Supervised Learning

Week 1

Supervised Learning Problem

Given a set of **input/output** pairs (**tranning set**) we wish to compute the functional relationship between the input and the output.

Learning approach

- Stable: finds something that is not chance part of set of examples.
- Efficient: infers(推断) solution in time polynomial in the size of the data
- Robust: should not be too sensitive to mislabelled/noisy examples

Supervised Learning Model

- Goal: Given training data(pattern, target) pairs

  $$\mathcal{S} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$$

  infer a function f_s such that

  $$f_{\mathcal{S}}(x_i) \approx y_i$$

  for the future data

  $$S' = \{(x_{m+1}, y_{m+1}), (x_{m+2}, y_{x+2}, \ldots\}$$

- $\mathcal{X}$: input space (eg, $\mathcal{X} \subseteq \mathbb{R}^d$), with elements $x, x', x_i, \ldots$
- $\mathcal{Y}$: output space, with elements $y, y', y_i, \ldots$

Learning Algorithm

- Traning set: $\mathcal{S} = \{(x_i, y_i)_{i=1}^m\} \subseteq \mathcal{X} \times \mathcal{Y}$
- A **learning algorithm** is a mapping $\mathcal{S} \to f_{\mathcal{S}}$
- A new input x is predicted as $f_{\mathcal{S}}(x)$

Learning Regression

$$\mathbf{X}\mathbf{w} = \mathbf{y}$$

the linear predictor

$$\hat{y} = \mathbf{w} \cdot \mathbf{x}$$

Note: $\mathbf{A} \cdot \mathbf{B} = A^T B$

Find a linear predictor $\hat{\mathbf{y}} = \mathbf{w} \cdot \mathbf{x}$ to minimize the square error over the data $\mathcal{S} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ thus

$$\textbf{Minimize}: \sum_{i=1}^{m} (y_i - \hat{y_i})^2 = \sum_{i=1}^{m} (y_i - \mathbf{w} \cdot \mathbf{x})$$

Thus in martix notation **empirical**(经验主义的) **mean (sqaure) error** of the linear predictor $\hat{y} = \mathbf{w} \cdot \mathbf{x}$ on the data sequence $\mathcal{S}$ is

$$\begin{aligned}
\varepsilon_{emp}(\mathcal{S}, \mathbf{w}) &= \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y_i})^2 \\
&= \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y_i})^2 \\
&= \sum_{i=1}^{m} (y_i - \mathbf{w} \cdot \mathbf{x})^2 \\
&= \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y_i})^2 \\
&= \sum_{i=1}^{m} (y_i - \sum_{j=1}^{n} w_j x_{i,j})^2 \\
&= \frac{1}{m} (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y})
\end{aligned}$$

To compute the minimum we solve for

$$\nabla_{\mathbf{w}} \varepsilon_{emp}(\mathcal{S}, \mathbf{w}) = 0$$

So we conclude that

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

K-nearest neighbours (KNN)

- Algorithm

  Let $N(\mathbf{x}; k)$ be the set of k nearest training inputs to $\mathbf{x}$ and

  $$I_{\mathbf{x}} = \{i : \mathbf{x_i} \in N(\mathbf{x}; k)\}$$

  the coresponding index set

$$f(\mathbf{x}) = \begin{cases} red & \text{if } \frac{1}{k} \sum_{i \in I_{\mathbf{x}}} y_i > \frac{1}{2} \\ green & \text{if } \frac{1}{k} \sum_{i \in I_{\mathbf{x}}} y_i \leq \frac{1}{2} \end{cases}$$

Perspectives(观点，前景) on supervised learning

Optimal Supervised Learning

- **Model**：We assume that the data is obatined by sampling from a **fixed but unknown** probability density $P(\mathbf{x}, y)$

  Expected error:

  $$\varepsilon(f) = E[(y - f(\mathbf{x})^2] = \int (y - f(\mathbf{x}))^2 dP(\mathbf{x}, y)$$

  Our goal is to minimize ε

- **Optimal solution**： $f^\star := \mathbf{argmin}_f \varepsilon(f)$ (Called Bayes estimator)

- Bayes estimator for square loss:

  Let us compute the optimal solution $f^\star$ for regression $\mathcal{Y} = \mathbb{R}$..

  Using the decomposition $P(y, \mathbf{x}) = P(y|\mathbf{x})P(\mathbf{x})$, we have

  $$\varepsilon(f) = \int_{\mathcal{X}} \{ \int_{\mathcal{Y}} (y - f(\mathbf{x}))^2 dP(y|\mathbf{x}) \} dP(\mathbf{x})$$

  So we may see that $f^\star$ is

  WHY?

  $$f^\star(x) = \int_{\mathcal{Y}} y dP(y|\mathbf{x})$$

  Deriving $f^\star$ with lighter notation

  $$f^\star(x) = \sum_{y \in Y} y p(y|x) = E[y|x]$$

  We now additionally assume there exist some underlying function F such that

  $$y = F(x) + \epsilon$$

  where ϵ is white noise, i.e., $E[\epsilon] = 0$ and finite variance.

  Thus the optimal prediction is

  $$f^\star(x) := E[y|x] = F(x)$$

  with square loss.

  We would like to understand the expected error by an arbitrary learner $A_S(x)$

  Our goal will be to understand the expected error at x'

  $$\varepsilon(A(x')) = E[(y' - A(x'))^2]$$

  where y' is a sample from $P(Y|x')$

  $$E[(y' - A(x'))^2] = E[(y' - f^\star(x'))^2] + (f^\star(x) - E[A(x')]^2) + E[(A(x') - E[A(x')])^2]$$

  - Bayes error: $E[(y' - f^\star(x'))^2]$

is the irreducible noise

- Bias: $(f^\star(x) - E[A(x')]^2)$

  describes the discrepancy(差异) between the algorithm and "truth"

- Variance: $E[(A(x') - E[A(x')])^2]$

  capture the variance of the algorithm between training sets.

- Bias and Variance Dilemma
  - The bias and variance tend to trade off against one another
  - Many parameters better flexibility to fit the data thus low bias but high variance
  - Few parameters give high bias but the fit between different data sets will not change much thus low variance
  - This exact decomposition only holds for the square loss.

Asymptoitic(渐近的) Optimality of k-NN

- As the number samples goes to infinity the error rate is no more than twice the Bayes error rate.

  TBC

Hypothesis Space

We introduce a **restricted** space of functions $\mathcal{H}$ called **hypothesis space**.

We minimize $\varepsilon_{emp}(\mathcal{S}, f)$ with $\mathcal{H}$. That is, our learning algorithm is:

$$f_{\mathcal{S}} = \mathbf{argmin}_{f \in \mathcal{H}} \varepsilon_{emp}(\mathcal{S}, f)$$

This approach is usually called **empirical error(risk) minimization**

For example (Least Squares):

$$\mathcal{H} = \{f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} : \mathbf{w} \in \mathbb{R}^n\}$$

Summary

- Data $\mathcal{S}$ sampled i.i.d from P (fixed but unknown)
- $f^\star$ is what we want, $f_{\mathcal{S}}$ is what we get
- Different approaches to attempt to estimate/approximate $f^\star$:
  - Minimize ε_{emp} in some restricted space of functions (eg, linear)
  - Compute local approximation of $f^\star$ (k-NN)
  - Estimate P and then use Bayes rule...

Model selection