Probability for ML Cheatsheet

Compiled by Tu Nguyen Ngoc. Material based on Kelvin Murphy's book (@probml). Please share comments, suggestions, and errors at github.com/nguyentuss/Probability-for-Machine-Learning-Cheatsheet.

Last Updated March 5, 2025

1 Introduction

1.1 Supervised Learning

The task T is to learn a mapping f from $x \in X$ to $y \in Y$. The x are also called the **features**. The output y is called the **label**. The experience E is given in the form of a set of N input-output pair $\mathcal{D} = \{(x_n, y_n)\}, n = 1 \to N$ is called **training set**. (N is called the **sample size**. The performance P depends on the type of output we want to predict.

1.2 Classification

In classification problem, the output space is a set of C labels called **classes**, $Y = \{1, 2, ..., C\}$. The problem predicting the class label given a input is called **pattern recognition**. The goal of supervised learning in classification problem is want to predict the label. A common way to measure the perform on this task is called **misclassification rate**.

$$\mathcal{L}(\boldsymbol{\theta}) \triangleq \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \mathbb{I}\left(y_n \neq f(x_n; \boldsymbol{\theta})\right)$$

Where $\mathbb{I}(e)$ is indicator function, which return 1 if the conditional is true, return 0 otherwise. We can also use the notation **loss function** $l(y, \hat{y})$.

$$\mathcal{L}(\boldsymbol{\theta}) \triangleq \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \ell\left(y_n, f(x_n; \boldsymbol{\theta})\right)$$

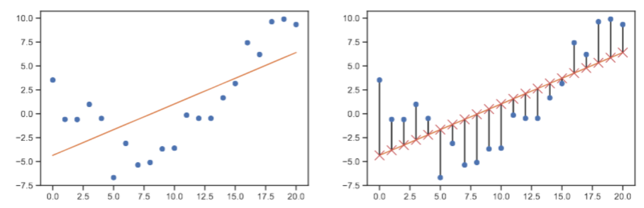
1.3 Regression

Similarly to the classification problem, but now the output in regression are a real-value $y \in \mathbb{R}$ instead the discrete value $y \in \{1, ..., C\}$; this is known as **regression**. So we need to use a different loss function. For regression, the most common choice is to use quadratic loss, or ℓ_2 loss (L2 normalization)

$$\ell_2(y, \hat{y}) = (y - \hat{y})^2$$

This penalizes large residuals $y - \hat{y}$. The empirical risk when use quadratic risk is equal to the **Mean squared error** or **MSE**.

$$MSE(\boldsymbol{\theta}) = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} (y_n - f(x_n; \boldsymbol{\theta}))^2$$



An example of the regression model in 1d data, we can fix the data using the **linear regression** model.

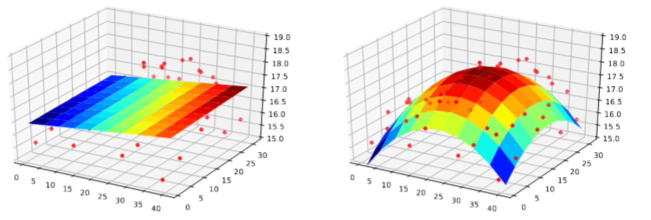
$$f(x; \boldsymbol{\theta}) = b + wx$$

Where w is the **slope**, b is the **bias**, and θ are the parameters of the model, we can minimize the sum square error.

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} MSE(\boldsymbol{\theta})$$

If we have multiple input features, we can write

$$f(\mathbf{x}; \boldsymbol{\theta}) = b + w_1 x_1 + ... + w_D x_D = b + \mathbf{w}^{\mathbf{T}}\mathbf{x}$$



(a) (b)

We can improve the fit by using a **Polynomial regression** model with degree $\mathcal{D}$. This now have the form

$$f(x; \mathbf{w}) = \mathbf{w}^{\mathbf{T}} \phi(x)$$

Where $\phi(x)$ are the feature vector derived from the input

$$\phi(x) = [1, x, x^2, ..., x^D]$$

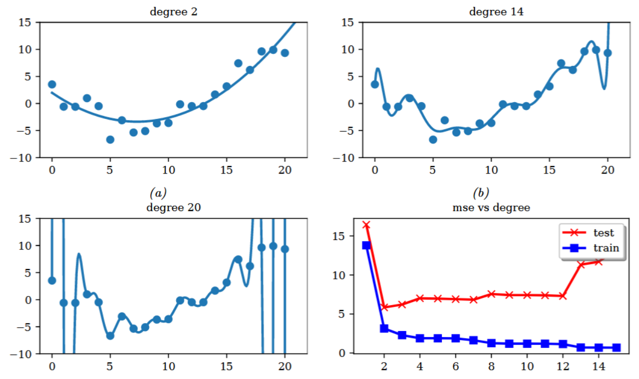
1.4 Overfitting

Empirical risk (training loss function)

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{\text{train}}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{\text{train}}} \ell(y, f(x; \boldsymbol{\theta}))$$

The difference $\mathcal{L}(\boldsymbol{\theta}; p^*) - \mathcal{L}(\boldsymbol{\theta}; D_{train})$ called **generalization gap**. If a model has a large generalization gap (i.e., low empirical risk but high population risk), it is a sign that it is overfitting. In practice we don't know p^*. However, we can partition the data we do have into two subsets, known as the training set and the **test set**. Then we can approximate the population risk using the **test risk**:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{\text{test}}) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{\text{test}}} \ell(y, f(x; \boldsymbol{\theta}))$$



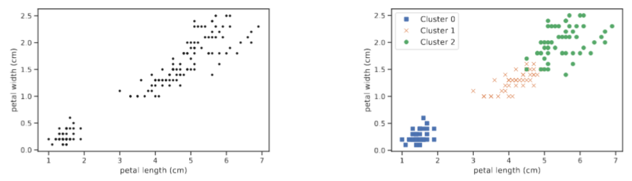
(a) (b) (c) (d)

We can make the training loss function to zero if we increase the degree $\mathcal{D}$, but it will increase the testing loss function. The purpose about the prediction accuarcy on new data, A model that fit the training data but which is too much complex. It will call the **overfitting**. If **D** is too small, the model will be **underfitting**.

1.5 Unsupervised learning

In supervised learning, we assume that each input x in training set have a output targets y, and our goal is to learn the input-output mapping. Although this is useful, and difficult, supervised learning is just a to find a mathematical function to fit the data points. So go back to the unsupervised learning, we will opposed to just learning a mapping. We just get $\mathcal{D} = \{(x_n : n = 1 : N)\}$ without any ouputs y_n.This is called **unsupervised learning**.
From a probabilistic perspective, we can view the task of unsupervised learning as fitting an unconditional model of the form $p(x)$, which can generate new data x, whereas supervised learning involves fitting a conditional model, $p(y|x)$, which specifies (a distribution over) outputs given inputs.
Unsupervised learning avoids the needs of collect large labeled datasets for training, which can be often time comsuming and expensive and does not rely on manually labeled data or predefined categories, unlike supervised learning, which learns from labeled examples. Instead, unsupervised learning finds patterns, structures, or groupings in the data based on inherent similarities or relationships.

1.6 Clustering



(a) (b)

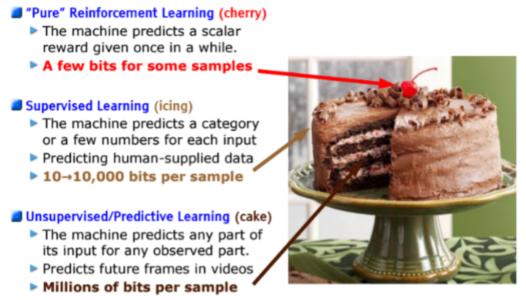
A simple example of unsupervised learning is the problem of finding clusters in data. The goal is to partition the input into regions that contain "similar" points.

1.7 Self-supervised learning

Self-supervised learning that automatically generating **labels** from **unlabeled data**. Try to learn to predict a color image from a grayscale image, or to mask out words in a sentence and then try to predict them given the surrounding context. The hope is that the resulting predictor $\hat{x}_1 = f(x_2; \boldsymbol{\theta})$. Where x_2 is the observed input and $\hat{x}_1$ is the predict output, will learn useful features from the data, that can be used in standard.

1.8 Reinforcement learning

The system or agent has to learn how to interact with its environment. For example, creating a bot playing Mario, a bot will interact and integration with the world, run left or right or the bot will jump if they see a block stone.(Click to see the detail)



- "Pure" Reinforcement Learning (cherry)
  - The machine predicts a scalar reward given once in a while.
  - A few bits for some samples
- Supervised Learning (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - 10→10,000 bits per sample
- Unsupervised/Predictive Learning (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - Millions of bits per sample

1.9 Preprocessing discrete input data

1.9.1 One-hot encoding

When we have categorical features, we need to scale it into numerical values, so that the compute makes sense. The standard way to preprocess such categorical variables is to use a **one-hot encoding**. one-hot(x) $= [\mathbb{I}(x = 1), ..., \mathbb{I}(x = K)]$. If a variable x has K values (3 colors red,green,blue), the corresponding one-hot vectors will be one-hot(red)=[1,0,0], one-hot(green)=[0,1,0], one-hot(blue)=[0,0,1].

1.9.2 Feature crosses

Converting the original datasets into a **wide format**, with more many columns. Suppose we want to predict the fuel efficiency of a vehicle given two categorical input variables:

- x_1: The type of car (SUV, Truck, Family car).
- x_2: The country of origin (USA, Japan).

Using one-hot encoding, we represent these variables as separate binary indicators:

$$\phi(x) = [1, I(x_1 = S), I(x_1 = T), I(x_1 = F), I(x_2 = U), I(x_2 = J)]$$

However, this encoding does not capture interactions between the features. For example, we expect trucks to be less fuel-efficient overall, but perhaps trucks from the USA are even less efficient than trucks from Japan. This cannot be captured using a simple linear model. We define a new composite feature representing all possible pairs:

$$(\text{Car type}, \text{Country}) = \{(S, U), (T, U), (F, U), (S, J), (T, J), (F, J)\}$$

The new model becomes:

$$f(x; w) = w^T \phi(x)$$

Express the equation we have:

$$\begin{aligned} f(x; w) = w_0 &+ w_1 I(x_1 = S) + w_2 I(x_1 = T) + w_3 I(x_1 = F) \\ &+ w_4 I(x_2 = U) + w_5 I(x_2 = J) + w_6 I(x_1 = S, x_2 = U) \\ &+ w_7 I(x_1 = T, x_2 = U) + w_8 I(x_1 = F, x_2 = U) \\ &+ w_9 I(x_1 = S, x_2 = J) + w_{10} I(x_1 = T, x_2 = J) \\ &+ w_{11} I(x_1 = F, x_2 = J) \end{aligned}$$

2 Probability Unvariate Models

2.1 Introduction

Calling **sample space** $\mathcal{X}$ are all possible experiences, and **event** will be a subset of the **sample space**.

2.1.1 Union

$$Pr(A \wedge B) = Pr(A, B)$$

If independent events

$$Pr(A \wedge B) = Pr(A)Pr(B)$$

We say a set of variables $X_1, ..., X_n$ is (mutually) independent if the joint can be written as a product of marginals for all subsets $\{X_1, ..., X_m\} \subseteq \{X_1, ..., X_n\}$

$$p(X_1, X_2, ..., X_n) = \prod_{i=1}^{m} p(X_i)$$

2.1.2 Disjoint

$$Pr(A \vee B) = Pr(A) + Pr(B) - Pr(A \wedge B)$$

2.1.3 Conditional probability

$$Pr(B|A) \triangleq \frac{Pr(A, B)}{Pr(A)}$$

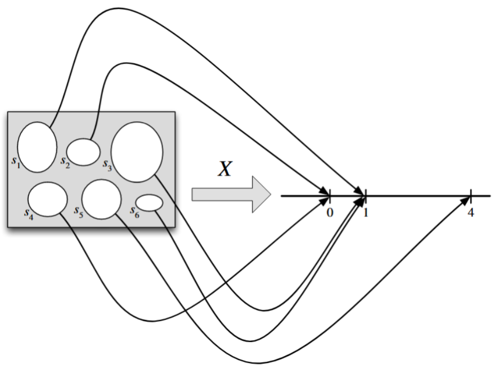
If events A and B are conditionally independent given event C

$$Pr(A, B|C) = Pr(A|C)Pr(B|C)$$

Be careful, we say X_1, X_2, X_3 are mutually independent if the following conditions hold:

$$\begin{aligned} p(X_1, X_2, X_3) &= p(X_1)p(X_2)p(X_3) \\ , p(X_1, X_2) &= p(X_1)p(X_2) \\ , p(X_1, X_3) &= p(X_1)p(X_3) \\ , p(X_2, X_3) &= p(X_2)p(X_3) \end{aligned}$$

2.2 Random variables



Given an experiment with sample space $\mathbb{S}$, a random variable(r.v.) is a function mapping from the sample $\mathbb{S}$ to the real value $\mathbb{R}$.

2.2.1 Discrete random variables

If sample space $\mathbb{S}$ have finite or countable, it is called a Discrete r.v. Denote probability of events in $\mathbb{S}$ and have value x by $Pr(X = x)$. This called probability mass function or **pmf** as a function which compute the probability of events which have the value x.

$$p(x) \triangleq Pr(X = x)$$

The pmf satisfied $0 \leq p(x) \leq 1$ and $\sum_{x \in \mathcal{X}} p(x) = 1$

2.2.2 Continuous random variables

If $X \in \mathbb{R}$, it is called the continuous r.v. The value or no longer create a finite set of distinct possible values it can take on.

2.2.3 Cumulative distribution function (cdf)

$$P(x) \triangleq Pr(X \leq x)$$

We can compute the probability of any interval

$$P(a \leq x \leq b) = P(b) - P(a - 1)$$

In discrete r.v, the cdf will compute

$$P(x) = \sum_{x \in \mathcal{X}} p(x)$$

In continuous r.v, the cdf will compute

$$P(x) = \int_{x \in \mathcal{X}} p(x)$$

2.2.4 Probability density function (pdf)

Define the pdf as a derivative of the cdf

$$p(x) \triangleq \frac{d}{dx} P(x)$$

As the size of interval get smaller, we can write

$$Pr(x < X < x + dx) \approx p(x)dx$$

2.2.5 Quantiles

If the cdf P is monotonically increase, it has an inverse, called the **inverse cdf**. If P is the cdf of X, then $P^{-1}(q)$ is the value x_q that $Pr(X \leq x_q) = q$; this call q'th quantiles of P.

2.2.6 Sets of related random variables

Suppose we have two r.v X and Y. We can define joint of distribution $p(x, y) = Pr(X = x, Y = y)$ for all possible value of x and y. We can respresent the all possible value by a 2d table. For example:

$p(X, Y)$	$Y = 0$	$Y = 1$
$X = 0$	0.2	0.3
$X = 1$	0.3	0.2

$Pr(X = 0, Y = 1) = 0.3$, $\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) = 1$

2.2.7 Moments of a distribution

Mean is the average of the distribution, other name is called **expected value**, denoted as μ. For continuous r.v, the mean is defined as follow:

$$\mathbb{E}[X] = \int_{x \in \mathcal{X}} xp(x)dx$$

If the integral is not finite, the mean is not defined. For discrete r.v, the mean is defined as follow:

$$\mathbb{E}[X] = \sum_{x \in \mathcal{X}} xp(x)dx$$

Since the mean is linear, we have the **linearity of expectation**:

$$\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$$

For n set random variables, we can show the sum of expectation as follow:

$$\mathbb{E}[\sum X_i] = \sum \mathbb{E}[X_i]$$

If they are independent, the expectation of product is defined:

$$\mathbb{E}[\prod X_i] = \prod \mathbb{E}[X_i]$$

When we have two or more dependent r.v, we can compute the moment of one given the others.

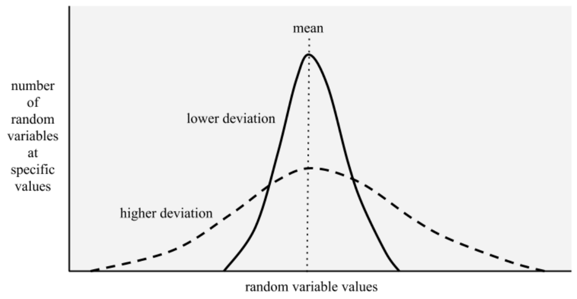
$$\mathbb{E}[X] = \mathbb{E}_Y[\mathbb{E}[X|Y]]$$

There is a similar formula for the variance.

$$\mathbb{V}[X] = \mathbb{E}_Y[\mathbb{V}[X|Y]] + \mathbb{V}_Y[\mathbb{E}[X|Y]]$$

Variance is a measure how "spread" a value compared with the mean of the distribution, denoted as σ^2. This is define as follow:

$$\begin{aligned} \mathbb{V}[X] \triangleq \mathbb{E}[(X - \mu)^2] &= \int (x - \mu)^2 p(x)dx \\ = \int x^2 p(x)dx + \mu \int p(x)dx &- 2\mu \int xp(x)dx \\ &= \mathbb{E}[X^2] - \mu^2 \end{aligned}$$



The **standard deviation** is defined as

$$std[X] = \sqrt{\mathbb{V}[X]} = \sigma$$

Lower deviation, the distribution is closer to the mean. High deviation, the distribution is far away from the mean.
The variance of a shifted and scaled version of a random variable is given by

$$\mathbb{V}[aX + b] = a^2\mathbb{V}[X]$$

If we have a set of n independent random variables, the variance of their sum is given by the sum of their variances:

$$\mathbb{V}[\sum X_i] = \sum \mathbb{V}[X_i]$$

The variance of their product can also be derived:

$$\begin{aligned}\mathbb{V}[\prod X_i] &= \mathbb{E}[(\prod X_i)^2] - (\mathbb{E}[(\prod X_i)])^2 \\ &= \prod(\sigma_i^2 + \mu_i^2) - \prod \mu_i^2\end{aligned}$$

Mode of a distribution
The **mode** of a distribution is the value with the highest probability mass or probability density

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x})$$

If the distribution is multimodal, this may not be unique. Like the function have 2 global extrema(means having 2 highest probability), this can may not be unique.

2.3 Bayes' Rule

2.3.1 Bayes' Rule, and with extra conditioning

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

$$P(A = a|B = b, C = c) = \frac{P(B = b|A = a, C = c)P(A = a|C = c)}{P(B = b|C = c)}$$

The term $p(A)$ represents what we know about possible values of A before we see any data; this is called the **prior distribution**. (If A has K possible values, then $p(A)$ is a vector of K probabilities, that sum to 1.) The term $p(B \mid A = a)$ represents the distribution over the possible outcomes B we expect to see if $A = a$; this is called the **observation distribution**. When we evaluate this at a point corresponding to the actual observations, b, we get the function $p(B = b \mid A = a)$, which is called the **likelihood**. (Note that this is a function of a, since b is fixed, but it is not a probability distribution, since it does not sum to one.) Multiplying the prior distribution $p(A = a)$ by the likelihood function $p(B = b \mid A = a)$ for each a gives the unnormalized joint distribution $p(A = a, B = b)$. We can convert this into a normalized distribution by dividing by $p(B = b)$, which is known as the **marginal likelihood**, since it is computed by marginalizing over the unknown A:

$$p(B = b) = \sum_{a' \in \mathcal{A}} p(A = a')p(B = b \mid A = a') = \sum_{a' \in \mathcal{A}} p(A = a', B = b)$$

Odds Form of Bayes' Rule

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

The *posterior odds* of A are the *likelihood ratio* times the *prior odds.*

2.4 Bernoulli and binomial distributions

Given experiment tossing a coin, where the probability of event it lands head is $0 \leq \theta \leq 1$, Y = 1 denoted that event, Y=0 denote the events that the coin lands tail. So $p(Y = 1) = \theta$ and $p(Y = 0) = 1 - \theta$, This called the **Bernoulli distribution**, it can be written as follows

$$Y \sim Ber(\theta)$$

Where $\sim$ is mean that "is sampled from" or "is distribution from", Y have model $Ber(\theta)$ The pmf is defined as follow:

$$\text{Ber}(y \mid \theta) = \begin{cases} 1 - \theta & \text{if } y = 0 \\ \theta & \text{if } y = 1 \end{cases}$$

We can also write this:

$$\text{Ber}(y \mid \theta) \triangleq \theta^y(1 - \theta)^{1-y}$$

Bernoulli is a special case of **Binomial distribution**. While Bernoulli tossing a coin 1 time, Binomial distribution tossing a coin N times, we observe a set of N Bernoulli trials, denoted $y_n \sim Ber(\cdot \mid \theta)$ (the dot mean that is the placeholder, it can be any specific value (eg. $y = 0$ or $y = 1$). Define s to be a number of heads, $s \triangleq \sum_{n=1}^{N} \mathbb{I}(y_n = 1)$. The distribution of s given by a binomial distribution:

$$Bin(s \mid \theta, N) \triangleq \binom{N}{s}\theta^s(1 - \theta)^{N-s}$$

When we want to predict a binary variable $y \in \{0, 1\}$ given some inputs $x \in X$, we need to use a conditional probability distribution of the form

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = Ber(y \mid f(\mathbf{x}; \boldsymbol{\theta}))$$

Where $f(\mathbf{x}; \boldsymbol{\theta})$ is some function that predict the mean parameter of the output distribution. But this will require that $0 \leq f(\mathbf{x}; \boldsymbol{\theta}) \leq 1$ to put into the conditional probability. So to avoid this, we can use any uncontrained function with this:

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = Ber(y \mid \sigma(f(\mathbf{x}; \boldsymbol{\theta})))$$

Where $\sigma()$ is the **sigmoid** or **logistic** function, defined as follows:

$$p = logistic(a) = \sigma(a) \triangleq \frac{1}{1 + e^{-a}} = \frac{e^x}{1 + e^x}$$

where $a = f(\mathbf{x}, \boldsymbol{\theta})$. The inverse of this is called the **logic function**.

$$a = logit(p) = \sigma^{-1}(p) \triangleq \log\left(\frac{p}{1 - p}\right)$$

So the sigmoid will tranform the function in $\mathbb{R}$ into probability $[0, 1]$ and the logit will transform the probability into real numbers, making it easier to model with linear regression techniques.
Some useful properties of these functions.

$$\sigma(x) \triangleq \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}$$

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x))$$

$$1 - \sigma(x) = \sigma(-x)$$

$$\sigma_+(x) \triangleq \log(1 + e^x) \triangleq \text{softplus}(x)$$

$$\frac{d}{dx}\sigma_+(x) = \sigma(x)$$

In particular, an issue commonly seen with activation functions is getting zero (or close to zero) gradient flow during backpropagation
When $x = 0$:

$$f(0) = \frac{1}{1 + e^0} = \frac{1}{2}$$

$$f'(0) = \frac{1}{2} \cdot \left(1 - \frac{1}{2}\right) = \frac{1}{4} = 0.25$$

This is **not very small** (but smaller than 1).
When $x \gg 0$ (large positive):

$$f(x) \to 1, \quad f'(x) = 1 \cdot (1 - 1) = 0$$

Gradient approaches 0.
When $x \ll 0$ (large negative):

$$f(x) \to 0, \quad f'(x) = 0 \cdot (1 - 0) = 0$$

Gradient also approaches 0.

2.5 Categorical and multinomial distributions

When the Bernoulli have only 2 classes, the categorical will represent a distribution over a finite sets of labels $y \in \{1..C\}$, which generalizes the Bernoulli to $C > 2$ values. The categorical discrete probability distribution with each parameter corresponding to one class.

$$\text{Cat}(y \mid \boldsymbol{\theta}) \triangleq \prod_{c=1}^{C} \theta_c^{\mathbb{I}(y=c)}$$

In other words, $p(y = c \mid \boldsymbol{\theta}) = \theta_c$. Note that the parameters are constrained so that $0 \leq \theta \leq 1$. We can write the categorical distribution in another way by converting the discrete variable y into a **one-hot vector** with C elements. For example, if $C = 3$, we encode the classes $1, 2$ and 3 as $(1, 0, 0), (0, 1, 0)$ and $(0, 0, 1)$. Consider a fair 6-sided die, where each face corresponds to an outcome:

$$y \in \{1, 2, 3, 4, 5, 6\}$$

Each face has an equal probability of occurring, so the categorical distribution parameter θ is:

$$\theta = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$$

If we roll a 3, then the one-hot encoding for $y = 3$ is:

$$(0, 0, 1, 0, 0, 0)$$

Substituting into the PMF:

$$P(y = 3 \mid \theta) = \theta_1^0\theta_2^0\theta_3^1\theta_4^0\theta_5^0\theta_6^0$$

Since any number raised to the power of 0 is **1**, this simplifies to:

$$P(y = 3 \mid \theta) = \theta_3 = \frac{1}{6}$$

Thus, for a fair die, the probability of rolling a 3 is $\frac{1}{6}$.
Using one-hot encoding, each possible outcome can be written as:

$$\begin{aligned} y = 1 &\to (1, 0, 0, 0, 0, 0) \\ y = 2 &\to (0, 1, 0, 0, 0, 0) \\ y = 3 &\to (0, 0, 1, 0, 0, 0) \\ y = 4 &\to (0, 0, 0, 1, 0, 0) \\ y = 5 &\to (0, 0, 0, 0, 1, 0) \\ y = 6 &\to (0, 0, 0, 0, 0, 1) \end{aligned}$$

Now, suppose we have a biased die with the following probabilities:

$$\theta = (0.10, 0.15, 0.30, 0.20, 0.15, 0.10)$$

The probability of rolling a 3 is now:

$$P(y=3 \mid \theta) = \theta_3 = 0.30$$

The categorical distribution is a special case of the **multinomial distribution**. Suppose we observe N categorical trials, $y_n \sim Cat(\cdot \mid \boldsymbol{\theta})$, for $n = 1:N$. Think of rolling a C-sided dice N times. Let us define y to be a vector that counts the number of times each face show up $y_c = N_c \triangleq \sum_{n=1}^{N} \mathbb{I}(y_n = c)$. Now $\mathbf{y}$ is no longer one-hot, but is "multi-hot", since it has a non-zero entry for every value of c that was observed across all N trials. The distribution of $\mathbf{y}$ is given by the **multinomial distribution**:

$$\mathcal{M}(\mathbf{y} \mid N, \boldsymbol{\theta}) \triangleq \binom{N}{y_1 \dots y_C} \prod_{c=1}^{C} \theta_c^{y_c} = \binom{N}{N_1 \dots N_C} \prod_{c=1}^{C} \theta_c^{N_c}$$

In the conditional case, we can define

$$p(y \mid x, \theta) = \text{Cat}(y \mid f(x;\theta))$$

which we can also write as

$$p(y \mid x, \theta) = \mathcal{M}(y \mid 1, f(x;\theta))$$

We require that $0 \le f_c(x;\theta) \le 1$ and $\sum_{c=1}^{C} f_c(x;\theta) = 1$.
To avoid the requirement that f directly predict a probability vector, it common to pass the output from f into the **softmax** function, also called the **multinomial logit**. It is defined as follows:

$$softmax(\mathbf{a}) \triangleq \left[\frac{e^{a_1}}{\sum_{c'=1}^{C} e^{c'}}, \dots, \frac{e^{a_c}}{\sum_{c'=1}^{C} e^{c'}}\right]$$

Similar with sigmoid, it will convert real value $\mathbb{R}^C$ into probability $[0,1]^C$. But there one weakness in this formula, suppose we have

$$p_c = \frac{e^c}{Z(\mathbf{a})} = \frac{e^c}{\sum_{c'=1}^{C} e^{c'}}$$

Where $\mathbf{a} = f(\mathbf{x}, \boldsymbol{\theta})$, are the logits. Supose $\mathbf{a} = (1000, 1000, 1000)$, the computer will compute inf. Similarly, $\mathbf{a} = (-1000, -1000, -1000)$ now it will compute 0. To avoid this, we use the following trick:

$$log \sum_{c'=1}^{C} e^{c'} = m + log \sum_{c'=1}^{C} e^{c'-m}$$

This will hold any m, so that the exponential will smaller. It is common to use $m = \max_c a_c$ which ensures the the largest will be zero, so you will definitely not overflow, and even if you underflow, the answer will be sensible. This is known as the **log-sum-exp trick**.

$$lse(\mathbf{a}) \triangleq \log \sum_{c'=1}^{C} e^{c'}$$

We have

$$p(y = c|\mathbf{x}) = \frac{exp(a_c - m)}{\sum_{c'} exp(a_{c'} - m)}$$
$$= e^{a_c - lse(\mathbf{a})}$$

The loss of the softmax function:

$$J(\mathbf{W}; \mathbf{x}, \mathbf{y}) = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ji} \log(a_{ji})$$
$$= -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ji} \log\left(\frac{\exp(\mathbf{W}_j^T \mathbf{x}_i)}{\sum_{k=1}^{C} \exp(\mathbf{W}_k^T \mathbf{x}_i)}\right)$$

Once again, we use **Stochastic Gradient Descent (SGD)** here.
For a single data point $(\mathbf{x}_i, \mathbf{y}_i)$, we have:

$$J_i(\mathbf{W}) \triangleq J(\mathbf{W}; \mathbf{x}_i, \mathbf{y}_i) = -\sum_{j=1}^{C} y_{ji} \log\left(\frac{\exp(\mathbf{W}_j^T \mathbf{x}_i)}{\sum_{k=1}^{C} \exp(\mathbf{W}_k^T \mathbf{x}_i)}\right)$$
$$= -\sum_{j=1}^{C} y_{ji} \mathbf{W}_j^T \mathbf{x}_i - y_{ji} \log\left(\sum_{k=1}^{C} \exp(\mathbf{W}_k^T \mathbf{x}_i)\right)$$
$$= -\sum_{j=1}^{C} y_{ji} \mathbf{W}_j^T \mathbf{x}_i + \log\left(\sum_{k=1}^{C} \exp(\mathbf{W}_k^T \mathbf{x}_i)\right)$$

In the final transformation, we used the observation: $\sum_{j=1}^{C} y_{ji} = 1$ since it represents the sum of probabilities.
Next, we use the formula:

$$\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}} = \left[\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}_1}, \frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}_2}, \dots, \frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}_C}\right]$$

For each column j, the gradient is computed using equation:

$$\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}_j} = -y_{ji}\mathbf{x}_i + \frac{\exp(\mathbf{W}_j^T \mathbf{x}_i)\mathbf{x}_i}{\sum_{k=1}^{C} \exp(\mathbf{W}_k^T \mathbf{x}_i)}$$
$$= -y_{ji}\mathbf{x}_i + a_{ji}\mathbf{x}_i = \mathbf{x}_i(a_{ji} - y_{ji})$$

$$= \mathbf{x}_i e_{ji}, \quad \text{where } e_{ji} = a_{ji} - y_{ji}$$

Here, the value e_{ji} can be interpreted as the **prediction error**.
At this point, we obtain a very elegant expression. Combining (4) and (5), we derive:

$$\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}} = \mathbf{x}_i[e_{i1}, e_{i2}, \dots, e_{iC}] = \mathbf{x}_i e_i^T$$

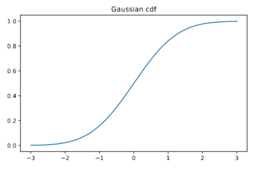
From this, we can conclude:

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} = \sum_{i=1}^{N} \mathbf{x}_i e_i^T = \mathbf{x}E^T$$

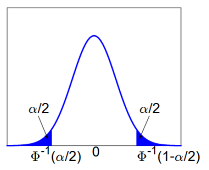
where $E = A - Y$. This general gradient formula enables easy application to **Batch Gradient Descent**, **Stochastic Gradient Descent (SGD)**, and **Mini-batch Gradient Descent**.
Assuming we use SGD, the weight matrix $\mathbf{W}$ update rule is:

$$\mathbf{W} = \mathbf{W} + \eta \mathbf{x}_i (y_i - a_i)^T$$

2.6 Univariate Gaussian (normal) distribution



(a)



(b)

We defined the cdf of the Gaussian by:

$$\phi(y; \mu, \sigma^2) \triangleq \int_{-\infty}^{y} \mathcal{N}(z, \mu, \sigma^2) dz$$

We can implemented using

$$\phi(y; \mu, \sigma^2) = \frac{1}{2}[1 + erf(z/\sqrt{2}]$$

where $z = (y - \mu)/\sigma$ and $erf(u)$ is the **error function**,defined as

$$erf(u) \triangleq \frac{2}{\sqrt{\pi}} \int_0^u e^{-t} dt$$

The pdf of the Gaussian is given by

$$\mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-\mu)^2}$$

Where $\sqrt{2\pi\sigma^2}$ is the normalization constant needed to ensure the density needed to ensure the density integrates to 1.
The mean of the distribution

$$\mathbb{E}[\mathcal{N}(\cdot \mid \mu, \sigma^2)] = \mu$$

The standard deviation is defined as

$$std[\mathcal{N}(\cdot \mid \mu, \sigma^2)] = \sigma$$

It can be helpful to make a parameters of the Gaussian be a function of some input variables, we want to create a conditional density model of the form

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid f_\mu(\mathbf{x}; \boldsymbol{\theta}), f_\sigma(\mathbf{x}; \boldsymbol{\theta})^2)$$

Where $f_\mu(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}$ predicts a mean, and $f_\sigma(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}_+$ predicts a variance.
Assume the variance is fixed, and is independent with the input. This is called **homoscedastic regression**. Furthermore, it is common to assume the mean is a linear function of the input.

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \mathbf{W}^T\mathbf{x} + b, \sigma^2)$$

Where $\boldsymbol{\theta} = (\mathbf{W}, b, \sigma^2)$. However we can also make the variance depend on the input; this is called **heteroskedastic regression**. In linear regression, we have

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \mathbf{W}_\mu^T \mathbf{x} + b, \sigma_+(\mathbf{W}_\sigma^T \mathbf{x}))$$

where $\boldsymbol{\theta} = (\mathbf{W}_\mu, \mathbf{W}_\sigma)$ are two forms of regression weights, and $\sigma_+(x)$ softplus function, that mas form $\mathbb{R}$ to $\mathbb{R}_+$.
The Gaussian are famous because, it only have two parameters, mean and variance. Second, sum of independent r.v will approximate be a Gaussian distribution, making a good choice of modeling the "noise" data point.
When the variance nearly to become 0. The gaussian will approach to infinity narrow. We can defined

$$\lim_{\sigma \to 0} \mathcal{N}(y \mid \mu, \sigma^2) \to \delta(y - \mu)$$

Where δ is the dirac delta function, is defined

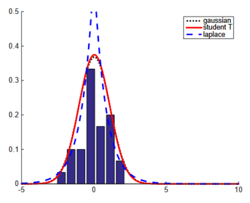
$$\delta(x) = \begin{cases} +\infty & \text{if x=0} \\ 0 & \text{if x} \neq 0 \end{cases}$$

We also defined

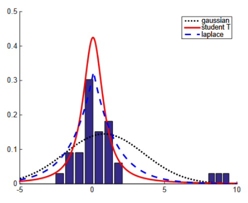
$$\delta_y(x) = \begin{cases} +\infty & \text{if x=y} \\ 0 & \text{if x} \neq y \end{cases}$$

Where $\delta_y(x) = \delta(x - y)$

2.7 Some common other unvariative distribution



(a)



(b)

2.7.1 Student t distribution

$$\mathcal{T}(y|\mu,\sigma^2,\nu) \propto \left[1+\frac{1}{\nu}\left(\frac{y-\mu}{\sigma}\right)^2\right]^{-\frac{\nu+1}{2}}$$

Difference with the Normal(Gaussian) distribution, the **student t** will have a heavy tails, which mean they will closer to the **mean**. Where μ is the mean, $\sigma > 0$ is the scale parameter(not standard deviation), and ν is a degree of freedom (since large ν will act like a Gaussian). Note that the Student distribution has the following properties:

$$\text{mean} = \mu, \quad \text{mode} = \mu, \quad \text{var} = \frac{\nu\sigma^2}{(\nu-2)}$$

The mean is only defined if $\nu > 1$. The variance is only defined if $\nu > 2$. For $\nu \gg 5$, the Student distribution rapidly approaches a Gaussian distribution and loses its robustness properties. It is common to use $\nu = 4$, which gives good performance in a range of problems.

2.7.2 Cauchy distribution

Since $\nu = 1$ the Student distribution is known as a **Cauchy** or **Lorentz** distribution.

$$\mathcal{C}(x \mid \mu,\gamma) = \frac{1}{\pi\gamma}\left[1+\left(\tfrac{x-\mu}{\gamma}\right)^2\right]^{-1}$$

Because the ν is small, they is a very heavy tail distribution.
The **Half Cauchy** distribution is a Cauchy (with $\mu = 0$) This has a form.

$$\mathcal{C}_{+}(x \mid \gamma) \triangleq \frac{2}{\gamma\pi}\left[1+\left(\tfrac{x}{\gamma}\right)^2\right]^{-1}$$

This is useful in Bayesian modeling, where we want to use a distribution over a positive real value, but finite density.

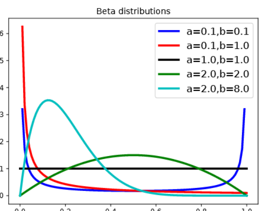
2.7.3 Laplace distribution

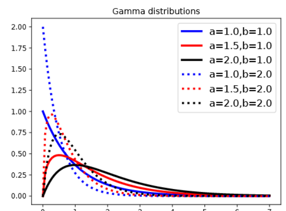
Another distribution with heavy tails is the **Laplace distribution**, also known as the **double sided exponential** distribution. This has the following pdf:

$$\text{Laplace}(y|\mu,b) \triangleq \frac{1}{2b}\exp\left(-\frac{|y-\mu|}{b}\right)$$

Here μ is a location parameter and $b > 0$ is a scale parameter. This distribution has the following properties:

$$\text{mean} = \mu, \quad \text{mode} = \mu, \quad \text{var} = 2b^2$$





2.7.4 Beta Distribution

The **beta distribution** has support over the interval $[0, 1]$ and is defined as follows:

$$\text{Beta}(x|a,b) = \frac{1}{B(a,b)}x^{a-1}(1-x)^{b-1}$$

where $B(a, b)$ is the **beta function**, defined by

$$B(a,b) \triangleq \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

where $\Gamma(a)$ is the Gamma function defined by

$$\Gamma(a) \triangleq \int_0^\infty x^{a-1}e^{-x}\,dx$$

We require $a, b > 0$ to ensure the distribution is integrable (i.e., to ensure $B(a, b)$ exists). If $a = b = 1$, we get the uniform distribution. If a and b are both less than 1, we get a bimodal distribution with "spikes" at 0 and 1; if a and b are both greater than 1, the distribution is unimodal.
Note that the distribution has the following properties

$$\text{mean} = \frac{a}{a+b}, \quad \text{mode} = \frac{a-1}{a+b-2}, \quad \text{var} = \frac{ab}{(a+b)^2(a+b+1)}$$

2.8 Transfomation of random variables

Suppose $\mathbf{x} \sim p()$ in some random variables, and $\mathbf{y} = f(\mathbf{x})$, is some transformation of it. We discuss how to compute $f(\mathbf{y})$

2.8.1 Discrete case

In discrete r.v, we can use pmf to compute the $p(y)$ by summing up the pmf of all $\mathbf{x}$ such that $f(\mathbf{x}) = \mathbf{y}$.

$$p_{\mathbf{y}}(\mathbf{y}) = \sum_{\mathbf{x}:f(\mathbf{x})=\mathbf{y}} p_{\mathbf{x}}(\mathbf{x})$$

Suppose $f(\mathbf{x}) = 1$ if $\mathbf{x}$ is even, $f(\mathbf{x}) = 0$ otherwise, and $p_{\mathbf{x}}(\mathbf{x})$ have a distribution $\{1, 2, ..., 10\}$, so $p_{\mathbf{y}}(1) = \sum_{x\in\{2,4,..,10\}} p_{\mathbf{x}}(x) = 0.5$ and $p_{\mathbf{y}}(0) = 0.5$ either. Note that in this example, $f(\mathbf{x})$ is **many-to-one** function.

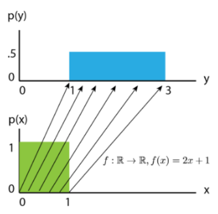
2.8.2 Continuous case

Unlikely with discrete r.v, the continuous does not have pmf, since $p_{\mathbf{x}}(\mathbf{x})$ is a density function. But we can work with cdf's.

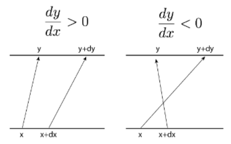
$$P_{\mathbf{y}}(\mathbf{y}) = Pr(Y \le \mathbf{y}) = Pr(f(\mathbf{x}) \le \mathbf{y}) = Pr(\mathbf{x} \in \{x \mid f(x) \le \mathbf{y}\}$$

If f invertible, we can derivative the cdf to have the pdf, but if not, we can use **Monte Carlo approximation** instead.

2.8.3 Invertible transformation (bijections or one-to-one)



(a)



(b)

We will discuss about the monotonic function (linear) and invertible function. Suppose $x \sim Uni(0, 1)$ and $y = f(x) = 2x + 1$. The transformation will like the figure (a) above. Now consider the general case for any $p_x(x)$ and any monotonic f: $\mathbb{R} \to \mathbb{R}$. Let $g = f^{-1}$, $y = f(x), x = g(y)$. We have

$$P_y(y) = Pr(f(X) \le y) = Pr(X \le f^{-1}(y)) = P_x(f^{-1}(y)) = P_x(g(y))$$

Taking derivative, we have

$$p_y(y) \triangleq \frac{d}{dy}P_y(y) = \frac{d}{dx}\frac{dx}{dy}P_x(g(y)) = \frac{dx}{dy}p_x(g(y))$$

Since $\frac{d}{dy} = \frac{dx}{dy}\frac{d}{dx}$ by the chain rule. We can derived the formula if f is a monotonically decrease function, to handle the general case we take the absolute to get

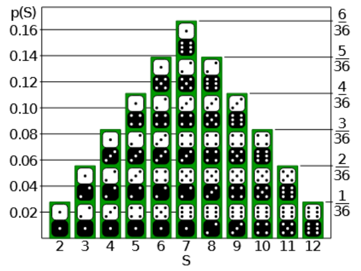
$$p_y(y) = p_x(x)|\frac{dx}{dy}|$$

This is call the **change variables** of the function.
Now, let discuss about the multivariate cases. Let f is invertible function that maps $\mathbb{R}^n \to \mathbb{R}^n$, with invertible g. $\mathbf{y} = f(\mathbf{x})$, defined

$$p_y(\mathbf{y}) = p_x(\mathbf{x})|det[\mathbf{J}_g(\mathbf{x})|$$

where $\mathbf{J}_g = \frac{d\mathbf{g}(\mathbf{y})}{d\mathbf{y}^{\mathbf{T}}}$ and $\mathbf{J}$ is Jacobian function.

2.8.4 Convolution theorem



Let $Y = X_1 + X_2$ such that X_1, X_2 is independent r.v. If this is the discrete case, define the formula

$$p(Y = y) = \sum_{-\infty}^{x} p(X_1 = x)p(X_2 = y - x)$$

In continuous case, we have

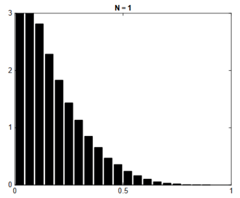
$$p(y) = \int p_1(x_1)p_2(y - x_1)dx_1$$

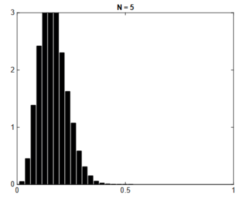
We can write equation as follows

$$p = p_1 \circledast p_2$$

where $\circledast$ represents the **convolution** operator.

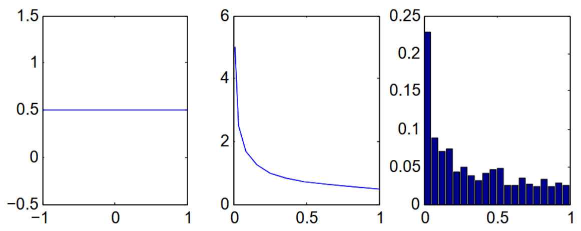
2.8.5 Central Limit theorem





Suppose we have N r.v(don't need to be a Gaussian) **independent** and **i.i.d**(independent and same distribution), sum of those r.v will approximate a Gaussian distribution.

2.8.6 Monte Carlo Approximation



Suppose $\mathbf{x}$ is a random variables, and $\mathbf{y} = f(\mathbf{x})$ is some function of $\mathbf{x}$. It is often difficult to compute $p(\mathbf{y})$. One simple but powerful is draw a specific number of samples from the $\mathbf{x}$ distribution, and then to use these samples (instead of the distribution) to approximate $p(\mathbf{y})$.
For example, suppose $x \sim \text{Unif}(-1, 1)$ and $y = f(x) = x^2$. We can approximate $p(y)$ by drawing many samples from $p(x)$ (using a uniform random number generator), squaring them, and computing the resulting empirical distribution, which is given by

$$p_s(y) \triangleq \frac{1}{N_s} \sum_{s=1}^{N_s} \delta(y - y_s)$$

3 Multivariate Models

3.1 Joint distributions for multiple random variables

3.1.1 Covariance

The **covariance** between two rv's X and Y measures the **direction** of **linear relationship** to which X and Y are (linearly) related. It quantifies how to random variables change together.

- Positive: If the one increase, the other also increase.
- Negative: If the one incrase while the other decrase
- Zero There are no relationship between the variables.

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

If $\mathbf{x}$ is a D-dimensional random vector, its **covariance matrix** is defined to be the following symmetric

$$\text{Cov}[\mathbf{x}] \triangleq \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^T] \triangleq \mathbf{\Sigma} =$$

$$\begin{pmatrix} \mathbb{V}[X_1] & \text{Cov}[X_1, X_2] & \cdots & \text{Cov}[X_1, X_D] \\ \text{Cov}[X_2, X_1] & \mathbb{V}[X_2] & \cdots & \text{Cov}[X_2, X_D] \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}[X_D, X_1] & \text{Cov}[X_D, X_2] & \cdots & \mathbb{V}[X_D] \end{pmatrix}$$

from which we can get the important result

$$\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T$$

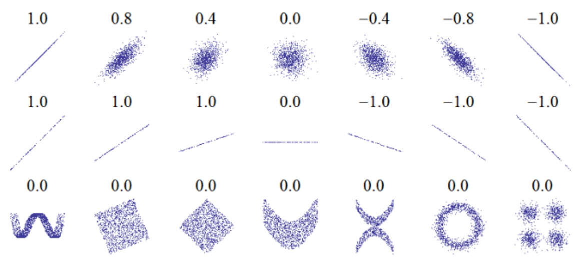
Another useful result is that the covariance of a linear transfomation

$$\text{Cov}[\mathbf{A}\mathbf{x} + b] = \mathbf{A}\text{Cov}[\mathbf{x}]\mathbf{A}^T$$

The **cross-covariance** between two random vector is define by

$$\text{Cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y} - \mathbb{E}[\mathbf{y}])^T]$$

3.1.2 Correlation



Covariance can be negative or positive infinity. Sometimes it is more convenient to work with a normalized measure, with a finite lower and upper bound. The **correltion coefficient** between X and Y is defined as

$$\rho \triangleq \text{Corr}[X, Y] \triangleq \frac{\text{Cov}[X, Y]}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}}$$

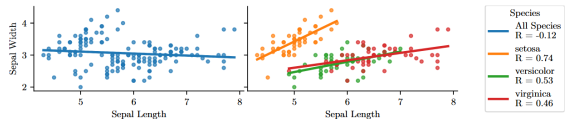
While the covariance can be any real value $\mathbb{R}$, correlation is always between $-1 \leq \rho \leq 1$. In the case of a vector $\mathbf{x}$ of related

$$\text{Corr}(\mathbf{x}) =$$

$$\begin{pmatrix} 1 & \frac{\mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)]}{\sigma_1\sigma_2} & \cdots & \frac{\mathbb{E}[(X_1-\mu_1)(X_D-\mu_D)]}{\sigma_1\sigma_D} \\ \frac{\mathbb{E}[(X_2-\mu_2)(X_1-\mu_1)]}{\sigma_2\sigma_1} & 1 & \cdots & \frac{\mathbb{E}[(X_2-\mu_2)(X_D-\mu_D)]}{\sigma_2\sigma_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathbb{E}[(X_D-\mu_D)(X_1-\mu_1)]}{\sigma_D\sigma_1} & \frac{\mathbb{E}[(X_D-\mu_D)(X_2-\mu_2)]}{\sigma_D\sigma_2} & \cdots & 1 \end{pmatrix}$$

But there is one fact that **Uncorrelated does nos imply independent**. Let $X \sim \text{Unif}(-1, -1)$ and $Y = X^2$. If X and Y are independent, meaning $p(X, Y) = p(X)p(Y)$, then $\text{Cov}[X, Y] = 0$ and $\text{Corr}[X, Y] = 0$ so it independent does not imply independent. But that not true, clearly Y depend on X . In general, there are several data sets where X and Y are clear dependence between X and Y, and yet the correlation coefficient is 0.

3.1.3 Simpson Paradox



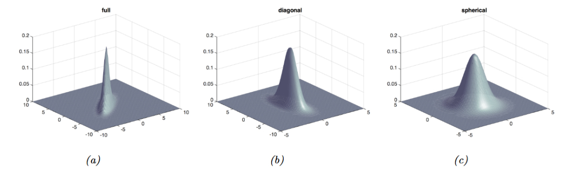
The Simpson say that a statistical trend or relationship that appears in several different group of data but can disappear if these group or combine.

3.2 The multivariate Gaussian (normal) distribution

The multivariate Gaussian (normal) distribution is a generalization of the univariate of the univariate gaussian to multiple dimensions.

$$\mathcal{N}(\mathbf{y}, \boldsymbol{\mu}, \mathbf{\Sigma}) \triangleq \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right)$$

where $\boldsymbol{\mu} = \mathbb{E}[\mathbf{y}] \in R^D$ is the mean vector, and $\mathbf{\Sigma} = \text{Cov}[\mathbf{y}]$ is the $D \times D$ **covariance matrix**.



In 2d, the MVN is know as the **Bivariate Gaussian** distribution. The pdf can be defined as $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$, where $\boldsymbol{\mu}, \mathbf{y} \in \mathbb{R}^2$ and

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

A diagonal covariance has D parameters, and 0s in off-diagonal. A sphere covariance, also called isotropic variance matrix is form by $\mathbf{\Sigma} = \sigma^2\mathbb{I}$.

3.2.1 Mahalanobis distance

The pdf of a **multivariate Gaussian** distribution

$$\log p(y \mid \boldsymbol{\mu}, \mathbf{\Sigma}) = -\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) + const$$

where $\mathbf{y}$ is a data point(vector), $\boldsymbol{\mu}$ is the mean vector, $\mathbf{\Sigma}$ is the covariance matrix. The quadratic term $(\mathbf{y} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})$ represents the **Mahalanobuis distance squared**.

$$\Delta^2 \triangleq (\mathbf{y} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})$$

A distance measure how far between a point and distribution, considering a correlation variables. It useful because it account for a shape of the data distribution. The notation is Δ between $\mathbf{y}$ and $\boldsymbol{\mu}$ is defined.
Thus contours of constant (log) probability are equivalent to contours of constant Mahalanobis distance.
To gain insight into the contours of constant Mahalanobis distance, we exploit the fact that Σ, and hence $\Lambda = \Sigma^{-1}$, are both positive definite matrices (by assumption). Consider the following eigendecomposition of Σ:

$$\Sigma = \sum_{d=1}^{D} \lambda_d u_d u_d^T$$

where u_d are the **eigenvectors** of $\mathbf{\Sigma}$, λ_d are the **eigenvalues** of $\mathbf{\Sigma}$. Since $\mathbf{\Sigma}$ is positive definite, we can similarly write the inverse

$$\Sigma^{-1} = \sum_{d=1}^{D} \frac{1}{\lambda_d} u_d u_d^T$$

This decomposition allows to **rotate and scale** the coordinate system. Define $z_d \triangleq u_d^T(\mathbf{y} - \boldsymbol{\mu})$, so $z = U(\mathbf{y} - \boldsymbol{\mu})$. Then we can write the Mahalanobis distance as follows

$$\begin{aligned}(\mathbf{y} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) &= (\mathbf{y} - \boldsymbol{\mu})^T (\sum_{d=1}^{D} \frac{1}{\lambda_d} u_d u_d^T)(\mathbf{y} - \boldsymbol{\mu}) \\ &= \sum_{d=1}^{D} \frac{1}{\lambda_d}(\mathbf{y} - \boldsymbol{\mu})^T u_d u_d^T(\mathbf{y} - \boldsymbol{\mu}) = \sum_{d=1}^{D} \frac{z_d^2}{\lambda_d}\end{aligned}$$

3.2.2 Marinals and conditionals of an MVN

Suppose $\mathbf{y} = (y_1, y_2)$ is jointly Gaussian with parameters

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \quad \mathbf{\Lambda} = \mathbf{\Sigma}^{-1} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}$$

where $\mathbf{\Lambda}$ is the **precision matrix**. Then the marginals are given by

$$p(\mathbf{y}_1) = \mathcal{N}(\mathbf{y}_1 \mid \boldsymbol{\mu}_1, \mathbf{\Sigma}_{11})$$
$$p(\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_2 \mid \boldsymbol{\mu}_2, \mathbf{\Sigma}_{22})$$

and the posterior conditional is given by

$$\begin{aligned} p(\mathbf{y}_1, \mathbf{y}_2) &= \mathcal{N}(\mathbf{y}_1 \mid \boldsymbol{\mu}_{1|2}, \mathbf{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \mathbf{\Sigma}_{11}\mathbf{\Sigma}_{22}^{-1}(y_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \Lambda_{11}^{-1}\Lambda_{12}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2) \\ &= \mathbf{\Sigma}_{1|2}(\Lambda_{11}\boldsymbol{\mu}_1 - \Lambda_{12}(\mathbf{y}_2 - \boldsymbol{\mu}_2)) \\ \mathbf{\Sigma}_{1|2} &= \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21} = \Lambda_{11}^{-1} \end{aligned}$$

Let us consider a 2d example. The covariance matrix is

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

The marginal $p(y_1)$ is a 1D Gaussian, obtained by projecting the joint distribution onto the y_1 line:

$$p(y_1) = \mathcal{N}(y_1 \mid \mu_1, \sigma_1^2)$$

Suppose we observe $Y_2 = y_2$; the conditional $p(y_1|y_2)$ is obtained by "slicing" the joint distribution through the $Y_2 = y_2$ line:

$$p(y_1|y_2) = \mathcal{N}\left(y_1 \middle| \mu_1 + \frac{\rho\sigma_1\sigma_2}{\sigma_2^2}(y_2 - \mu_2), \sigma_1^2 - \frac{(\rho\sigma_1\sigma_2)^2}{\sigma_2^2}\right)$$

If $\sigma_1 = \sigma_2 = \sigma$, we get

$$p(y_1|y_2) = \mathcal{N}\left(y_1 \middle| \mu_1 + \rho(y_2 - \mu_2), \sigma^2(1 - \rho^2)\right)$$

For example, suppose $\rho = 0.8$, $\sigma_1 = \sigma_2 = 1$, $\mu_1 = \mu_2 = 0$, and $y_2 = 1$. We see that $\mathbb{E}[y_1|y_2 = 1] = 0.8$, which makes sense, since $\rho = 0.8$ means that we believe that if y_2 increases by 1 (beyond its mean), then y_1 increases by 0.8. We also see

$$\mathrm{Var}(y_1|y_2 = 1) = 1 - 0.8^2 = 0.36.$$

This also makes sense: our uncertainty about y_1 has gone down, since we have learned something about y_1 (indirectly) by observing y_2. If $\rho = 0$, we get $p(y_1|y_2) = \mathcal{N}(y_1|\mu_1, \sigma_1^2)$, since y_2 conveys no information about y_1 if they are uncorrelated (and hence independent).

3.2.3 Example Missing data

Suppose we sampled $N = 10$ vectors from a $8D$ Gaussian, and then hid 50% of data. Then compute the missing given the observed entries and the true model parameters. More precisely, for each example of n in the data matrix, we compute $p(\mathbf{y}_{n,h}|\mathbf{y}_{n,v}, \boldsymbol{\theta})$, y is the data point, v are the indices of the visible features, and h are the remaining indices of the hidden features, and $\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{\Sigma})$. So we can compute the marginal distribution if the distribution of each missing variable $i \in \mathbf{h}, p(y_{n,i} \mid y_{n,\mathbf{v}}, \boldsymbol{\theta})$. From the marginal, we compute the posterior mean

$$\bar{y}_{n,i} = \mathbb{E}\left[y_{n,i} \mid \mathbf{y}_{n,v}, \boldsymbol{\theta}\right].$$

The posterior mean represents our "best guess" about the true value of that entry, in the sense that it minimizes our expected squared error. We can use $\mathbb{V}[y_{n,i} \mid y_{n,v}, \boldsymbol{\theta}]$ as a measure of the confidence in this guess. Because the prediction are the mean of the marginal distribution, if the variance is small, our prediction. The conditional distribution estimate all the possible value of the missing data. If the variance is small, the distribution will "close" to the mean, and actual value will have the higher chance to be same as the mean (prediction value), otherwise, if the variance is big, the distribution will "spread" and far a way a mean, and the actual value will have the lower chance to be same as the mean.

3.3 Linear Gaussian systems

We will extend this approach to handle the noise observation. Let $z \in \mathbb{R}^L$ be unknown data value, and $y \in \mathbb{R}^D$ be some noise measurement of zs, we assume this variables related to the joint distribution.

$$p(z) = \mathcal{N}(z \mid \boldsymbol{\mu}_z, \mathbf{\Sigma}_z)$$
$$p(\mathbf{y}|z) = \mathcal{N}(\mathbf{y} \mid \mathbf{W}\mathbf{x} + \mathbf{b}, \mathbf{\Sigma}_y)$$

where $\mathbf{W}$ is a matrix of size $D \times L$. This is an example of a **linear Gaussian system**.
The corresponding joint distribution, $p(z, \mathbf{y}) = p(z)p(\mathbf{y}|z)$, is the and $L + R$ dimensional Gaussian, with mean and covariance

$$\mu = \begin{pmatrix} \mu_z \\ \mathbf{W}\mu_z + \mathbf{b} \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \Sigma_z & \Sigma_z\mathbf{W}^T \\ \mathbf{W}\Sigma_z & \Sigma_y + \mathbf{W}\Sigma_z\mathbf{W}^T \end{pmatrix}$$

Bayes rule for Gaussians

$$\begin{aligned} p(z|\mathbf{y}) &= \mathcal{N}(z|\boldsymbol{\mu}_{z|\mathbf{y}}, \mathbf{\Sigma}_{z|\mathbf{y}}) \\ \mathbf{\Sigma}_{z|\mathbf{y}}^{-1} &= \mathbf{\Sigma}_z^{-1} + \mathbf{W}^T\mathbf{\Sigma}_z^{-1}\mathbf{W} \\ \boldsymbol{\mu}_{z|\mathbf{y}} &= \mathbf{\Sigma}_{z|\mathbf{y}}[\mathbf{W}^T\mathbf{\Sigma}_y^{-1}(\mathbf{y} - \mathbf{b}) + \mathbf{\Sigma}_z^{-1}\boldsymbol{\mu}_z] \end{aligned}$$

The normalization constant of the posterior is given by

$$\begin{aligned} p(\mathbf{y}) &= \int \mathcal{N}(z|\boldsymbol{\mu}_z, \mathbf{\Sigma}_z)\mathcal{N}(\mathbf{y}|\mathbf{W}z + \mathbf{b}, \mathbf{\Sigma}_y)dz \\ &= \mathcal{N}(\mathbf{y}|\mathbf{W}\boldsymbol{\mu}_z + b, \mathbf{\Sigma}_z + \mathbf{W}\mathbf{\Sigma}_z\mathbf{W}^T) \end{aligned}$$

Derivation
The log of the joint distribution

$$\log p(z, \mathbf{y}) = -\frac{1}{2}(z - \boldsymbol{\mu}_z)^T\mathbf{\Sigma}_z^{-1}(z - \boldsymbol{\mu}_z)$$
$$-\frac{1}{2}(\mathbf{y} - \mathbf{W}z - \mathbf{b})^T\mathbf{\Sigma}_y^{-1}(\mathbf{y} - \mathbf{W}z - \mathbf{b})$$

4 Linear Models

4.1 Linear Discriminant Analysis

Consider classification models of the following form

$$p(y = c \mid \mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x}|y = c, \boldsymbol{\theta})p(y = c|\boldsymbol{\theta})}{\sum_{c'} p(\mathbf{x}|y = c', \boldsymbol{\theta})p(y = c'|\boldsymbol{\theta})}$$

where $p(y = c|\boldsymbol{\theta})$ is the prior distribution, and $p(\mathbf{x}|y = c, \boldsymbol{\theta})$ is the class conditional density for class c.
The overall model is called a **generative classifier**, by generating features $\mathbf{x}$ which have the class c, define $p(\mathbf{x}|y = c, \boldsymbol{\theta})$. By contrast, a **discriminative classifier** directly model the class posterior $p(y = c|\mathbf{x}, \boldsymbol{\theta})$.
If we choose the class conditional densities in a special way, we will see that the posterior over classes is a linear function of $\mathbf{x}$, $\log p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \mathbf{W}\mathbf{x} + \mathbf{b}$.
Gaussian discriminant analysis
Consider generative classifierby a multivariate Gaussian

$$p(x|y = c, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \mathbf{\Sigma}_c)$$

The posterior has the form

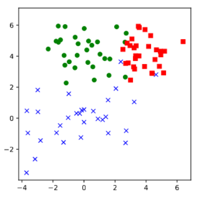
$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) \propto \pi_c\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \mathbf{\Sigma}_c)$$

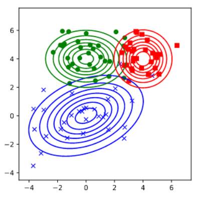
where $\pi_c = p(y = c|\boldsymbol{\theta})$ the prior probability of the label c,. It have a shorter name **GDA**.

The log posterior over class is given by

$$\log p(y = c \mid \mathbf{x}, \boldsymbol{\theta}) = \log \pi_c - \frac{1}{2}\log|2\pi\mathbf{\Sigma}_c|$$
$$-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c) + const$$

This is called **discriminant function**, decision between 2 classes, say c and c', will be a quadratic function x. Hence this is known as a **quadratic discriminant analysis(QDA)**.


(a)


(b)

For example, give the 3 classes, each of them fit the Gaussian distribuion (fig a), and plot it (fig b). The feaures in the blue class, there have som correlated, while in the green are independent, and whereas red class are independent and isotropic.

4.1.1 Linear boundaries decision

When we have a special case of Gaussian distribution, which each class have the same covariance $\mathbf{\Sigma}_c = \mathbf{\Sigma}$. We define

$$\begin{aligned} \log p(y = c|\mathbf{x}, \boldsymbol{\theta}) &= \log \pi_c - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top\mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) + \text{const} \\ &= \underbrace{\log \pi_c - \frac{1}{2}\boldsymbol{\mu}^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu}}_{\gamma_c} + \underbrace{\mathbf{x}^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu}}_{\beta_c} + \text{const} \underbrace{- \frac{1}{2}\mathbf{x}^\top\mathbf{\Sigma}^{-1}\mathbf{x}}_{\kappa} \\ &= \gamma_c + \mathbf{x}^\top\beta_c + \kappa \end{aligned}$$

The final term is independent of c. Although we use a quadratic discriminant analysis(QDA), we see that the discriminant function is a linear function of $\mathbf{x}$. We see that

$$\begin{aligned} &(\mathbf{x} - \mu_c)^\top\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu_c) \\ &= \mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x} - 2\mathbf{x}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_c + \boldsymbol{\mu}_c^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_c \end{aligned}$$

Where we compared 2 classes (c and c') $\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x}$ is same for both.
An interesting connection between LDA and logistic regression, we can write

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{\beta_c^T\mathbf{x} + \gamma_c}}{\sum_{c'} e^{\beta_{c'}^T\mathbf{x} + \gamma_{c'}}} = \frac{e^{\mathbf{w}^T[1,\mathbf{x}]}}{\sum_{c'} e^{\mathbf{w}^T[1,\mathbf{x}]}}$$

where $\mathbf{w}_c = [\gamma_c, \beta_c]$. We has the same form as multinomial logistic regression model. The key difference is that in LDA, we first fit the Gaussians (and class prior) to maximize the join likelihood $p(x, y \mid \boldsymbol{\theta})$ the join likelihood $p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$, and then we derive $\mathbf{w}$ from $\boldsymbol{\theta}$. By contrast, in logistic regression, we estimate $\mathbf{w}$ directly to maximize the conditional likelihood $p(y \mid \mathbf{x}, \mathbf{w})$. In general these can give different results.
Suppose we train the following binary classifier via maximum likelihood.

a. GaussI: A generative classifier, where the class-conditional densities are Gaussian, with both covariance matrices set to $\mathbb{I}$ (identity matrix), i.e., $p(x \mid y = c) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \mathbf{\Sigma}_c)$ We assume $p(y)$ is uniform

b. GaussX: as for GaussI, but the covariance matrices are unconstrained, i.e., $p(x \mid y = c) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \mathbf{\Sigma}_c)$

c. LinLog: A logistic regression model with linear features.

d. QuadLog: A logistic regression model, using linear and quadratic features(i.e., polynomial basis function expansion of degree 2) $(w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2)$

After training we compute the performance of each model M on the training set as follows

$$L(M) = \frac{1}{n}\sum_{i=1}^{n} \log p(y_i \mid \mathbf{x}_i, \hat{\boldsymbol{\theta}}, M)$$

(Note that this is the *conditional* log-likelihood $p(y|\mathbf{x}, \hat{\boldsymbol{\theta}})$ and not the joint log-likelihood $p(y, \mathbf{x}|\hat{\boldsymbol{\theta}})$.) We now want to compare the performance of each model. We will write $L(M) \leq L(M')$ if model M must have lower (or equal) log-likelihood (on the training set) than M', for any training set (in other words, M is worse than M', at least as far as training set log probability is concerned). For each of the following model pairs, state whether $L(M) \leq L(M')$, $L(M) \geq L(M')$, or whether no such statement can be made (i.e., M might sometimes be better than M' and sometimes worse); also, for each question, briefly (1-2 sentences) explain why.

Answer:

1. GaussI:
   - Parameters: for two classes in d-dimension, you estimate two mean vector (2d parameters) plus a class prior
   - The resulting decision boundary is a linear(because identical, spherical covariances lead to $\|x - \boldsymbol{\mu}_1\|^2 - \|x - \boldsymbol{\mu}_2\|^2 = const$
   - Weekness: It generative the features $\mathbf{x}$ by sphere Gaussians ($\boldsymbol{\Sigma} = \sigma^2\mathbf{I}_d$,$\mathbf{I}_d$ is the $d \times d$ matrix), so it is quite restrictive and tends not to fit training data as flexibly as purely disciminative models do.
2. LinLog
   - Parameters: a single weight vector $\mathbf{w} \in \mathbb{R}^D$ plus a bias b,
   - Decision boundary is also linear, but here the parameters are learned purely by maximizing the conditional likelihood $p(y \mid \mathbf{x}, \boldsymbol{\theta})$. Empirically this usually fits the training set better than the GaussI.
3. QuadLog
   - Parameters: includes all linear terms plus all pair product x_ix_j, be a quadratic matrix.
   - This is a significantly large parameter set than LinLog, and it can shape a more flexible (quadratic) decision boundary in x-shape. This will achieve a lower training-set negative log-likelihood than purely linear model
4. Gauss2
   - Prameters: for two classes, each has a mean vector(d parameters) plus a full covariance matrix(on the order of $\frac{d(d+1)}{2}$ independent parameters). All doubled because there is two classes, plus class priors.
   - The log-odds decisions boundary is the difference of two quadratic forms in x, giving even more freedom than a single quadratic function. As a result, on a giving training set, it tends to achieve the **lowest** negative log-likelihood, also overfit the easiest.

On the **training set**, the typical ordering (worst $\rightarrow$ best)

$$\text{GaussI} > \text{LinLog} > \text{QuadLog} > \text{Gauss2}$$