Super VIP Cheatsheet: Aprendizaje Automático

Afshine Amidi y Shervine Amidi

6 de octubre de 2018

Índice

Traducido por Fernando Diaz, Juan P. Chavat, Erick Gabriel Mendoza Flores, Fernando González-Herrera, Mariano Ramirez, Alonso Melgar López, Gustavo Velasco-Hernández, David Jiménez Paredes, Fermin Ordaz, Jaime Noel Alvarez Luna y Juan Manuel Nava Zamudio.

1. Aprendizaje Supervisado

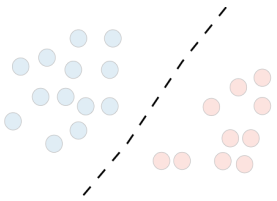
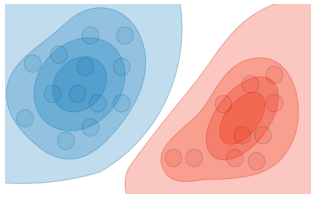
1.1. Introducción al aprendizaje supervisado

Dado un conjunto de puntos $\{x^{(1)},...,x^{(m)}\}$ asociado a un conjunto de etiquetas $\{y^{(1)},...,y^{(m)}\}$, queremos construir un clasificador que aprenda cómo predecir y dado x.

❒ **Tipo de predicción** – Los diferentes tipos de modelos de predicción se resumen en la siguiente tabla:

	Regresión	**Clasificador**
Etiqueta	Continuo	Clase
Ejemplos	Regresión lineal	Regresión logística, SVM, Naive Bayes

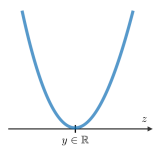
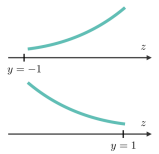
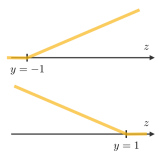
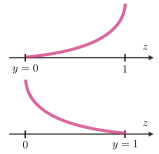
❒ **Tipo de modelo** – Los diferentes tipos de modelos se resumen en la siguiente tabla:

	Modelo discriminatorio	**Modelo generativo**
Objetivo	Estima directamente $P(y\|x)$	Estima $P(x\|y)$ para deducir $P(y\|x)$
Qué se aprende	Límite de decisión	Distribución de los datos
Ilustración		
Ejemplos	Regresiones, SVMs	GDA, Naive Bayes

1.2. Notación y conceptos generales

❒ **Hipótesis** – La hipótesis se representa con h_θ y es el modelo que elegimos. Para un dato de entrada $x^{(i)}$, la predicción dada por el modelo se representa como $h_\theta(x^{(i)})$.

❒ **Función de pérdida** – Una función de pérdida es una función $L : (z,y) \in \mathbb{R} \times Y \longmapsto L(z,y) \in \mathbb{R}$ que toma como entrada el valor predicho z y el valor real esperado y, dando como resultado qué tan diferentes son ambos. Las funciones de pérdida más comunes se detallan en la siguiente tabla:

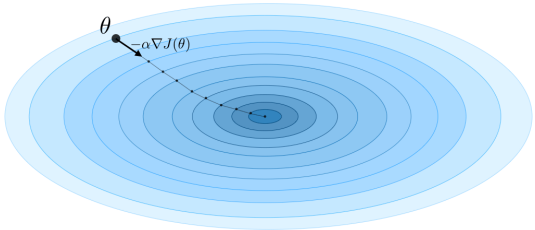
Error cuadrático	**Logística**	**Bisagra**	**Entropía cruzada**
$\frac{1}{2}(y-z)^2$	$\log(1+\exp(-yz))$	máx$(0,1-yz)$	$-\left[y\log(z)+(1-y)\log(1-z)\right]$
			
Regresión lineal	Regresión logística	SVM	Red neuronal

❒ **Función de costo** – La función de costo J es comúnmente utilizada para evaluar el rendimiento de un modelo y se define utilizando la función de pérdida L de la siguiente forma:

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

❒ **Descenso de gradiente** – Siendo $\alpha \in \mathbb{R}$ la tasa de aprendizaje, la regla de actualización de descenso en gradiente se expresa junto a la tasa de aprendizaje y la función de costo J de la siguiente manera:

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



Observación: El descenso en gradiente estocástico (en inglés, Stochastic Gradient Descent) actualiza el parámetro basándose en cada ejemplo de entenamiento, mientras que el descenso por lotes realiza la actualización del parámetro basándose en un conjunto (un lote) de ejemplos de entrenamiento.

❒ **Verosimilitud** – La verosimilitud de un modelo $L(\theta)$ dados los parámetros θ es utilizada para hallar los valores óptimos de θ a través de la verosimilitud. En la práctica se utiliza la log-verosimilitud $\ell(\theta) = \log(L(\theta))$ la cual es fácil de optimizar. Tenemos:

$$\boxed{\theta^{\text{opt}} = \arg\max_{\theta} L(\theta)}$$

❒ **Algoritmo de Newton** – El algoritmo de Newtow es un método numérico para hallar θ tal que $\ell'(\theta) = 0$. Su regla de actualización es:

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

Observación: la generalización multidimensional, también conocida como método de Newton-Raphson, tiene la siguiente regla de actualización:

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

1.3. Modelos lineales

1.3.1. Regresión lineal

Asumimos que $y|x;\theta \sim \mathcal{N}(\mu,\sigma^2)$

❒ **Ecuaciones normales** – Sea X la matriz de diseño, el valor de θ que minimiza la función de costo es una solución en forma cerrada tal que:

$$\boxed{\theta = (X^T X)^{-1} X^T y}$$

❒ **Algoritmo LMS** – Sea α la tasa de aprendizaje, la regla de actualización del algoritmo LMS (en inglés, *Least Mean Squares*) para el entrenamiento de m puntos, conocida también como tasa de aprendizaje de Widrow-Hoff, se define como:

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

Observación: la regla de actualización es un caso particular del ascenso de gradiente.

❒ **LWR** – Regresión local ponderada, LWR (en inglés, *Locally Weighted Regression*) es una variante de la regresión lineal que pondera cada ejemplo de entrenamiento en su función de costo utilizando $w^{(i)}(x)$, la cual se define con el parámetro $\tau \in \mathbb{R}$ as:

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

1.3.2. Clasificación y regresión logística

❒ **Función sigmoide** – La función sigmoide g, también conocida como la función logística, se define de la siguiente forma:

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

❒ **Regresión logística** – Asumiendo que $y|x;\theta \sim \text{Bernoulli}(\phi)$, tenemos la siguiente forma:

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

Observación: no existe solución en forma cerrada para los casos de regresiones logísticas.

❒ **Regresión softmax** – La regresión softmax, también llamada regresión logística multiclase, es utilizada para generalizar regresiones logísticas cuando hay más de dos clases resultantes. Por convención, se define $\theta_K = 0$, lo que hace al parámetro de Bernoulli ϕ_i de cada clase i igual a:

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

1.3.3. Modelos lineales generalizados

❒ **Familia exponencial** – Se dice que una clase de distribuciones está en una familia exponencial si es posible escribirla en términos de un parámetro natural, también llamado parámetro canónico o función de enlace, η, un estadístico suficiente $T(y)$ y una función de log-partición (log-partition function) $a(\eta)$ de la siguiente manera:

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

Observación: comúnmente se tiene $T(y) = y$. Además, $\exp(-a(\eta))$ puede ser visto como un parámetro de normalización que asegura que las probabilidades sumen uno.

La siguiente tabla presenta un resumen de las distribuciones exponenciales más comunes:

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

❒ **Supuestos de los modelos GLM** – Los modelos lineales generalizados (en inglés, *Generalized Linear Models*) (GLM) tienen como objetivo la predicción de una variable aleatoria y como una función de $x \in \mathbb{R}^{n+1}$ bajo los siguientes tres supuestos:

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

Observación: los métodos de mínimos cuadrados ordinarios y regresión logística son casos particulares de los modelos lineales generalizados.

1.4. Máquinas de vectores de soportes

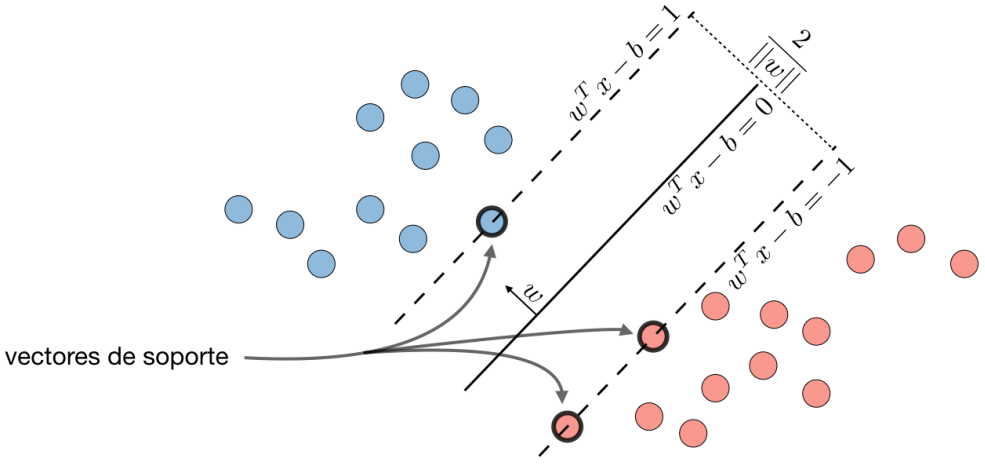
El objetivo de las máquinas de vectores de soportes (en inglés, *Support Vector Machines*) es hallar la línea que maximiza la mínima disancia a la línea.

❒ **Clasificador de margen óptimo** – El clasificador de márgen óptimo h se define de la siguiente manera:

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

donde $(w,b) \in \mathbb{R}^n \times \mathbb{R}$ es la solución del siguiente problema de optimización:

$$\boxed{\text{mín } \frac{1}{2}||w||^2} \quad \text{tal que} \quad \boxed{y^{(i)}(w^T x^{(i)} - b) \geqslant 1}$$



Observación: la línea se define como $\boxed{w^T x - b = 0}$.

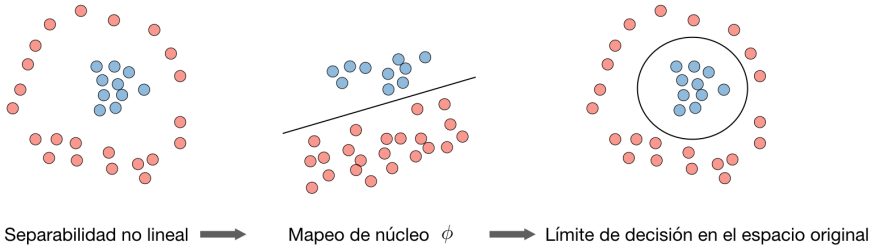
❒ **Función de pérdida de tipo bisagra** – La función de pérdida de tipo bisagra (en inglés, *Hinge loss*) es utilizada en la configuración de SVMs y se define de la siguiente manera:

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

❒ **Núcleo** – Dado un mapeo de características ϕ, se define el núcleo K (en inglés, *Kernel*) como:

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

En la práctica, el núcleo K definido por $K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$ es conocido como núcleo Gaussiano y es comúnmente utilizado.



Observación: decimos que utilizamos el "truco del núcleo"(en inglés, kernel trick) para calcular la función de costo porque en realidad no necesitamos saber explícitamente el mapeo ϕ que generalmente es muy complicado. En cambio, solo se necesitan los valores $K(x,z)$.

❒ **Lagrangiano** – Se define el Lagrangiano $\mathcal{L}(w,b)$ de la siguiente manera:

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

Observación: los coeficientes β_i son llamados multiplicadores de Lagrange.

1.5. Aprendizaje generativo

Un modelo generativo primero trata de aprender como se generan los datos estimando $P(x|y)$, lo que luego podemos utilizar para estimar $P(y|x)$ utilizando el Teorema de Bayes.

1.5.1. Análisis discriminante Gaussiano

❒ **Marco** – El Análisis discriminante Gaussiano (en inglés, *GDA - Gaussian Discriminant Analysis*) asume que y, $x|y=0$ y $x|y=1$ son de la siguiente forma:

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

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

❒ **Estimación** – La siguiente tabla resume las estimaciones encontradas al maximizar la probabilidad:

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

1.5.2. Naive Bayes

❒ **Supuestos** – El modelo Naive Bayes supone que las características de cada punto de los dato son todas independientes:

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

❒ **Soluciones** – Maximizar la log-probabilidad da las siguientes soluciones, con $k \in \{0,1\}$,$l \in [\![1,L]\!]$

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

Observación: Naive Bayes es comúnmente utilizado para la clasificación de texto y la detección de correo no deseado (spam).

1.6. Métodos basados en árboles y conjuntos

Estos métodos pueden ser utilizados tanto en problemas de regresión como de clasificación.

❒ **CART** – Árboles de clasificación y regresión (en inglés, *Classification and Regression Trees*) (CART), comúnmente conocidos como árboles de decisión, pueden ser representados como árboles binarios. Presentan la ventaja de ser muy interpretables.

❒ **Bosques aleatórios** – Es una téctica (en inglés, *Random Forest*) basada en árboles que utiliza una gran cantidad de árboles de decisión construidos a partir de conjuntos de características

seleccionadas al azar. A diferencia del árbol de decisión simple, la solución del método de bosques aleatórios es difícilmente interpretable aunque por su frecuente buen rendimiento es un algoritmo muy popular.

Observación: el método de bosques aleatorios es un típo de método de conjuntos.

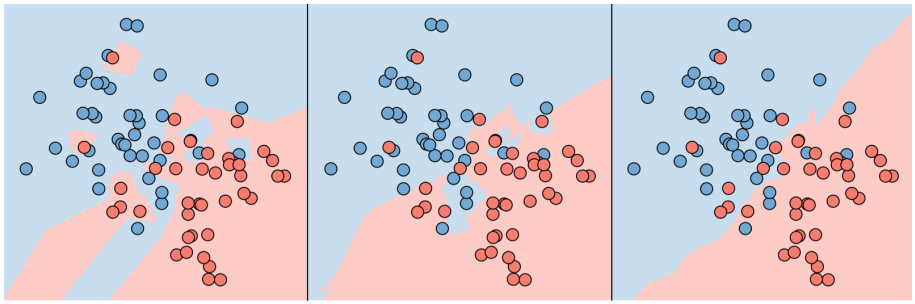
❒ **Potenciación** – La idea de la potenciación (en inglés, *Boosting*) es combinar varios métodos de aprendizaje débiles para conformar uno más fuerte. La siguiente tabla resume los principales tipos de potenciamiento:

Potenciamiento adaptativo	Potenciamiento del gradiente
- Se pondera fuertemente en los errores para mejorar en el siguiente paso del potenciamiento	- Los métodos de aprendizaje débiles entrenan sobre los errores restantes

1.7. Otros métodos no paramétricos

❒ **k vecinos más cercanos** – El algorítmo de k vecinos más cercanos (en inglés, *k-nearest neighbors algorith*), comunmente conocido como k-NN, es un método no paramétrico en el que la respuesta a un punto de los datos está determinada por la naturaleza de sus k vecinos del conjunto de datos. El método puede ser utilizado tanto en clasificaciones como regresiones.

Observación: Cuanto mayor es el parámetro k, mayor es el sesgo, y cuanto menor es el parámetro k, mayor la varianza.

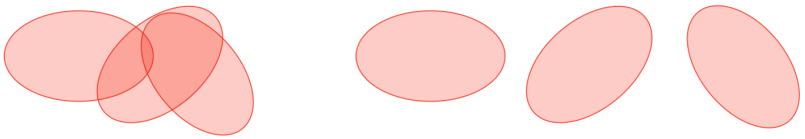


$k = 1$ $k = 3$ $k = 11$

1.8. Teoría del aprendizaje

❒ **Desigualdad de Boole** – Sean $A_1, ..., A_k$ k eventos, tenemos que:

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



$A_1 \cup A_2 \cup A_3$ A_1 A_2 A_3

❒ **Desigualdad de Hoeffding** – Sean $Z_1, .., Z_m$ m variables iid extraídas de una distribución de Bernoulli de parámetro ϕ. Sea $\widehat{\phi}$ su media emprírica y $\gamma > 0$ fija. Tenemos que:

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

Observación: esta desigualdad se conoce también como el límite de Chernoff.

❒ **Error de entrenamiento** – Para un clasificador dado h, se define el error de entrenamiento $\widehat{\epsilon}(h)$, también conocido como riesgo empírico o error empírico, de la siguiente forma:

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

❒ **Aprendizaje correcto probablemente aproximado (PAC)** – PAC es un marco bajo el cual se probaron numerosos resultados en teoría de aprendizaje, y presenta los siguientes supuestos:

- los conjuntos de entrenamiento y de prueba siguen la misma distribución
- los ejemplos de entrenamiento son escogidos de forma independiente

❒ **Shattering** – Dado un conjunto $S = \{x^{(1)},...,x^{(d)}\}$, y un conjunto de clasificadores $\mathcal{H}$, decimos que $\mathcal{H}$ destroza (shatters) S si para cualquier conjunto de etiquetas $\{y^{(1)}, ..., y^{(d)}\}$, tenemos que:

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

❒ **Teorema de la frontera superior** – Sea $\mathcal{H}$ una clase de hipótesis finita tal que $|\mathcal{H}| = k$ y sea δ y el tamaño de la muestra m fijo. Entonces, con probabilidad de al menos $1 - \delta$, tenemos:

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

❒ **Dimensión VC** – La dimensión de Vapnik-Chervonenkis (VC) de una clase de hipótesis finita $\mathcal{H}$, denotada como VC($\mathcal{H}$), es el tamaño del conjunto más grande destrozado (shattered) por $\mathcal{H}$.

Observación: la dimensión VC de $\mathcal{H}$ = {set of linear classifiers in 2 dimensions} es 3.

❒ **Teorema (Vapnik)** – Dado $\mathcal{H}$, con VC($\mathcal{H}$) $= d$ y m el número de ejemplos de entrenamiento. Con probabilidad de al menos $1 - \delta$, tenemos que:

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

2. Aprendizaje no Supervisado

2.1. Introducción al Aprendizaje no Supervisado

❒ **Motivación** – El objetivo del aprendizaje no supervisado es encontrar patrones ocultos en datos no etiquetados $\{x^{(1)},...,x^{(m)}\}$.

❒ **Desigualdad de Jensen** – Sea f una función convexa y X una variable aleatoria. Tenemos la siguiente desigualdad:

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

2.2. Agrupamiento

2.2.1. Expectativa-Maximización

❒ **Variables latentes** – Las variables latentes son variables ocultas/no observadas que dificultan los problemas de estimación y a menudo son denotadas como z. Estos son los ajustes más comunes en los que hay variables latentes:

Ajustes	Variance latente z	$x\|z$	Comentarios
Mezcla de k gaussianos	Multinomial(ϕ)	$\mathcal{N}(\mu_j,\Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
Análisis factorial	$\mathcal{N}(0,I)$	$\mathcal{N}(\mu+\Lambda z,\psi)$	$\mu_j \in \mathbb{R}^n$

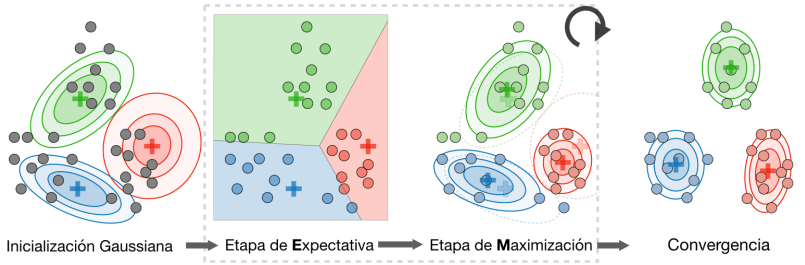
❒ **Algoritmo** – El algoritmo Expectativa-Maximización (EM) proporciona un método eficiente para estimar el parámetro θ a través de la estimación por máxima verosimilitud construyendo repetidamente un límite inferior en la probabilidad (E-step) y optimizando ese límite inferior (M-step) de la siguiente manera:

- E-step: Evalúa la probabilidad posterior $Q_i(z^{(i)})$ de que cada punto de datos $x^{(i)}$ provenga de un determinado clúster $z^{(i)}$ de la siguiente manera:

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

- M-step: Usa las probabilidades posteriores $Q_i(z^{(i)})$ como pesos específicos del clúster en los puntos de datos $x^{(i)}$ para re-estimar por separado cada modelo de clúster de la siguiente manera:

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

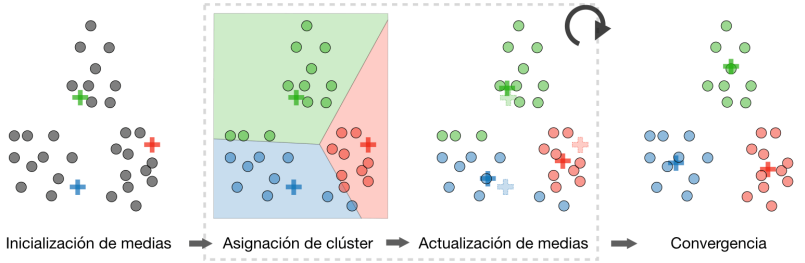


2.2.2. Agrupamiento k-means

Denotamos $c^{(i)}$ al clúster de puntos de datos i, y μ_j al centro del clúster j.

❒ **Algoritmo** – Después de haber iniciado aleatoriamente los centroides del clúster $\mu_1,\mu_2,...,\mu_k \in \mathbb{R}^n$, el algoritmo k-means repite el siguiente paso hasta la convergencia:

$$\boxed{c^{(i)} = \arg\min_j ||x^{(i)} - \mu_j||^2} \quad \text{y} \quad \boxed{\mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}}$$



❒ **Función de distorsión** – Para ver si el algoritmo converge, observamos la función de distorsión definida de la siguiente manera:

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

2.2.3. Agrupación jerárquica

❒ **Algoritmo** – Es un algoritmo de agrupamiento con un enfoque de aglomeramiento jerárquico que construye clústeres anidados de forma sucesiva.

❒ **Tipos** – Hay diferentes tipos de algoritmos de agrupamiento jerárquico que tienen por objetivo optimizar diferentes funciones objetivo, que se resumen en la tabla a continuación:

Enlace de Ward	Enlace promedio	Enlace completo
Minimizar dentro de la distancia del clúster	Minimizar la distancia promedio entre pares de clúster	Minimizar la distancia máxima entre pares de clúster

2.2.4. Métricas de evaluación de agrupamiento

En un entorno de aprendizaje no supervisado, a menudo es difícil evaluar el rendimiento de un modelo ya que no contamos con las etiquetas verdaderas, como en el caso del aprendizaje supervisado.

❒ **Coeficiente de silueta** – Sea a y b la distancia media entre una muestra y todos los demás puntos en la misma clase, y entre una muestra y todos los demás puntos en el siguiente grupo más cercano, el coeficiente de silueta s para una muestra individual se define de la siguiente manera:

$$\boxed{s = \frac{b-a}{\text{máx}(a,b)}}$$

❒ **Índice de Calinski-Harabaz** – Sea k el número de conglomerados, B_k y W_k las matrices de dispersión entre y dentro de la agrupación, respectivamente definidas como:

$$B_k = \sum_{j=1}^{k} n_{c^{(i)}} (\mu_{c^{(i)}} - \mu)(\mu_{c^{(i)}} - \mu)^T, \qquad W_k = \sum_{i=1}^{m} (x^{(i)} - \mu_{c^{(i)}})(x^{(i)} - \mu_{c^{(i)}})^T$$

el índice de Calinski-Harabaz $s(k)$ indica qué tan bien un modelo de agrupamiento define sus grupos, de tal manera que cuanto mayor sea la puntuación, más denso y bien separados estarán los conglomerados. Se define de la siguiente manera:

$$\boxed{s(k) = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N-k}{k-1}}$$

2.3. Reducción de la dimensionalidad

2.3.1. Análisis de componentes principales

Análisis de componentes principales (en inglés, *Principal Component Analysis*) es una técnica de reducción de la dimensionalidad que encuentra la varianza maximizando las direcciones sobre las cuales se proyectan los datos.

❒ **Autovalor, Autovector** – Dada una matriz $A \in \mathbb{R}^{n \times n}$, se dice que λ es un autovalor (en inglés, *Eigenvalue*) de A si existe un vector $z \in \mathbb{R}^n \backslash \{0\}$, llamado autovector (en inglés, *Eigenvector*), de tal manera que tenemos:

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

❒ **Teorema espectral** – Sea $A \in \mathbb{R}^{n \times n}$. Si A es simétrica, entonces A es diagonalizable a través de una matriz ortogonal real $U \in \mathbb{R}^{n \times n}$. Al observar $\Lambda = \text{diag}(\lambda_1,...,\lambda_n)$, tenemos:

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

Observación: el autovector asociado con el autovalor más grande se denomina autovector principal de la matriz A.

❒ **Algoritmo** – El procedimiento de Análisis de Componentes Principales (ACP) es una técnica de reducción de la dimensionalidad que proyecta los datos en k dimensiones maximizando la varianza de los datos de la siguiente manera:

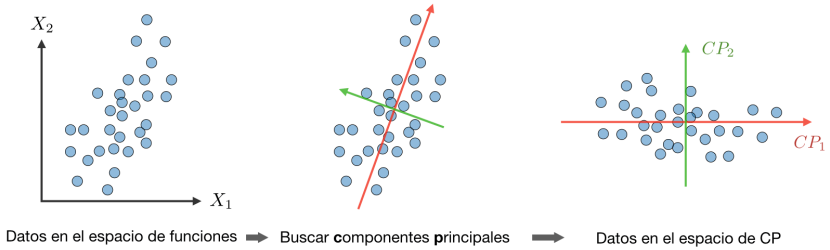
- Paso 1: Normalizar los datos para obtener una media de 0 y una desviación estándar de 1.

$$\boxed{x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j}} \quad \text{donde} \quad \boxed{\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}} \quad \text{y} \quad \boxed{\sigma_j^2 = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2}$$

- Paso 2: Calcular $\Sigma = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} x^{(i)^T} \in \mathbb{R}^{n \times n}$, que es simétrico con autovalores reales.

- Paso 3: Calcular $u_1, ..., u_k \in \mathbb{R}^n$ los k autovectores ortogonales principales de Σ, es decir, los autovectores ortogonales de los k mayores autovalores.

- Paso 4: Proyectar los datos en $\text{span}_{\mathbb{R}}(u_1,...,u_k)$. Este procedimiento maximiza la varianza entre todos los espacios k-dimensionales.



2.3.2. Análisis de componentes independientes

Es una técnica destinada a encontrar las fuentes generadoras subyacentes.

❒ **Suposiciones** – Suponemos que nuestros datos x han sido generados por el vector fuente n-dimensional $s = (s_1,...,s_n)$, donde s_i son variables aleatorias independientes; a través de una matriz A de mezcla y no singular, de la siguiente manera:

$$\boxed{x = As}$$

El objetivo es encontrar la matriz separadora $W = A^{-1}$.

❒ **Algoritmo ICA de Bell y Sejnowski** – Este algoritmo encuentra la matriz separadora W siguiendo los siguientes pasos:

- Escribir la probabilidad de $x = As = W^{-1}s$ como:

$$p(x) = \prod_{i=1}^{n} p_s(w_i^T x) \cdot |W|$$

- Escriba la probabilidad dado nuestros datos de entrenamiento $\{x^{(i)}, i \in [\![1,m]\!]\}$ y denotando g, la función sigmoide, como:

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

Por lo tanto, la regla de aprendizaje de ascenso de gradiente estocástica es tal que para cada ejemplo de entrenamiento $x^{(i)}$, actualizamos W de la siguiente manera:

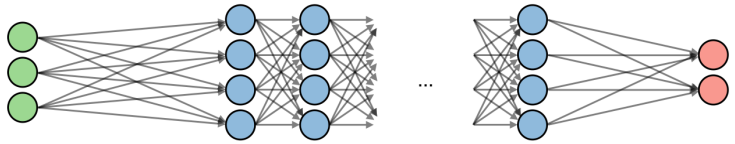
$$\boxed{W \longleftarrow W + \alpha \left(\begin{pmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{pmatrix} {x^{(i)}}^T + (W^T)^{-1} \right)}$$

3. Aprendizaje profundo

3.1. Redes neuronales

Las redes neuronales (en inglés, *Neural Networks*) son una clase de modelos construidos a base de capas. Los tipos más utilizados de redes neuronales incluyen las redes neuronales convolucionales y las redes neuronales recurrentes.

❒ **Arquitectura** – El vocabulario en torno a arquitecturas de redes neuronales se describe en la siguiente figura:

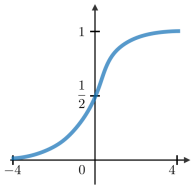
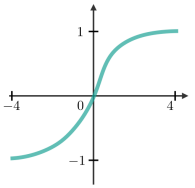
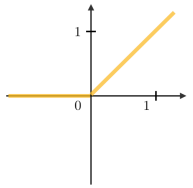
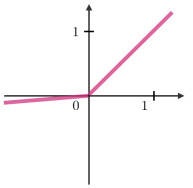


Denotando i en la i-ésima capa de la red y j en la j-ésima unidad oculta de la capa, tenemos:

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

donde w, b y z son el peso, el sesgo y la salida, respectivamente.

❒ **Función de activación** – Las funciones de activación son utilizadas al final de una unidad oculta para introducir complejidades no lineales al modelo. A continuación las más comunes:

Sigmoide	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 $\epsilon \ll 1$
			

❒ **Pérdida de cross-entropy** – En el contexto de las redes neuronales, la pérdida de cross-entropy $L(z,y)$ es utilizada comúnmente y definida de la siguiente manera:

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

❒ **Velocidad de aprendizaje** – La velocidad de aprendizaje (en inglés, *Learning rate*), denotada como α o algunas veces η, indica a que ritmo los pesos son actualizados. Este valor puede ser fijo o cambiar de forma adaptativa. El método más popular en este momento es llamado Adam, que es un método que adapta a la velocidad de aprendizaje.

❒ **Retropropagación** – La retropropagación (en inglés, *Backpropagation*), o propagación inversa, es un método de actualización de los pesos en una red neuronal, teniendo en cuenta la

salida actual y la salida esperada. La derivada respecto al peso w es calculada utilizando la regla de la cadena y se expresa de la siguiente forma:

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

Como resultado, el peso es actualizado de la siguiente forma:

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

❒ **Actualizando pesos** – En una red neuronal, los pesos son actualizados de la siguiente forma:

- Paso 1 : Tomar un lote de los datos de entrenamiento.
- Paso 2 : Realizar propagación hacia adelante para obtener la pérdida correspondiente.
- Paso 3 : Propagar inversamente la pérdida para obtener los gradientes.
- Paso 4 : Utiliza los gradientes para actualizar los pesos de la red.

❒ **Retiro** – El retiro (en inglés, *Dropout*) es una técnica para prevenir el sobreajuste de los datos de aprendizaje descartando unidades en una red neuronal. En la práctica, las neuronas son retiradas con una probabilidad de p o se mantienen con una probabilidad de $1-p$.

3.2. Redes neuronales convolucionales

❒ **Requisito de la capa convolucional** – Notando que W es el volumen de la entrada, F el tamaño de las neuronas de la capa convolucional, P la cantidad de relleno con ceros, entonces el número de neuronas N que entran en el volumen dado es tal que:

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

❒ **Normalización por lotes** – Es un paso de híperparámetro γ, β que normaliza el grupo $\{x_i\}$. Denotando μ_B, σ_B^2 la media y varianza del lote que queremos corregir, se realiza de la siguiente manera:

$$\boxed{x_i \longleftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta}$$

Se realiza usualmente después de una capa completamente conectada/convolucional y antes de una capa no-lineal y su objetivo es permitir velocidad más altas de aprendizaje y reducir su fuerte dependencia sobre la inicialización.

3.3. Redes neuronales recurrentes

❒ **Tipos de puerta** – A continuación, tenemos los diferentes tipos de compuertas que encontramos en una red neuronal recurrente típica:

Puerta de entrada	Puerta de olvido	Puerta	Puerta de salida
¿Escribir?	¿Borrar?	¿Cuánto escribir?	¿Cuánto revelar?

❒ **LSTM** – Una red de memoria a corto y largo plazo (en inglés, *Long Short Term Memory*) es un tipo de modelo de red neuronal recurrente que evita el problema del desvanecimiento del gradiente añadiendo puertas de 'olvido'.

3.4. Aprendizaje por refuerzo

El objetivo del aprendizaje por refuerzo es hacer que un agente aprenda como evolucionar en un entorno.

❒ **Procesos de decisión de Markov** – Un procesos de decisión de Markov (en inglés, *Markov Decision Process*) es una 5-tupla $(\mathcal{S}, \mathcal{A}, \{P_{sa}\}, \gamma, R)$ donde:

- $\mathcal{S}$ es el conjunto de estados
- $\mathcal{A}$ es el conjunto de acciones
- $\{P_{sa}\}$ son las probabilidades de transición de estado para $s \in \mathcal{S}$ y $a \in \mathcal{A}$
- $\gamma \in [0,1[$ es el factor de descuento
- $R : \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$ o $R : \mathcal{S} \longrightarrow \mathbb{R}$ es la función recompensa que el algoritmo pretende maximizar

❒ **Política** – Una política π es una función $\pi : \mathcal{S} \longrightarrow \mathcal{A}$ que asigna estados a acciones.

Observación: decimos que ejecutamos una política π dada si dado un estado s tomamos la acción $a = \pi(s)$.

❒ **Función valor** – Para una política dada π y un estado dado s, definimos el valor de la función V^π de la siguiente manera:

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

❒ **Ecuación de Bellman** – Las ecuaciones óptimas de Bellman, caracterizan la función valor V^{π^*} de la política óptima π^*:

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

Observación: denotamos que la política óptima π^* para un estado dado s es tal que:

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

❒ **Algoritmo de iteración valor** – El algoritmo de iteración valor es en dos pasos:

- Inicializamos el valor:

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

- Iteramos el valor con base en los valores de antes:

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

❒ **Estimación por máxima verosimilitud** – El estimación de probabilidad máximo para las probabilidades de transición de estado son como se muestra a continuación:

$$\boxed{P_{sa}(s') = \frac{\#\text{veces que se tomó la acción } a \text{ en el estado } s \text{ y llevó a } s'}{\#\text{veces que se tomó la acción } a \text{ en el estado}}}$$

❒ **Q-learning** – Q-learning es una estimación libre de modelo de Q, que se realiza de la siguiente forma:

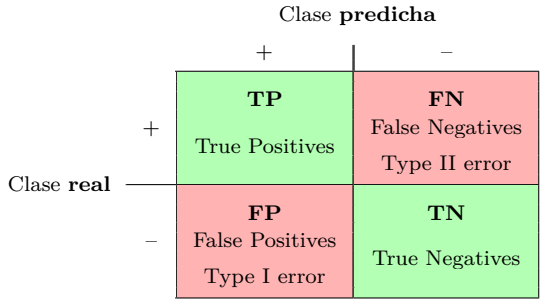
$$\boxed{Q(s,a) \leftarrow Q(s,a) + \alpha\left[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]}$$

4. Consejos y trucos sobre Aprendizaje Automático

4.1. Métricas para clasificación

En el contexto de una clasificación binaria, estas son las principales métricas que son importantes seguir para evaluar el rendimiento del modelo.

❒ **Matriz de confusión** – La matriz de confusión (en inglés, *Confusion matrix*) se utiliza para tener una visión más completa al evaluar el rendimiento de un modelo. Se define de la siguiente manera:



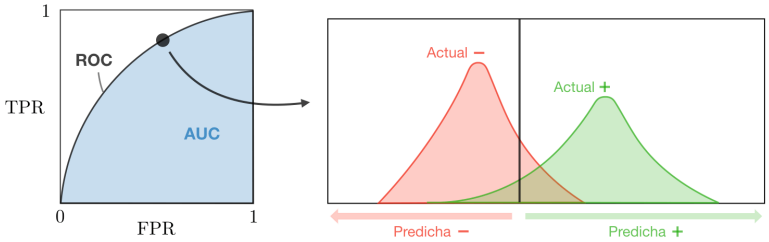
❒ **Métricas principales** – Las siguientes métricas se utilizan comúnmente para evaluar el rendimiento de los modelos de clasificación:

Métrica	Fórmula	Interpretación
Exactitud	$\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$	Rendimiento general del modelo
Precisión	$\frac{\text{TP}}{\text{TP}+\text{FP}}$	Que tan precisas son las predicciones positivas
Exhaustividad Sensibilidad	$\frac{\text{TP}}{\text{TP}+\text{FN}}$	Cobertura de la muestra positiva real
Especificidad	$\frac{\text{TN}}{\text{TN}+\text{FP}}$	Cobertura de la muestra negativa real
F1 score	$\frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$	Métrica híbrida útil para clases desbalanceadas

❒ **ROC** – La curva Característica Operativa del Receptor (en inglés, *Receiver Operating Curve*), también conocida como ROC, es una representación gráfica de la sensibilidad frente a la especificidad según se varía el umbral. Estas métricas se resumen en la siguiente tabla:

Métrica	Fórmula	Interpretación
True Positive Rate TPR	$\frac{\text{TP}}{\text{TP}+\text{FN}}$	Exhaustividad, sensibilidad
False Positive Rate FPR	$\frac{\text{FP}}{\text{TN}+\text{FP}}$	1-especificidad

❒ **AUC** – El área bajo la curva Característica Operativa del Receptor, también conocida como AUC o AUROC (en inglés, *Area Under the Receiving Operating Curve*), es el área debajo del ROC, como se muestra en la siguiente figura:



4.2. Métricas de regresión

❒ **Métricas básicas** – Dado un modelo de regresión f, las siguientes métricas se usan comúnmente para evaluar el rendimiento del modelo:

Suma total de cuad.	Suma de cuad. explicada	Suma residual de cuad.
$\mathrm{SS}_{\mathrm{tot}} = \sum_{i=1}^{m}(y_i - \overline{y})^2$	$\mathrm{SS}_{\mathrm{reg}} = \sum_{i=1}^{m}(f(x_i) - \overline{y})^2$	$\mathrm{SS}_{\mathrm{res}} = \sum_{i=1}^{m}(y_i - f(x_i))^2$

❒ **Coeficiente de determinación** – El coeficiente de determinación, a menudo indicado como R^2 o r^2, proporciona una medida de lo bien que los resultados observados son replicados por el modelo y se define de la siguiente manera:

$$\boxed{R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}}$$

❒ **Métricas principales** – Las siguientes métricas se utilizan comúnmente para evaluar el rendimiento de los modelos de regresión, teniendo en cuenta la cantidad de variables n que tienen en consideración:

Cp de Mallow	AIC	BIC	R^2 ajustado
$\frac{\mathrm{SS}_{\mathrm{res}} + 2(n+1)\widehat{\sigma^2}}{m}$	$2\big[(n+2) - \log(L)\big]$	$\log(m)(n+2) - 2\log(L)$	$1 - \frac{(1-R^2)(m-1)}{m-n-1}$

donde L es la probabilidad y $\widehat{\sigma^2}$ es una estimación de la varianza asociada con cada respuesta.

4.3. Selección de modelo

❒ **Vocabulario** – Al seleccionar un modelo, distinguimos 3 partes diferentes de los datos que tenemos de la siguiente manera:

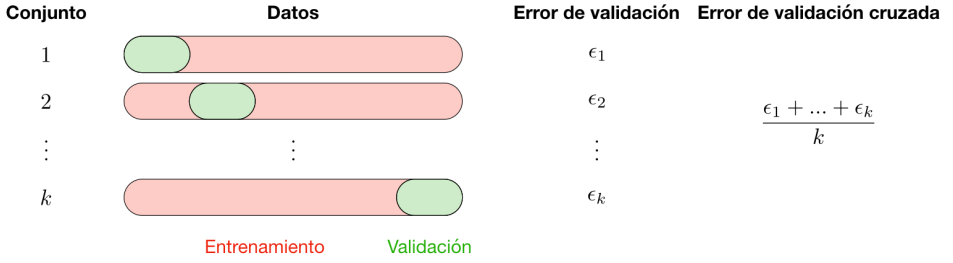
Entrenamiento	Validación	Prueba
- Modelo es entrenado - Generalmente el 80 % del conjunto de datos	- Modelo es evaluado - Generalmente 20 % - También llamado hold-out o conjunto de desarrollo	- Modelo da predicciones - Datos no vistos

Una vez que se ha elegido el modelo, se entrena sobre todo el conjunto de datos y se testea sobre el conjunto de prueba no visto. Estos están representados en la figura a continuación:

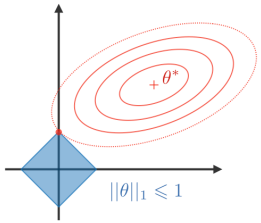
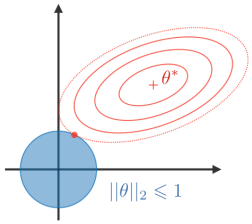
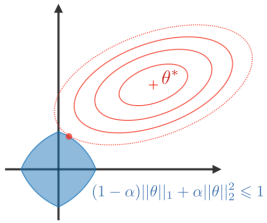
❒ **Validación cruzada** – La validación cruzada, también denominada CV (en inglés, *Cross validation*), es un método que se utiliza para seleccionar un modelo que no confíe demasiado en el conjunto de entrenamiento inicial. Los diferentes tipos se resumen en la tabla a continuación:

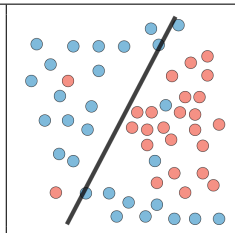
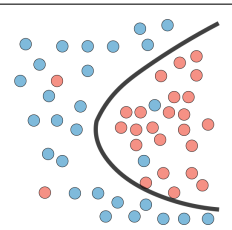
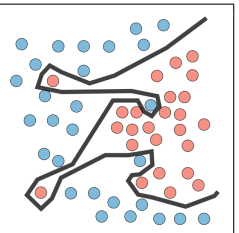
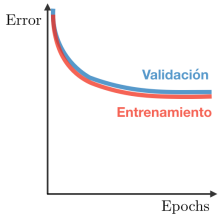
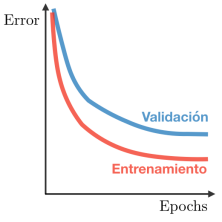
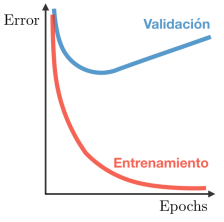
k-fold	Leave-p-out
- Entrenamiento sobre los conjuntos $k-1$ y evaluación en el restante - Generalmente $k = 5$ o 10	- Entrenamiento en observaciones $n-p$ y evaluación en los p restantes - El caso $p = 1$ se llama *leave-one-out*

El método más comúnmente utilizado se denomina validación cruzada k-fold y divide los datos de entrenamiento en k conjuntos para validar el modelo sobre un conjunto mientras se entrena el modelo en los otros $k-1$ conjuntos, todo esto k veces. El error luego se promedia sobre los k conjuntos y se denomina error de validación cruzada.



❒ **Regularización** – El procedimiento de regularización tiene como objetivo evitar que el modelo se sobreajuste a los datos y, por lo tanto, resuelve los problemas de alta varianza. La siguiente tabla resume los diferentes tipos de técnicas de regularización comúnmente utilizadas:

LASSO	Ridge	Elastic Net
- Reduce los coeficientes a 0 - Bueno para la selección de variables	Hace que los coeficientes sean más pequeños	Compensación entre la selección de variables y los coeficientes pequeños
		
$... + \lambda\|\|\theta\|\|_1$ $\lambda \in \mathbb{R}$	$... + \lambda\|\|\theta\|\|_2^2$ $\lambda \in \mathbb{R}$	$... + \lambda\left[(1-\alpha)\|\|\theta\|\|_1 + \alpha\|\|\theta\|\|_2^2\right]$ $\lambda \in \mathbb{R}$, $\alpha \in [0,1]$

Clasificación			
Deep Learning			
Soluciones	- Incrementar la complejidad del modelo - Agregar más funciones - Entrenar más tiempo		- Realizar la regularización - Obtener más datos

4.4. Diagnóstico

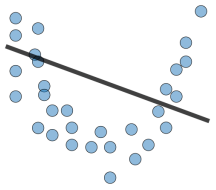
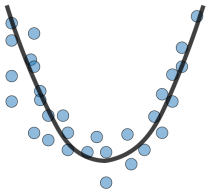
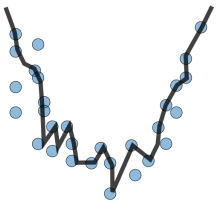
❒ **Sesgo** – El sesgo (en inglés, *Bias*) de un modelo es la diferencia entre la predicción esperada y el modelo correcto que tratamos de predecir para determinados puntos de datos.

❒ **Varianza** – La varianza (en inglés, *Variance*) de un modelo es la variabilidad de la predicción del modelo para puntos de datos dados.

❒ **Corrección de sesgo/varianza** – Cuanto más simple es el modelo, mayor es el sesgo, y cuanto más complejo es el modelo, mayor es la varianza.

❒ **Análisis de errores** – El análisis de errores analiza la causa raíz de la diferencia de rendimiento entre los modelos actuales y perfectos.

❒ **Análisis ablativo** – El análisis ablativo analiza la causa raíz de la diferencia en el rendimiento entre los modelos actuales y de referencia.

	Underfitting	Just right	Overfitting
Síntomas	- Error de entrenamiento alto - Error de entrenamiento cercano al error de prueba - Sesgo alto	- Error de entrenamiento légèrement inférieure à l'erreur de test	- Error de entrenamiento muy bajo - Error de entrenamiento mucho más bajo que el error de prueba - Varianza alta
Regresión			

5. Repaso

5.1. Probabilidades y Estadísticas

5.1.1. Introducción a la probabilidad y combinatoria

❒ **Espacio muestral** – El conjunto de todos los posibles resultados de un experimento es conocido como el espacio muestral del experimento y se denota como S.

❒ **Evento** – Cualquier subconjunto E del espacio muestral es conocido como un evento. Esto significa que un evento es un conjunto de posibles resultados de un experimento. Si el resultado de un experimento esta contenido en E, entonces decimos que el evento E ha ocurrido.

❒ **Axiomas de la probabilidad** – Para cada evento E, denota $P(E)$ como la probabilidad de que el evento E ocurra.

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

❒ **Permutación** – Una permutación es un arreglo de r objetos tomados de un grupo de n objetos, en un order arbitrario. El número de estos arreglos es dado por $P(n,r)$, definido como:

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

❒ **Combinación** – Una combinación es un arreglo de r objetos tomados de un grupo de n objetos, donde el orden no importa. El número de estos arreglos es dado por $C(n,r)$, definido como:

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

Observación: cabe resaltar que para $0 \leqslant r \leqslant n$, se tiene $P(n,r) \geqslant C(n,r)$.

5.1.2. Probabilidad condicional

❒ **Regla de Bayes** – Para eventos A y B tal que $P(B) > 0$, se tiene:

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

Observación: Se tiene $P(A \cap B) = P(A)P(B|A) = P(A|B)P(B)$.

❒ **Partición** – Sea $\{A_i, i \in [\![1,n]\!]\}$ tal que para todo i, $A_i \neq \varnothing$. Se dice entonces que $\{A_i\}$ es una partición si se cumple:

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

Observación: Para cualquier evento B del espacio muestral, se cumple $P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$.

❒ **Regla de Bayes extendida** – Sea $\{A_i, i \in [\![1,n]\!]\}$ una partición del espacio muestral. Se cumple:

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

❒ **Independencia** – Dos events A y B son independientes si y solo si se cumple:

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

5.1.3. Variables aleatorias

❒ **Variable aleatoria** – Una variable aleatoria, generalmente denotada por X, es una función que asocia cada elemento de un espacio muestral a una linea real.

❒ **Función de distribución acumulada (FDA)** – La función de distribución acumulada F (en inglés *CDF - Cumulative distribution function*), la cual es monótonamente creciente y es tal que:

$$\lim_{x \to -\infty} F(x) = 0 \quad \text{y} \quad \lim_{x \to +\infty} F(x) = 1$$

es definida como:

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

Observación: Se tiene $P(a < X \leqslant B) = F(b) - F(a)$.

❒ **Función de densidad de probabilidad (FDP)** – La función de densidad de probabilidad f (en inglés *PDF - Probability density function*) es la probabilidad que X tome valores entre dos ocurrencias adyacentes de la variable aleatoria.

❒ **Relaciones entre la FDA y FDP** – Estas son las propiedades mas importantes para conocer en los casos discreto (D) y contínuo (C).

Caso	FDA F	FDP f	Propiedades de PDF
(D)	$F(x) = \sum_{x_i \leqslant x} P(X = x_i)$	$f(x_j) = P(X = x_j)$	$0 \leqslant f(x_j) \leqslant 1$ and $\sum_j f(x_j) = 1$
(C)	$F(x) = \int_{-\infty}^{x} f(y)dy$	$f(x) = \frac{dF}{dx}$	$f(x) \geqslant 0$ and $\int_{-\infty}^{+\infty} f(x)dx = 1$

❒ **Varianza** – La varianza de una variable aleatoria, frecuentemente denotada por $\text{Var}(X)$ o σ^2, es la medida de dispersión de su función de distribución. Esta determinada de la siguiente manera:

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

❒ **Desviación estándar** – La desviación estándar de una variable aleatoria, frecuentemente denotada por σ, es una medida de la dispersión de su función de distribución la cual es compatible con las unidades de la correspondiente variable aleatoria. Se determina de la siguiente manera:

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

❒ **Valor esperado y momentos de la distribución** – Aquí están las expresiones del valor esperado $E[X]$, valor esperado generalizado $E[g(X)]$, $k^{ésimo}$ momento $E[X^k]$ y función característica $\psi(\omega)$ para los casos discreto y contínuo:

Caso	$E[X]$	$E[g(X)]$	$E[X^k]$	$\psi(\omega)$
(D)	$\sum_{i=1}^{n} x_i f(x_i)$	$\sum_{i=1}^{n} g(x_i) f(x_i)$	$\sum_{i=1}^{n} x_i^k f(x_i)$	$\sum_{i=1}^{n} f(x_i) e^{i\omega x_i}$
(C)	$\int_{-\infty}^{+\infty} x f(x) dx$	$\int_{-\infty}^{+\infty} g(x) f(x) dx$	$\int_{-\infty}^{+\infty} x^k f(x) dx$	$\int_{-\infty}^{+\infty} f(x) e^{i\omega x} dx$

Observación: se tiene $e^{i\omega x} = \cos(\omega x) + i\sin(\omega x)$.

❒ **Transformación de variables aleatorias** – Sean las variables X y Y asociadas por alguna función. Denotemos como f_X y f_Y la función de distribución de X y Y respectivamente, se tiene:

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

❒ **Regla integral de Leibniz** – Sea g una función de x y posiblemente de c, y además sea a, b, un intervalo que puede depender de c. Se tiene:

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

❒ **Desigualdad de Chebyshev** – Sea X una variable aleatoria con valor esperado μ. Para $k, \sigma > 0$, se tiene la siguiente desigualdad:

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

5.1.4. Variables aleatorias conjuntas

❒ **Densidad condicional** – La densidad condicional de X con respecto a Y, frecuentemente denotada como $f_{X|Y}$, es definida como:

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

❒ **Independencia** – Dos variables aleatorias X y Y son consideradas independientes si se tiene:

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

❒ **Densidad marginal y distribución acumulada** – De la función conjunta de densidad de probabilidad f_{XY}, se tiene:

Caso	Densidad marginal	Función acumulativa
(D)	$f_X(x_i) = \sum_j f_{XY}(x_i, y_j)$	$F_{XY}(x,y) = \sum_{x_i \leqslant x} \sum_{y_j \leqslant y} f_{XY}(x_i, y_j)$
(C)	$f_X(x) = \int_{-\infty}^{+\infty} f_{XY}(x,y) dy$	$F_{XY}(x,y) = \int_{-\infty}^{x} \int_{-\infty}^{y} f_{XY}(x',y') dx' dy'$

❒ **Covarianza** – Definimos la covarianza de dos variables aleatorias X y Y, denotada como σ_{XY}^2 o comúnmente como $\text{Cov}(X,Y)$, de la siguiente manera:

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

❒ **Correlación** – Sean σ_X, σ_Y las desviaciones estándard de X y Y, definimos la correlación entre estas variables, denotada como ρ_{XY}, de la siguiente manera:

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

Observaciones 1: cabe resaltar que para X, Y, variables aleatorias cualesquiera, se tiene que $\rho_{XY} \in [-1,1]$. Si X y Y son independientes, entonces $\rho_{XY} = 0$.

❒ **Distribuciones importantes** – Aquí están las distribuciones más importantes para tomar en cuenta:

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

5.1.5. Estimación de parámetros

❒ **Muestra aleatoria** – Una muestra aleatoria es una colección de n variables aleatorias $X_1, ..., X_n$ que son independientes e idénticamente distribuidas a X.

❒ **Estimador** – Un estimador es una función de los datos que es usada para inferir el valor de un parámetro desconocido en un modelo estadístico.

❒ **Sesgo** – El sesgo de un estimador $\hat{\theta}$ se define como la diferencia entre el valor esperado de la distribución de $\hat{\theta}$ y el valor exacto, esto es:

$$\boxed{\text{Sesgo}(\hat{\theta}) = E[\hat{\theta}] - \theta}$$

Observación: se dice que un estimador es no sesgado cuando se tiene $E[\hat{\theta}] = \theta$.

❒ **Media de la muestra** – La media de la muestra aleatoria se usa para estimar el valor exacto de la media μ de la distribución, se denota frecuentemente como $\overline{X}$ y se define de la siguiente manera:

$$\boxed{\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i}$$

Observación: la media de la muestra es no sesgada, esto es $E[\overline{X}] = \mu$.

❒ **Media de la muestra** – La media de la muestra aleatoria se usa para estimar el valor exacto de la media μ de la distribución, se denota frecuentemente como $\overline{X}$ y se define de la siguiente manera:

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

Observación: la varianza de la muestra es no sesgada, esto es $E[s^2] = \sigma^2$.

❒ **Teorema del Límite Central** – Sea $X_1, ..., X_n$ una muestra aleatoria que sigue una distribución con media μ y varianza σ^2, entonces se tiene:

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

5.2. Álgebra Lineal y Cálculo

5.2.1. Notaciones Generales

❒ **Vector** – Sea $x \in \mathbb{R}^n$ un vector con n entradas, donde $x_i \in \mathbb{R}$ es la i-ésima entrada:

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

❒ **Matriz** – Sea $A \in \mathbb{R}^{m\times n}$ una matriz con m filas y n columnas; donde $A_{i,j} \in \mathbb{R}$ es el valor ubicado en la i-ésima fila y la j-ésima columna:

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

Observación: el vector x definido arriba puede ser visto como una matriz $n \times 1$ y es particularmente llamado vector-columna.

❒ **Matriz identidad** – La matriz identidad $I \in \mathbb{R}^{n\times n}$ es una matriz cuadrada con valores 1 en su diagonal y ceros en el resto:

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

Observación: para todas las matrices $A \in \mathbb{R}^{n\times n}$, se cumple que $A \times I = I \times A = A$.

❒ **Matriz diagonal** – Una matriz diagonal $D \in \mathbb{R}^{n\times n}$ es una matriz cuadrada con valores diferentes de zero en su diagonal y cero en el resto:

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

Observación: también se denota D como $diag(d_1,...,d_n)$.

5.2.2. Operaciones de matrices

❒ **Vector-vector** – Hay dos tipos de multiplicaciones vector-vector:

- Producto interno: para $x,y \in \mathbb{R}^n$, se tiene que:

$$\boxed{x^T y = \sum_{i=1}^{n} x_i y_i \in \mathbb{R}}$$

- Producto diádico : para $x \in \mathbb{R}^m, y \in \mathbb{R}^n$, se tiene que:

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

❒ **Matriz-vector** – El producto de la matriz $A \in \mathbb{R}^{m\times n}$ y el vector $x \in \mathbb{R}^n$, es un vector de tamaño $\mathbb{R}^m$; tal que:

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

donde $a_{r,i}^T$ son los vectores fila y $a_{c,j}$ son los vectores columna de A, y x_i son las entradas de x.

❒ **Matriz-matriz** – El producto de las matrices $A \in \mathbb{R}^{m\times n}$ y $B \in \mathbb{R}^{n\times p}$ es una matriz de tamaño $\mathbb{R}^{n\times p}$, tal que:

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

donde $a_{r,i}^T, b_{r,i}^T$ son los vectores fila y $a_{c,j}, b_{c,j}$ son los vectores columna de A y B respectivamente.

❒ **Transpuesta** – La transpuesta de una matriz $A \in \mathbb{R}^{m\times n}$, denotada A^T, es tal que sus entradas se intercambian de la siguiente forma:

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

Observación: dadas las matrices A,B, se cumple que $(AB)^T = B^T A^T$.

❒ **Inversa** – La inversa de una matriz cuadrada invertible A es denotada como A^{-1} y es la única matriz tal que:

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

Observación: no todas las matrices cuadradas son invertibles. Además, para las matrices A,B, se cumple que $(AB)^{-1} = B^{-1}A^{-1}$.

❒ **Traza** – La traza de una matriz cuadrada A, denotada $\text{tr}(A)$, es la suma de los elementos en su diagonal:

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

Observación: dadas las matrices A,B, se cumple que $tr(A^T) = tr(A)$ y $tr(AB) = tr(BA)$.

❒ **Determinante** – El determinante de una matriz cuadrada $A \in \mathbb{R}^{n\times n}$, denotado $|A|$ o $\det(A)$, es expresado recursivamente en términos de $A_{\backslash i,\backslash j}$, que es la matriz A sin su i-ésima fila ni su j-ésima columna, de la siguiente forma:

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

Observación: A es invertible si y solo si $|A| \neq 0$. Además, $|AB| = |A||B|$ y $|A^T| = |A|$.

5.2.3. Propiedades de matrices

❒ **Descomposición simétrica** – Una matriz dada A puede ser expresada en términos de sus partes simétricas y antisimétricas de la siguiente forma:

$$\boxed{A = \underbrace{\frac{A + A^T}{2}}_{\text{Simétrica}} + \underbrace{\frac{A - A^T}{2}}_{\text{Antisimétricas}}}$$

❒ **Norma** – Una norma (o módulo) es una función $N : V \longrightarrow [0, +\infty[$ donde V es un vector espacial tal que para todo x,$y \in V$, se cumple que:

- $N(x + y) \leqslant N(x) + N(y)$
- $N(ax) = |a|N(x)$ siendo a un escalar
- si $N(x) = 0$, entonces $x = 0$

Para $x \in V$, las normas comúnmente utilizadas se resumen en la siguiente tabla:

Norma	Notación	Definición	Caso de uso
Manhattan, L^1	$\|\|x\|\|_1$	$\sum_{i=1}^{n} \|x_i\|$	LASSO
Euclidean, L^2	$\|\|x\|\|_2$	$\sqrt{\sum_{i=1}^{n} x_i^2}$	Ridge
p-norma, L^p	$\|\|x\|\|_p$	$\left(\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}}$	Desigualdad de Hölder
Infinito, L^∞	$\|\|x\|\|_\infty$	$\max_i \|x_i\|$	Convergencia

❒ **Dependencia lineal** – Un conjunto de vectores se dice que es linealmente dependiente si uno de los vectores del conjunto puede ser definido como una combinación lineal de los restantes.

Observación: si ningún vector puede ser escrito de esta forma, entonces se dice que los vectores son linealmente independientes.

❒ **Rango matricial** – El rango de una matriz dada A es denotado $\text{rank}(A)$ y es la dimensión del espacio vectorial generado por sus columnas. Esto es equivalente al máximo número de columnas linealmente independientes de A.

❒ **Matriz semi-definida positiva** – Una matriz $A \in \mathbb{R}^{n\times n}$ es semi-defininda positiva (PSD), lo cual se denota como $A \succeq 0$, si se cumple que:

$$\boxed{A = A^T} \quad \text{y} \quad \boxed{\forall x \in \mathbb{R}^n, \quad x^T Ax \geqslant 0}$$

Observación: de igual forma, una matriz A se dice definida positiva, lo cual se denota con $A \succ 0$, si es una matriz PSD que satisface para todos los vectores x diferentes de cero, $x^T Ax > 0$.

❒ **Valor propio, vector propio** – Dada una matriz $A \in \mathbb{R}^{n\times n}$, λ se dice que es un valor propio (*eigenvalue*, en inglés) de A si existe un vector $z \in \mathbb{R}^n \backslash \{0\}$, llamado vector propio (*eigenvector*, en inglés), tal que:

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

❒ **Teorema espectral** – Sea $A \in \mathbb{R}^{n\times n}$. Si A es simétrica, entonces A es diagonalizable a través de una matriz real ortogonal $U \in \mathbb{R}^{n\times n}$. Denotando $\Lambda = \text{diag}(\lambda_1,...,\lambda_n)$, se tiene que:

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

❒ **Descomposición en valores singulares** – Para una matriz dada A de dimensiones $m \times n$, la descomposición en valores singulares (SVD, por sus siglas en inglés de *Singular-Value*

Decomposition) es una técnica de factorización que garantiza la existencia de las matrices U $m \times m$ unitaria, Σ $m \times n$ diagonal y V $n \times n$ unitaria, tal que:

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

5.2.4. Cálculo de matrices

❒ **Gradiente** – Sea $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ una función y $A \in \mathbb{R}^{m \times n}$ una matriz. El gradiente de f con respecto a A es una matriz de $m \times n$, denotada por $\nabla_A f(A)$, tal que:

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

Observación: el gradiente de f se define únicamente cuando f es una función que retorna un escalar.

❒ **Hessiana** – Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función y $x \in \mathbb{R}^n$ un vector. La hessiana de f con recpecto a x es una matriz simétrica $n \times n$, denotada $\nabla^2_x f(x)$, tal que:

$$\boxed{\left(\nabla^2_x f(x)\right)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}}$$

Observación: la matriz hessiana de f se define únicamente cuando f es una función que devuelve un escalar.

❒ **Operaciones de gradiente** – Dadas las matrices A,B,C, las siguientes propiedades de gradiente merecen ser tenidas en cuenta:

$$\boxed{\nabla_A \text{tr}(AB) = B^T} \qquad \boxed{\nabla_{A^T} f(A) = (\nabla_A f(A))^T}$$

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