INT104 ARTIFICIAL INTELLIGENCE

Review II

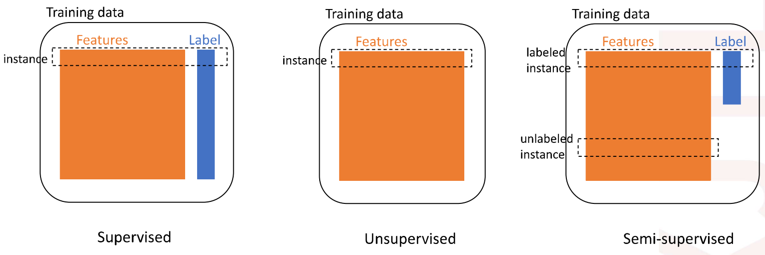
Fang Kang
Fang.kang@xjtlu.edu.cn

Xi'an Jiaotong-Liverpool University
西交利物浦大学

CONTENT

- Supervised methods
  - Classification and Regression
  - SVM
  - Decision Tree
  - Random Forest
- Unsupervised methods
  - K-means
  - Hierarchical clustering
  - GMM

Supervised vs. unsupervised

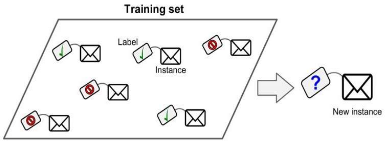


Supervised

Classification

Classification: Classification algorithms find a function that determines which category the input data belongs to.

Binary Classification is a supervised learning algorithm that classifies new observations into one of two classes.



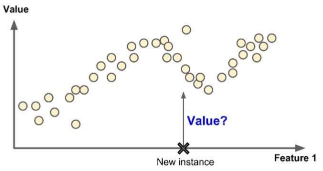
Multiclass/Multilabel Classification

- **Multiclass classification** refers to classification tasks that can distinguish between more than two classes.
- **Multilabel classification** refers to classification system that outputs multiple binary tags.

Regression

Regression attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).

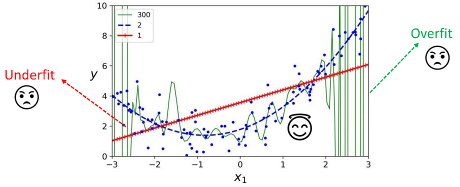
$$x \xrightarrow{f(x)} \hat{y} = f(x)$$



Learning Curves

$$\hat{y} = ax_2 + bx_1 + c$$
$$x_2 = x_1^2$$

If you perform high-degree Polynomial Regression, you will likely fit the training data much better than with plain Linear Regression. (Is high-degree polynomial always better?)

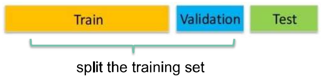


Bias: refers to the error from erroneous assumptions in the learning algorithm. (inability to capture the underlying patterns in the data).

Variance: refers an error from sensitivity to small fluctuations in the training data. (difference in fits between data sets)

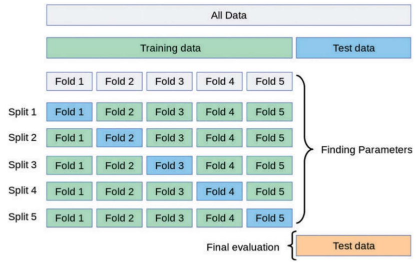
Cross Validation

- Train/test/validation split
- To avoid selecting the parameters that perform best on the test data but maybe not the parameters that generalize best, we can further split the training set into training fold and validation fold
- Can maximize the accuracy on the training data



- **Training fold**: used to fit the model
- **Validation fold**: used to estimate prediction error for model selection
- **Test set**: used for assessment of the prediction error of the final chosen model

K-fold Cross-Validation



All Data

Training data | Test data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Finding Parameters

Final evaluation | Test data

Confusion Matrix

		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}} \qquad \text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Support Vector Machine (SVM)

Linear SVM Classification

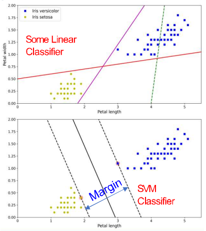
- Linear separability

- Fitting widest possible "street" between classes

Performs better with new data

- **Large Margin Classification**

- Margin, Support Vectors



Hard Margin SVM

All instances being off the street and on the right side

Soft Margin SVM

Allow margin violations

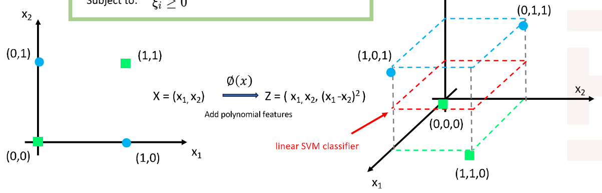
Support Vectors

- Decision boundary is not affected by more training instances
- It is determined by support vectors （instances located on the edge of street）

Nonlinear SVM Classification

Minimize: $\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i$ (Slack variable)

Subject to: $\xi_i \geq 0$



Nonlinear transformation $\emptyset(x)$: not only one form

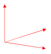
Cover's theorem: High-dimensional space is more likely to be linearly separable than in a low-dimensional space.

https://en.wikipedia.org/wiki/Cover's_theorem

Nonlinear SVM: Kernel Trick

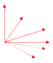
Input Space: dimension n

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$



High-dimensional Feature Space: dimension N >> n

$$\phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \phi_3(x) \\ \vdots \\ \phi_N(x) \end{bmatrix} \quad N \gg n$$



Expensive operation and requires large memory

$$K(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^T \boldsymbol{b} = [a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_n] \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

$$K(\phi(\boldsymbol{a}), \phi(\boldsymbol{b})) = \underbrace{\phi(\boldsymbol{a})^T \phi(\boldsymbol{b})} = [\phi_1(a) \quad \phi_2(a) \quad \phi_3(a) \quad \dots \quad \phi_N(a)] \begin{bmatrix} \phi_1(b) \\ \phi_2(b) \\ \phi_3(b) \\ \vdots \\ \phi_n(b) \end{bmatrix}$$

$$\phi(\boldsymbol{a})^T \phi(\boldsymbol{b}) = \text{function } (\boldsymbol{a}^T \boldsymbol{b})$$

Kernel Trick

Common kernels:

$$\boxed{\begin{aligned} \text{Linear: } K(\mathbf{a}, \mathbf{b}) &= \mathbf{a}^\top \mathbf{b} \\ \text{Polynomial: } K(\mathbf{a}, \mathbf{b}) &= (\gamma \mathbf{a}^\top \mathbf{b} + r)^d \\ \text{Gaussian Radial Basis Function: } K(\mathbf{a}, \mathbf{b}) &= \exp(-\gamma \|\mathbf{a} - \mathbf{b}\|^2) \\ \text{Sigmoid: } K(\mathbf{a}, \mathbf{b}) &= \tanh(\gamma \mathbf{a}^\top \mathbf{b} + r) \end{aligned}}$$

Universal approximator.
Corresponding feature space
$\phi(x)$ is infinite dimensional space

non-linearly separable data

⬇

infinite-dimensional space

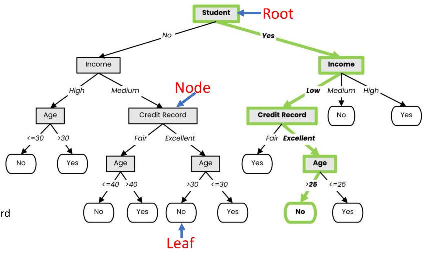
Decision Tree Definition

- A tree-like model that illustrates series of events leading to certain decisions
- Each node represents a test on an attribute and each branch is an outcome of that test

Who to loan?

- Not a student
- 45 years old
- Medium income
- Fair credit record
- ➢ Yes

- Student
- 27 years old
- Low income
- Excellent credit record
- ➢ No

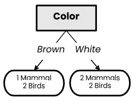


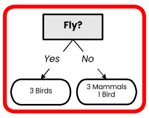
Depth: the length of the longest path from the root node to a leaf node

Best attribute = highest information gain

In practice, we compute $entropy(X)$ only once!

Does it fly?	Color	Class
No	Brown	**Mammal**
No	White	**Mammal**
Yes	Brown	**Bird**
Yes	White	**Bird**
No	White	**Mammal**
No	Brown	**Bird**
Yes	White	**Bird**





$$entropy\ (X) = -p_{\text{mammal}} \log_2 p_{\text{mammal}} - p_{\text{bird}} \log_2 p_{\text{bird}} = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} \approx 0.985$$

$$entropy\ (X_{color=brown}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} \approx 0.918 \qquad entropy\ (X_{color=white}) = 1$$

$$\boldsymbol{gain\ (X, color) = 0.985 - \frac{3}{7} \cdot 0.918 - \frac{4}{7} \cdot 1 \approx 0.020}$$

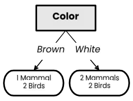
$$entropy\ (X_{fly=yes}) = 0 \qquad entropy\ (X_{fly=no}) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} \approx 0.811$$

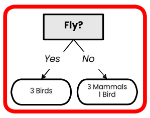
$$\boldsymbol{gain\ (X, fly) = 0.985 - \frac{3}{7} \cdot 0 - \frac{4}{7} \cdot 0.811 \approx \boxed{0.521}}$$

Best attribute = lowest Gini impurity

In practice, we compute $gini(X)$ only once!

Does it fly?	Color	Class
No	Brown	**Mammal**
No	White	**Mammal**
Yes	Brown	**Bird**
Yes	White	**Bird**
No	White	**Mammal**
No	Brown	**Bird**
Yes	White	**Bird**





$$gini\ (X_{color=brown}) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 \approx 0.444 \qquad gini\ (X_{color=white}) = 0.5$$

$$\boldsymbol{gini\ (X, color) = \frac{3}{7} \cdot 0.444 + \frac{4}{7} \cdot 0.5 \approx 0.476}$$

$$gini\ (X_{fly=yes}) = 0 \qquad gini\ (X_{fly=no}) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 \approx 0.375$$

$$\boldsymbol{gini\ (X, fly) = \frac{3}{7} \cdot 0 + \frac{4}{7} \cdot 0.375 \approx \boxed{0.214}}$$

Ensemble Learning

Ensemble : A group of predictors

Voting Classifier

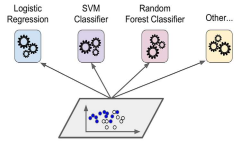


Figure 7-1. Training diverse classifiers

Hard Voting

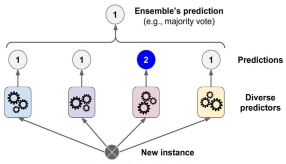
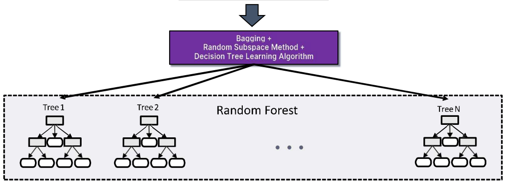


Figure 7-2. Hard voting classifier predictions

Random Forests

Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	**Apple**
121	Orange	Rough	3.9	**Orange**
85	Red	Smooth	3.3	**Apple**
101	Orange	Smooth	3.7	**Orange**
111	Green	Rough	3.5	**Apple**
...				
117	Red	Rough	3.4	**Orange**



Ensemble method

- **Random Forests** are one of the most common examples of ensemble learning.
- Other commonly-used ensemble methods:
  - ➢ Bagging: multiple models on random subsets of data samples.
  - ➢ Random Subspace Method: multiple models on random subsets of features.
  - ➢ Boosting: train models iteratively, while making the current model focus on the mistakes of the previous ones by increasing the weight of misclassified samples.
  - ➢ Stacking: instead of using hard voting to aggregate the predictions of all predictors in an ensemble, train a model to perform this aggregation.

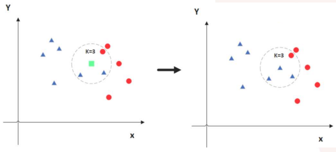
K Nearest Neighbors (KNN)

As in the general problem of classification, we have a set of data points for which we know the correct class labels

When we get a new data point, we compare it to each of our existing data points and find similarity

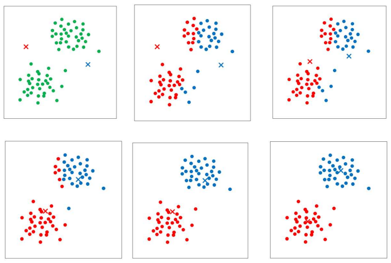
Take the most similar k data points (k nearest neighbours)

From these k data points, take the majority vote of their labels.
The winning label is the label / class of the new data point



Unsupervised

K-means clustering algorithm



Euclidean distance $d(p,q)=\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\cdots+(p_n-q_n)^2}.$

Goal: Assign all data points to k clusters

Step 1: Pick k *random* initial cluster centroids

Step 2: Paint the data points that are closer to red centroid red, and those closer to blue centroid blue

Step 3: Update the positions of centroids
Red centroid := average of current red points
Blue centroid := average of current blue points

Repeat

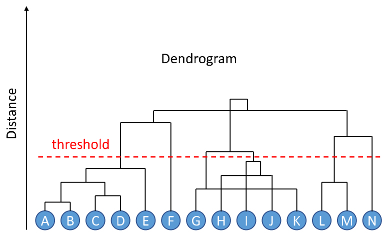
Until no more pointes need to be repainted, i.e., the centroids no longer change

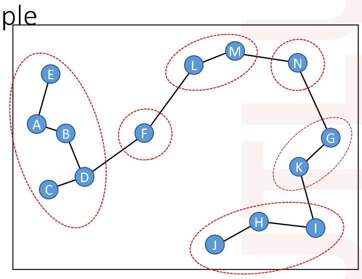
Clustering is done

Hierarchical clustering

- Hierarchical Clustering is a set of clustering methods that aim at building a hierarchy of clusters
  - ➢ A cluster is composed of smaller clusters
- There are two strategies for building the hierarchy of clusters:
  - ➢ Agglomerative (bottom-up): we start with each point in its own cluster and we merge pairs of clusters until only one cluster is formed.
  - ➢ Divisive (top-down): we start with a single cluster containing the entire set of points and we recursively split until each point is in its own cluster.
- The most popular strategy in practical use is bottom-up (agglomerative)!

Agglomerative clustering example

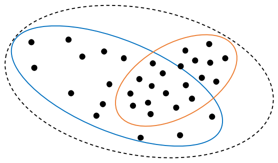




Gaussian mixture model (GMM)

K-means make *hard* assignments to data points: $x^{(i)}$ must belong to one of the clusters $1,2,\cdots,K$

Sometimes, one data point can belong to multiple clusters



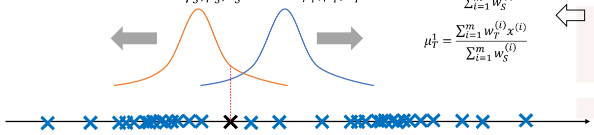
- Clusters may overlap
- Hard assignment may be simplistic
- Need a *soft* assignment:
  data points belong to clusters with different ***probabilities***

Demonstration with k = 2, 1-D Gaussian

Maximize likelihood of the whole data: $\mathcal{L}(\theta) = p(X|\Theta) = \prod_{i=1}^{m} p\left(x^{(i)}, z^{(i)}|\Theta\right) = \prod_{i=1}^{m}\sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}|\Theta)}{Q_i(z^{(i)})}$

Random initialization

Salmon: $\phi_S^0, \mu_S^0, \sigma_S^0$ Tuna: $\phi_T^0, \mu_T^0, \sigma_T^0$



These points tend to have larger $w_S^{(i)}$

$w_S^{(i)} = 0.6$

$w_T^{(i)} = 0.4$

E-step

These points tend to have larger $w_T^{(i)}$

$$\mu_S^1 = \frac{\sum_{i=1}^m w_S^{(i)} x^{(i)}}{\sum_{i=1}^m w_S^{(i)}}$$

$$\mu_T^1 = \frac{\sum_{i=1}^m w_T^{(i)} x^{(i)}}{\sum_{i=1}^m w_S^{(i)}}$$

M-step