# Supervised learning

- $g_\theta(x)$ is our estimate of $\eta(x) = \mathbb{E}[Y \mid X = x]$.
- Given iid $\{x_i, y_i\}_{i=1}^n$ $\quad l_i(\theta) = l(g_\theta(x_i), y_i)$

$$R_n(\Theta) = \frac{1}{n}\sum_{i=1}^{n} l_i(\theta) \xrightarrow[n\to\infty]{\mathbb{P}} R(\theta)$$

def So empirical risk minimization (ERM) make sense

$$\min_{\theta \in \Theta} R_n(\theta) \qquad \left(\hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} R_n(\theta)\right)$$

Let $\hat{\theta}$ be the ERM then $\mathbb{E} R_n(\hat{\theta}) \leq R(\hat{\theta})$ (why?) which motivates training-test split,

$$R_{train}(\theta) = \frac{1}{n_0}\sum_{i=1}^{n_0} l_i(\theta) \quad \text{is training error}$$

$$R_{test}(\theta) = \frac{1}{n-n_0}\sum_{i=n_0+1}^{n} l_i(\theta) \quad \text{is test error}$$

by iid $R_{train}(\theta) \xrightarrow[n_0\to\infty]{\mathbb{P}} R(\theta)$ and $R_{test}(\underbrace{\hat{\theta}}_{\text{ERM}}) \xrightarrow[n-n_0\to\infty]{\mathbb{P}} R(\hat{\theta})$.

notes API

```
class Learner:
    :
    def fit(self, train):
    def predict(self, test_X):
```

Validation

If there is a tuning parameter ($k$) then split into 3 sets

| train | Validation | Test |
| --- | --- | --- |

$$\hat{\theta}_k = \min_{\theta \in \Theta_k} R_{train}(\theta)$$

$$\hat{k} = \min R_{valid}(\hat{\theta}_k)$$