Module1 I – Syllabus

1.1 Introduction to Machine Learning, 1.2 Examples of Machine Learning applications – 1.3 Learning associations, 1.4 Classification, 1.5 Regression, 1.6 Unsupervised Learning, 1.7 Reinforcement Learning. 1.8 Supervised learning- 1.9 Input representation, 1.10 Hypothesis class, 1.11 Version space, 1.12 Vapnik-Chervonenkis (VC) Dimension

1.1 Introduction To Machine Learning

➢ **What do you mean by Machine Learning?**

"the field of study that gives computers the ability to learn without being explicitly programmed." - Arthur Samuel

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. **The primary aim is to allow the computers learn automatically** without human intervention or assistance and adjust actions accordingly.

Types of Machine Learning

- Supervised – Classification, Regression, Association learning
- Unsupervised – Clustering
- Reinforcement Learning – Q learning

A computer program is said to ***learn*** from **experience E** with respect to some class of **tasks T** and **performance measure P** , if its performance at tasks T , as measured by P , improves with experience E.

Example

Handwriting recognition learning problem

• Task T : Recognising and classifying handwritten words within images

• Performance P : Percent of words correctly classified

• Training experience E: A dataset of handwritten words with given classifications

1.2 Examples of Machine Learning Applications (**can include topics in 1.3 to 1.7**)

1.3 Learning Associations

➢ **Explain association rule learning with an example**

In the case of retail—for example, a supermarket chain—one application of machine learning is *basket analysis*, which is finding associations between products bought by customers: If people who buy X typically also buy Y, and if there is a customer who buys X and does not buy Y, then is he or she a potential Y customer. Once we find such customers, we can target them for cross-selling.

In finding an *association rule*, we are interested in learning a conditional probability of the form $P(Y|X)$ where Y is the product we would like to condition on X, which is the product or

the set of products which we know that the customer has already purchased.

Let us say, going over our data, we calculate that *P(*chips | beer*)* = 0.7.

Then, we can define the rule:

70 percent of customers who buy beer also buy chips.

We may want to make a distinction among customers and toward this, estimate *P(Y | X , D)* where *D* is the set of customer attributes, for example, gender, age, marital status, and so on, assuming that we have access to this information

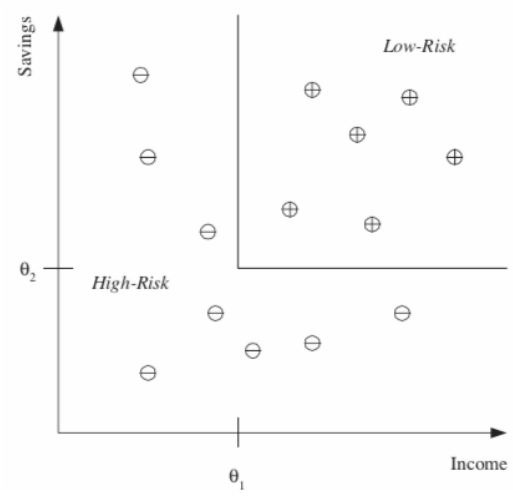
1.4 Classification

- **Explain the 2 types of Supervised learning problems (classification and regression)**
- **Explain Classification problem with examples**
- **Differentiate between binary and multi class classification**
- **Explain pattern recognition technique and its applications**
- **What do you mean by Outlier detection?**
- **What do you mean by discriminant function in case of classification?**

In machine learning, *classification* is the problem of identifying to which set of categories a new observation belongs to, on the basis of a training set of data containing observations (or instances) whose category membership is known.

Consider the following example - It is important for the bank to be able to predict in advance the risk associated with a loan, which is the probability that the customer will default and not pay the whole amount back. In *credit scoring* the bank calculates the risk given the amount of credit and the information about the customer. The information about the customer includes data we have access to and is relevant in calculating his or her financial capacity—namely, income, savings, collaterals, profession, age, past financial history, and so forth. The bank has a record of past loans containing such customer data and whether the loan was paid back or not. From this data of particular applications, the aim is to infer a general rule coding the association between a customer's attributes and his risk. That is, the machine learning system fits a model to the past data to be able to calculate the risk for a new application and then decides to accept or refuse it accordingly. (can shorten the example and write)

This is an example of a ***classification*** **problem** where there are **two classes (binary classification)**:- low-risk and high-risk customers. The information about a customer makes up the *input* to the classifier whose task is to assign the input to one of the two classes.



sification rule learned may be of the form

IF income> θ_1 AND savings> θ_2 THEN low-risk ELSE high-risk

for suitable values of θ_1 and θ_2 (see figure). This is an example of a ***discriminant*** - it is a function that separates the examples of different classes. A *discriminant* of a classification problem is a rule or

a function that is used to assign labels to new observations.

Figure shows Example of a training dataset where each circle corresponds to one data instance with input values in the corresponding axes and its sign indicates the class. For simplicity, only two customer attributes, income and savings, are taken as input and the two classes are low-risk (‘+’) and high-risk (‘−’). An example discriminant that separates the two types of examples is also shown

Having a rule like this, the main application is *prediction*: Once we have a rule that fits the past data, if the future is similar to the past, then we can make correct predictions for novel instances. Given a new application with a certain income and savings, we can easily decide whether it is low- risk or high-risk.

The above example is an example for **binary classification** as there are **2 classes** , if there are **more than 2 classes it’s a multi-class classification problem**.

In some cases, instead of making a 0/1 (low-risk/high-risk) type decision, we may want to calculate a probability, namely, $P(Y \mid X)$, where X are the customer attributes and Y is 0 or 1 respectively for low-risk and high-risk. From this perspective, we can see classification as learning an association from X to Y. Then for a given $X = x$, if we have $P(Y = 1|X = x) = 0.8$, we say that the customer has an 80 percent probability of being high-risk, or equivalently a 20 percent probability of being low-risk. We then decide whether to accept or refuse the loan depending on the possible gain and loss.

Other examples of discriminant functions

IF Score1 + Score2 ≥ 60, THEN “Pass” ELSE “Fail”.

IF Score1 ≥ 20 AND Score2 ≥ 40 THEN “Pass” ELSE “Fail”

Pattern recognition is another type of Machine Learning Classification task application. Where the aim is to classify or identify a pattern from an input set. Following are some of pattern recognition examples.

- ***optical character recognition***, which is recognizing character codes from their images. This is an example where there are multiple classes, as many as there are characters we would like to recognize. We may be interested in classifying handwritten digits or alphabets. Differences in handwriting styles make the classification challenging.
- ***face recognition***, the input is an image, the classes are people to be recognized, and the learning program should learn to associate the face images to identities. This problem is more difficult than optical character recognition because there are more classes, input im- age is larger, and a face is three-dimensional and differences in pose and lighting cause significant changes in the image
- ***medical diagnosis***, the inputs are the relevant information we have about the patient and the classes are the illnesses. The inputs contain the patient’s age, gender, past medical history, and current symptoms.
- ***speech recognition***, the input is acoustic and the classes are words that can be uttered. This time the association to be learned is from an acoustic signal to a word of some language. Different people, because of differences in age, gender, or accent, pronounce the same word differently, which makes this task rather difficult. Another difference of speech is that the input is *temporal*; words are uttered in time as a

sequence of speech phonemes and some words are longer than others

- ***Biometrics*** is recognition or authentication of people using their physiological and/or behavioural characteristics that requires an integration of inputs from different modalities. Examples of physiological characteristics are images of the face, fingerprint, iris, and palm; examples of behavioral characteristics are dynamics of signature, voice, gait, and key stroke.

Knowledge Extraction – Learning a rule from data allows *knowledge extraction*. The rule is a simple model that explains the data, and looking at this model we have an explanation about the process underlying the data. For example, once we learn the discriminant separating low-risk and high-risk customers, we have the knowledge of the properties of low-risk customers. We can then use this information to target potential low-risk customers more efficiently, for example, through advertising.

Outlier Detection - Another use of machine learning is *outlier detection*, which is finding the instances that do not obey the rule and are exceptions. In this case, after learning the rule, we are not interested in the rule but the exceptions not covered by the rule, which may imply anomalies requiring attention— for example, fraud.

Compression - Learning also performs *compression* in that by fitting a rule to the data, we get an explanation that is simpler than the data, requiring less memory to store and less computation to process. Once you have the rules of addition, you do not need to remember the sum of every possible pair of numbers.

Some other classification application includes Spam Filtering – where the task is to classify a mail as spam or not based on various attributes, Natural Language processing, machine Translation.

1.5 Regression (*also refer module 3 for detailed description*)

- ➢ **Differentiate between Classification and Regression technique**
- ➢ **Explain regression (linear and polynomial) with example**

In machine learning, a *regression problem* is the problem of predicting the value of a numeric variable based on observed values of the variable. The value of the output variable may be a number, such as an integer or a floating point value. These are often quantities, such as amounts and sizes. The input variables may be discrete or real-valued.

Let us say we want to have a system that can predict the price of a used car. Inputs are the car attributes—brand, year, engine capacity, mileage, and other information—that we believe affect a car's worth. The output is the price of the car. **Such problems where the output is a number are *regression* problems.**

Let X denote the car attributes and Y be the price of the car. Again surveying the past transactions, we can collect a training data and the machine learning program fits a function to this data to learn Y as a function of X. An example is given in figure below where the fitted function is of the form
$\boldsymbol{y = w x + w_0}$, for suitable values of w and w_0.

The approach in machine learning is that we assume a model defined up to a set of parameters:

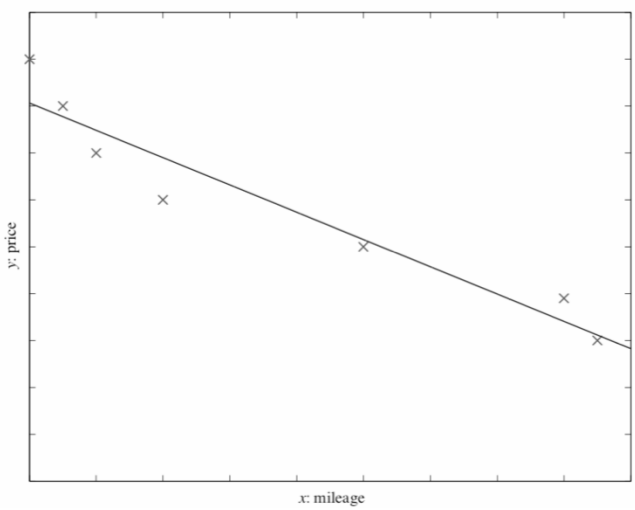
$y = g(x|\theta)$

where $g(\cdot)$ is the model and θ are its parameters.

***Y* is a number in regression and is a class code (e.g., 0/1) in the case of classification.**

***g(·)* is the regression function and in classification, it is the discriminant function separating the instances of different classes**.

The machine learning program optimizes the parameters, θ, such that the approximation error is minimized, that is, our estimates are as close as possible to the correct values given in the training set. (*Refer module 3 to see how the error is calculated)*
For example in figure , the model is linear (ie : $y = w x + w_0$)and w and w_0 are the



parameters optimized for best fit to the training data .

The figure shows A training dataset of used cars and the function fitted. For simplicity, mileage is taken as the only input attribute and a linear model is used.

Now given mileage of a car not in training set, the model can be used to predict its approximate price.

In cases where the linear model is too restrictive, one can use for example a quadratic

$y = w_2x^2 + w_1x + w_0$

or a higher-order polynomial, or any other nonlinear function of the input, this time optimizing its parameters for best fit.

Types of Regression

➢ **Explain/Differentiate the 2 types of Supervised Learning methods (also draw figures for both from above explanations)**

Regression	**Classification**
• Regression is used to predict **continuous values**. • Examples – once a model is trained based on sample data ○ Predicting price of a house given the area, no of bedrooms ○ Predicting amount of rainfall	• Classification is used to predict which class a data point is part of (**discreet value**) • Example – ○ Classifying mail as spam or not spam ○ Identifying a fruit based on size, color, length , diameter etc

given temperature, humidity etc	○ Identifying if a tumor is begin or malignant

Both regression and classification are *supervised learning* problems where there is an input, X, an output, Y, and the task is to learn the mapping from the input to the output.

The approach in machine learning is that we assume a model defined up to a set of parameters:

$y = g(x|\theta)$, where $g(\cdot)$ is the model and θ are its parameters.
***Y* is a number in regression and is a class code (e.g., 0/1) in the case of classification.**

***g(·)* is the regression function and in classification, it is the discriminant function separating the instances of different classes**.

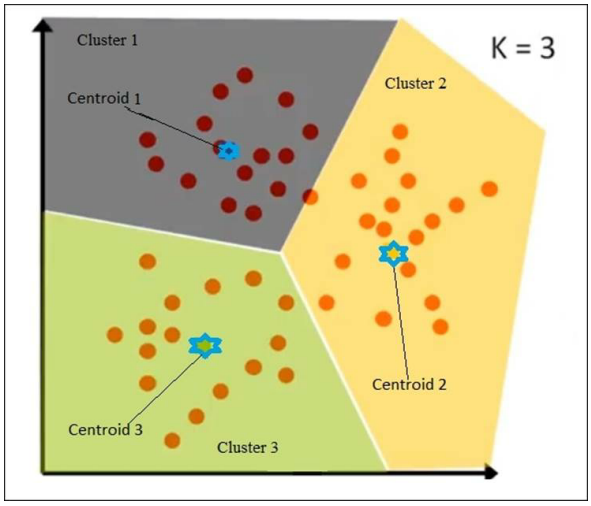
1.6 Unsupervised Learning

- ➢ **Compare Supervised and Unsupervised learning with example**
- ➢ **Explain Unsupervised learning with example**
- ➢ **Explain some applications of Unsupervised learning**

- In supervised learning, the aim is to learn a mapping from the input to an output whose correct values are provided by a supervisor. In unsupervised learning, there is no such supervisor and we only have input data.
- The aim is to find the regularities in the input. There is a structure to the input space such that certain patterns occur more often than others, and we want to see what generally happens and what does not. In statistics, this is called *density estimation*.

Example - Clustering (refer module 6 for detailed description)

- Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.
- Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:
  - ○ Recommendation engines
  - ○ Market segmentation
  - ○ Social network analysis
  - ○ Search result grouping
  - ○ Medical imaging
  - ○ Image segmentation
  - ○ Anomaly detection



Clustering for Image Compression

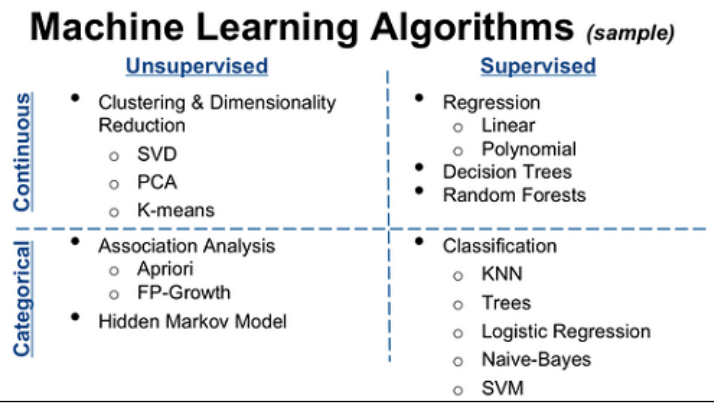
In this case, the input instances are image pixels represented as RGB values. A clustering program groups pixels with similar colors in the same group, and such groups correspond to the colors occurring frequently in the image. If in an image, there are only shades of a small number of colors, and if we code those belonging to the same group with one color, for example, their average, then the image is quantized. Let us say the pixels are 24 bits to represent 16 million colors, but if there are shades of only 64 main colors, for each pixel we need 6 bits instead of 24. For example, if the scene has various shades of blue in different parts of the image, and if we use the same average blue for all of them, we lose the details in the image but gain space in storage and transmission.

Clustering for Document Clustering

In *document clustering*, the aim is to group similar documents. For example, news reports can be subdivided as those related to politics, sports, fashion, arts, and so on. Commonly, a document is represented as a *bag of words*, that is, we predefine a lexicon of N words and each document is an N-dimensional binary vector whose element i is 1 if word i appears in the document; suffixes "–s" and "–ing" are removed to avoid duplicates and words such as "of," "and," and so forth, which are not informative, are not used. Documents are then grouped depending on the number of shared words.

Supervised Learning	**Unsupervised Learning**
1. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output. Y = f(X) The goal is to approximate the mapping function so well that when you have	1. Unsupervised learning is where you only have input data (X) and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

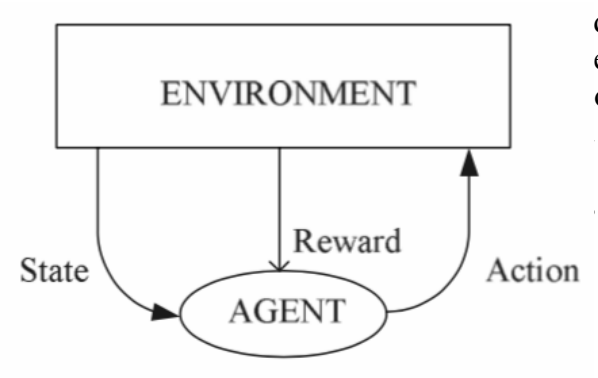
new input data (x) that you can predict the output variables (Y) for that data.	
2. It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.	2. These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data.
3. Supervised learning problems can be further grouped into regression and classification problems.	3. Unsupervised learning problems can be further grouped into clustering and association problems.
4. Explain regression/classification technique as example with fig	4. Explain Clustering technique with example fig



1.7 Reinforcement Learning

➢ **Explain Reinforcement learning with an example, how is it different from Supervised and Unsupervised learning.**

In some applications, the output of the system is a sequence of *actions*. In such a case, a single action is not important; what is important is the *policy* that is the sequence of correct actions to reach the goal.



decision maker, called the ***agent*,** that is placed ent is in a certain ***state*** .The decision maker has osen and taken, the state changes. The solution we get feedback, in the form of a ***reward***. The actions to solve a problem where "best" is the maximum cumulative reward. Such is the

The mathematical framework for defining a solution in reinforcement learning scenario is called **Markov Decision Process**. This can be designed as:

- Set of states, S
- Set of actions, A
- Reward function, R
- Policy, π
- Value, V

We have to take an action (A) to transition from our start state to our end state (*S*). In return getting rewards (R) for each action we take. Our actions can lead to a positive reward or negative reward.
The set of actions we took define our policy (π) and the rewards we get in return defines our value (V). Our task here is to maximize our rewards by choosing the correct policy.

A good example is *game playing* where a single move by itself is not that important; it is the sequence of right moves that is good. A move is good if it is part of a good game playing policy.

A robot navigating in an environment in search of a goal location is another application area of reinforcement learning. At any time, the robot can move in one of a number of directions. After a number of trial runs, it should learn the correct sequence of actions to reach to the goal state from an initial state, doing this as quickly as possible and without hitting any of the obstacles.

Other examples include

- Adaptive Traffic signal optimization
- Adaptive power grid distribution

1.8 Supervised Learning

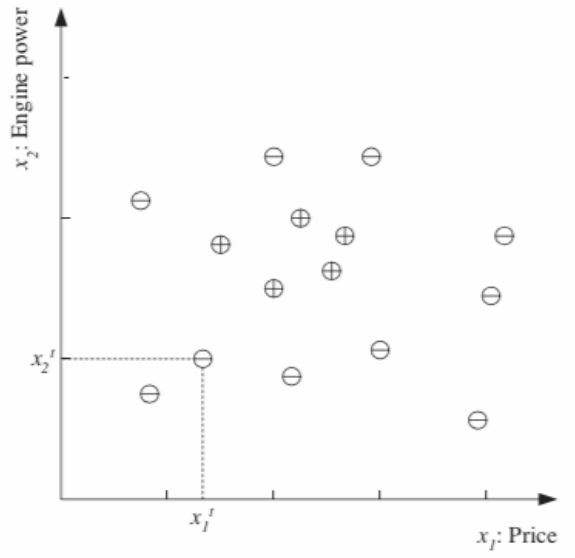
- ➢ **How do we learn a Class from positive and negative examples**
- ➢ **What do you mean by Learning from a class of examples**
- ➢ **Explain Input Representation with an example**
- ➢ **Explain Training set with an example**
- ➢ **What do you mean by hypothesis Class , how can we set an hypothesis**
- ➢ **What do you mean by empirical error, explain with an example**
- ➢ **Explain the cases of Generalization and Specialized Hypothesis with example.**
- ➢ **Explain concept of Version Space with example**

- **Why is it considered best to choose a Margin in between of the Version Space**
- **How does doubts arise when labelling data samples**

Learning from a Class of Examples
Suppose we want to learn the *class*, C, of a "family car." We have a set of examples of cars, and we have a group of people that we survey to whom we show these cars. The people look at the cars and label them; the cars that they believe are family cars are ***positive examples***, and the other cars are ***negative examples***.

Class learning is finding a description that is shared by all positive examples and none of the negative examples. Doing this, we can make a prediction: Given a car that we have not seen before, by checking with the description learned, we will be able to say whether it is a family car or not.

After an analysis experts reach conclusion that among all features a car may have, the features that separate a family car from other cars are the price and engine power. These two attributes are the ***inputs*** to the class recognizer. Note that when we decide on this particular ***input representation***, we are ignoring various other attributes as irrelevant. (other features like seating capacity, mileage etc have been ignored for simplicity)



nily car."(left)

Each data point corresponds to one example car, and the coordinates of the point indicate the price and engine power of that car. '+' denotes a positive example of the class (a family car), and '−' denotes a negative example (not a family car);

We can denote price as the first input attribute x_1 and engine power as the second attribute x_2 (e.g., engine volume in cubic cms).

Thus we represent each car using two numeric values price(x1)and engine power(x2)(**Input Representation**)

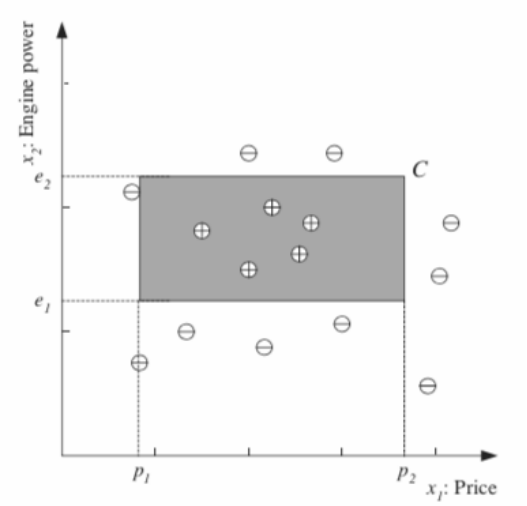
Input representation: we need to decide what attributes (features) to use to describe the input patterns (examples, instances). This implies ignoring other attributes as irrelevant.

Training Set: Each car is represented by such an ordered pair *(x,r)* and the **training set** contains *N* such examples

$\mathbf{X} = \{ \boldsymbol{x}^t, \boldsymbol{r}^t \}_{t=1}^{N}$, where *t* indexes different examples in the set.

And **Class Label** (r) denoted by

Example of a hypothesis class- The class of family car is a rectangle in the price-engine



power space

On plotting examples in 2D we see that for a car to be a family car (positive example) , price and engine power of the car should be in a certain range

$(p_1 \leq \text{price} \leq p_2)$ AND $(e_1 \leq \text{engine power} \leq e_2)$

for suitable values of p_1, p_2, e_1, and e_2 we assume Class C to be a rectangle in the price-engine power space

Hypothesis class H: the *hypothesis class* from which we believe class C (contains set of positive example, family cars) is drawn, namely, the set of rectangles. The learning algorithm then finds the particular *hypothesis*, $h \in H$, to approximate C as closely as possible.

Though the expert defines this hypothesis class(set of rectangles), the values of the parameters are not known; that is, though we choose H (to be a rectangle) , we do not know which particular $h \in H$ is equal, or closest, to C (ie range value of the rectangle, ie 4 parameters p_1, p_2, e_1, and e_2)

The aim is to find $h \in H$ that is as similar as possible to C. Let us say the hypothesis h makes a prediction for an instance x such that

In real life we do not know $C(x)$, so we cannot evaluate how well $h(x)$ matches $C(x)$. What we have is the training set X, which is a small subset of the set of all possible x.

The ***empirical error*** is the proportion of training instances where *predictions* of h do not match the *required values* given in X. (that is for example when a family car is not identified as a family car by the hypothesis).

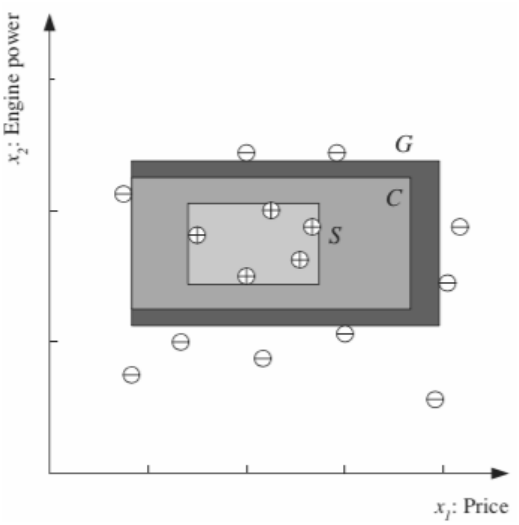
The error of hypothesis *h* given the training set X is

,where $1(a \neq b)$ is 1 if $a \neq b$ and is 0 if $a = b$

In our family car example, the hypothesis class H is the set of all possible rectangles. Each quadruple $(p_1^h, p_2^h, e_1^h, e_2^h)$ defines one hypothesis, h, from H , and we need to choose the best one, or in other words, we need to find the values of these four parameters given the training set, to include all the positive examples and none of the negative examples. There are infinitely many such h for which this is satisfied, namely, for which the error, E, is 0,

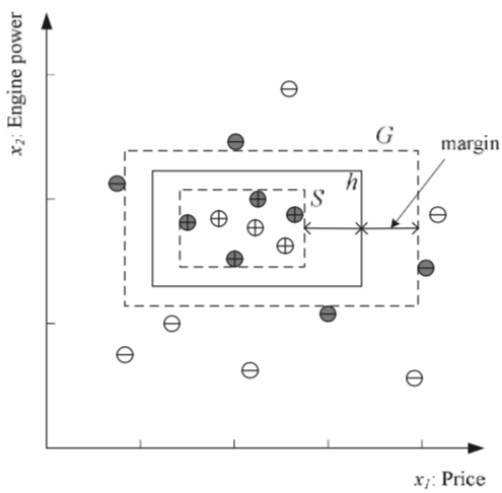
But given a future example somewhere close to the boundary between positive and negative examples, different candidate hypotheses may make different predictions. This is the problem of ***generalization***—that is, how well our hypothesis will correctly classify future examples that are not part of the training set.

Most specific hypothesis, ***S***, that is the tightest rectangle that includes all the positive examples and none of the negative examples This gives us one hypothesis, $h = S$, as our induced class. Note that the actual class C may be larger than S but is never smaller. The ***most general hypothesis*, *G***, is the largest rectangle we can draw that includes all the positive examples and none of the negative examples **Any $h \in H$ between *S* and *G* is a valid hypothesis with no error, said to be *consistent* with the training set, and such *h* make up**



the *version space*.

Depending on X and H , there may be several S_i and G_j which respectively make up the *S*-set and the *G*-set. Every member of the *S*-set is consistent with all the instances, and there are no consistent hypotheses that are more specific. Similarly, every member of the *G*-set is consistent with all the instances, and there are no consistent hypotheses that are more general. These two make up the boundary sets and any hypothesis between them is consistent and is part of the **version space**.



from the version space and use it as our hypothesis, *h*. **halfway between *S* and *G*; this is to increase the** ı the boundary and the instances closest to it. For our *h* with the maximum margin, we should use an error s whether an instance is on the correct side of the . Instead of *h(x)* that returns 0/1, we need to have a carries a measure of the distance to the boundary and ıich uses it, different from the one that checks for

If an instance falls between S and G we consider it to be a **doubt**, ie we cannot label it with certainty.

Thus we can summarize, a model for learning consist of

To find **Class** C, say a "family car"

- **Prediction**: Is car *x* a family car?
- **Knowledge extraction**: What do people expect from a family car?

- **Output**: **Training Set**
  Positive (+) and negative (–) examples of family cars
- **Input representation**:
  x_1: price, x_2 : engine power
- **Hypothesis h** with the largest margin(best separation) in Version Space ie which has the least error.

1.9 Vapnik-Chervonenkis (VC) Dimension

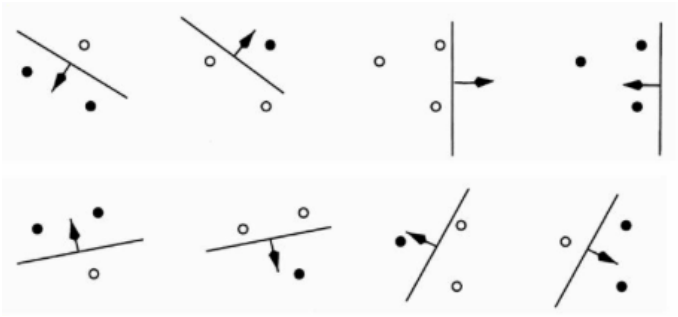
- ➢ **Explain the concept of VC dimensions with example**
- ➢ **When is an hypothesis said to shatter N points**
- ➢ **How can we determine the VC dimension of a Hypothesis**
- ➢ **How can we measure the capacity of a Hypothesis**
- ➢ **Show that the VC dimension of hypothesis being rectangle is four and that of a line is three.**
- ➢ **Justify, can a the VC dimension of a rectangle class be greater than four.**

If we have a dataset containing *N* points. These *N* points can be labeled in 2^N ways as positive and negative. Therefore, 2^N different learning problems can be defined by *N* data points.

If for any of these problems, we can find a hypothesis $h \in H$ that separates the positive examples from the negative, then we say H ***shatters*** *N* points.

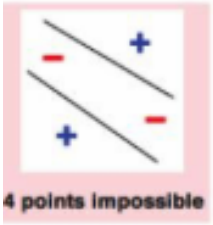
That is, any learning problem definable by *N* examples can be learned with no error by a

hypothesis drawn from H . **The maximum number of points that can be shattered by H is called the *Vapnik-Chervonenkis* (VC) *dimension* of H, is denoted as *VC(*H*)*, and measures the *capacity* of H**

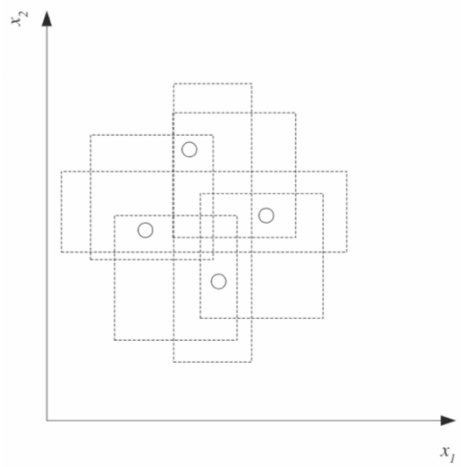


ays). Consider black circles as one class

Assuming the hypothesis to be a separating line, it can separate the point combinations. Thus we can say the hypothesis shatters 3 points and the VC dimension of the hypothesis is 3.



ble to shatter 4 points using a single separation line as Hypothesis. Thus we dimension of the Hypothesis is 3, ie: the maximum points that can be



r four points in two dimensions. Then *VC (H), when H is rectangles in two dimensions, is four*. In calculating the find four points that can be shattered.

It is not necessary that we be able to shatter *any* four points in two dimensions. For example, four points placed on a line cannot be shattered by rectangles, also 3 points placed on a straight line cannot be shattered by a separating line

However we cannot place five points in two dimensions *anywhere* such that a rectangle can separate the positive and negative examples for all possible labelings.

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*.**

In real life, the world is smoothly changing, instances close by most of the time have the same labels, and we need not worry about *all possible labelings*. There are a lot of datasets containing many more data points than four that are learnable by our hypothesis class. So even hypothesis classes with small VC dimensions are applicable and are preferred over those with large VC dimensions

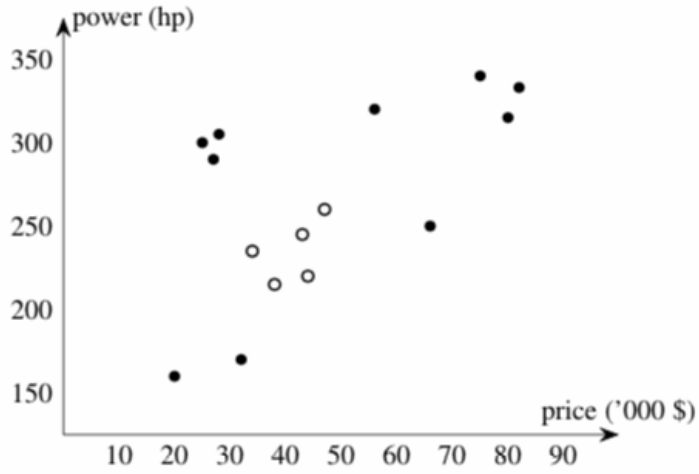
Sample problem

Consider the problem of assigning the label “family car” (indicated by “1”) or “not family car” (indicated by “0”) to cars. Given the following examples for the problem and assuming that the hypothesis space is as IF (p1 < price < p2) AND (e1 < power <

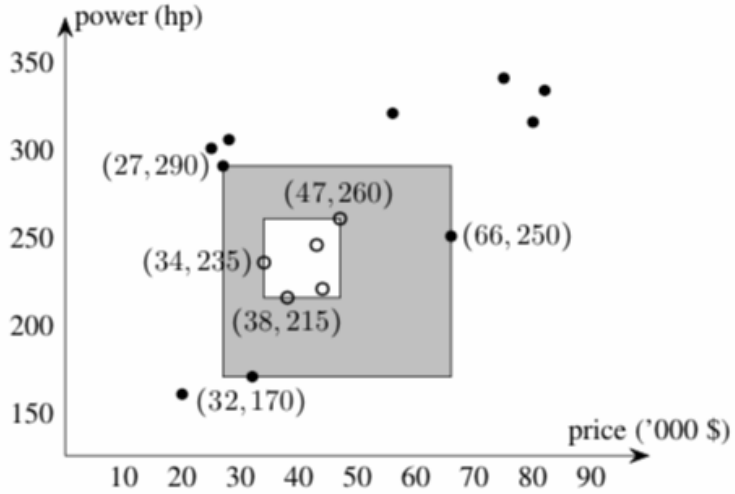
e2) THEN “1” ELSE ”0” . Find the the version space for the problem.

a scatter plot of the given data is drawn . In the figure, the data with class label “1” (family car) is shown as hollow circles and the data with class labels “0” (not family car) are shown as solid dots.

A hypothesis as given by IF (p1 < price < p2) AND (e1 < power < e2) THEN “1” ELSE ”0” with specific values for the parameters p_1, p_2, e_1 and e_2 specifies an axis-aligned rectangle as shown in Figure. So the hypothesis space for the problem can be thought as the set of axis-aligned rectangles in the price-power plane.



Scatter plot of price-power data (hollow circles indicate positive examples and solid dots indicate negative examples)



The version space consists of hypotheses corresponding to axis-aligned rectangles contained in the shaded region

The version space consists of all hypotheses specified by axis-aligned rectangles contained in the shaded region in Figure.

The inner rectangle is defined by

(34 < price < 47) AND (215 < power < 260) and the

outer rectangle is defined by

(27 < price < 66) AND (170 < power < 290).