02-620 Week 1

Machine Learning for Scientists

Aidan Jan

January 16, 2026

Introduction

Machine learning has many uses, including

- Visual: image segmentation, object recognition, classification
- Auditory and Textual: speech recognition, web scraping, chat bots
- Robotics: self driving cars, movement imitation

Modern biology needs machine learning.

- Traditional biology: one locus at a time
  - Huntington disease and mutations in the *HTT* gene [MacDonald 1993 *Cell*]
  - Cystic fibrosis and mutations in the *CFTR* gene [Riordan 1989 *Science*]
- Modern high-throughput biology: genome wide analysis
  - Use **all genetic variants** to predict risk complex traits (e.g., type II diabetes) [Weissbrod 2022 *Nat Genet*]
  - Aims to identify **all causal genes** for diseases such as Alzheimer's disease [Kunkle 2019 *Nat Genet*]

Some direct examples of machine learning used in biological contexts include

- finding genes on the human genome with hidden Markov models
- inferring genetic ancestry (finding SNPs) with principal component analysis
- predicting cancer subtypes from gene expression data
- learning gene regulatory networks

General Machine Learning Formulation

Data is represented as N data points with D features. This is represented in a matrix of form $X \in \mathbb{R}^{N \times D}$. Or,

$$x_1, \ldots, x_n \in \mathbb{R}^D$$

Optionally, the data points can have labels (for reinforcement/supervised learning). Labels are represented as a vector of form $y \in \mathbb{R}^N$. Or,

$$y_1, \ldots, y_n \in \mathbb{R}$$

A **Model** is a parametric function, $f(\cdot; \theta) : \mathbb{R}^D \to \mathbb{R}$ with learnable parameters θ. Finally, **learning** is the process of estimating parameters $\hat{\theta}$ from the data (X, y) by optimizing an objective function. An example of an objective function is letting

$$y_i \approx f(x_i; \hat{\theta})$$

Inference is the useful case of machine learning; for a new data point $x_{i'}$, predict $\hat{y}_{i'} = f(x_{i'}; \hat{\theta})$.

Example: Univariate Linear Regression

- **Data:**
  - N data points with $D = 1$ features: $x_i \in \mathbb{R}$
  - Labels: $y_i \in \mathbb{R}$
- **Model:**
  - Parametric function: $f(x_i; \theta) = \theta_1 x_1 + \theta_0$
  - Parameters: $\theta = (\theta_0, \theta_1)$
- **Learning:**
  - Estimate parameters $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1)$ from data (X, y) by letting $y_i \approx \hat{\theta}_1 x_i + \hat{\theta}_0$
- **Inference:**
  - For a **new** data point $x_{i'}$, predict $\hat{y}_{i'} = \hat{\theta}_1 x_{i'} + \hat{\theta}_0$

Taxonomy of Machine Learning Models

Supervised Learning: learning to predict label y_i from features x_i using labeled data (X, y)

- **Regression:** $y_i \in \mathbb{R}$. E.g., linear regression, etc.
- **Classification:** $y_i \in \{0, 1\}$. E.g., Naïve bayes classifier, logistic regression, decision tree, etc.
- Models: deterministic $y_i = f(x_i; \theta)$ or probabilistic $P(Y_i | X_i = x_i; \theta)$

Unsupervised Learning: learning structure or patterns from unlabeled data X. e.g., clustering, dimensionality reduction, etc.

- Models: deterministic $f(x_i; \theta)$ or probabilistic $P(X_i; \theta)$.

Examples of Supervised Learning

Learn to perform the task "Given input x_i, decide output y_i"

- Given blood test results x_i, decide diagnosis y_i. ($y_i = 0$ for healthy, $y_i = 1$ for heart disease)
- Given image x_i, decide whether an object is present y_i. ($y_i = 0$ for no, $y_i = 1$ for yes)
- Given image x_i, decide whether an object is a cat y_i. ($y_i = 0$ for no, $y_i = 1$ for yes)

Examples of Unsupervised Learning

Learn to perform the task "Find interesting patterns in X"

- Given a set of images X, find groups of related images. (e.g., cluster images of animals into horses, cats, dogs, humans, etc.)
- Given a set of documents X, find groups of documents with related topics. (for example, cluster NYTimes articles according to topics, e.g., politics, business, entertainment)

Supervised vs. Unsupervised Learning

	Supervised Learning	Unsupervised Learning
Data Format	Input X, Output y	Input X
Model	$y_i = f(x_i; \theta)$ or $P(Y_i \| X_i = x_i)$	$f(x_i; \theta)$ or $P(X_i; \theta)$
Learning	With a teacher	Without a teacher

Taxonomy of Learning Methods

- **Learning: Estimating Parameters θ**
  - $\mathcal{D}$: data or evidence (e.g., (X, y))
- **Empirical Risk Minimization (ERM)** (general principle)
  - Minimize a predefined loss function (e.g., squared error loss):

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - f(x_i; \theta))^2$$

- **Maximum Likelihood Estimation (MLE)**
  - *Probabilistic instantiation of ERM with **deterministic** θ*

$$\hat{\theta} = \arg\max_{\theta} P(\mathcal{D}; \theta)$$

- **Maximum a Posteriori (MAP)**
  - *Probabilistic instantiation of ERM with **random** Θ*

$$\hat{\theta} = \arg\max_{\theta} P(\Theta = \theta | \mathcal{D})$$

Solving Maximum Likelihood Estimation

Probabilistic instantiation of ERM with deterministic θ

$$\begin{aligned}
\hat{\theta} = \arg\max_{\theta} P(\mathcal{D}; \theta) &= \arg\max_{\theta} P(X, y; \theta) \\
&= \arg\max_{\theta} \prod_{i=1}^{N} P(X_i = x_i, Y_i = y_i; \theta) \\
&= \arg\max_{\theta} \log \prod_{i=1}^{N} P(X_i = x_i, Y_i = y_i; \theta) \\
&= \arg\max_{\theta} \sum_{i=1}^{N} \log P(X_i = x_i, Y_i = y_i; \theta)
\end{aligned}$$

The resulting equation is the log likelihood of the data point i.

- We are able to apply the log to convert the product into a sum since (1) it makes the derivative much easier, and (2), log is a monotonically increasing function.

To solve this optimization problem, we have a few options.

$$\hat{\theta} = \arg\max_{\theta} P(\mathcal{D}; \theta)$$

- Closed-form solution (when available): solve $\frac{\partial P(\mathcal{D};\theta)}{\partial \theta} = 0$
- Numerical optimization (general case): gradient descent, Newton-Raphson, etc.

Bayes Rule in Machine Learning

- $\mathcal{D}$: data / evidence (e.g., (X, y))
- Θ: unknown parameters (random variables)

$$P(\Theta|\mathcal{D}) = \frac{P(\mathcal{D}|\Theta)P(\Theta)}{P(\mathcal{D})}$$

- The **posterior** is the belief on the unknown quantity **after** you see data $\mathcal{D}$, and is represented by $P(\Theta|\mathcal{D})$
- **Likelihood** is how likely the observed data is under the particular unknown quantity Θ, and is represented by $P(\mathcal{D}|\Theta)$
- The **prior** is the belief on the unknown quantity **before** seeing data $\mathcal{D}$, and is represented by $P(\Theta)$

Maximum a Posteriori (MAP) Estimation

Probabilistic instantiation of ERM with random Θ

$$\begin{aligned}
\hat{\theta} = \arg\max_{\theta} P(\Theta|\mathcal{D}) &= \arg\max_{\theta} \frac{P(\mathcal{D}|\theta)P(\Theta)}{P(\mathcal{D})} \\
&= \arg\max_{\theta} P(\mathcal{D}|\theta)P(\Theta) \\
&= \arg\max_{\theta} P(\Theta) \prod_{i=1}^{N} P(X_i = x_i, Y_i = y_i|\Theta) \\
&= \arg\max_{\theta} \left[\log P(\Theta = \theta) + \sum_{i=1}^{N} \log P(Y_i = y_i, X_i = x_i|\Theta = \theta) \right]
\end{aligned}$$

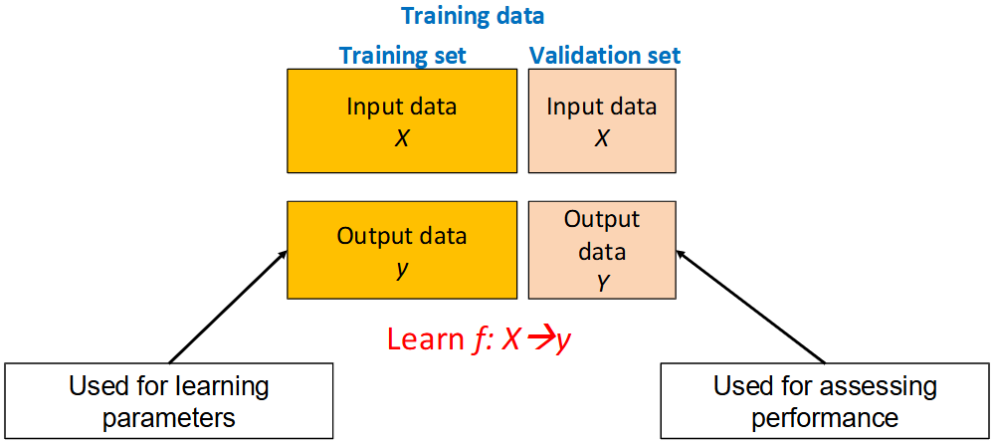
In this case, the first log term ($\log P(\Theta = \theta)$) is the log prior, and the second term (the sum) is the log likelihood of data point i. In essence, MAP is the same as MLE except with a nonzero prior term. In a similar way to MLE, we can solve this optimization problem in two ways:

$$\arg\max_{\theta} P(\Theta|\mathcal{D})$$

- Closed-form solution (when available): solve $\frac{\partial P(\Theta|\mathcal{D})}{\partial \theta} = 0$
- Numerical optimization (general case): gradient descent, Newton-Raphson, etc.

Avoid Overfitting For Model Performance Assessment

Data used for assessing performance should be independent of data used for learning parameters (training)



Common Machine Learning Questions

Data

- N data points with D features
$$x_1, \cdots, x_N \in \mathbb{R}^D$$
- (Optional) labels
$$y_1, \cdots, y_N \in \mathbb{R}$$
- What are features and labels?
- Are the labels binary or continuous?
- Are data points independent?
- Are the feature/labels accurate or noisy?
- How many features relative to the number of samples?

Model

- Parametric function $f(\cdot; \theta) : \mathbb{R}^D \to \mathbb{R}$ with learnable parameters θ

Learning

- Estimate parameters $\hat{\theta}$ from data (X, y) by optimizing an objective function. E.g., letting $y_i \approx f(x_i; \hat{\theta})$

Inference

- For a new data point $x_{i'}$, predict $\hat{y}_{i'} = f(x_{i'}; \hat{\theta})$
- Do model assumptions match the data? Linear or non-linear? Dense or sparse? Noise distribution?
- How should the parameters be learned? MLE or MAP?
- How can the optimization be computed efficiently? Closed-form solution or numerical optimization?
- How can inference be performed efficiently?

Summary

- General machine learning formulation: data + model + learning + inference
- Supervised learning (with labels) versus unsupervised learning (without labels)
- Regression (continuous labels) vs. classification (binary labels)
- Learning methods: MLE vs. MAP
- Optimization methods: close-form vs. numerical methods.
- Model performance should be assessing using independent data

Regression

Beginning with an example, genome-wide association studies (GWASs) aim to find genetic variants (SNPs) associated with disease. A Single Nucleotide Polymorphism (SNP) is a letter of the genome that differs in different individuals. (e.g., G/T).

GWAS as Linear Regression

- Profile genotype and phenotype across a large cohort
- Correlation each SNP with phenotype to find associated SNPs

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

- X is a matrix containing N individuals (rows) and D SNPs (columns)
- x_{ij} is the number of mutations (alternative allele) at individual i and SNP j. E.g., when reference/alternative alleles are A/T, 0 if AA; 1 if AT; 2 if TT.
- y is a $N \times 1$ matrix that represent the phenotype, or labels. In this case, they are trait values or disease status. E.g., BRCA gene expression level, blood pressure, insulin level, if having breast cancer, etc.

A SNP is associated if it is strongly correlated with the phenotype.

Other Examples of Regression

- Predict weight from gender, height, age, etc.
- Predict Google stock price today from Google, Yahoo, MSFT prices yesterday
- Predict each pixel intensity in a robot's current camera image, from previous image and previous action.

Univariate Linear Regression

Data

- N data points with $D = 1$ features: $x_i \in \mathbb{R}$
- Labels: $y_i \in \mathbb{R}$

Model

- Parametric function: $f(x_i; \theta) = \theta_1 x_i + \theta_0$
- Parameters: $\theta = (\theta_0, \theta_1)$

Learning

- Estimate parameters $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1)$ from data (X, y) by letting $y_i \approx \hat{\theta}_1 x_i + \hat{\theta}_0$

Inference

- For a new data point $x_{i'}$, predict $\hat{y}_{i'} = \hat{\theta}_1 x_{i'} + \hat{\theta}_0$