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 February 23, 2025

Introduction

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 $\mathcal{D} = \{(x_n, y_n)\}, n = 1 \rightarrow N$ is called **training set**. (N is called the **sample size**. The performance P depends on the type of output we want to predict.

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**.

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

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** $l(y, \hat{y})$.

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

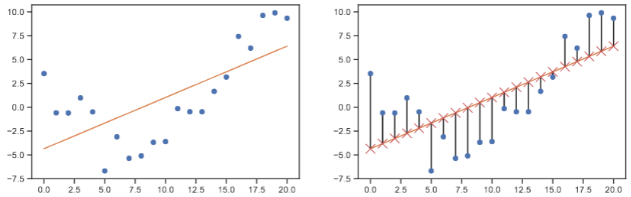
Regression

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**.

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



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

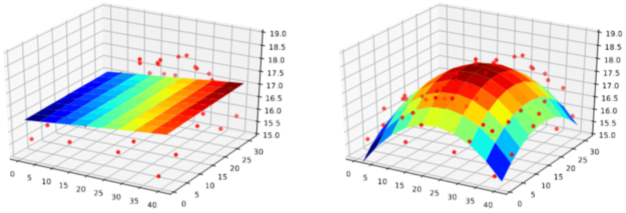
$$f(x; \boldsymbol{\theta}) = b + wx$$

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

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

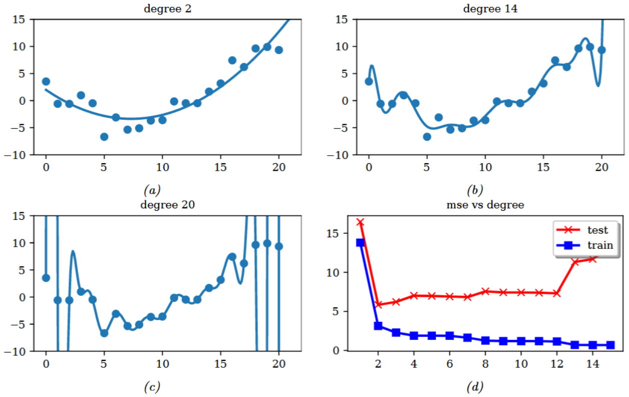
Overfitting

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**.

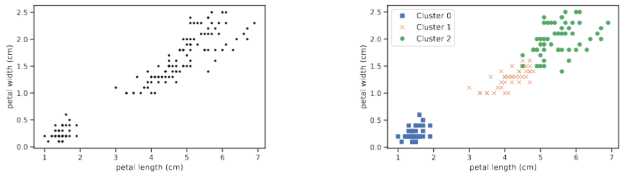
Unsupervised learning

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.

Clustering



(a)

(b)

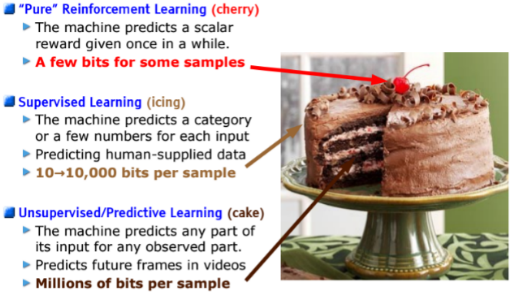
A simple example of unsupervised learning is the problem of finding clusters in data. The goal is to partition the input into regions that contain "similar" points.

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.

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

Preprocessing discrete input data

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].
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:

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

The new model becomes:

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

Express the equation we have:

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

Probability Unvariate Models

Introduction

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

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

If independent events

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

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 $\{X_1, ..., X_m\} \subseteq \{X_1, ..., X_n\}$

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

Disjoint

$$Pr(A \lor B) = Pr(A) + Pr(B) - Pr(A \land B)$$

Conditional probability

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

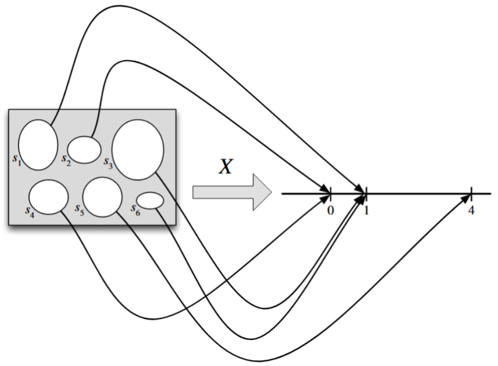
If events A and B are conditionally independent given event C

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

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

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

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 $\mathbb{R}$.
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 $\sum_{x \in \mathcal{X}} p(x) = 1$
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.
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

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

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

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

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.
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

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

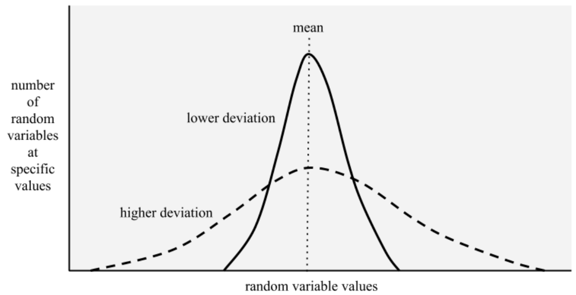
$$\mathbb{E}[X] = \mathbb{E}_Y[\mathbb{E}[X|Y]]$$

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.

Bayes' Rule

Bayes' Rule, and with extra conditioning (just add in C!)

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

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

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*.

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

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