Homework 3 for Machine learning: Theory and Algorithms

Gabriele Genovese

11 October 2024

1 Exercise 1

Let $f : R^n \to R$ be a convex function.

1.1 Point *a*

Let's prove that if f is concave, then f is also affine, that is $f(x) = ax + b$.

If f is convex, by definition $\forall x, y$ in its domain, and $\forall \lambda \in [0, 1]$, we have

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

$\lambda f(x) + (1 - \lambda)f(y)$ is rappresent the function that traverse the inside of f and it's an affine function.

Let's assume that f is also concave. We say that f is concave if $-f$ is convex and using the above formula we have

$$-f(\lambda x + (1 - \lambda)y) \leq -\lambda f(x) - (1 - \lambda)f(y) \implies f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$

Therefore, we have that

$$\lambda f(x) + (1 - \lambda)f(y) \leq f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

So, we have that

$$f(\lambda x + (1 - \lambda)y) = \lambda f(x) + (1 - \lambda)f(y)$$

f is a straight line and therefore it's an affine function.

1.2 Point *b*

Assume that f is convex and that $f(x) < M$ for all $x \in R^n$, where M is some upper bound. Let's prove by contradiction that f must be constant. Consider two distinct points $x_1 \neq x_2 \in R^n$. Let L be the line passing through the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$.

Since f is convex, the graph of f must lie on or above the line L connecting these two points, that is:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2), \quad \forall \lambda \in [0, 1]$$

Since $f(x) < M$ for all x, this implies that the line L must intersect the horizontal line $y = M$ at some point. However, this would contradict the assumption that f is strictly less than M everywhere.

It must be that $x_1 = x_2, \ \forall x_1, x_2 \in R^n$. This implies that $f(x)$ is constant for all $x \in R^n$.

2 Exercise 2

2.1 Point 1

By fixing i in (1), we get that for

$i = 1,\ h_n(x) = \{a_1 \text{ if } x \in (\frac{1}{2}, 1);\ 0 \text{ otherwise}\}$

$i = 2,\ h_n(x) = \{a_2 \text{ if } x \in (\frac{1}{3}, \frac{1}{2});\ 0 \text{ otherwise}\}$

$i = 3,\ h_n(x) = \{a_3 \text{ if } x \in (\frac{1}{4}, \frac{1}{3});\ 0 \text{ otherwise}\}$

$i = 4,\ h_n(x) = \{a_4 \text{ if } x \in (\frac{1}{5}, \frac{1}{4});\ 0 \text{ otherwise}\}$

...

On the other hand, by fixing i in (2), we get that for:

$i = 1,\ f_1(x) = \{1 \text{ if } x \in (\frac{1}{3}, \frac{1}{2});\ 0 \text{ otherwise}\}$

$i = 2,\ f_2(x) = \{1 \text{ if } x \in (\frac{1}{5}, \frac{1}{4});\ 0 \text{ otherwise}\}$

...

We can observe that the labeling function f_1 is correctly represented by a_2, that is a learner over n. The same holds for f_2 with a_4. We can say that the labeling function f_i is perfectly represented by a_{2i}. For larger i, the hypothesis class does not cover the corresponding intervals exactly, so there is some approximation error. The approximation error decreases as n increases: with a larger n, the hypothesis class includes more intervals, allowing for better representations of the labeling functions and the alignment between the H and the labeling functions improves, reducing in the approximation error.

2.2 Point 2

The bias-complexity tradeoff reflects the balance between approximation error and estimation error.

- **With small** n, the hypothesis class is too simple to capture the true function f, leading to high approximation error but low estimation error since the model is less prone to overfitting.
- **As n increases**, the model becomes more flexible reducing approximation error by better approximating f. However, this increase in complexity will also increase the estimation error, as more parameters need to be considered, leading to greater risk of overfitting.