Comprehensive Notes on Probabilistic Frameworks

Joseph Margaryan

December 27, 2024

Contents

1 Foundations of Probabilistic Frameworks

1.1 Probability Basics

1.1.1 Definition of probability

Probability measures the likelihood of an event occurring, ranging from 0 (impossible event) to 1 (certain event). It quantifies the uncertainty and is defined as:

$$P(A) = \frac{\textbf{Number of favorable outcomes}}{\textbf{Total number of outcomes}}$$

1.1.2 Conditional Probability

Conditional probability is the probability of an event A occurring, given event B has already occurred:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Where:

- Where $P(A \cap B)$ is the probability of both A and B occur
- $P(B) \neq 0$

1.1.3 Joint Probabilities

The joint probability is the probability of two events A and B occurring simultaneously:

$$P(A \cap B) = P(A|B)P(B)$$

or

$$P(A \cap B) = P(B|A)P(A)$$

1.1.4 Expectation, Variance and Covariance

The **Expected value** for a discrete random variable is given by:

$$\mathbb{E}[X] = \sum_{x} x \cdot P(X = x)$$

For a continuous random variable:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \cdot f(x)dx$$

The **Variance** is given by the expected squared deviation from the mean:

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$$

The **Covariance** is to measure the linear relationship between two random variables X and Y

$$Cov(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

The **Correlation coefficient** (p) is a normalized measure of linear relationship:

$$p(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}}$$

1.1.5 Covariance Matrix

For a random vector $\mathbf{X} = [X_1, X_2, \cdots, X_N]^T$, the covariance matrix is defined as:

$$\Sigma = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^T]$$

Suppose we have two features X_1 and X_2 The covariance matrix will be:

$$\Sigma = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_1, X_2) & Var(X_2) \end{bmatrix}$$

1.1.6 Probability Mass Function (PMF)

1.1.7 Probability Density Function (PDF)

1.1.8 Cumulative Distribution Function (CDF)

1.1.9 Bayes rule

When classifying targets based on conditional probabilities, we can set decision boundaries based on rules:
Bayes optimal classification rule uses posterior probabilities:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Bayes classification rule always chooses the class that yields the highest probability

$$\hat{y} = \underset{y}{argmax} P(y|X)$$

Under the $[0 - 1]$ loss, this will always yield the lowest risk compared to a probabilistic classifier. These are called Maxmimum a Posterior (MAP) predictions
When dealing with continuous targets, we predict the μ and σ of a given probability distribution, often the gaussian distribution $\mathcal{N}(\mu, \sigma^2)$

1.2 Entropy

Entropy quantifies the uncertainty of a probability distribution. It measures how unpredictable the outcome of a random variable is;

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x)$$

where:

- X is a random variable
- $\mathcal{X}$ is the set of all possible outcomes
- $P(X)$ is the probability of x

- High entropy means high uncertainty in a system
- Low entropy means low uncertainty in a system

For a two random variables X and Y, joint entropy measures the uncertainty in their joint distribution:

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) \log P(x, y)$$

Conditional entropy measure the entropy of Y given X

$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) \log P(y|x)$$

1.3 Maximum Entropy

The Maximum Entropy Principle states that among all probability distributions consistent with the known constraints, the one with the highest entropy should be chosen. This is because it is the least biased and makes the fewest assumptions about unknown information.
Find a probability distribution $P(X)$ that maximizes entropy:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

subject to known constraints, such as:

- $\Sigma_{x \in \mathcal{X}} P(x) = 1$: The distribution must sum to one
- Example comstraint could be that we the expected value: $\mathbb{E}[f(x)] = \sum_{x \in \mathcal{X}} P(x) f(x)$

Solution via Langrage multipliers:
The probability distribution satisfying the constraints is often exponential in form:

$$P(x) = \frac{1}{Z} exp\left(-\lambda f(x)\right)$$

Where $Z = \sum_{x \in \mathcal{X}} exp\left(-\lambda f(x)\right)$ is the partioning function

1.4 Mutual Information

Mutual information quantifies how much knowing one random variable X is gained by knowing another random variable Y. It measures the dependency between two variables:

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

Alternative representation using entropy H:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

In terms of conditional entropy:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Interpretation: How much knowing Y reduces the uncertainty of X, and vice versa.

1.5 Kullback-Leibler Divergence

K-L Divergence measures the distance between two probability distributions P and Q:

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

1.6 Cross-Entropy

Measures the expected number of bits needed to encode samples from P using model Q:

$$H(P,Q) = -\sum_{x \in \mathcal{X}} P(x) \log Q(x)$$

2 Probabilistic Distributions

2.1 Boltzmann Distribution

The Boltzmann Distribution describes the probability of a system being in a state i with energy E_i at a given temperature T:

$$P(E_i) = \frac{1}{Z} e^{\frac{-E_i}{k_B T}}$$

Where

- E_i Energy of state i

- T Temperature (in Kelvin)
- k_B Boltzmann constant

with Partioning function:

$$Z = \sum_i e^{\frac{-E_i}{k_B T}}$$

Properties:

- Assigns higher probabilities to states with lower energy.
- Probability decreases exponentially with increasing energy.

2.2 Gaussian Distribution

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

2.3 Bernoulli Distribution

2.4 Binomial Distribution

2.5 Poisson Distribution

2.6 Poisson Distribution

2.7 Drischlet Distribution

2.8 Multivariate Gaussian Distribution

3 Constrained Lagrangian Multipliers

4 Bayesian Inference

Bayesian inference is a framework for reasoning under uncertainty. It updates our beliefs about unknown parameters or hypotheses in light of new evidence. In the case of BNNs, we model each weight as a probability distribution, often a gaussian $\mathcal{N}(\mu, \sigma^2)$ with μ and σ^2

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

Where:

- $P(\mathbf{w}|\mathcal{D})$: The **posterior distribution** over the weights
- $P(\mathcal{D}|\mathbf{w})$: **Likelihood** of the data given the weights
- $P(\mathbf{w})$: The **prior** distribution over the weights
- $P(\mathcal{D})$ The **Evidence**, ensuring the posterior inetgrates to one

To make predictions for new inputs x^*, BNNs marginalize over the posterior distribution of weights

$$P(y^*|x^*, \mathcal{D}) \int P(y^*|x^*, \mathbf{w}) P(\mathbf{w}|\mathcal{D}) d\mathbf{w}$$

This accounts for all plausible weight configurations, resulting in predictions that inherently include uncertainty. This way, we get the uncertainty associated with each prediction rather than just point estimates.

5 Sampling Techniques

5.1 Importance Sampling

Importance Sampling is a statistical technique used to estimate properties of a distribution while using samples from a different distribution.
The main idea of importance sampling is to estimate the expectation of a function $f(x)$ under a target distribution $p(x)$

$$\mathbb{E}_p[f(x)] = \int f(x)p(x)dx$$

When direct sampling from $p(x)$ is difficult, we use a proposal distribution $q(x)$ that is easier to sample from. The target distribution $p(x)$ is "reweighted" using $q(x)$, leading to:

$$\mathbb{E}_p[f(x)] = \int f(x)\frac{p(x)}{q(x)}q(x)dx$$

The $w(x) = \frac{p(x)}{q(x)}$ is called **the importance weight**

5.1.1 Monte Carlo Approximation

With samples $x_1, x_2, \cdots, x_n \sim q(x)$, the expectation can be approximated as:

$$\mathbb{E}_p[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)w(x_i)$$

Where:

$$w(x_i) = \frac{p(x_i)}{q(x_i)}$$

Here, $w(x_i)$ adjusts for the fact that the samples are not drawn from $p(x_i)$

5.2 Rejection Sampling

- Accept/reject based on a scaled proposal distribution.

5.3 Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms used to generate samples from a probability distribution when direct sampling is infeasible. These methods are widely used in Bayesian statistics, statistical physics, and machine learning for approximating complex integrals, simulating distributions, and performing inference.
MCMC combines two fundamental concepts: **Markov chains** and **Monte Carlo sampling**. The essence of MCMC lies in constructing a Markov chain whose stationary distribution matches the target distribution, and then using samples from this chain to approximate expectations or probabilities.

5.3.1 Markov Chain

A **Markov Chain** is a stochastic process defined on a state space $\mathcal{X}$ where the probability of transitioning to a new state depends only on the previous state. This property is mathematically expressed as:

$$P(X_{t+1} = x_{t+1}|X_t = x_t, X_{t-1}, \cdots X_0) = P(X_{t+1} = x_{t+1}|X_t = x_t)$$

Stationary Distribution $\pi(x)$ is when the chain has reached a point where it remains in $\pi(x)$ after any number of steps:

$$\pi(x') = \sum_{x \in \mathcal{X}} \pi(x) P(x \to x'),\ \forall x' \in \mathcal{X}$$

The idea is that we want our markov chain to converge to a stationary distribution which is the target distribution. This process allows us to approximate high-dimensional integrals by transitioning between points in the state space, where the probability of visiting each point is proportional to the target distribution's density

5.3.2 Monte Carlo Sampling

Monte Carlo sampling approximate expectations by random sampling. If we want to compute the expectation of a function $f(x)$ under a target distribution $\pi(x)$, the monte carlo estimate is:

$$\mathbb{E}_\pi[X] = \int_{\mathcal{X}} f(x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$$

For complex distributions, it is infeasable to directly sample from $\pi(x)$. MCMC addresses this by using a markov chain to generate dependant samples that converge to $\pi(x)$

5.4 Metropolis-Hasting

5.5 Gibbs Sampling

Given a joint probability distribution $P(x_1, x_2, \cdots x_n)$ over d random variable, our goal is to infer properties of the distribution, such as expectations or probabilities, by sampling from it.
The core idea is that we cant sample directly from the joint probability distribution $P(x_1, x_2, \cdots x_n)$, we iteratively sample from each condition probability distribution, we construct a Markov chain whose stationary distribution is the target joint distribution.

Algorithm 1 Gibbs Sampling

Require: Target joint distribution $P(x_1, x_2, \ldots, x_d)$ with $x = (x_1, x_2, \ldots, x_d)$
Ensure: Samples from $P(x_1, x_2, \ldots, x_d)$
1: **Initialization**: Start with an initial guess $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_d^{(0)})$
2: **for** $t = 1, 2, \ldots, T$ **do** ▷ Iterate for T iterations
3: **for** $i = 1, 2, \ldots, d$ **do** ▷ Sequentially update each variable
4: Sample $x_i^{(t+1)} \sim P(x_i | x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_d^{(t)})$
5: **end for**
6: **end for**
7: **Return** $(x_1^{(t)}, x_2^{(t)}, \ldots, x_d^{(t)})$ for $t >$ burn-in period

5.5.1 Example Motif Discovery

Problem setup
We have a multiple sequence alignment and we would like to estimate the joint probability of motif start positions $z = (z_1, z_2, \cdots, z_N)$ for each sequence $S_1, S_2, \cdots, S_N$ of length L using a position weight matrix (PWM) $\mathbf{W}$.
The goal is to infer these positions probabilistically by maximizing the joining probability distribution of the motif positions and the PWM $\mathbf{W}$ The joint probability distribution can be expressed:

$$P(S, z|\mathbf{W}) = \prod_{i=1}^{N} P(S_i|z_i, \mathbf{W}) P(z_i)$$

where:

- $P(S, z|\mathbf{W})$ is the likelihood of the motif at position z_i in sequence S_i computed using PWM

The conditional distribution for the start position z of the motif in a sequence, conditioned on the motif positions in all other sequences, can be expressed as a softmax function:

$$P(z|\mathbf{W}, S) = \frac{exp(S_w(z, S))}{\sum_i exp(S_w(i, S))}$$

Algorithm 2 Motif Discovery Using Gibbs Sampling

Require: Multiple sequence alignment $S = \{S_1, S_2, \ldots, S_N\}$ of N sequences, each of length L; motif length w.

Initialize: Randomly assign start positions $z = \{z_1^0, z_2^0, \ldots, z_N^0\}$ for the motif in each sequence.

while not converged **do**

  for $i = 1, 2, \ldots, N$ **do** ▷ Iterate through all sequences

    Hold-Out Step: Temporarily remove S_i from the model.

    Recompute PWM: Using the current motif positions z_{-i} (excluding z_i):

$$W[j, a] = \frac{\#\text{ occurrences of } a \text{ at position } j + \alpha}{\text{Total motifs at position } j + 4\alpha}, \quad a \in \{A, C, G, T\}.$$

    Compute Likelihood: For each possible start position $z_i' \in \{1, 2, \ldots, L - w + 1\}$:

$$P(S_i | z_i', \mathbf{W}) = \prod_{j=1}^{w} W[j, S_i[z_i' + j]].$$

    Normalize: Compute probabilities for all z_i':

$$P(z_i = z_i' | z_{-i}, S, \mathbf{W}) = \frac{P(S_i | z_i', \mathbf{W})}{\sum_{z_i'=1}^{L-w+1} P(S_i | z_i', \mathbf{W})}.$$

    Sample: Sample a new start position z_i for S_i using the normalized probabilities.

  end for

end while

Output: Final motif positions $z = \{z_1, z_2, \ldots, z_N\}$ and PWM $\mathbf{W}$.

5.6 Hamiltonian Monte Carlo (HMC)

- Simulating trajectories in phase space.
- Leapfrog integration and energy conservation.

5.7 No U-Turn Sampler (NUTS)

6 Variational Inference

6.1 Goal

- Approximate the posterior $P(\theta|\text{Data})$ with a simpler distribution $q(\theta)$.

6.2 Evidence Lower Bound (ELBO)

- Maximizing:

$$\text{ELBO} = \mathbb{E}_{q(\theta)}[\log P(\text{Data}|\theta)] - \text{KL}(q(\theta)||\pi(\theta)). \tag{1}$$

6.3 Applications

- Variational autoencoders.
- Scalable Bayesian models.

7 Stochastic Processes

7.1 Brownian Motion

Arima ARCH reinforcement learning

8 Applications

8.1 Hidden Markov Model

8.2 Deep Markov Model

8.3 Gaussian Processes

8.4 Gaussian Mixture Models

8.5 Bayesian Neural Network

8.6 Solving ordinary differential equations

9 Interpretability and Theoretical Bounds

9.1 Probabilistic Model Interpretability

- Posterior Predictive Checks: Diagnostics for model evaluation.
- Calibration: Evaluating the reliability of probabilistic predictions.

9.2 Theoretical Foundations

- Markovs inequality
- Chebyshev inequality
- Hoeffdings inequality
- PAC-Bayesian Framework: Combining probabilistic reasoning with generalization bounds.
- VC Dimension: Understanding capacity and generalization in probabilistic models.
- Information-Theoretic Generalization Bounds: Connecting mutual information to learning efficiency.

10 Neural Network

10.1 Gradient Descent

10.2 Linear Layer

10.3 Activation functions

10.4 Dropout

10.5 Convolutional layer

10.6 Max pooling layer

10.7 Batch Normalization

10.8 Embeddings

10.9 Positional Encodings

10.10 Self Attention Mechanism

10.11 Transformer

10.12 Variational Auotencoders

10.13 Diffusion Models

10.14 Generative Adversarial Networks

11 Bioinformatics

11.1 Gibbs Free Energy

11.2 Potentials of Mean Force

11.3 RMSD

11.4 De Novo Prediction

11.5 Rosseta

11.9 Free Energy Minimization

Zuker and Stiegler's model. Partition function for ensembles of structures.