https://introml.mit.edu/

6.390 Intro to Machine Learning

Fall 24 Final Review

Shen Shen

December 10, 2024

Outline

- Rundown
- Q&A
- Past Exams Walk-through

Week 1 - IntroML

- Terminologies
  - Training, validation, testing
  - Identifying overfitting and underfitting
- Concrete processes
  - Learning algorithm
  - Validation and Cross-validation
  - Concept of hyperparameter

Week 2 - Regression

- Problem Setup
- Analytical solution formula $\theta^* = \left(\tilde{X}^\top \tilde{X}\right)^{-1} \tilde{X}^\top \tilde{Y}$ (and what's $\tilde{X}$)
- When $\tilde{X}^\top \tilde{X}$ not invertible (optimal solution*s* still exist; just not via the "formula")
  - Practically (two scenarios)
  - Visually (obj fun no longer of "bowl" shape, instead has "half-pipe" shape)
  - Mathematically (loss of solution uniqueness)
- Regularization
  - Motivation, how to, when to

Week 3 - Gradient Descent

- The gradient vector (both analytically and conceptually)
- The gradient-descent algorithm and the key update formula
- (Convex + small-enough step-size + gradient descent + global min exists + run long enough) guarantee convergence to a global min
    - What happens when any of these conditions is violated
- How does the stochastic variant differ (Set up, run-time behavior, and conclusion)

Week 4 - Classification

- (Binary) linear classifier (sign based)
- (Binary) Logistic classifiers (sigmoid, NLL loss)
- Linear separator (the equation form, visual form with normal vector)
- Linear separability (interplay with features)
- How to handle multiple classes
  - Softmax generalization (Softmax, cross-entropy)
  - Multiple sigmoids
  - One-vs-one, one-vs-all

Week 5 - Features

- Feature transformations
  - Apply a fixed feature transformation
  - Hand-design feature transformation (e.g. towards getting linear separability)
  - Interplay between the number of features, the quality of features, and the quality of learning algorithms
- Feature encoding
  - One-hot, thermometer, factored, numerical, standardization
  - When and why to use any of those

Week 6 - Neural Networks

- Forward-pass (for evaluation)
- Backward-pass (via backpropogation, for optimization)
- Source of expressiveness
- Output layer design
  - dimension, activation, loss
- Hand-designing weights
  - to match some given function form
  - achieve some goal (e.g. separate a given data set)

Week 7 - Auto-encoders

- Unsupervised learning setup
- Auto-encoder:
  - The idea of compression and reconstruction
  - Mechanically, can use any vanilla classical or neural architecture

Week 8 - CNN

- Forward pass: convolution operation; max-pooling and the typical "pyramid" stack.
- Backward pass: back-propagation to learn filter weights/bias.
- The convolution/max-pooling operation
    - various hyper-parameters (filter size, padding size, stride) in spatial dimension;
    - the 3rd channel/depth dimension
    - reason about in/out shapes.
- Conceptually: weight sharing, "pattern matching" template, independent and parallel processing.

Week 9 - Transformers

- A single input (think one sentence), tokenized into a sequence: n tokens, each token x is d dimensional
- the attention mechanism (one head)
  - learn weights W_q, W_k, W_v to turn raw x inputs into (query, key, value)
  - the mechanics, softmax(raw attention score), shapes
  - masking: why and how
- parallel-processing machines
  - each head is processed in parallel
  - inside a head, each token is processed in parallel

Week 10 - Clustering

- Unsupervised learning set up
- The k-means algorithm
  - cluster assignment; cluster center updates
  - convergence criterion
- The initialization matters
- The choice of hyper-parameter k matters

Week 11 - MDPs

- Definition (the five tuple)
  - π, V, and Q : definition and interpretation
- Policy evaluation: given $\pi(s)$, calculate $V(s)$
  - via summation, or via Bellman recursion or equation
- Policy optimization: finding optimal policy $\pi^*(s)$
  - toy setup: solve via heuristics; more generally: Q value-iteration
- Interpretation of optimal policy
  - how various setup changes optimal policy R, γ, h

Week 12 - Reinforcement Learning

- How RL setup differs from MDP
- Q-learning algorithm
    - Forward thinking: given experiences, work out Q-values.
    - Backward thinking: given realized Q-values, work out experiences.
    - Two new hyper-parameters (compared with MDP value iteration):
        - $\epsilon-$greedy action selection
        - α the learning rate
- The idea of fitting parameterized Q-functions via regression, can handle larger or continuous state/action space

Week 13 - Non-parametric methods

- Decision trees:
  - Flow chart; if/else statement; human-understandable
  - Split dimension, split value, tree structure (root/decision node and leaf)
  - Largest leaf size k matters
  - For classification: weighted-average-entropy or accuracy; for regression, MSE
- $k-$nearest neighbors:
  - memorizes data
  - scaling matters, k matters
  - inefficient in test/prediction time

Course Evaluations

http://registrar.mit.edu/subjectevaluation

We'd love to hear your thoughts on the course: this provides valuable feedback for us and other students, for future semesters! Thank you!🙏

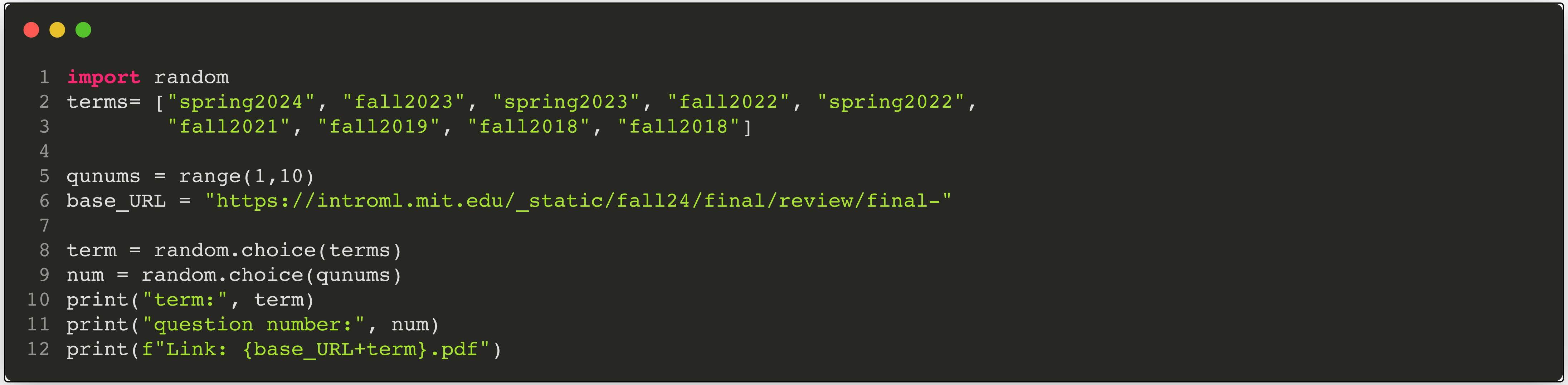
https://shenshen.mit.edu/tree

(The demo won't embed in PDF. But the direct link below works.)

https://shenshen.mit.edu/tree

Resources

- All the released materials Week 1 - Week 13
- Review Question Sampler



```python
import random
terms= ["spring2024", "fall2023", "spring2023", "fall2022", "spring2022",
        "fall2021", "fall2019", "fall2018", "fall2018"]

qunums = range(1,10)
base_URL = "https://introml.mit.edu/_static/fall24/final/review/final-"

term = random.choice(terms)
num = random.choice(qunums)
print("term:", term)
print("question number:", num)
print(f"Link: {base_URL+term}.pdf")
```

- (wip) past exams video walk-through
- Piazza and OHs

General problem-solving tips

Polya's Problem Solving Techniques

In 1945 George Polya published the book *How To Solve It* which quickly became his most prized publication. It sold over one million copies and has been translated into 17 languages. In this book he identifies four basic principles of problem solving.

Polya’s First Principle: Understand the problem

This seems so obvious that it is often not even mentioned, yet studens are often stymied in their efforts to solve problems simply because they don’t understand it fully, or even in part. Polya taught teachers to ask students questions such as:

- Do you understand all the words used in stating the problem?
- What are you asked to find or show?
- Can you restate the problem in your own words?
- Can you think of a picture or diagram that might help you understand the problem?
- Is there enough information to enable you to find a solution?

Polya's Second Principle: Devise a plan

Polya mentions that there are many reasonable ways to solve problems. The skill at choosing an appropriate strategy is best learned by solving many problems. You will find choosing a strategy increasingly easy. A partial list of strategies is included:

- Guess and check
- Make an orderly list
- Eliminate possibilities
- Use symmetry
- Consider special cases
- Use direct reasoning
- Solve an equation
- Look for a pattern
- Draw a picture
- Solve a simpler problem
- Use a model
- Work backwards
- Use a formula
- Be ingenious

Polya's Third Principle: Carry out the plan

This step is usually easier than devising the plan. In general, all you need is care and patience, given that you have the necessary skills. Persist with the plan that you have chosen. If it continues not to work discard it and choose another. Don't be misled, this is how mathematics is done, even by professionals.

Polya's Fourth Principle: Look back

Polya mentions that much can be gained by taking the time to reflect and look back at what you have done, what worked, and what didn't. Doing this will enable you to predict what strategy to use to solve future problems.

More detailed CliffsNotes

Exam-taking tips

- Arrive 5min early to get settled in.
- Bring a pencil (and **eraser**), a watch, and some water.
- Look over whole exam and strategize for the order you do problems.

Good luck!

6.390

https://introml.mit.edu/

Thanks for the Fall24 semester!