Supervised Learning (cont)

Chap2

1. Noise — any anomaly in the data. Difficult/impossible to get zero error.

2. Imprecision in recording input attributes. Data points may be in incorrect positions.

3. Errors in labeling the data points (teacher noise).

4. There may be additional hidden attributes that affect the label of an instance.

No simple rectangle is able to separate the true examples ☑ from the false examples ☐.

A more complex classifier may be needed. But there are tradeoffs.

The rectangle classifier:

1. is simple to use.
2. is simple to train and use with few parameters.
3. is simple to explain.

Occam’s Razor — simpler explanations are more plausible.
Learning Multiple Classes

In general, we have $K$ classes

$C_i : i = 1, 2, \ldots, K$.

Training set $X = \bigcup x^t, r^t \bigcup_{t=1}^N$

$r^t_i = \left\{ \begin{array}{ll} 1, & y^t e C_i \\ 0, & y^t e C_j : j \neq i. \end{array} \right.$

$r = (r_1, r_2 \ldots r_K)$

- Family Car
- Sports Car
- Luxury Sedan

We need to learn a $K$-hypothesis

$h_i(x^t) = \left\{ \begin{array}{ll} 1, & y^t \in C_i \\ 0, & y^t \in C_j : j \neq i. \end{array} \right.$

For most data, there will be a unique label.
For some data, there may be no clear class assignment.
Regression

Output is a numeric value (i.e., not $y$)

Training Set \( X = \{ x_t, r_t \}_{i=1}^{N} \) \( r_t \in \mathbb{R} \)

called interpolation if no noise.

Find a function \( f(x) \) s.t. \( r_t = f(x_t) \)

polynomial interpolation, given \( N \) points, we find the \((N-1)\)st degree polynomial.

Regression, noise is added.

\[ r_t = f(x_t) + \varepsilon_t \quad \varepsilon_t \text{- random noise} \]

Noise - unknown hidden variables.

\[ r_t = f^*(x_t, z_t) \quad z_t \text{- hidden variable} \]

Want to approximate the output by a function \( g(x) \)

Empirical error: \[ \mathbb{E}(g|X) = \frac{1}{N} \sum_{i=1}^{N} (r_t - g(x_t))^2 \]
Regression (cont)

Goal: find \( g(\cdot) \) that minimizes the empirical error.

Assume a hypothesis class for \( g(\cdot) \)

E.g. \( g(\cdot) \) is linear

\[
g(x) = \omega_1 x_1 + \cdots + \omega_d x_d + \omega_0 = \sum_{j=1}^{d} \omega_j x_j + \omega_0
\]

Back to simple case - single attribute \( x_1 \)

\[
E(\omega_1, \omega_0 | x) = \frac{1}{N} \sum_{t=1}^{N} (y^t - (\omega_1 x_1^t + \omega_0))^2
\]

Minimize: \( \frac{\partial E}{\partial \omega_1} = 0 \) \& \( \frac{\partial E}{\partial \omega_0} = 0 \)

solution

\[
\begin{align*}
\hat{\omega}_1 &= \frac{\overline{x}^t \overline{r}^t - \overline{x} \overline{r} N}{\sum_{t=1}^{N} (x^t r^t - x \overline{r})^2} \\
\hat{\omega}_0 &= \frac{\overline{x} \overline{r}^t - \omega_1 \overline{x} \overline{r}}{\sum_{t=1}^{N} (x^t r^t - x \overline{r})^2}
\end{align*}
\]

where \( \overline{x} = \frac{1}{N} \sum_{t=1}^{N} x_t \), \( \overline{r} = \frac{1}{N} \sum_{t=1}^{N} r_t \)

A "richer" model can be used.

E.g. \( g(x) = \omega_2 x^2 + \omega_1 x + \omega_0 \)

Too high an order follow the data too closely.
Model Selection and Generalization

Consider learning a Boolean function

If d inputs, 2^d examples (at most)
Each example can be labeled 0 or 1.
Hence 2^d possible Boolean functions of d variables

Each training example removes half the hypotheses (see table)

Learning as a way to remove hypotheses consistent with the data
But we need to see 2^d examples to learn!

Ill-posed: Data is not sufficient to determine the hypothesis (unless we have all the data)
Learning is ill-posed, we cannot learn unless we have an inductive bias
- assumptions about the set of hypotheses

But each hypothesis class has finite capacity and can only learn some functions
Model Selection (cont)

Learning needs an inductive bias.

Model selection: how to choose the right bias?

Want the model to be able to generalize — predict new data — even more than fitting the training dataset.

Best generalization requires matching the complexity of the hypothesis with the complexity of the function underlying the data.

Underfitting — if hypothesis too simple.
Overfitting — if hypothesis too complex.

Trade-off between three factors:
• The complexity of the hypothesis we fit to data, the capacity of the hypothesis class
• The amount of training data
• The generalization error on new examples

As training data increases, generalization error decreases.
As complexity of model increases, generalization error then increases.
Model Selection (cont.)

To measure generalization ability, we need new data — validation set. Or divide dataset into two parts: training data and validation set.

Cross-validation: pick hypothesis that does best on validation set.

Use training set to create hypothesis; validation set to do model selection; test set to evaluate performance.
Recapitulation & Generalize.

Sample \( X = \{ (x_i, r_i) \}^n \).

Sample is i.i.d. (Independent & Identically Distributed).
All instances drawn from the same, but unknown, distribution \( P(x, r) \).

Need to choose:

(i) the model \( g(x; \theta) \)
   define the hypothesis class
(ii) loss function - error function.
   \( E(\theta | x) = \mathbb{E}_r \{ L(r, g(x; \theta)) \} \)
(iii) optimization procedure
   to find \( \hat{\theta} = \arg \min \ E(\theta | x) \)

Following conditions required:

- Hypothesis class \( g(\cdot) \) must be big enough.
- Enough training data to find best hypothesis.
- Need good optimization method.