Lecture 22: Recap – Part 1

COMP90049
Introduction to Machine Learning

Semester 2, 2020

Hadi Khorshidi, CIS

Roadmap

This lecture

- Details on the exam
- Recap of part 1 of this course

Exam Details

Date, time, format...

- The exam will be on **November 17th at 3pm**
- The exam will be **2 hours (strict time limit)**, with an additional **15 minutes of reading time** and a **30 minute slack period for technical overhead** (including uploading solutions).
- The exam will be a **Canvas Quiz**
- The exam will not be invigilated, and it will be an **open book** exam.

Exam Content Details

- Worth **40% of your grade**
- A number of questions of three different categories (coming up next)
- You should attempt all questions (no pick-and-choose)
- Questions have different weight (!)
- **The exam is worth 120 marks**, i.e., $\approx$ 1 mark per minute. The marks associated with a question will give you an idea about how much time you should spend on it.

Question Types 1 / 3: Multiple-choice questions (MCQ)

Section A: Multiple-choice Questions

- **Resembling Quizzes posted throughout the semester**
- one or more correct answers
- typically require some calculation
- you will answer the question directly in Canvas (by ticking the appropriate box(es))

Section A: Multiple-choice Questions

Question 2

1 pts

Which statement is FALSE about Gradient Descent? You may have more than one answer.

- [ ] It is guaranteed to find a local optimum
- [ ] It is guaranteed to find a global minimum
- [ ] The learning rate influences the step size
- [ ] It is useful when we cannot compute the derivative of the target function

THE UNIVERSITY OF MELBOURNE

Question Types 1 / 3: Multiple-choice questions (MCQ)

Question 2

1 pts

Calculate the entropy for the following clustering outcome:

	Class = yes	Class = No
Cluster 1	3	1
Cluster 2	2	4

- ◯ 0.45
- ◯ 1
- ◯ 0.87
- ◯ 0.92

THE UNIVERSITY OF MELBOURNE

Question Types 2 / 3: Method Questions (METHOD)

Section B: Method Questions

- **Resembling Workshop Questions**
- Demonstrate your conceptual understanding of the methods that we have studied in this subject.
- usually involve some calculations, and you will need to show your calculations (i.e., not just state the answer)
- You will answer these questions in a **text box**, with the option to include **images of your hand-written solution** (e.g., of formulas or diagrams). If you combine images with typed text, all information must be presented in logical order, easy to follow for the marker. You are welcome to upload only an image of your hand-written solution (no typing).

THE UNIVERSITY OF MELBOURNE

Section B: Method Questions

Question 9 10 pts

K-means [METHOD]

With respect to the following data set of 6 instances with 3 attributes and two classes F and T, plus a single test instance labelled "?":

instance #	ele	fed	aus	CLASS
1	1	1	1	F
2	1	0	0	F
3	1	1	0	T
4	1	1	0	T
5	1	1	1	T
6	1	1	1	T
7	0	0	0	?

Exclude the classlabels from the dataset, and cluster all 7 instances using the method of "k-means". Apply the Manhattan Distance as a similarity measure; use the second (1,0,0) and third (1,1,0) instances as seeds. Show your mathematical working.

12pt Paragraph

p 0 words </>

Section B: Method Questions

Question 9 — 10 pts

K-means [METHOD]

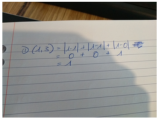
With respect to the following data set of 6 instances with 3 attributes and two classes F and T, plus a single test instance labelled "?":

instance #	ele	fed	aus	CLASS
1	1	1	1	F
2	1	0	0	F
3	1	1	0	T
4	1	1	0	T
5	1	1	1	T
6	1	1	1	T
7	0	0	0	?

Exclude the classlabels from the dataset, and cluster all 7 instances using the method of "k-means". Apply the Manhattan Distance as a similarity measure; use the second (1,0,0) and third (1,1,0) instances as seeds. Show your mathematical working.

12pt Paragraph

In order to arrive at the clustering, we first ...... blah blah ...



....

Overall, we arrive at cluster A with instances [...] and cluster B with instances [...]

p 24 words

Question Types 3 / 3: Long answer questions (LONG_A)

Section C: Long Answer Questions: Design and Application Questions

- **Resembling Assignment Questions**
- Demonstrate that you have gained a high-level understanding of the methods and algorithms covered in this subject, and can apply that understanding.
- Expected answer to each question to be from one third of a page to one full page in length.
- Require significantly more thought than MCQ or METHOD, and should be attempted last.
- You will answer these questions in a **text box** just in typing.

Question Types 3 / 3: Long answer questions (LONG_A)

Section C: Long Answer Questions: Design and Application Questions

Question 10 — 20 pts

College Admissions [LONG_A]

You are a manager of a life insurance company and want to provide optimal insurance quotes to your potential customers. The quotes fall into one of three categories 'high', 'medium' or 'low' premium. Your company is so popular that you cannot sort through all applications manually. Instead, you want to pre-sort applications into meaningful groups. Each application comes with features such as

- Name of applicant
- Age of applicant
- Favorite color of applicant
- Longest period spent in hospital
- Marital status of applicant
- Gender of applicant

Please answer the following questions with respect to the machine learning problem introduced above.

1. Describe the machine learning concept and features underlying this task. [3 marks]
2. Assume you have access to the following ML methods: (a) Decision trees; (b) neural networks; (c) k-means. For each algorithm, state whether it is appropriate in this situation as well as a reason for your decision [6 marks]
3. Now assume a slightly different situation where you (a) have access to a set of 50 admission decisions from previous years. Describe how this new information will change (a) your machine learning approach. [8 marks]
4. Further questions e.g., on evaluation or feature selection ... [3 marks]

12pt Paragraph

p — 0 words </>

Machine Learning Definitions, Terminology, and Concepts

What is machine learning?

"We are drowning in information, but we are starved for knowledge"

John Naisbitt, Megatrends

Our definition of Machine Learning

automatic extraction of **valid, novel, useful and comprehensible knowledge** (rules, regularities, patterns, constraints, models, ...) from arbitrary sets of data

Three ingredients for machine learning

... and related questions

Three ingredients for machine learning

... and related questions

1. Data

- Discrete vs continuous vs ...
- Big data vs small data
- Labeled data vs unlabeled data
- Public vs sensitive data

... and related questions

Models

- function mapping from inputs to outputs
- motivated by a data *generating* hypothesis
- probabilistic machine learning models
- geometric machine learning models
- parameters of the function are unknown

Three ingredients for machine learning

... and related questions

Learning

- Improving (on a task) after data is taken into account
- Finding the best model parameters (for a given task)
- Supervised vs. unsupervised learning

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**

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

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
- Possible attribute types (levels of measurement):
  1. nominal
  2. ordinal
  3. continuous

Classification

Classification

- **Lazy learning:**
  No model training, Instance-based Learning
- **Eager learning:**
  Train a model using training data and use the model to predict test instances

K-NN

- K-NN is a lazy learner
- Algorithm
    - Measure the similarity (or distance) between the test instance and training data
    - Find K-Nearest neighbours
    - Return the class of the test instance using the corresponding labels of the K-Nearest neighbours
- Advantages:
    - Intuitive
    - No assumptions
    - Evolve and adapt immediately
- Concerns:
    - What similarity (or distance) measure?
    - How to aggregate the labels of the neighbours?
    - What K value?
    - Expensive if the data set is large

Eager learning

- **Linear Classification**
- **Non-Linear Classification**

Linear Classification

Naive Bayes I

- Task: classify an instance $D = \langle x_1, x_2, ..., x_n \rangle$ according to one of the classes $c_j \in C$

$$
\begin{aligned}
c &= \operatorname*{argmax}_{c_j \in C} P(c_j | x_1, x_2, ..., x_n) && (1)\\
&= \operatorname*{argmax}_{c_j \in C} \frac{P(c_j) P(x_1, x_2, ..., x_n | c_j)}{P(x_1, x_2, ..., x_n)} && (2)\\
&= \operatorname*{argmax}_{c_j \in C} P(c_j) P(x_1, x_2, ..., x_n | c_j) && (3)\\
&= \operatorname*{argmax}_{c_j \in C} P(c_j) \prod_i P(x_i | c_j) && (4)
\end{aligned}
$$

Posterior $P(c_j | x_1, x_2, ..., x_n) = \frac{prior * likelihood}{evidence}$

- Predicts D belongs to c_i iff the probability $P(c_i | D)$ is the highest among all the $P(c_k | D)$ for all the K classes
- Why can we go from (2) to (3)?
- What does the equality between (3) and (4) imply?

The problem with unseen features

- If any term $P(x_m|y) = 0$ then the class probability $P(y|x) = 0$
- But, we already established that in any realistic scenario we won't see every class-feature combination during training
- A single zero renders many additional meaningful observations irrelevant
- **Solution:** no event is impossible: $P(x_m|y) > 0 \forall x_m \forall y$
- We need to readjust the remaining model parameters to maintain valid probability distributions ($\sum_i \psi_i = 1$)

1. Epsilon Smoothing
2. Laplace Smoothing

Naive Bayes IV: Estimation

- What are the parameters to be learnt for a NB model?

- How are these parameters learnt?

Naive Bayes IV: Estimation

- What are the parameters to be learnt for a NB model?

$p(y)$

$p(x_m|y)$

- How are these parameters learnt?

Maximum Likelihood Estimation

Closed-form Optimisation Recipe

Logistic Regression I

The model

- Is a **binary** classification model
- Is a **probabilistic discriminative model**
- Optimizes $P(y|x)$ directly
- Learns to optimally discriminate between inputs which belong to different classes
- No model of $P(x|y) \rightarrow$ no conditional feature independence assumption

Logistic Regression II

- Let's assume a **binary** classification task, y is `true` (1) or `false` (0).
- We model **probabilities** $P(y = 1|x; \theta) = p(x)$ as a function of observations x under parameters θ. [What about $P(y = 0|x; \theta)$?]
- We want to use a **regression** approach

Logistic Regression II

- Let's assume a **binary** classification task, y is `true` (1) or `false` (0).
- We model **probabilities** $P(y=1|x;\theta) = p(x)$ as a function of observations x under parameters θ. [What about $P(y=0|x;\theta)$?]
- We want to use a **regression** approach
- The logistic function returns the probability of $P(y=1)$ given an in put x

$$P(y=1|x_1, x_2, ..., x_F; \theta) = \frac{1}{1+\exp(-(\theta_0 + \sum_{f=1}^{F} \theta_f x_f))} = \sigma(x;\theta)$$

- We define a **decision boundary**, e.g., predict $y=1$ if $P(y=1|x_1, x_2, ..., x_F; \theta) > 0.5$ and $y=0$ otherwise

Perceptron: Definition I

- The Perceptron is a **minimal neural network**
- **Neural networks** are inspired by the brain – a complex net of **neurons**
- A (computational) neuron is defined as follows:
  - input = a vector x of numeric inputs ($\langle 1, x_1, x_2, ...x_n \rangle$)
  - output = a scalar $y_i \in \mathbb{R}$
  - hyper-parameter: an **activation function** f
  - parameters: $\theta = \langle \theta_0, \theta_1, \theta_2, ...\theta_n \rangle$
- Mathematically:

$$y^i = f\left(\left[\sum_j \theta_j x_j^i\right]\right) = f(\theta^T x^i)$$

The Perceptron Algorithm

$D = \{(\mathbf{x}^i, y^i) | i = 1, 2, ..., N\}$ the set of training instances

Initialize the weight vector $\theta \leftarrow 0$

t $\leftarrow$ 0

repeat

  t $\leftarrow$ t+1

  for each training instance $(x^i, y^i) \in D$ **do**

    compute $\hat{y}^{i,(t)} = f(\theta^T x^i)$

    if $\hat{y}^{i,(t)} \neq y^i$ **then**

      for each each weight θ_j **do**

        update $\theta_j^{(t)} \leftarrow \theta_j^{(t-1)} + \eta(y^i - \hat{y}^{i,(t)})x_j^i$

    else

      $\theta_j^{(t)} \leftarrow \theta_j^{(t-1)}$

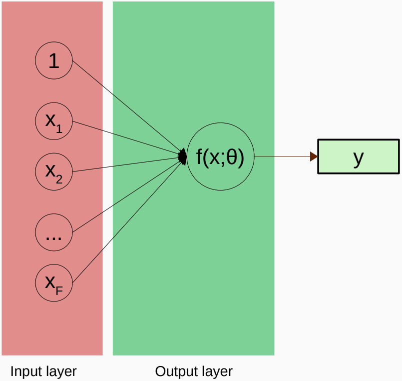
until tired

Return $\theta^{(t)}$

Non-Linear Classification

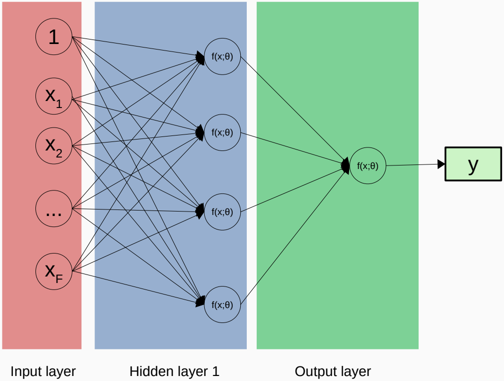
Multi-layer Perceptron I

- **Input layer** with input units x: the first layer, takes features x as inputs
- **Output layer** with output units y: the last layer, has one unit per possible output (e.g., 1 unit for binary classification)
- **Hidden layers** with hidden units h: all layers in between.



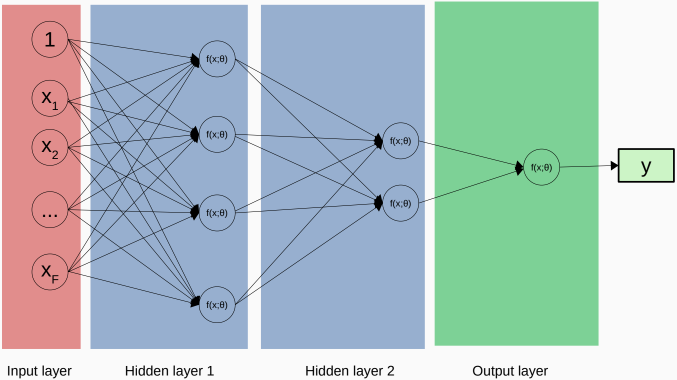
Multi-layer Perceptron I

- **Input layer** with input units x: the first layer, takes features x as inputs
- **Output layer** with output units y: the last layer, has one unit per possible output (e.g., 1 unit for binary classification)
- **Hidden layers** with hidden units h: all layers in between.



Multi-layer Perceptron I

- **Input layer** with input units x: the first layer, takes features x as inputs
- **Output layer** with output units y: the last layer, has one unit per possible output (e.g., 1 unit for binary classification)
- **Hidden layers** with hidden units h: all layers in between.



Designing Neural Networks

- Inputs and feature functions
- Activation Functions
- Network Structure (depth and width)
- Output Function
- Loss Functions

THE UNIVERSITY OF MELBOURNE

Recall Perceptron learning:

- Pass an input through and compute $\hat{y}$
- Compare $\hat{y}$ against y
- Weight update $\theta_i \leftarrow \theta_i + \eta(y - \hat{y})x_i$

Learning the Multi-layer Perceptron

Recall Perceptron learning:

- Pass an input through and compute $\hat{y}$
- Compare $\hat{y}$ against y
- Weight update $\theta_i \leftarrow \theta_i + \eta(y - \hat{y})x_i$

Problems

- This update rule depends on **true target outputs** y
- We only have access to true outputs for the **final layer**
- We do not know the **true activations** for the **hidden layers**. Can we **generalize** the above rule to also update the hidden layers?

Backpropagation provides us with an efficient way of computing partial derivatives of the error of an MLP wrt. each individual weight.

Backpropagation: The Generalized Delta Rule

- The Generalized Delta Rule

$$\triangle\theta_{ij}^2 = \eta\frac{\partial E}{\partial\theta_{ij}^2} = \eta(y^p - \hat{y}^p)g'(z_i)a_j = \eta\, \delta_i\, a_j$$

$$\delta_i = (y^p - \hat{y}^p)g'(z_i)$$

- The above δ_i can only be applied to output units, because it relies on the **target outputs** y^p.
- We do not have target outputs y for the intermediate layers

Backpropagation: The Generalized Delta Rule

- Instead, we **backpropagate** the errors (δs) from right to left through the network

$$\triangle\theta^1_{jk} = \eta\ \delta_j\ a_k$$

$$\delta_j = \sum_i \theta^1_{ij}\ \delta_i\ g'(z_j)$$

More Thoughts

Questions to think about I

Choosing a classification (or any ML) Algorithm

- Probabilistic interpretation?
- Restrictive assumptions on features?
- Restrictive assumptions on the problem?
- How well does it perform?
- How long does it take to train?
- How interpretable is it?
- How much data does it require?

Questions to think about II

How do we know we succeeded?

- Choose the right evaluation metric (accuracy, precision, recall, ...)
- Know the mechanics behind the metrics.
- What is **overfitting** and how do we prevent it?
- Choose the right evaluation strategy, maximizing the utility of your data (cross-validation, hold-out, ...). What to consider?

Questions to think about III

Theoretical considerations and optimization

- Is the problem linearly separable?
- Is my classifier powerful enough to solve my problem?
- What does the objective function of my classifier look like? And what optimization strategy should I choose?

Summary

Source https://www.aitrends.com/machine-learning/here-are-six-machine-learning-success-stories/

Summary

Source `https://www.aitrends.com/machine-learning/here-are-six-machine-learning-success-stories/`

- Understand fundamental mathematical concepts in machine learning (including probability and optimization)
- Understand the theory behind a variety machine learning algorithms
- Identify the correct ML model given a specific data set
- Meaningfully evaluate the output of a ML model in the context of a specific problem
- Apply a variety of ML algorithms
- Python programming: ML model implementation, data processing, evaluation
- Problem solving, Academic writing and presentation

And finally...

Please participate in the university feedback survey!

- What worked well?
- Suggestions for improvements?

Capstone / PhDs

I am looking for motivated master (capstone) and PhD students, working at solving medical problem by machine learning. Feel free to get in touch if you're interested!

THE UNIVERSITY OF MELBOURNE