1.

i.

The parameters are $p_1, p_2, p_3, p_4, p_5, p_6, q_1, q_2, q_3, q_4, q_5, q_6$.

However, as $\sum_{i=1}^{6} p_i = 1$, we only need 11 parameters now.

Thus, parameters of this process are $p_1, p_2, p_3, p_4, p_5, q_1, q_2, q_3, q_4, q_5, q_6$.

(for the convenience of expression, all following questions in Q1 will still have p_6, and it refers to $1 - \sum_{i=1}^{5} p_i$)

ii.

$$L[\theta|(a_1, b_1) \dots (a_n, b_n)] = P[(a_1, b_1) \dots (a_n, b_n)|\theta]$$

As these n pairs are independent,

$$P[(a_1, b_1) \dots (a_n, b_n)|\theta] = \prod_{i=1}^{n} P[(a_i, b_i)|\theta] = \prod_{i=1}^{n} p_{a_i} q_{a_i} e^{-q_{a_i} b_i}$$

iii.

$$L[\theta|(a_1, b_1) \dots (a_n, b_n)] = \prod_{i=1}^{n} p_{a_i} q_{a_i} e^{-q_{a_i} b_i} \propto \sum_{i=1}^{n} \ln(p_{a_i} q_{a_i} e^{-q_{a_i} b_i})$$

Given a set of (a_i, b_i), let $\sum_{i=1}^{n} \ln(p_{a_i} q_{a_i} e^{-q_{a_i} b_i})$ be function $f(\theta)$

Now that we need to find the maximum likelihood estimate of θ, that is to compute $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} L[\theta|(a_1, b_1) \dots (a_n, b_n)] = \underset{\theta}{\operatorname{argmax}} f(\theta)$. What we need to do is to find the θ with zero gradient.

Suppose $\mathcal{K}_i$ is a subset of all n pairs of observations with $a = i$. And its size is K_i,

$$\frac{\partial f}{\partial p_i} = \sum_{j=1}^{K_i} \frac{1}{p_i} + \sum_{j=1}^{n-K_i} -\frac{1}{1-p_i} = \frac{K_i}{p_i} - \frac{n-K_i}{1-p_i}$$

Solve equation $\frac{K_i}{p_i} - \frac{n-K_i}{1-p_i} = 0$,

$$\widehat{p_i} = \frac{K_i}{n}$$

$$\frac{\partial f}{\partial q_i} = \sum_{j \in \mathcal{K}_i} \frac{1}{q_i} - b_j$$

Solve equation $\sum_{j \in \mathcal{K}_i} \frac{1}{q_i} - b_j = 0$,

$$\widehat{q_i} = \frac{n}{\sum_{j \in \mathcal{K}_i} b_j}$$

iv.

$$P[b_1 \dots b_n|\theta] = \sum_{i=1}^{6} \sum_{j=1}^{n} p_i q_i e^{-q_i b_j}$$

v.

Initialize parameters θ arbitrarily

While true **do**:

for i **in** $1 \dots n$ **do**:

$\widehat{a_i} \leftarrow \underset{j}{\operatorname{argmax}}\, p_j q_j e^{-q_j b_i}$

end

$divide\ n\ pairs\ (\ \widehat{a_i}, b_i\ )\ into\ 6\ subsets\ \mathcal{K}_1 \dots \mathcal{K}_6$

for i **in** $1 \dots 6$ **do**:

$\widehat{p_i} \leftarrow \frac{K_i}{n}$

$\widehat{q_i} \leftarrow \frac{n}{\sum_{j \in \mathcal{K}_i} b_j}$

end

if $all\ p_i = \widehat{p_i}\ and\ q_i = \widehat{q_i}$:

break

end

for i **in** $1 \dots 6$ **do**:

$p_i \leftarrow \widehat{p_i}$

$q_i \leftarrow \widehat{q_i}$

end

end

2.

As the penalty of wrong prediction is different, the classification criteria is now $\mathrm{P}[1] = c\mathrm{P}[0]$. That is, when $\mathrm{P}[1] \geq c\mathrm{P}[0]$, make prediction 1, else make prediction 0.

i.

$$\mathrm{P}[x=1] = \pi_1 N\left(x; \mu = 2, \sigma^2 = \frac{1}{4}\right) = \frac{1}{3}\frac{1}{\sqrt{2\pi\frac{1}{4}}}\exp\left(-\frac{1}{2}\frac{(x-2)^2}{\frac{1}{4}}\right) = \frac{1}{3}\frac{2}{\sqrt{2\pi}}\exp(-2(x-2)^2)$$

$$\mathrm{P}[x=0] = \pi_0 N(x; \mu = 0, \sigma^2 = 1) = \frac{2}{3}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right)$$

To solve $\mathrm{P}[x=1] \geq c\mathrm{P}[x=0]$, is to solve

$$\exp(-2(x-2)^2) \geq c\exp\left(-\frac{1}{2}x^2\right)$$

$$-2(x-2)^2 \geq \ln(c) + (-\frac{1}{2}x^2)$$

$$-\frac{3}{2}x^2 + 8x - 8 \geq \ln(c)$$

Here, note that the maximum value of $-\frac{3}{2}x^2 + 8x - 8$ is $\frac{8}{3}$, and maximum value of $\ln(c)$ is $\ln(14)$. As $\frac{8}{3} > \ln(14)$, it is guaranteed that there is some x will satisfy this inequality.

$$-\frac{3}{2}x^2 + 8x - [8 + \ln(c)] \geq 0$$

$$\frac{8 - \sqrt{16 - 6\ln(c)}}{3} \leq x \leq \frac{8 + \sqrt{16 - 6\ln(c)}}{3}$$

Thus, the subset of prediction 1 is $[\frac{8-\sqrt{16-6\ln(c)}}{3}, \frac{8+\sqrt{16-6\ln(c)}}{3}]$.

ii.

In this case, the inequality $\mathrm{P}[x = 1] \geq c\mathrm{P}[x = 0]$ is still $-\frac{3}{2}x^2 + 8x - 8 \geq \ln(c)$, however now the minimum value of $\ln(c)$ is $\ln(15)$ and $\frac{8}{3} < \ln(15)$, which indicates no x will hold this inequality. Thus the classifier will always predict 0, and the region of prediction 1 is $\emptyset$.

3.

i.

a.

Suppose the VC dimension for $\mathcal{F}_1$ is $VCdim(\mathcal{F}_1) = d_1$, and the VC dimension for $\mathcal{F}_2$ is $VCdim(\mathcal{F}_2) = d_2$.

(i). The lower bound of $VCdim(\mathcal{F}_1 \cup \mathcal{F}_2)$ is $\max(d_1, d_2)$.

This is pretty straight forward, the VC dimension for $\mathcal{F}_1 \cup \mathcal{F}_2$ cannot be lower than VC dimension of set $\mathcal{F}_1$ or $\mathcal{F}_2$, as it is the union of two sets, it contains all classifiers in these two sets.

(ii). The upper bound of $VCdim(\mathcal{F}_1 \cup \mathcal{F}_2)$ is $d_1 + d_2 + 1$.

First, introduce the growth function. Growth function is also called shatter coefficient or shattering number. It measures the richness of a set family. (*https://en.wikipedia.org/wiki/Growth_function, Vapnik, V. N.; Chervonenkis, A. Ya. (1971). "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". Theory of Probability & Its Applications.* ***16*** *(2): 264. doi:10.1137/1116025.* This is an English translation, by B. Seckler, of the Russian paper: *"On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". Dokl. Akad. Nauk.* ***181*** *(4): 781. 1968.* The translation was reproduced as: *Vapnik, V. N.; Chervonenkis, A. Ya. (2015). "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". Measures of Complexity. p. 11. doi:10.1007/978-3-319-21852-6_3. ISBN 978-3-319-21851-9.*)

And for every two classifiers sets, $Growth(\mathcal{F}_1 \cup \mathcal{F}_2, m) \leq Growth(\mathcal{F}_1, m) + Growth(\mathcal{F}_2, m)$. (https://en.wikipedia.org/wiki/Growth_function, *Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning. USA, Massachusetts: MIT Press. ISBN 9780262018258.*, especially Section 3.2)

According to Sauer's Lemma, if $Growth(\mathcal{F}, m) = d$, then for all m, $Growth(\mathcal{F}, m) \leq \sum_{i=0}^{d} \binom{m}{i}$. (https://en.wikipedia.org/wiki/Growth_function, *Shalev-Shwartz, Shai; Ben-David, Shai (2014). Understanding Machine Learning - from Theory to Algorithms. Cambridge university press. ISBN 9781107057135.*)

Thus,

$$Growth(\mathcal{F}_1 \cup \mathcal{F}_2, m) \leq Growth(\mathcal{F}_1, m) + Growth(\mathcal{F}_2, m)$$

$$Growth(\mathcal{F}_1 \cup \mathcal{F}_2, m) \leq \sum_{i=0}^{d_1} \binom{m}{i} + \sum_{i=0}^{d_2} \binom{m}{i}$$

As $\binom{m}{i} = \binom{m}{m-i}$, we can rewrite it as

$$Growth(\mathcal{F}_1 \cup \mathcal{F}_2, m) \leq \sum_{i=0}^{d_1} \binom{m}{i} + \sum_{i=0}^{d_2} \binom{m}{m-i}$$

$$Growth(\mathcal{F}_1 \cup \mathcal{F}_2, m) \leq \sum_{i=0}^{d_1} \binom{m}{i} + \sum_{i=m-d_2}^{m} \binom{m}{i}$$

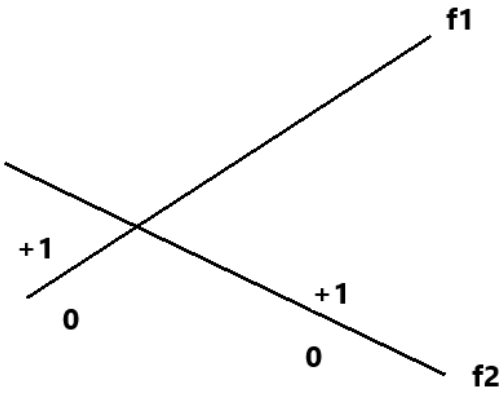
When $m > d_1 + d_2 + 1$

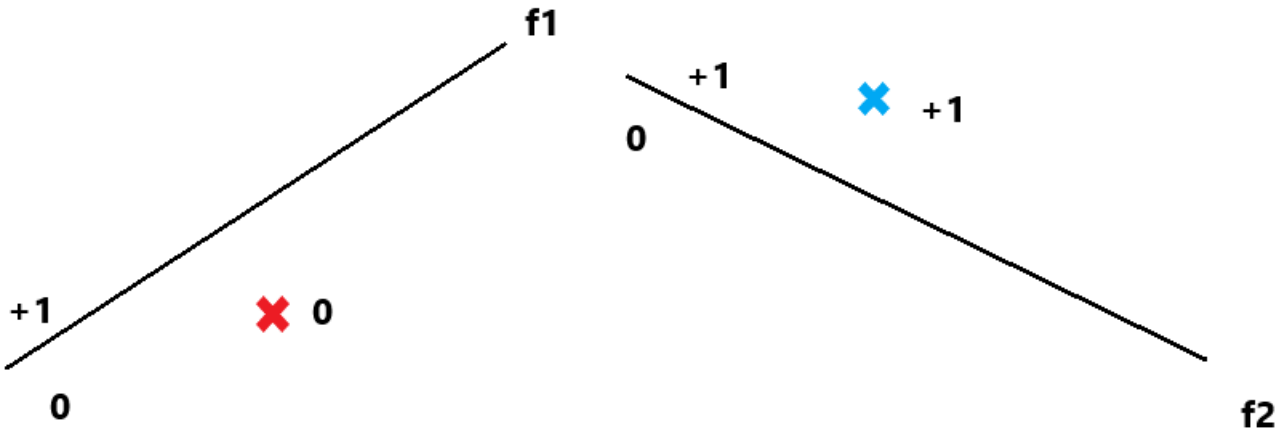
$$Growth(\mathcal{F}_1 \cup \mathcal{F}_2, m) \leq \sum_{i=0}^{m} \binom{m}{i} - \binom{m}{d+1} = 2^m - \binom{m}{d+1} < 2^m$$

And because of $VCdim(\mathcal{F}) \geq d$ if and only if $Growth(\mathcal{F}) = 2^d$, we have

$$VCdim(\mathcal{F}_1 \cup \mathcal{F}_2) \leq m = d_1 + d_2 + 1$$

Now, give an example that VC dimension of union set can actually reach the upper bound, suppose $\mathcal{F}_1$ is a set with only one liner classifier f_1, it's VC dimension $d_1 = 0$, and $\mathcal{F}_2$ is a set with only anther liner classifier f_2, it's VC dimension $d_2 = 0$, the VC dimension for union $\mathcal{F}_1 \cup \mathcal{F}_2$, $VCdim(\mathcal{F}_1 \cup \mathcal{F}_2) = 0 + 0 + 1 = 1$.



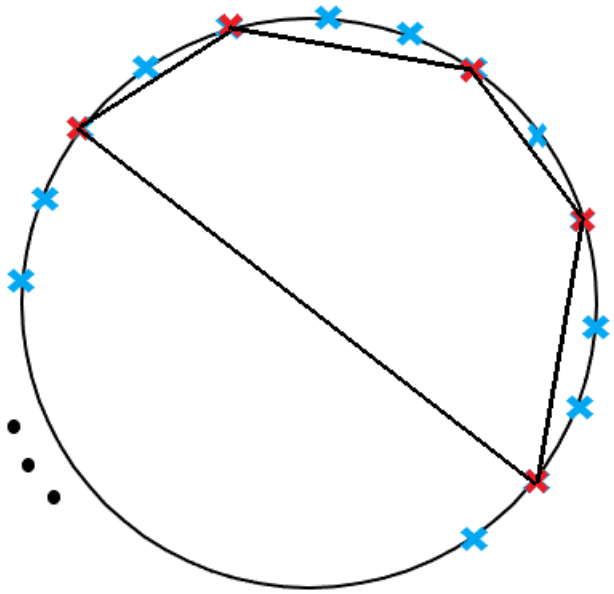


Thus, the VC dimension for union $\mathcal{F}_1 \cup \mathcal{F}_2$ is $\max(d_1, d_2) \leq VCdim(\mathcal{F}_1 \cup \mathcal{F}_2) \leq d_1 + d_2 + 1$, where $d_1 = VCdim(\mathcal{F}_1)$, $d_2 = VCdim(\mathcal{F}_2)$.

b.

VC dimension of convex polygons in $\mathcal{R}^2$ is ∞.

Suppose points placed on a circle with an arbitrary label assignment, if we connect all same-labeled neighbor points with a straight line, the shape of the lines is a convex polygon.



Thus, for any number of points, any ways of labeling, we can always find a convex polygon so that the points are shattered.

ii.

Suppose that we are using a voting algorithm $\mathcal{B}$, running algorithm $\mathcal{A}$ m times (m is odd) and let the m models vote for the output (to have the mode as output).

According to the question, we need to have algorithm $\mathcal{B}$ with at least $1 - \delta$ probability return an output which complies

$$\mathrm{err}\left(f_n^{\mathcal{B}}\right) - \inf_{f \in \mathcal{F}} \mathrm{err}(f) \leq \epsilon$$

thus, we need to guarantee within m times of running algorithm $\mathcal{A}$, we have at least $1-\delta$ probability that there are more 'correct' models than 'wrong' models, that is

$$\sum_{i=0}^{\frac{m-1}{2}} (0.45)^{m-i}(0.55)^{i} \leq \delta$$

$$(0.45)^{m}(0.55)^{0} + (0.45)^{m-1}(0.55)^{1} + \cdots + (0.45)^{\frac{m+1}{2}}(0.55)^{\frac{m-1}{2}} \leq \delta$$

$$\frac{(0.45)^{m} - (0.45)^{\frac{m+1}{2}}(0.55)^{\frac{m-1}{2}}\frac{0.55}{0.45}}{1-\frac{0.55}{0.45}} \leq \delta$$

$$\frac{(0.45)^{m} - (0.45)^{\frac{m}{2}}(0.55)^{\frac{m}{2}}\sqrt{\frac{0.45}{0.55}}\frac{0.55}{0.45}}{1-\frac{0.55}{0.45}} \leq \delta$$

$$1.1(0.45)^{\frac{m}{2}}(0.55)^{\frac{m}{2}} - (0.45)^{m} \leq 0.22\delta$$

For the convenience of computation, let's set the bound stricter,

$$1.1(0.45)^{\frac{m}{2}}(0.55)^{\frac{m}{2}} - (0.45)^{m} < 1.1(0.45)^{\frac{m}{2}}(0.55)^{\frac{m}{2}} \leq 0.22\delta$$

$$\frac{m}{2}\ln(0.2475) + \ln(1.1) \leq \ln(0.22\delta)$$

$$\frac{m}{2}\ln(0.2475) \leq \ln(0.2\delta)$$

$$0.7m \geq \ln(\frac{5}{\delta})$$

$$m \geq 1.43\ln(\frac{5}{\delta})$$

Consequently, we have

$$n' = mn \geq 1.43\ln(\frac{5}{\delta})O(\frac{1}{\epsilon^{2}})$$

$1.43\ln(\frac{5}{\delta})O(\frac{1}{\epsilon^{2}})$ is a polynomial of $\frac{1}{\delta}$ and $\frac{1}{\epsilon}$, thus $\mathcal{F}$ is efficiently PAC-learnable.

4.

i.

Let $f(x_i) = \sum_{i,j}(\|x_i - x_j\| - D_{ij})^2$

$$\frac{\partial f}{\partial x_i} = \sum_{i,j} 2(\|x_i - x_j\| - D_{ij})\frac{(x_i - x_j)}{\|x_i - x_j\|}$$

ii.

The code will be submitted in Coursework, here we will present the pseudo code.

Initialize position(pos), threshold(δ), updating step(η), position updating matrix($pos\_$)

while $pos\_ > \delta$:

$$pos\_(i) \leftarrow \sum_{i,j} 2(\|x_i - x_j\| - D_{ij}) \frac{(x_i - x_j)}{\|x_i - x_j\|}$$

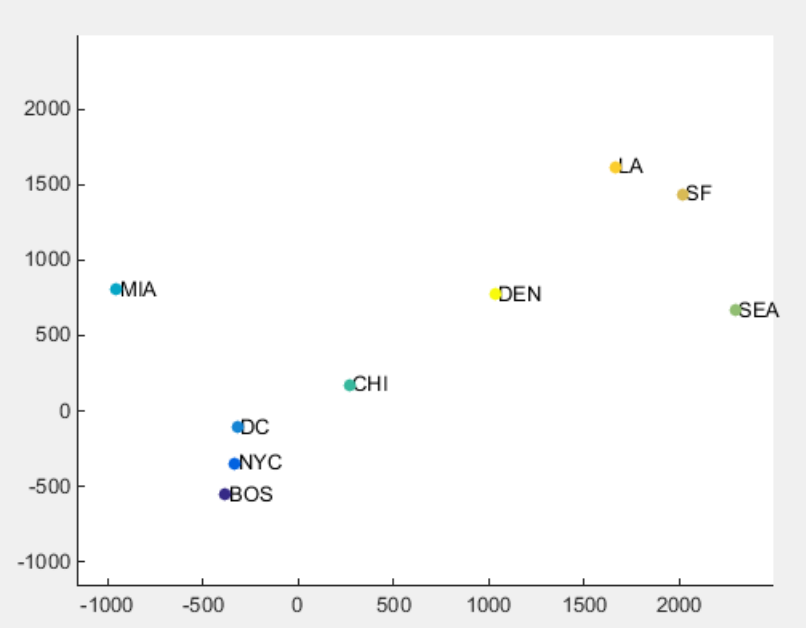
$$pos \leftarrow pos - pos\_ * \eta$$

end

iii.

One thing to note here, the result is sensitive to initial state, thus, choosing an appropriate initial value is very important.

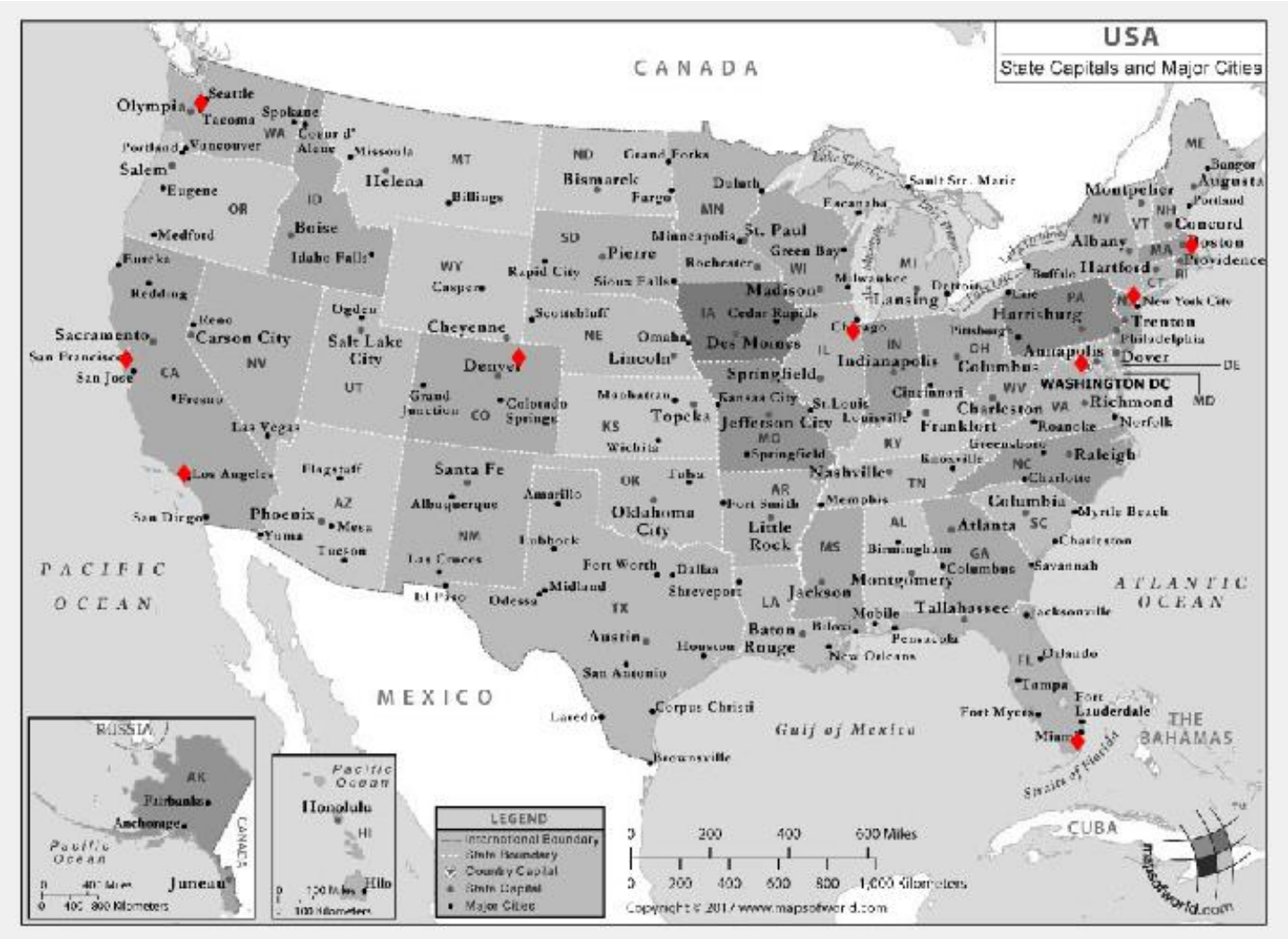
The learnt result, the relative positions of cities are basically right. However, it still looks different from a real map, as the angle is not set.



Here is a real map of US major cities. (https://www.mapsofworld.com/usa/usa-capital-and-major-cities-map.html)

Now, we can manually select the nine cities in this map, and use the coordinate of these points, we can compute the transform matrix from the relative position we have to the real position. Then, apply transformation matrix to the relative position, we can draw the result on the real map to evaluate the result.

The real map is drawn in gray image as it is too colorful. We have to change it to gray image to see the result clearly.



5.

For this question, I used linear regression method.

Firstly, I tried to use neural networks to make regression, but the score didn't meet the base line.

Then, I use linear regression and have an acceptable result. For details of the method, check this link. (http://mezeylab.cb.bscb.cornell.edu/labmembers/documents/supplement%205%20-%20multiple%20regression.pdf)

I also tried to use PCA on raw data, then apply linear regression. However, the result score decreased a little. So I gave up using PCA.