Learning From Data: An Overview

Shao-Lun Huang
2017/09/22 @ TBSI

TBSI
Tsinghua-Berkeley Shenzhen Institute

The Age of Big Data

Machine Learning

- Different types of learning:
  - Supervised learning: given some input (label) and output training data, learn the structure of the machine from the training data.
  - Unsupervised learning: no labels are given in prior, finding hidden structure or pattern from the data their own.
  - Reinforcement learning: the learning machine is presented in an interactive manner to a dynamic environment.

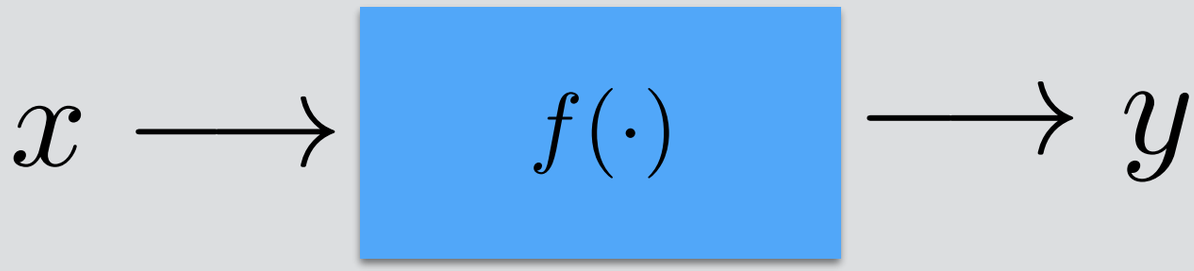
Supervised Learning

- Fundamental elements of supervised learning

$$x \longrightarrow \boxed{f(\cdot)} \longrightarrow y$$

- Input x (can be a vector), output y (the label)
- A machine f generating y from x

Supervised Learning



- Given training data of x and y, the goal is:
  - Inference: knowing the structure of f, find good models to describe f.
  - Prediction: given future data samples of x, predict the corresponding output data y.

Statistical Learning

- Statistical learning assumes the data are i.i.d. generated from some unknown distributions.
- Key elements of statistical learning:
  - Input generating distribution $P(x)$
  - Output generating distribution $P(y|x)$
  - A collection of learning machines $f(x) \in \mathcal{F}$
- Find the best learning machine to predict y from x.

Statistical Learning

- How to measure the performance?
  - The loss function $L(y, f(x))$
  - The risk $R = E[L(y, f(x))]$
  - Empirical risk: given data samples (x_i, y_i)

$$R_{\text{emp}} = \sum_{i=1}^{n} L(y_i, f(x_i))$$

- Find the best $f(x)$ in $\mathcal{F}$ to minimize the empirical risk.
- Many ML algorithms can be viewed as finding the optimal f in a given $\mathcal{F}$ for certain loss function.

Statistical Learning

- Some examples of loss functions:
  - The mean-squared error (MSE)

$$L(y, f(x)) = (y - f(x))^2$$

  - The empirical risk $= \sum_{i=1}^{n} (y - f(x))^2$
  - The indicator function, for label $y = \{0,1\}$

$$L(y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{if } y \neq f(x) \end{cases}$$

The Linear Regression

- In the statistical learning, apply the MSE loss function
- The set of functions $\mathcal{F}$ are linear functions of x.
- Linear regression: let input be $x = (x^{(1)}, \ldots, x^{(k)})$
- Given data samples $(y_i, x_i^{(1)}, \ldots, x_i^{(k)}),\ \text{for}\ i = 1, \ldots, n$
- Find the optimal linear coefficient c_i to minimize the risk

$$R = \sum_{i=1}^{n} \left(y_i - \left(c_1 x_i^{(1)} + \cdots + c_k x_i^{(k)} \right) \right)^2$$

The Linear Regression

- Written in vector form

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(k)} \\ x_2^{(1)} & \cdots & x_2^{(k)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \cdots & x_n^{(k)} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix}$$

- The optimal coefficients minimizing the risk are

$$\mathbf{c}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- Orthogonality: $(\mathbf{y} - \mathbf{X}\mathbf{c}^*)^T \cdot \mathbf{X} = 0$

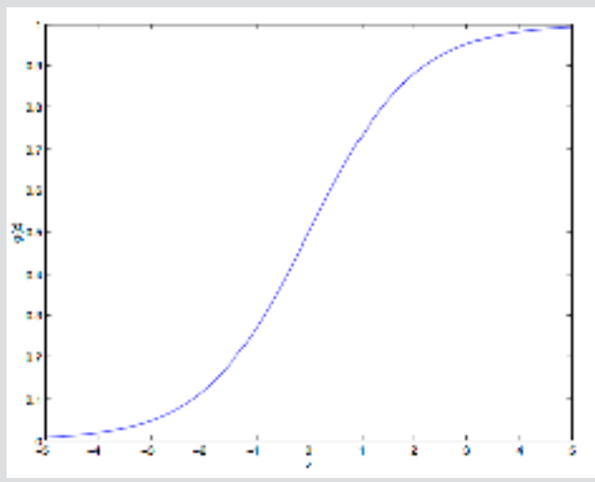
The Linear Regression

- The advantages:
  - Have a very simple close form solution, easy to analyze.
  - Can be computed with a linear time algorithm
- The disadvantages:
  - Can only capture linear features
  - If the hidden structure behind the data is very complicated, the performance is often very bad

Logistic Regression

- In many data classification problems, y is the binary label $\{0,1\}$.
  - Linear regression is not good to approximate the discrete data.
- Instead of using linear functions of data x to predict y, we use the sigmoid function:

$$g(z) = \frac{1}{1 + e^{-z}}$$



- $g(z)$ takes value in $(0,1)$
- Can be viewed as the probability of the event $\{y = 1\}$.

Logistic Regression

- From the statistical learning point of view:
  - The collection of learning machine: $f(x) = \dfrac{1}{1 + e^{-\sum_{j=1}^{k} c_j x^{(j)}}}$
  - The loss function: $y \log h(x) + (1 - y) \log(1 - h(x))$
  - The empirical risk:

$$\sum_{i=1}^{n} y_i \log h(x_i) + (1 - y_i) \log(1 - h(x_i))$$

- The algorithm of finding optimal coefficients:
  - The gradient decent algorithm:

$$c_j \leftarrow c_j + \alpha(y_i - h(x_i))x_i^{(j)}$$

Softmax Regression

- Logistic regression can be generalized to multi-classification $y = \{1, 2, \ldots, M\}$.
- The softmax regression:
  - The collection of learning machine:

$$f(x) = \left[\frac{e^{\sum_{j=1}^{k} c_{j,1} x^{(j)}}}{\sum_{m=1}^{M} e^{\sum_{j=1}^{k} c_{j,m} x^{(j)}}}, \ldots, \frac{e^{\sum_{j=1}^{k} c_{j,M} x^{(j)}}}{\sum_{m=1}^{M} e^{\sum_{j=1}^{k} c_{j,m} x^{(j)}}} \right]$$

  - The empirical risk:

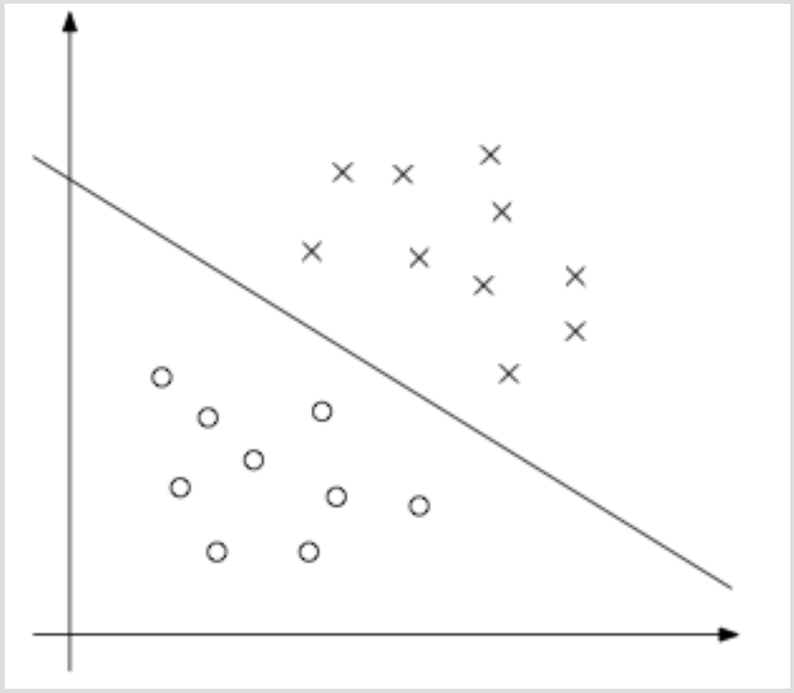
$$\sum_{i=1}^{n} \log \Pi_{m=1}^{M} \left(\frac{e^{\sum_{j=1}^{k} c_{j,1} x_i^{(j)}}}{\sum_{m=1}^{M} e^{\sum_{j=1}^{k} c_{j,m} x_i^{(j)}}} \right)^{\mathbb{I}\{y_i=1\}}$$

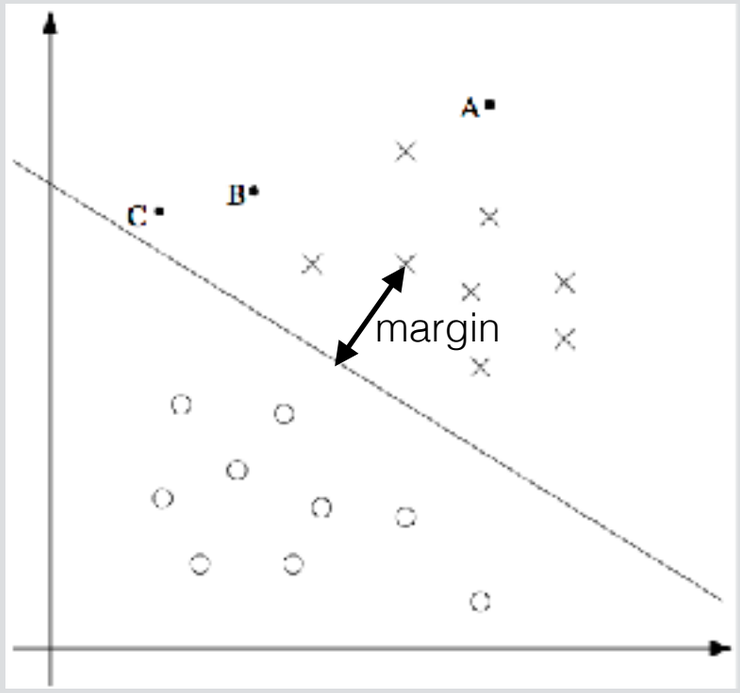
  - Can be implemented by similar gradient decent algorithm.

Support Vector Machine

- In logistic regression, how do we make a prediction of a future label y from an observed data x?
  - We find $\sum_{j=1}^{k} c_j x^{(j)}$ and see if it is greater than 0 or not.
  - For an $y = 1$, as $\sum_{j=1}^{k} c_j x^{(j)}$ is larger, we have more confidence in our prediction.
- Given training samples (y_i, x_i), suppose they are linearly separable, we want to find a hyperplane to separate labels $y = 1$ and $y = 0$.

Support Vector Machine





- Not only want the hyperplane to separate data, but want them to be separated reliably.
- We hope the minimal margin between the data points and the hyperplane can be maximized

Support Vector Machine

- From now on, change the linear coefficients c to w.
- The mathematical formulation:

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y_i \ (w^T x_i \ + b) \geq 1, \quad i = 1, \ldots, m \end{aligned}$$

- The dual problem:

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i \ y_i \ \alpha_i \alpha_j \langle x_i \ , x_i \ \rangle. \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y_i \ = 0, \end{aligned}$$

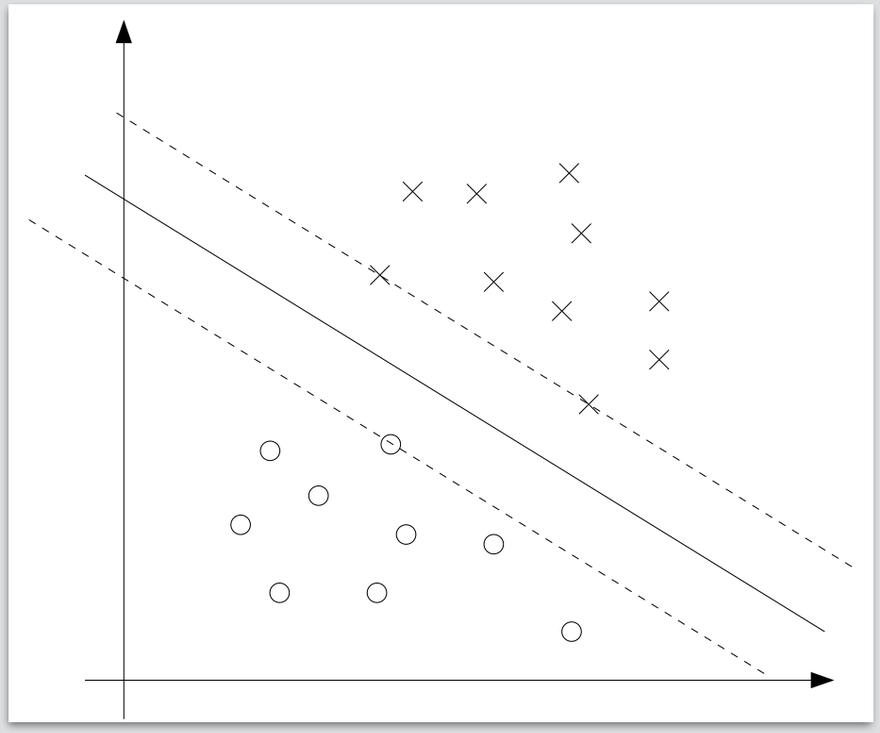
-

Support Vector Machine

- Solving the Lagrange multiplier, we get for the optimal α_i:

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

- The x_i with $\alpha_i \neq 0$ are called the support vectors.



Support Vector Machine

- To make a prediction for a future data x, we need to compute

$$w^T x + b = \sum_{i=1}^{m} \alpha_i y_i \langle x_i , x \rangle + b$$

- Only the support vector is needed to make the prediction
- Only have linear combinations of the data for prediction
- To generalize: use the kernel on the data:

$$\text{Data space} \qquad x \mapsto \phi(x) \qquad \text{Feature space}$$

- Kernel SVM: apply linear SVM on the kernel space.

Neural Networks

- Again, consider the label $y = \{0,1\}$.
- We want to use the linear indicator function to predict y.

$$f(x) = \theta\left(\sum_{j=1}^{k} w_j x^{(j)}\right)$$

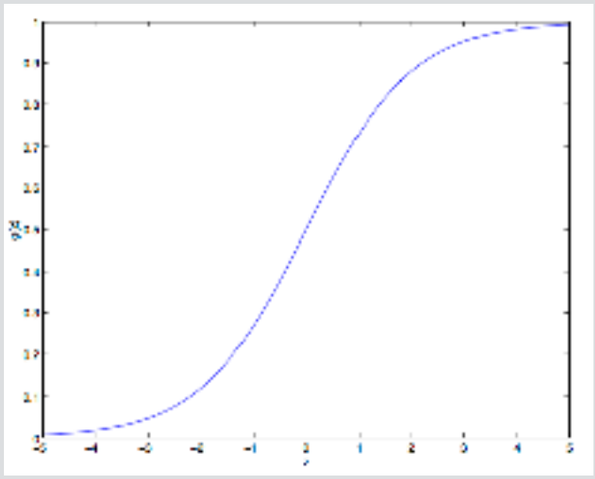
- The empirical risk is: $R_{\text{emp}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$
  - This is the empirical rate of making wrong decision.
- It is impossible to use regular gradient-based methods to find the optimal solution of this empirical risk.
  - Approximate the indicator by sigmoid functions.

Neural Networks

- Recall the sigmoid function:

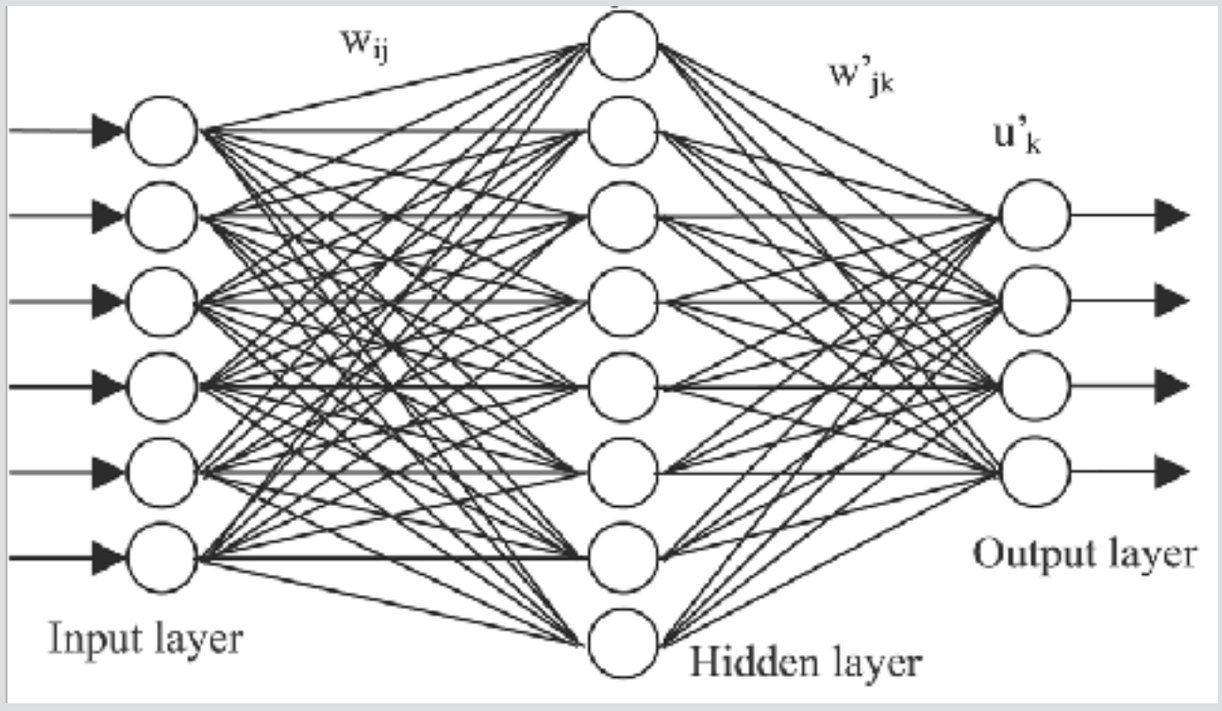
$$f(x) = g\left(\sum_{j=1}^{k} w_j x^{(j)}\right)$$

$$= \frac{1}{1 + e^{-\sum_{j=1}^{k} w_j x^{(j)}}}$$



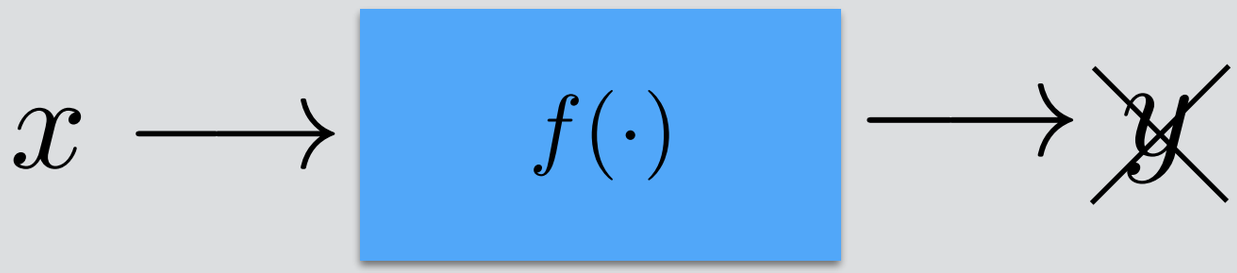
- The empirical risk: $R_{\text{emp}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$
  - Can apply gradient decent method to sigmoid function
- What if sigmoid function is not good approximation?
  - Iteratively using sigmoid function, back propagation.

Neural Networks



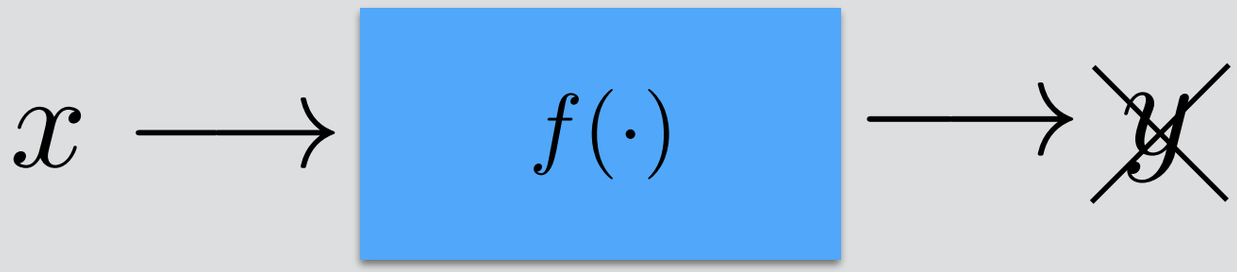
- The more layers of sigmoid functions used, the better the approximation is.
  - Too many layers = high complexity + overfitting.
  - The systematic design guideline is still an open problem.

Unsupervised Learning



- Similar to the supervised learning, but without labels.
  - Still want to learn the machine *f*.
  - Significantly harder in general.
- Often deal with clustering or classification problems.

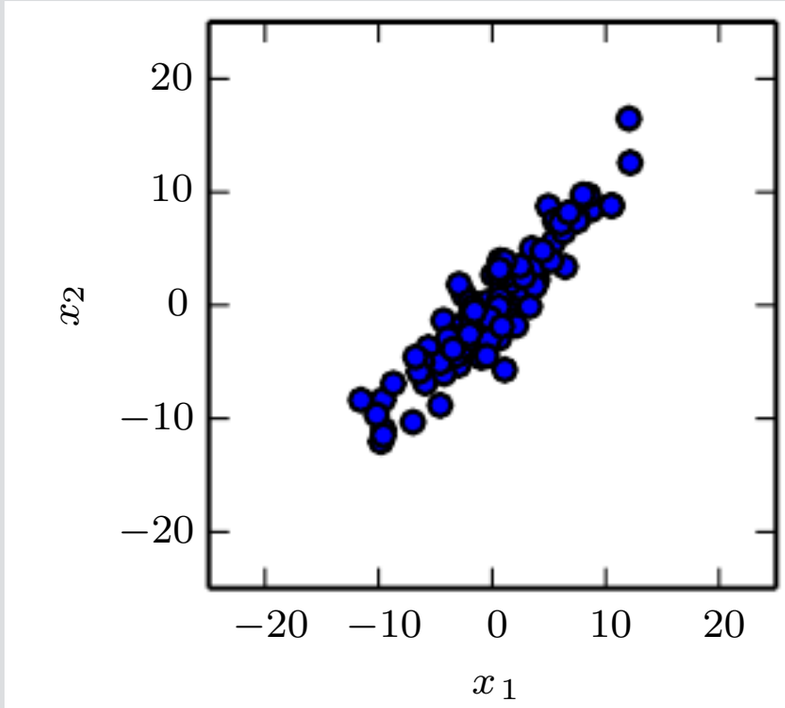
Unsupervised Learning

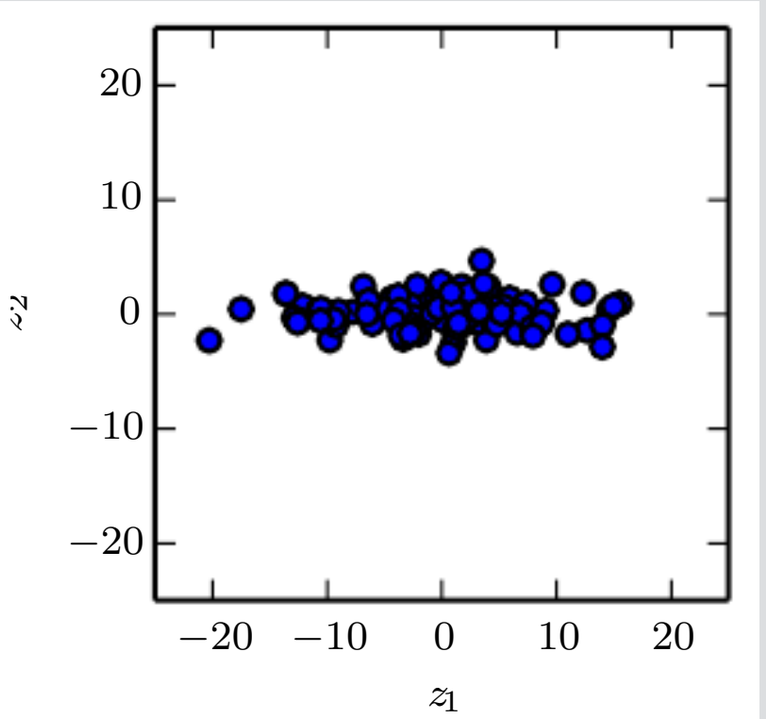


- How can we learn without label?
  - The data x are often in the form of vectors.
  - Investigate hidden structures behind data vectors x.
  - Example: two groups {0000, 1111, 0011, 1100} and {0101, 1010}.
- Unsupervised learning can often be viewed as extracting some kind of common information shared among data.

Principal Component Analysis

- Given some 2-d data vectors, find a direction so that these vectors are aligned.





Reinforcement Learning

- A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle), without a teacher explicitly telling it whether it has come close to its goal. Another example is learning to play a game by playing against an opponent.
- Alphago!
  - Markov decision process
  - Value iteration and Policy iteration