Lecture 2: Machine Learning Concepts

COMP90049
Introduction to Machine Learning

Semester 2, 2022

Lida Rashidi, CIS

Acknowledgement: Lea Frermann

Roadmap

Last lecture

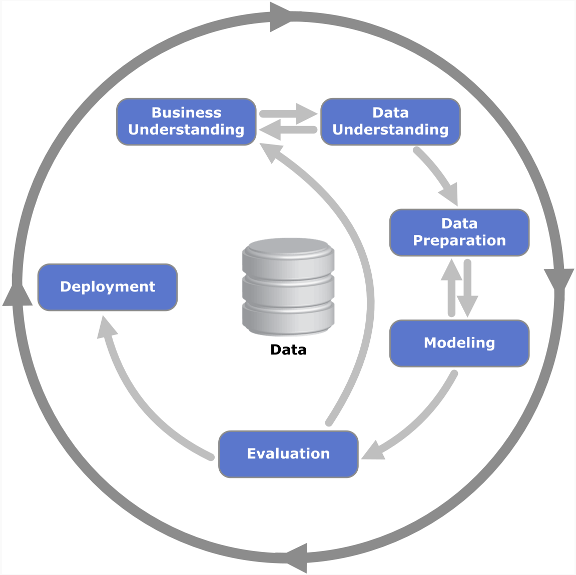
- Warm-up
- Housekeeping COMP90049
- Machine Learning

This lecture

- Establishing terminology
- Basic concepts of ML: instances, attributes, learning paradigms, ...
- Python demo

Basics of ML: Instances, Attributes and Learning Paradigms

Typical Workflow



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

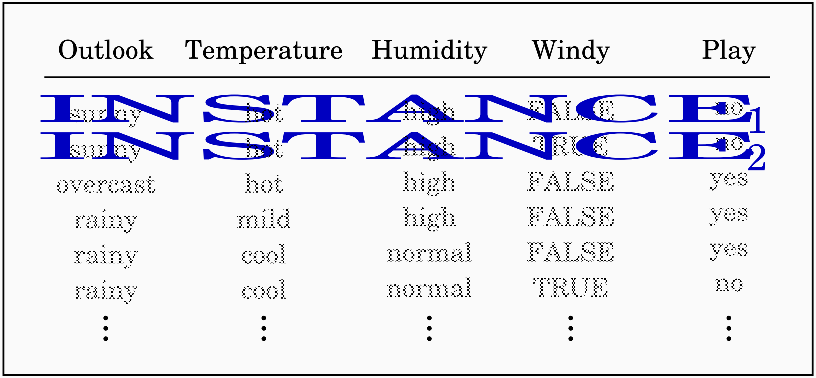
Terminology

- 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
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
⋮	⋮	⋮	⋮	⋮

INSTANCE$_1$
INSTANCE$_2$

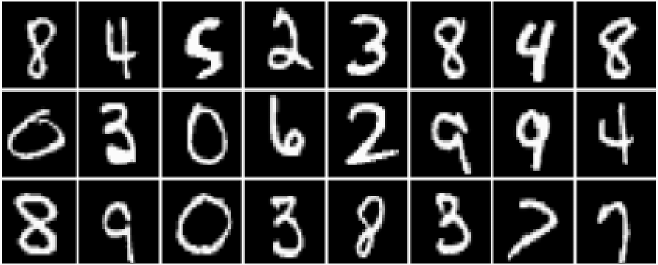
Example: `weather.nominal` Dataset

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$_2$

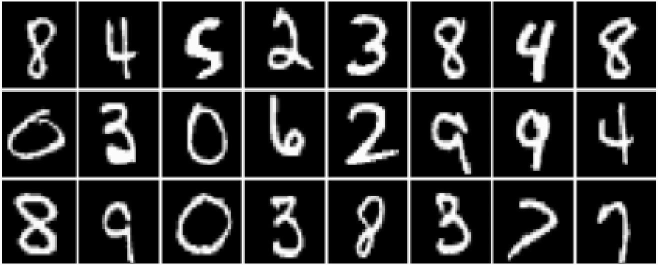
Quiz I



The MNIST digit classification data set

- How many **instances** do you see in the dataset above?

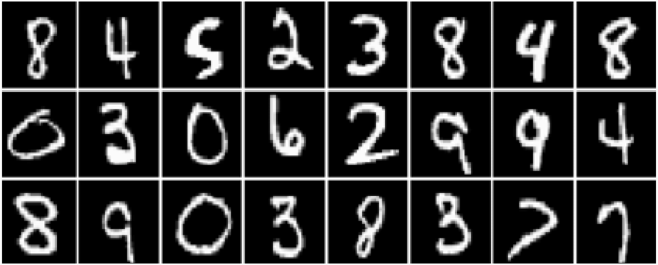
Quiz I



The MNIST digit classification data set

- How many **instances** do you see in the dataset above?
- What are these instances?

Quiz I



The MNIST digit classification data set

- How many **instances** do you see in the dataset above?
- What are these instances?
- What **features** or **attributes** do we have for the instances?

What’s a Concept?

“Concepts” that we aim to learn:

- Predicting a discrete class (Classification)
- Grouping similar instances into clusters (Clustering)
- Predicting a numeric quantity (Regression)
- Detecting associations between attribute values (Association Learning)

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 inventory of classes, and instead discover groups of ‘similar’ examples in a given dataset. Two flavors :

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 inventory of classes, and instead discover groups of ‘similar’ examples in a given dataset. Two flavors :

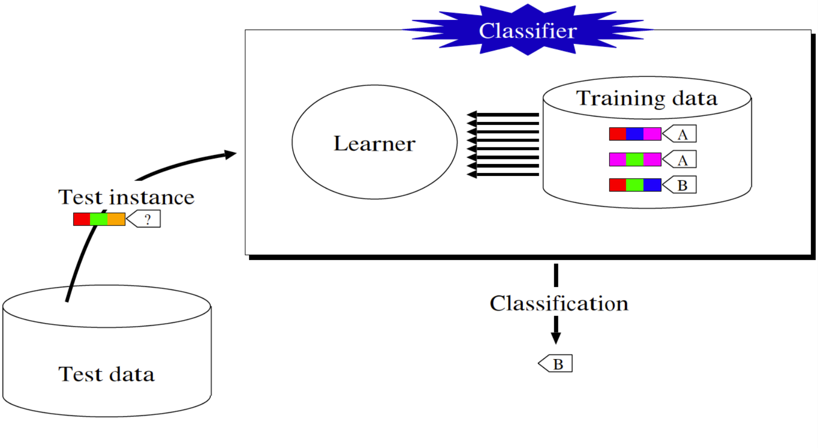
  - dynamically discover the “classes” (implicitly derived from grouping of instances) in the process of categorising the instances [STRONG]

  ... *OR* ...

  - categorise instances as certain labels without the aid of pre-classified data [WEAK]

Classification

- Assigning an instance a discrete class label
- Classification learning is **supervised**
- 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**)


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

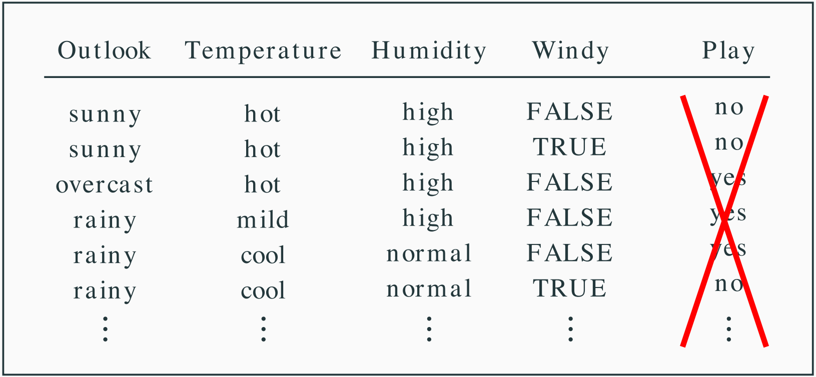
Example Predictions for `weather.nominal`

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

Clustering

- 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~~
⋮	⋮	⋮	⋮	~~⋮~~

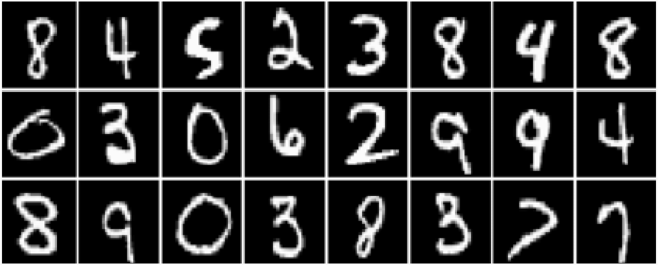
Regression

- 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	71.2
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. What ‘concept(s)’ could be learnt?
- 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
- Attribute Data types
  1. discrete: nominal (also: categorical) or ordinal
  2. continuous: numeric

What about class label data types?

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**
- Special case: dichotomy (“Boolean” attribute)
- No relation is implied among nominal values (no ordering or distance measure), and only equality tests can be performed

Ordinal Quantities

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

Numeric Quantities

- Numeric quantities are real-valued attributes
- Scalar (a single number): attribute `distance`
- Vector-valued (a vector of numbers each pertaining to a feature or feature value): attribute `position` (x,y coordinate)
- All mathematical operations are allowed (of which addition, subtraction, scalar multiplication are most salient, but other operations are relevant in some contexts)

Attribute Types and Machine Learning Models

Most machine learning algorithms assume a certain type of attribute

- Naive Bayes: nominal or numeric
- Logistic/Linear Regression: numeric
- Perceptron/Neural networks: numeric

If we have the wrong attribute type for our algorithm (or attributes with different types for each instance), we can

- Select only attributes with the correct type
- Change the model assumptions to match the data
- **Change the attributes to match the model**

Converting Nominal to Numeric Attributes

Option 1: Map category names to numbers

- “red”, “blue”, “green”, “yellow”
- 0, 1, 2, 3

Graphical representation:

- Problem: creates an artificial ordering. Some categories will appear more similar to each other than others
- Especially problematic with a large number of categories

Converting Nominal to Numeric Attributes

Option 2: One-hot encoding

"red"	=	[1, 0, 0, 0]
"blue"	=	[0, 1, 0, 0]
"green"	=	[0, 0, 1, 0]
"yellow"	=	[0, 0, 0, 1]

Graphical representation:

- Better way of encoding categorical attributes in a numeric way
- Problem: increases the dimensionality of the feature space

Numeric Feature Normalization

Features of vastly different scale can be problematic

- Some machine learning models assume features to follow a normal distribution
- Some learning algorithms are overpowered by large feature values (and ignore smaller ones)

- Feature **standardization** rescales features to be distributed around a 0 mean with a unit standard deviation.

$$x' = \frac{x - \mu}{\sigma}$$

- Feature **scaling** rescales features to a given range. For example, **Min-max scaling** rescales values between 0 and 1 using the minimum and maximum feature value observed in the data

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Converting Numeric to Nominal Attributes

Distcretization: Group numeric values into a pre-defined set of distinct categories. E.g., map housing prices to {“high”, “medium”, “low”}

To do this, we

- First, decide on the number of categories
- Secondly, decide on the category boundaries

Converting Numeric to Nominal Attributes

Distcretization: Group numeric values into a pre-defined set of distinct categories. E.g., map housing prices to {“high”, “medium”, “low”}

To do this, we

- First, decide on the number of categories
- Secondly, decide on the category boundaries

Option 1: Equal widths discretisation

- Find the minimum and maximum of the data
- Partition the values into *n* bins of width (max-min)/n bins
- **Problem 1:** outliers
- **Problem 2:** bins may end up with vastly different number of items
- **Problem 3:** how to select *n*?

Converting Numeric to Nominal Attributes

Distcretization: Group numeric values into a pre-defined set of distinct categories. E.g., map housing prices to {“high”, “medium”, “low”}

To do this, we

- First, decide on the number of categories
- Secondly, decide on the category boundaries

Option 2: Equal frequency discretisation

- Sort the values
- Partition them into *n* bins such that each bin has an identical number of items
- **Problem 1:** boundaries could be hard to interpret
- **Problem 2:** how to select *n*?

Converting Numeric to Nominal Attributes

Distcretization: Group numeric values into a pre-defined set of distinct categories. E.g., map housing prices to {“high”, “medium”, “low”}

To do this, we

- First, decide on the number of categories
- Secondly, decide on the category boundaries

Option 3: Clustering

- Use unsupervised machine learning to group the value into *n* clusters
- For example: K-means clustering (more on that later)
- **Problem 1:** how to evaluate the result?
- **Problem 2:** how to select K?

ML in the Wild

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
- Typical cases of out-of-range values:
  - Types: unknown, unrecorded, irrelevant
  - Reasons:
    - malfunctioning equipment
    - changes in experimental design
    - collation of different datasets
    - measurement not possible
- Most models assume that values are **missing at random**
- Missing value may have significance in itself (e.g. missing test in a medical examination) → **Missing not at random**. `Missing` may need to be coded discretely.

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 →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 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!
- You can never know your data too well (**or can you?**)

Coding Demo!

Intended take-aways

- Starting Jupyter Notebook
- Reading in a dataset (using basic Python)
- Reading in a dataset (using the `pandas` library)
- Formatting a dataset into lists (of instances)
- 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: K-Nearest Neighbors