Lecture 2

谢丹
清华大学数学系

September 17, 2025

Foundations of Probability Theory I

Key Concepts and Definitions

Probability Density Function (PDF)

A probability distribution for a continuous random variable X is described by a function $p(x)$ satisfying:

$$p(x) \geq 0 \quad \text{(Non-negativity)}$$

$$\int p(x)dx = 1 \quad \text{(Normalization)}$$

Key Densities for Multivariate Distributions

Foundations of Probability Theory II

Key Concepts and Definitions

For two random variables X and Y, we define:

- **Joint Density:** $p(x, y)$
- **Marginal Density:** $p(x) = \int p(x, y)dy$ ("Summing out" the other variable)
- **Conditional Density:** $p(x|y)$ and $p(y|x)$

The Fundamental Product Rule

The relationship between these densities is given by:

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x)$$

This rule is the foundation for **Bayes' Theorem**:
$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$.

Important Characteristics of a Distribution

Foundations of Probability Theory III

Key Concepts and Definitions

- **Mean (μ)**: Expected value, measuring central tendency. $\mathbb{E}[X] = \mu = \int x p(x) dx$
- **Variance (σ^2)**: Measures the spread or dispersion around the mean. $\text{Var}(X) = \sigma^2 = \int (x - \mu)^2 p(x) dx$
- **Entropy (H)**: Measures the average uncertainty or information content. $H(X) = -\sum p(x) \log_2 p(x)$ (for discrete variables). We have joint entropy $H(X, Y)$, conditional entropy $H(X|Y), H(Y|X)$ and the mutual information $I(X; Y)$. The relative entropy $D(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} dx$ plays a crucial role.

The use of entropy

Theorem (Maximum Entropy Distribution)

Among all continuous probability distributions $p(x)$ *with a fixed mean* μ *and variance* σ^2*, the Gaussian distribution*

$$q(x) = \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

achieves the **maximum differential entropy***:*

$$h(p) = -\int p(x) \log p(x) dx \leq \frac{1}{2} \log(2\pi e\sigma^2) = h(q)$$

Equality holds if and only if $p(x) = q(x)$*.*

$\Rightarrow$ The Gaussian is the **least informative** distribution for a given mean and variance.

Proof Setup: KL Divergence

The proof uses the non-negativity of the **Kullback-Leibler (KL) Divergence**.

KL Divergence

The KL divergence from q to p measures the "distance" between distributions:

$$D_{\mathsf{KL}}(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

Its key property is:

$$D_{\mathsf{KL}}(p \parallel q) \geq 0$$

with equality *if and only if* $p(x) = q(x)$ almost everywhere.

Step 1: Expand the KL Divergence

Let's expand $D_{\mathsf{KL}}(p \parallel q)$ for our target Gaussian q:

$$\begin{aligned} D_{\mathsf{KL}}(p \parallel q) &= \int p(x) \log \frac{p(x)}{q(x)} dx \\ &= \int p(x) \log p(x) dx - \int p(x) \log q(x) dx \\ &= -h(p) - \int p(x) \log q(x) dx \end{aligned}$$

Since $D_{\mathsf{KL}}(p \parallel q) \geq 0$, we have:

$$-h(p) - \int p(x) \log q(x) dx \geq 0 \quad \Rightarrow \quad h(p) \leq - \int p(x) \log q(x) dx \tag{1}$$

$\Rightarrow$ We now have an **upper bound** for $h(p)$.

Step 2: Compute the Upper Bound

We need to compute $-\int p(x) \log q(x) dx$. First, write down $\log q(x)$ for the Gaussian $q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$:

$$\begin{aligned}\log q(x) &= \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{(x-\mu)^2}{2\sigma^2} \\ &= -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}\end{aligned}$$

Now plug this into the integral:

$$\begin{aligned}-\int p(x) \log q(x) dx &= -\int p(x)\left(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}\right) dx \\ &= \int p(x)\left(\frac{1}{2}\log(2\pi\sigma^2) + \frac{(x-\mu)^2}{2\sigma^2}\right) dx\end{aligned}$$

Step 3: Use the Constraints

Distribute the integral and use the constraints on $p(x)$:

$$-\int p(x)\log q(x)dx = \frac{1}{2}\log(2\pi\sigma^2)\int p(x)dx + \frac{1}{2\sigma^2}\int p(x)(x-\mu)^2dx$$

By definition, our distribution $p(x)$ satisfies:

- $\int p(x)dx = 1$ (Normalization)
- $\int p(x)(x-\mu)^2dx = \sigma^2$ (Definition of Variance)

Substituting these in:

$$\begin{aligned}-\int p(x)\log q(x)dx &= \frac{1}{2}\log(2\pi\sigma^2)\cdot 1 + \frac{1}{2\sigma^2}\cdot\sigma^2 \\ &= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}\end{aligned}$$

Step 4: Final Manipulation and Result

Simplify the expression:

$$\begin{aligned}\frac{1}{2}\log(2\pi\sigma^2)+\frac{1}{2} &= \frac{1}{2}\left(\log(2\pi\sigma^2)+1\right)\\ &= \frac{1}{2}\left(\log(2\pi\sigma^2)+\log(e)\right) \quad \text{(since } 1=\log e\text{)}\\ &= \frac{1}{2}\log(2\pi e\sigma^2)\end{aligned}$$

But this is precisely the differential entropy of the Gaussian distribution $q(x)$:

$$h(q)=\frac{1}{2}\log(2\pi e\sigma^2)$$

Step 5: Conclusion

Recall our inequality from Step 1:

$$h(p) \leq -\int p(x) \log q(x) dx$$

We have just shown that:

$$-\int p(x) \log q(x) dx = \frac{1}{2} \log(2\pi e \sigma^2) = h(q)$$

Therefore, we conclude:

$$h(p) \leq h(q)$$

Equality Condition

Equality holds if and only if $D_{\mathsf{KL}}(p \parallel q) = 0$, which happens *if and only if* $p(x) = q(x)$ almost everywhere.

Important Probability Distributions

1. **Gaussian (Normal) Distribution**:

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

   Mean μ, variance σ^2, which are the parameters.

2. **Bernoulli Distribution** (discrete):

$$\text{Ber}(x|\mu) = \mu^x(1-\mu)^{1-x}$$

   - $P(x=0) = 1-\mu$, $P(x=1) = \mu$
   - Mean μ, variance $\mu(1-\mu)$

Multivariate Distributions

Multivariate (n dimensional) Gaussian

$$P(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Mean vector $\boldsymbol{\mu}$ (n dimensional vector), covariance matrix Σ ($n \times n$ matrix).

Categorical Distribution

For K classes:

$$P(t=i) = p_i \quad (i = 1, \ldots, K), \quad \sum_{i=1}^{K} p_i = 1$$

Compact representation:

$$P(\mathbf{t}) = \prod_{i}^{K} p_i^{t_i}$$

One-Hot Encoding Representation

- $\mathbf{t}$ is a one-hot encoded n-dimensional vector
- t_i denotes the i-th component of $\mathbf{t}$

Examples:

- Class 1: $\mathbf{t} = [1, 0, 0, \ldots, 0]$
- Class 2: $\mathbf{t} = [0, 1, 0, \ldots, 0]$
- Class K: $\mathbf{t} = [0, 0, \ldots, 1]$

Exactly one component is 1, all others are 0

Normal Distribution

A cornerstone of continuous multivariate probability

A random vector $\mathbf{X} = [X_1, X_2, ..., X_D]^T$ follows a multivariate Gaussian distribution if its probability density function (PDF) is:

Probability Density Function (PDF)

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

- $\boldsymbol{\mu} \in \mathbb{R}^D$: Mean vector (center of the distribution)
- $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$: Covariance matrix (symmetric, positive definite)
  - Diagonal elements Σ_{ii}: Variances of each variable X_i
  - Off-diagonal elements Σ_{ij}: Covariance between variables X_i and X_j

Notation

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Key Properties and Partitioning the Vector

The distribution is fully defined by its mean and covariance:

$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu}$$
$$\text{Cov}[\mathbf{X}] = \boldsymbol{\Sigma}$$

Partitioning the Vector and Matrices

To analyze marginals and conditionals, we partition the vector and its parameters:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_a \\ \mathbf{X}_b \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix}$$

- $\mathbf{X}_a$ is $p \times 1$, $\mathbf{X}_b$ is $q \times 1$ $(p + q = D)$.
- $\boldsymbol{\Sigma}_{aa}$: Covariance of $\mathbf{X}_a$.
- $\boldsymbol{\Sigma}_{bb}$: Covariance of $\mathbf{X}_b$.
- $\boldsymbol{\Sigma}_{ab} = \boldsymbol{\Sigma}_{ba}^T$: Cross-covariance between $\mathbf{X}_a$ and $\mathbf{X}_b$.

Marginal Distributions

The distribution of a subset of variables

Theorem (Marginal is Gaussian)

If $\begin{bmatrix} \mathbf{X}_a \\ \mathbf{X}_b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \right)$, *then the marginal distributions are also Gaussian:*

$$p(\mathbf{X}_a) = \mathcal{N}(\mathbf{X}_a | \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$$
$$p(\mathbf{X}_b) = \mathcal{N}(\mathbf{X}_b | \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb})$$

Interpretation

To get the marginal distribution of any subset of variables:

1. Extract the corresponding subvector from the mean $\boldsymbol{\mu}$.
2. Extract the corresponding submatrix from the covariance $\boldsymbol{\Sigma}$.

The marginal distribution **ignores** (integrates out) the other variables but retains their influence via the covariances in its own submatrix.

Conditional Distributions

The distribution of a subset *given* the others

Theorem (Conditional is Gaussian)

The conditional distribution $p(\mathbf{X}_a|\mathbf{X}_b)$ *is also a Gaussian:*

$$p(\mathbf{X}_a|\mathbf{X}_b) = \mathcal{N}(\mathbf{X}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$$

with parameters:

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{X}_b - \boldsymbol{\mu}_b)$$
$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}$$

Interpretation

- The **conditional mean** $\boldsymbol{\mu}_{a|b}$ is a linear function of the value $\mathbf{X}_b$.
- The **conditional covariance** $\boldsymbol{\Sigma}_{a|b}$ is *constant* (it does not depend on the value of $\mathbf{X}_b$). This is a special property of the Gaussian distribution.

Summary: The Multivariate Gaussian

- **Defined by**: Mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
- **Linear Transformations**: Any linear transformation of a Gaussian vector is itself Gaussian.
  $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b} \implies \mathbf{Y} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T)$.
- **Marginal Distributions**: Are Gaussian. Their parameters are found by **selecting** the relevant sub-vectors and sub-matrices of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- **Conditional Distributions**: Are Gaussian. Their parameters are found by a **matrix inversion and multiplication** on the blocks of $\boldsymbol{\Sigma}$.
- **Special Case: Independence**: If $\boldsymbol{\Sigma}_{ab} = \mathbf{0}$ (blocks are independent), then $p(\mathbf{X}_a|\mathbf{X}_b) = p(\mathbf{X}_a)$ and $\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a$, $\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa}$.

The family of Gaussian distributions is closed under marginalization and conditioning.

Table: Common Probability Distributions

Distribution	**Probability Mass/Density Function (PMF/PDF)**	**Parameters & Support**	**Mean & Variance**
Discrete Distributions			
Bernoulli	$P(X = x) = \begin{cases} p & \text{if } x = 1 \\ 1-p & \text{if } x = 0 \end{cases}$	$p \in [0,1]$ (success prob.) $x \in \{0,1\}$	$\mu = p$ $\sigma^2 = p(1-p)$
Binomial	$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$	$n \in \mathbb{N}$ (number of trials) $p \in [0,1]$ (success prob.) $k \in \{0,1,...,n\}$	$\mu = np$ $\sigma^2 = np(1-p)$
Poisson	$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$	$\lambda > 0$ (rate) $k \in \mathbb{Z}_{\geq 0}$	$\mu = \lambda$ $\sigma^2 = \lambda$
Geometric	$P(X = k) = (1-p)^{k-1} p$	$p \in (0,1]$ (success prob.) $k \in \mathbb{Z}^+$ (# trials until success)	$\mu = \frac{1}{p}$ $\sigma^2 = \frac{1-p}{p^2}$
Negative Binomial	$P(X = k) = \binom{k-1}{r-1} (1-p)^{k-r} p^r$	$r \in \mathbb{Z}^+$ (# successes) $p \in (0,1]$ (success prob.) $k \in \{r, r+1, ...\}$	$\mu = \frac{r}{p}$ $\sigma^2 = \frac{r(1-p)}{p^2}$

Distribution	Probability Mass/Density Function (PMF/PDF)	Parameters & Support	Mean & Variance
Continuous Distributions			
Uniform	$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$	$a, b \in \mathbb{R},\ a < b$ $x \in [a,b]$	$\mu = \frac{a+b}{2}$ $\sigma^2 = \frac{(b-a)^2}{12}$
Normal (Gaussian)	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$	$\mu \in \mathbb{R}$ (location) $\sigma > 0$ (scale) $x \in \mathbb{R}$	μ σ^2
Exponential	$f(x) = \lambda e^{-\lambda x}$	$\lambda > 0$ (rate) $x \geq 0$	$\mu = \frac{1}{\lambda}$ $\sigma^2 = \frac{1}{\lambda^2}$
Gamma	$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$	$\alpha > 0$ (shape), $\beta > 0$ (rate) $x > 0$	$\mu = \frac{\alpha}{\beta}$ $\sigma^2 = \frac{\alpha}{\beta^2}$
Beta	$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$	$\alpha > 0, \beta > 0$ (shape) $x \in [0,1]$	$\mu = \frac{\alpha}{\alpha+\beta}$ $\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

Modeling High-Dimensional Distributions

We have described families of one-dimensional distributions. For higher-dimensional distributions—besides the multivariate Gaussian—we will primarily consider the following models:

1. **Independent and Identically Distributed (i.i.d.)**
   The random variables are mutually independent and share the same distribution parameterized by w.

$$p(x_1, \ldots, x_n \mid w) = \prod_{i=1}^{n} p(x_i \mid w)$$

2. **Markov Chains**
3. **Graphical Models** (Probabilistic models encoded by a graph)

We will discuss Markov chains and graphical models later. For now, our focus will be on the first class: **i.i.d. models**.

What does i.i.d. mean?

i.i.d. stands for Independent and Identically Distributed.

A Fundamental Assumption

It is a common and crucial assumption about a collection of random variables in statistics and machine learning.

Let's break it down for a sequence of random variables $X_1, X_2, X_3, \ldots, X_n$.

Part 1: Identically Distributed (i.d.)

All variables follow the same probability distribution.

- They have the same mean:
  $E[X_1] = E[X_2] = \ldots = E[X_n] = \mu$
- They have the same variance:
  $\text{Var}(X_1) = \text{Var}(X_2) = \ldots = \text{Var}(X_n) = \sigma^2$
- They have the same underlying probability law (PDF/PMF).

Example:

- $X_1, X_2, \ldots, X_{10}$ represent 10 rolls of a fair die.
- Each X_i has PMF: $P(X_i = k) = \frac{1}{6}$ for $k = 1, 2, \ldots, 6$.
- They are *identically distributed*.

Part 2: Independent (i.)

The outcome of one variable does not affect the others.

- Knowing the value of X_j tells you *nothing* about X_i (for $i \neq j$).
- Joint probability is the product of individual probabilities.

For any two variables X_i and X_j and values a, b:

$$P(X_i \leq a,\ X_j \leq b) = P(X_i \leq a) \cdot P(X_j \leq b)$$

Example (cont.):

- The result of the nth1 die roll does not influence the nth5 roll.
- $P(X_1 = 1,\ X_5 = 6) = P(X_1 = 1) \cdot P(X_5 = 6) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$.
- They are *independent*.

Putting It All Together: i.i.d.

If our die rolls are both Independent *and* Identically Distributed, they form an **i.i.d. sequence**.

Formal Definition

The sequence $X_1, X_2, \ldots, X_n$ is i.i.d. if:

1. $(X_1, X_2, \ldots, X_n)$ are mutually **independent**.
2. All X_i are drawn from the same distribution F (they are **identically distributed**).

Why is the i.i.d. Assumption Important?

It simplifies mathematics and is the foundation of many key theorems.

Law of Large Numbers

For an i.i.d. sequence with mean μ:

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{P} \mu$$

The sample average converges to the true mean.

Central Limit Theorem

For an i.i.d. sequence with mean μ and variance σ^2:

$$\sqrt{n}(\bar{X}_n - \mu) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

The sample mean is approximately normally distributed.

This assumption is critical for many statistical inferences and machine learning algorithms.

The Bayesian Perspective I

Machine learning models are often based on a parameterized probabilistic model. The central goal is to estimate the parameters of this model from observed data.

A systematic approach is Bayesian estimation, which treats the parameters $\mathbf{w}$ as random variables themselves. This leads to Bayes' theorem:

Bayes' Theorem for Parameter Estimation

$$p(\mathbf{w} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{w})\, p(\mathbf{w})}{p(\mathbf{x})}$$

- $p(\mathbf{w})$: **Prior** probability — our belief about $\mathbf{w}$ *before* seeing data.
- $p(\mathbf{x} \mid \mathbf{w})$: **Likelihood** — probability of the data given the parameters.

The Bayesian Perspective II

- $p(\mathbf{w} \mid \mathbf{x})$: **Posterior** probability — our updated belief about $\mathbf{w}$ *after* seeing data.
- $p(\mathbf{x})$: **Evidence** or marginal likelihood.

The Intractable Posterior Problem

In Bayesian estimation, we want the full posterior distribution for parameters $\mathbf{w}$ given data $\mathcal{D}$:

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{p(\mathcal{D})}$$

The Challenge

The marginal likelihood $p(\mathcal{D}) = \int p(\mathcal{D} \mid \mathbf{w}) p(\mathbf{w})\, d\mathbf{w}$ is often intractable for complex models. We cannot compute the posterior in closed form.

Common Solutions

- MCMC sampling (accurate, but slow), will discuss it later
- Variational Inference (fast, but biased), will discuss it later
- MAP and Laplace Approximation (fast, based on optimization)

Maximum A Posteriori (MAP) Estimation:

$$\mathbf{w}^*_{\text{MAP}} = \arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathbf{x}) = \arg\max_{\mathbf{w}} p(\mathbf{x} \mid \mathbf{w})\, p(\mathbf{w})$$

Maximum Likelihood Estimation (MLE) is a special case with a uniform prior:

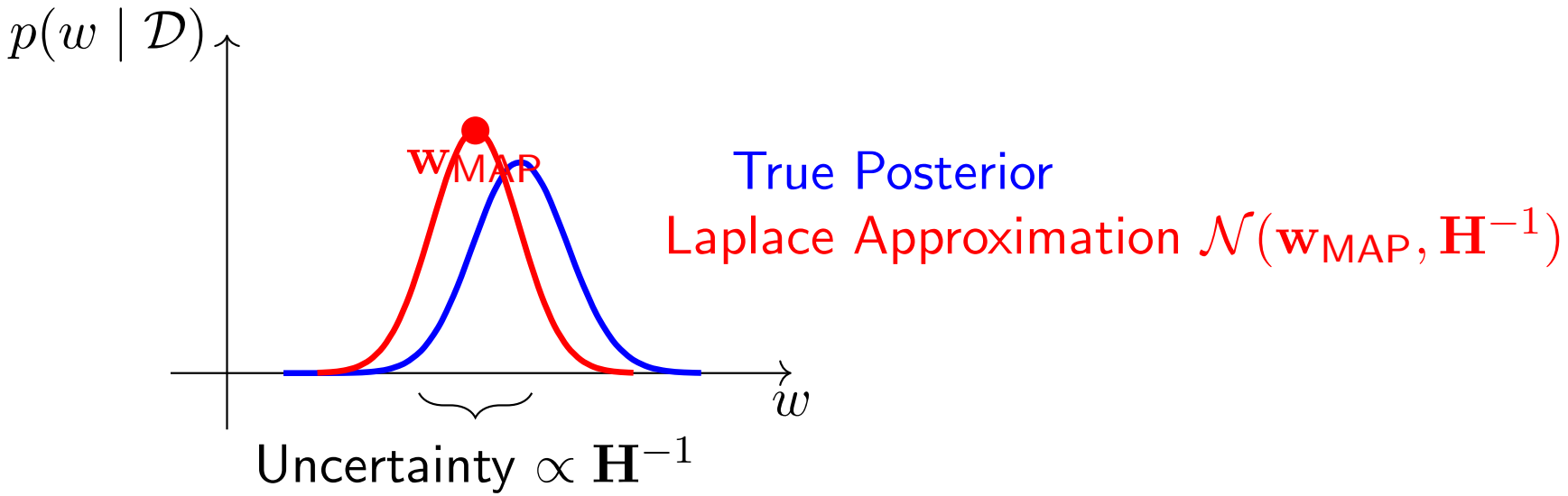
$$\mathbf{w}^*_{\text{MLE}} = \arg\max_{\mathbf{w}} p(\mathbf{x} \mid \mathbf{w}) = \arg\max_{\mathbf{w}} \prod_{i=1}^{N} p(x_i \mid \mathbf{w})$$

Laplace approximation

Laplace Approximation: Core Idea

Approximate the true posterior $p(\mathbf{w} \mid \mathcal{D})$ with a Gaussian distribution $q(\mathbf{w})$ by:

1. Finding its mode (**MAP estimate**).
2. Matching its curvature at that mode.



Mathematical Derivation

Step 1: Find the Mode

Find the Maximum A Posteriori (MAP) estimate:

$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathcal{D}) = \arg\min_{\mathbf{w}} \underbrace{[-\log p(\mathcal{D} \mid \mathbf{w}) - \log p(\mathbf{w})]}_{E(\mathbf{w})}$$

Step 2: Taylor Expand around the Mode

Expand the negative log-posterior $E(\mathbf{w}) = -\log p(\mathbf{w} \mid \mathcal{D})$:

$$E(\mathbf{w}) \approx E(\mathbf{w}_{\text{MAP}}) + \underbrace{(\mathbf{w} - \mathbf{w}_{\text{MAP}})^T \nabla E(\mathbf{w}_{\text{MAP}})}_{=0} + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{\text{MAP}})^T \underbrace{\nabla\nabla E(\mathbf{w}_{\text{MAP}})}_{\mathbf{H}} (\mathbf{w} - \mathbf{w}$$

$$= \text{const} + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{\text{MAP}})^T \mathbf{H} (\mathbf{w} - \mathbf{w}_{\text{MAP}})$$

Step 3: Exponentiate to get the approximate Gaussian Posterior

$$p(\mathbf{w} \mid \mathcal{D}) \propto \exp(-E(\mathbf{w})) \approx \exp\left(-\frac{1}{2}(\mathbf{w} - \mathbf{w}_{\text{MAP}})^T \mathbf{H} (\mathbf{w} - \mathbf{w}_{\text{MAP}})\right)$$

$$\Rightarrow q(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{w}_{\text{MAP}}, \mathbf{H}^{-1})$$

Summary, Advantages, and Limitations

The Laplace Approximation

$$p(\mathbf{w} \mid \mathcal{D}) \approx \mathcal{N}(\mathbf{w} \mid \mathbf{w}_{\text{MAP}}, \mathbf{H}^{-1})$$

where $\mathbf{H} = \nabla\nabla \left[-\log p(\mathcal{D} \mid \mathbf{w}) - \log p(\mathbf{w}) \right] \Big|_{\mathbf{w}=\mathbf{w}_{\text{MAP}}}$ is the Hessian matrix.

Advantages

- Simple and intuitive.
- Turns integration into optimization.
- Provides a full distribution estimate, not just a mode.

Limitations

- Local approximation. Poor for multi-modal, skewed, or heavy-tailed posteriors.
- Requires computing and inverting the Hessian, which is $O(PD^3)$ for D parameters.
- Is a Gaussian approximation, which might be unsuitable.

Numerical Considerations in Maximum Likelihood I

In practice, working directly with the product of probabilities presents numerical challenges:

- Multiplying many probabilities (all < 1) results in extremely small numbers.
- This can lead to arithmetic underflow (numbers too small for finite precision).

A standard solution is to use the negative log-likelihood:

$$\begin{aligned} \mathbf{w}^*_{\text{MLE}} &= \arg\max_{\mathbf{w}} \prod_{i=1}^{N} p(x_i \mid \mathbf{w}) \\ &= \arg\min_{\mathbf{w}} \left(-\sum_{i=1}^{N} \log p(x_i \mid \mathbf{w}) \right) \end{aligned}$$

Why is this better?

Numerical Considerations in Maximum Likelihood II

- Products become sums, which are numerically stable.
- The log function compresses the dynamic range of values.
- Minimization is often more standard in optimization frameworks.

Thus, maximum likelihood estimation becomes finding the minimum of the negative log-likelihood function.

CHAPTER 2: Linear model

Linear Regression: A Supervised Learning Approach I

A fundamental supervised learning task: given a dataset of N input-output pairs.

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$$

- $\mathbf{x}_n \in \mathbb{R}^D$: **Independent variable** (D-dimensional feature vector)
- $y_n \in \mathbb{R}$: **Dependent variable** (one-dimensional real-valued target)

Linear Regression: A Supervised Learning Approach II

Probabilistic Model

We assume the target y is a $\boxed{linear function}$ of the inputs $\mathbf{x}$, corrupted by Gaussian noise:

$$P(y \mid \mathbf{x}, \mathbf{w}, \sigma^2) = \mathcal{N}(y \mid \mathbf{w}^T \mathbf{x}, \sigma^2)$$

The mean is a linear combination of the features:

$$\mu = \mathbf{w}^T \mathbf{x} = w_0 + w_1 x_1 + \ldots + w_D x_D$$

w_0 is called bias. $\mathbf{w}$ is a $D+1$ dimensional row vector, and $\mathbf{x} = [1, x_1, \ldots, x_D]^T$ is $D+1$ dimensional Column vector.

From Likelihood to Loss Function

Linear Regression: A Supervised Learning Approach III

Maximizing the likelihood (uniform priori for w) is equivalent to minimizing the negative log-likelihood:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \log \mathcal{N}(y_n \mid \mathbf{w}^T \mathbf{x}_n, \sigma^2) \propto \sum_{n=1}^{N} (y_n - \mathbf{w}^T \mathbf{x}_n)^2$$

$$\boxed{E(\mathbf{w}; \mathcal{D}) = \frac{1}{2} \sum_{n=1}^{N} (y_n - \mathbf{w}^T \mathbf{x}_n)^2}$$

The maximum likelihood solution is given by the normal equations:

$$\mathbf{w}_{\mathsf{ML}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Preventing Overfitting: Regularization I

Regularized Loss Function

To prevent overfitting and improve generalization (The model fits data too well and has bad generalization), we introduce a penalty term to the loss function:

$$E(\mathbf{w}; \mathcal{D}, \lambda) = \sum_{n=1}^{N} (y_n - \mathbf{w}^T \mathbf{x}_n)^2 + \lambda \|\mathbf{w}\|_2^2$$

This specific form is known as L_2 regularization or ridge regression.

- $\lambda \geq 0$: **Hyperparameter** controlling the regularization strength.
- $\|\mathbf{w}\|_2^2 = \sum_j w_j^2$: The squared L_2 norm of the weight vector.

Bayesian Interpretation

Preventing Overfitting: Regularization II

This formulation has a natural interpretation in the Bayesian framework. The regularization term is equivalent to placing a Gaussian (normal) prior on the parameters:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \lambda^{-1}\mathbf{I})$$

Maximizing the posterior distribution $p(\mathbf{w} \mid \mathbf{X}, \mathbf{y})$ (MAP estimation) leads directly to the minimization of $E(\mathbf{w})$.

Solution

Unlike the comment in the original text, this regularized problem **does have a closed-form solution**:

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

The matrix $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})$ is always invertible, which is a key advantage of L_2 regularization.

Logistic Regression (Classification)

Now the data points are $(t_1, x_1), \ldots, (t_N, x_N)$, with t_i dependent variables taking two values $(0, 1)$. x_i is also a D dimensional vector.

Binary Classification Model

$$\boxed{P(t|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x})^t(1 - \sigma(\mathbf{w}^T\mathbf{x}))^{1-t}}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function

The joint probability is then

$$p(D|x) = \prod_{n=1}^{N} \sigma(\mathbf{w}^T \mathbf{x}_n)^{t_n} (1 - \sigma(\mathbf{w}^T \mathbf{x}_n))^{1-t_n}$$

Loss Function

Negative log-likelihood:

$$E(\mathbf{w}; \mathcal{D}) = -\sum_{n=1}^{N} \left[t_n \log \sigma(\mathbf{w}^T \mathbf{x}_n) + (1 - t_n) \log(1 - \sigma(\mathbf{w}^T \mathbf{x}_n)) \right]$$

Multiclass Classification

The data is $(t_1, x_1), \ldots, (t_N, x_N)$, and t_i takes K discrete values.

Softmax Regression

Probability for class i:

$$p_i(w; x) = \frac{\exp(\mathbf{w}_i^T \mathbf{x})}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x})}$$

The probability distribution is then

$$p(t|w) = \prod_{i=1}^{K} p_i(w; x)^{t_i}, \quad p(\mathcal{D}|w) = \prod_{n=1}^{N} \prod_{i=1}^{K} p_i(w; x_n)^{t_{ni}}$$

The loss function from negative log-likelihood is

$$\boxed{E(\mathcal{D}; w) = -\log p(\mathcal{D}|w) = -\prod_{n=1}^{N} \prod_{i=1}^{K} t_{ni} \log p_i(w; x_n)}$$

This loss-function is called cross entropy.

Summary

Given the probability assumptions and using the MAP estimation of Bayes method, one get a loss function for the parameters

$$E(\mathcal{D}; w)$$

where $\mathcal{D}$ consists of known data, and the next goal is to find the minimal value w^* of the function $E(\mathcal{D}; w)$. Usually one can not find the exact solution.

The Optimization Problem

Most machine learning involves minimizing a loss function $J(\mathbf{w})$ with respect to model parameters $\mathbf{w}$.

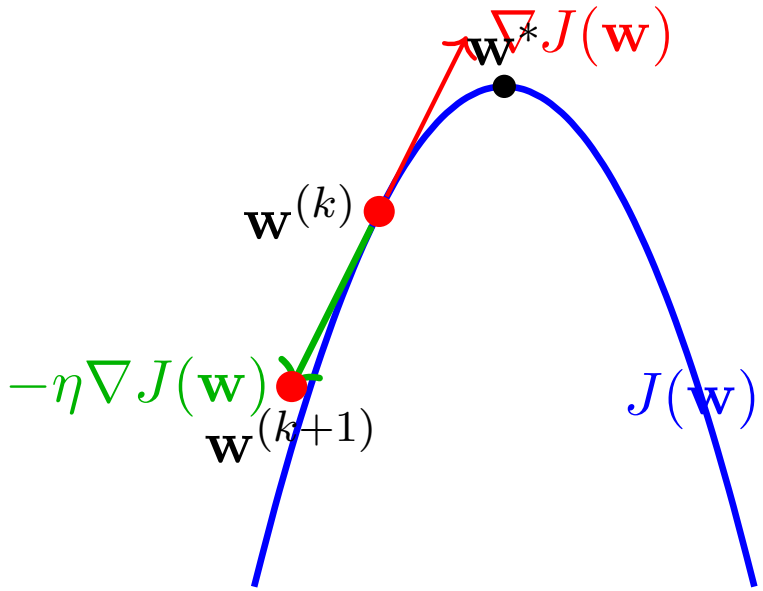
$$\mathbf{w}^* = \arg\min_{\mathbf{w}} J(\mathbf{w})$$

The Challenge

For complex models (e.g., neural networks), finding an analytical solution is impossible. We need an iterative algorithm.

Gradient Descent is the fundamental algorithm for this task.

Intuition: Walking Down a Hill



- The gradient $\nabla J(\mathbf{w})$ points uphill (direction of steepest ascent).
- To minimize, we move in the opposite direction: $-\nabla J(\mathbf{w})$.
- The step size is scaled by the learning rate η.

The Algorithm: Core Update Rule

Gradient Descent Update Step

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \nabla_{\mathbf{w}} J(\mathbf{w}^{(k)})$$

Parameters:

- $\mathbf{w}^{(k)}$: Parameters at iteration k
- η: Learning rate ($\eta > 0$)
- $\nabla_{\mathbf{w}} J$: Gradient of J w.r.t. $\mathbf{w}$

Stopping Criteria:

- Max iterations reached
- $\|\nabla J(\mathbf{w})\| < \epsilon$
- Change in loss $|J^{(k+1)} - J^{(k)}| < \epsilon$

Variants of Gradient Descent I

Batch Gradient Descent

Uses the entire training set to compute the gradient.

Pro: True gradient direction. **Con:** Slow for large datasets.

$$\nabla J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \nabla J_i(\mathbf{w})$$

Stochastic Gradient Descent (SGD)

Uses a single, random training example (x_i, y_i) to compute the gradient.

Pro: Very fast per step. **Con:** Noisy updates.

$$\nabla J(\mathbf{w}) \approx \nabla J_i(\mathbf{w})$$

Mini-batch Gradient Descent (Most Common)

Variants of Gradient Descent II

A compromise: uses a small random subset (mini-batch) of size B.
Pro: Smoother and more efficient than SGD.

$$\nabla J(\mathbf{w}) \approx \frac{1}{B} \sum_{i=1}^{B} \nabla J_i(\mathbf{w})$$

The Critical Role of the Learning Rate (η)

- Too Small: Slow convergence, can get stuck in local minima.
- Too Large: Overshoots, oscillates, or even diverges.
- **Just Right**: Efficient and stable convergence to a (local) minimum.

Advanced optimizers (Adam, RMSProp) adapt η during training (will be discussed later).

Probabilistic Inference for Regression I

After finding optimal parameters $\mathbf{w}^*$ for our linear regression model, we can perform predictive inference for a new target y_* given new input $\mathbf{x}_*$.

Point Prediction

The simplest prediction is the mean of the distribution:

$$\mathbb{E}[y_* \mid \mathbf{x}_*, \mathcal{D}] = \mathbf{w}^{*T}\mathbf{x}_*$$

Full Predictive Distribution (Bayesian Approach)

Probabilistic Inference for Regression II

To capture uncertainty, we compute the full predictive distribution by integrating over all possible parameters, weighted by their posterior probability:

$$p(y_* \mid \mathbf{x}_*, \mathcal{D}) = \int p(y_*, \mathbf{w} \mid \mathbf{x}_*, \mathcal{D})\, d\mathbf{w}$$
$$= \int p(y_* \mid \mathbf{x}_*, \mathbf{w})\, p(\mathbf{w} \mid \mathcal{D})\, d\mathbf{w}$$

- **Likelihood:** $p(y_* \mid \mathbf{x}_*, \mathbf{w}) = \mathcal{N}(y_* \mid \mathbf{w}^T \mathbf{x}_*, \sigma^2)$
- **Posterior:** $p(\mathbf{w} \mid \mathcal{D})$ represents our updated belief about the parameters after seeing the data.

Probabilistic Inference for Classification I

For a classification model with a new input $\mathbf{x}_*$, we predict the target y_*.

Point Prediction (Maximum a Posteriori - MAP)

We can use the optimized parameters $\mathbf{w}^*$ to get class probabilities and choose the most likely class:

$$\hat{\mathbf{p}} = \text{softmax}(\mathbf{w}^{*T}\mathbf{x}_*), \quad \hat{y}_* = \arg\max_i \hat{p}_i$$

More generally, we can sample from the top-K classes based on these probabilities.

Full Predictive Distribution (Bayesian Approach)

Probabilistic Inference for Classification II

To properly account for model uncertainty, we again integrate over the posterior distribution of the parameters:

$$p(y_* = c \mid \mathbf{x}_*, \mathcal{D}) = \int p(y_* = c \mid \mathbf{x}_*, \mathbf{w})\, p(\mathbf{w} \mid \mathcal{D})\, d\mathbf{w}$$

- **Likelihood:** $p(y_* = c \mid \mathbf{x}_*, \mathbf{w}) = [\mathrm{softmax}(\mathbf{w}^T \mathbf{x}_*)]_c$

The Three Pillars of a Machine Learning Model

1. **Model Formulation** $\rightarrow$ loss function $E(w; \mathcal{D})$
2. **Parameter Estimation (Training)**
3. **Prediction (Inference)**

1. Model Formulation

Define a probabilistic model that describes how the data is generated.

- Core component: A family of distributions $p(\text{data} \mid \mathbf{w})$.
- **Goal:** Find parameters $\mathbf{w}$ that make the observed data $\mathcal{D}$ most probable.
- This leads to an objective function (loss) to optimize:

$$E(\mathcal{D}; \mathbf{w}; \boldsymbol{\lambda}) = -\log p(\mathcal{D} \mid \mathbf{w}) + \text{Penalty}(\mathbf{w}, \boldsymbol{\lambda})$$

- $\boldsymbol{\lambda}$ represents hyperparameters (e.g., regularization strength) which are set manually, not optimized.

2. Parameter Estimation (Training)

The process of finding the optimal parameters $\mathbf{w}^*$.

- **Algorithm:** Typically a variant of Stochastic Gradient Descent (SGD).
- **Hyperparameters:** Learning rate η, batch size B, number of epochs.
- **Output:** A trained model with fixed parameters $\mathbf{w}^*$.

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} E(\mathcal{D}; \mathbf{w}; \boldsymbol{\lambda})$$

3. Prediction (Inference)

Using the trained model to make predictions on new, unseen data.

- **Input:** New independent variable $\mathbf{x}_{\text{new}}$.
- **Output:** Prediction for dependent variable y_{new}.
- For a probabilistic model: Can output a full predictive distribution $p(y_{\text{new}} \mid \mathbf{x}_{\text{new}}, D)$.