Lecture 2: Machine Learning Concepts

COMP90049
Introduction to Machine Learning

Semester 2, 2020

Hadi Khorshidi, CIS

The presentation adapted from the slides prepared by Lea Frermann, CIS

Basics of ML: Instances, Attributes and Learning Paradigms

Roadmap

Last lecture

- Warm-up
- Housekeeping COMP90049
- Machine Learning

This lecture

- Terminology
- Basic concepts: instances, attributes and learning paradigms
- Python demo

Typical Workflow

https://commons.wikimedia.org/wiki/File:CRISP-DM_Process_Diagram.png

Typical Workflow

1. Identify task, collect data
2. Choose a good data representation (feature engineering)
3. Pick a suitable model and learning algorithm
4. Train the model
5. Evaluate your model (on a separate test data sets)
6. probably go back to previous steps.

Terminology

- The input to a machine learning system consists of:
  - **Instances**: the individual, independent examples of a concept

    *also known as **exemplars***
  - **Attributes**: measuring aspects of an instance

    *also known as **features***
  - **Concepts**: things that we aim to learn

  *generally in the form of **labels** or **classes***

Example: `weather.nominal` Dataset

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
⋮	⋮	⋮	⋮	⋮

Example: `weather.nominal` Dataset

	Outlook	Temperature	Humidity	Windy	Play
INSTANCE$_1$	sunny	hot	high	FALSE	no
INSTANCE$_2$	sunny	hot	high	TRUE	no
	overcast	hot	high	FALSE	yes
	rainy	mild	high	FALSE	yes
	rainy	cool	normal	FALSE	yes
	rainy	cool	normal	TRUE	no
	⋮	⋮	⋮	⋮	⋮

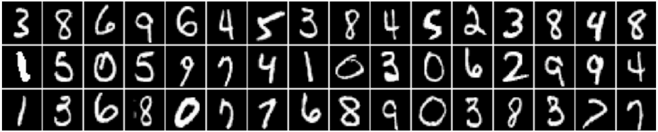
Example: `weather.nominal` Dataset

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
⋮	⋮	⋮	⋮	⋮

ATTRIBUTE$_1$

ATTRIBUTE$_2$

Quiz I



The MNIST digit classification data set

- How many **instances** do you see in this picture?
- What are these instances?
- How many **features** does each instance have?
- What could these features be?

A Word on Supervision

- **Supervised** methods have prior knowledge of a closed set of classes and set out to discover and categorise new instances according to those classes
- **Unsupervised** do not have access to an invertory of classes, and instead discover groups of ‘similar’ examples in a given dataset

THE UNIVERSITY OF MELBOURNE

What's a Concept?

- Styles of "concepts" that we aim to learn:
    - Classification learning:

        predicting a discrete class
    - Clustering:

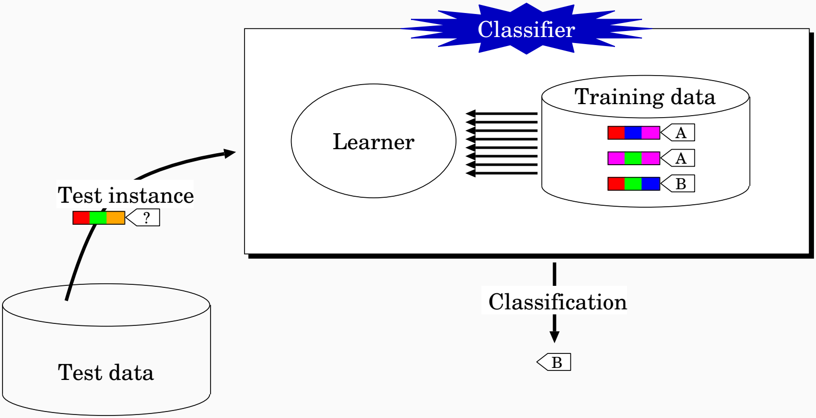
        grouping similar instances into clusters
    - Regression:

        predicting a numeric quantity
    - Association learning:

        detecting associations between attribute values

Classification Learning

- Scheme is provided with actual outcome or **class**
- The learning algorithm is provided with a set of classified **training data**
- Measure success on “held-out” data for which class labels are known (**test data**)
- Classification learning is **supervised**


Classifier
Learner
Training data
A
A
B
Test instance
?
Test data
Classification
B

Example Predictions for `weather.nominal`

Outlook	Temperature	Humidity	Windy	True Label	Classified
sunny	hot	high	FALSE	no	
sunny	hot	high	TRUE	no	
overcast	hot	high	FALSE	yes	
rainy	mild	high	FALSE	yes	
rainy	cool	normal	FALSE	yes	
rainy	cool	normal	TRUE	no	
overcast	cool	normal	TRUE	yes	
sunny	mild	high	FALSE	no	
sunny	cool	normal	FALSE	yes	
rainy	mild	normal	FALSE	yes	
sunny	mild	normal	TRUE	yes	no
overcast	mild	high	TRUE	yes	yes
overcast	hot	normal	FALSE	yes	yes
rainy	mild	high	TRUE	no	yes

- Finding groups of items that are similar
- Clustering is **unsupervised** — the learner operates without a set of labelled training data
- The class of an example is not known ... or at least, not given to the learning algorithm
- Success often measured subjectively; evaluation is problematic

Clustering over `weather.nominal`

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	~~no~~
sunny	hot	high	TRUE	~~no~~
overcast	hot	high	FALSE	~~yes~~
rainy	mild	high	FALSE	~~yes~~
rainy	cool	normal	FALSE	~~yes~~
rainy	cool	normal	TRUE	~~no~~
⋮	⋮	⋮	⋮	⋮

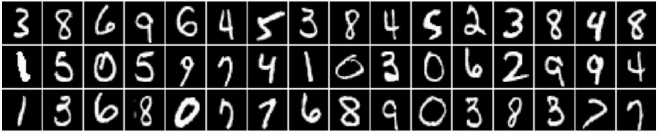
Regression

- Classification learning, but class is continuous (**numeric prediction**)
- Learning is supervised
- Why is this distinct from Classification?
  - In Classification, we can exhaustively enumerate all possible labels for a given instance; a correct prediction entails mapping an instance to the label which is truly correct
  - In Regression, infinitely many labels are possible, we cannot conceivably enumerate them; a "correct" prediction is when the numeric value is acceptably close to the true value

Example Predictions for `weather`

Outlook	Humidity	Windy	Play	Actual Temp	Classified Temp
sunny	85	FALSE	no	85	
sunny	90	TRUE	no	80	
overcast	86	FALSE	yes	83	
rainy	96	FALSE	yes	70	
rainy	80	FALSE	yes	68	
rainy	70	TRUE	no	65	
overcast	65	TRUE	yes	64	
sunny	95	FALSE	no	72	
sunny	70	FALSE	yes	69	
rainy	80	FALSE	yes	75	
sunny	70	TRUE	yes	75	68.8
overcast	90	TRUE	yes	72	76.2
overcast	75	FALSE	yes	81	70.6
rainy	91	TRUE	no	71	76.5

Quiz II



The MNIST digit classification data set

- Design a **classification** task given this data set
- Could we perform **clustering** instead? What would change?
- Can you think of a meaningful **regression** task?

Instance Topology

- Instances characterised as “feature vectors”, defined by a predetermined set of attributes
- Input to learning scheme: set of instances/dataset
  - Flat file representation
  - No relationships between objects
  - No explicit relationship between attributes

What's in an Attribute?

- Each instance is described by a fixed feature vector
- Possible attribute types (levels of measurement):
    1. nominal
    2. ordinal
    3. continuous

Nominal Quantities

- Values are distinct symbols (e.g. $\{\texttt{sunny},\texttt{overcast},\texttt{rainy}\}$)
  - values themselves serve only as labels or names
- Also called **categorical**, or **discrete** (NB. "discrete" implies an order which tends not to exist)
- No relation is implied among nominal values (no ordering or distance measure), and only equality tests can be performed
- Special case: dichotomy ("Boolean" attribute)

Ordinal Quantities

- An explicit order is imposed on the values (e.g. {`hot`,`mild`,`cool`} where `hot` > `mild` > `cool`)
- No distance between values defined and addition and subtraction don't make sense
- Example rule: `temperature` < `hot` →`play` = `yes`
- Distinction between nominal and ordinal not always clear (e.g. `outlook`)

Continuous Quantities

- Continuous quantities are real-valued attributes with a well-defined zero point and no explicit upper bound
- Example: attribute `distance`
    - Distance between an object and itself is zero
- All mathematical operations are allowed (of which addition, subtraction, scalar multiplication are most salient, but other operations are relevant in some contexts)

THE UNIVERSITY OF MELBOURNE

ML in the Wild

Attribute Types Used in Practice

- Many data schemes/learners accommodate continuous attributes, and they are very commonly observed
- Many also support nominal attributes, and they are commonly observed
- Some support ordinal attributes, which are occasionally observed (but often treated as one of the other types)

THE UNIVERSITY OF MELBOURNE

Transforming Nominal to Boolean

- Simple transformation allows nominal attribute with *n* values to be coded using *n* Boolean attributes ("one–hot")
- Example: attribute `temperature`

$$\texttt{hot} = [1, 0, 0]$$
$$\texttt{mild} = [0, 1, 0]$$
$$\texttt{cool} = [0, 0, 1]$$

THE UNIVERSITY OF MELBOURNE

Preparing the Input

- 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/storage may be required
- Critical: type and level of data aggregation

Missing Values

- The number of attributes may vary in practice
  - missing values
  - inter-dependent attributes
- Missing values are prevalent in data analysis
  - 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)
- How to deal with missing values
  - Remove instances with missing values
  - `missing` may need to be coded discretely
  - Imputation

Inaccurate Values

- Cause: a given data mining application is often not known at the time logging is set up
- Result: errors and omissions that don't affect original purpose of data (e.g. age of customer)
- Typographical errors in nominal attributes $\rightarrow$values need to be checked for consistency
- Typographical and measurement errors in numeric attributes $\rightarrow$outliers need to be identified
- Errors may be deliberate (e.g. wrong post codes)

Getting to Know the Data

- Simple visualization tools are very useful
  - Nominal attributes: histograms (distribution consistent with background knowledge?)
  - Numeric attributes: scatter plots (any obvious outliers?)
- 2-D and 3-D plots show dependencies
- Need to consult domain experts
- Too much data to inspect? Take a sample!
- Imbalanced data? Re-sampling!
- You can never know your data **too** well

Coding Demo!

Intended take-aways

- Jupyter Notebook
- Looking at a data set
- Reading in data
- Separating features from class labels (for each instance)

Summary

Today: Establishing common vocabulary

- What are instances, attributes and concepts?
- Learning paradigms: supervised and unsupervised
- Concepts: Regression, Classification, Clustering
- Attributes: types and encodings
- Python and Jupyter

Next: Probabilities (recap) and probabilistic modeling