Part I: The Elements of Machine Learning

Xijia Liu
xijia.liu.18@gmail.com

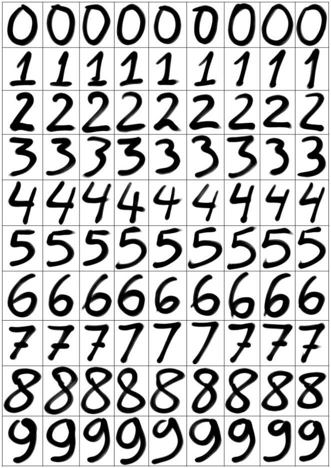
Department of mathematics and mathematical statistics
Umeå University

December 2, 2019

Outline

Philosophy

- Examples.

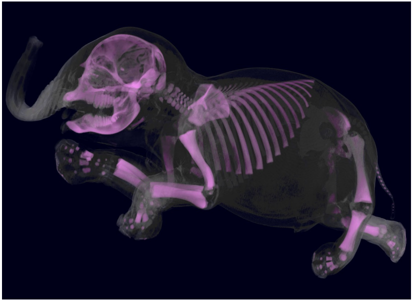


Philosophy

- Examples.

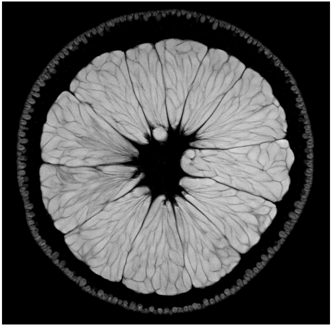
Philosophy

- Examples.



Philosophy

- Examples.



Philosophy

- Examples.
- Two kinds of information

Philosophy

- Examples.
- Two kinds of information
  - $\mathbf{x}$ information: accessible, cheap.

Philosophy

- Examples.
- Two kinds of information
  - **x** information: accessible, cheap.
  - **y** information: expensive, annotation by human.

Philosophy

- Examples.
- Two kinds of information
  - **x** information: accessible, cheap.
  - **y** information: expensive, annotation by human.
- Aim: find the connection between **x** and **y**, e.g. a map f from **x** to **y**

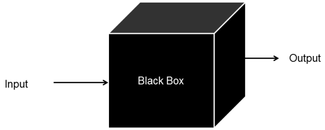
x **y**

Philosophy

- Examples.
- Two kinds of information
    - **x** information: accessible, cheap.
    - **y** information: expensive, annotation by human.
- Aim: find the connection between **x** and **y**, e.g. a map f from **x** to **y**
- Prediction: automated, accurate, safe and objective solution.

Philosophy

- Examples.
- Two kinds of information
  - **x** information: accessible, cheap.
  - **y** information: expensive, annotation by human.
- Aim: find the connection between **x** and **y**, e.g. a map f from **x** to **y**
- Prediction: automated, accurate, safe and objective solution.
- Ability of prediction V.S. interpretation.



Internal behavior of the code is unknown

Supervised Learning

Observations from my lovely son ♡

Observations from my lovely son ♡

- At beginning, he repeatedly asked me: "**What is this, papa?**"

Observations from my lovely son ♡

- At beginning, he repeatedly asked me: "**What is this, papa?**"
- After a while, his question style was switched from special to general. "**Papa, is this an apple?**"

Observations from my lovely son ♡

- At beginning, he repeatedly asked me: "**What is this, papa?**"
- After a while, his question style was switched from special to general. "**Papa, is this an apple?**"
- At last, he just became an amazing classifier about apple! **Accuracy is** 100%

Observations from my lovely son ♡

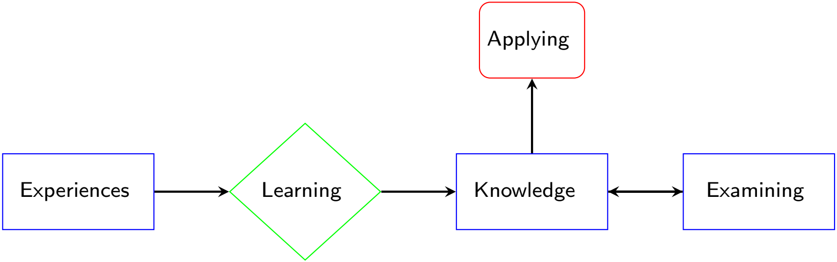
- At beginning, he repeatedly asked me: "**What is this, papa?**"

- After a while, his question style was switched from special to general. "**Papa, is this an apple?**"

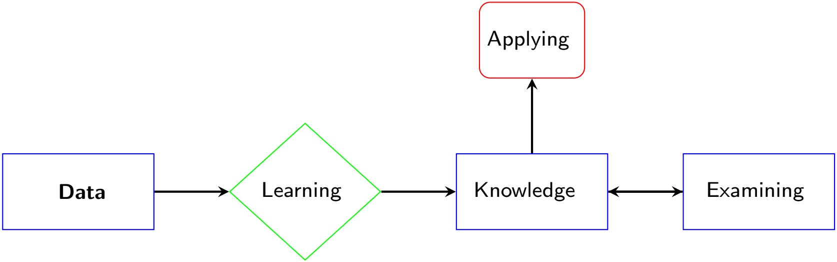
- At last, he just became an amazing classifier about apple! **Accuracy is** 100%

- It's so magical! I did not summarize any rule for him!

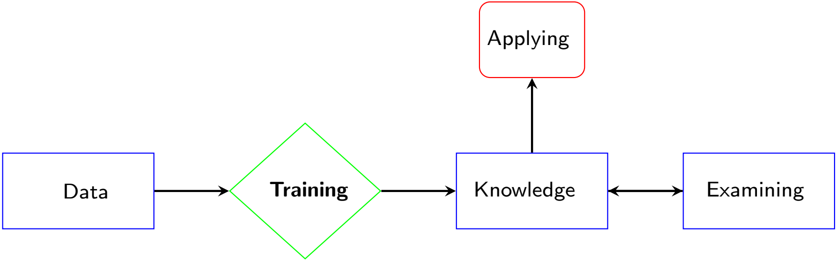
Our learning process



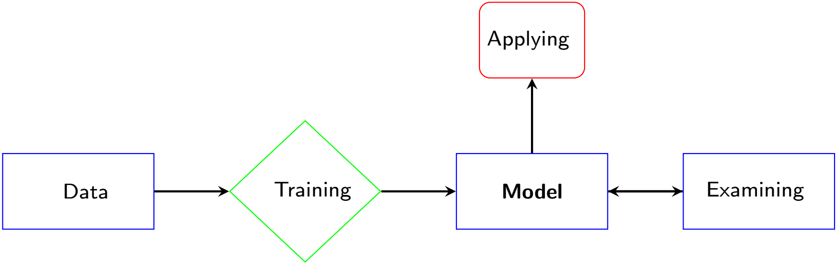
Our learning process



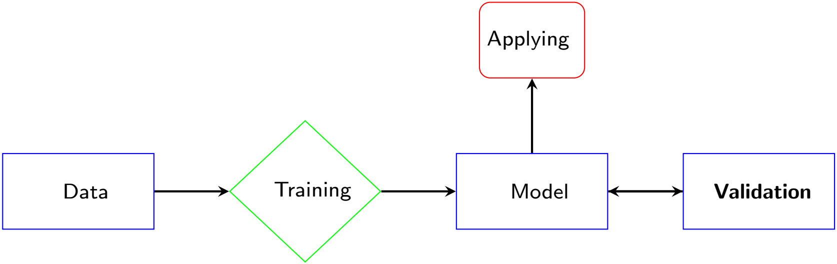
Our learning process



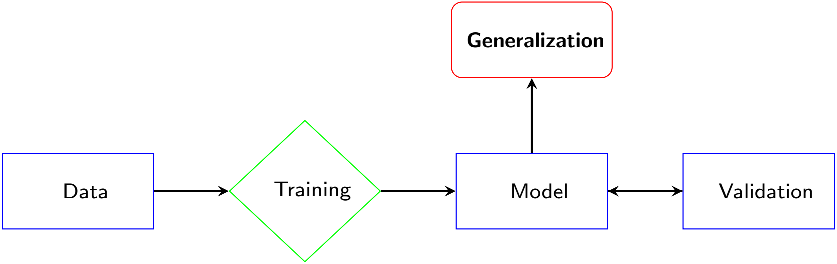
Our learning process



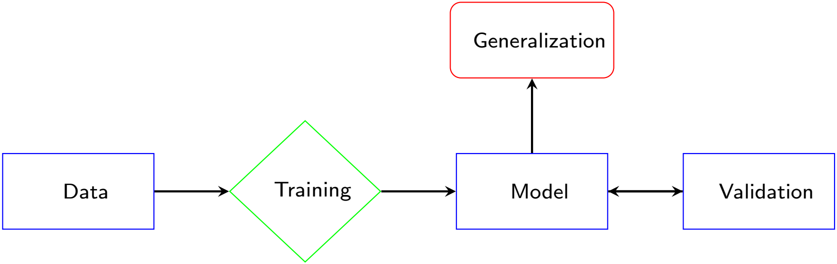
Our learning process



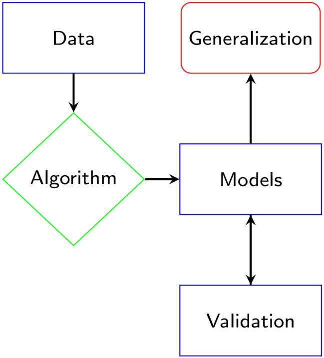
Our learning process



Our learning process

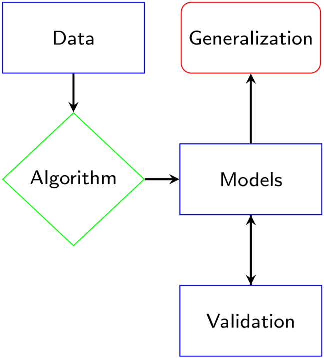


Supervised Learning



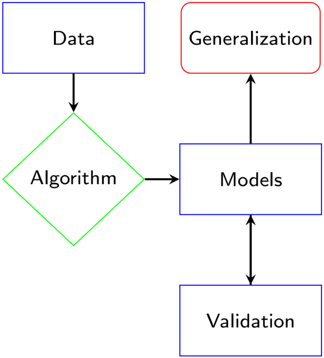
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.



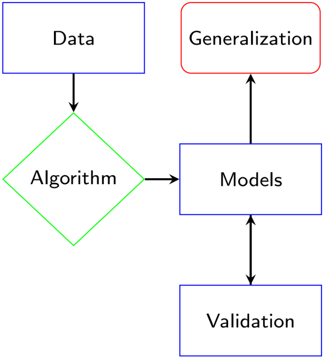
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:



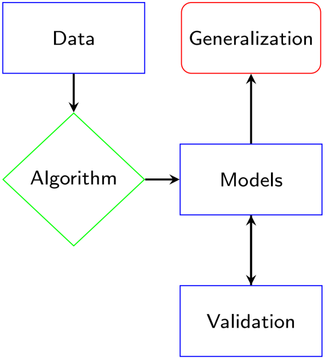
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



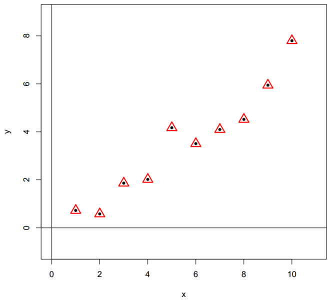
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).



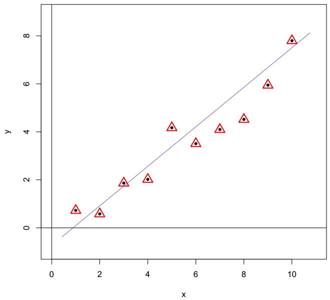
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



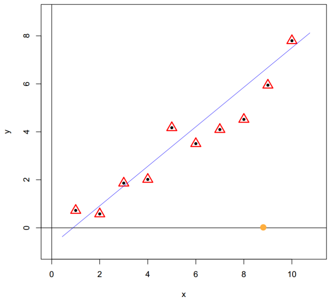
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



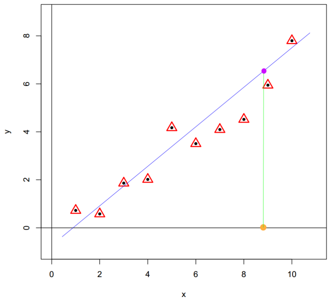
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



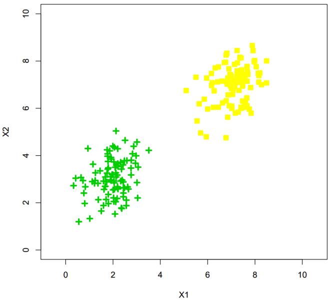
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



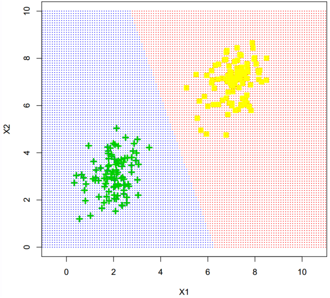
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



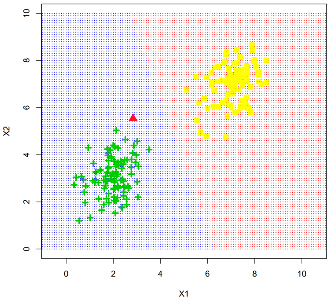
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



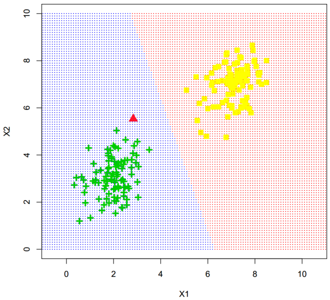
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).



Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).

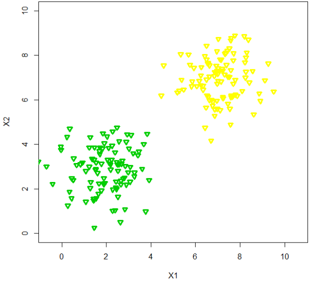


Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.

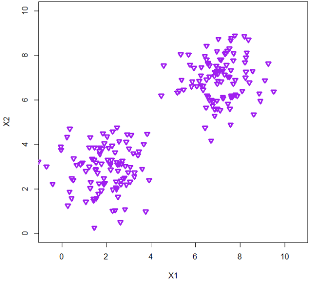
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.



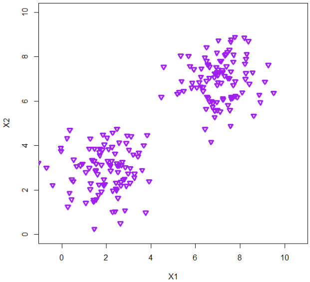
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.



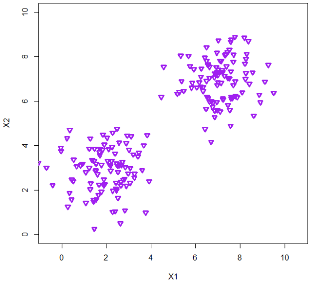
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.



Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.



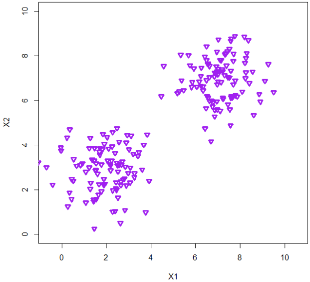
Supervised Learning

$x_1 \cdots x_n$ $\quad$ $x_i \sim N(\mu, \sigma^2)$

$g(x)$

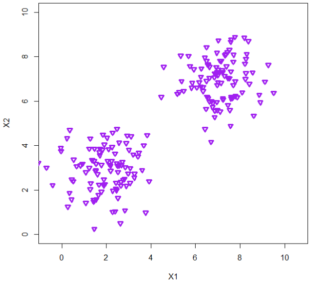
- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.

$z = w^T x$



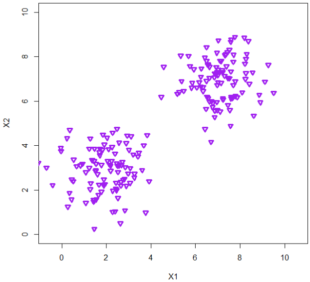
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.



Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.



Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.

Supervised Learning

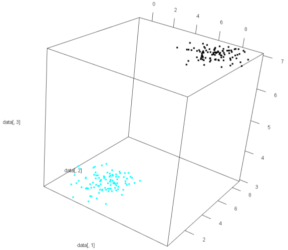
- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:

Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.

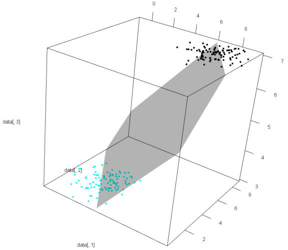
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



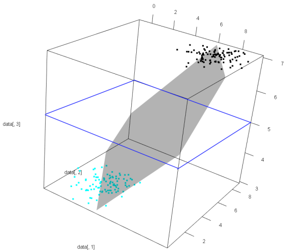
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



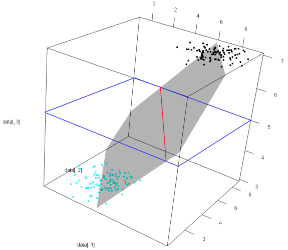
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



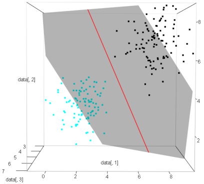
Supervised Learning

$[0,1]$ $\quad$ R

$P = E(Y|x) = w^T x$

$P(Y=1)$

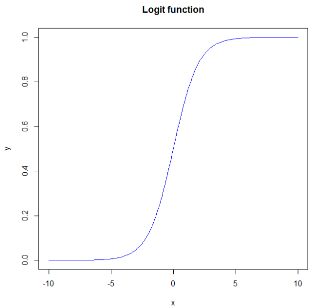
- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



Supervised Learning

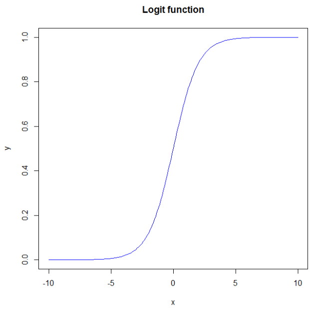
$P(y=1|x) = \phi(w^T x) \in [0,1] \quad \phi(s) = \frac{1}{1+e^{-s}}$

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



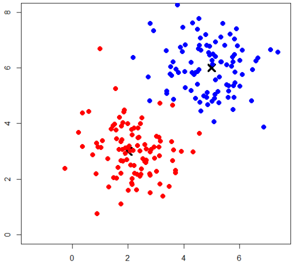
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



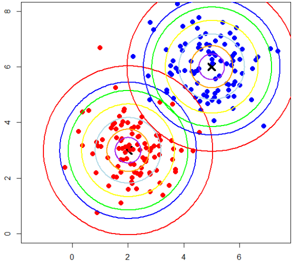
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



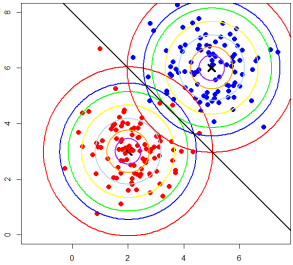
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.



Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.
  - Key: $\widehat{y} = arg\max_k \Pr(y = k|\mathbf{x})$

Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.
  - Key: $\widehat{y} = arg\max_k \Pr(y = k|\mathbf{x})$
  - Maximum margin classifier

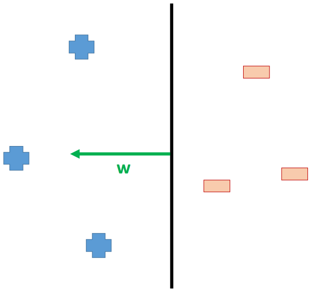
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.
  - Key: $\widehat{y} = arg\max_k \Pr(y = k|\mathbf{x})$
  - Maximum margin classifier
  - Perceptron classfier: $Sign(\mathbf{w}'\mathbf{x})$

$$Sign(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \end{cases}$$

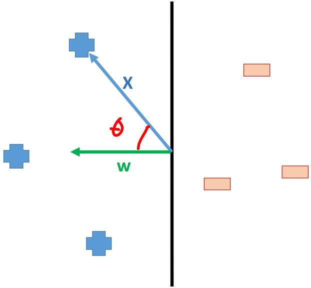
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.
  - Key: $\widehat{y} = arg\max_k \Pr(y = k|\mathbf{x})$
  - Maximum margin classifier
  - Perceptron classfier: $Sign(\mathbf{w}'\mathbf{x})$



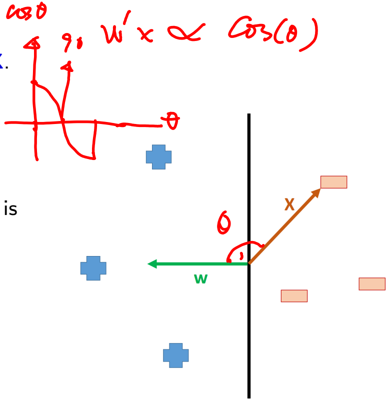
Supervised Learning

- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \rightarrow y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \rightarrow z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.
  - Key: $\widehat{y} = arg \max_k \Pr(y = k|\mathbf{x})$
  - Maximum margin classifier
  - Perceptron classfier: $Sign(\mathbf{w}'\mathbf{x})$



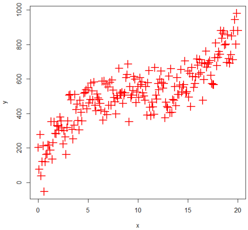
Supervised Learning

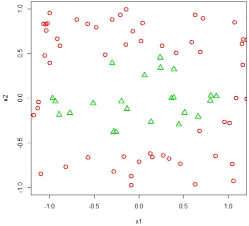
- Data: Target variable y, feature variables $\mathbf{X}$.
- Types of learning:
  - Supervised Learning: $f(\mathbf{X}) \to y$. regression, classification (classifier, decision boundary).
  - Unsupervised Learning: $g(\mathbf{X}) \to z$, z is latent variable. Density estimation, Grouping, Features extraction.
- Supervised Learning algorithms:
  - Linear classifier: Logistic regression, Gaussian Discriminant.
  - Key: $\widehat{y} = arg\max_k \Pr(y = k|\mathbf{x})$
  - Maximum margin classifier
  - Perceptron classfier: $Sign(\mathbf{w}'\mathbf{x})$

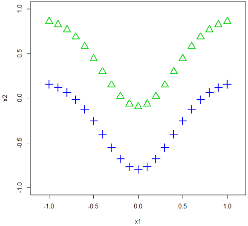


Non-linear Extension

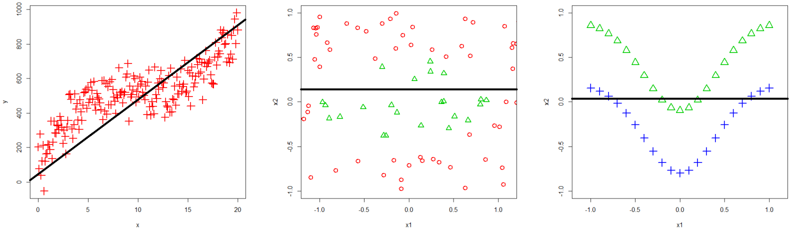
Help!


y
x
0
200
400
600
800
1000
0
5
10
15
20

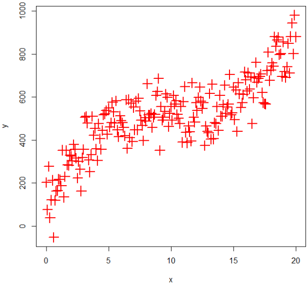

x2
x1
-1.0
-0.5
0.0
0.5
1.0


x2
x1
-1.0
-0.5
0.0
0.5
1.0

Help!

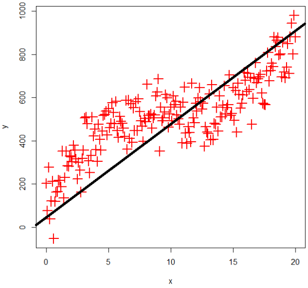


Polynomial Regression



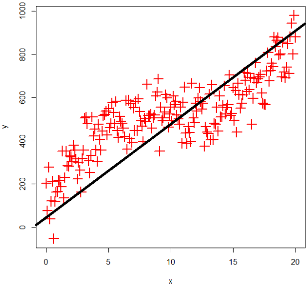
Polynomial Regression

- In simple linear regression, we fit the data as a linear function of x



Polynomial Regression

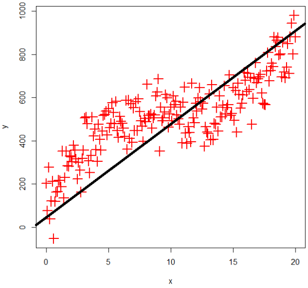
- In simple linear regression, we fit the data as a linear function of x
- Here, it seems that we need to fit our data as a curve.



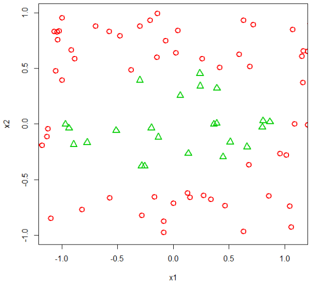
Polynomial Regression

- In simple linear regression, we fit the data as a linear function of x
- Here, it seems that we need to fit our data as a curve.
- Polynomial regression

$$y_i = w_0 + w_1 x_i + w_2 x_i^2 + \ldots + w_k x_i^k + \epsilon_i$$

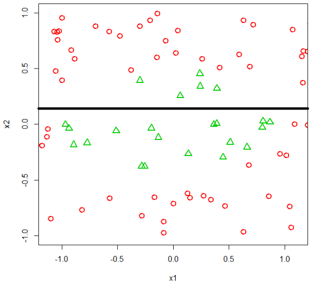


Non linear Separable



Non linear Separable

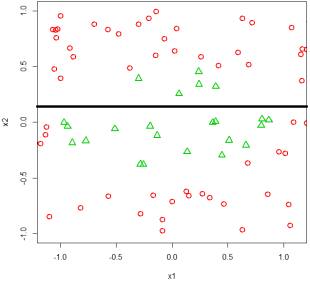
- Non linear separable. Hyper plain decision boundary seems does not work here...



Non linear Separable

- Non linear separable. Hyper plain decision boundary seems does not work here...
- It seems that the decision boundary is an ellipse.

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = 1$$



$(x_1, x_2) \longrightarrow (x_1^2, \sqrt{2}x_1 x_2, x_2^2)$

Augmented Features Space

- In the first problem, we did a transformation $\phi : \mathbb{R} \to \mathbb{R}^k$

$$x \to (x, x^2, ..., x^k)$$

Augmented Features Space

- In the first problem, we did a transformation $\phi : \mathbb{R} \to \mathbb{R}^k$

$$x \to (x, x^2, ..., x^k)$$

- In the second problem, we did a transformation $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

$$(x_1, x_2) \to (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

Augmented Features Space

- In the first problem, we did a transformation $\phi : \mathbb{R} \to \mathbb{R}^k$

$$x \to (x, x^2, ..., x^k)$$

- In the second problem, we did a transformation $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

$$(x_1, x_2) \to (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

- ϕ is called feature mapping or basis function.

Augmented Features Space

- In the first problem, we did a transformation $\phi : \mathbb{R} \to \mathbb{R}^k$

$$x \to (x, x^2, ..., x^k)$$

- In the second problem, we did a transformation $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

$$(x_1, x_2) \to (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

- ϕ is called feature mapping or basis function.
- Examples:
  - Identical transform x
  - Polynomial transform x^k
  - Non-linear function: $\log(x)$, $\sqrt{x}$

Augmented Features Space

- In the first problem, we did a transformation $\phi : \mathbb{R} \to \mathbb{R}^k$

$$x \to (x, x^2, ..., x^k)$$

- In the second problem, we did a transformation $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

$$(x_1, x_2) \to (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

- ϕ is called feature mapping or basis function.
- Examples:
  - Identical transform x
  - Polynomial transform x^k
  - Non-linear function: $\log(x)$, $\sqrt{x}$
  - Indicator function: $\mathbf{1}_{x\in(a,b)}$ local regression, spline.
  - Fourier transform, Wavelet transform.

Augmented Features Space

- In the first problem, we did a transformation $\phi : \mathbb{R} \to \mathbb{R}^k$

$$x \to (x, x^2, ..., x^k)$$

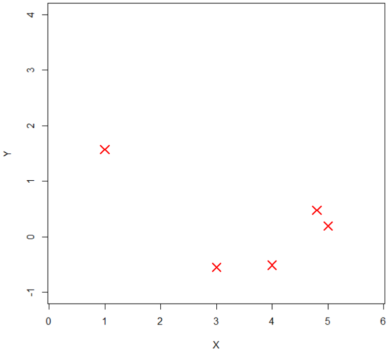
- In the second problem, we did a transformation $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

$$(x_1, x_2) \to (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

- ϕ is called feature mapping or basis function.
- Examples:
  - Identical transform x
  - Polynomial transform x^k
  - Non-linear function: $\log(x)$, $\sqrt{x}$
  - Indicator function: $\mathbf{1}_{x\in(a,b)}$ local regression, spline.
  - Fourier transform, Wavelet transform.

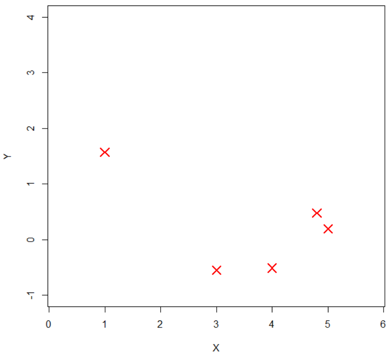
Over Fitting Problem

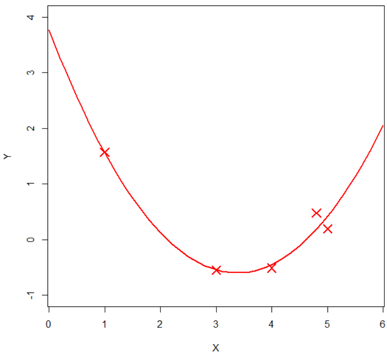
Training V.S. Generalization



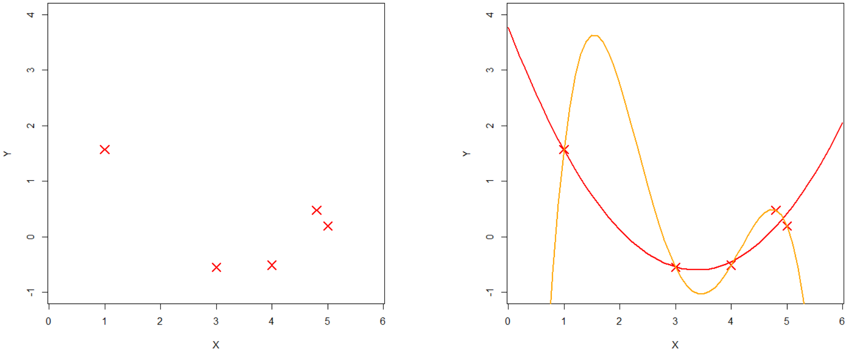
Training V.S. Generalization

$Y = \beta_0 + \beta_1 X_0 + \beta_2 X^2 + \beta_3 X^3 + \beta_2 X^4$

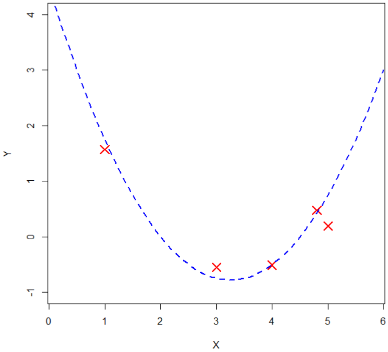


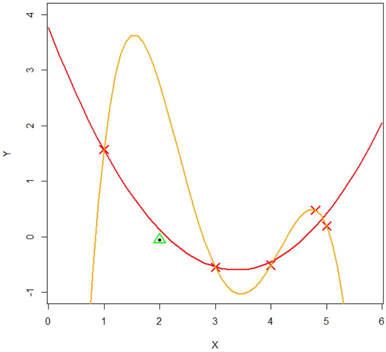


Training V.S. Generalization

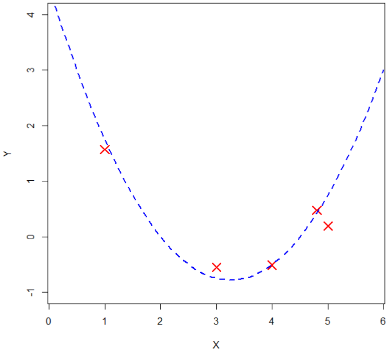


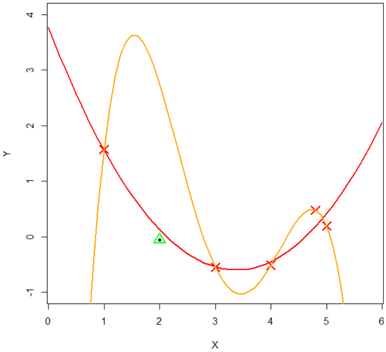
Training V.S. Generalization



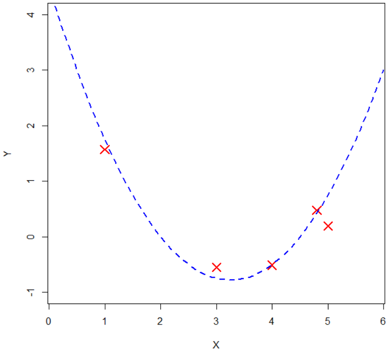


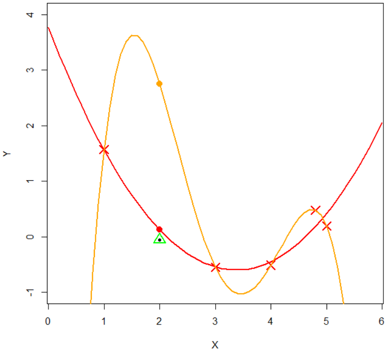
Training V.S. Generalization





Training V.S. Generalization





Overfitting

- The final aim of building a model is generalization.

Overfitting

- The final aim of building a model is generalization.
- Training set V.S. validation set.

Overfitting

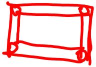
- The final aim of building a model is generalization.
- Training set V.S. validation set.
- Not only take into account the errors within the training set, E_T, but also the prediction error from validation set, E_V

Overfitting

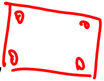
- The final aim of building a model is generalization.
- Training set V.S. validation set.
- Not only take into account the errors within the training set, E_T, but also the prediction error from validation set, E_V
- E_T is the measure of the goodness of fit; E_V is a measure of the overfitting.

	Orange	Red
$E_T(m)$	Perfect	Good enough
$E_V(m)$	Poor	Good
Complexity	Complicate	Simple

Overfitting



- The final aim of building a model is generalization.
- Training set V.S. validation set.



- Not only take into account the errors within the training set, E_T, but also the prediction error from validation set, E_V
- E_T is the measure of the goodness of fit; E_V is a measure of the overfitting.

	Orange	Red
$E_T(m)$	Perfect	Good enough
$E_V(m)$	Poor	Good
Complexity	Complicate	Simple

- Overfitting: Model the errors; Consequence is bad Generalization.

Overfitting

- The final aim of building a model is generalization.
- Training set V.S. validation set.
- Not only take into account the errors within the training set, E_T, but also the prediction error from validation set, E_V
- E_T is the measure of the goodness of fit; E_V is a measure of the overfitting.

	Orange	Red
$E_T(m)$	Perfect	Good enough
$E_V(m)$	Poor	Good
Complexity	Complicate	Simple

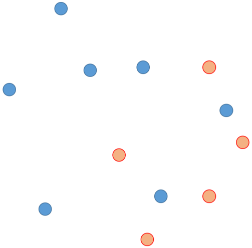
- Overfitting: Model the errors; Consequence is bad Generalization.

K Nearest Neighbors, follwing the local majority

- Predict the label as the majority of the K surroundings.

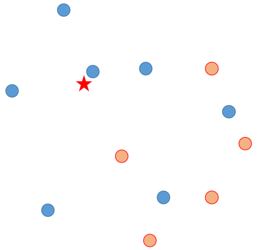
K Nearest Neighbors, follwing the local majority

- Predict the label as the majority of the K surroundings.
- Annotated observations $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$



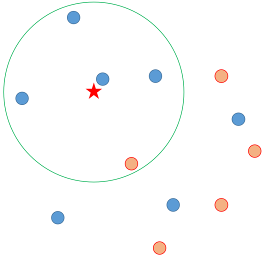
K Nearest Neighbors, follwing the local majority

- Predict the label as the majority of the K surroundings.
- Annotated observations $\{\mathbf{x}_i, y_i\}_{i=1}^N$
- For a new observation $\mathbf{x}_{new}$, find the K nearest neighbors given a metric.



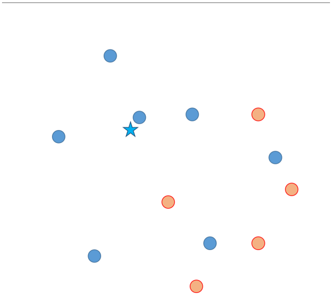
K Nearest Neighbors, follwing the local majority

- Predict the label as the majority of the K surroundings.
- Annotated observations $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$
- For a new observation $\mathbf{x}_{new}$, find the K nearest neighbors given a metric.



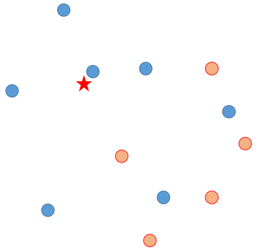
K Nearest Neighbors, follwing the local majority

- Predict the label as the majority of the K surroundings.
- Annotated observations $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$
- For a new observation $\mathbf{x}_{new}$, find the K nearest neighbors given a metric.



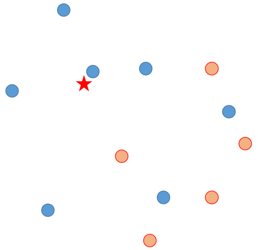
K Nearest Neighbors, follwing the local majority

- Predict the label as the majority of the K surroundings.
- Annotated observations $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$
- For a new observation $\mathbf{x}_{new}$, find the K nearest neighbors given a metric.
- It also can be applied for regression problem.



K Nearest Neighbors, follwing the local majority

- Predict the label as the majority of the K surroundings.
- Annotated observations $\{\mathbf{x}_i, y_i\}_{i=1}^N$
- For a new observation $\mathbf{x}_{new}$, find the K nearest neighbors given a metric.
- It also can be applied for regression problem.
- Memory based method and lazy learner.



Hyper-parameters and Model selection

- Model parameters:

- Hyper parameters:

Hyper-parameters and Model selection

- Model parameters:
    - (Logistic) regression, the regression coefficients.
- Hyper parameters:

Hyper-parameters and Model selection

- Model parameters:
  - (Logistic) regression, the regression coefficients.
  - LDA, the parameters of each Gaussian model.
- Hyper parameters:

Hyper-parameters and Model selection

- Model parameters:
  - (Logistic) regression, the regression coefficients.
  - LDA, the parameters of each Gaussian model.
  - Perceptron, the weight parameters.
- Hyper parameters:

Hyper-parameters and Model selection

- Model parameters:
  - (Logistic) regression, the regression coefficients.
  - LDA, the parameters of each Gaussian model.
  - Perceptron, the weight parameters.
- Hyper parameters:
  - Polynomial regression, the order of polynomial.

Hyper-parameters and Model selection

$X'X$

- Model parameters:
  - (Logistic) regression, the regression coefficients.
  - LDA, the parameters of each Gaussian model.
  - Perceptron, the weight parameters.
- Hyper parameters:
  - Polynomial regression, the order of polynomial.
  - Principal component regression, the number of components.

Hyper-parameters and Model selection

- Model parameters:
  - (Logistic) regression, the regression coefficients.
  - LDA, the parameters of each Gaussian model.
  - Perceptron, the weight parameters.
- Hyper parameters:
  - Polynomial regression, the order of polynomial.
  - Principal component regression, the number of components.
  - KNN, the number of neighbors K.

Hyper-parameters and Model selection

- Model parameters:
  - (Logistic) regression, the regression coefficients.
  - LDA, the parameters of each Gaussian model.
  - Perceptron, the weight parameters.
- Hyper parameters:
  - Polynomial regression, the order of polynomial.
  - Principal component regression, the number of components.
  - KNN, the number of neighbors K.
- How to choose (tuning) the hyper-parameters?

Hyper-parameters and Model selection

- Model parameters:
  - (Logistic) regression, the regression coefficients.
  - LDA, the parameters of each Gaussian model.
  - Perceptron, the weight parameters.
- Hyper parameters:
  - Polynomial regression, the order of polynomial.
  - Principal component regression, the number of components.
  - KNN, the number of neighbors K.
- How to choose (tuning) the hyper-parameters?
- Choose the "best" model.

Model Evaluation and Selection

Evaluation

- Regression:
  - Mean Square Error (MSE): $\frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2$
  - Mean Absolute Error (MSA): $\frac{1}{N}\sum_{i=1}^{N}|\widehat{y}_i - y_i|$
  - R square and F-statistic
- Classification:
  - Accuracy

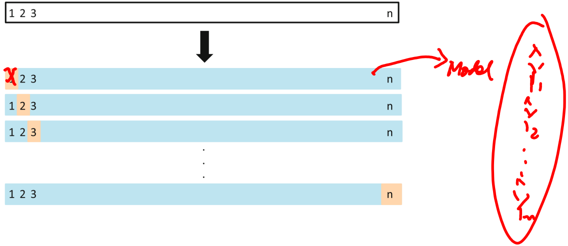
$$\frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{\widehat{y}_i \neq y_i\}$$

  - Confusion matrix
  - Kappa statistics

Leave One Out Cross Validation

- Suppose the sample size of the training data is N
- For each observation i
  - Leave the ith case out, and train the model with the rest.
  - Predict the ith case using the trained model.
- Calculate the average accuracy or other metrics.

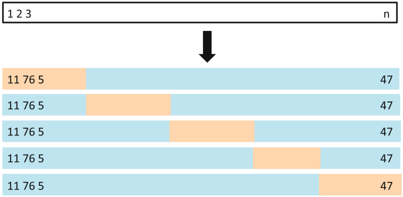
For $k = 1 \dots k$



- Computational cost is high.

k-Fold Cross Validation

- Randomly divide the training data set into k even parts.
- For each group i,
    - Excluding the ith group, train the model with the rest.
    - Predict the ith group using the trained model.
- Calculate the average performance.



- Provide a way to estimate the standard deviation of the estimation of performance.

Procedure

- Spliting the data into three parts: training set, validation set and testing set.
- Fit and validate different models with training/validatoin set and the corresponding validation method.
- Select the model with the "best" performance.
- Train the "best" model by using train set + validation set.
- Estimate the performance of the model by using the testing set.