$. Otherwise,

$$\Phi_\sigma(x) \cdot \Phi_\sigma(x') = \int_{-\sigma}^{+\sigma} e^{-\frac{2}{1-(\frac{|\alpha|}{\sigma})^2}} d\alpha = C(\sigma)$$

$C(\sigma)$ is a constant w.r.t σ.
Hence, simply comparing x and x' could easily compute $\Phi_\sigma(x) \cdot \Phi_\sigma(x')$.

(iii) (a) Applying $\boldsymbol{\omega}$ and b:

$$\begin{aligned} sign(\langle \boldsymbol{\omega}, \boldsymbol{\phi}(x) \rangle + b) &= sign(\frac{1}{2}(||\boldsymbol{c}_-||^2 - 2\boldsymbol{c}_- \cdot \boldsymbol{\phi}(x) + ||\boldsymbol{\phi}(x)||^2) \\ &\qquad - \frac{1}{2}(||\boldsymbol{c}_+||^2 - 2\boldsymbol{c}_+ \cdot \boldsymbol{\phi}(x) + ||\boldsymbol{\phi}(x)||^2)) \\ &= sign(\frac{1}{2}(\boldsymbol{c}_- - \boldsymbol{\phi}(x))^2 - \frac{1}{2}(\boldsymbol{c}_+ - \boldsymbol{\phi}(x))^2) \\ &= sign((\boldsymbol{c}_- - \boldsymbol{\phi}(x))^2 - (\boldsymbol{c}_+ - \boldsymbol{\phi}(x))^2) \\ &= sign(||\boldsymbol{c}_- - \boldsymbol{\phi}(x)|| - ||\boldsymbol{c}_+ - \boldsymbol{\phi}(x)||) \\ &= h(x) \end{aligned}$$

Proven.

(b)

$$
\begin{aligned}
h(x) &= sign((\boldsymbol{c}_+ - \boldsymbol{c}_-) \cdot \boldsymbol{\phi}(x) + \frac{1}{2}(||\boldsymbol{c}_-||^2 - ||\boldsymbol{c}_+||^2)) \\
&= sign(\frac{1}{m_+} \sum_{i:y_i=1} \boldsymbol{\phi}(x_i) \cdot \boldsymbol{\phi}(x) - \frac{1}{m_-} \sum_{i:y_i=-1} \boldsymbol{\phi}(x_i) \cdot \boldsymbol{\phi}(x) \\
&\qquad + \frac{1}{2}(\frac{1}{m_-^2} \sum_{i:y_i=-1} \sum_{j:y_j=-1} \boldsymbol{\phi}(x_i) \cdot \boldsymbol{\phi}(x_j) - \frac{1}{m_+^2} \sum_{i:y_i=1} \sum_{j:y_j=1} \boldsymbol{\phi}(x_i) \cdot \boldsymbol{\phi}(x_j))) \\
&= sign(\frac{1}{m_+} \sum_{i:y_i=1} K(x_i, x) - \frac{1}{m_-} \sum_{i:y_i=-1} K(x_i, x) \\
&\qquad + \frac{1}{2}(\frac{1}{m_-^2} \sum_{i:y_i=-1} \sum_{j:y_j=-1} K(x_i, x_j) - \frac{1}{m_+^2} \sum_{i:y_i=1} \sum_{j:y_j=1} K(x_i, x_j)))
\end{aligned}
$$

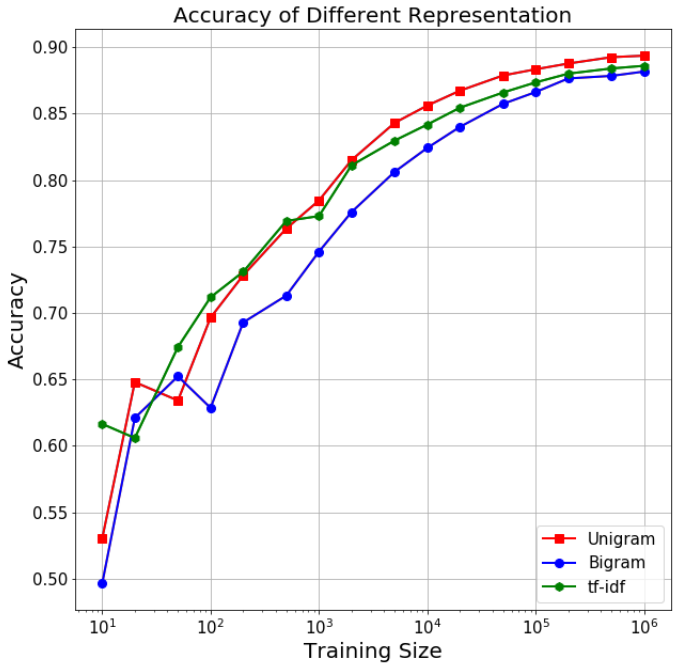
Hence, eachc term of $h(x)$ could be expressed via $K(\cdot, \cdot)$, which could be efficiently computed.

Problem 5

(i) Some words which get low weight in unigram would be considered meaningless, but in fact, it could combine with its predeccesor or successor to form a meaningful phrase in bigram. Besides, consider an example: 'disapointed' always apears with 'not' in the training set, hence after training it would be considered as a positive word. If the test data really means 'disapointed', the model will make mistake. This could be avoided via bigram.

(ii) The code is submitted via Courseworks.

(iii) The plot of the accuracy of different presentations is shown below:



We can see that when the training size is large, unigram presentation performs better than tf-idf and bigram. Decreasing the training size would cause that the accuracy also decreases.

(iv) Using the whole 1 million data to train the model by unigram, the 10 most positive and negative words are:

Positive:	Negative:
perfect, phenomenal, exceeded, heavenly, heaven, incredible, disappoint, skeptical, gem, perfection	worst, poisoning, underwhelmed, lacked, mediocre, flavorless, inedible, tasteless, underwhelming, meh

Problem 6

(i)

$$\frac{\partial \sigma}{\partial x} = -\frac{1}{(1+e^{-x})^2} \times (-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2}$$
$$\sigma(x)(1-\sigma(x)) = \frac{1}{1+e^{-x}}(1 - \frac{1}{1+e^{-x}}) = \frac{e^{-x}}{(1+e^{-x})^2}$$
$$\frac{\partial \sigma}{\partial x} = \sigma(x)(1-\sigma(x))$$

Proven.

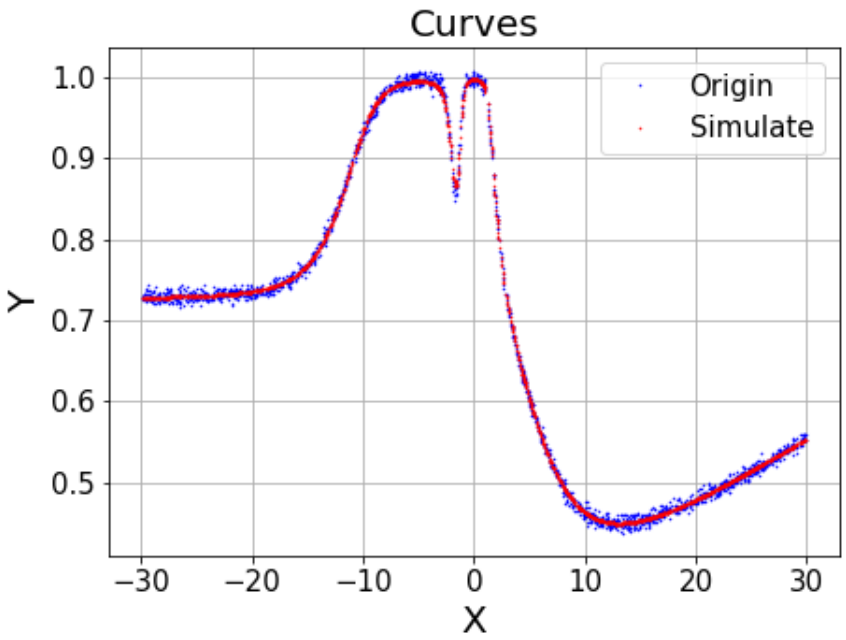
(ii)

$$E(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{k=1}^{O}(\sigma(\boldsymbol{w}_k^T \boldsymbol{x}_i + b_k) - y_{ik})^2 \quad (Let\ (\boldsymbol{w}_k^T \boldsymbol{x}_i + b_k)\ be\ I_k)$$
$$\frac{\partial E}{\partial b_k} = \frac{1}{2n}\sum_{i=1}^{n} 2(\sigma(I_k) - y_{ik})\frac{\partial \sigma(I_k)}{\partial b_k}$$
$$= \frac{1}{2n}\sum_{i=1}^{n} 2(\sigma(I_k) - y_{ik})\sigma(I_k)(1-\sigma(I_k))\frac{\partial (I_k)}{\partial b_k}$$
$$= \frac{1}{n}\sum_{i=1}^{n}(\sigma(I_k) - y_{ik})\sigma(I_k)(1-\sigma(I_k))$$
$$\frac{\partial E}{\partial \boldsymbol{b}} = \frac{1}{n}\sum_{i=1}^{n}(\sigma(\boldsymbol{I}) - \boldsymbol{y}_i) * \sigma(\boldsymbol{I}) * (\boldsymbol{1} - \sigma(\boldsymbol{I}))$$
$$\frac{\partial E}{\partial \boldsymbol{w}_k} = \frac{1}{2n}\sum_{i=1}^{n} 2(\sigma(I_k) - y_{ik})\frac{\partial \sigma(I_k)}{\partial \boldsymbol{w}_k}$$
$$= \frac{1}{2n}\sum_{i=1}^{n} 2(\sigma(I_k) - y_{ik})\sigma(I_k)(1-\sigma(I_k))\frac{\partial (I_k)}{\partial \boldsymbol{w}_k}$$
$$= \frac{1}{n}\sum_{i=1}^{n}(\sigma(I_k) - y_{ik})\sigma(I_k)(1-\sigma(I_k))\boldsymbol{x}_i$$
$$\frac{\partial E}{\partial \boldsymbol{W}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i((\sigma(\boldsymbol{I}) - \boldsymbol{y}_i) * \sigma(\boldsymbol{I}) * (\boldsymbol{1} - \sigma(\boldsymbol{I})))^T$$

The $*$ means elementwise multiply. $\boldsymbol{1}$ denotes a matrix with the same shape of $\boldsymbol{I}$ whose elements are all 1. $\boldsymbol{I}$ is the matrix consisting of I_k (i.e., $\boldsymbol{W}^T\boldsymbol{x}_i + \boldsymbol{b}$). $\sigma(\boldsymbol{I})$ is an elementwise function.

(iii) The code is submitted via Courseworks.
We set two boundaries to terminate the algorithm: 1) when the total error is small enough; 2) when all dimensions of all gradients are small enough.
The first means the fitting is good enough; the second means it's near the locally optimal point, i.e., it converges.

(iv) The figure we get is shown below:



The blue line is the original data curve; while the red line is the one we learned.