Changelog: 14 Oct 2016, 30 Oct 2016

Basic Concepts

Weka Workbench and its terminology

Lecture 2 Part 2

Marina Santini

Acknowledgements
Slides borrowed and adapted from:
Data Mining by I. H. Witten, E. Frank and M. A. Hall

Outline

- Concepts, instances, attributes
- How to prepare the input: ARFF, attributes, missing values, getting to know data

Getting ready for the in-class lab !

Terminology

- Components of the input:
  - Concepts: kinds of things that can be learned
    - Ex: different types of iris flowers
  - Instances: the individual, independent examples of a concept
    - Ex: description of each iris flower
  - Attributes: measuring aspects of an instance
    - Ex: Sepal length, petal width, etc.

What's a concept?

- Different style of learning: classification, association, clustering regression etc.
- Concept: thing to be learned
- Concept description: output of learning scheme

Classification learning

- Example problems: weather data, contact lenses, irises, labor negotiations
- Classification learning is *supervised*
  - Scheme is provided with actual outcome
- Outcome is called the *class* of the example
- Measure success on fresh data for which class labels are known (*test data*)

Association learning

- Can be applied if no class is specified and any kind of structure is considered "interesting"
- Difference to classification learning:
  - Can predict any attribute's value, not just the class, and more than one attribute's value at a time
  - Hence: far more association rules than classification rules
  - Thus: constraints are necessary
    - Minimum coverage and minimum accuracy

Clustering

- Finding groups of items that are similar
- Clustering is *unsupervised*
  - The class of an example is not known

	Sepal length	Sepal width	Petal length	Petal width	Type
1	5.1	3.5	1.4	0.2	Iris setosa
2	4.9	3.0	1.4	0.2	Iris setosa
...					
51	7.0	3.2	4.7	1.4	Iris versicolor
52	6.4	3.2	4.5	1.5	Iris versicolor
...					
101	6.3	3.3	6.0	2.5	Iris virginica
102	5.8	2.7	5.1	1.9	Iris virginica
...					

Numeric prediction

- Variant of classification learning where "class" is numeric (also called "regression")
- Learning is supervised
  - Scheme is being provided with target value
- Measure success on test data

Outlook	Temperature	Humidity	Windy	Play-time
Sunny	Hot	High	False	5
Sunny	Hot	High	True	0
Overcast	Hot	High	False	55
Rainy	Mild	Normal	False	40
...	...	...	...	...

What's in an example?

- Instance: specific type of example
  - Thing to be classified, associated, or clustered
  - Individual, independent example of target concept
  - Characterized by a predetermined set of attributes
- Input to learning scheme: set of instances/ dataset
  - Represented as a single relation/flat file
- Rather restricted form of input
  - No relationships between objects
- Most common form in practical data mining

What's in an attribute?

- Each instance is described by a fixed predefined set of features, its "attributes"
- But: number of attributes may vary in practice
- Possible attribute types ("levels of measurement"):
  - *Nominal, ordinal, interval* and *ratio*

Nominal quantities

- Values are distinct symbols
  - Values themselves serve only as labels or names
  - *Nominal* comes from the Latin word for name
- Example: attribute "outlook" from weather data
  - Values: "sunny","overcast", and "rainy"
- No relation is implied among nominal values (no ordering or distance measure)
- Only equality tests can be performed

Ordinal quantities

- Impose order on values
- But: no distance between values defined
- Example:
  attribute "temperature" in weather data
  - Values: "hot" > "mild" > "cool"
- Note: addition and subtraction don't make sense
- Example rule:
  temperature < hot ⇒ play = yes
- Distinction between nominal and ordinal not always clear (e.g. attribute "outlook")

Interval quantities

- Interval quantities are not only ordered but measured in fixed and equal units
- Example 1: attribute “temperature” expressed in degrees Fahrenheit
- Example 2: attribute “year”
- Difference of two values makes sense
- Sum or product doesn’t make sense
  - Zero point is not defined!

Ratio quantities

- Ratio quantities are ones for which the measurement scheme defines a zero point
- Example: attribute “distance”
  - Distance between an object and itself is zero
- Ratio quantities are treated as real numbers
  - All mathematical operations are allowed
- But: is there an “inherently” defined zero point?
  - Answer depends on scientific knowledge (e.g. Fahrenheit knew no lower limit to temperature)

Attribute types used in practice

- Most schemes accommodate just two levels of measurement: nominal and ordinal
- Nominal attributes are also called “categorical”, ”enumerated”, or “discrete”
  - But: “enumerated” and “discrete” imply order
- Special case: dichotomy (“boolean” attribute)
- Ordinal attributes are called “numeric”, or “continuous”
  - But: “continuous” implies mathematical continuity

Preparing the input

- Denormalization is not the only issue
- Problem: different data sources (e.g. sales department, customer billing department, …)
  - Differences: styles of record keeping, conventions, time periods, data aggregation, primary keys, errors
  - Data must be assembled, integrated, cleaned up
  - “Data warehouse”: consistent point of access
- External data may be required (“overlay data”)
- Critical: type and level of data aggregation

The ARFF format

```
%
% ARFF file for weather data with some numeric features
%
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {true, false}
@attribute play? {yes, no}

@data
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
...
```


Additional attribute types

- ARFF supports *string* attributes:

```
@attribute description string
```

  - Similar to nominal attributes but list of values is not pre-specified
- It also supports *date* attributes:

```
@attribute today date
```

  - Uses the ISO-8601 combined date and time format *yyyy-MM-dd-THH:mm:ss*

Relational attributes

- Allow multi-instance problems to be represented in ARFF format
  - The value of a relational attribute is a *separate* set of instances

```
@attribute bag relational
       @attribute outlook { sunny, overcast, rainy }
       @attribute temperature numeric
       @attribute humidity numeric
       @attribute windy { true, false }
@end bag
```

  - Nested attribute block gives the structure of the referenced instances

Mulit-instance ARFF

```
%
% Multiple instance ARFF file for the weather data
%
@relation weather

@attribute bag_ID { 1, 2, 3, 4, 5, 6, 7 }
@attribute bag relational
       @attribute outlook {sunny, overcast, rainy}
       @attribute temperature numeric
       @attribute humidity numeric
       @attribute windy {true, false}
       @attribute play? {yes, no}
@end bag

@data
1, "sunny, 85, 85, false\nsunny, 80, 90, true", no
2, "overcast, 83, 86, false\nrainy, 70, 96, false", yes
...
```


Sparse data

- In some applications most attribute values in a dataset are zero
  - E.g.: word counts in a text categorization problem
- ARFF supports sparse data

```
0, 26, 0,  0, 0 ,0, 63, 0, 0, 0, "class A"
0,  0, 0, 42, 0, 0,  0, 0, 0, 0, "class B"
```

```
{1 26, 6 63, 10 "class A"}
{3 42, 10 "class B"}
```

- This also works for nominal attributes (where the first value corresponds to "zero")

Attribute types

- Interpretation of attribute types in ARFF depends on learning scheme
  - Numeric attributes are interpreted as
    - ordinal scales if less-than and greater-than are used
    - ratio scales if distance calculations are performed (normalization/standardization may be required)
  - Instance-based schemes define distance between nominal values (0 if values are equal, 1 otherwise)
- Integers in some given data file: nominal, ordinal, or ratio scale?

Missing values

- Frequently indicated by out-of-range entries
  - Types: unknown, unrecorded, irrelevant
  - Reasons:
    - malfunctioning equipment
    - changes in experimental design
    - collation of different datasets
    - measurement not possible
- Missing value may have significance in itself (e.g. missing test in a medical examination)
  - Most schemes assume that is not the case: "missing" may need to be coded as additional value

Inaccurate values

- Reason: data has not been collected for mining it
- Result: errors and omissions that don't affect original purpose of data (e.g. age of customer)
- Typographical errors in nominal attributes ⇒ values need to be checked for consistency
- Typographical and measurement errors in numeric attributes ⇒ outliers need to be identified
- Errors may be deliberate (e.g. wrong zip codes)
- Other problems: duplicates, stale data

Getting to know the data

- Simple visualization tools are very useful
  - Nominal attributes: histograms (Distribution consistent with background knowledge?)
  - Numeric attributes: graphs (Any obvious outliers?)
- 2-D and 3-D plots show dependencies
- Need to consult domain experts
- Too much data to inspect? Take a sample!

Machine Learning in Practice

Start Loop

1. **Understand the domain**
2. **Data pre-processing**
3. **Learning models**
4. **Interpreting results**

End Loop

The end