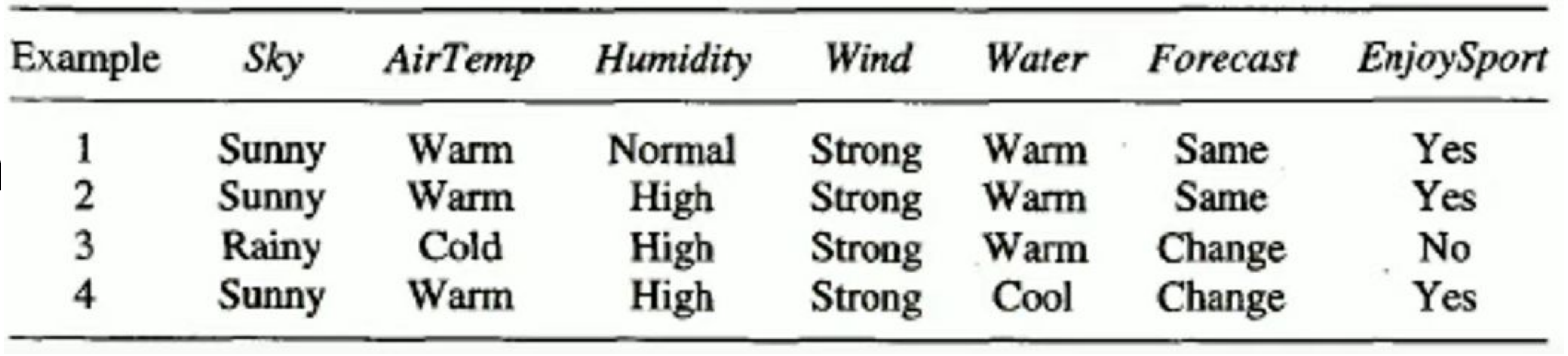
Concept Learning

- Inducing general functions from specific training examples is a main issue of machine learning.
- • Concept Learning: Acquiring the definition of a general category from given sample positive and negative training examples of the category.
- • Concept Learning can seen as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples.
- • The hypothesis space has a general-to-specific ordering of hypotheses, and the search can be efficiently organized by taking advantage of a naturally occurring structure over the hypothesis space.

Con



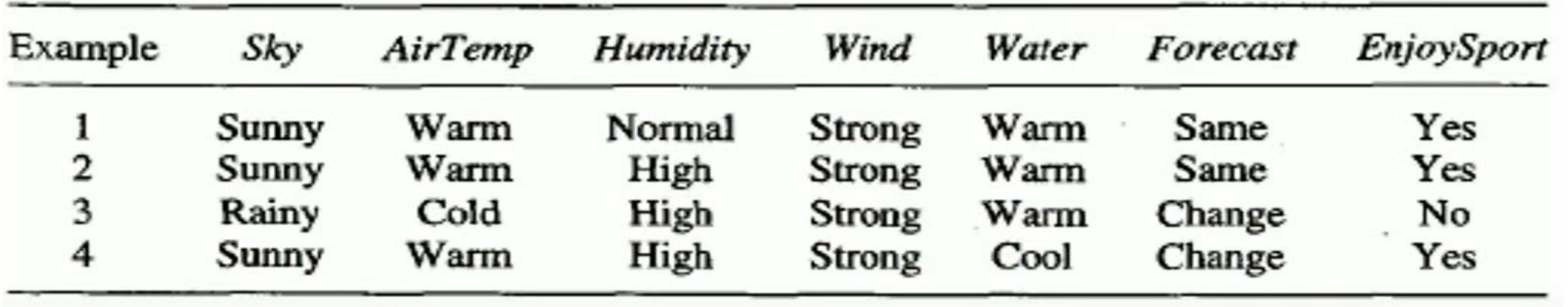
Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- For each attribute, the hypothesis will either indicate by a "?' that any value is acceptable for this attribute,
- specify a single required value (e.g., Warm) for the attribute, or indicate by a " ϕ " that no value is acceptable.
- The most general hypothesis-that every day is a positive example-is represented by (?, ?, ?, ?, ?, ?)
- and the most specific possible hypothesis-that no day is a positive example is represented by (ϕ,ϕ,ϕ,ϕ,ϕ,ϕ)

Find S Algorithm: Finding a Maximally specific Hypotheses

1. Initialize h to the most specific hypothesis in H
   - $h_0 = < \phi,\phi,\phi,\phi,\phi,\phi >$
2. For each positive training instance x
   - For each attribute
   - If the constraint is satisfied by x
   - Then do nothing
   - Else replace attribute in h by the next most general constraint that is satisfied by x

x_1= <Sunny,Warm,Normal,Strong,Warm,Same>
h_1= <Sunny,Warm,Normal,Strong,Warm,Same>



Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- Iteration 2

  x_2= <Sunny,Warm,High,Strong,Warm,Same>

  h_2= <Sunny,Warm,?,Strong,Warm,Same>

- Iteration 3: its a negative case therefore previous hypothesis is kept as it is

  h_3= <Sunny,Warm,?,Strong,Warm,Same>

- Iteration 4: Next outcome is positive, so we consider it for hypothesis generation

  x_4= <Sunny,Warm,High,Strong,Cool,Change>

  h_4= <Sunny,Warm,?,Strong,?,?>

h_4 is the maximally specific hypothesis.

Candidate Elimination Algorithm

- 1. Firstly define the most specific and most generic boundries,.
- 2. If the first outcome is positive, we go to the most generic boundary and check the consistency. If it is consistent we retain it otherwise we write the next general hypothesis. Then we go to the specific boundary.
- 3. If the first outcome is negative, we go to the most specific boundary and check the consistency. If it is consistent we retain it otherwise we write the next specific hypothesis. After that we go to the general boundary.
- 4. Eliminate the hypotheses those are not consistent with the given data.
- 4. After applying above step for each outcome, we will get the most specific and most generic boundries.

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- $S_0 =$ < $\phi,\phi,\phi,\phi,\phi,\phi$ > as ϕ is not matching with any attribute, So S1 will be generated from attributes.
- $S_1 =$ <Sunny,Warm,Normal,Strong,Warm,Same>
- $S_2 =$ <Sunny,Warm, ? ,Strong,Warm,Same>

S_2: S_3: ⟨Sunny,Warm, ?, Strong, Warm, Same⟩

S_4 ⟨Sunny, Warm, ?, Strong, ?, ?⟩

- G_0 = < ?,?,?,?,?,? > As first outcome is +ve So we go with generic boundary. Also as ? Will match with all, therefore next G1 will be same. After that go to specific boundary.
- G_1 = < ?,?,?,?,?,? > As the next outcome is also+ve So we go with generic boundary. Also as ? Will match with all, therefore next G2 will be same. Then go to the specific boundary.

G_3: ⟨Sunny,?,?,?,?,?⟩ ⟨?,Warm,?,?,?,?⟩ ⟨?,?,Normal,?,?,?⟩ ⟨?, ?,?,?,Cool,?⟩ ⟨?,?,?,?,?,Same⟩

G_4: ⟨Sunny, ?, ?, ?, ?, ?⟩ ⟨?, Warm, ?, ?, ?, ?⟩

S ⟨*Sunny, Warm, ?, Strong, ?, ?*⟩

⟨*Sunny, ?, ?, Strong, ?, ?*⟩ ⟨*Sunny, Warm, ?, ?, ?, ?*⟩ ⟨*?, Warm, ?, Strong, ?, ?*⟩

G ⟨*Sunny, ?, ?, ?, ?, ?*⟩ ⟨*?, Warm, ?, ?, ?, ?*⟩

Bayes Theorem

- In machine learning we are often interested in determining the best hypothesis from some space H, given the observed training data D. One way to specify what we mean by the best hypothesis is to say that we demand the most probable hypothesis, given the data D plus any initial knowledge about the prior probabilities of the various hypotheses in H.

- Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.

- Bayes theorem is the cornerstone of Bayesian learning methods because it provides a way to calculate the posterior probability P(h|D), from the prior probability P(h), together with P(D) and P(D|h).

Bayes theorem:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- P(h) - the initial probability that hypothesis h holds, before we have observed the training data.
- P(D) - the prior probability that training data D will be observed (i.e., the probability of D given no knowledge about which hypothesis holds).
- P(D|h) - the probability of observing data D given some world in which hypothesis h holds.
- P (h|D) - the posterior probability of h, because it reflects our confidence that h holds after we have seen the training data D.
- **MAP-** In many learning scenarios, the learner considers some set of candidate hypotheses H and is interested in finding the most probable hypothesis h ϵ H given the observed data D. Any such maximally probable hypothesis is called a maximum a posteriori (MAP) hypothesis.
- We can determine the MAP hypotheses by using Bayes theorem to calculate the posterior probability of each candidate hypothesis.

$$\begin{aligned} h_{MAP} &\equiv \underset{h\in H}{\operatorname{argmax}}\, P(h|D) \\ &= \underset{h\in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)} \\ &= \underset{h\in H}{\operatorname{argmax}}\, P(D|h)P(h) \end{aligned}$$

Example- Does Patient have Cancer or not

- The available data is from a particular laboratory test with two possible outcomes: +(positive) and -(negative).
- We have prior knowledge that over the entire population of people only .008 have this disease.
- Furthermore, the lab test is only an imperfect indicator of the disease. The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present. In other cases, the test returns the opposite result.
- The above situation can be summarized by the following probabilities:

$$P(cancer) = 0.008 \qquad P(\neg cancer) = 0.992$$

$$P(+|cancer) = 0.98 \qquad P(-|cancer) = 0.02$$

$$P(+|\neg cancer) = 0.03 \qquad P(-|\neg cancer) = 0.97$$

- Observe a new patient for whom the lab test returns a positive result and diagnose the patient is having cancer or not?

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$$P(cancer|+) = P(+|cancer) * P(cancer) = 0.98 * 0.008 = 0.0078$$

$$P(\neg cancer|+) = P(+|\neg cancer) * P(\neg cancer) = 0.03 * 0.992 = 0.0298$$

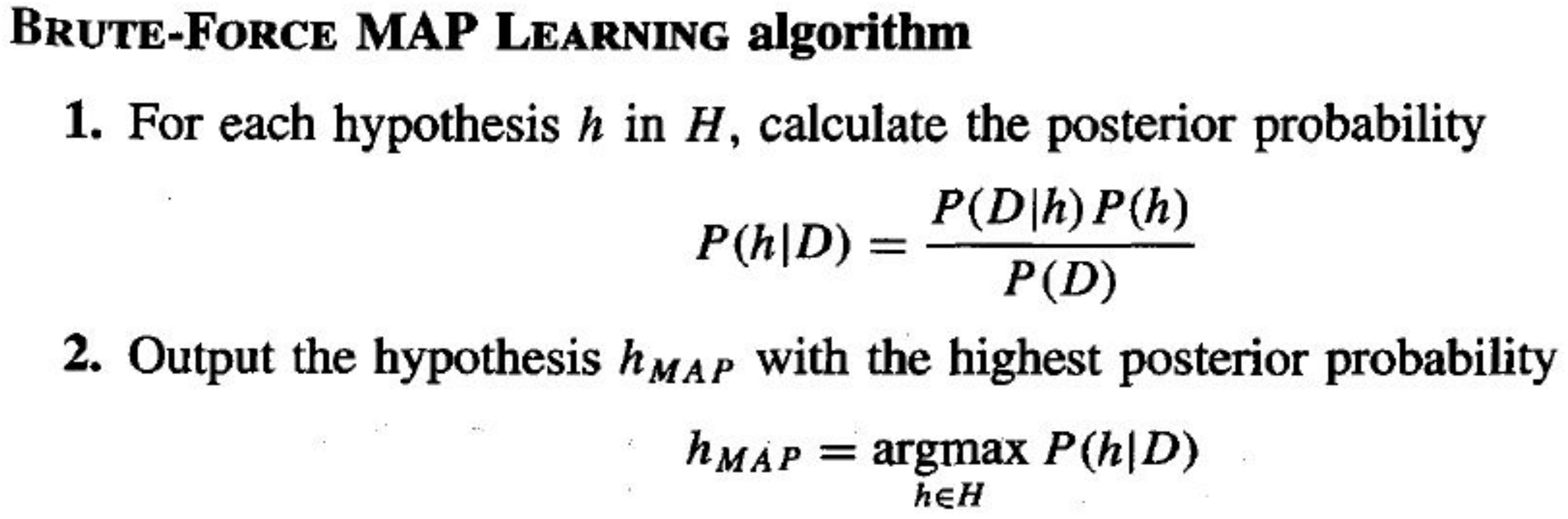
- Observe a new patient for whom the lab test returns a negative result and diagnose the patient is having cancer or not using Bayes rule?

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$$P(cancer|-) = P(-|cancer) * P(cancer) = 0.02 * 0.008 = 0.00016$$

$$P(\urcorner cancer|-) = P(-|\urcorner cancer) * P(\urcorner cancer) = 0.97 * 0.992 = 0.96224$$

Base Theorem and Concept Learning
Brute-Force based concept learning



BRUTE-FORCE MAP LEARNING algorithm

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)\,P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\, P(h|D)$$

We may choose the probability distributions P(h) and P(D1h) in any way we wish, to describe our prior knowledge about the learning task. Here let us choose them to be consistent with the following assumptions:

1. The training data D is noise free (i.e., di = c(xi)).

2. The target concept c is contained in the hypothesis space H

3. We have no a priori reason to believe that any hypothesis is more probable

than any other.

Given these assumptions, what values should we specify for P(h)? Given no prior knowledge that one hypothesis is more likely than another, it is reasonable to assign the same prior probability to every hypothesis h in H.

$$P(h) = \frac{1}{|H|} \quad \text{for all } h \text{ in } H$$

- The probability of observing the target values D = (dl . . .dm) for the fixed set of instances (XI . . . x,), given a world in which hypothesis h holds (i.e., given a world in which h is the correct description of the target concept c). Since we assume noise-free training data, the probability of observing classification di given h is just 1 if di = h(xi) and 0 if di = h(xi).

$$P(D|h) = \begin{cases} 1 \text{ if } d_i = h(x_i) \text{ for all } d_i \text{ in } D \\ 0 \text{ otherwise} \end{cases}$$

In other words, the probability of data D given hypothesis h is 1 if D is consistent with h, and 0 otherwise

Let us consider the first step of this algorithm, which uses Bayes theorem to compute the posterior probability ***P(h1D)*** of each hypothesis ***h*** given the observed training data ***D.***

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

First consider the case where *h* is inconsistent with the training data *D. P(D)h)* to be *0* when *h* is inconsistent with *D,* we have

$$P(h|D) = \frac{0 \cdot P(h)}{P(D)} = 0 \text{ if } h \text{ is inconsistent with } D$$

The posterior probability of a hypothesis inconsistent with *D* is zero. Now consider the case where *h* is consistent with *D.* *P(Dlh)* to be 1 when *h* is consistent with *D,* we have

$$\begin{aligned} P(h|D) &= \frac{1 \cdot \frac{1}{|H|}}{P(D)} \\ &= \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}} \\ &= \frac{1}{|VS_{H,D}|} \text{ if } h \text{ is consistent with } D \end{aligned}$$

where *VSH,*is~ th e subset of hypotheses from H that are consistent with *D* i.e., *VSH,*is t he version space of H with respect to *D*

Bayes Optimal Classifier

- Using Bayes theorem the most probable hypothesis can be found. Now for the new instance the most probable classification is to be found. This can easily be done using MAP, maximum a posteriori.
- To get better result we use Bayes optimal classifier.
- Consider a hypothesis space containing three hypotheses, hl, h2, and h3. Suppose that the posterior probabilities of these hypotheses given the training data are .4, **.3,** and **.3** respectively.
- Suppose a new instance x is encountered, which is classified positive by ***hl,*** but negative by *h2* and h3. Taking all hypotheses into account, the probability that x is positive is .4 (the probability associated with *hi)*
- the probability that it is negative is therefore .6. The most probable classification (negative) in this case is different from the classification generated by the MAP hypothesis.
- **P(x is +) = 0.4**
- **P(x is -) = 0.3+0.3=0.6**

- In general, the most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities. P(Vj|D) is the probability that correctly classify the new instance.

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- **Base Optimal Clas**

$$\underset{v_j \in V}{\operatorname{argmax}} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

The set of possible classifications of the new instance is $V = \{\oplus, \ominus\}$

$$P(h_1|D) = .4,\ P(\ominus|h_1) = 0,\ P(\oplus|h_1) = 1$$

$$P(h_2|D) = .3,\ P(\ominus|h_2) = 1,\ P(\oplus|h_2) = 0$$

$$P(h_3|D) = .3,\ P(\ominus|h_3) = 1,\ P(\oplus|h_3) = 0$$

- So for +ve outcome

0.4+0+0=0.4

$$\sum_{h_i \in H} P(\oplus|h_i) P(h_i|D) = .4$$

and for –ve outcome

0+0.3+0.3=0.6 and

$$\sum_{h_i \in H} P(\ominus|h_i) P(h_i|D) = .6$$

Maximum is 0.6

- No other classification method using the same hypothesis space and same prior knowledge can outperform this method on average. This method maximizes the probability that the new instance is classified correctly, given the available data, hypothesis space, and prior probabilities over the hypotheses.

Naïve Bayes Classifier

- One highly practical Bayesian learning method is the naive Bayes learner, often called the ***naive Bayes classifier. Here we don't deal with the hypotheses.***
- The naive Bayes classifier applies to learning tasks where each instance ***x*** is described by a conjunction of attribute values and where the target function f ***(x)*** can take on any value from some finite set V. A set of training examples of the target function is provided, and a new instance is presented, described by the attribute values ***(al, a2.. .a,).*** The learner is asked to predict the target value, or classification, for this new instance.
- The Bayesian approach to classifying the new instance is to assign the most probable target value, VMAPg, iven the attribute values ***(al,a 2 . . .a ,)*** that describe the instance.

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n)$$

We can use Bayes theorem to rewrite this expression as

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

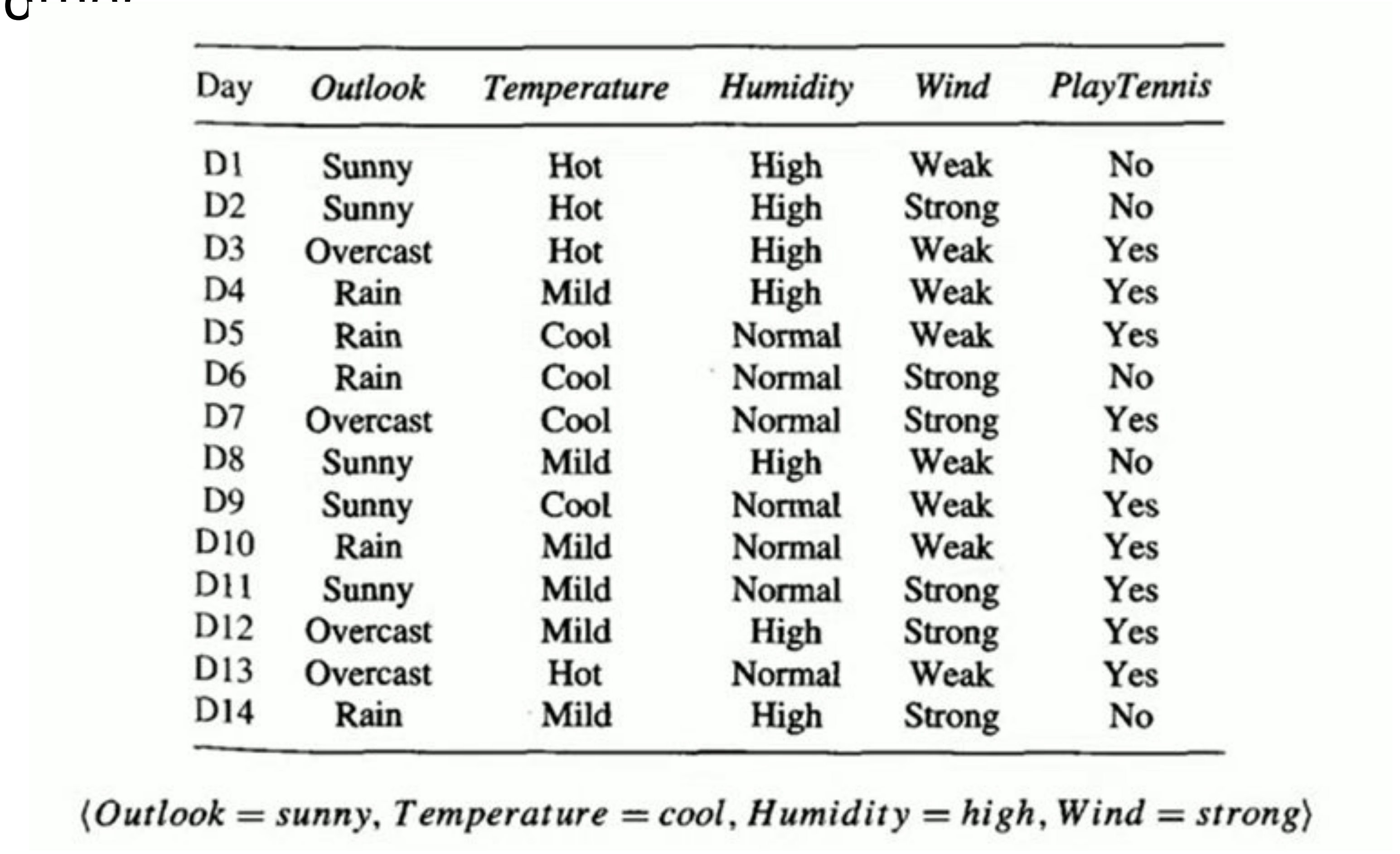
Naive Bayes classifier:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

Naive Bayes learning method involves a learning step in which the various P(vj) and P(ai|vj) terms are estimated, based on their frequencies over the training data. The set of these estimates corresponds to the learned hypothesis. This hypothesis is then used to classify each new instance by applying the rule in Equation. Whenever the naive Bayes assumption of conditional independence is satisfied, this naive Bayes classification VNB is identical to the MAP classification.

One interesting difference between the naive Bayes learning method and other learning methods we have considered is that there is no explicit search through the space of possible hypotheses (in this case, the space of possible hypotheses is the space of possible values that can be assigned to the various P(vj) and P(ailvj) terms). Instead, the hypothesis is formed without searching, simply by counting the frequency of various data combinations within the training examples.

- Naïve Bayes classifier is based on Bayes theorem with am assumption that all the attributes are independent of each other



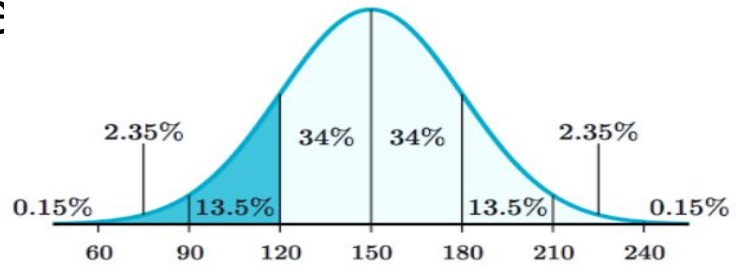
Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

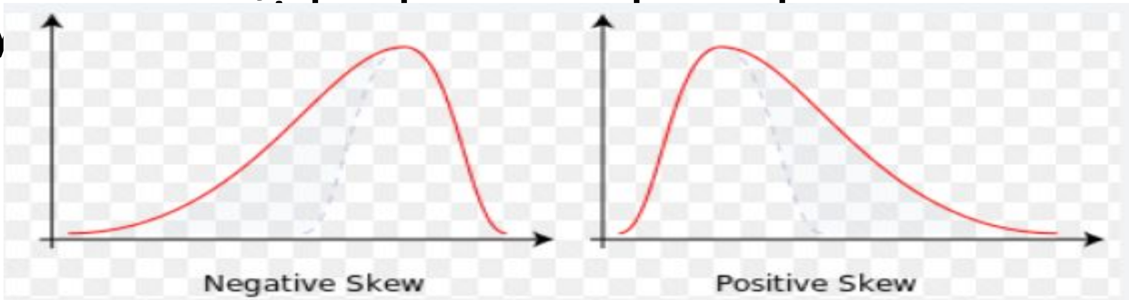
$\langle Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong\rangle$

Consider a data set with 14 rows and with 4 attributes. New instance is also given for which classification is to be done.

EM Algorithm
Expectation and Maximization Algorithm

- F=ma
- F is observed indirectly but m and a can be observed directly.
- Similarly some parameters can directly be measured and some can not be directly measured. These are called unobservable data.
- These unobservable data can be measured by observable data.
- Normal distribution- A probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean and is also known as Gaussian distribution. Like most of the students in class will have average weight and very less will have underweight and overweight. Its of bell shaped curve.
- Skewed distribution-A skewed distribution is neither symmetric nor normal be off more sharply





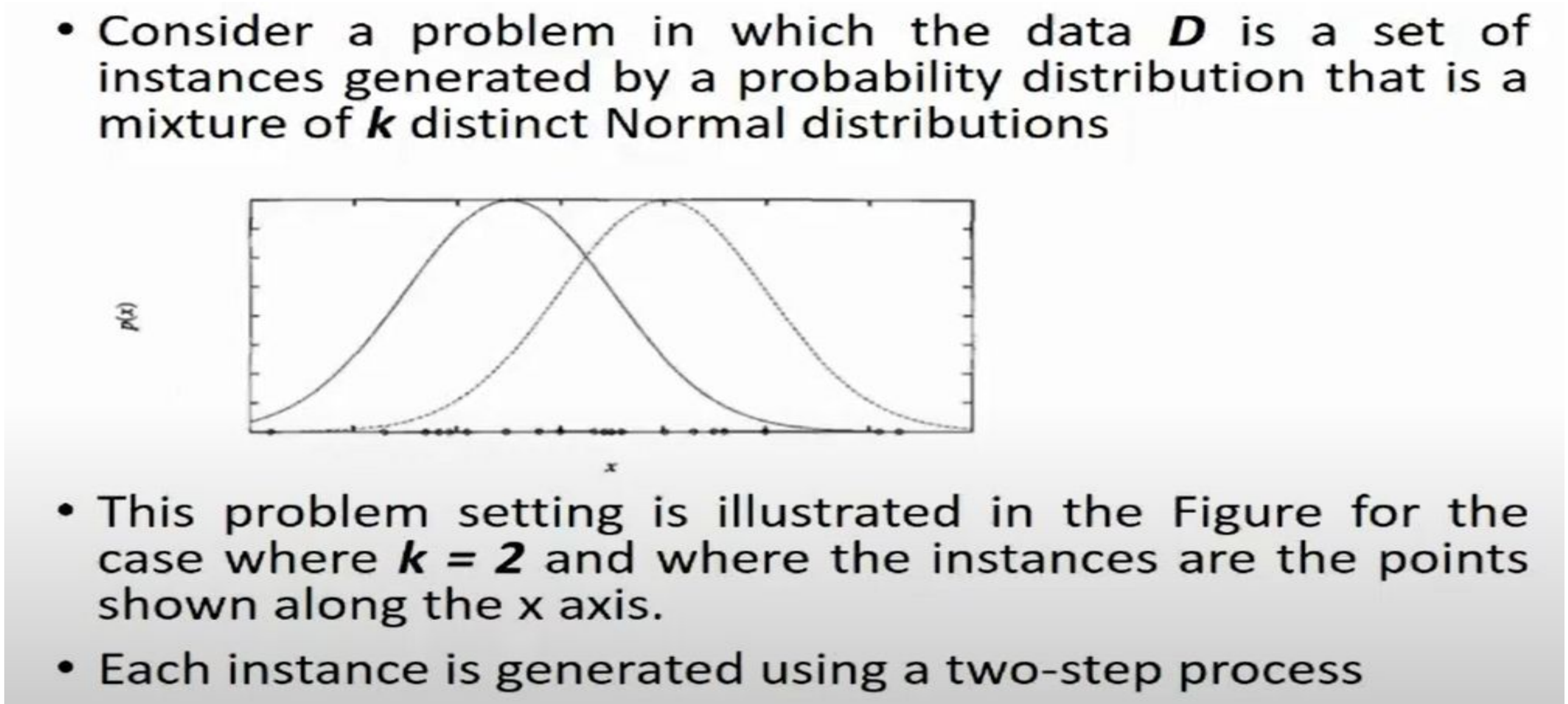
- The learning system is given known values and its trained on those values. Based on learning the distribution will be cleared and now the system can estimate the unknown values also.
- In Expectation and maximization also firstly the values of unobserved data are estimated and then in maximization we maximize their probability of occurrence in data set. If its not maximized then estimate different values and again maximize. Its done iteratively. If the generated hypothesis has maximized probability then algorithm converge.

EM Algorithm
Expectation and Maximization Algorithm

- In the real world applications of machine learning, it is very common that there are many relevant features available for learning but only a small subset of them are observable.
- If some variable is sometimes observed and sometimes not, then we can use the cases for which it has been observed to learn to predict its values when it is not.
- The EM algorithm can be used for the latent variables ie the variables that are not directly observable and are actually inferred from the values of the other observed variables.
- The EM algorithm can be used even for variables whose value is never directly observed, provided the general form of the probability distribution governing these variables is known.
- The EM algorithm has been used to train Bayesian belief networks as well as radial basis function networks.
- This algorithm is the base for many unsupervised clustering algorithms in the field of machine learning.

Estimating mean of k Gaussians

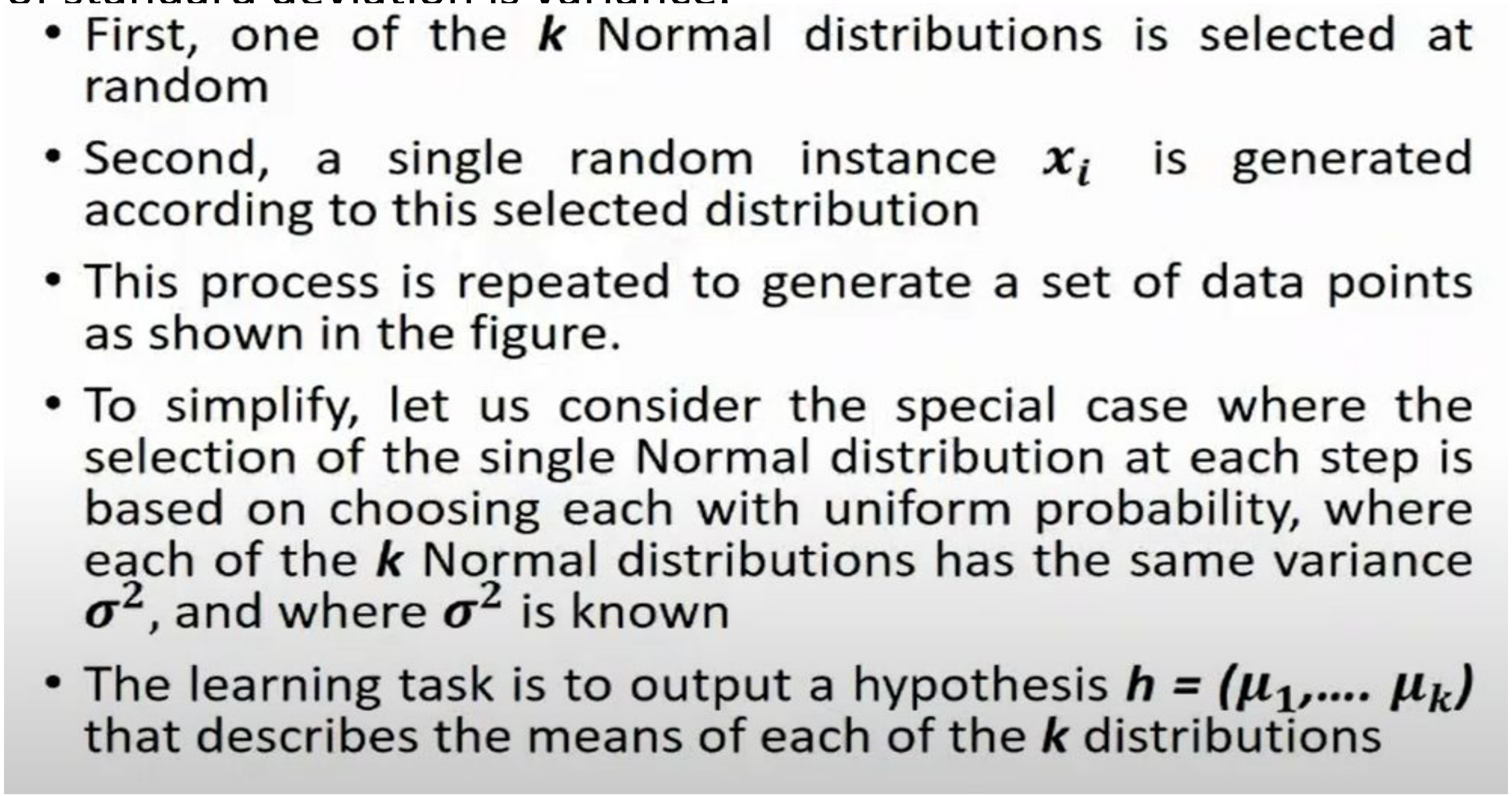
- Training examples are producing two types of normal distributions with different means. Now we don't know which data set belongs to which distribution.



- Consider a problem in which the data ***D*** is a set of instances generated by a probability distribution that is a mixture of ***k*** distinct Normal distributions

- This problem setting is illustrated in the Figure for the case where ***k = 2*** and where the instances are the points shown along the x axis.
- Each instance is generated using a two-step process

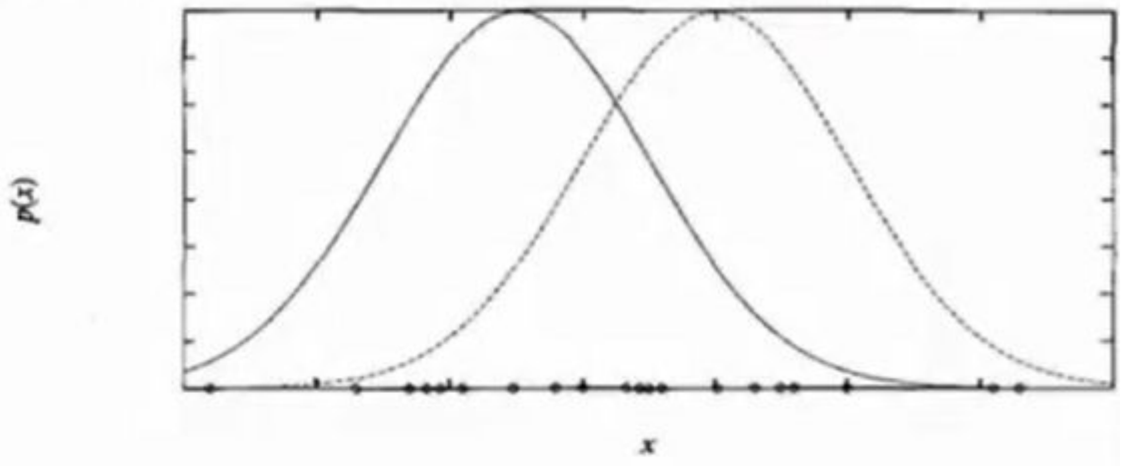
Standard deviation ives the deviation from the mean and square root of standard deviation is variance.



- First, one of the ***k*** Normal distributions is selected at random
- Second, a single random instance $\boldsymbol{x_i}$ is generated according to this selected distribution
- This process is repeated to generate a set of data points as shown in the figure.
- To simplify, let us consider the special case where the selection of the single Normal distribution at each step is based on choosing each with uniform probability, where each of the ***k*** Normal distributions has the same variance $\boldsymbol{\sigma^2}$, and where $\boldsymbol{\sigma^2}$ is known
- The learning task is to output a hypothesis ***h* = (**$\boldsymbol{\mu_1}$**,.... **$\boldsymbol{\mu_k}$**)** that describes the means of each of the ***k*** distributions

- We would like to find a maximum likelihood hypothesis for these means; that is, a hypothesis ***h*** that maximizes ***p(D|h)***
- It is easy to calculate the maximum likelihood hypothesis for the mean of a single Normal distribution given the observed data instances $\boldsymbol{x_1}$, $\boldsymbol{x_2}$,..., $\boldsymbol{x_m}$ drawn from this single distribution
- We already know that the maximum likelihood hypothesis is the one that minimizes the sum of squared errors over the ***m*** training instances
- We can write as $\mu_{ML} = \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{m} (x_i - \mu)^2$

- In the example of this Figure, we can think of the full description of each instance as the triple (x_i, z_{i1}, z_{i2}), where x_i is the observed value of the i^{th} instance and where z_{i1} and z_{i2} indicate which of the two Normal distributions was used to generate the value x_i



- In particular, z_{ij} has the value 1 if x_i was created by the j^{th} Normal distribution and 0 otherwise
- Here x_i is the observed variable in the description of the instance, and z_{i1} and z_{i2} are hidden variables

- If the values of z_{i1} and z_{i2} were observed, we could use Equation

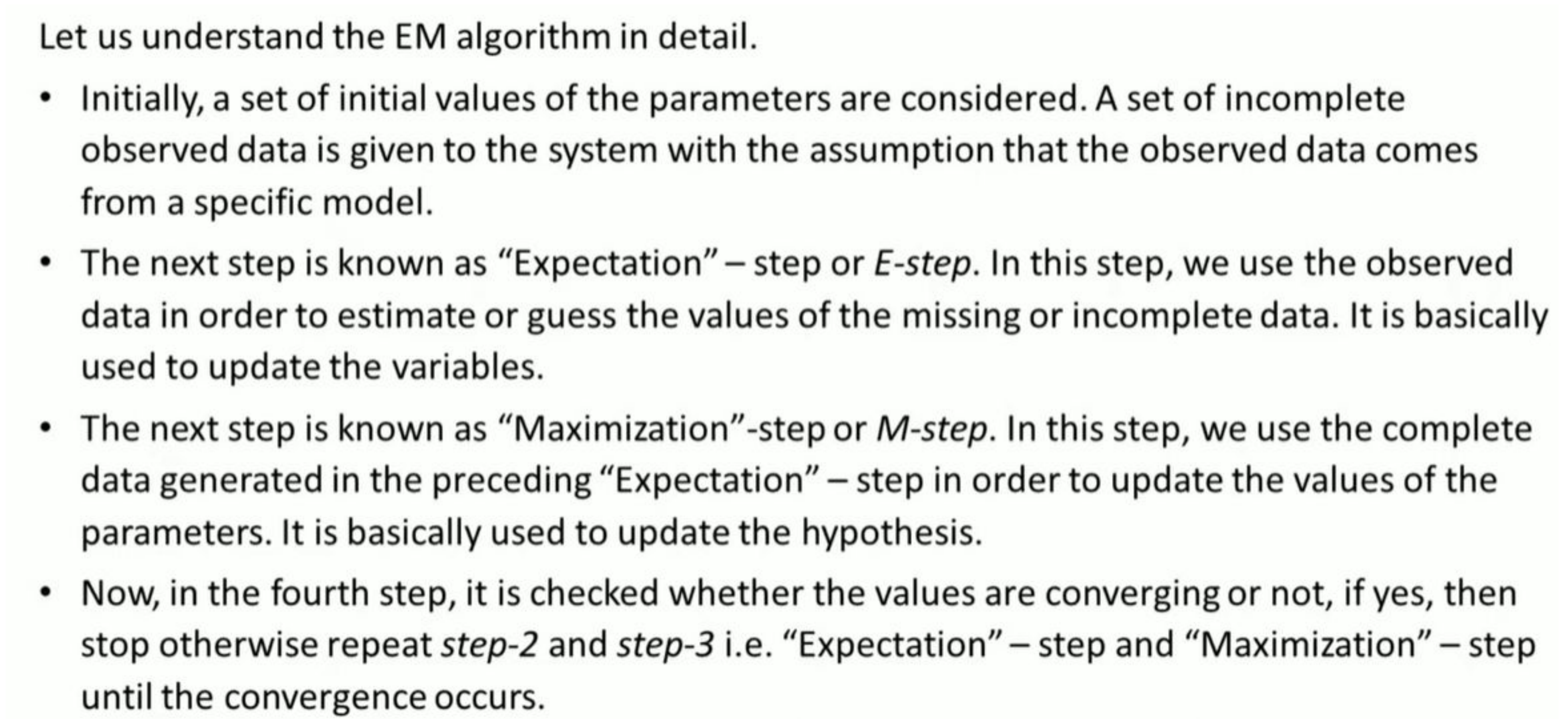
$$\mu_{ML} = \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{m} (x_i - \mu)^2$$

to solve for the means μ_1 and μ_2. Because they are not, we will instead use the EM algorithm

- Applied to the problem of estimating the two means for the Figure, the EM algorithm first initializes the hypothesis to ***h*** = (μ_1, μ_2), where μ_1 and μ_2 are arbitrary initial values.
- It then iteratively re-estimates ***h*** by repeating the following two steps until the procedure converges to a stationary value for ***h***

- Consider a problem in which the data D is a set of instances generated by a probability distribution that is a mixture of k distinct Normal distributions. For the case where k = 2 and where the instances are the points shown along the ***x*** axis. Each instance is generated using a two-step process. First, one of the k Normal distributions is selected at random. Second, a single random instance ***xi*** is generated according to this selected distribution. This process is repeated to generate a set of data points as shown in the figure

STEPS



Let us understand the EM algorithm in detail.

- Initially, a set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model.
- The next step is known as “Expectation” – step or *E-step*. In this step, we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables.
- The next step is known as “Maximization”-step or *M-step*. In this step, we use the complete data generated in the preceding “Expectation” – step in order to update the values of the parameters. It is basically used to update the hypothesis.
- Now, in the fourth step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat *step-2* and *step-3* i.e. “Expectation” – step and “Maximization” – step until the convergence occurs.