Super VIP Cheatsheet: Machine Learning

Afshine Amidi and Shervine Amidi

August 6, 2018

Contents

1 Supervised Learning

1.1 Introduction to Supervised Learning

Given a set of data points $\{x^{(1)}, ..., x^{(m)}\}$ associated to a set of outcomes $\{y^{(1)}, ..., y^{(m)}\}$, we want to build a classifier that learns how to predict y from x.

❒ **Type of prediction** – The different types of predictive models are summed up in the table below:

Model	Outcome	Examples
Regression	Continuous	Linear regression
Classifier	Class	Logistic regression, SVM, Naive Bayes

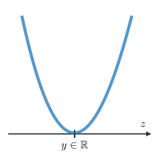
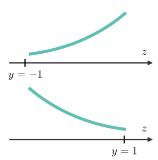
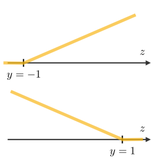
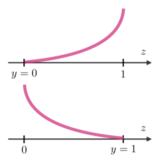
❒ **Type of model** – The different models are summed up in the table below:

Model	Goal	Examples
Discriminative	Estimate $P(y\|x)$	Regressions, SVMs
Generative	Estimate $P(x\|y)$ Used to deduce $P(y\|x)$	GDA, Naive Bayes

1.2 Notations and general concepts

❒ **Hypothesis** – The hypothesis is noted h_θ and is the model that we choose. For a given input data $x^{(i)}$, the model prediction output is $h_\theta(x^{(i)})$.

❒ **Loss function** – A loss function is a function $L : (z,y) \in \mathbb{R} \times Y \longmapsto L(z,y) \in \mathbb{R}$ that takes as inputs the predicted value y corresponding to the real data value y and outputs how different they are. The common loss functions are summed up in the table below:

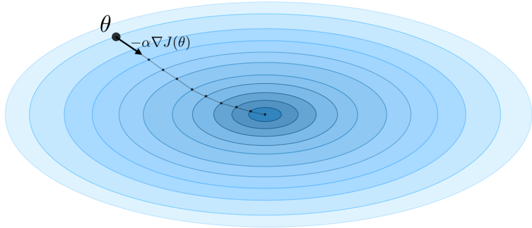
Least squared	Logistic	Hinge	Cross-entropy
$\frac{1}{2}(y-z)^2$	$\log(1+\exp(-yz))$	$\max(0,1-yz)$	$-\Big[y\log(z)+(1-y)\log(1-z)\Big]$
			
Linear regression	Logistic regression	SVM	Neural Network

❒ **Cost function** – The cost function J is commonly used to assess the performance of a model, and is defined with the loss function L as follows:

$$\boxed{J(\theta)=\sum_{i=1}^m L(h_\theta(x^{(i)}), y^{(i)})}$$

❒ **Gradient descent** – By noting $\alpha \in \mathbb{R}$ the learning rate, the update rule for gradient descent is expressed with the learning rate and the cost function J as follows:

$$\boxed{\theta \longleftarrow \theta - \alpha\nabla J(\theta)}$$



Remark: Stochastic gradient descent (SGD) is updating the parameter based on each training example, and batch gradient descent is on a batch of training examples.

❒ **Likelihood** – The likelihood of a model $L(\theta)$ given parameters θ is used to find the optimal parameters θ through maximizing the likelihood. In practice, we use the log-likelihood $\ell(\theta) = \log(L(\theta))$ which is easier to optimize. We have:

$$\boxed{\theta^{\textrm{opt}}=\underset{\theta}{\textrm{arg max }}L(\theta)}$$

❒ **Newton's algorithm** – The Newton's algorithm is a numerical method that finds θ such that $\ell'(\theta) = 0$. Its update rule is as follows:

$$\boxed{\theta\leftarrow\theta-\frac{\ell'(\theta)}{\ell''(\theta)}}$$

Remark: the multidimensional generalization, also known as the Newton-Raphson method, has the following update rule:

$$\theta\leftarrow\theta-\left(\nabla_\theta^2\ell(\theta)\right)^{-1}\nabla_\theta\ell(\theta)$$

1.3 Linear models

1.3.1 Linear regression

We assume here that $y|x;\theta\sim\mathcal{N}(\mu,\sigma^2)$

❒ **Normal equations** – By noting X the matrix design, the value of θ that minimizes the cost function is a closed-form solution such that:

$$\boxed{\theta=(X^TX)^{-1}X^Ty}$$

❒ **LMS algorithm** – By noting α the learning rate, the update rule of the Least Mean Squares (LMS) algorithm for a training set of m data points, which is also known as the Widrow-Hoff learning rule, is as follows:

$$\boxed{\forall j,\quad \theta_j \leftarrow \theta_j+\alpha\sum_{i=1}^m\left[y^{(i)}-h_\theta(x^{(i)})\right]x_j^{(i)}}$$

Remark: the update rule is a particular case of the gradient ascent.

❒ **LWR** – Locally Weighted Regression, also known as LWR, is a variant of linear regression that weights each training example in its cost function by $w^{(i)}(x)$, which is defined with parameter $\tau \in \mathbb{R}$ as:

$$\boxed{w^{(i)}(x) = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)}$$

1.3.2 Classification and logistic regression

❒ **Sigmoid function** – The sigmoid function g, also known as the logistic function, is defined as follows:

$$\forall z \in \mathbb{R}, \quad \boxed{g(z) = \frac{1}{1 + e^{-z}} \in]0,1[}$$

❒ **Logistic regression** – We assume here that $y|x;\theta \sim \text{Bernoulli}(\phi)$. We have the following form:

$$\boxed{\phi = p(y = 1|x;\theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)}$$

Remark: there is no closed form solution for the case of logistic regressions.

❒ **Softmax regression** – A softmax regression, also called a multiclass logistic regression, is used to generalize logistic regression when there are more than 2 outcome classes. By convention, we set $\theta_K = 0$, which makes the Bernoulli parameter ϕ_i of each class i equal to:

$$\boxed{\phi_i = \frac{\exp(\theta_i^T x)}{\displaystyle\sum_{j=1}^{K} \exp(\theta_j^T x)}}$$

1.3.3 Generalized Linear Models

❒ **Exponential family** – A class of distributions is said to be in the exponential family if it can be written in terms of a natural parameter, also called the canonical parameter or link function, η, a sufficient statistic $T(y)$ and a log-partition function $a(\eta)$ as follows:

$$\boxed{p(y;\eta) = b(y)\exp(\eta T(y) - a(\eta))}$$

Remark: we will often have $T(y) = y$. Also, $\exp(-a(\eta))$ can be seen as a normalization parameter that will make sure that the probabilities sum to one.

Here are the most common exponential distributions summed up in the following table:

Distribution	η	$T(y)$	$a(\eta)$	$b(y)$
Bernoulli	$\log\left(\frac{\phi}{1-\phi}\right)$	y	$\log(1 + \exp(\eta))$	1
Gaussian	μ	y	$\frac{\eta^2}{2}$	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2}\right)$
Poisson	$\log(\lambda)$	y	e^η	$\frac{1}{y!}$
Geometric	$\log(1-\phi)$	y	$\log\left(\frac{e^\eta}{1-e^\eta}\right)$	1

❒ **Assumptions of GLMs** – Generalized Linear Models (GLM) aim at predicting a random variable y as a function fo $x \in \mathbb{R}^{n+1}$ and rely on the following 3 assumptions:

$$(1) \quad \boxed{y|x;\theta \sim \text{ExpFamily}(\eta)} \qquad (2) \quad \boxed{h_\theta(x) = E[y|x;\theta]} \qquad (3) \quad \boxed{\eta = \theta^T x}$$

Remark: ordinary least squares and logistic regression are special cases of generalized linear models.

1.4 Support Vector Machines

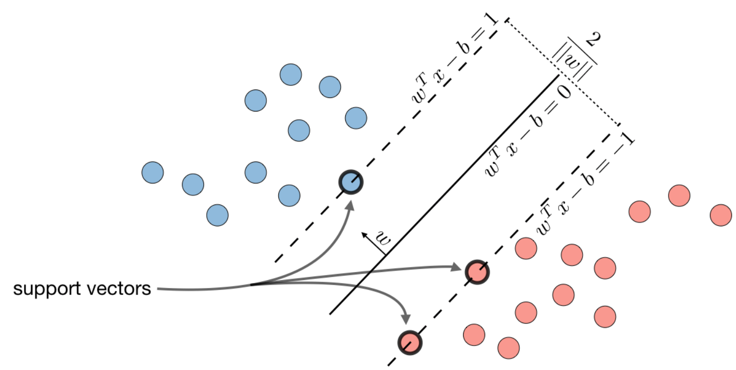
The goal of support vector machines is to find the line that maximizes the minimum distance to the line.

❒ **Optimal margin classifier** – The optimal margin classifier h is such that:

$$\boxed{h(x) = \text{sign}(w^T x - b)}$$

where $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ is the solution of the following optimization problem:

$$\boxed{\min \frac{1}{2}||w||^2} \qquad \text{such that} \qquad \boxed{y^{(i)}(w^T x^{(i)} - b) \geqslant 1}$$



Remark: the line is defined as $\boxed{w^T x - b = 0}$.

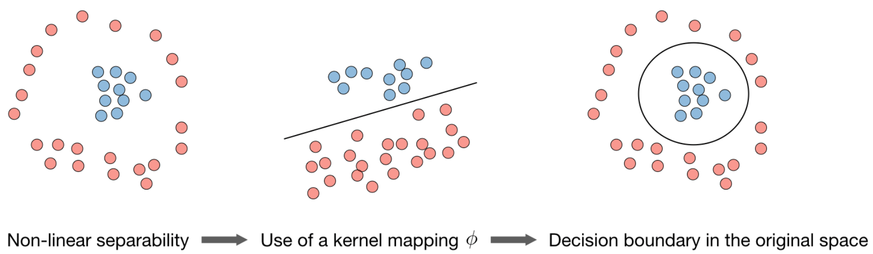
❒ **Hinge loss** – The hinge loss is used in the setting of SVMs and is defined as follows:

$$\boxed{L(z,y) = [1 - yz]_+ = \max(0,1 - yz)}$$

❒ **Kernel** – Given a feature mapping ϕ, we define the kernel K to be defined as:

$$\boxed{K(x,z)=\phi(x)^T\phi(z)}$$

In practice, the kernel K defined by $K(x,z)=\exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$ is called the Gaussian kernel and is commonly used.



Remark: we say that we use the "kernel trick" to compute the cost function using the kernel because we actually don't need to know the explicit mapping ϕ, which is often very complicated. Instead, only the values $K(x,z)$ are needed.

❒ **Lagrangian** – We define the Lagrangian $\mathcal{L}(w,b)$ as follows:

$$\boxed{\mathcal{L}(w,b)=f(w)+\sum_{i=1}^l\beta_ih_i(w)}$$

Remark: the coefficients β_i are called the Lagrange multipliers.

1.5 Generative Learning

A generative model first tries to learn how the data is generated by estimating $P(x|y)$, which we can then use to estimate $P(y|x)$ by using Bayes' rule.

1.5.1 Gaussian Discriminant Analysis

❒ **Setting** – The Gaussian Discriminant Analysis assumes that y and $x|y=0$ and $x|y=1$ are such that:

$$\boxed{y\sim\textrm{Bernoulli}(\phi)}$$

$$\boxed{x|y=0\sim\mathcal{N}(\mu_0,\Sigma)}\quad\textrm{and}\quad\boxed{x|y=1\sim\mathcal{N}(\mu_1,\Sigma)}$$

❒ **Estimation** – The following table sums up the estimates that we find when maximizing the likelihood:

$\widehat{\phi}$	$\widehat{\mu_j}\quad(j=0,1)$	$\widehat{\Sigma}$
$\displaystyle\frac{1}{m}\sum_{i=1}^m1_{\{y^{(i)}=1\}}$	$\displaystyle\frac{\sum_{i=1}^m1_{\{y^{(i)}=j\}}x^{(i)}}{\sum_{i=1}^m1_{\{y^{(i)}=j\}}}$	$\displaystyle\frac{1}{m}\sum_{i=1}^m(x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T$

1.5.2 Naive Bayes

❒ **Assumption** – The Naive Bayes model supposes that the features of each data point are all independent:

$$\boxed{P(x|y)=P(x_1,x_2,...|y)=P(x_1|y)P(x_2|y)...=\prod_{i=1}^nP(x_i|y)}$$

❒ **Solutions** – Maximizing the log-likelihood gives the following solutions, with $k\in\{0,1\},l\in[\![1,L]\!]$

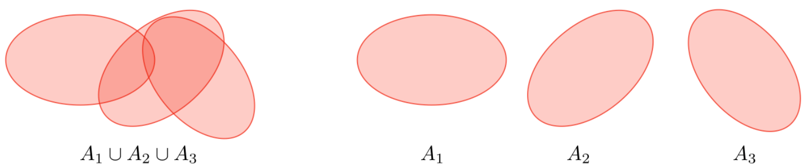
$$\boxed{P(y=k)=\frac{1}{m}\times\#\{j|y^{(j)}=k\}}\quad\textrm{and}\quad\boxed{P(x_i=l|y=k)=\frac{\#\{j|y^{(j)}=k\textrm{ and }x_i^{(j)}=l\}}{\#\{j|y^{(j)}=k\}}}$$

Remark: Naive Bayes is widely used for text classification.

1.6 Learning Theory

❒ **Union bound** – Let $A_1,...,A_k$ be k events. We have:

$$\boxed{P(A_1\cup...\cup A_k)\leqslant P(A_1)+...+P(A_k)}$$



❒ **Hoeffding inequality** – Let $Z_1,..,Z_m$ be m iid variables drawn from a Bernoulli distribution of parameter ϕ. Let $\widehat{\phi}$ be their sample mean and $\gamma>0$ fixed. We have:

$$\boxed{P(|\phi-\widehat{\phi}|>\gamma)\leqslant2\exp(-2\gamma^2m)}$$

Remark: this inequality is also known as the Chernoff bound.

❒ **Training error** – For a given classifier h, we define the training error $\widehat{\epsilon}(h)$, also known as the empirical risk or empirical error, to be as follows:

$$\boxed{\widehat{\epsilon}(h)=\frac{1}{m}\sum_{i=1}^m1_{\{h(x^{(i)})\neq y^{(i)}\}}}$$

❒ **Probably Approximately Correct (PAC)** – PAC is a framework under which numerous results on learning theory were proved, and has the following set of assumptions:

- the training and testing sets follow the same distribution
- the training examples are drawn independently

❒ **Shattering** – Given a set $S=\{x^{(1)},...,x^{(d)}\}$, and a set of classifiers $\mathcal{H}$, we say that $\mathcal{H}$ shatters S if for any set of labels $\{y^{(1)},...,y^{(d)}\}$, we have:

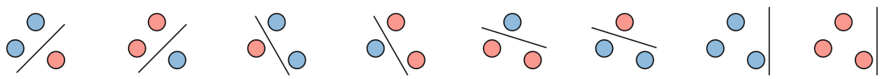
$$\boxed{\exists h\in\mathcal{H},\quad \forall i\in[\![1,d]\!],\quad h(x^{(i)})=y^{(i)}}$$

❒ **Upper bound theorem** – Let $\mathcal{H}$ be a finite hypothesis class such that $|\mathcal{H}|=k$ and let δ and the sample size m be fixed. Then, with probability of at least $1-\delta$, we have:

$$\boxed{\epsilon(\widehat{h})\leqslant\left(\min_{h\in\mathcal{H}}\epsilon(h)\right)+2\sqrt{\frac{1}{2m}\log\left(\frac{2k}{\delta}\right)}}$$

❒ **VC dimension** – The Vapnik-Chervonenkis (VC) dimension of a given infinite hypothesis class $\mathcal{H}$, noted $\textrm{VC}(\mathcal{H})$ is the size of the largest set that is shattered by $\mathcal{H}$.

Remark: the VC dimension of $\mathcal{H}=\{\textrm{set of linear classifiers in 2 dimensions}\}$ is 3.



❒ **Theorem (Vapnik)** – Let $\mathcal{H}$ be given, with $\textrm{VC}(\mathcal{H})=d$ and m the number of training examples. With probability at least $1-\delta$, we have:

$$\boxed{\epsilon(\widehat{h})\leqslant\left(\min_{h\in\mathcal{H}}\epsilon(h)\right)+O\left(\sqrt{\frac{d}{m}\log\left(\frac{m}{d}\right)+\frac{1}{m}\log\left(\frac{1}{\delta}\right)}\right)}$$

2 Unsupervised Learning

2.1 Introduction to Unsupervised Learning

❒ **Motivation** – The goal of unsupervised learning is to find hidden patterns in unlabeled data $\{x^{(1)},...,x^{(m)}\}$.

❒ **Jensen's inequality** – Let f be a convex function and X a random variable. We have the following inequality:

$$\boxed{E[f(X)]\geqslant f(E[X])}$$

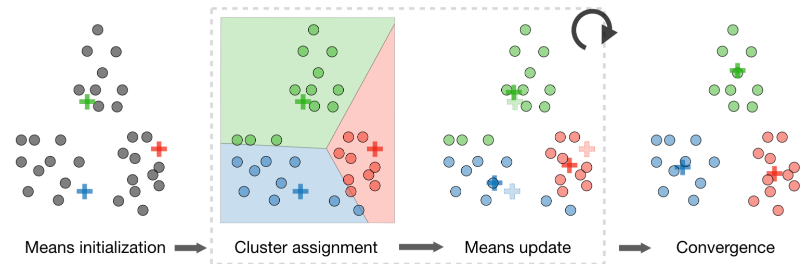
2.2 Clustering

2.2.1 k-means clustering

We note $c^{(i)}$ the cluster of data point i and μ_j the center of cluster j.

❒ **Algorithm** – After randomly initializing the cluster centroids $\mu_1,\mu_2,...,\mu_k\in\mathbb{R}^n$, the k-means algorithm repeats the following step until convergence:

$$\boxed{c^{(i)}=\underset{j}{\textrm{arg min}}||x^{(i)}-\mu_j||^2}\quad\textrm{and}\quad\boxed{\mu_j=\frac{\displaystyle\sum_{i=1}^m1_{\{c^{(i)}=j\}}x^{(i)}}{\displaystyle\sum_{i=1}^m1_{\{c^{(i)}=j\}}}}$$



❒ **Distortion function** – In order to see if the algorithm converges, we look at the distortion function defined as follows:

$$\boxed{J(c,\mu)=\sum_{i=1}^m||x^{(i)}-\mu_{c^{(i)}}||^2}$$

2.2.2 Expectation-Maximization

❒ **Latent variables** – Latent variables are hidden/unobserved variables that make estimation problems difficult, and are often denoted z. Here are the most common settings where there are latent variables:

Setting	Latent variable z	$x\|z$	Comments
Mixture of k Gaussians	Multinomial(ϕ)	$\mathcal{N}(\mu_j,\Sigma_j)$	$\mu_j\in\mathbb{R}^n,\phi\in\mathbb{R}^k$
Factor analysis	$\mathcal{N}(0,I)$	$\mathcal{N}(\mu+\Lambda z,\psi)$	$\mu_j\in\mathbb{R}^n$

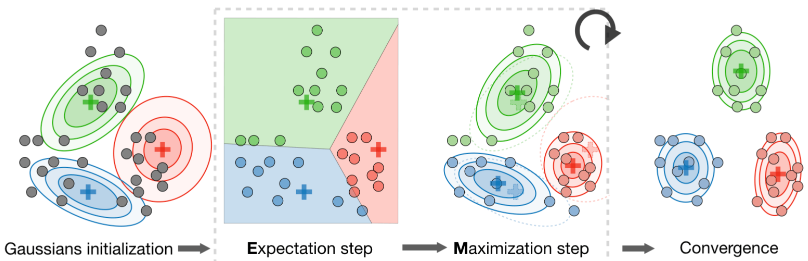
❒ **Algorithm** – The Expectation-Maximization (EM) algorithm gives an efficient method at estimating the parameter θ through maximum likelihood estimation by repeatedly constructing a lower-bound on the likelihood (E-step) and optimizing that lower bound (M-step) as follows:

- E-step: Evaluate the posterior probability $Q_i(z^{(i)})$ that each data point $x^{(i)}$ came from a particular cluster $z^{(i)}$ as follows:

$$\boxed{Q_i(z^{(i)})=P(z^{(i)}|x^{(i)};\theta)}$$

- M-step: Use the posterior probabilities $Q_i(z^{(i)})$ as cluster specific weights on data points $x^{(i)}$ to separately re-estimate each cluster model as follows:

$$\boxed{\theta_i=\underset{\theta}{\textrm{argmax }}\sum_i\int_{z^{(i)}}Q_i(z^{(i)})\log\left(\frac{P(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}\right)dz^{(i)}}$$



2.3 Dimension reduction

2.3.1 Principal component analysis

It is a dimension reduction technique that finds the variance maximizing directions onto which to project the data.

❒ **Eigenvalue, eigenvector** – Given a matrix $A\in\mathbb{R}^{n\times n}$, λ is said to be an eigenvalue of A if there exists a vector $z\in\mathbb{R}^n\backslash\{0\}$, called eigenvector, such that we have:

$$\boxed{Az=\lambda z}$$

❒ **Spectral theorem** – Let $A\in\mathbb{R}^{n\times n}$. If A is symmetric, then A is diagonalizable by a real orthogonal matrix $U\in\mathbb{R}^{n\times n}$. By noting $\Lambda=\textrm{diag}(\lambda_1,...,\lambda_n)$, we have:

$$\boxed{\exists\Lambda\textrm{ diagonal},\quad A=U\Lambda U^T}$$

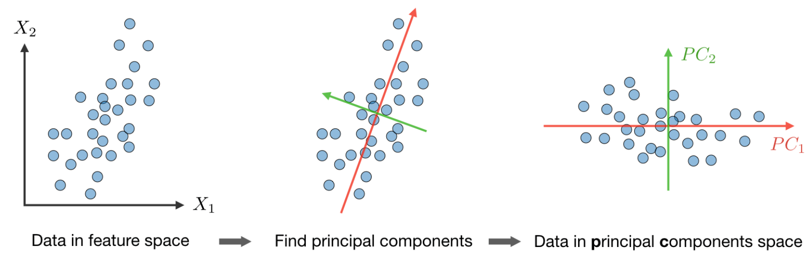
Remark: the eigenvector associated with the largest eigenvalue is called principal eigenvector of matrix A.

❒ **Algorithm** – The Principal Component Analysis (PCA) procedure is a dimension reduction technique that projects the data on k dimensions by maximizing the variance of the data as follows:

- Step 1: Normalize the data to have a mean of 0 and standard deviation of 1.

$$\boxed{x_j^{(i)}\leftarrow\frac{x_j^{(i)}-\mu_j}{\sigma_j}}\quad\textrm{where}\quad\boxed{\mu_j=\frac{1}{m}\sum_{i=1}^mx_j^{(i)}}\quad\textrm{and}\quad\boxed{\sigma_j^2=\frac{1}{m}\sum_{i=1}^m(x_j^{(i)}-\mu_j)^2}$$

- Step 2: Compute $\Sigma=\frac{1}{m}\sum_{i=1}^mx^{(i)}{x^{(i)}}^T\in\mathbb{R}^{n\times n}$, which is symmetric with real eigenvalues.
- Step 3: Compute $u_1,...,u_k\in\mathbb{R}^n$ the k orthogonal principal eigenvectors of Σ, i.e. the orthogonal eigenvectors of the k largest eigenvalues.
- Step 4: Project the data on $\textrm{span}_\mathbb{R}(u_1,...,u_k)$. This procedure maximizes the variance among all k-dimensional spaces.



2.3.2 Independent component analysis

It is a technique meant to find the underlying generating sources.

❒ **Assumptions** – We assume that our data x has been generated by the n-dimensional source vector $s=(s_1,...,s_n)$, where s_i are independent random variables, via a mixing and non-singular matrix A as follows:

$$\boxed{x=As}$$

The goal is to find the unmixing matrix $W=A^{-1}$ by an update rule.

❒ **Bell and Sejnowski ICA algorithm** – This algorithm finds the unmixing matrix W by following the steps below:

- Write the probability of $x=As=W^{-1}s$ as:

$$p(x)=\prod_{i=1}^np_s(w_i^Tx)\cdot|W|$$

- Write the log likelihood given our training data $\{x^{(i)}, i \in [\![1,m]\!]\}$ and by noting g the sigmoid function as:

$$l(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \log \left(g'(w_j^T x^{(i)}) \right) + \log |W| \right)$$

Therefore, the stochastic gradient ascent learning rule is such that for each training example $x^{(i)}$, we update W as follows:

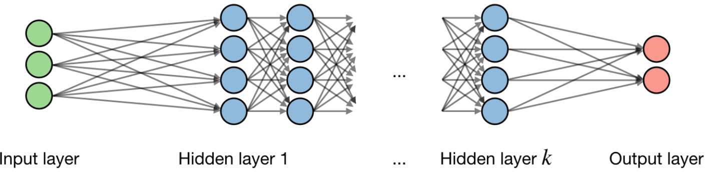
$$\boxed{W \longleftarrow W + \alpha \left(\begin{pmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{pmatrix} {x^{(i)}}^T + (W^T)^{-1} \right)}$$

3 Deep Learning

3.1 Neural Networks

Neural networks are a class of models that are build with layers. Commonly used types of neural networks include convolutional and recurrent neural networks.

❒ **Architecture** – The vocabulary around neural networks architectures is described in the figure below:

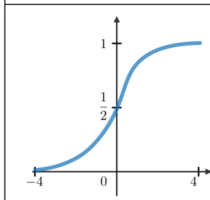
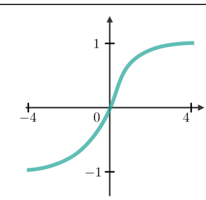
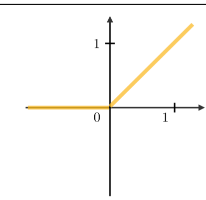
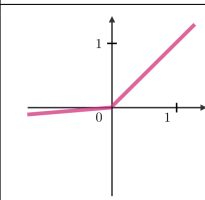


By noting i the i^{th} layer of the network and j the j^{th} hidden unit of the layer, we have:

$$\boxed{z_j^{[i]} = {w_j^{[i]}}^T x + b_j^{[i]}}$$

where we note w, b, z the weight, bias and output respectively.

❒ **Activation function** – Activation functions are used at the end of a hidden unit to introduce non-linear complexities to the model. Here are the most common ones:

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \dfrac{1}{1+e^{-z}}$	$g(z) = \dfrac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0,z)$	$g(z) = \max(\epsilon z,z)$ with $\epsilon \ll 1$
			

❒ **Cross-entropy loss** – In the context of neural networks, the cross-entropy loss $L(z,y)$ is commonly used and is defined as follows:

$$\boxed{L(z,y) = -\Big[y \log(z) + (1-y)\log(1-z)\Big]}$$

❒ **Learning rate** – The learning rate, often noted η, indicates at which pace the weights get updated. This can be fixed or adaptively changed. The current most popular method is called Adam, which is a method that adapts the learning rate.

❒ **Backpropagation** – Backpropagation is a method to update the weights in the neural network by taking into account the actual output and the desired output. The derivative with respect to weight w is computed using chain rule and is of the following form:

$$\boxed{\frac{\partial L(z,y)}{\partial w}=\frac{\partial L(z,y)}{\partial a}\times\frac{\partial a}{\partial z}\times\frac{\partial z}{\partial w}}$$

As a result, the weight is updated as follows:

$$\boxed{w\longleftarrow w-\eta\frac{\partial L(z,y)}{\partial w}}$$

❒ **Dropout** – Dropout is a technique meant at preventing overfitting the training data by dropping out units in a neural network. In practice, neurons are either dropped with probability p or kept with probability $1-p$.

3.2 Convolutional Neural Networks

❒ **Convolutional layer** – By noting W the input volume size, F the size of the convolutional layer neurons, P the amount of zero padding, then the number of neurons N that fit in a given volume is such that:

$$\boxed{N=\frac{W-F+2P}{S}+1}$$

3.3 Recurrent Neural Networks

❒ **LSTM** – A long short-term memory (LSTM) network is a type of RNN model that avoids the vanishing gradient problem by adding 'forget' gates.

3.4 Reinforcement Learning and Control

The goal of reinforcement learning is for an agent to learn how to evolve in an environment.

❒ **Markov decision processes** – A Markov decision process (MDP) is a 5-tuple $(S,A,\{P_{sa}\},\gamma,R)$ where:

- $\mathcal{S}$ is the set of states
- $\mathcal{A}$ is the set of actions
- $\{P_{sa}\}$ are the state transition probabilities for $s\in\mathcal{S}$ and $a\in\mathcal{A}$
- $\gamma\in[0,1[$ is the discount factor
- $R:\mathcal{S}\times\mathcal{A}\longrightarrow\mathbb{R}$ or $R:\mathcal{S}\longrightarrow\mathbb{R}$ is the reward function that the algorithm wants to maximize

❒ **Policy** – A policy π is a function $\pi:\mathcal{S}\longrightarrow\mathcal{A}$ that maps states to actions.
Remark: we say that we execute a given policy π if given a state a we take the action $a=\pi(s)$.

❒ **Value function** – For a given policy π and a given state s, we define the value function V^π as follows:

$$\boxed{V^\pi(s)=E\Big[R(s_0)+\gamma R(s_1)+\gamma^2R(s_2)+...|s_0=s,\pi\Big]}$$

❒ **Bellman equation** – The optimal Bellman equations characterizes the value function V^{π^*} of the optimal policy π^*:

$$\boxed{V^{\pi^*}(s)=R(s)+\max_{a\in\mathcal{A}}\gamma\sum_{s'\in S}P_{sa}(s')V^{\pi^*}(s')}$$

Remark: we note that the optimal policy π^* for a given state s is such that:

$$\boxed{\pi^*(s)=\underset{a\in\mathcal{A}}{\textrm{argmax}}\sum_{s'\in\mathcal{S}}P_{sa}(s')V^*(s')}$$

❒ **Value iteration algorithm** – The value iteration algorithm is in two steps:

- We initialize the value:

$$\boxed{V_0(s)=0}$$

- We iterate the value based on the values before:

$$\boxed{V_{i+1}(s)=R(s)+\max_{a\in\mathcal{A}}\left[\sum_{s'\in\mathcal{S}}\gamma P_{sa}(s')V_i(s')\right]}$$

❒ **Maximum likelihood estimate** – The maximum likelihood estimates for the state transition probabilities are as follows:

$$\boxed{P_{sa}(s')=\frac{\#\textrm{times took action }a\textrm{ in state }s\textrm{ and got to }s'}{\#\textrm{times took action }a\textrm{ in state }s}}$$

❒ **Q-learning** – Q-learning is a model-free estimation of Q, which is done as follows:

$$\boxed{Q(s,a)\leftarrow Q(s,a)+\alpha\Big[R(s,a,s')+\gamma\max_{a'}Q(s',a')-Q(s,a)\Big]}$$

4 Machine Learning Tips and Tricks

4.1 Metrics

In a context of a binary classification, here are the main metrics that are important to track to assess the performance of the model.

❒ **Confusion matrix** – By noting True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), the confusion matrix is defined as follows:

	Predicted class +	**Predicted** class –
Actual class +	TP True Positives	FN False Negatives
Actual class –	FP False Positives	TN True Negatives

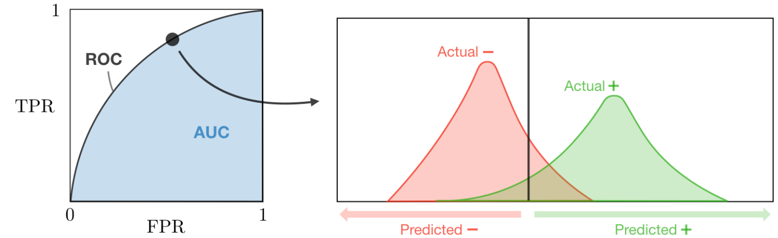
❒ **Main metrics** – Here are the main metrics to track:

Metric	Formula	Interpretation
Accuracy	$\frac{\textrm{TP}+\textrm{TN}}{\textrm{TP}+\textrm{TN}+\textrm{FP}+\textrm{FN}}$	Overall performance of model
Precision	$\frac{\textrm{TP}}{\textrm{TP}+\textrm{FP}}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{\textrm{TP}}{\textrm{TP}+\textrm{FN}}$	Coverage of actual positive sample
Specificity	$\frac{\textrm{TN}}{\textrm{TN}+\textrm{FP}}$	Coverage of actual negative sample
F1 score	$\frac{\textrm{2TP}}{\textrm{2TP}+\textrm{FP}+\textrm{FN}}$	Hybrid metric useful for unbalanced classes

❒ **ROC** – The receiver operating curve, also noted ROC, is the plot of TPR versus FPR by varying the threshold. These metrics are are summed up in the table below:

Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{\textrm{TP}}{\textrm{TP}+\textrm{FN}}$	Recall, sensitivity
False Positive Rate FPR	$\frac{\textrm{FP}}{\textrm{TN}+\textrm{FP}}$	1-specificity

❒ **AUC** – The area under the receiving operating curve, also noted AUC or AUROC, is the area below the ROC as shown in the following figure:



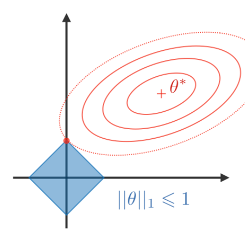
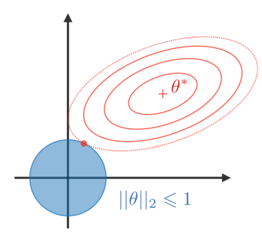
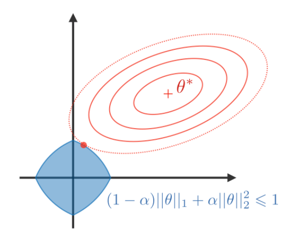
4.2 Model selection

❒ **Vocabulary** – When selecting a model, we distinguish 3 different parts of the dataset as follows:

- Training set: this is the part of the dataset on which the model is trained.
- Validation set: also called hold-out, development set or dev set, it is used to assess the performance of the previously trained model.
- Test set: once the model has been chosen, it is trained on the training+validation set and tested on the unseen test set.

❒ **Cross-validation** – Cross-validation is a model selection technique aimed at assessing which model would generalize well. While several types of cross-validation exist, the most common one is the k-fold cross-validation where the dataset is split into k parts and where the model is used as follows:

❒ **Regularization** – The regularization procedure aims at avoiding the model to overfit the data and thus deals with high variance issues. The following table sums up the different types of commonly used regularization techniques:

LASSO	Ridge	Elastic Net
- Shrinks coefficients to 0 - Good for variable selection	Makes coefficients smaller	Tradeoff between variable selection and small coefficients
		
$...+\lambda\lvert\lvert\theta\rvert\rvert_1$ $\lambda\in\mathbb{R}$	$...+\lambda\lvert\lvert\theta\rvert\rvert_2^2$ $\lambda\in\mathbb{R}$	$...+\lambda\Big[(1-\alpha)\lvert\lvert\theta\rvert\rvert_1+\alpha\lvert\lvert\theta\rvert\rvert_2^2\Big]$ $\lambda\in\mathbb{R},\quad\alpha\in[0,1]$

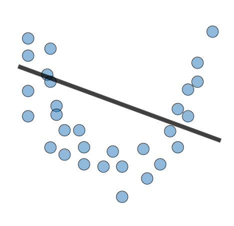
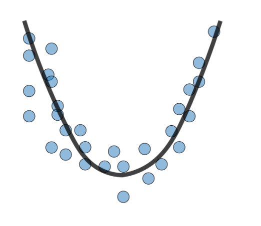
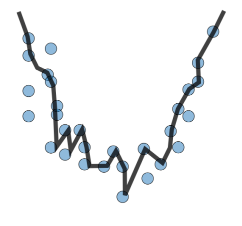
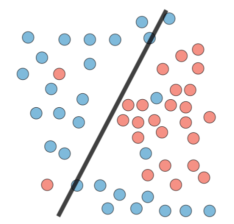
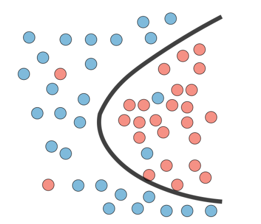
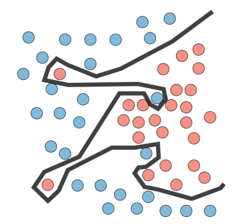
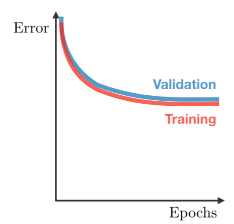
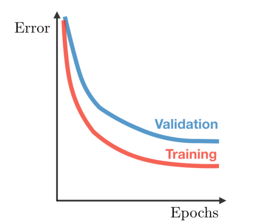
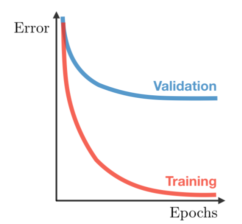
❒ **Model selection** – Train model on training set, then evaluate on the development set, then pick best performance model on the development set, and retrain all of that model on the whole training set.

4.3 Diagnostics

❒ **Bias** – The bias of a model is the difference between the expected prediction and the correct model that we try to predict for given data points.

❒ **Variance** – The variance of a model is the variability of the model prediction for given data points.

❒ **Bias/variance tradeoff** – The simpler the model, the higher the bias, and the more complex the model, the higher the variance.

	Underfitting	**Just right**	**Overfitting**
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Deep learning			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data

❒ **Error analysis** – Error analysis is analyzing the root cause of the difference in performance between the current and the perfect models.

❒ **Ablative analysis** – Ablative analysis is analyzing the root cause of the difference in performance between the current and the baseline models.

5 Refreshers

5.1 Probabilities and Statistics

5.1.1 Introduction to Probability and Combinatorics

❒ **Sample space** – The set of all possible outcomes of an experiment is known as the sample space of the experiment and is denoted by S.

❒ **Event** – Any subset E of the sample space is known as an event. That is, an event is a set consisting of possible outcomes of the experiment. If the outcome of the experiment is contained in E, then we say that E has occurred.

❒ **Axioms of probability** – For each event E, we denote $P(E)$ as the probability of event E occuring. By noting $E_1,...,E_n$ mutually exclusive events, we have the 3 following axioms:

$$(1)\quad \boxed{0\leqslant P(E)\leqslant 1}\quad (2)\quad \boxed{P(S)=1}\quad (3)\quad \boxed{P\left(\bigcup_{i=1}^n E_i\right)=\sum_{i=1}^n P(E_i)}$$

❒ **Permutation** – A permutation is an arrangement of r objects from a pool of n objects, in a given order. The number of such arrangements is given by $P(n, r)$, defined as:

$$\boxed{P(n,r)=\frac{n!}{(n-r)!}}$$

❒ **Combination** – A combination is an arrangement of r objects from a pool of n objects, where the order does not matter. The number of such arrangements is given by $C(n, r)$, defined as:

$$\boxed{C(n,r)=\frac{P(n,r)}{r!}=\frac{n!}{r!(n-r)!}}$$

Remark: we note that for $0\leqslant r\leqslant n$, we have $P(n,r)\geqslant C(n,r)$.

5.1.2 Conditional Probability

❒ **Bayes' rule** – For events A and B such that $P(B) > 0$, we have:

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

Remark: we have $P(A\cap B)=P(A)P(B|A)=P(A|B)P(B)$.

❒ **Partition** – Let $\{A_i, i\in[\![1,n]\!]\}$ be such that for all i, $A_i\neq\varnothing$. We say that $\{A_i\}$ is a partition if we have:

$$\boxed{\forall i\neq j, A_i\cap A_j=\emptyset\quad\textrm{and}\quad\bigcup_{i=1}^n A_i=S}$$

Remark: for any event B in the sample space, we have $P(B)=\displaystyle\sum_{i=1}^n P(B|A_i)P(A_i)$.

❒ **Extended form of Bayes' rule** – Let $\{A_i, i\in[\![1,n]\!]\}$ be a partition of the sample space. We have:

$$\boxed{P(A_k|B)=\frac{P(B|A_k)P(A_k)}{\displaystyle\sum_{i=1}^nP(B|A_i)P(A_i)}}$$

❒ **Independence** – Two events A and B are independent if and only if we have:

$$\boxed{P(A\cap B)=P(A)P(B)}$$

5.1.3 Random Variables

❒ **Random variable** – A random variable, often noted X, is a function that maps every element in a sample space to a real line.

❒ **Cumulative distribution function (CDF)** – The cumulative distribution function F, which is monotonically non-decreasing and is such that $\underset{x\rightarrow-\infty}{\lim}F(x)=0$ and $\underset{x\rightarrow+\infty}{\lim}F(x)=1$, is defined as:

$$\boxed{F(x)=P(X\leqslant x)}$$

Remark: we have $P(a<X\leqslant B)=F(b)-F(a)$.

❒ **Probability density function (PDF)** – The probability density function f is the probability that X takes on values between two adjacent realizations of the random variable.

❒ **Relationships involving the PDF and CDF** – Here are the important properties to know in the discrete (D) and the continuous (C) cases.

Case	**CDF F**	**PDF f**	**Properties of PDF**
(D)	$F(x)=\displaystyle\sum_{x_i\leqslant x}P(X=x_i)$	$f(x_j)=P(X=x_j)$	$0\leqslant f(x_j)\leqslant1$ and $\displaystyle\sum_jf(x_j)=1$
(C)	$F(x)=\displaystyle\int_{-\infty}^xf(y)dy$	$f(x)=\displaystyle\frac{dF}{dx}$	$f(x)\geqslant0$ and $\displaystyle\int_{-\infty}^{+\infty}f(x)dx=1$

❒ **Variance** – The variance of a random variable, often noted Var(X) or σ^2, is a measure of the spread of its distribution function. It is determined as follows:

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

❒ **Standard deviation** – The standard deviation of a random variable, often noted σ, is a measure of the spread of its distribution function which is compatible with the units of the actual random variable. It is determined as follows:

$$\boxed{\sigma=\sqrt{\textrm{Var}(X)}}$$

❒ **Expectation and Moments of the Distribution** – Here are the expressions of the expected value $E[X]$, generalized expected value $E[g(X)]$, k^{th} moment $E[X^k]$ and characteristic function $\psi(\omega)$ for the discrete and continuous cases:

Case	$E[X]$	$E[g(X)]$	$E[X^k]$	$\psi(\omega)$
(D)	$\displaystyle\sum_{i=1}^nx_if(x_i)$	$\displaystyle\sum_{i=1}^ng(x_i)f(x_i)$	$\displaystyle\sum_{i=1}^nx_i^kf(x_i)$	$\displaystyle\sum_{i=1}^nf(x_i)e^{i\omega x_i}$
(C)	$\displaystyle\int_{-\infty}^{+\infty}xf(x)dx$	$\displaystyle\int_{-\infty}^{+\infty}g(x)f(x)dx$	$\displaystyle\int_{-\infty}^{+\infty}x^kf(x)dx$	$\displaystyle\int_{-\infty}^{+\infty}f(x)e^{i\omega x}dx$

Remark: we have $e^{i\omega x}=\cos(\omega x)+i\sin(\omega x)$.

❒ **Revisiting the k^{th} moment** – The k^{th} moment can also be computed with the characteristic function as follows:

$$\boxed{E[X^k]=\frac{1}{i^k}\left[\frac{\partial^k\psi}{\partial\omega^k}\right]_{\omega=0}}$$

❒ **Transformation of random variables** – Let the variables X and Y be linked by some function. By noting f_X and f_Y the distribution function of X and Y respectively, we have:

$$\boxed{f_Y(y)=f_X(x)\left|\frac{dx}{dy}\right|}$$

❒ **Leibniz integral rule** – Let g be a function of x and potentially c, and a,b boundaries that may depend on c. We have:

$$\boxed{\frac{\partial}{\partial c}\left(\int_a^bg(x)dx\right)=\frac{\partial b}{\partial c}\cdot g(b)-\frac{\partial a}{\partial c}\cdot g(a)+\int_a^b\frac{\partial g}{\partial c}(x)dx}$$

❒ **Chebyshev's inequality** – Let X be a random variable with expected value μ and standard deviation σ. For $k,\sigma>0$, we have the following inequality:

$$\boxed{P(|X-\mu|\geqslant k\sigma)\leqslant\frac{1}{k^2}}$$

5.1.4 Jointly Distributed Random Variables

❒ **Conditional density** – The conditional density of X with respect to Y, often noted $f_{X|Y}$, is defined as follows:

$$\boxed{f_{X|Y}(x)=\frac{f_{XY}(x,y)}{f_Y(y)}}$$

❒ **Independence** – Two random variables X and Y are said to be independent if we have:

$$\boxed{f_{XY}(x,y)=f_X(x)f_Y(y)}$$

❒ **Marginal density and cumulative distribution** – From the joint density probability function f_{XY}, we have:

Case	Marginal density	Cumulative function
(D)	$f_X(x_i)=\displaystyle\sum_j f_{XY}(x_i,y_j)$	$F_{XY}(x,y)=\displaystyle\sum_{x_i\leqslant x}\sum_{y_j\leqslant y} f_{XY}(x_i,y_j)$
(C)	$f_X(x)=\displaystyle\int_{-\infty}^{+\infty} f_{XY}(x,y)dy$	$F_{XY}(x,y)=\displaystyle\int_{-\infty}^{x}\int_{-\infty}^{y} f_{XY}(x',y')dx'dy'$

❒ **Distribution of a sum of independent random variables** – Let $Y=X_1+...+X_n$ with $X_1,...,X_n$ independent. We have:

$$\boxed{\psi_Y(\omega)=\prod_{k=1}^n \psi_{X_k}(\omega)}$$

❒ **Covariance** – We define the covariance of two random variables X and Y, that we note σ_{XY}^2 or more commonly $\textrm{Cov}(X,Y)$, as follows:

$$\boxed{\textrm{Cov}(X,Y)\triangleq\sigma_{XY}^2=E[(X-\mu_X)(Y-\mu_Y)]=E[XY]-\mu_X\mu_Y}$$

❒ **Correlation** – By noting σ_X,σ_Y the standard deviations of X and Y, we define the correlation between the random variables X and Y, noted ρ_{XY}, as follows:

$$\boxed{\rho_{XY}=\frac{\sigma_{XY}^2}{\sigma_X\sigma_Y}}$$

Remarks: For any X,Y, we have $\rho_{XY}\in[-1,1]$. If X and Y are independent, then $\rho_{XY}=0$.

❒ **Main distributions** – Here are the main distributions to have in mind:

Type	Distribution	PDF	$\psi(\omega)$	$E[X]$	$\textrm{Var}(X)$
(D)	$X\sim\mathcal{B}(n,p)$ Binomial	$P(X=x)=\displaystyle\binom{n}{x}p^xq^{n-x}$ $x\in[\![0,n]\!]$	$(pe^{i\omega}+q)^n$	np	npq
	$X\sim\textrm{Po}(\mu)$ Poisson	$P(X=x)=\dfrac{\mu^x}{x!}e^{-\mu}$ $x\in\mathbb{N}$	$e^{\mu(e^{i\omega}-1)}$	μ	μ
(C)	$X\sim\mathcal{U}(a,b)$ Uniform	$f(x)=\dfrac{1}{b-a}$ $x\in[a,b]$	$\dfrac{e^{i\omega b}-e^{i\omega a}}{(b-a)i\omega}$	$\dfrac{a+b}{2}$	$\dfrac{(b-a)^2}{12}$
	$X\sim\mathcal{N}(\mu,\sigma)$ Gaussian	$f(x)=\dfrac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $x\in\mathbb{R}$	$e^{i\omega\mu-\frac{1}{2}\omega^2\sigma^2}$	μ	σ^2
	$X\sim\textrm{Exp}(\lambda)$ Exponential	$f(x)=\lambda e^{-\lambda x}$ $x\in\mathbb{R}_+$	$\dfrac{1}{1-\frac{i\omega}{\lambda}}$	$\dfrac{1}{\lambda}$	$\dfrac{1}{\lambda^2}$

5.1.5 Parameter estimation

❒ **Random sample** – A random sample is a collection of n random variables $X_1,...,X_n$ that are independent and identically distributed with X.

❒ **Estimator** – An estimator $\hat{\theta}$ is a function of the data that is used to infer the value of an unknown parameter θ in a statistical model.

❒ **Bias** – The bias of an estimator $\hat{\theta}$ is defined as being the difference between the expected value of the distribution of $\hat{\theta}$ and the true value, i.e.:

$$\boxed{\textrm{Bias}(\hat{\theta})=E[\hat{\theta}]-\theta}$$

Remark: an estimator is said to be unbiased when we have $E[\hat{\theta}]=\theta$.

❒ **Sample mean and variance** – The sample mean and the sample variance of a random sample are used to estimate the true mean μ and the true variance σ^2 of a distribution, are noted $\overline{X}$ and s^2 respectively, and are such that:

$$\boxed{\overline{X}=\frac{1}{n}\sum_{i=1}^n X_i}\quad\textrm{and}\quad\boxed{s^2=\hat{\sigma}^2=\frac{1}{n-1}\sum_{i=1}^n(X_i-\overline{X})^2}$$

❒ **Central Limit Theorem** – Let us have a random sample $X_1,...,X_n$ following a given distribution with mean μ and variance σ^2, then we have:

$$\boxed{\overline{X}\underset{n\rightarrow+\infty}{\sim}\mathcal{N}\left(\mu,\frac{\sigma}{\sqrt{n}}\right)}$$

5.2 Linear Algebra and Calculus

5.2.1 General notations

❒ **Vector** – We note $x\in\mathbb{R}^n$ a vector with n entries, where $x_i\in\mathbb{R}$ is the i^{th} entry:

$$x=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}\in\mathbb{R}^n$$

❒ **Matrix** – We note $A\in\mathbb{R}^{m\times n}$ a matrix with n rows and m, where $A_{i,j}\in\mathbb{R}$ is the entry located in the i^{th} row and j^{th} column:

$$A=\begin{pmatrix}A_{1,1}&\cdots&A_{1,n}\\\vdots&&\vdots\\A_{m,1}&\cdots&A_{m,n}\end{pmatrix}\in\mathbb{R}^{m\times n}$$

Remark: the vector x defined above can be viewed as a $n\times 1$ matrix and is more particularly called a column-vector.

❒ **Identity matrix** – The identity matrix $I\in\mathbb{R}^{n\times n}$ is a square matrix with ones in its diagonal and zero everywhere else:

$$I=\begin{pmatrix}1&0&\cdots&0\\0&\ddots&\ddots&\vdots\\\vdots&\ddots&\ddots&0\\0&\cdots&0&1\end{pmatrix}$$

Remark: for all matrices $A\in\mathbb{R}^{n\times n}$, we have $A\times I=I\times A=A$.

❒ **Diagonal matrix** – A diagonal matrix $D\in\mathbb{R}^{n\times n}$ is a square matrix with nonzero values in its diagonal and zero everywhere else:

$$D=\begin{pmatrix}d_1 & 0 & \cdots & 0\\ 0 & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & d_n\end{pmatrix}$$

Remark: we also note D as diag(d_1,...,d_n).

5.2.2 Matrix operations

❒ **Vector-vector multiplication** – There are two types of vector-vector products:

- inner product: for $x,y\in\mathbb{R}^n$, we have:

$$\boxed{x^Ty=\sum_{i=1}^n x_iy_i\in\mathbb{R}}$$

- outer product: for $x\in\mathbb{R}^m,y\in\mathbb{R}^n$, we have:

$$\boxed{xy^T=\begin{pmatrix}x_1y_1 & \cdots & x_1y_n\\ \vdots & & \vdots\\ x_my_1 & \cdots & x_my_n\end{pmatrix}\in\mathbb{R}^{m\times n}}$$

❒ **Matrix-vector multiplication** – The product of matrix $A\in\mathbb{R}^{m\times n}$ and vector $x\in\mathbb{R}^n$ is a vector of size $\mathbb{R}^n$, such that:

$$\boxed{Ax=\begin{pmatrix}a_{r,1}^Tx\\ \vdots\\ a_{r,m}^Tx\end{pmatrix}=\sum_{i=1}^n a_{c,i}x_i\in\mathbb{R}^n}$$

where $a_{r,i}^T$ are the vector rows and $a_{c,j}$ are the vector columns of A, and x_i are the entries of x.

❒ **Matrix-matrix multiplication** – The product of matrices $A\in\mathbb{R}^{m\times n}$ and $B\in\mathbb{R}^{n\times p}$ is a matrix of size $\mathbb{R}^{n\times p}$, such that:

$$\boxed{AB=\begin{pmatrix}a_{r,1}^Tb_{c,1} & \cdots & a_{r,1}^Tb_{c,p}\\ \vdots & & \vdots\\ a_{r,m}^Tb_{c,1} & \cdots & a_{r,m}^Tb_{c,p}\end{pmatrix}=\sum_{i=1}^n a_{c,i}b_{r,i}^T\in\mathbb{R}^{n\times p}}$$

where $a_{r,i}^T,b_{r,i}^T$ are the vector rows and $a_{c,j},b_{c,j}$ are the vector columns of A and B respectively.

❒ **Transpose** – The transpose of a matrix $A\in\mathbb{R}^{m\times n}$, noted A^T, is such that its entries are flipped:

$$\boxed{\forall i,j,\qquad A_{i,j}^T=A_{j,i}}$$

Remark: for matrices A,B, we have $(AB)^T=B^TA^T$.

❒ **Inverse** – The inverse of an invertible square matrix A is noted A^{-1} and is the only matrix such that:

$$\boxed{AA^{-1}=A^{-1}A=I}$$

Remark: not all square matrices are invertible. Also, for matrices A,B, we have $(AB)^{-1}=B^{-1}A^{-1}$

❒ **Trace** – The trace of a square matrix A, noted tr(A), is the sum of its diagonal entries:

$$\boxed{\textrm{tr}(A)=\sum_{i=1}^n A_{i,i}}$$

Remark: for matrices A,B, we have $tr(A^T)=tr(A)$ and $tr(AB)=tr(BA)$

❒ **Determinant** – The determinant of a square matrix $A\in\mathbb{R}^{n\times n}$, noted $|A|$ or det(A) is expressed recursively in terms of $A_{\backslash i,\backslash j}$, which is the matrix A without its i^{th} row and j^{th} column, as follows:

$$\boxed{\det(A)=|A|=\sum_{j=1}^n(-1)^{i+j}A_{i,j}|A_{\backslash i,\backslash j}|}$$

Remark: A is invertible if and only if $|A|\neq 0$. Also, $|AB|=|A||B|$ and $|A^T|=|A|$.

5.2.3 Matrix properties

❒ **Symmetric decomposition** – A given matrix A can be expressed in terms of its symmetric and antisymmetric parts as follows:

$$\boxed{A=\underbrace{\frac{A+A^T}{2}}_{\textrm{Symmetric}}+\underbrace{\frac{A-A^T}{2}}_{\textrm{Antisymmetric}}}$$

❒ **Norm** – A norm is a function $N:V\longrightarrow[0,+\infty[$ where V is a vector space, and such that for all $x,y\in V$, we have:

- $N(x+y)\leqslant N(x)+N(y)$
- $N(ax)=|a|N(x)$ for a scalar
- if $N(x)=0$, then $x=0$

For $x\in V$, the most commonly used norms are summed up in the table below:

Norm	Notation	Definition	Use case
Manhattan, L^1	$\|\|x\|\|_1$	$\displaystyle\sum_{i=1}^n \|x_i\|$	LASSO regularization
Euclidean, L^2	$\|\|x\|\|_2$	$\displaystyle\sqrt{\sum_{i=1}^n x_i^2}$	Ridge regularization
p-norm, L^p	$\|\|x\|\|_p$	$\displaystyle\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}}$	Hölder inequality
Infinity, L^∞	$\|\|x\|\|_\infty$	$\underset{i}{\max}\ \|x_i\|$	Uniform convergence

❒ **Linearly dependence** – A set of vectors is said to be linearly dependent if one of the vectors in the set can be defined as a linear combination of the others.
Remark: if no vector can be written this way, then the vectors are said to be linearly independent.

❒ **Matrix rank** – The rank of a given matrix A is noted rank(A) and is the dimension of the vector space generated by its columns. This is equivalent to the maximum number of linearly independent columns of A.

❒ **Positive semi-definite matrix** – A matrix $A \in \mathbb{R}^{n\times n}$ is positive semi-definite (PSD) and is noted $A \succeq 0$ if we have:

$$\boxed{A = A^T}\quad\textrm{and}\quad\boxed{\forall x\in\mathbb{R}^n,\quad x^TAx\geqslant 0}$$

Remark: similarly, a matrix A is said to be positive definite, and is noted $A \succ 0$, if it is a PSD matrix which satisfies for all non-zero vector x, $x^TAx > 0$.

❒ **Eigenvalue, eigenvector** – Given a matrix $A \in \mathbb{R}^{n\times n}$, λ is said to be an eigenvalue of A if there exists a vector $z \in \mathbb{R}^n\backslash\{0\}$, called eigenvector, such that we have:

$$\boxed{Az=\lambda z}$$

❒ **Spectral theorem** – Let $A \in \mathbb{R}^{n\times n}$. If A is symmetric, then A is diagonalizable by a real orthogonal matrix $U \in \mathbb{R}^{n\times n}$. By noting $\Lambda = \textrm{diag}(\lambda_1,...,\lambda_n)$, we have:

$$\boxed{\exists\Lambda\textrm{ diagonal},\quad A=U\Lambda U^T}$$

❒ **Singular-value decomposition** – For a given matrix A of dimensions $m \times n$, the singular-value decomposition (SVD) is a factorization technique that guarantees the existence of U $m\times m$ unitary, Σ $m \times n$ diagonal and V $n \times n$ unitary matrices, such that:

$$\boxed{A=U\Sigma V^T}$$

5.2.4 Matrix calculus

❒ **Gradient** – Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a function and $A \in \mathbb{R}^{m\times n}$ be a matrix. The gradient of f with respect to A is a $m \times n$ matrix, noted $\nabla_A f(A)$, such that:

$$\boxed{\Big(\nabla_A f(A)\Big)_{i,j}=\frac{\partial f(A)}{\partial A_{i,j}}}$$

Remark: the gradient of f is only defined when f is a function that returns a scalar.

❒ **Hessian** – Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function and $x \in \mathbb{R}^n$ be a matrix. The hessian of f with respect to x is a $n \times n$ symmetric matrix, noted $\nabla^2_x f(x)$, such that:

$$\boxed{\Big(\nabla_x^2 f(x)\Big)_{i,j}=\frac{\partial^2 f(x)}{\partial x_i\partial x_j}}$$

Remark: the hessian of f is only defined when f is a function that returns a scalar.

❒ **Gradient operations** – For matrices A,B,C, the following gradient properties are worth having in mind:

$$\boxed{\nabla_A\textrm{tr}(AB)=B^T}\quad\boxed{\nabla_{A^T}f(A)=\left(\nabla_Af(A)\right)^T}$$

$$\boxed{\nabla_A\textrm{tr}(ABA^TC)=CAB+C^TAB^T}\quad\boxed{\nabla_A|A|=|A|(A^{-1})^T}$$