# Sequential Search

1. Hyperparam response surface

* Find the hyperparams that minimize/maximize a performance metric
* A: Algorithm
  + Aλ: Hyperparameters
* X: dataset
* L: Function to optimize
  + Performance metric

1. Hyperparam tuning – Challenges

* We can’t define a formula to find the hyperparam -> black box function
* The response surface is not differentiable -> it does not have a gradient
* Try different combinations of hyperparam and evaluate model performance

1. Hyperparameter search

* For every hyperparameter combination, use cross-validation – train K models & estimate K metrics
* If models are simple (i.e., linear models, tree based algorithms):
* Grid search
* Random search
* If models are complex (i.e., NN)
* Training the model is very costly (time and money)
* Trying all possible combinations is not an option
* Select smartly which hyperparams (combinations) to evaluate

1. Sequential search

* **Grid search and random search** generate all the candidate points up front and evaluate them in parallel
* **Sequential search** techniques pick a few hyperparam settings, evaluate their quality, then decide where to sample next
* Iterative and sequential process
* Not parallelizable
* Goal: make fewer evaluations, only of those most promising candidate hyperparams

1. Sequential search trade-off

* Trade off: less ML model training time >< time to estimate where to sample next
* Sequential search makes sense when the **evaluation procedure** (training the model - performance) **takes much longer than** the process of **evaluating where to sample** next

1. Bayesian optimization

* Bayesian optimization is a sequential strategy for **global optimization** **of black-box functions**, which does not assume any functional forms
* Bayesian optimization is usually employed to optimize **expensive-to-evaluate** functions

# Bayesian optimization

1. Overview

* A sequential approach for global optimization of black-box objective functions, that are costly to evaluate
* Mathematically, we want to find the global maximizer (or minimizer) of an unknown (black-box) object function f:

Where x are the hyperparams

1. The objective function f

* F is continuous
* F is difficult to evaluate -> too much time or money
* F lacks known structure, like concavity or linearity -> f is black-box
* F has no derivative -> we can’t evaluate a gradient
* F can be evaluated at arbitrary points of x (the hyperparams)
* We can make point-wise observations of f

1. Bayesian optimization

* F is unknown
* In Bayesian optimization, we treat f as a random function and place a **prior** over it (prior: a function that captures the belief-distribution, behavior of f)
* Then, we evaluate f at certain points
* With new data, the prior (f original belief) is updated to a new **posterior** distribution
* The posterior distribution is used to construct an **acquisition function to determine the next query point**

1. Estimating the prior

* Gaussian processes
* Tree-parzen estimator
* Random Forests

1. Acquisition function

* Expected improvement (EI)
* Gaussian process upper confidence bound (UCB)

# Bayes Inference – Introduction

1. Bayesian Inference – Foundations

* Bayesian inference is the reallocation of probability (credibility) across possibilities
* The possibilities across which the probability (credibility) is reallocated are usually the parameters (or hyperparams) of a mathematical model
* In Bayesian statistics, the **probability (credibility) expresses the degree of belief** in an event
* The degree of belief can be based on prior knowledge about the event, i.e., data from an experiment, or on personal beliefs about the event
* These beliefs (probabilities) can be **updated** when we gather **new information** about the event
* We use **Bayes’ rule** to update the belief (probability)

1. Intuition

* See slides for reference
* Reassignment of probabilities whenever new inf about the event is gathered (posterior becomes the new prior when new info is available)

1. Probability reallocation

* After gathering new info, we reallocate our beliefs
* We hypothesize a range of possible distributions, models or generators (priors) and from data we determine their credibility (posterior)

1. Prior and posterior probability

* **Prior prob**. is the **unconditional probability** assigned to an event **before** any relevant info is taken into account
* **Posterior prob.** is the **conditional probability** of an event that is assigned **after** taking into account the new evidence
* **Prior and posterior** are mathematically related by Bayes’ rule

# Joint and conditional probability

1. Data – counts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **German Sheppard** | **Dobermann** | **Labrador** | **Marginal (Dysplasia)** |
| **None** | 0 | 30 | 60 | 100 |
| **Mild** | 30 | 40 | 210 | 280 |
| **Severe** | 170 | 30 | 30 | 230 |
| **Marginal (Breed)** | 200 | 100 | 300 | 600 |

1. Marginal probability

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **German Sheppard** | **Dobermann** | **Labrador** | **Marginal (Dysplasia)** |
| **None** | 0.017 | 0.050 | 0.100 | 0.167 |
| **Mild** | 0.050 | 0.067 | 0.350 | 0.467 |
| **Severe** | 0.283 | 0.050 | 0.050 | 0.383 |
| **Marginal (Breed)** | 0.333 | 0.167 | 0.500 | 1.000 |

* Probability that we select a dog of our population and it is a German Sheppard: 0.333
* P(breed) -> marginal probability
* Probability that we select a dog of our population and it is a Doberman with mild dysplasia: 0.067
* P(breed, dysplasia) -> joint probability

1. Joint probability

* The probability of particular combinations of 2 events taking place
* The joint probability is symmetric
* P(A, B) = P(B, A)

1. Marginal probability

* The probability of particular event, collapsed across the values of the other events
* The sum of the joint probabilities
* P(A) = sum(P(A, B))

1. Conditional probability

* A German Sheppard comes to the vet
* What is the probability of dysplasia given that it is a German Sheppard?

|  |  |  |
| --- | --- | --- |
|  | German Sheppard | Divide by 0.333 |
| None | 0.017 | 0.050 |
| Mild | 0.050 | **0.150** |
| Severe | 0.283 | **0.851** |
| Marginal (Breed) | 0.333 | 1.000 |

* Probability of 1 outcome, given that the other is true
* Probability of A given B: P(A|B)
* Conditional probability is NOT symmetric
* P(A|B) ≠ P(B|A)
* In our example
* Prior and Posterior probability
* Prior probability: P(dysplasia)
* Posterior probability: P(dysplasia | breed = German Sheppard)

# Bayes’ rule

1. Bayes’ Rule (Theorem)

* The mathematical relationship between the **prior allocation** of credibility (probability) and the **posterior reallocation**, conditioned on the new evidence

1. Conditional probability

* P(A|B) = **P(A, B)** / P(B)
* P(A|B) \* P(B) = P(A, B)
* P(B|A) = P(A, B) / P(A)
* P(B|A) \* P(A) = **P(A, B)**
* A and B are events, like breed and dysplasia
* P(A|B) is the posterior (conditional) probability of A taking place given the new evidence B
* P(B|A) is also a conditional probability, of B taking place given A
* P(A) and P(B) are the marginal probability of A and B taking place independently
* Bayes’ rule gets us form the prior p(A) to the posterior distribution P(A|B) when focusing on a specific value of B

1. Bayes’ rule in action: Fraud

* A = fraud, B = ML decision
* 1 in 1000 applications are fraudulent -> P(fraud) = 0.001
* ML model correctly identifies 99% of fraudsters -> P(positive | fraud) = 0.99
* ML model incorrectly flags 5% of non-fraudsters -> P(positive | non-fraud) = 0.05
* If application is flagged by ML model, what is the likelihood that it is fraudulent?

1. Bayes’ rule value

* Key application of Bayes’ rule when A is data / generalization error and B is (hyper)parameters
* A model specifies **P(data | parameters)** and the prior **P(parameters)**
* We use Bayes’ rule to convert that into what we are interested in:
* How strongly should we believe in the parameters given the data -> the posterior **P(parameters | data)**?

# Sequential Model-based optimization

1. Bayes’ rule

* With Bayes’ rule we infer a posterior P(A|B) from a prior P(A)
* P(A|B) is proportional with P(B|A) – likelihood and P(A) – prior

1. Hyperparam optimization

* Let w be f(x) and D be the available data
* w is unknown, so we treat it as a random function and place a **prior** over it -> **P(w)**
* P(w) captures our beliefs about the possible values of w
* Given D and the **likelihood model P(D | w)**, we can infer the **posterior P(w | D)** using Bayes’ rule
* The posterior **P(w | D)** represents our updated belief of w after contemplating D
* Given the posterior distribution P(w | D), we construct an **acquisition function** to determine the next query point to sample w
* F(x) is unknown, we do not know its shape
* We need a way to produce a prior, and a posterior -> surrogate model for f(x)
* Gaussian Process Regression
* We also need a way to estimate where to sample next -> acquisition function
* Acquisition function values are high where the mean of f(x) is low (or high) -> exploitation
* Acquisition function values are high when the variance of f(x) is high -> exploration
* The surrogate and the acquisition function need to be cheaper to evaluate than f(x)

# Gaussian distribution

1. Gaussian distribution overview

* Bell shape
* μ = mean value
* center of distribution
* σ = standard deviation
* measure of dispersion
* Symmetric
* Most observations occur around the center
* Probabilities for values further away from the center decrease equally in both directions
* Extreme values in both tails of the distribution are similarly unlikely
* ~ 50% of the observations within xmean ± 0.67 σ
* ~ 99% of the observations within xmean ± 2.7 σ
* A variable is normally distributed
* X ~ N(μ, σ)
* X1 ~ N(μ=0, σ2=1) – standard normal distribution

1. Mean

* Average of the variable values
* n = number of observations

1. Variance and standard deviation

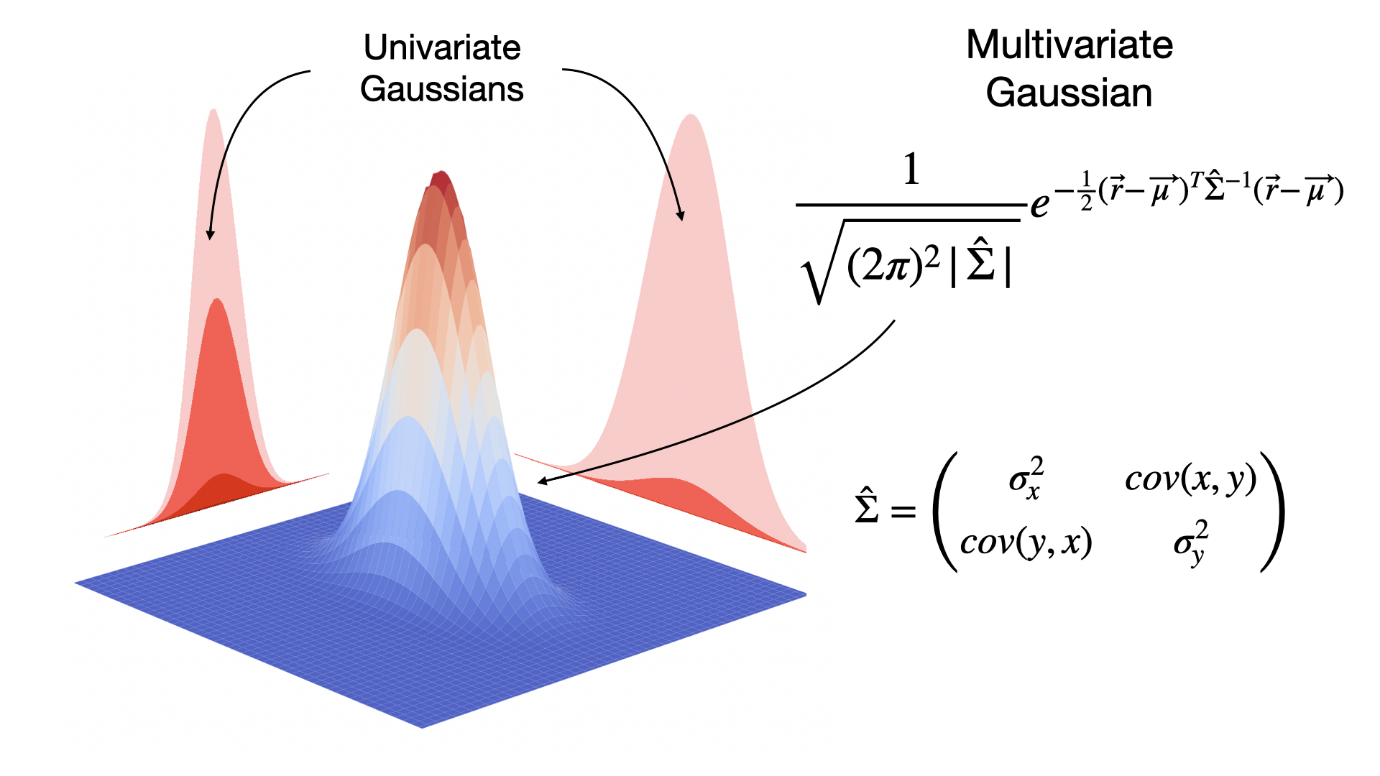
* Measure of dispersion of the data, away from the mean
* N = number of observations
* σ = standard deviation

# Multivariate Gaussian Distribution

1. Gaussian distribution

* Univariate Gaussian distributions are determined by μ (mean) and σ (standard deviation)

1. Multivariate Gaussian distribution



* The probability of a value x occurring is given by the joint probability of x1 and x2
* μ1 and μ2
* σ2 1 and σ2 2
* X1 ~ N(μ1 = 0, σ2 1 = 1)
* X2 ~ N(μ2 = 0, σ2 2 = 1)
* Need a function to estimate probability of x in multivariate Gaussian distribution
* Ellipse = variance + covariance
* Mean = center of joint distribution
* Covariance matrix
* Generalizes the univariate Gaussian distribution to higher dimensions
* More than 1 variable
* Instead of values, we now have vectors
* Multivariate Gaussian distributions need μ, σ2 and the covariance Ʃ
* Covariance matrix: captures σ2 and Ʃ

1. Covariance

* Measure of joint probability of 2 random variables
* Examples:
* Cov(x1, x2) = 0
* x1 and x2 are not correlated
* Cov(x1, x2) > 0
* The bigger x1, the bigger x2
* Cov(x1, x2) < 0
* The bigger x1, the smaller x2

1. Covariance matrix

* Square matrix with the covariance of each pair of variables
* Symmetric
* The diagonal contains the variances, i.e., the covariance of each variable with itself
* The covariance matrix provides a succinct way to summarize the covariance of all pairs of variables
* Covariance matrix – general
* Examples:
* Cov(x1, x2) = 0
* Cov(x1, x2) > 0
* Cov(x1, x2) < 0

1. Summary

* Multivariate Gaussian Distribution functions
* Cov(x1, x2) = 0
* Cov(x1, x2) > 0
* Cov(x1, x2) < 0
* The probability of a value x occurring is given by the joint probability of x1 and x2

# Gaussian Process

1. Gaussian process

* **Gaussian distribution** -> probability distribution of **scalars**
* **Multivariate Gaussian** -> probability distribution of **vectors**
* **Gaussian process** -> probability distribution of **functions**
* Hyperparameter response function

1. Overview

* Gaussian process = GP = f(x)
* GP = Gaussian distribution over functions
* x are functions
* Think of a function, x, as a **very long vector**, each entry in the vector specifying the function value f(x)
* The distribution of a Gaussian process is the joint distribution of all those (infinitely many) random variables / functions
* The joint distribution functions determines the shape of the GP

1. Formula

* f(x) ~ GP(m(x), K(x, x’))
* m(x) = E[f(x)]
* K(x, x’) = E[(f(x) – m(x)) \* (f(x’) – m(x’))]
* Computationally -> infinitely dimensional
* If we evaluate f(x) for a finite number of x, we can predict f(x) for all other values with an estimate plus an uncertainty
* GP -> prior
* F(x) evaluations -> Data
* Update f(x) -> the posterior

1. Joint probability

* See slides

1. Multivariate Gaussian Theorem

If

* The marginal:
* The posterior:

1. Gaussian Regression