CS 412 — Introduction to Machine Learning (UIC)	February 11, 2025
Lecture [7]	
Instructor: Aadirupa Saha	*Scribe(s): Farhad Ferdowsi*

[This draft is not fully proofread. Please email any typos/errors to the instructor or directly edit the latex file.]

Overview

In the last lecture, we covered the following main topics:

1. MLE interpretation & Logistic regression
2. MLE interpretation of linear regression
3. MAP interpretation on regularized linear regression

This lecture focuses on:

1. Quick review of Bernoulli, Binomial, Categorical, and Multinomial Distribution
2. MLE interpretation of logistic regression
3. MAP interpretation on regularized logistic regression
4. Convex functions

1 Common Probability Distribution Functions in Machine Learning

In machine learning, feature vectors are numerical representations of data points that capture meaningful information for modeling and prediction. Different types of probability distributions play a key role in constructing and interpreting these feature vectors, as they describe the likelihood of various outcomes and help in probabilistic modeling. Below, we discuss how the commonly encountered probability distributions relate to different types of feature vectors in machine learning.

1.1 Bernoulli Distribution

The Bernoulli distribution is ideal for feature vectors containing binary attributes, where each feature takes values in 0,1. This is common in classification problems, such as representing the presence (1) or absence (0) of a feature in a dataset, like detecting spam in emails. Additionally, in logistic regression, the Bernoulli distribution underlies the probabilistic model for binary classification, where the probability of class membership is modeled using the sigmoid function.

Definition 1.1: Bernoulli distribution

The probability mass function (PMF) for a Bernoulli-distributed random variable X with success probability p is:

$$P(X = x) = p^x(1-p)^{1-x}, \quad x \in \{0, 1\} \tag{1}$$

Which simply means:

$$P(X = 1) = p, \quad P(X = 0) = 1 - p. \tag{2}$$

1.2 Binomial Distribution

The Binomial distribution applies to feature vectors where features represent counts of binary events across multiple trials. The binomial distribution extends the Bernoulli distribution by repeating the experiment multiple times. It models the number of successes in n independent trials, each with the same success probability p.

Definition 1.2: Binomial Distribution

The PMF for a binomial random variable X, with k representing the number of successes, is:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, 2, \ldots, n \tag{3}$$

where k is number of success, and $\binom{n}{k}$ is the binomial coefficient ("n choose k"), defined as:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}. \tag{4}$$

This distribution applies to scenarios like counting how many heads appear in n coin tosses or how many people in a group respond "yes" to a survey.

1.3 Categorical Distribution

The Categorical distribution is a generalization of the Bernoulli distribution for multiple categories. It applies to feature vectors with discrete, non-numeric values. Instead of just two outcomes (like success or failure), we have K possible categories (e.g., choosing a color from a set of colors), each with its own probability. In Categorical distribution instead of just two outcomes (like success or failure), we have K possible categories (e.g., choosing a color from a set of colors), each with its own probability.

Definition 1.3: Categorical distribution

If X can take one of K possible outcomes with probabilities $p_1, p_2, \ldots, p_K$, the categorical PMF is:

$$P(X = i) = p_i, \quad \text{for } i = 1, 2, \ldots, K \tag{5}$$

where the probabilities must sum to 1:

$$\sum_{i=1}^{K} p_i = 1. \tag{6}$$

1.4 Multinomial Distribution

The Multinomial distribution extends the binomial case to multiple categories and is used for feature vectors where features correspond to the number of times each category appears in multiple trials. It models the number of times each of K possible outcomes occurs over n independent trials.

Definition 1.4: Multinomial distribution

If $X_1, X_2, \ldots, X_K$ represent the counts of each outcome, and each outcome has probability $p_1, p_2, \ldots, p_K$, then the multinomial PMF is:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_K = x_K) = \frac{n!}{x_1!x_2! \ldots x_K!} p_1^{x_1} p_2^{x_2} \ldots p_K^{x_K} \tag{7}$$

where the counts sum to n:

$$\sum_{i=1}^{K} X_i = n. \tag{8}$$

This distribution is used for problems like rolling a die multiple times and tracking how often each face appears, or analyzing survey responses with multiple possible answers.

Remark 1. *The terms Probability Mass Function (PMF) and Probability Distribution Function are sometimes used interchangeably, but they have distinct meanings. A PMF applies to discrete random variables and gives the probability that the variable takes a specific value, denoted as $P(X = x)$. In contrast, a Probability Distribution Function is a more general term that describes the probability distribution of a random variable. For discrete variables, this can be the PMF, while for continuous variables, it is typically represented by the Probability Density Function (PDF) or the Cumulative Distribution Function (CDF). The word "distribution" can refer to both PMFs and PDFs, as it broadly describes how probabilities are assigned to different values of a random variable.*

2 Logistic Regression and Maximum Likelihood Estimation

Assume we have a dataset D defined as:

$$D = \{(X_i, y_i)\}_{i=1}^{N} \tag{9}$$

where $X_i \in \mathbb{R}^d$. Given that our task is binary classification, the labels y_i can take values of either 0 or 1:

$$y_i \in \{0, 1\} \tag{10}$$

We further assume that the labels y_i are generated from a Bernoulli distribution:

$$y_i \sim \text{Bernoulli}[\sigma(W^T X)] \tag{11}$$

where $\sigma(W^T X)$ represents the sigmoid function:

$$\sigma(W^T X) = \frac{1}{1 + e^{-W^T X}} \tag{12}$$

The sigmoid function is illustrated in Figure 1. To enable the model to predict the labels y, we need to determine an appropriate parameter W. This can be achieved by maximizing the likelihood function with respect to W. The likelihood function is defined as:

$$L_D(W) = \prod_{i=1}^{n} P(y_i|X_i; W) \tag{13}$$

Since working with sums is more convenient, we instead maximize the log-likelihood function:

$$W^* = \underset{W\in\mathbb{R}^d}{argmax}\ log(L_D(W)) = \underset{W\in\mathbb{R}^d}{argmax}\ \sum_{i=1}^{n} log(P(y_i|X_i; W)) \tag{14}$$

Using our assumption that y_i come from Bernoulli distribution $y_i \sim \text{Bernoulli}[\sigma(W^T X)]$, we can expand the expression as follows:

$$\begin{aligned} W^* &= \underset{W\in\mathbb{R}^d}{argmax}\ \sum_{i=1}^{n} log[\sigma(W^T X)^{y_i} + (1-\sigma(W^T X))^{1-y_i}] \\ &= \underset{W\in\mathbb{R}^d}{argmax}\ \sum_{i=1}^{n} \left[y_i log(\frac{1}{1+e^{-W^T X}}) + (1-y_i) log(1 - \frac{1}{1+e^{-W^T X}}) \right] \\ &= -\left[-\underset{W\in\mathbb{R}^d}{argmax}\ \sum_{i=1}^{n} \left[y_i log(\frac{1}{1+e^{-W^T X}}) + (1-y_i) log(\frac{1}{1+e^{W^T X}}) \right] \right] \end{aligned} \tag{15}$$

Since the negative of a maximization problem is equivalent to minimization, we rewrite the expression as:

$$W^* = \underset{W\in\mathbb{R}^d}{argmin}\ \sum_{i=1}^{n} \left[-y_i log(\frac{1}{1+e^{-W^T X}}) - (1-y_i) log(\frac{1}{1+e^{W^T X}}) \right] \tag{16}$$

which simplifies to:

$$W^* = \underset{W\in\mathbb{R}^d}{argmin}\ \sum_{i=1}^{n} \left[y_i log(1+e^{-W^T X}) + (1-y_i) log(1+e^{W^T X}) \right] \tag{17}$$

This optimization problem is precisely the one solved in logistic regression, where we define the logistic loss function as:

$$l_{logistic}(y, p) = -\sum [y_i log(p_i) + (1-y_i) log(1-p_i)] \tag{18}$$

where y_i is the true label and p_i is the predicted probability:

$$p_i = \sigma(W^T X_i) \tag{19}$$

Thus, minimizing the logistic loss over the dataset D is equivalent to maximizing the likelihood function with respect to W, under the assumption that y_i follows a Bernoulli distribution parameterized by a sigmoid function.

This provides the Maximum Likelihood Estimation interpretation of logistic regression. However, this formulation does not account for normalization or prior knowledge about the parameters. How can we incorporate regularization into the maximum likelihood framework?

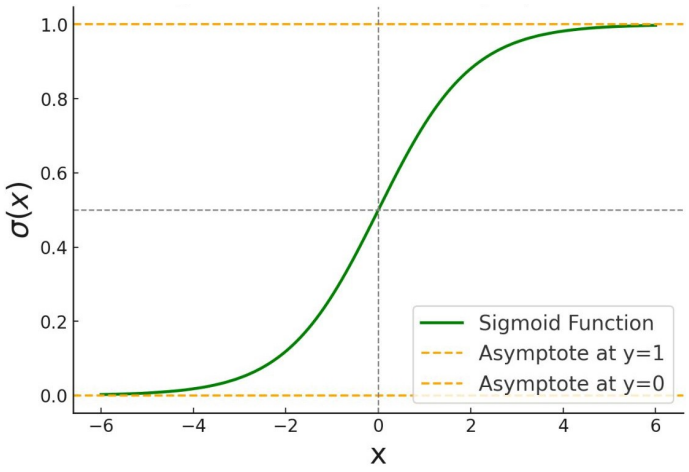

Figure 1: Sigmoid function

Regularization in logistic regression can be understood through the lens of Maximum A Posteriori (MAP) estimation. Instead of treating W as a parameter to be estimated solely from data, we introduce a prior distribution over W. This allows us to control the complexity of the model and prevent overfitting.
In some cases, we need to incorporate prior knowledge into our model. A prior serves a role similar to regularization, as it imposes constraints on the parameter space. If our dataset reflects real-world observations but we already have some prior belief (such as the assumption that a coin might have a certain bias) then the parameter W is not just an arbitrary estimate but instead follows a probability distribution.
To obtain the L1 or L2 regularized form of logistic loss, we must assume a prior distribution over W. Specifically:

- $L2$ regularization (Ridge) arises when we assume a Gaussian prior on W.
- $L1$ regularization (Lasso) arises when we assume a Laplacian prior on W.

By incorporating a prior, we move from a Maximum Likelihood Estimation (MLE) framework to a Maximum A Posteriori (MAP) estimation, which naturally leads to regularized logistic regression.

3 Regularization in Maximum Likelihood Estimation

In many cases, we introduce regularization to prevent overfitting and to incorporate prior knowledge about the parameters. Regularization in logistic regression can be interpreted in the framework of Maximum A Posteriori (MAP) estimation, where we assume a prior distribution over W.

3.1 L2 Regularization (Ridge Regularization)

For $L2$ regularization we assume that the weight vector W follows a Gaussian prior. The Gaussian Probability Density Function (PDF), also known as the normal distribution, is a fundamental concept in probability and statistics. It describes how the values of a random variable are distributed in a symmetrical, bell-shaped curve.

Definition 3.1: One dimensional Gaussian probability density function

In one dimension, a Gaussian PDF is denoted as $\mathcal{N}(\mu, \sigma^2)$ where μ is the mean (center) of the distribution and σ^2 is the variance, representing the spread of the distribution. If a random variable x follows this distribution, its probability density function is given by:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right); \quad \mu, \sigma \in \mathbb{R} \tag{20}$$

This can be extended to multiple dimensions, leading to the multivariate Gaussian distribution.

Definition 3.2: multivariate Gaussian probability density function

In multiple dimensions, multivariate Gaussian PDF denoted as $\mathcal{N}(\mu, \Sigma)$ where μ is an d-dimensional mean vector that represents the expected value of X:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix} \tag{21}$$

and Σ is a $d \times d$ covariance matrix that describes variances and covariances between variables. If a d-dimensional random vector $X = (x_1, x_2, ..., x_d)$ follows this distribution, its probability density function is given by:

$$P(X) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right); \quad \mu \in \mathbb{R}^d, \sigma \in \mathbb{R}^{d\times d} \tag{22}$$

Where $|\Sigma|$ is the determinant of the covariance matrix, and Σ^{-1} is the inverse of the covariance matrix.

Remark 2. *In multivariate Gaussian PDF, if Σ is diagonal, then the components of X are uncorrelated and independent.*

For $L2$ regularization, we assume that the weight vector W follows a d-dimensional Gaussian prior:

$$W \sim \mathcal{N}(0, \lambda I_{d\times d}) \tag{23}$$

Here we set μ to zero vector and The covariance matrix Σ is assumed to be a scaled identity matrix $\lambda I_{d\times d}$ where λ is a positive scalar. This implies that the elements of W are independent and have the same variance λ. Thus, the covariance matrix takes the following diagonal form:

$$\Sigma_{d\times d} = \begin{bmatrix} \lambda & 0 & 0 & \cdots & 0 \\ 0 & \lambda & 0 & \cdots & 0 \\ 0 & 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda \end{bmatrix}_{d\times d} \tag{24}$$

To estimate the Maximum A Posteriori (MAP) value of W, we need to find the W that maximizes the posterior probability $P(W|D)$:

$$W^* = \underset{W \in \mathbb{R}^d}{argmax}\ P(W|D) \tag{25}$$

Using **Bayes' theorem**, we can express this as:

$$W^* = \underset{W \in \mathbb{R}^d}{argmax}\ P(W|D) = \underset{W \in \mathbb{R}^d}{argmax}\ (P(D|W)P(W)) \tag{26}$$

Since maximizing a function is equivalent to maximizing its logarithm, we take the log transformation:

$$W^* = \underset{W \in \mathbb{R}^d}{argmax}\ log(P(W|D)) = \underset{W \in \mathbb{R}^d}{argmax}\ [(log(P(D|W)) + log(P(W))] \tag{27}$$

The first term, $P(D|W)$, is computed using the Maximum Likelihood Estimation. Thus, we have to analyze the prior probability $P(W)$. Given our covariance matrix Σ represented in equation (24), and noting that for this covariance matrix:

$$\Sigma^{-1} = \frac{1}{\lambda} I_{d \times d}, \quad |\Sigma| = \lambda \tag{28}$$

the prior probability distribution of W is:

$$P(W) = \frac{1}{(2\pi)^{d/2}\lambda} \exp(-\frac{1}{2} W^T \frac{I_{d \times d}}{\lambda} W) \tag{29}$$

Since $I_{d \times d} W = W$ we get:

$$W^T I_{d \times d} W = W^T W = \|W\|_2^2 \tag{30}$$

Therefor, we have:

$$\begin{aligned} W^* &= \underset{W \in \mathbb{R}^d}{argmax}\ [(log(P(D|W)) + log(P(W))] \\ &= \underset{W \in \mathbb{R}^d}{argmax}\ \sum_{i=1}^{n} log(P(y_i|X_i; W) + \left[log \frac{1}{(2\pi)^{d/2}\lambda} - \frac{\|W\|_2^2}{2\lambda} \right] \\ &= \underset{W \in \mathbb{R}^d}{argmax}\ \sum_{i=1}^{n} - \left[y_i log(1 + e^{-W^T X_i}) + (1 - y_i) log(1 + e^{W^T X_i}) \right] + \left[log \frac{1}{(2\pi)^{d/2}\lambda} - \frac{\|W\|_2^2}{2\lambda} \right] \end{aligned} \tag{31}$$

Since the term $log(\frac{1}{(2\pi)^{d/2})\lambda})$ is a constant, it does not affect the maximization process and can therefore be ignored. By multiplying the expression by -1, we convert this maximization problem into a minimization one, which results in the $L2$-regularized logistic regression formulation:

$$W^* = \underset{W \in \mathbb{R}^d}{argmin}\ \sum_{i=1}^{n} \left[y_i log(1 + e^{-W^T X_i}) + (1 - y_i) log(1 + e^{W^T X_i}) \right] + \frac{\|W\|_2^2}{2\lambda} \tag{32}$$

For clarity, the regularization term is given by:

$$\frac{1}{2\lambda} \|W\|_2^2 = \frac{1}{2\lambda} \sum_{j=1}^{d} W_j^2 \tag{33}$$

Where W_j represents the weight of the j-th feature and λ is a tuning parameter that controls the strength of the regularization. Smaller values of λ encourage smaller weights, leading to simpler models.

This Gaussian prior assumption plays a crucial role in $L2$ regularization, as it encourages smaller weight values and helps prevent overfitting by penalizing large coefficients.

3.2 L1 Regularization (Lasso Regularization)

For $L1$ regularization, we assume that the weight vector W follows a Laplace prior. The Laplace Probability Density Function is commonly used in Bayesian statistics to impose sparsity on parameter estimates. The Laplace distribution has sharper peaks at zero compared to the Gaussian distribution, encouraging many weights to be exactly zero.

Definition 3.3: One dimensional Laplace probability density function

In one dimension, a Laplace PDF is denoted as $Laplace(\mu, b)$, where μ is the location parameter (center of the distribution) and b is the scale parameter, which controls the spread of the distribution. If a random variable x follows this distribution, its probability density function is given by:

$$P(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right); \quad \mu, b \in \mathbb{R} \tag{34}$$

This can be extended to multiple dimensions, leading to the multivariate Laplace distribution

Definition 3.4: multivariate Laplace probability density function

In multiple dimensions, multivariate Laplace PDF denoted as $Laplace(\mu, B)$, where μ is a d -dimensional mean vector:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix} \tag{35}$$

and B is a $d \times d$ scale matrix, which describes the dispersion of the distribution. If a d-dimensional random vector $X = (x_1, x_2, ..., x_d)$ follows this distribution, for the case of diagonal scale matrix B, its probability density function is given by:

$$P(X) = \frac{1}{2^d|B|} \exp\left(-\sum_{i=1}^{d} \frac{|x_i - \mu_i|}{b_i}\right); \quad \mu \in \mathbb{R}^d,\ B \in \mathbb{R}^{d\times d} \tag{36}$$

Where the term b_i represents the scale parameter associated with the $i-th$ component of the random vector X. It controls the dispersion (spread) of the $i-th$ variable around its mean μ_i. $|B|$ is the determinant of the scale matrix.

Remark 3. *In multivariate Laplace PDF, if B is diagonal, then the components of X are uncorrelated and independent.*

Remark 4. *Dispersion of the distribution, generally refers to how widely or narrowly the values are spread out from the center, and in the case of the Laplace prior in L1 regularization, it controls the likelihood of obtaining sparse weight vectors.*

For $L1$ regularization, we assume that the weight vector W follows a d-dimensional Laplace prior:

$$W \sim Laplace(0, \lambda I_{d\times d}) \tag{37}$$

Here as well, we set μ to the zero vector, and the scale matrix B is assumed to be a scaled identity matrix $\lambda I_{d\times d}$, where λ is a positive scalar. This implies that the elements of W are independent and have the same variance, leading to the following diagonal form:

$$B_{d\times d} = \begin{bmatrix} \lambda & 0 & 0 & \cdots & 0 \\ 0 & \lambda & 0 & \cdots & 0 \\ 0 & 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda \end{bmatrix}_{d\times d} \tag{38}$$

In this case as well, in order to estimate the Maximum A Posteriori (MAP) value of W, we need to find the W that maximizes the posterior probability $P(W|D)$:

$$W^* = \arg\max_{W\in\mathbb{R}^d} P(W|D) \tag{39}$$

By considering Laplace prior probability distribution for W and Considering our choice for the B matrix in equation (38), and knowing that for this matrix determinant is equal to λ, the probability distribution of W is as follows:

$$P(W) = \frac{1}{2^d\lambda} \exp\left(-\frac{\|W\|_1}{\lambda}\right) \tag{40}$$

where $\|W\|_1 = \sum_{j=1}^{d} |W_j|$ represents the $L1$ norm of W. By Using Bayes' theorem, and following a similar derivation as for $L2$ regularization, we can derive $L1$-regularized logistic regression formulation as follows:

$$W^* = \arg\max_{W\in\mathbb{R}^d} \left[\sum_{i=1}^{n} \log P(y_i|X_i;W) + \log\left(\frac{1}{2^d\lambda}\right) - \frac{\|W\|_1}{\lambda}\right] \tag{41}$$

By ignoring $\log(1/2^d\lambda)$ and converting this maximization problem into a minimization one, $L1$-regularized logistic regression formulation is:

$$W^* = \arg\min_{W\in\mathbb{R}^d} \sum_{i=1}^{n} \left[y_i \log(1 + e^{-W^T X_i}) + (1 - y_i) \log(1 + e^{W^T X_i})\right] + \frac{\|W\|_1}{\lambda} \tag{42}$$

For clarity, the regularization term is given by:

$$\frac{1}{\lambda}\|W\|_1 = \frac{1}{\lambda}\sum_{j=1}^{d} |W_j| \tag{43}$$

where W_j represents the weight of the j-th feature, and λ is a tuning parameter that controls the strength of the regularization.

Smaller values of λ encourage sparser weights, leading to feature selection by forcing some weights to be exactly zero. Larger values of λ result in more nonzero weights, leading to a denser model. Thus, $L1$ regularization not only prevents overfitting but also enforces sparsity, making it useful for feature selection. To conclude this section, Table **??** provides a comparative summary of $L1$ (Lasso) and $L2$ (Ridge) regularization, outlining their key differences and respective advantages.

Feature	**L1 Regularization (Lasso)**	**L2 Regularization (Ridge)**
Prior Assumption	Laplacian distribution: $W \sim \text{Laplace}(0, \lambda I)$	Gaussian distribution: $W \sim N(0, \lambda I)$
Regularization Term	$\frac{1}{\lambda} \sum \lvert W_j \rvert$ (absolute values of weights)	$\frac{1}{2\lambda} \sum W_j^2$ (squared values of weights)
Effect on Weights	Induces sparsity: some weights become exactly zero	Shrinks all weights smoothly, but none become zero
Feature Selection	Can be used for feature selection (eliminates irrelevant features)	Retains all features but reduces their impact
Optimization	Non-differentiable at zero (requires subgradient methods)	Differentiable everywhere (easier to optimize with gradient-based methods)
Bias-Variance Tradeoff	Higher bias, lower variance (better for sparse models)	Lower bias, higher variance (better for dense models)
When to Use?	When feature selection is important (reducing irrelevant features)	When all features contribute to prediction but need to be controlled
Interpretability	Easier to interpret due to zero coefficients	Harder to interpret as all coefficients remain nonzero
Computational Complexity	Can be computationally expensive for high-dimensional data due to feature selection	Computationally more stable and efficient in high dimensions

Table 1: Comparison of L1 (Lasso) and L2 (Ridge) Regularization

4 Convex functions

In machine learning we always encounter with minimizing task which we have to find value or vectors which minimize a function typically Loss functions or inverse of MLE or MAP functions we saw in previouse section. Gradient descent is one of the most effective to fullfill this goal. However, this method only works well if certain conditions are met and convexity of the loss function is one of these conditions. Therefore, before explaining the gradient descent method, we need to elaborate on convexity. Convexity of a function means that that function dont have local minimums and has only global minumum. this ensures that in the process of minimizing does not get stuck in local minimums.
In machine learning, we frequently encounter optimization problems where we aim to minimize a function, such as a loss function or the negative log-likelihood derived from Maximum Likelihood Estimation or Maximum A Posteriori estimation. Many of the techniques we use, such as gradient descent, are effective only under certain conditions, and one of the most critical conditions is convexity. Convexity of a function ensures that the function lies below the straight line connecting any two points on its graph, forming a ”bowl” shape. If the inequality is strict for all $x_1 \neq x_2$, the function is strictly convex, meaning it has a unique

global minimum. Convexity is crucial because it guarantees that the function has no local minima apart from the global minimum, ensuring that gradient-based methods like gradient descent will not get trapped in suboptimal points. Additionally, optimization methods remain efficient and converge reliably, providing computational stability when finding optimal parameters.
For example, in logistic regression, we optimize the negative log-likelihood function which is convex in W, meaning gradient descent will efficiently find the optimal parameters. When we add regularization terms to logistic regression, convexity is preserved, which ensures stable and efficient optimization. How can we deteremine if a function is convex?

To discuss the main definition of a convex function, it is useful to begin by introducing the concept of a convex combination.

Definition 4.1: Convex combination

For any two fixed points $x_1, x_2 \in \mathbb{R}$, we define the function $A(\lambda)$ as follows:

$$A(\lambda, x_1, x_2) = \lambda x_1 + (1-\lambda)x_2; \quad \lambda \in [0,1] \tag{44}$$

This function is called a convex combination.

This function generates any point between x_1 and x_2 by varying λ within the interval $[0, 1]$. For example the function generates x_2 when $\lambda = 0$ and x_1 when $\lambda = 1$.

$$A(\lambda, x_1, x_2) = \begin{cases} x_1, & \lambda = 1 \\ x_2, & \lambda = 0 \end{cases} \tag{45}$$

For any intermediate value of λ, the function produces a point between x_1 and x_2. By choosing an appropriate λ, one can obtain any desired interpolation between these two points.

4.1 Convexity main definition

The main definition of convex functions can be expressed as follows:

Definition 4.2: Convexity main definition

A one dimenensional function $f : D \to \mathbb{R}$ is convex in a specific domain $D \subseteq \mathbb{R}$ if, for every $x_1, x_2 \in D$ and $0 \leq \lambda \leq 1$, the following inequality holds:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \tag{46}$$

It is strictly convex if, for every $x_1, x_2 \in D$ and $0 \leq \lambda \leq 1$, the following stricter inequality holds:

$$f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda) f(x_2) \tag{47}$$

Remark 5. *Using the concept of a convex combination, the convexity condition can be equivalently expressed as:*

$$f(A(\lambda, x_1, x_2)) \leq A(\lambda, f(x_1), f(x_2)); \quad \lambda \in [0,1],\ x_1, x_2 \in \mathbb{R} \tag{48}$$

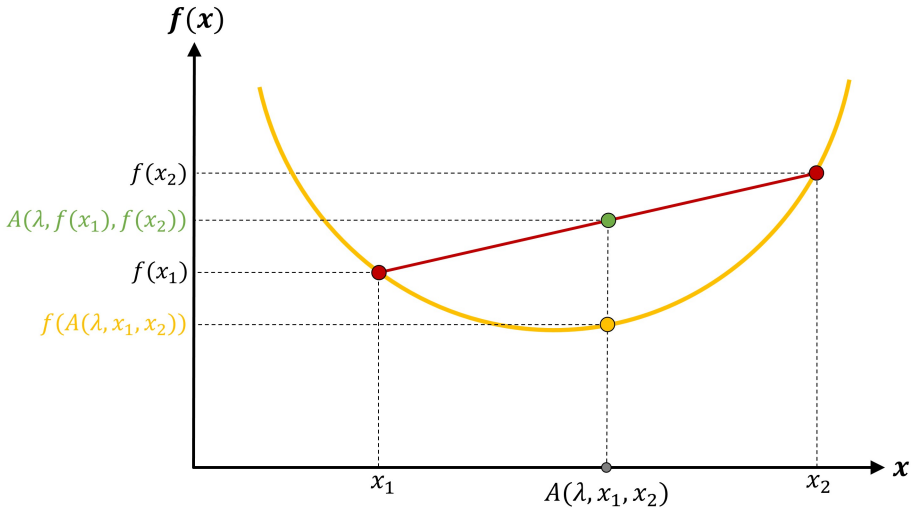

Figure 2: Graphical representation of a convex function

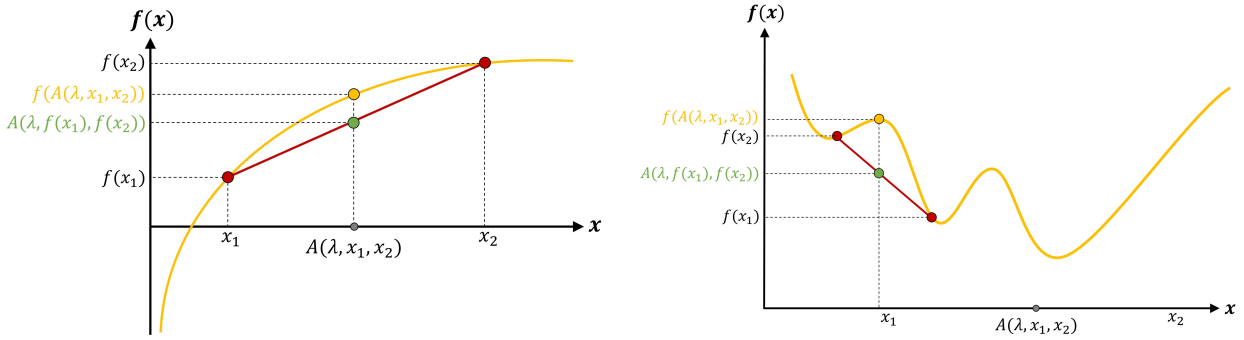

Figure 3: Graphical representation of Non-Convex functions

This condition implies that, within the convexity domain of the function, the function's value at any convex combination of two points is always less than or equal to the convex combination of their function values. Geometrically, this means that the graph of a convex function always lies on or below the straight line connecting any two points on the function. This is illustrated in the Figure 2 where the yellow curve represents the convex function $f(x)$, two points, $(x_1, f(x_1))$ and $(x_2, f(x_2))$, are chosen on the function and the red line segment connecting these points represents the convex combination of their function values. The yellow dot shows the function's value at an intermediate point, while the green dot represents the convex combination of the function values. Since the yellow dot is always below (or on) the red line, the convexity condition is visually demonstrated. Further examples are shown in Figure 3. Unlike the previous case, in these examples, there exist points on the straight line connecting $f(x_1)$ and $f(x_2)$ that lie below (or on) the function's value at a corresponding x. This indicates that these curves are not convex, as they fail to satisfy the convexity condition.

It is evident that the convexity of a function should be rigorously examined using mathematical conditions, as this is the most reliable method. This approach will be discussed in more detail in the next session. However, a simple rule of thumb for visually assessing convexity from a graph is to check whether any straight line

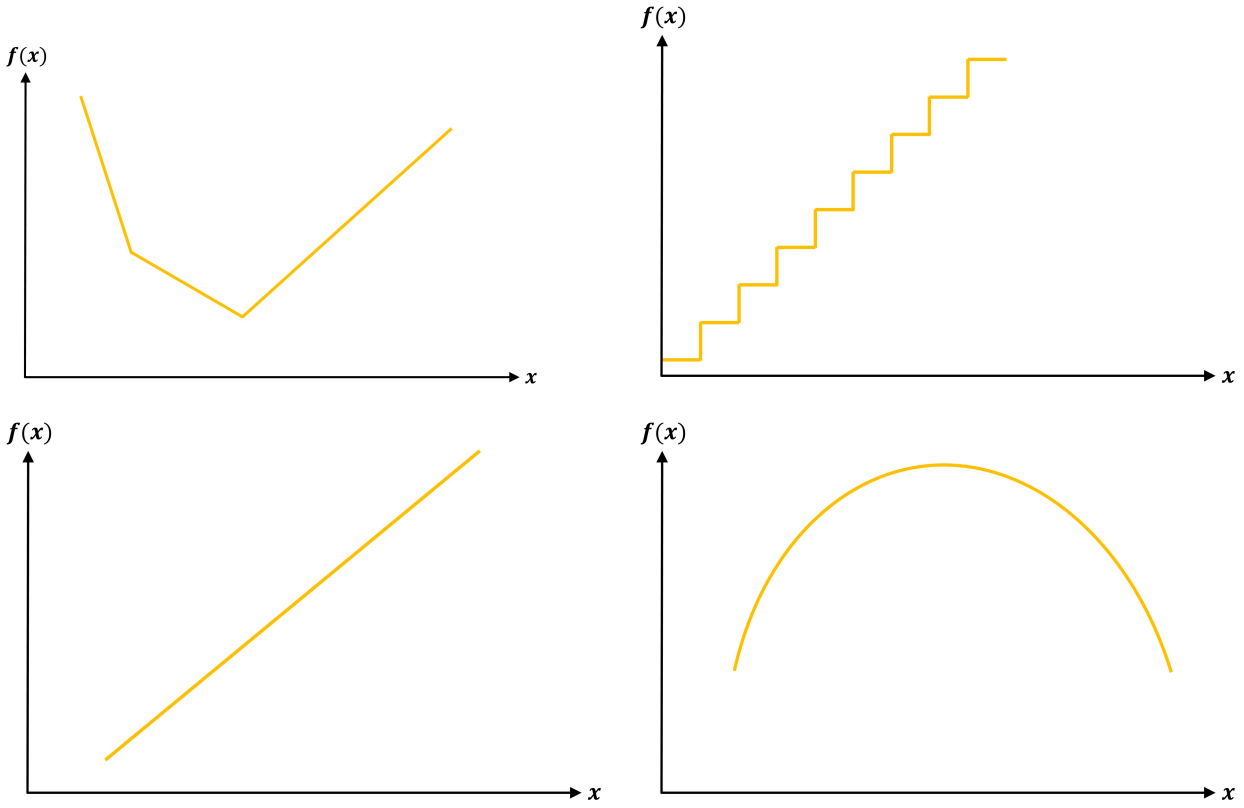

Figure 4: Convexity Check Exercise

drawn between two points on the curve lies at least partly below the function or crosses it. If this occurs, the function is not convex.
As an exercise, try checking the convexity of some graphs shown in figure 4.

Exercise 4.1: Convexity

Determine whether the graphs in figure 4 are convex.

Next Lecture

The next lecture will cover the following topics:
(i) Convexity,
(ii) Key properties of a loss function that improve gradient descent convergence.,
(iii) Introduction to Gradient Descent .

References:

1. Book ”Pattern Recognition and Machine Learning” by Christopher M. Bishop, Chapter 2: Probability Distributions, Chapter 4:Linear models for classification.

2. Book ”Mathematical Statistics and Data Analysis” by John A. Rice, Chapter 2: Random Variables.
3. Book ”The Elements of Statistical Learning” by Hastie, Tibshirani, and Friedman, Chapter 4: Linear Methods Classification.
4. Book ”Machine Learning: A Probabilistic Perspective” by Kevin P. Murphy, Chapter 8: Bayesian Logistic Regression.
5. Lecture note by Andrew Ng on Logistic Regression, MLE for logistic regression.
6. Lecture note by Henry Chai & Zack Lipton on MLE & MAP, Machine Learning Lecture 6