HPI Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

WEKA

Overview over WEKA

- Waikato Environment for Knowledge Analysis
- Open Source Java library (GPL)
- Since 1997 in Java
- Includes CLI and GUI

- Provides
  - **Preprocessors, classifiers**
  - Association rule miner, clusterer

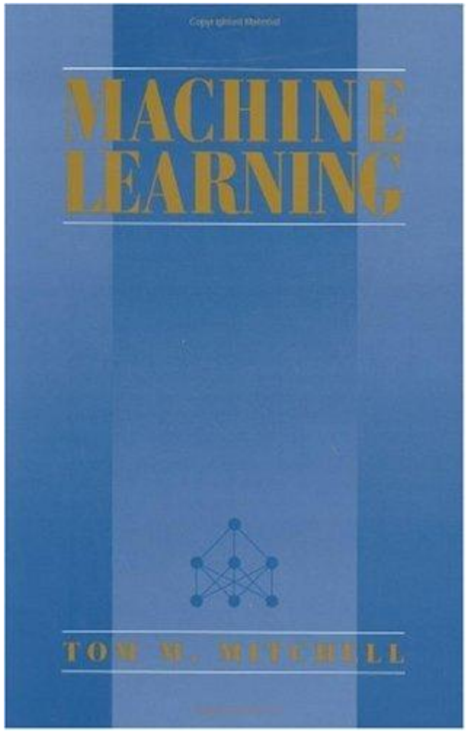
General Machine Learning

- Classification
  - Assign a label or value to a given data instance
- Learn rules from a set of train instances
- Apply them to new instances to classify them

- **Supervised**
  - The train instances have to be labeled manually
  - Most classifiers
- Unsupervised
  - No labels needed, mostly statistical data
  - Cluster algorithms, SVD

Literature

- Machine Learning, Tom Mitchell, McGraw Hill, 1997
- Slides: http://www.cs.cmu.edu/~tom/mlbook-chapter-slides.html



Exemplary Data Matrix

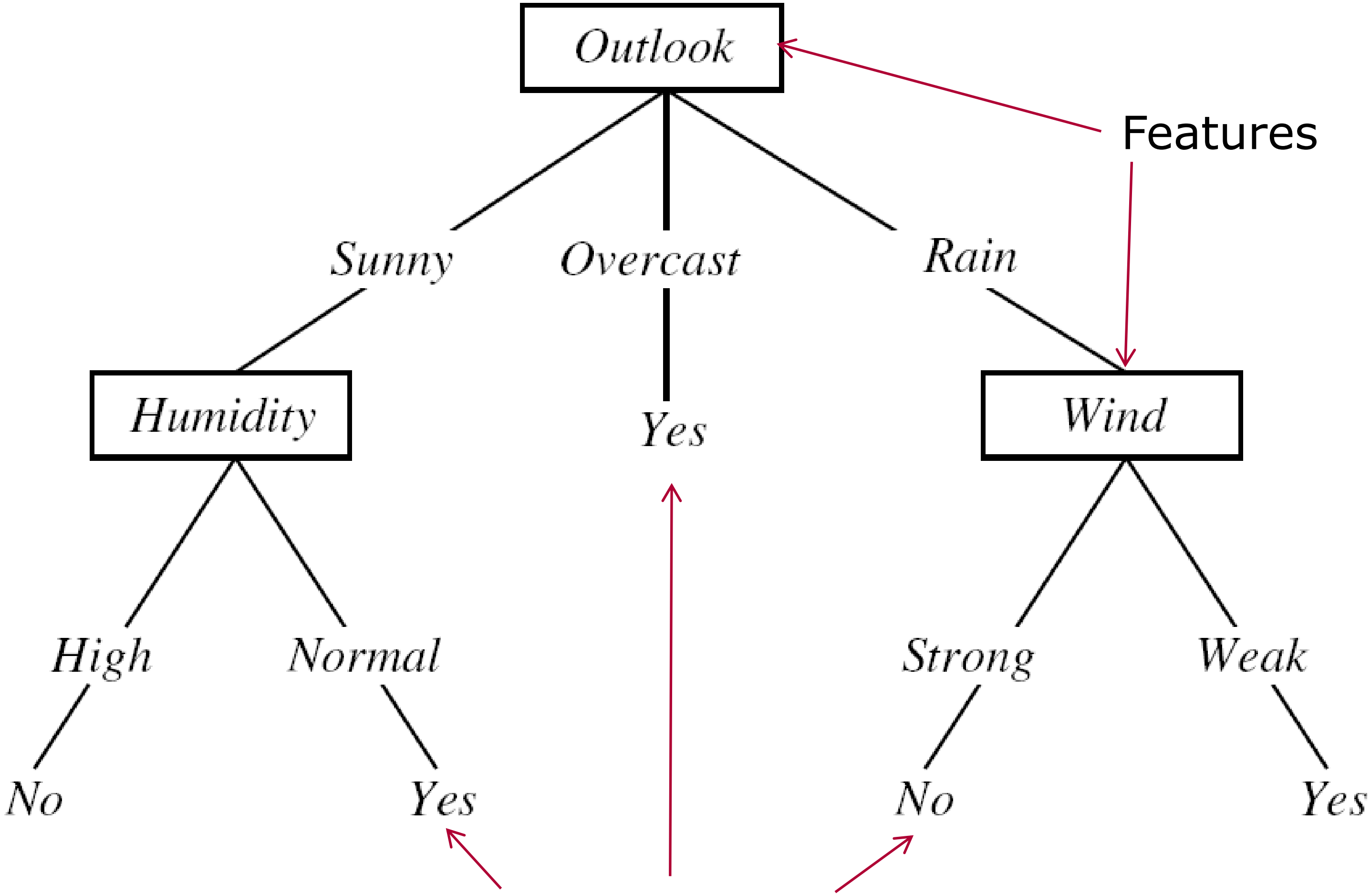
Instance ID

4 Features

Binary class label

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Learned Classifier: Decision Tree

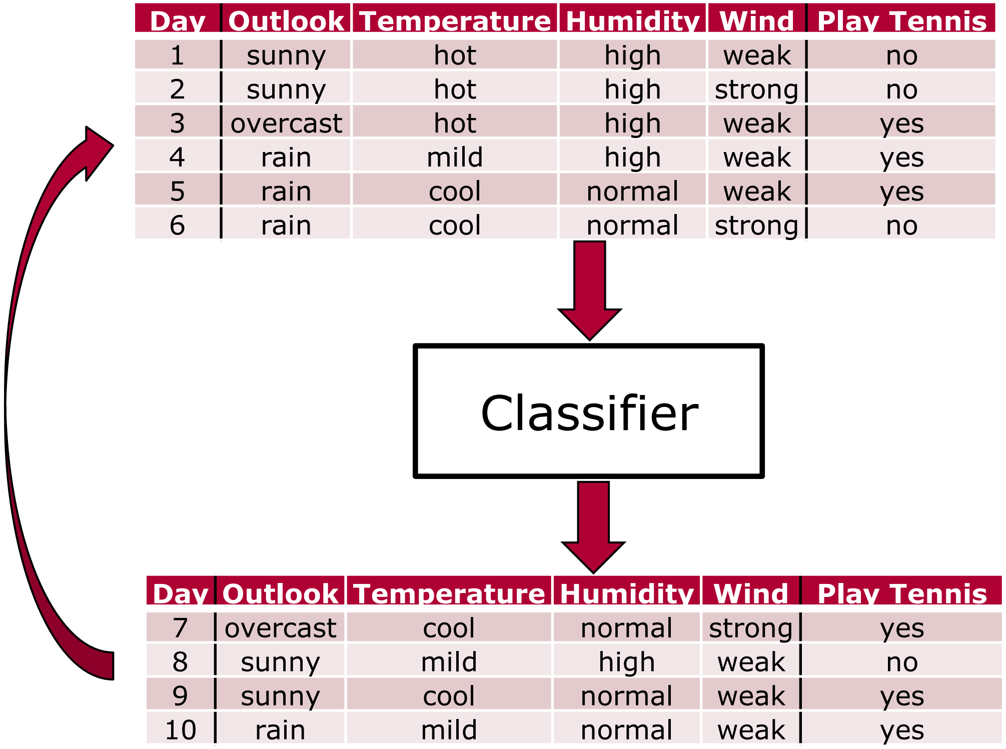


Features

Binary class label

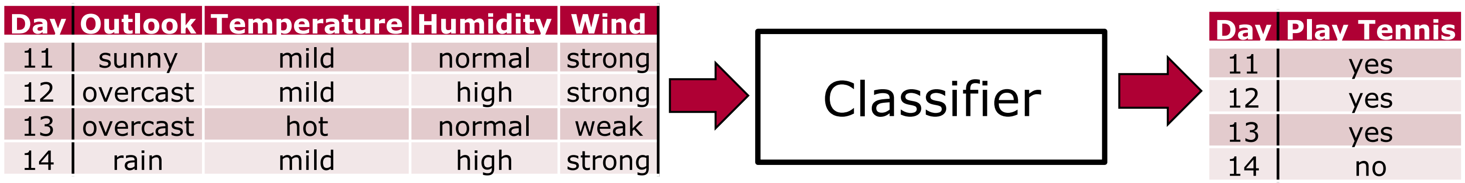
Normal Workflow

- Gather training instances
- Learn
- Validate
- Apply



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes



Day	Outlook	Temperature	Humidity	Wind
11	sunny	mild	normal	strong
12	overcast	mild	high	strong
13	overcast	hot	normal	weak
14	rain	mild	high	strong

Day	Play Tennis
11	yes
12	yes
13	yes
14	no

Gather training instances

- All classifiers are dumb!
  - □ They don’t abstract anything
  - □ Mostly statistical
  - □ Good results only if trained with data of same characteristics
- Need to have a representative training set
  - □ Large variety of instances
  - □ Balance class representatives!
  - □ Needs good counter-examples
  - □ Remove outliers!
- [Some E-Mail classification examples]
  - □ Number of instances (balance spam and ham)
  - □ Length (most spam is short, need short ham and long spam)
  - □ Words (price often in spam but also in some ham)

Learn

- Black box for us
- If you are interested -> IfI

- However important to know some characteristics
  - Support for floating point class values?
  - How much data is needed at minimum?
  - How much data at maximum?
  - Is the order important? Randomize?

Validate

- Tests how well the classifier performs on training data
- If it is near baseline (=bad)
  - Features are insufficient
  - Data is too noisy
  - Bad type of classifier
  - Too many data instances

- If it is near perfect (=probably bad)
  - Overfitted
  - Too few data instances
  - Too clean data (removed too many “outliers”)
- Tenfold cross-validation is state of the art (ten times as slow!)

Apply

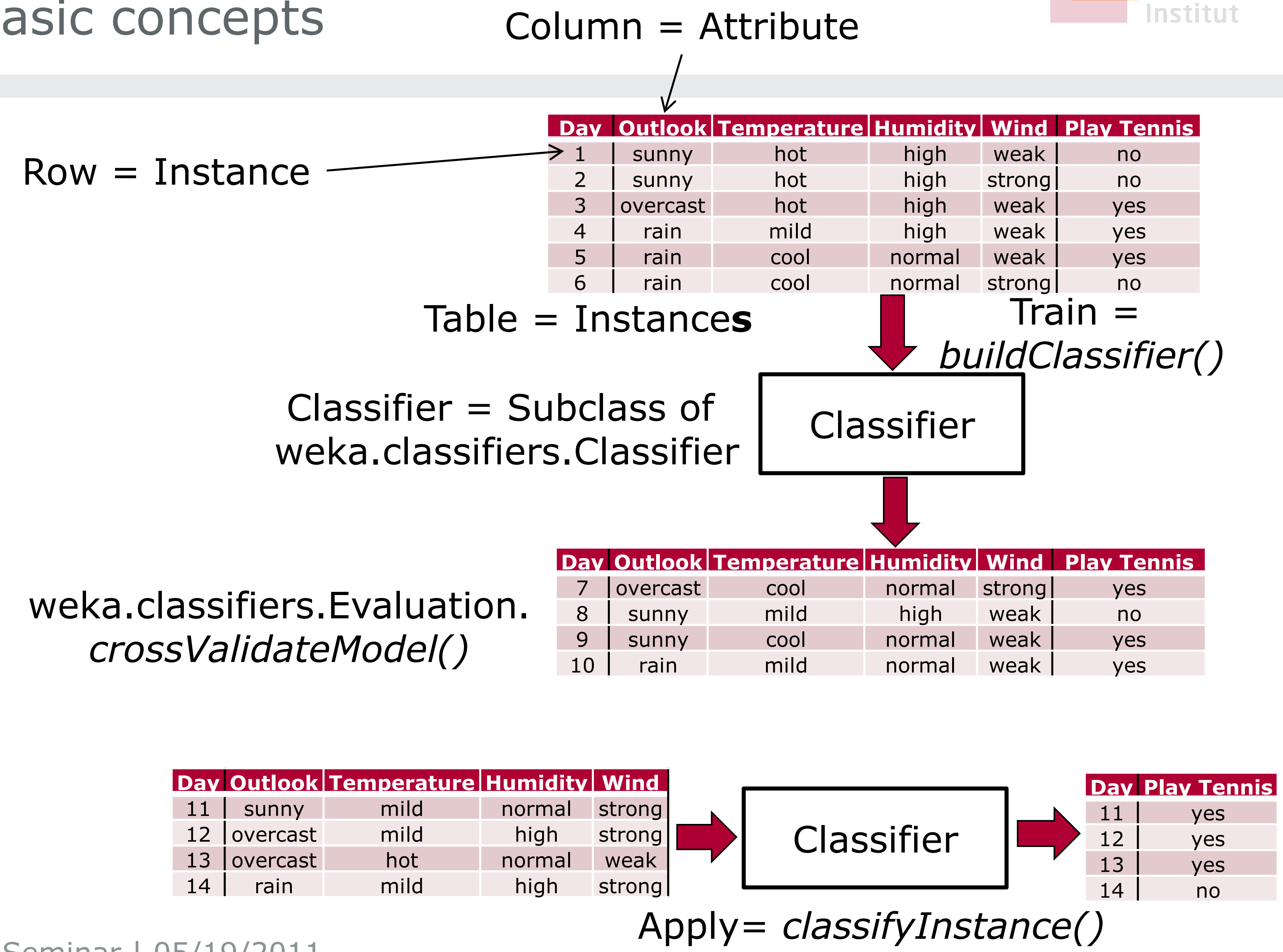
- Again black box

- Check random samples
- If results are obviously wrong
  - Most probably bad test data
  - Bug

Finally WEKA

- Great tutorials and wiki on official site
- Book (at the chair)
- Good Javadoc
- When having troubles, commit & post request @ mailing list

Basic concepts



Column = Attribute

Row = Instance

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no

Table = Instance**s**

Train = *buildClassifier()*

Classifier = Subclass of weka.classifiers.Classifier

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes

weka.classifiers.Evaluation.
crossValidateModel()

Day	Outlook	Temperature	Humidity	Wind
11	sunny	mild	normal	strong
12	overcast	mild	high	strong
13	overcast	hot	normal	weak
14	rain	mild	high	strong

Day	Play Tennis
11	yes
12	yes
13	yes
14	no

Apply= *classifyInstance()*

Source Code: Create Instances

```
// 1. set up attributes
FastVector atts = new FastVector();
atts.addElement(new Attribute("Outlook"));
atts.addElement(new Attribute("Temperature")); ...

FastVector classVal = new FastVector(2);
classVal.addElement("yes");
classVal.addElement("no");
atts.addElement(new Attribute("Play Tennis", classVal));

// 2. create Instances object
Instances trainInstances = new Instances("Tennis Data", atts, 0);

// 3. fill with data
trainInstances.add(new Instance(...));
```

Source Code: Create Instance

```
Instance instance = new Instance(attributes.length + 1);

// set value for first attribute
instance.setValue(0, 42);
// or
instance.setValue(attributes[1], 42);

instance.setClassValue("yes");
```

Source Code: Train Classifier

```
Classifier classifier =
    Classifier.forName("weka.classifiers.bayes.NaiveBayes", new String[0]);
// or
Classifier classifier = new J48();

classifier.buildClassifier(trainInstances);
```

Source Code: Cross Validation

```
// tenfold cross validation
Evaluation evaluation = new Evaluation(trainInstances);
evaluation.crossValidateModel(classifier, trainInstances, 10,
         trainInstances.getRandomNumberGenerator(1));

System.out.println(evaluation.toSummaryString());
System.out.println(evaluation.toMatrixString());
```

Source Code: Apply Classifier

```
double classValue = classifier.classifyInstance(instance);

double[] distribution = classifier.distributionForInstance(instance);
```

Weka Classifier

- Naïve Bayes - weka.classifiers.bayes.NaiveBayes
  - Fast, good starting point
- Support Vector Machine – weka.classifiers.functions.SMO
  - Slow but precise
- Decision Tree - weka.classifiers.trees.J48
  - Easy interpretation and may yield interesting insights
- Regression - weka.classifiers.functions.LinearRegression
  - Handles floating point classes

- Many more and some combinations
  - Experimentation is necessary
  - Share insights via mailing lists