Machine Learning - HW3

Kai Liao

September 2020

All the questions that are not answered in this file can be found in related Jupyter Notebook

1 Statistical Learning: Hoeffding's Inequality

1.1 a

Proof. By Markov's inequality,

$$\Pr\left(X \geq \mu_X + t\right) = \Pr\left(X - \mu_X \geq t\right) \leq \mathbb{E}\left[e^{\lambda(X-\mu_X)}\right] e^{-\lambda t} \tag{1}$$

This inequality is true for each $\lambda > 0$, therefore,

$$\Pr\left(X \geq \mu_X + t\right) \leq \min_{\lambda \geq 0} \mathbb{E}\left[e^{\lambda(X-\mu_X)}\right] e^{-\lambda t} = \min_{\lambda \geq 0} M_{X-\mu_X}(\lambda) e^{-\lambda t} \tag{2}$$

□

1.2 b

$$\begin{aligned} \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu_{X_i}) \geq t\right) &= \mathbb{P}\left(\sum_{i=1}^{n}(X_i - \mu_{X_i}) \geq nt\right) \\ \text{By Chernoff bound, } &\leq \mathbb{E}\left[\exp\left(\lambda \sum_{i=1}^{n}(X_i - \mu_{X_i})\right)\right] e^{-\lambda nt} \\ &= \left(\prod_{i=1}^{n} \mathbb{E}\left[e^{\lambda\left(X_i - \mu_{X_i}\right)}\right]\right) e^{-\lambda nt} \\ \text{By Hoeffding's Lemma, } &\leq \left(\prod_{i=1}^{n} e^{\frac{\lambda^2(b-a)^2}{8}}\right) e^{-\lambda nt} \end{aligned} \tag{3}$$

Note that, the Chernoff bound is true for each $\lambda > 0$, therefore,

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu_{X_i}) \geq t\right) = \min_{\lambda \geq 0} \exp\left(\frac{n\lambda^2(b-a)^2}{8} - \lambda nt\right) = \exp\left(-\frac{2nt^2}{(b-a)^2}\right) \tag{4}$$

1.3 c

Consider a simple case with $n = 1$. Let X has a distribution as follow:

$$\begin{aligned} &\mathbb{P}\left(X = -0.001\right) = 1 - 10^{-3} \\ &\mathbb{P}\left(X = 10^3\right) = 10^{-3} \end{aligned} \tag{5}$$

Let $t = 1$, then the LHS is,

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu_{X_i}) \geq t\right) = \mathbb{P}\left((X - \mu_X) \geq 1\right) = 10^{-3} \tag{6}$$

The RHS is,

$$\exp\left(-\frac{2nt^2}{(b-a)^2}\right) = 0.99998 \tag{7}$$

We can see that the information provide by this inequality is very limited.

2 SVM and the power of kernels

2.1 a

By the Representer Theorem, the solution to the SVM problem,

$$f^* \in \operatorname{argmin}_{f \in H_k} \sum_{i=1}^{n} \ell\left(f\left(x_i\right), y_i\right) + \Omega\left(\|f\|_{H_k}^2\right) \tag{8}$$

can all be expressed in the form

$$f^* = \sum_{i=1}^{n} \alpha_i k\left(x_i, \cdot\right) \tag{9}$$

It suffice to show that

$$f^*(x_i) = y_i \tag{10}$$

for each i.

We can show that

$$f^*(x_i) = \sum_{k=1}^{n} \alpha_k k\left(x_k, x_i\right) = e_i^T K\alpha \tag{11}$$

where e_i is the i th standard basis vector. Write all i in metric form.

$$K\alpha = y \tag{12}$$

Suppose K is inevitable (for example, symmetric and strictly positive definite), set $\alpha = K^{-1}y$, we have $f^*(x_i) = y_i$ as desired.

2.2 b

Since this is a separable SVM, the decision boundary must be right at the middle of a and b:

$$x_d = \frac{a+b}{2} \tag{13}$$

The relation of λ and x_d is given by

$$\lambda x_d + \lambda_0 = 0 \tag{14}$$

WLOG, suppose a is a support vector. Then

$$\lambda a + \lambda_0 = 1 \tag{15}$$

We have 3 unknown variables and 3 equations. Therefore, solve for λ:

$$\lambda = \frac{2}{a-b} \tag{16}$$

Thus,

$$f(x) = \frac{2x}{a-b} - \frac{a+b}{a-b} \tag{17}$$

The dual problem constrain

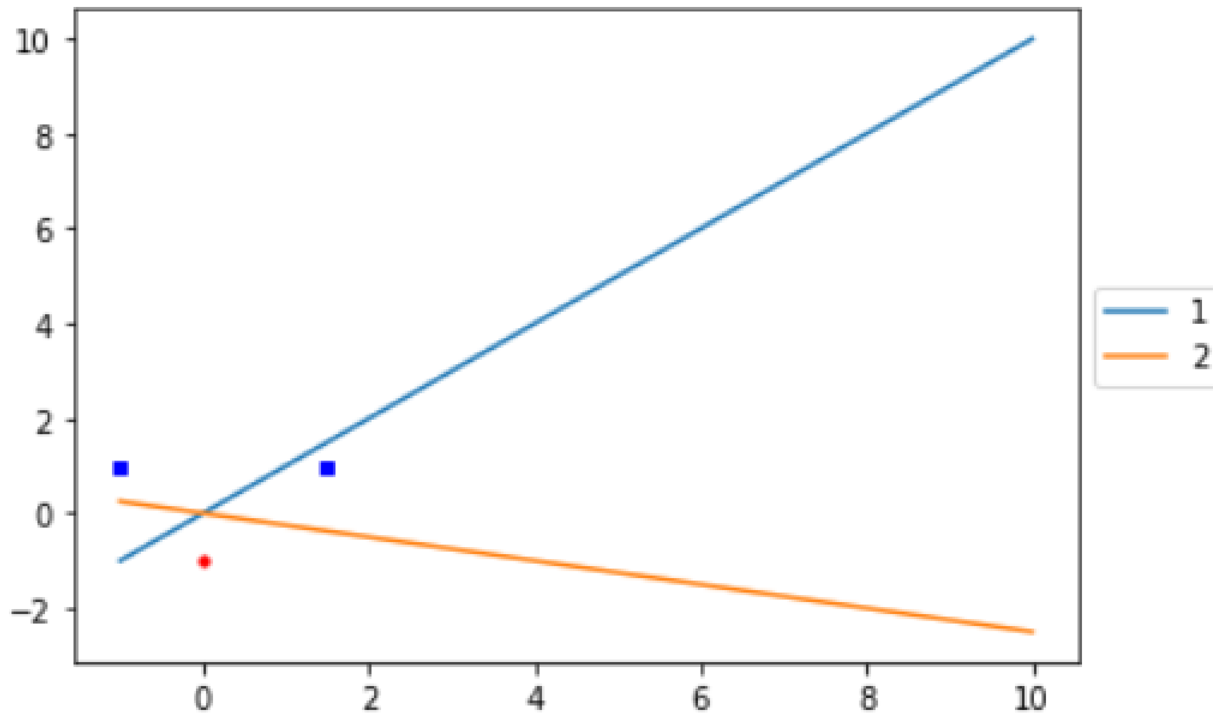
$$\alpha_a + \sum_{i \neq a} \alpha_i y_i = 0 \tag{18}$$

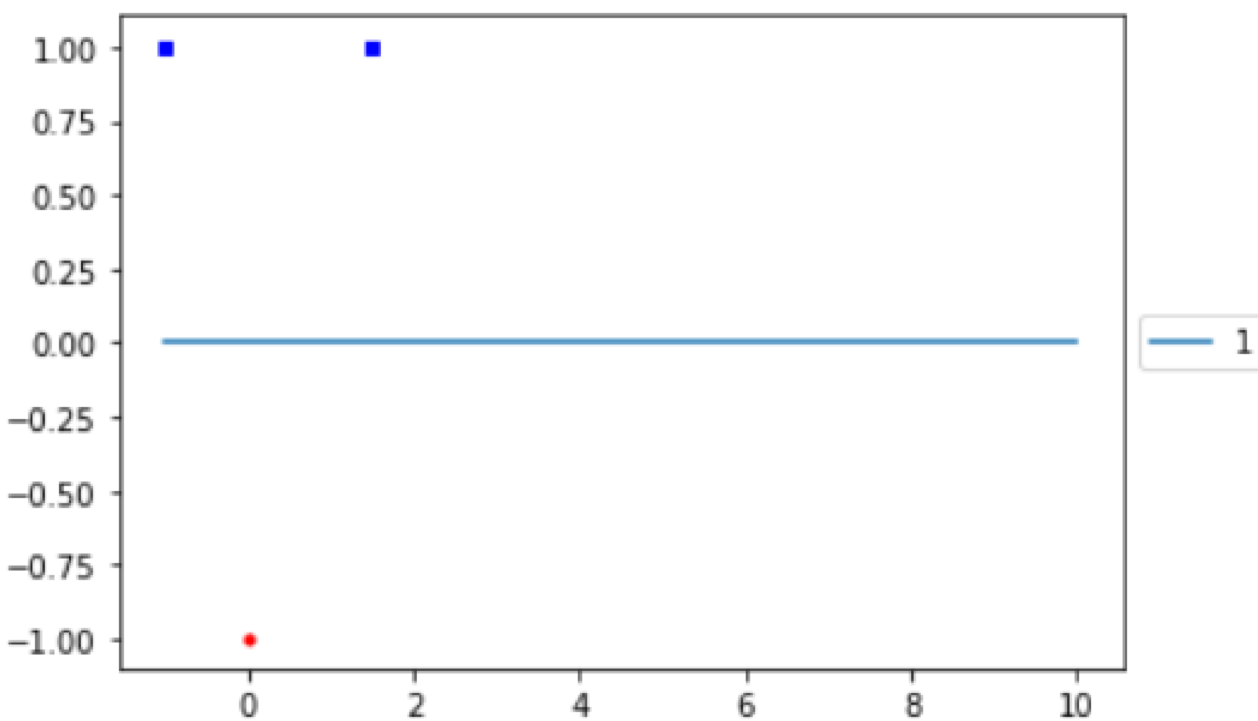
where $\alpha_a > 0$ suggests that $\sum_{i \neq a} \alpha_i y_i < 0$. Therefore, there must be at least one more support vector in the summation.

3 Perceptron: Theoretical

3.1 a

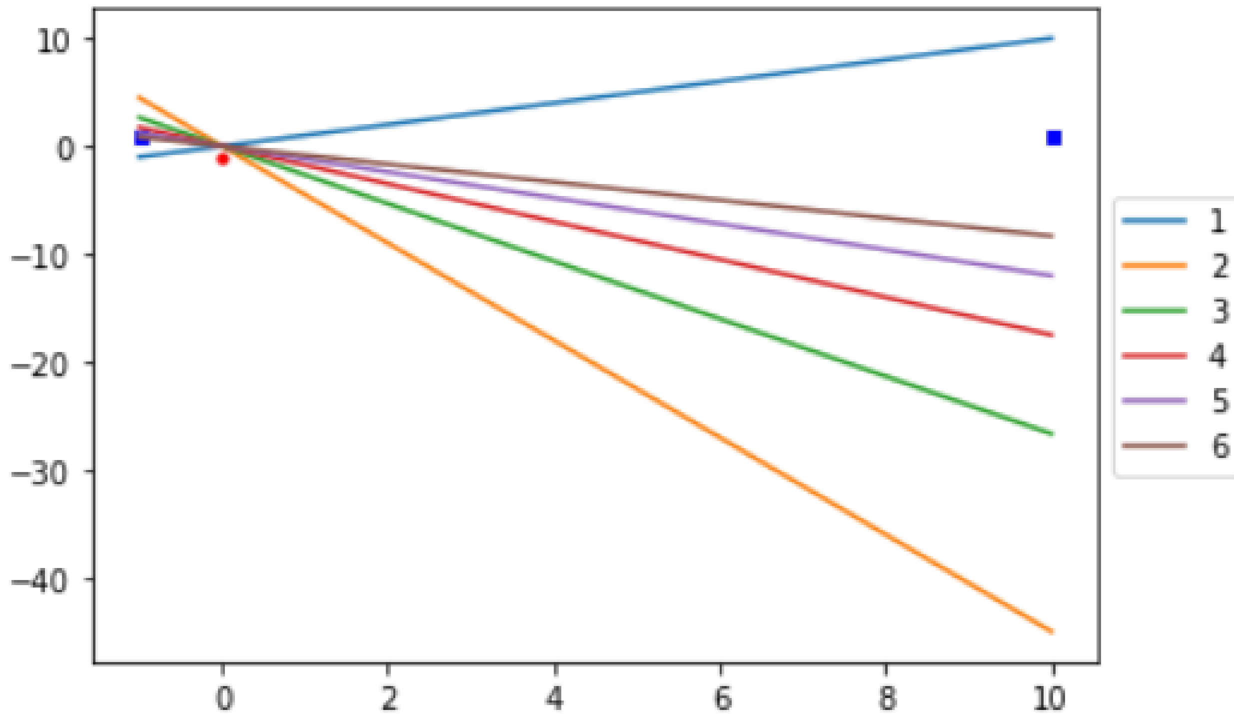
If the algorithm starts with point x_1, it takes 2 mistakes until convergence. We can see the boundary in the first figure below. If the algorithm stats with point x_2, it only takes 1 mistake until convergence. We can see the boundaries in the second figure below.

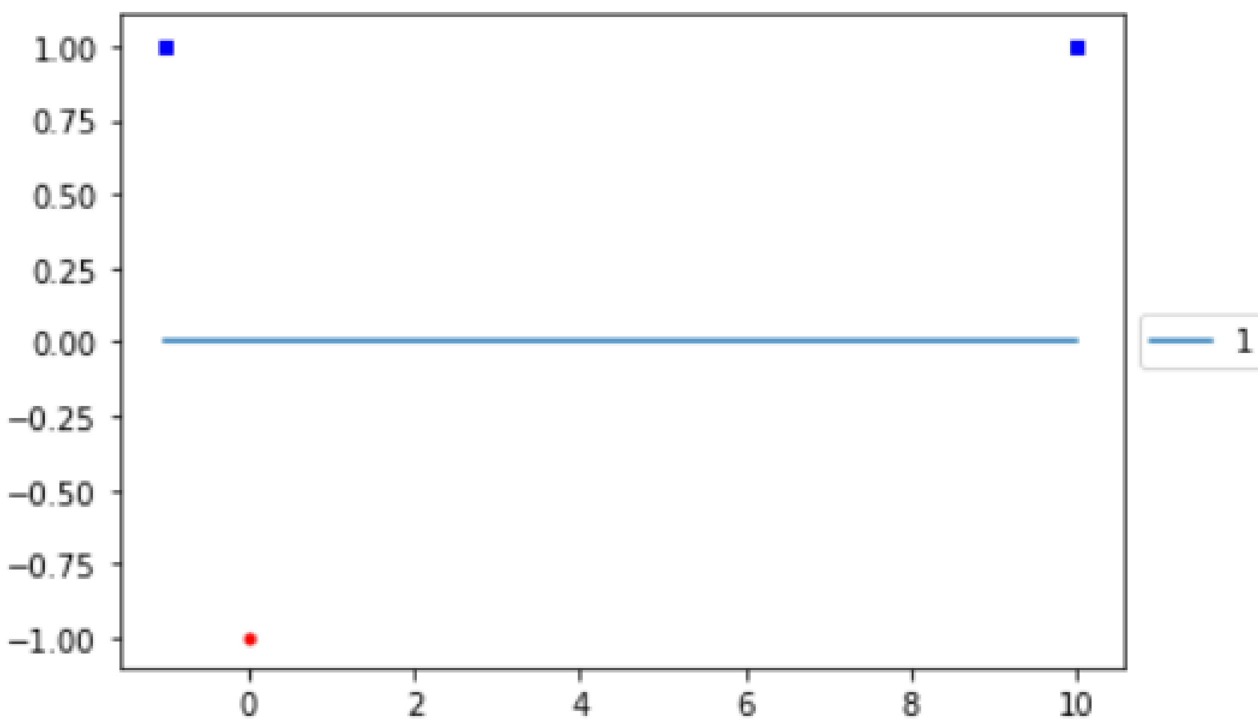




3.2 b

If the algorithm starts with point x_1, it takes 6 mistakes until convergence. We can see the boundary in the first figure below. If the algorithm stats with point x_2, it only takes 1 mistake until convergence. We can see the boundaries in the second figure below.





3.3 c

Firstly, we choose the first data point which can maximize the number of misclassified data points. Second;y, choose the closest misclassified data point to the boundary. This procedure makes sure that this new mistake (if it is a mistake) only provide very limited information, i.e., the boundary only moves a tiny bit after this update. Repeat the second step until we go over all data points.