Learning

- **Learning process:**
  - Learner (a computer program) processes data ***D*** representing the past experiences and tries to either derive something reasonable about the data seen or to develop some appropriate response to future data
- **Example:**
  - Learner sees a set of patient cases with corresponding disease labels and tries to predict the disease for future patient cases

Types of learning

Three main types:

- **Supervised learning**
  - Learning mapping between inputs x and desired outputs y
  - Teacher gives me y's for the learning purposes
- **Unsupervised learning**
  - Learning relations between data components
- **Reinforcement learning**
  - Learning mapping between inputs x and desired outputs y
  - Critic does not give me y's but instead a signal (reinforcement) of how good my answer was

Supervised learning

Data: $D = \{d_1, d_2, .., d_n\}$ **a set of *n* examples**

$d_i = < \mathbf{x}_i, y_i >$

$\mathbf{x}_i$ is input vector, and *y* is desired output (given by a teacher)

Objective: learn the mapping $f : X \rightarrow Y$

s.t. $y_i \approx f(x_i)$ for all $i = 1, .., n$

Two types of problems:

- **Regression:** Y is **continuous**
  Example: earnings, product orders → company stock price
- **Classification:** Y is **discrete**
  Example: temperature, heart rate → disease

Unsupervised learning

- **Data:** $D = \{d_1, d_2, .., d_n\}$
  $d_i = \mathbf{x}_i$ vector of values
  No target value (output) y to learn

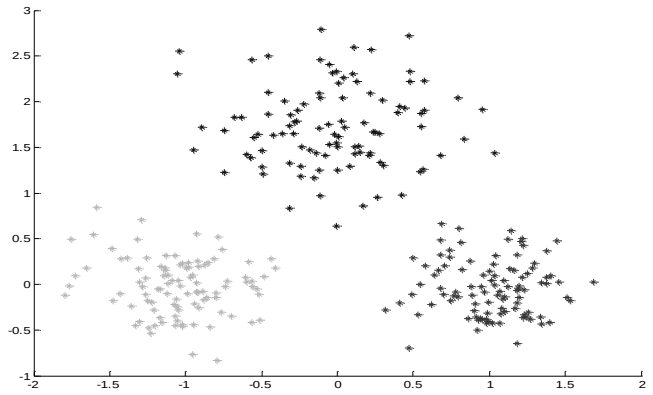
- **Objective:**
  - learn relations between samples, components of samples

Types of problems:

- **Clustering**
  Group together "similar" sample instances, e.g. patient cases
- **Density estimation**
  - Model probabilistically the population of samples

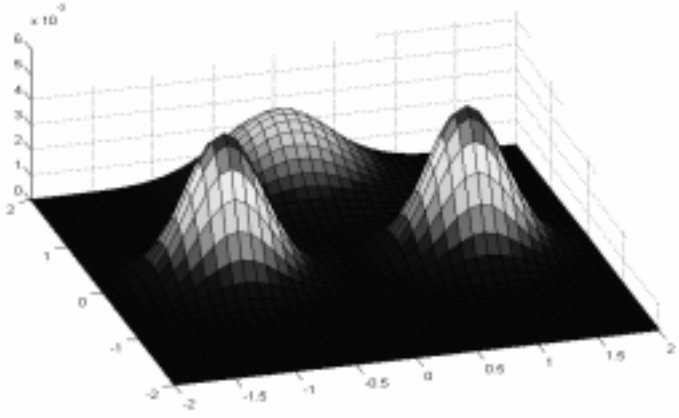
Unsupervised learning. Density estimation

- We want to build the probability model of a population from which we draw samples $d_i = \mathbf{x}_i$



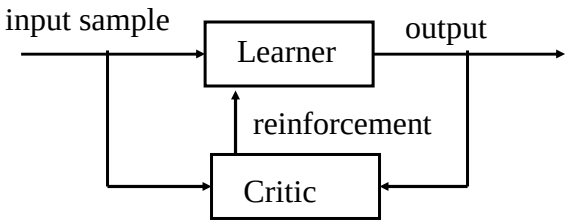
Unsupervised learning. Density estimation

- Mixture of Gaussians – gives a probability distribution of a point in two dimensional space being seen)



Reinforcement learning

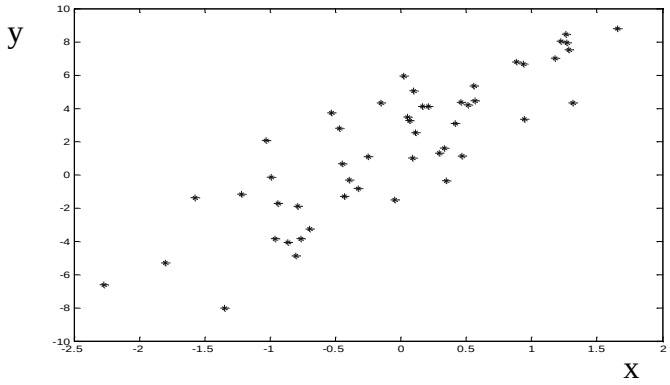
- We want to learn: $f : X \rightarrow Y$
- We see samples of **x** but not *y*
- Instead of *y* we get a feedback (reinforcement) from a **critic** about how good our output was



- The goal is to select output that leads to the best reinforcement

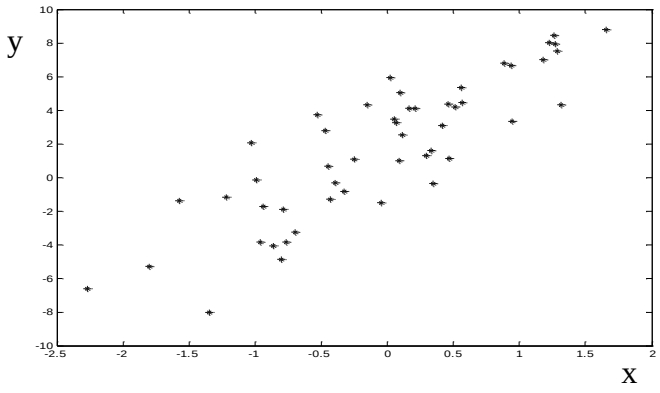
Learning

- Assume we see examples of pairs (**x** , *y*) and want to learn the mapping $f : X \rightarrow Y$ for all possible values of **x**
- We get the data what should we do?



Learning bias

- **Problem:** many possible functions $f : X \to Y$ exists for representing the mapping between x and y
- Which one to choose? Many samples still unseen!



Learning bias

- Problem is easier when we make an assumption about the model, say, $f(x) = ax + b + \varepsilon$

  $\varepsilon = N(0, \sigma)$ - random (normally distributed) noise
- Restriction to the linear model is an example of learning bias