03 Supervised Learning

Dr. Nuhman Ul Haq

CUI Abbottabad

February 25, 2025

Outline

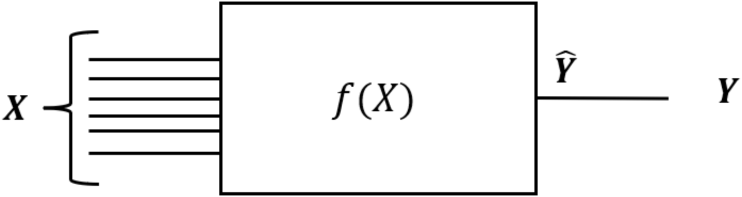
Concepts involved in Supervised Learning

1. Learning Class from Examples
2. Vapnik-Chervonenkis Dimension
3. Probably Approximately Correct Learning
4. Noise
5. Learning Multiple Classes
6. Regression
7. Model Selection and Generalization
8. Dimensions of a Supervised Machine Learning Algorithm

Supervised Learning

1. The supervised Learning Task is problem of mapping some input to predefined outputs

$f(X)$ **: Mapping function Known as model in Machine Learning**



X **is input data**

Y **is predetermined outcome/label**

$\widehat{Y}$ **is predicted outcome of the model**

Learning a Class from Examples (Classification)

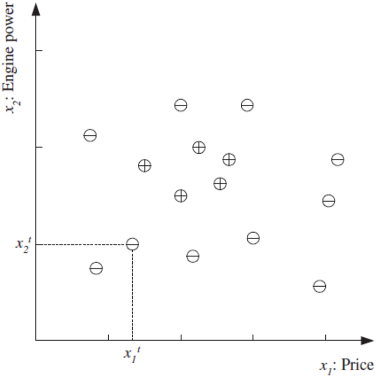
Problem: Learning of class C as a ”Family Car”

- **Data Preparation:**
  The people look at the cars and label them; the cars that they believed family cars are **positive examples**, and the other cars are **negative examples**.
- Class learning is finding a description that is shared by all the positive examples and none of the negative examples.
- After some discussions with experts in the field, it is concluded that two features **price** and **engine power** separate family cars from other cars.
- Note that when we decide on this particular input **representation**, we are ignoring various other attributes as irrelevant.

Learning a Class from Examples (Classification)

Problem: Feature Space representation of class C as a "Family Car"

Training set for the class of a "family car." Each data point corresponds to one example car, and the coordinates of the point indicate the price and engine power of that car.



Learning a Class from Examples (Classification) Data Representation

Problem: Feature Space representation of class C as a "Family Car" Let us denote price as the first input attribute x_1 and engine power as the second attribute x_2
Each car is represented by two numeric values as input

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and the labels are denoted as

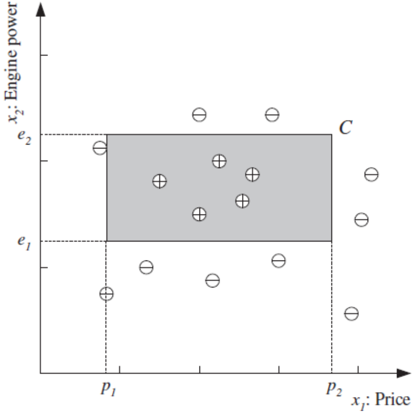
$$f(x) = \begin{cases} 1 & \text{if x is positive example} \\ 0 & \text{if x is negative example} \end{cases}$$

Each car data is represented ordered pair (x, r) and training set contains m such examples that is represented as

$$\mathcal{X} = \{x^t, r^t\}_{t=1}^{m}$$

Visualization of Hypothesis class with input features

- Training data is plotted in the two dimensional (x_1, x_2) space where each instance t is data point
  - At coordinates (x_1^t, x_2^t)
  - Hypothesis class as $\oplus$ for positive class and $\ominus$ for negative class.



How learning Model looks?

1. Model is the function that maps some input representation to some output representation.
2. What we have:
   1. training set $\mathcal{X}$
   2. Hypothesis $h(x)$
   3. target function $C(x)$
   4. **Objective:** evaluate how well $h(x)$ matches $C(x)$
   5. **Evaluation:** *empirical error* is a portion of training instances where *prediction* of h do not match the required values given in X.

The error of hypothesis h given the training set X is

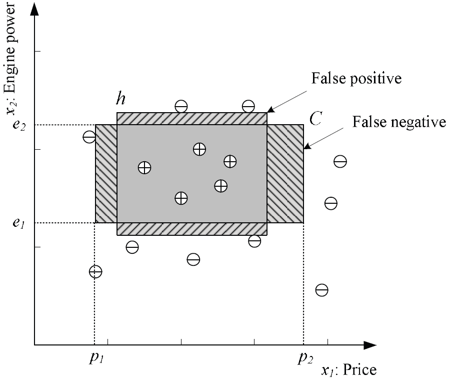
$$E(h|X) = \sum_{i=1}^{m} 1 \text{ if } h(x^t) \neq r^t$$

Hypothesis Space $\mathcal{H}$

1. Hypothesis Space ($\mathcal{H}$) is the set of all hypothesis based on input domain.
2. Each quadruple $< p_1^h, p_2^h, e_1^h, e_2^h >$ defines one hypothesis, h, from $\mathcal{H}$
3. **Generalization:** how well our hypothesis will correctly classify future examples that are not part of the training set.
4. **Most Specific Hypothesis:** One possibility is to find the most specific hypothesis, S, that is the tightest rectangle that includes all the positive examples and none of the negative examples.
5. **The most general hypothesis**, G, is the largest rectangle we can draw that includes all the positive examples and none of the negative examples
6. **Version Space:** The set of all hypothesis that holds for correctly classify the training examples

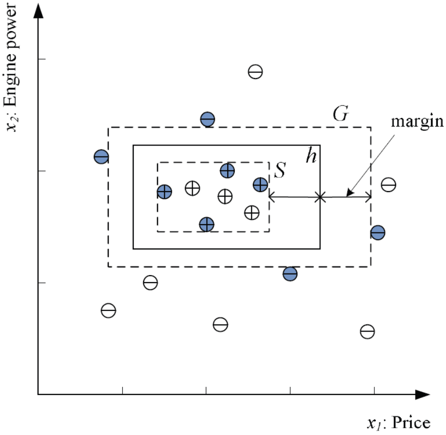
Possibility of Errors

$\mathcal{C}$ is the actual class and h is our induced hypothesis. The point where $\mathcal{C}$ is 1 but h is 0 is a false negative, and the point where $\mathcal{C}$ is 0 but h is 1 is a false positive. Other points—namely, true positives and true negatives—are correctly classified.



Best strategies for class separation

We choose the hypothesis with the largest margin, for best separation. The shaded instances are those that define (or support) the margin; other instances can be removed without affecting *h*.

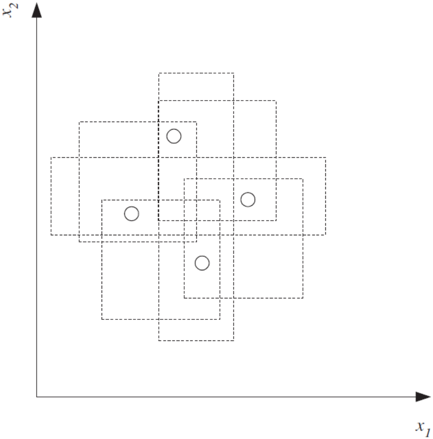


Vapnik-Chervonenkis Dimension

1. Let us say we have a dataset containing N points. These N points can be labeled in 2^N ways as positive and negative leads to 2^N learning problems.
2. If for any of these problems, we can find a hypothesis $h \in \mathcal{H}$ that separates the positive examples from the negative, then we say $\mathcal{H}$ shatters N points.
3. The maximum number of points that can be shattered by $\mathcal{H}$ is called the Vapnik-Chervonenkis (VC) dimension of $\mathcal{H}$, is denoted as $\mathcal{VC}(\mathcal{H})$, and measures the capacity of $\mathcal{H}$.

Vapnik-Chervonekis Dimensions

1. An axis-aligned rectangle can shatter four points. Only rectangles covering two points are shown.
2. VC dimension may seem pessimistic. It tells us that using a rectangle as our hypothesis class, we can learn only datasets containing four points and not more. A learning algorithm that can learn datasets of four points is not very useful. However, this is because the VC dimension is independent of the probability distribution from which instances are drawn.



Probably Approximately Correct Learning

In *probably approximately correct (PAC)* learning, given a class, C, and examples drawn from some unknown but fixed probability distribution, $p(x)$, we want to find the number of examples, N, such that with probability at least $1 - \delta$, the hypothesis h has error at most ϵ, for arbitrary $\delta \leq \frac{1}{2}$ and $\epsilon > 0$

$$P\{C \Delta h \leq \epsilon\} \geq 1 - \delta$$

where $C \Delta h$ is the region of difference between C and h.
In our case, because S is the tightest possible rectangle, the error region between C and $h = S$ is the sum of four rectangular strips

Probably Approximately Correct Learning

1. We would like to make sure that the probability of a positive example falling in here (and causing an error) is at most ϵ.
2. Probability is upper bounded by $\frac{\epsilon}{4}$, the error is at most $4(\frac{\epsilon}{4}) = \epsilon$.
3. The probability that a randomly drawn example misses this strip is $1 - \epsilon/4$.
4. The probability that all N independent draws miss the strip is $(1 - \epsilon/4)^N$, and the probability that all N independent draws miss any of the four strips is at most $4(1 - \epsilon/4)^N$, which we would like to be at most δ.

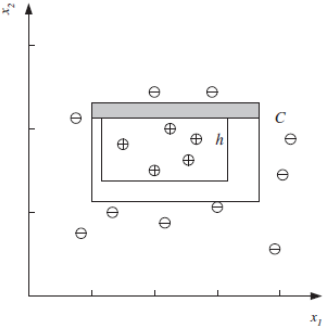


Figure: The difference between h and C is the sum of four rectangular strips, one of which is shaded.

Probably Approximately Correct Learning

1. We have the inequality

$$(1 - x) \leq \exp[-x]$$

2. For appropriate N and δ the equation will be

$$4\exp[-\epsilon N/4] \leq \delta$$

3. we can also write $4(1 - \epsilon/4)^N \leq \delta$. Dividing both sides by 4, taking (natural) log and rearranging terms, we have

$$N \leq (4/\epsilon)\log(4/\delta)$$

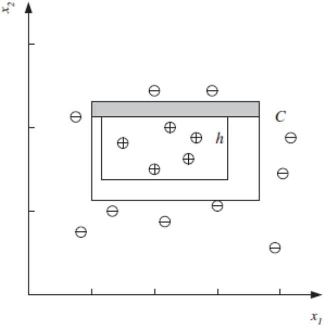
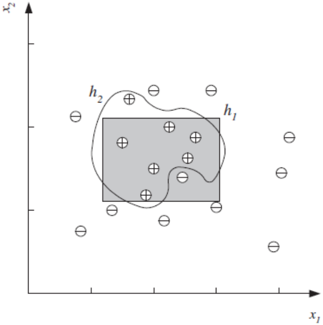


Figure: The difference between h and C is the sum of four rectangular strips, one of which is shaded.

Noise

Noise is any unwanted anomaly in the data and due to noise, the class may be more difficult to learn and zero error may be infeasible with a simple hypothesis class

1. There may be imprecision in recording the input attributes, which may shift the data points in the input space.
2. There may be errors in labeling the data points, which may relabel positive instances as negative and vice versa. This is sometimes called teacher noise.
3. There may be some attributes which are hidden or latent and may be non-observable and modeled as random components.



Learning Multiple Classes

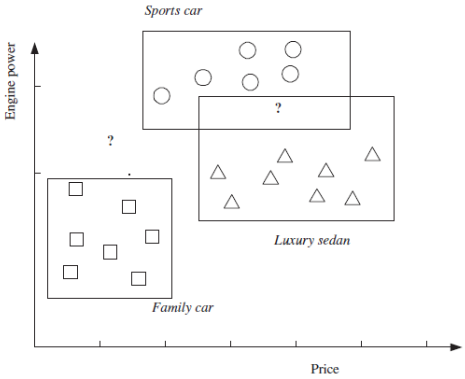


Figure: There are three classes: family car, sports car, and luxury sedan. There are three hypotheses induced, each one covering the instances of one class and leaving outside the instances of the other two classes. '?' are reject regions where no, or more than one, class is chosen.

Learning Multiple Classes

1. The example in previous slides is a two class problem
2. In the general case, we have K classes denoted as $C_i, i = 1, 2, ..., K$ and an input instance belongs to one and exactly one of them.
3. The training set $\mathcal{X}$ is now

$$\mathcal{X} = \{x^t, r^t\}_{t=1}^N$$

Where r has K dimensions

$$r_i^t = \begin{cases} 1 & \text{if } x^t \in \mathcal{C}_i \\ 0 & \text{if } x^t \in \mathcal{C}_j \ , \ j \neq i \end{cases}$$

4. The total empirical error takes sum over the prediction of all the classes over all instances:

$$E(\{h_i\}_{i=1}^K | \mathcal{X}) = \sum_{t=1}^N \sum_{i=1}^K 1(h_i(x^t) \neq r_i^t)$$

Regression

1. Classification Problem generate Boolean of Discrete values. And the learning function / target function as $\mathcal{C}(x) \in \{0, 1, 2, ..., K\}$
2. When the output is a numeric value and the target function is numeric function.
3. The training data is represented as

$$\mathcal{X} = \{x_t, r_t\}_{t=1}^{N} \text{ where } r^t \in \mathcal{R}.$$

If there is no noise, the task is interpolation. We would like to find the function $f(x)$ that passes through these points such that we have

$$r_t = f(x_t)$$

4. The empirical error is measured as **Mean Squared Error (MSE)**

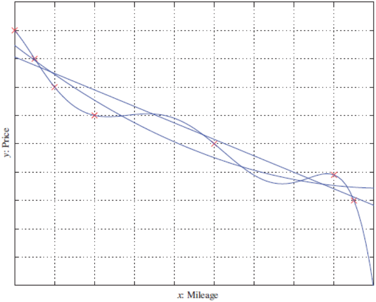
$$E(g|X) = \frac{1}{N} \sum_{t=1}^{N} [r_t - g(x_t)]^2$$

Regression

1. The approximation function presented in Error equation is the model that need to be learn to minimize the error. The model $g(x)$ is linear , and shape of the model

$$g(x) = w_1 x_1 + \cdots + w_d x_d + w_0$$

$$= \sum_{j=1}^{d} w_j x_j + w_0$$



Model Selection and Generalization

1. In a Boolean function, all inputs and the output are binary. There are 2^d possible ways to write d binary values and therefore, with d inputs, the training set has at most 2^d examples.
2. Generally, each training example removes some hypothesis among hypothesis space. The objective of the learning function is to get distinct hypothesis.
3. If hypothesis space is not tends to distinct hypothesis, such problems are known as **ill posed problem**, the reason is insufficient data.
4. For ill posed problem problem we consider some extra assumptions reach unique solution, such assumptions are known as **inductive bias** of learning algorithm.
5. In learning the class of family cars, there are infinitely many ways of separating the positive examples from the negative examples.
6. Assuming the shape of a rectangle is one inductive bias

Model Selection and Generalization (Matching data and model complexity)

1. For example in linear regression of one variable, we use other complex hypothesis like polynomial regression.
2. Thus learning is not possible without inductive bias, and now the question is how to choose the right bias?
3. This is called model selection, which is choosing between possible $\mathcal{H}$.
4. How well a model trained on the training set predicts the right output for new instances is called **generalization**.
5. If $\mathcal{H}$ is less complex than the function, we have under-fitting
6. if there is noise, an over-complex hypothesis may learn not only the underlying function but also the noise in the data and may make a bad fit, for example, when fitting a sixth-order polynomial to noisy data sampled from a third-order polynomial. This is called **over-fitting**

Dimensions of a Supervised Machine Learning Algorithm

1. Let us now recapitulate and generalize.
2. **Data:**

$$\mathcal{X} = \{x_t, r_t\}_{t=1}^{N}$$

3. The sample is independent and identically distributed (IID); the ordering is not important and all instances are drawn from the same joint distribution $p(x, r)$.
4. t indexes one of the N instances, x^t is the arbitrary dimensional input, and r^t is the associated desired output.
5. r^t is $0/1$ for **binary classification** learning, for **multiple classification** a $K - \textit{dimensional}$ binary vector (where exactly one of the dimensions is 1 and all others 0) and is a **real value in regression**.

Dimensions of a Supervised Machine Learning Algorithm

1. The aim is to build a good and useful approximation to r^t using the model $\mathbf{g}(x^t|\theta)$.
2. To achieve good approximation we consider three things
    1. **Model** we use in learning, denoted as $\mathbf{g}(\mathcal{X})|\theta$, where $\mathbf{g}(.)$ is the model, with $\mathcal{X}$ as the input and θ as learning parameters.
    2. **Loss Function** $L(.)$ is the difference between the desired output, r^t , and our approximation to it, $g(x^t|\theta)$, given the current value and the parameters, θ. The approximation error, or loss, is the sum of losses over the individual instances

       $$E(\theta|X) = \sum_t L(r_t, g(x_t|\theta))$$

    3. **Optimization procedure** to find θ^* that minimizes the total error

       $$\theta^* = \arg\min_\theta E(\theta|X)$$