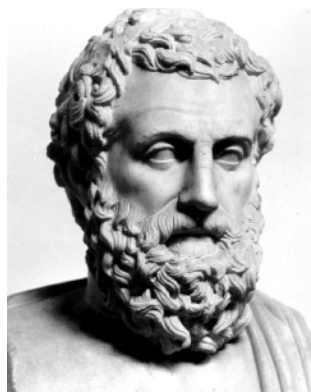


Artificial Intelligence For NLP

Lesson- 08 - 09

人工智能与自然语言处理课程组

2019.August. 17



Review

- In Last lesson:
 - 1. What's the machine learning?
 - 2. What's the *model* ?
 - 3. What's the overfitting and under fitting, why?
 - 4. What's the classification and regression?
 - 5. What's the categorical and numerical?
 - 6. What's the outlier?

Outline

- Supervised Learning
 - 1. Classification and Regression
 - 2. Logistic Regression
 - 3. Bayes Models
 - 4. KNN Models
 - 5. SVM Models
 - 6. Decision Tree
- Scikit-Learning
- Unsupervised Learning
 - K-means, hierarchy clustering, embedding clustering
- Semi-Supervised Learning
 - Active Learning
 - Machine Learning + Search Policy
- Importance of Preprocessing

Supervised Learning

Data: $D = \{D_1, D_2, \dots, D_n\}$ a set of n examples

$$D_i = \langle \mathbf{x}_i, y_i \rangle$$

$\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$ is an input vector of size d

y_i is the desired output (given by a teacher)

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

s.t. $y_i \approx f(\mathbf{x}_i)$ for all $i = 1, \dots, n$

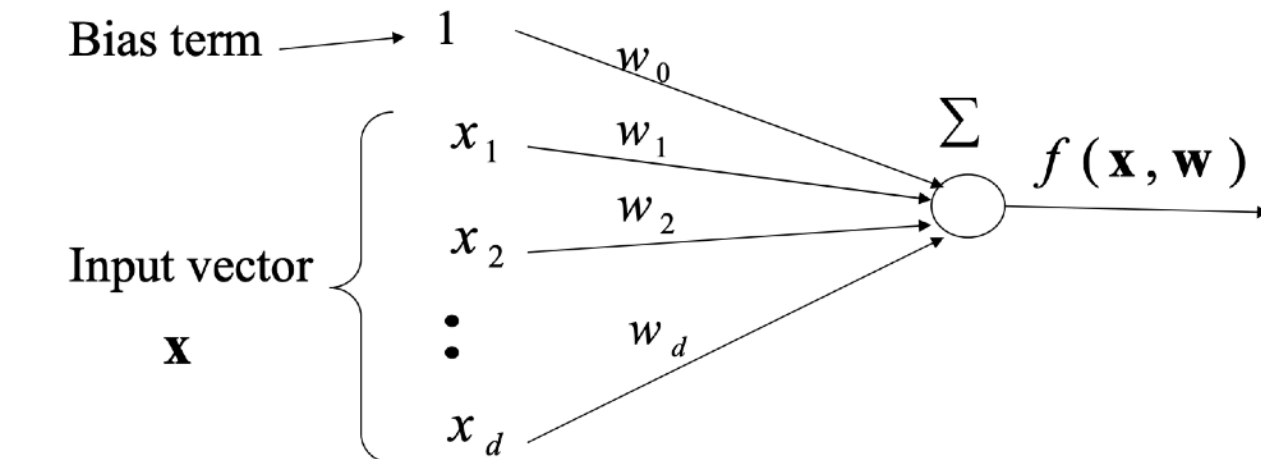
- **Regression:** Y is **continuous**
Example: earnings, product orders \rightarrow company stock price
 - **Classification:** Y is **discrete**
Example: handwritten digit in binary form \rightarrow digit label
-

Linear Regression & Logistic Regression

- Someone called 'LR' model.
- **Function** $f : X \rightarrow Y$ is a linear combination of input components

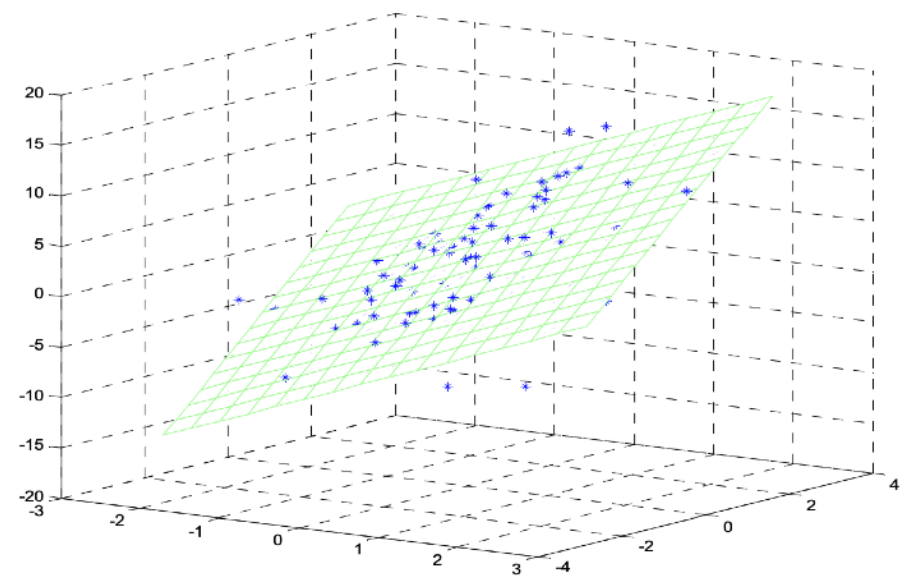
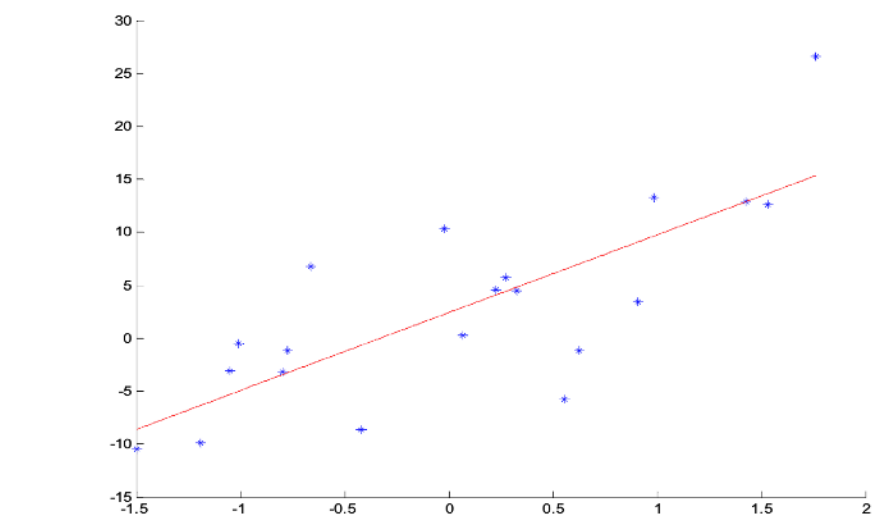
$$f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \dots w_dx_d = w_0 + \sum_{j=1}^d w_jx_j$$

w_0, w_1, \dots, w_k - **parameters (weights)**

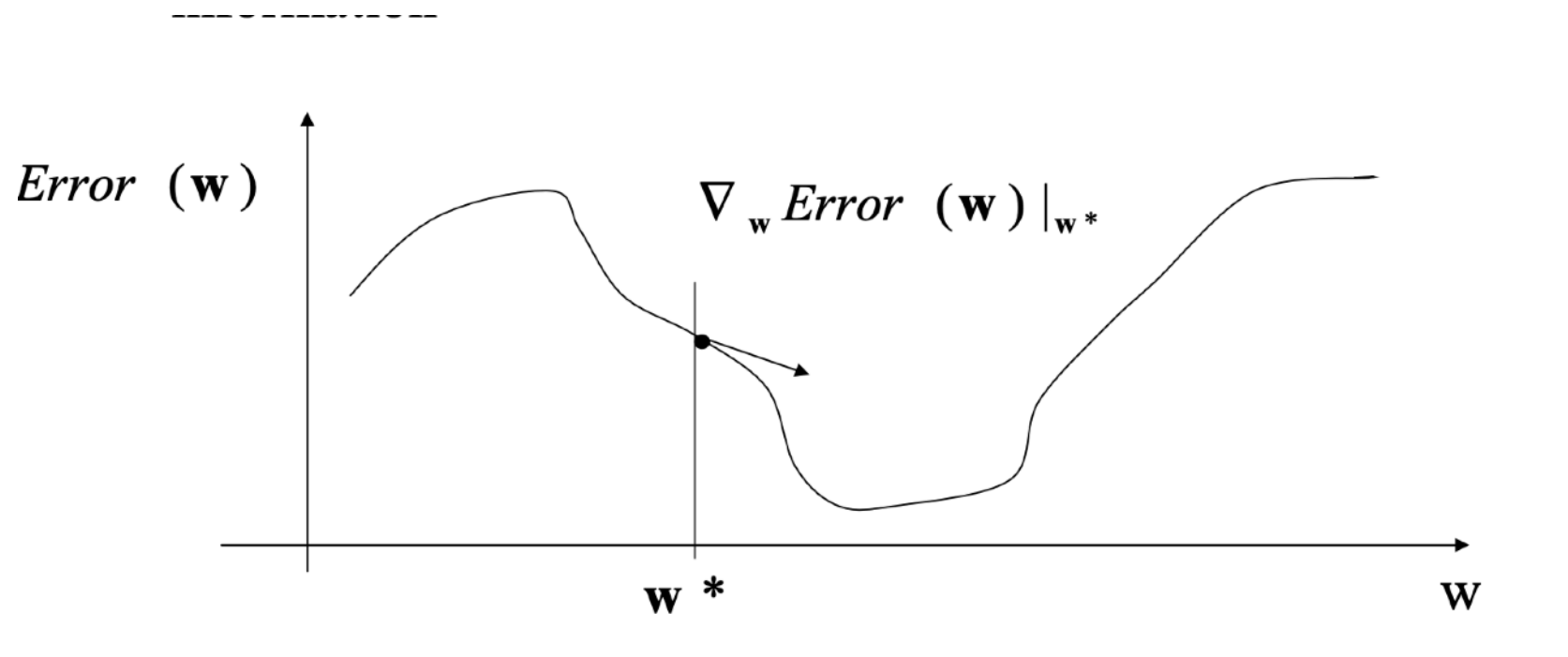


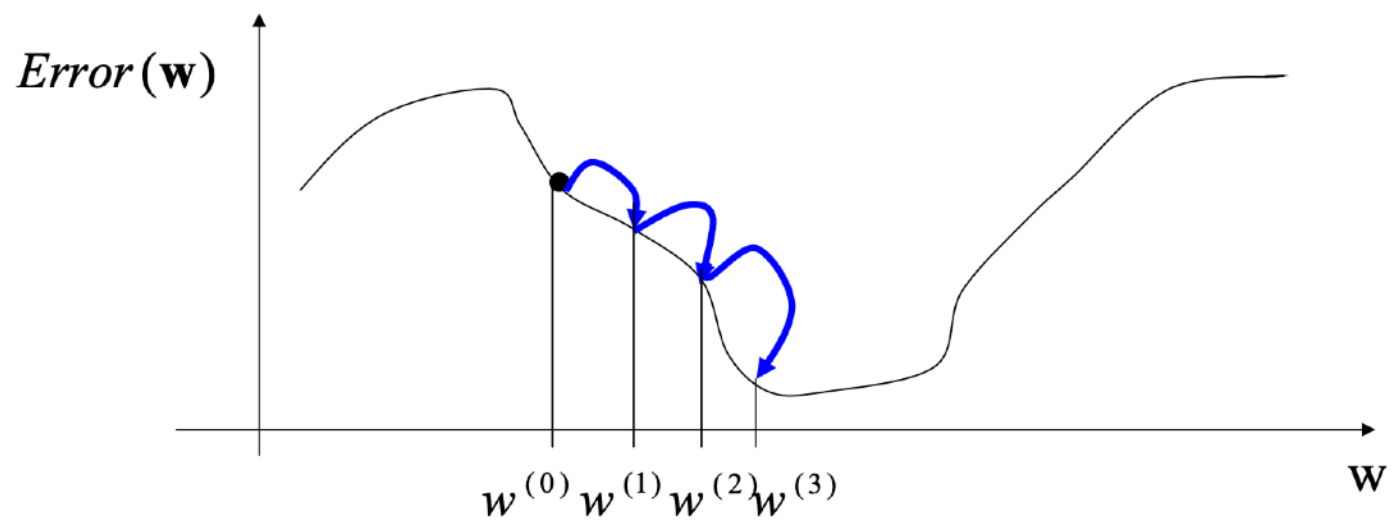
Mean Squared Error

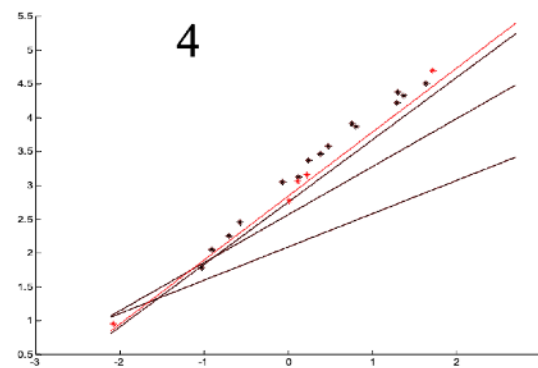
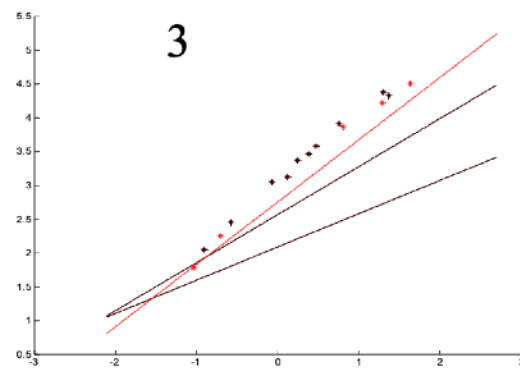
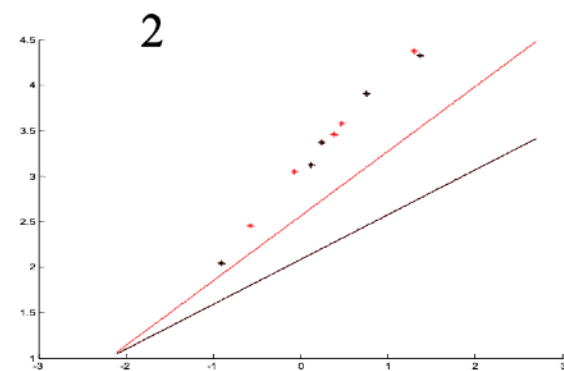
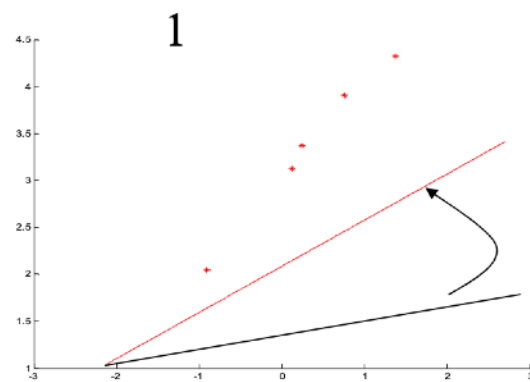
- Convex Function



Gradient Descent





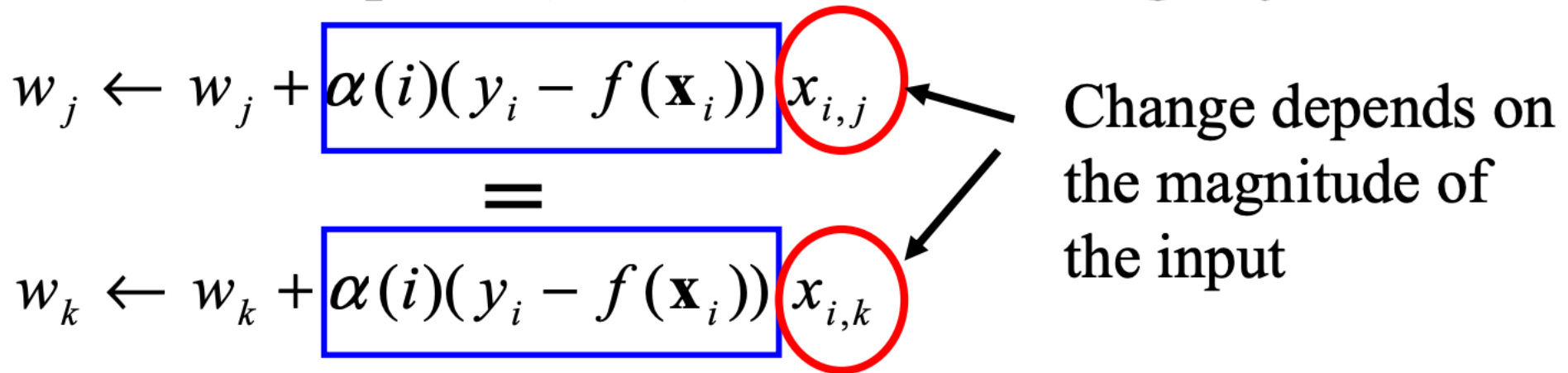


Practical Concerns

- Normalization and Scalar

$$\begin{aligned} w_j &\leftarrow w_j + \boxed{\alpha(i)(y_i - f(\mathbf{x}_i))} \boxed{x_{i,j}} \\ &= \\ w_k &\leftarrow w_k + \boxed{\alpha(i)(y_i - f(\mathbf{x}_i))} \boxed{x_{i,k}} \end{aligned}$$

Change depends on the magnitude of the input



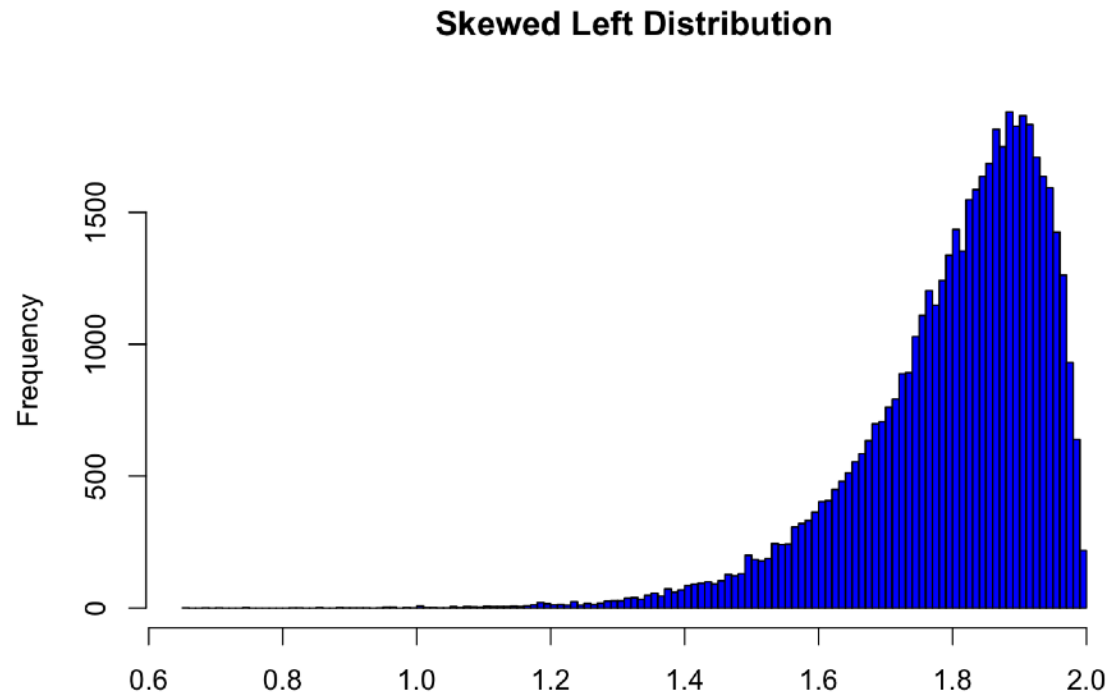
$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j}$$

$$\sigma_j^2 = \frac{1}{n-1} \sum_{i=1}^n (x_{i,j} - \bar{x}_j)^2$$

New input: $\tilde{x}_{i,j} = \frac{(x_{i,j} - \bar{x}_j)}{\sigma_j}$

Feature function

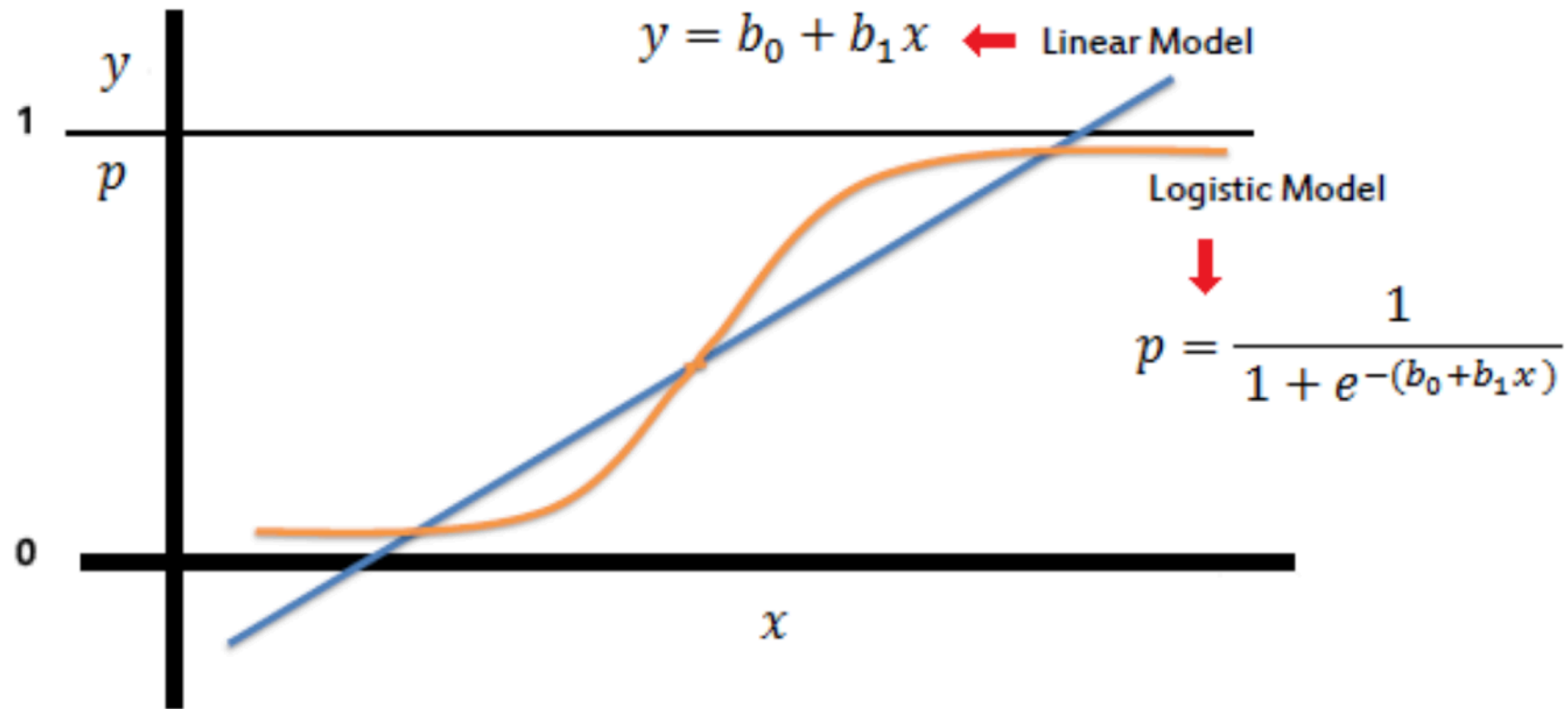
- Nonlinear attributes:
 - Bank Deposit
 - Timing
 - Etc



Scikit-learning

- Scikit learning is our friend.
- > We search it in Google.

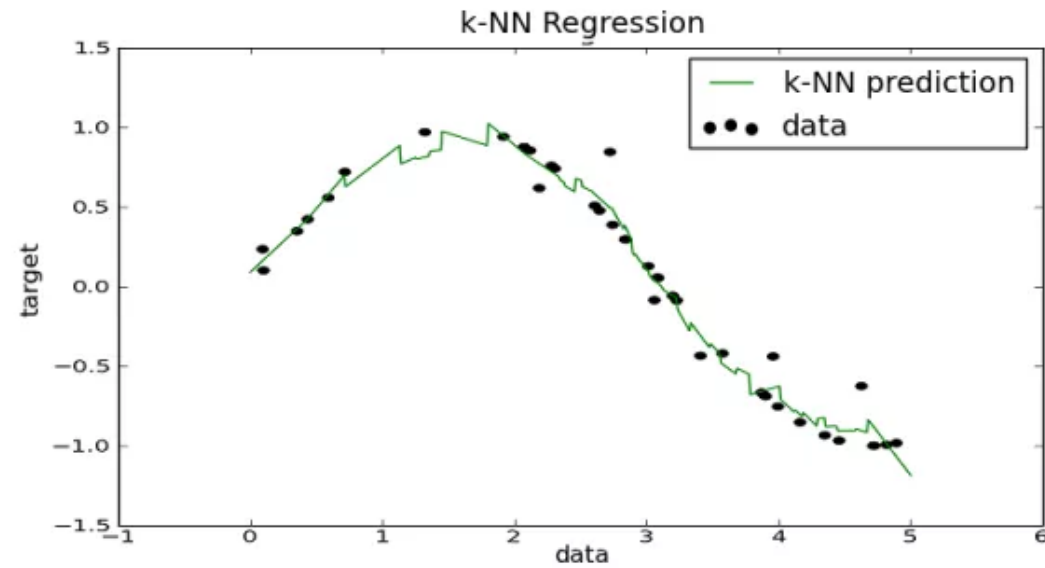
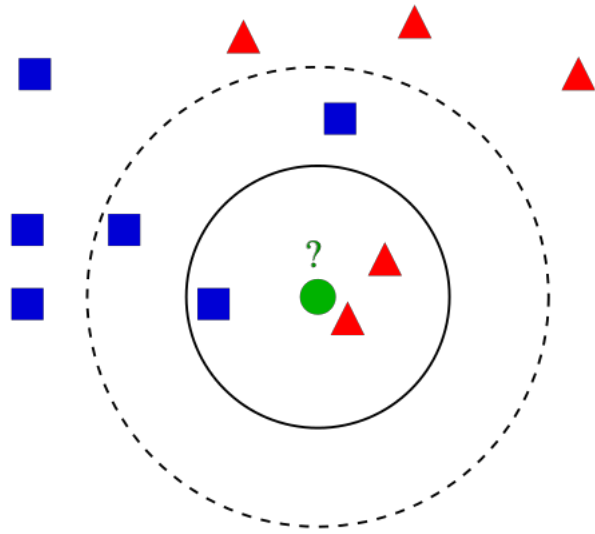
Logistic Regression



Loss Function and Gradient Descent

- Practical Reference: https://github.com/SSaishruthi/LogisticRegression_Vectorized_Implementation/blob/master/Logistic_Regression.ipynb
- Data Description: <https://archive.ics.uci.edu/ml/datasets/iris>

K-Nearest-Neighbors

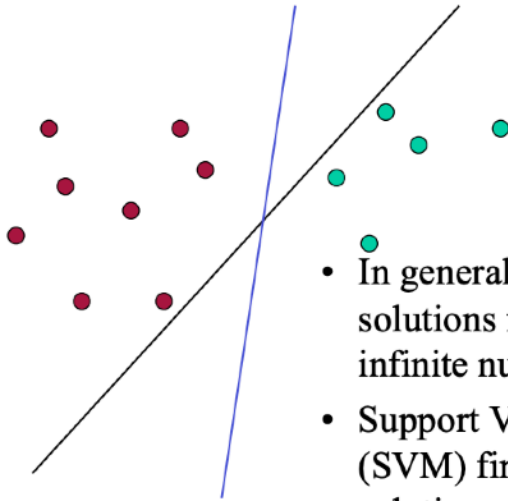


KNN

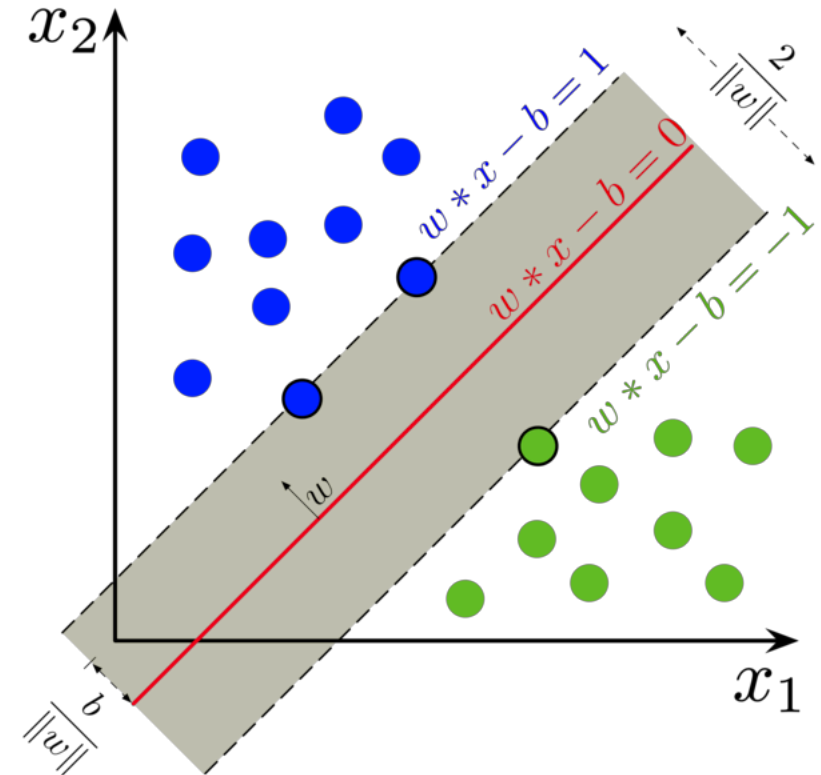
- If we want write a program solve it, it's simple.
- 1-nearest neighbor classifier / regression
- Weighted nearest neighbor classifier / regression
- Maybe all the machine learning problem could use this method.
 - Dimension reduction
 - With the dimension increase, we need much more data to fit.
 - Euclidean distance is unhelpful in high dimensions because all vectors are almost equidistant to the search query vector. (*Image a circle*)
 - Outlinear
 - Memory Consuming
 - Timing Cusuming *in high dimension*

Support Vector Machine

Recall from 1-layer nets : Which Separating Hyperplane?

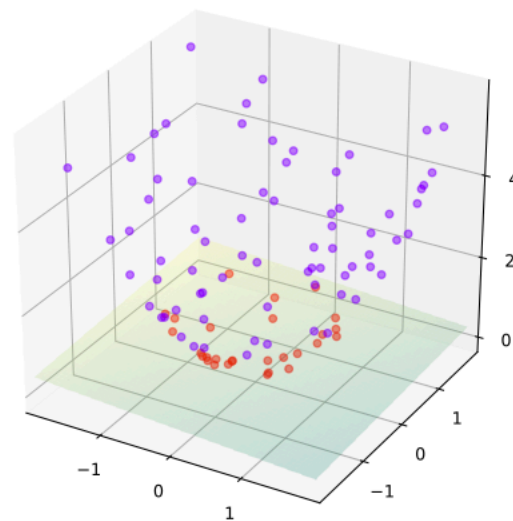
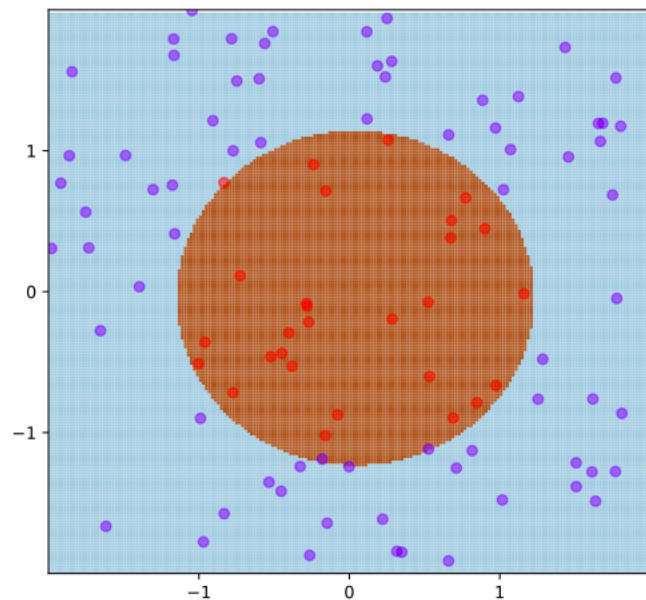


- In general, lots of possible solutions for a, b, c (an infinite number!)
- Support Vector Machine (SVM) finds an optimal solution



$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta)$$

Kernel Function



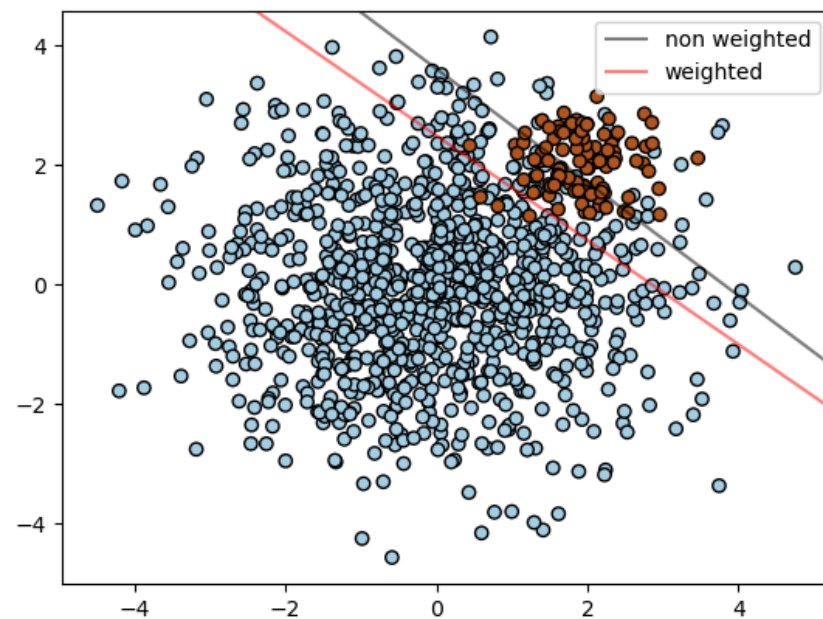
$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}.$$

- **Polynomial (homogeneous)**: $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j)^d$.
 - **Polynomial (inhomogeneous)**: $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d$.
 - **Gaussian radial basis function**: $k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2)$ for $\gamma > 0$. Sometimes parametrized using $\gamma = 1/(2\sigma^2)$.
 - **Hyperbolic tangent**: $k(\vec{x}_i, \vec{x}_j) = \tanh(\kappa \vec{x}_i \cdot \vec{x}_j + c)$ for some (not every) $\kappa > 0$ and $c < 0$.
-

- <https://cs.stanford.edu/~karpathy/svmjs/demo/>

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}.$$

Unbalance Problem



Important of Preprocessing

- **Balance Analysis:** We need to know the balance information. *Spam email*.
- **Remove Noise.** Our models assumes that your input and output variables are not noisy. Consider using data cleaning operations that let you better expose and clarify the signal in your data. This is most important for the output variable and you want to remove outliers in the output variable (y) if possible.
- **Remove Collinearity.** Model over-fit your data when you have highly correlated input variables. Consider calculating pairwise correlations for your input data and removing the most correlated.
- **Rescale Inputs:** Model will often make more reliable predictions if you rescale input variables using standardization or normalization.

Bayes & Gaussian Bayesian Classifier

- Gaussian Bayesian:

- When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a

continuous attribute, x

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Decision Tree

Random Forest

- 1. Create a 'bootstrapped' data set

Bam!!! We've created a bootstrapped dataset!!!

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Random Forest

- 1. Create a 'bootstrapped' data set

Bam!!! We've created a bootstrapped dataset!!!

Original Dataset

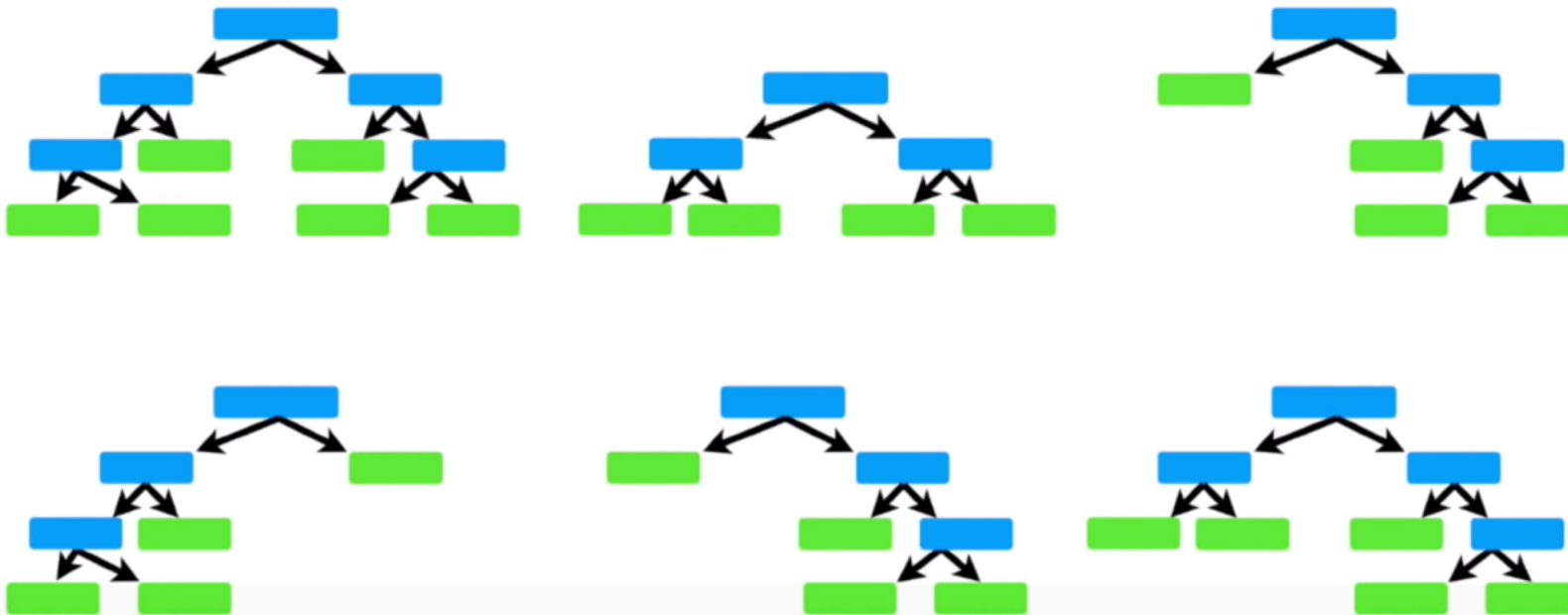
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Random Forest

Ideally, you'd do this 100's of times, but we only have space to show 6... but you get the idea.



1. Predicate using the forest

2. Before predicating, we use Out-Of-Bagging sample to evaluate our model.

3. Using the evaluated result, we could choose the right variables number to be chosen in Random Forest.

K-means Cluster

Embedding Cluster

Semi Supervised & Activate Learning

Assignments

- 1. Using the new data to test different models
 - KNN, Logistic Regression, SVM, Bayes, Random Forest, XGBoost
 - To Classify if a new a by Xinhua.
- 2. Summary the advantages and disadvantages about Linear Regression, Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Bayes, etc.