

10-315 Learning Objectives

Spring 2025 - Midterm 1

Course Level Learning Outcomes

1. Course Level
 - a. Implement and analyze existing learning algorithms, including well-studied methods for classification and regression
 - b. Integrate multiple facets of practical machine learning in a single system: data preprocessing, learning, and model selection
 - c. Describe the formal properties of models and algorithms for learning and explain the practical implications of those results
 - d. Employ calculus, linear algebra, and optimization in order to develop new predictive models or learning methods
 - e. Given a description of an ML technique, analyze it to identify (1) the expressive power of the technique; (2) the computational properties of the algorithm: any guarantees (or lack thereof) regarding termination, convergence, correctness, accuracy, or generalization power.

ML Basics

1. Course Overview
 - a. Formulate a well-posed learning problem for a real-world task by identifying the task, performance measure, and training experience
 - b. Describe common learning paradigms in terms of the type of data available and when, the form of prediction, and the structure of the output prediction
 - c. Reason about the decision boundary for a classification algorithm (e.g. compare k-NN, decision trees, and logistic regression)
2. Decision Trees
 - a. Formalize a learning problem by identifying the input space, output space, hypothesis, and hypothesis space
 - b. Implement decision tree training and prediction
 - c. Use effective splitting criteria for decision trees and be able to define entropy, conditional entropy, and mutual information
 - d. Explain the difference between memorization and generalization [C1ML]
 - e. Describe the inductive bias of a decision tree
 - f. Judge whether a decision tree is "underfitting" or "overfitting"
 - g. Explain how a pruning or early stopping method affects overfitting in decision tree learning

- h. Describe the greedy nature of decision tree algorithms and the implications of achieving an optimal solution
 - i. Explain how decision trees could be used for regression tasks as well as classification tasks
- 3. k-Nearest Neighbor
 - a. Describe a dataset as points in a high dimensional space [C1ML]
 - b. Implement k-nearest neighbor with $O(MN)$ prediction
 - c. Describe the inductive bias of a k-NN classifier and relate it to the scale of the features
 - d. Explain how k-nearest neighbor could be used for regression tasks as well as classification tasks
- 4. Model Selection
 - a. Plan an experiment that uses training, validation, and test datasets to predict the performance of a classifier on unseen data (without cheating)
 - b. Explain the difference between (1) training error, (2) cross-validation error, and (3) test error
 - c. For a given learning technique, identify the model, learning algorithm, parameters, and hyperparameters
 - d. Describe trade-offs between different options for cross-validation

Numerical Optimization for ML

- 1. Optimization for ML
 - a. Identify the relationships between the concepts of loss, risk, empirical risk, and empirical risk minimization as they relate to finding a hypothesis function / model in machine learning
 - b. Given a loss function and hypothesis function structure (e.g. linear regression structure), formulate the machine learning optimization problem
 - c. Explain why there could be a need for more general loss functions than just zero-one loss for classification tasks
 - d. Demonstrate how squared error loss and mean squared error related to empirical risk minimization for regression
 - e. Apply (batch) gradient descent, stochastic gradient descent, and mini-batch gradient descent to optimize an objective function
 - f. Apply knowledge of zero derivatives to identify a closed-form solution (if one exists) to an optimization problem
 - g. Distinguish between convex, concave, and nonconvex functions
- 2. Calculus
 - a. Derive the gradient of a differentiable, multi-variable objective function by applying knowledge of multivariable calculus, including:
 - i. Partial derivatives involving scalars, vectors, and matrices
 - ii. Numerator and denominator layout of partial derivatives
 - iii. Multivariable chain rule
 - b. Derive matrix calculus properties by expanding linear algebra into scalar form as needed

3. Linear Regression
 - a. Implement learning for linear regression using four optimization techniques: (1) closed form, (2) (batch) gradient descent, (3) stochastic gradient descent, and mini-batch stochastic gradient
 - b. Analyze the tradeoffs of computational complexity vs. convergence speed for linear regression optimization techniques on a particular dataset
 - c. Identify properties of datasets that affect the number of unique solutions to linear regression
4. Logistic Regression
 - a. Describe the differences between linear and logistic regression, including modeling, optimization, and application
 - b. Implement logistic regression for binary or multi-class classification
 - c. Explain the differences and similarities between binary and multi-class logistic regression models and loss functions
 - d. Prove that the decision boundary of binary logistic regression is linear
5. Feature Engineering
 - a. Engineer appropriate features for a new task
 - b. Convert a non-linear dataset (or linearly inseparable dataset) to a linear dataset (or linearly separable dataset) in higher dimensions
 - c. Explain why linear models are still considered linear despite their ability to be applied to tasks with highly non-linear datasets
6. Neural Networks
 - a. Explain the biological motivations for a neural network
 - b. Combine simpler models (e.g. linear regression, binary logistic regression, multi-class logistic regression) as components to build up feed-forward neural network architectures
 - c. Explain the reasons why a neural network can model nonlinear decision boundaries for classification
 - d. Compare and contrast feature engineering with learning features
 - e. Identify (some of) the options available when designing the architecture of a neural network
 - f. Implement and train a feed-forward neural network from scratch including training with the backpropagation algorithm
 - g. Instantiate an optimization method (e.g. SGD) when the parameters of a model are comprised of several matrices corresponding to different layers of a neural network

References

[CIML] Several of these learning objectives are copied or adapted from [Daume III \(2018\)](#) "[CIML](#)".

Built from 10-301/601 Learning objectives