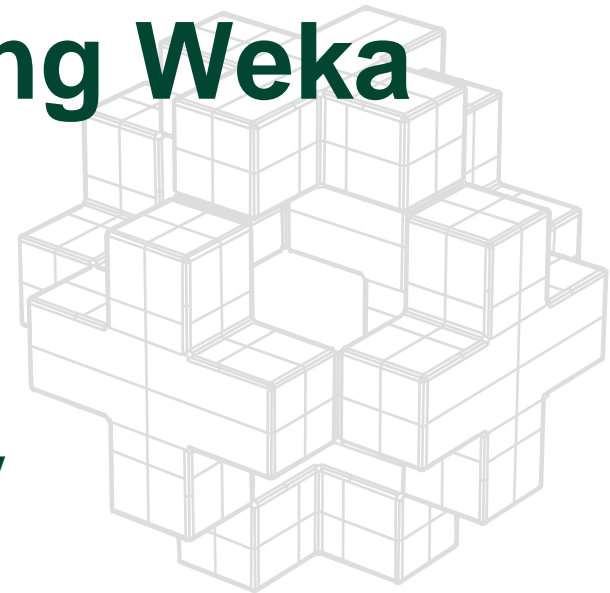




Classification using Weka

2009-09-21

Byoung-Hee Kim
Bilintelligence Lab, CSE,
Seoul National University





Agenda



❖ Introduction to Weka

❖ Classification

- General Concept
- We will see two popular classifiers
 - Neural Networks, Decision Trees
- Evaluation criteria

❖ How to classify data using Weka?

- Data format: ARFF
- Neural Networks, Decision Trees in Weka
- Using Experimenter (optional)

❖ Weka 3: Data Mining Software in Java

- Weka is a collection of machine learning algorithms for data mining tasks
- What you can do with weka?
 - data pre-processing, **classification**, regression, clustering, association rules, and visualization
- Weka is an open source software issued under the GNU General Public License
- How to get?
<http://www.cs.waikato.ac.nz/ml/weka/> or just type 'Weka' in google.
- Demo



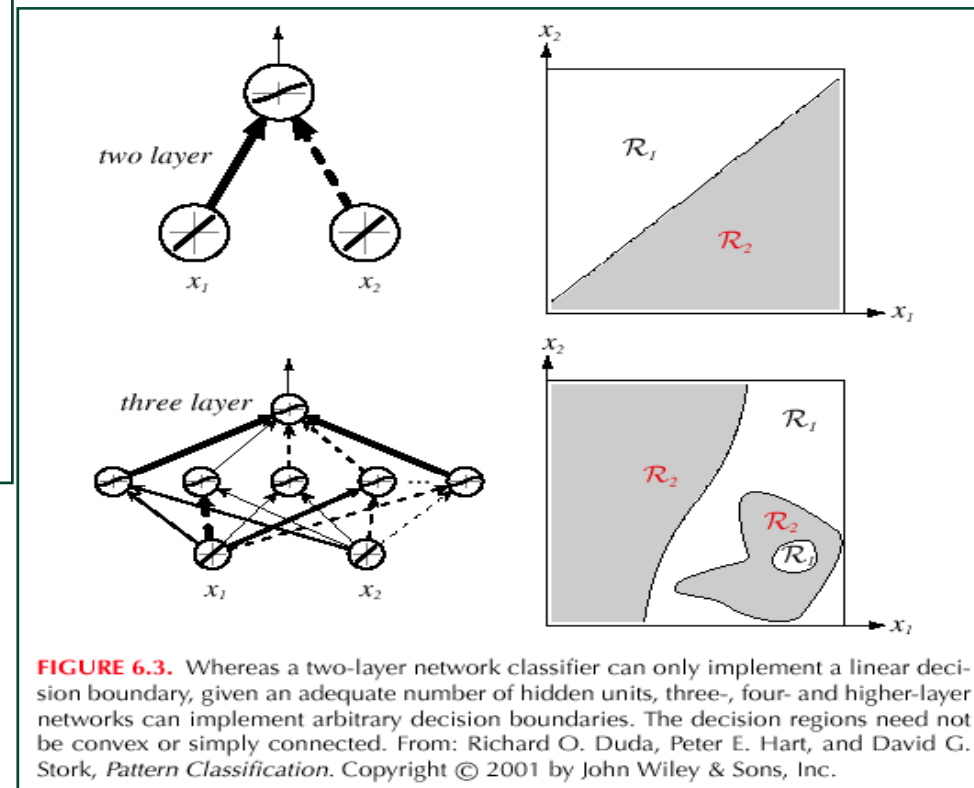
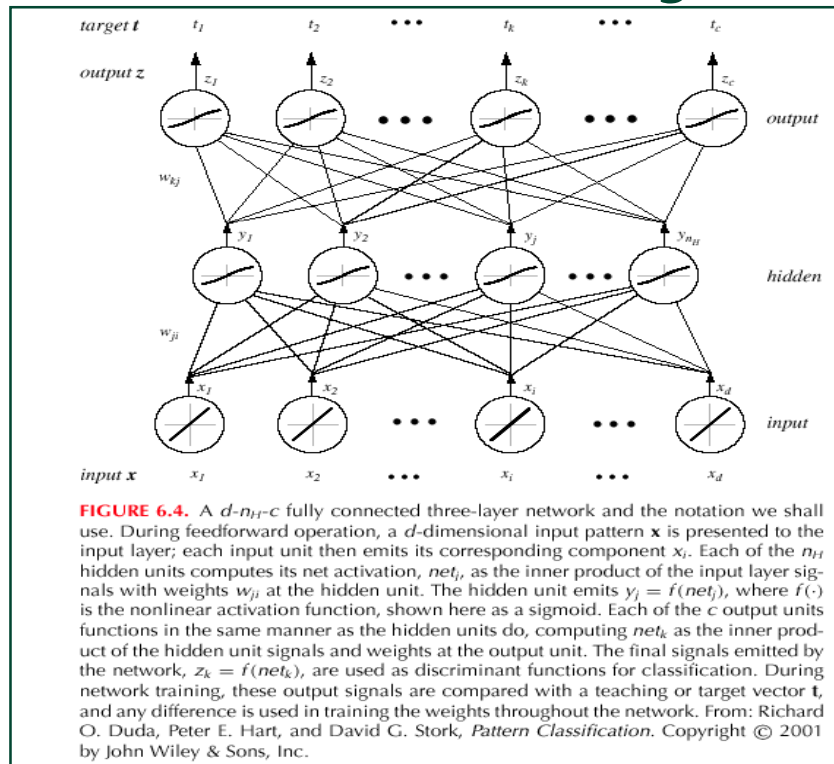
Concept of Classification



- ❖ Feature or attribute
- ❖ Training or learning
- ❖ Data: training set, test set
- ❖ Supervised learning

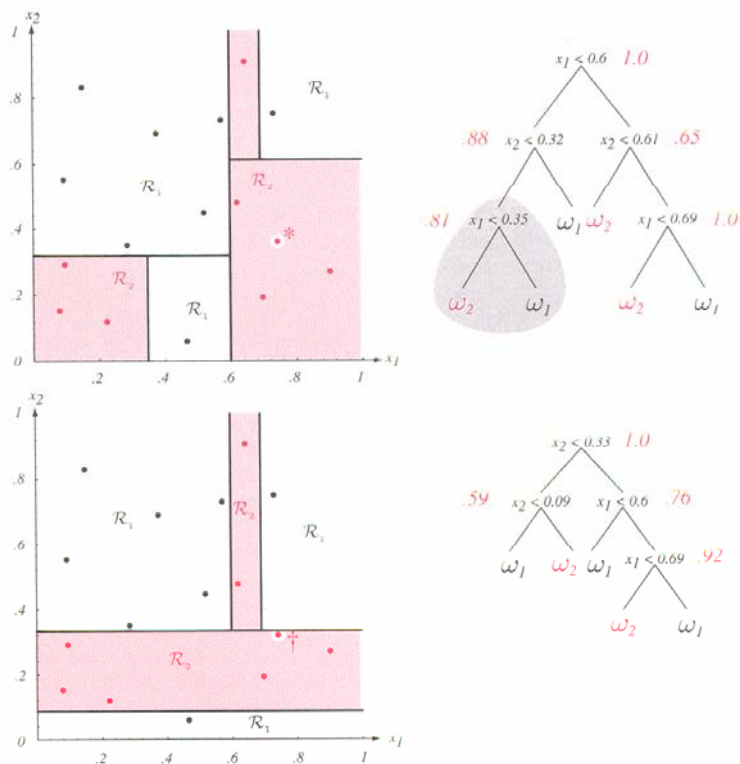
Neural Networks

❖ MLP (Multilayer Perceptron)



Decision Trees

❖ ID3, J48 (C4.0)



Training data and associated (unpruned) tree are shown at the top. The entropy impurity at nonterminal nodes is shown in red and the impurity at each leaf is 0. If the single training point marked * were instead slightly lower (marked †), the resulting tree and decision regions would differ significantly, as shown at the bottom.

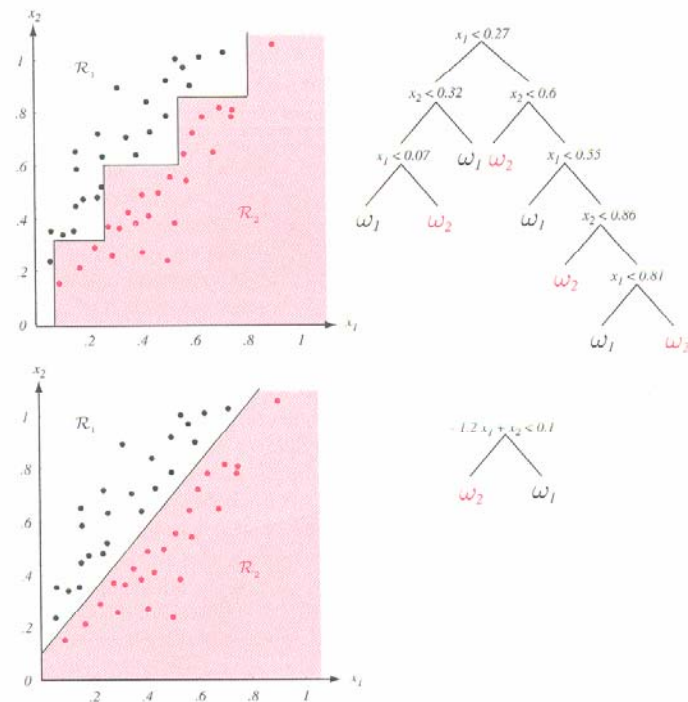


FIGURE 8.5. If the class of node decisions does not match the form of the training data, a very complicated decision tree will result, as shown at the top. Here decisions are parallel to the axes while in fact the data is better split by boundaries along another direction. If, however, “proper” decision forms are used (here, linear combinations of the features), the tree can be quite simple, as shown at the bottom.

❖ Data: Iris data set

❖ Data format for Weka

- [.ARFF](#)
- Header + data (CSV type)
- You need to add a header to usual text file

❖ Classifiers

- Neural Networks (Multilayer Perceptron)
- Decision Tree (J48)

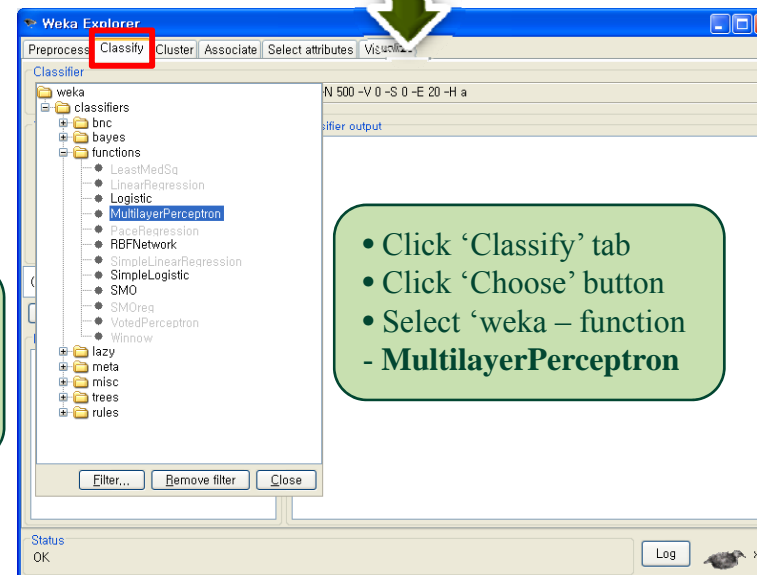
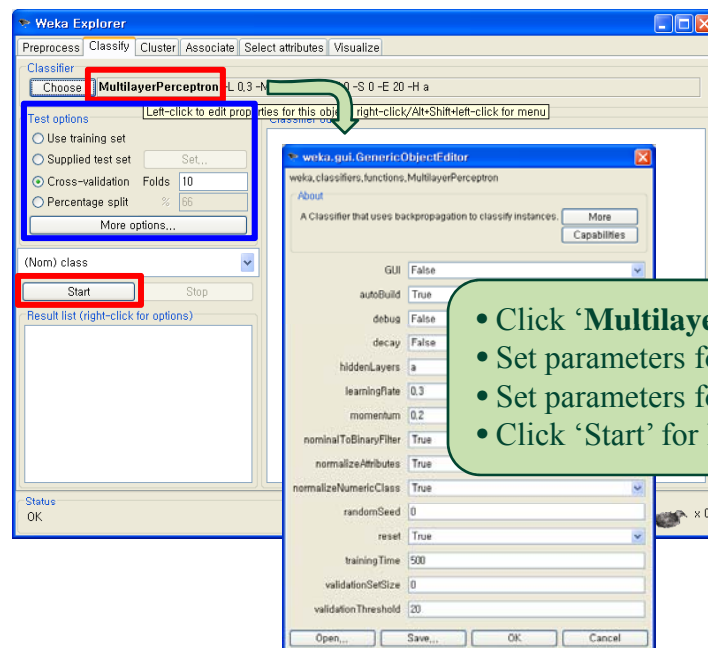
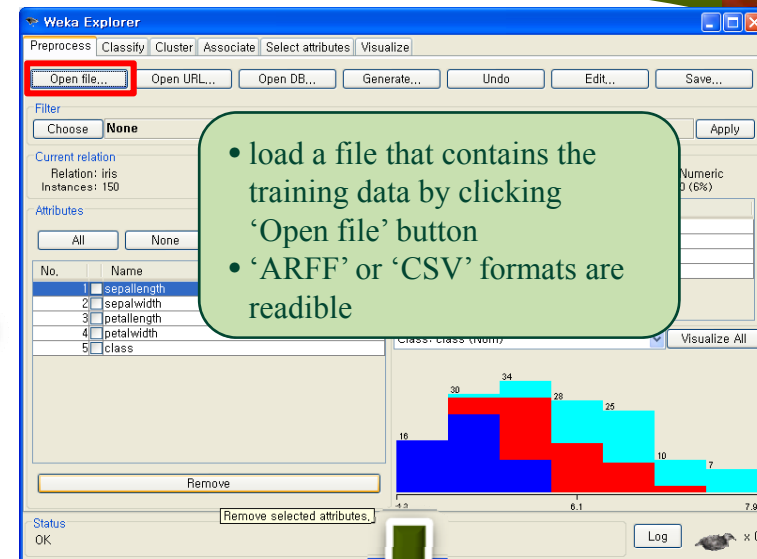
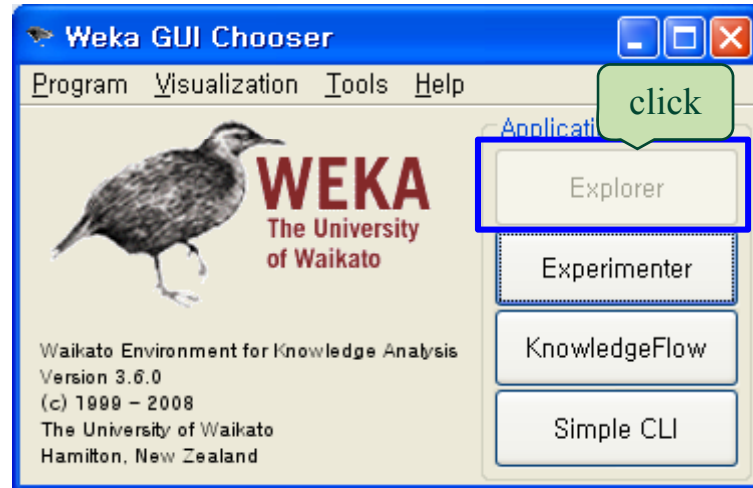
Iris data set

❖ Data

- 150 instances (50 instances * 3 classes)
- 4 numeric, predictive attributes and the class
 - sepal length/width in cm
 - petal length/width in cm
- Class labels – species of *iris* flowers
 - Iris setosa
 - Iris virginica
 - Iris versicolor
- Note: classical, but very famous dataset, first used by Fisher (1936)



Neural Networks in Weka





Some Notes on the Parameter Setting



❖ **Parameter Setting = Car Tuning**

- need much experience or many times of trial
- you may get worse results if you are unlucky

❖ **Multilayer Perceptron (MLP)**

- Main parameters: hiddenLayers, learningRate, momentum, training time, randomSeed

❖ **J48**

- many parameters are related to the size of the result tree, i.e. pruning

Running MultilayerPerceptron in Weka

The screenshot shows the Weka Explorer window with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'MultilayerPerceptron' with parameters '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a'. The 'Test options' section on the left is highlighted with a green box, showing 'Cross-validation' selected with 'Folds' set to 5. The 'Classifier output' pane on the right is highlighted with a red box, displaying the following results:

Time taken to build model: 0.63 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	144	96	%
Incorrectly Classified Instances	6	4	%
Kappa statistic	0.94		
Mean absolute error	0.0376		
Root mean squared error	0.1448		
Relative absolute error	8.4538 %		
Root relative squared error	30.7184 %		
Total Number of Instances	150		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC
1	1	0	1	1	1	1
0.9	0.9	0.01	0.978	0.9	0.938	0
0.98	0.98	0.05	0.907	0.98	0.942	0
Weighted Avg.	0.96	0.02	0.962	0.96	0.96	0

=== Confusion Matrix ===

a	b	c	<-- classified as
50	0	0	a = Iris-setosa
0	45	5	b = Iris-versicolor
0	1	49	c = Iris-virginica

The status bar at the bottom shows 'Status OK' and a 'Log' button.

You need to understand the meaning of these options

You need to understand this result screen



How to Evaluate the Performance? (1/2)



- ❖ Usually, build a Confusion Matrix out of given data
- ❖ Evaluation Metrics
 - Accuracy
 - Precision
 - Recall
 - Many other metrics: F-measure, Kappa score, etc.
- ❖ For fair evaluation, the 'cross-validation' scheme is used

How to Evaluate the Performance? (2/2)

❖ Confusion Matrix

<div>Real Prediction</div>	True	False	
Positive	TP	FP	All with positive Test
Negative	FN	TN	All with Negative Test
	All with Disease	All without Disease	Everyone

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

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

As **recall** ↑ **precision** ↓

conversely:

As **recall** ↓ **precision** ↑

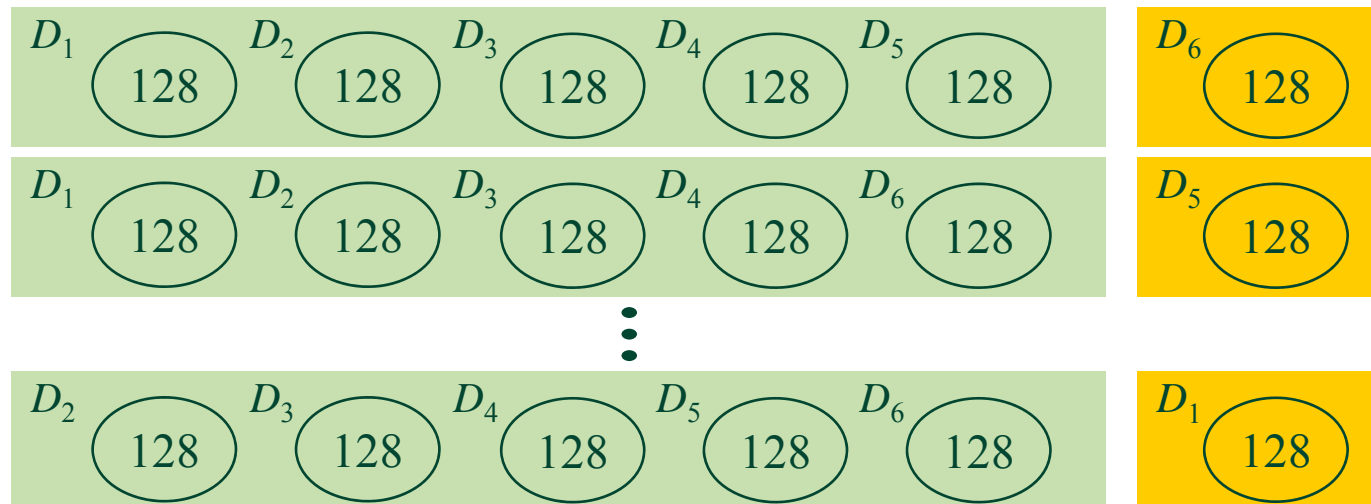
Cross Validation (1/2)

❖ K-fold Cross Validation

- The data set is randomly divided into k subsets.
- One of the k subsets is used as the 'test set' and the other $k-1$ subsets are put together to form a 'training set'.

$$Error = \frac{1}{k} \sum_{i=1}^k Error_i$$

6-fold cross validation





How to Show Classification Results?



❖ Cross validation and Confusion Matrix

- At least 10 runs for your k value.

Run	Test Error
1	
2	
...	...
10	
Average	

- Show the confusion matrix for the best result of your experiments.

Using Experimenter in Weka

❖ Tool for 'Batch' experiments

The diagram illustrates the workflow for using the Weka Experimenter tool, consisting of three main steps:

- Step 1: Weka GUI Chooser**
 - Click on the **Experimenter** button.
- Step 2: Weka Experiment Environment (Setup)**
 - Click **'New'** to create a new experiment.
 - Set experiment type/iteration control.
 - Set datasets / algorithms.
- Step 3: Weka Experiment Environment (Run/Analyse)**
 - Select **'Run'** tab and click **'Start'**.
 - If it has finished successfully, click **'Analyse'**.

The final state shows the **Weka Experiment Environment** with the **Analyse** tab selected, displaying the test results.