# 10-315 Learning Objectives

## Spring 2025 - Midterm 2

## Course Level Learning Outcomes

1. Course Level
    a. Implement and analyze existing learning algorithms, including well-studied methods for classification, regression, clustering, and feature learning
    b. Integrate multiple facets of practical machine learning in a single system: data preprocessing, learning, regularization, and model selection
    c. Describe the formal properties of models and algorithms for learning and explain the practical implications of those results
    d. Compare and contrast different paradigms for learning (supervised, unsupervised, self-supervised)
    e. Design experiments to evaluate and compare different machine learning techniques on real-world problems
    f. Employ probability, statistics, calculus, linear algebra, and optimization in order to develop new predictive models or learning methods
    g. 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 logistic regression and linear/quadratic discriminant analysis)
2. 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 (3) test error, and (4) true error (error with respect to a true generation function c*(x))
    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. Feature Engineering / Regularization
    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

    —-

    d. Identify symptoms that indicate that a model is overfitting
    e. Add a regularizer to an existing objective in order to combat overfitting
    f. Use regularization penalties that encourage sparsity to identify and remove irrelevant features

# Neural Network Applications

1. Neural Network Applications
    a. Combine various layers and loss functions to create, train, and deploy task-specific neural network architectures, such as convolutional neural networks and autoencoders
    b. Describe the potential benefits of using convolutional layers rather than fully connected layers
    c. Describe the parameters used in convolutional neural networks
    d. Explain how pooling and kernel stride length can reduce the number of values as the network gets deeper

    e. Explain the reasons why a neural network can model nonlinear decision boundaries for classification
    f. Compare and contrast feature engineering with learning features
    g. Identify (some of) the options available when designing the architecture of a neural network
    h. Implement and train a feed-forward neural network from scratch including training with the backpropagation algorithm
    i. Utilize pre-trained parameters of a neural network to fine-tune training a new network such that it can be applied to a different but related task
    j. Leverage functionality within a deep learning toolkit, such as PyTorch, to train and deploy a neural network for a specific task
2. Language models
    a. Build a language model using word embeddings by formulating the objective using encoding and decoding linear layers, deriving the gradient update equations, and training the model in a self-supervised manner
    b. Adjust basic language models (n-grams and word embeddings) to use longer context than just a single token and discuss the impact of increased context size on model size, training data, and model performance.
    c. Given an input token context, generate the next token by sampling from the probabilities generated by a trained language model
    d. Compare different techniques for basic language models, specifically n-grams and word embeddings
    e. Describe the motivation for using attention mechanisms rather than a uniform combination of input content embeddings
    f. Describe the motivation for using multiple attention heads rather than a single attention head
    g. Give various examples of the role of linear operations in GPT language models
    h. Identify the size of and difference between parameters and computed values (activations) in various components within transformer models, especially as the context size changes.

# MLE, MAP, and Generative Models

1. MLE and MAP
    a. Recall probability basics, including but not limited to: discrete and continuous random variables, probability mass functions, probability density functions, events vs. random variables, expectation and variance, joint probability distributions, marginal probabilities, conditional probabilities, independence, and conditional independence
    b. Describe common probability distributions such as Bernoulli, Categorical, Gaussian, etc.
    c. State the principle of maximum likelihood estimation and explain what it tries to accomplish
    d. State the principle of maximum a posteriori estimation and explain why we use it

    e. Derive the MLE or MAP parameters of a simple model in closed form
    f. Define maximum conditional likelihood estimation (MCLE) and describe how it differs from MLE
    g. Provide a probabilistic interpretation of linear regression
    h. For linear regression, show that the parameters that minimize squared error are equivalent to those that maximize conditional likelihood
    i. Describe detailed probabilistic relationships between the likelihood, posterior, and prior on the parameters
    j. Show that the Gaussian and Laplace priors correspond to specific regularization penalties
    k. Explain the practical reasons why we work with the log of the likelihood
2. Generative vs. Discriminative
    a. Write a generative story given classification or regression task, i.e., how was the supervised data generated?
    b. Contrast generative vs. discriminative modeling using properties of probability distributions
    c. Describe the tradeoffs of generative vs. discriminative models
    d. Describe the detailed probabilistic relationship between class priors, class conditional distributions, the likelihood, and (posterior) conditional likelihood
    e. Formulate the MLE for a generative model, including precise use of notation to capture how specific parameters are used in the model
    f. Explain how the distinction between MLE and MAP differs from the distinction between generative and discriminative models
3. Gaussian (Linear and Quadratic) Discriminant Analysis
    a. Formulate and solve the MLE optimization for models with Bernoulli/Categorical class priors and Gaussian class conditional distributions (discriminant analysis)
    b. Derive the equation for a decision boundary given the parameters for a discriminant analysis model
    c. Explain the difference between linear and quadratic discriminant analysis, including the conditions that lead to each
    d. Describe the difference between Gaussian discriminant analysis and logistic regression
    e. Draw estimated Gaussian contours and decision boundaries for a discriminant analysis model on a simple dataset with 1-D or 2-D input features
4. Naive Bayes
    a. Write the generative story for naive Bayes
    b. Create a new naive Bayes classifier using your favorite probability distribution for the features given the class
    c. Apply the principle of maximum likelihood estimation (MLE) to learn the parameters of a naive Bayes model
    d. Apply the principle of maximum a posteriori (MAP) estimation to learn the parameters of a naive Bayes model
    e. Motivate the need for MAP estimation through the deficiencies of MLE

    f. Identify the relationship between the covariance matrix of a class-conditional Gaussian distribution and the naive Bayes assumption

# Unsupervised Learning

1. Dimensionality Reduction, PCA, and Autoencoders
    a. Identify examples of high-dimensional data and common use cases for dimensionality reduction
    b. Draw the principal components of a given toy dataset
    c. Establish the equivalence of minimization of reconstruction error with maximization of variance
    d. Given a set of principal components, project from high to low-dimensional space and do the reverse to produce a reconstruction
    e. Explain the connection between PCA, eigenvectors, eigenvalues, and covariance matrix
    f. Use common methods in linear algebra to obtain the principal components
    g. Discuss the tradeoffs involved in choosing the number of principal components to retain
    h. Describe the difference between PCA and autoencoders
    i. Give example applications of PCA and autoencoders
2. K-means Clustering
    a. Define an objective function that gives rise to a "good" clustering
    b. Apply block coordinate descent to an objective function preferring each point to be close to its nearest objective function to obtain the K-means algorithm
    c. Implement the K-means algorithm
    d. Connect the nonconvexity of the K-means objective function with the (possibly) poor performance of random initialization
3. Variational Autoencoders and Variational Inference
    a. Describe the similarities and differences between autoencoders and variational autoencoders
    b. Train a VAE to generate images from a random sample from the latent space
    c. Understand the trade-offs between autoencoder reconstruction error and the KL divergence regularization added to the VAE objective
    d. Show the relationship between KL divergence, cross-entropy, and entropy
    e. Use variational inference and Jensen's inequality to derive the evidence lower bound (ELBO) for the log likelihood of variational autoencoders

# Additional techniques

1. Recommender Systems
    a. Briefly describe item-based, user-based, and collaborative filtering approaches to recommender systems
    b. Describe how a common feature space of users and items can be learned and applied to predict new ratings

c. Formulate collaborative filtering as a latent variable optimization problem, derive the gradients, and implement the training of the recommender system

# References

Built from 10-301/601 learning objectives