Machine Learning HW5

Kailun Chen
UNI: kc3327

2020.05.04

1. For the algorithm A, we can develop k times independent sampling to generate models $f_{n1}^A, f_{n2}^A, ..., f_{nk}^A$. Each of the model satisfies the following property:
for all $\epsilon > 0$, with probability 0.55

$$err(f_{ni}^A) - inf_{f \in F} err(f) <= \epsilon_1$$

From this, we know that the probability for all models that all the error not bounded is $(1 - 0.55)^k$. Then the probability that at least one model satisfying the property is $1 - (1 - 0.55)^k$.Of course, we have confidence that $1 - (1 - 0.5)^k$ that at least one model satisfying the property
Let $\epsilon_1 = \frac{\epsilon}{2}$, we need to calculate the number of k such that after ruining k times of the algorithm A, we will have confidence level $1 - \delta$ that for all $\epsilon_1 > 0$, there exist $i <= k$, the error is bounded.

$$1 - \delta = 1 - (1 - 0.5)^k$$

$$k = \frac{log(\delta)}{log(\frac{1}{2})}$$

According to the Chernoff-Hoeffding bound lemma, we have for any i within k

$$P(err(f_{ni}^A) - inf_{f \in F} err(f) >= \epsilon_1) <= 2e^{-2\epsilon_1^2 m}$$

$$\sum_{0<=i<=k} P(err(f_{ni}^A) - inf_{f \in F} err(f) >= \epsilon_1) <= 2ke^{-2\epsilon_1^2 m}$$

$$\sum_{0<=i<=k} P(err(f_{ni}^A) - inf_{f \in F} err(f) >= \epsilon_1) <= 2ke^{-2\epsilon_1^2 m} <= \delta$$

$$1 - \sum_{0<=i<=k} P(err(f_{ni}^A) - inf_{f \in F} err(f) >= \epsilon_1) >= 1 - \delta$$

We have for all $f \in F$ with probability $1 - \delta$, the error is bounded.
From the inequality, we have

$$n^{'} = m >= \frac{log(\frac{2k}{\delta})}{2\epsilon_1^2}$$

$$n^{'} >= \frac{2log(\frac{2k}{\delta})}{\epsilon^2}$$

where $n^{'}$ is the sample size for algorithm B.

For the constructed algorithm B, by the finite size F theorem, with sample size $n' >= \frac{2log(\frac{2k}{\delta})}{\epsilon^2}$ in the form of polynomial of $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$. This is enough to return a model such that the model class is efficiently PAC learn-able.

2. (a)

$$\frac{\partial}{\partial y_i} 2 \sum_{1<=j<=n} (||y_i - y_j|| - \pi_{ij})^2$$

$$\frac{\partial}{\partial y_i} 2 \sum_{1<=j<=n} ||y_i - y_j||^2 - 2\pi_{ij}||y_i - y_j||$$

$$\frac{\partial}{\partial y_i} 2 \sum_{1<=j<=n} (y_i - y_j)^T (y_i - y_j) - 2\pi_{ij}||y_i - y_j||$$

$$\frac{\partial}{\partial y_i} 2 \sum_{1<=j<=n} y_i^T y_i - 2y_i^T y_j - 2\pi_{ij}||y_i - y_j||$$

$$4 \sum_{1<=j<=n} y_i - y_j - \pi_{ij} \frac{y_i - y_j}{||y_i - y_j||}$$

(b)

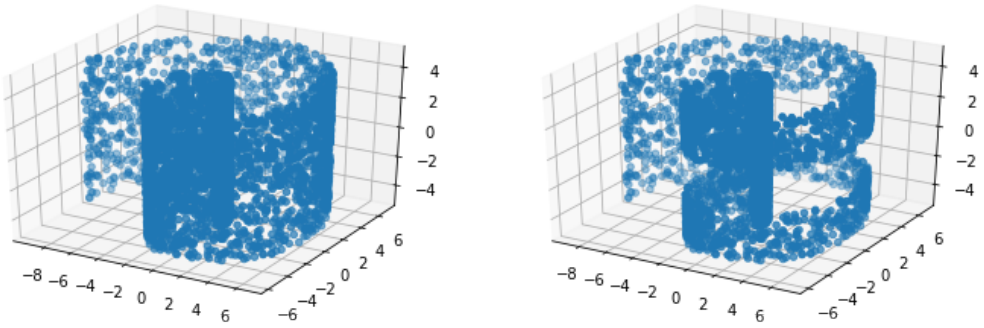
$$S = 4 \sum_{1<=j<=n} y_i - y_j - \pi_{ij} \frac{y_i - y_j}{||y_i - y_j||}$$

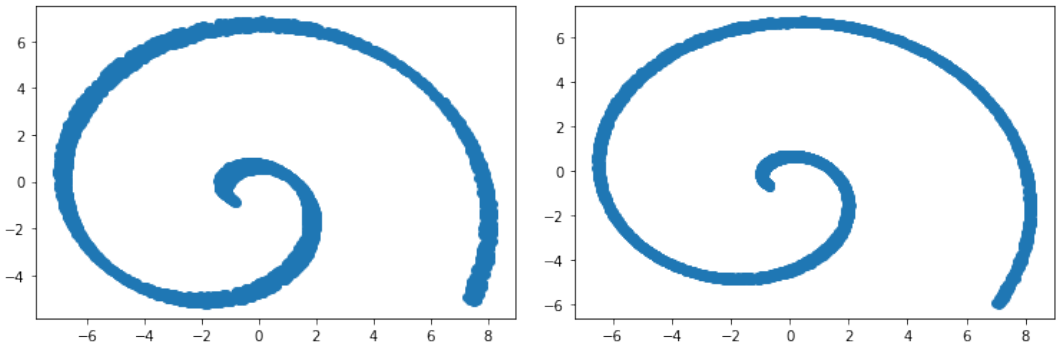
$$\frac{\partial}{\partial y_i} S = 4 \sum_{1<=j<=n} 1 - \pi_{ij} \frac{||y_i - y_j|| - \frac{(y_i - y_j)^2}{||y_i - y_j||}}{||y_i - y_j||^2} = 4n > 0$$

Thus, the second derivative is larger than zero, the function is convex.

(c) The code has been submitted via Coursework.

(d) The pca is a linear dimensionality reduction so that it can not capture some special features of the data. This is the reason why PCA of two datasets have the same pattern. 2D embedding may have the better performance than pca.




6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6
8
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
8