UNIVERSIDAD DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA QUÍMICA, BIOTECNOLOGÍA Y MATERIALES

VHL-Hunter, Servicio web de clasificación de mutaciones en VHL

DAVID ALFREDO MEDINA ORTIZ

Profesor Tutor: ÁLVARO OLIVERA NAPPA, PH.D.

Resumen Desarrollo de Proyecto,
Seminario 02

Santiago – Chile
Julio, 2018

TABLA DE CONTENIDOS

página

ÍNDICE DE FIGURAS

página

ÍNDICE DE TABLAS

página

RESUMEN

La enfermedad de von Hippel-Lindau (VHL) es un trastorno genético con afectación multisistémica. Se asocia con tumores múltiples, incluidos retina y el sistema nervioso central, hemangioblastomas, carcinoma renal de células claras y el feocromocitoma. Siendo asociado al resultado de mutaciones en el gen VHL. Se han reportado miles de mutaciones asociadas con VHL. Sin embargo, para un gran porcentaje de ellas, su relevancia clínica aún no está clara, esta es la razón por la que VHL-Hunter se propone como un sistema de clasificación para mutaciones en VHL.

La herramienta procesa la mutación puntual y entrega como resultado si es clínicamente relevante o no. VHL-Hunter calcula la información filogenética y las energías de interacción y estabilidad, a través del uso de las herramientas MOSST y SDM. Además, se considera la información topológica de la proteína, por lo que la mutación se clasifica previamente según su posición en la proteína en algunos de los sectores de interacción proteína-proteína que han sido notificados, lo que permite utilizar el modelo de clasificación que corresponde según su sitio de interacción.

VHL-Hunter se compone de 13 modelos de clasificación, correspondientes a cada sector de interacción proteína-proteína. Todos los modelos fueron entrenados aplicando una estrategia exploratoria de algoritmos de clasificación y variación de parámetros, obteniendo distribuciones de medidas de desempeño, lo que permitió obtener un conjunto de modelos para cada sector de interacción, los cuales fueron filtrados por técnicas estadísticas. Los modelos se obtuvieron con tasas de verdaderos positivos y negativos, altos, agregando un alto valor de precisión. En los casos en que los modelos no permitieron obtener las características deseadas, se aplicaron técnicas de meta-learning para aumentar el rendimiento.

1. INTRODUCCIÓN

1.1. Von Hippel-Lindau

La enfermedad de von Hippel-Lindau (VHL) es un síndrome de neoplasia dominante autosómica que resulta de una mutación en la línea germinal en el gen VHL [37, 38, 39, 40, 41, 42, 43]. VHL se caracteriza por ser un síndrome de cáncer hereditario caracterizado por el desarrollo de tumores vasculares del sistema nervioso central y retina, carcinomas renales de células claras, feocromocitomas, tumores de células de islotes pancreáticos, tumores del saco endolinfático y quistes benignos que afectan una variedad de órganos [44].

El producto canónico de la proteína VHL, $pVHL$ en su isoforma 1 ($pVHL_{30}$), tiene dos dominios estructuralmente diferentes: un dominio desordenado de 53 aminoácidos N-terminal no necesario para la supresión tumoral y un dominio ordenado C-terminal que consiste en un dominio α-helicoidal (residuos 155-192) y un dominio principalmente de hoja β (residuos 63-154 y 193-204). pVHL forma un complejo ternario con las proteínas elongina C y elongina B (Complejo VCB) [47, 48] el cual es crítico para la estabilidad y la función en pVHL [49].

Las mutaciones que afectan a los residuos de unión a pVHL en elongina C se han descrito en ccRCC, lo que respalda la hipótesis de que los efectos tumorogénicos de las mutaciones de VHL se relacionan con la disfunción del complejo VCB. Por lo tanto, todo el complejo de VCB debe considerarse como una sola entidad cuando se evalúan los efectos estructurales y funcionales de las mutaciones de VHL [45].

VHL presenta una incidencia de 1 en 36.000 pacientes, los cuales se predisponen al desarrollo de hemangioblastoma retiniano, cerebeloso y espinal, quistes y carcinoma de páncreas y RCC [46], adicional a ello, han sido documentadas, sobre 1000 mutaciones en VHL, de las cuales cerca del 52 % se desconoce su relevancia clínica o no se tiene una mayor información de ésta [45].

Las características viscerales del trastorno incluyen quistes y carcinomas renales, los feocromocitomas, los quistes pancreáticos y los tumores neuroendocrinos, así como los

cistadenomas del epidídimo y del ligamento ancho [43], cuyos índices de aparición se exponen en la Figura 1.1.

	Mean (range) age of onset (years)	Frequency in patients (%)
CNS		
Retinal haemangioblastomas	25 (1–67)	25–60%
Endolymphatic sac tumours	22 (12–50)	10%
Craniospinal haemangioblastomas		
Cerebellum	33 (9–78)	44–72%
Brainstem	32 (12–46)	10–25%
Spinal cord	33 (12–66)	13–50%
Lumbosacral nerve roots	Unknown (..)	<1%
Supratentorial	Unknown (..)	<1%
Visceral		
Renal cell carcinoma or cysts	39 (16–67)	25–60%
Phaeochromocytomas	30 (5–58)	10–20%
Pancreatic tumour or cyst	36 (5–70)	35–70%
Epididymal cystadenoma	Unknown (..)	25–60%
Broad ligament cystadenoma	Unknown (16–46)	Unknown

Figura 1.1: Resumen de incidencias de afecciones relacionadas a VHL.

El diagnóstico de la enfermedad de von Hippel-Lindau a menudo se basa en criterios clínicos. Los pacientes con antecedentes familiares y un hemangioblastoma del SNC (incluidos los hemangioblastomas retinianos), un feocromocitoma o un carcinoma renal de células claras son diagnosticados con la enfermedad. Los que no tienen antecedentes familiares relevantes deben tener dos o más hemangioblastomas del SNC o un hemangioblastoma del SNC y un tumor visceral (a excepción de quistes epididimarios y renales, que son frecuentes en la población general) para cumplir con los criterios diagnósticos [50, 51, 52].

Las correlaciones específicas de genotipo y fenotipo han surgido en las familias afectadas. Ahora se reconocen varios fenotipos familiares de la enfermedad de von Hippel-Lindau, que proporcionan información útil para examinar y aconsejar a las personas afectadas. Las familias tipo 1 tienen un riesgo muy reducido de feocromocitomas, pero pueden desarrollar todos los otros tipos de tumores generalmente asociados con la enfermedad. Las familias de tipo 2 tienen feocromocitomas, pero tienen un bajo riesgo (tipo 2A) o alto riesgo (tipo 2B) para los carcinomas de células renales. Las familias de tipo 2C solo tienen feocromocitomas, sin otros hallazgos neoplásicos de VHL [43].

Los avances en las pruebas genéticas para la detección de la enfermedad incluyen *Southern blotting cualitativo y cuantitativo*, que se ha agregado al análisis de la secuencia de ADN. Esto mejoró para uso personal. No obstante, su reproducción es sólo con permiso de

The Lancet Publishing Group. La prueba ha aumentado la tasa de detección de mutaciones de ADN en leucocitos de sangre periférica del 75 % a casi el 100 % [43].

Se han desarrollado diversos software que permiten explorar el efecto de las mutaciones en las proteínas que han sido detectadas y como afectan a su estabilidad térmica y qué efectos provocan en la proteína, siendo una de éstas SDM [53], el cual predice los cambios en la estabilidad esperados sobre la mutación en base a la consideración de información estructural y su secuencia, dado el uso de Tablas Específicas de entornos de sustitución (ESSTs). Por otro lado, se han desarrollado softwares que facilitan el estudio de mutaciones, mediante la evaluación de información filogenética y los cambios en la secuencia que se han observado en una familia de proteínas y una proteína de interés. Este es el caso de MOSST [54], el cual estima mediante cálculos matemáticos la propensión de mutaciones en la proteína a estudiar, con respecto a la familia de contraste. Otro de los software de interés que han sido desarrollados para la evaluación de mutaciones en VHL, según el riesgo de ésta es el expuesto en [45], en el cual, por medio de algoritmo Random Forest, se propone un sistema de clasificación de mutaciones según el riesgo que éstas presentan.

Dado a lo anterior, se ha denotado la falta de sistemas de clasificación de riesgo de mutaciones detectadas en VHL, las cuales han sido reportadas pero se desconoce si representarán un riesgo o no para el paciente. En base a esto y a los sistemas que se han desarrollado para trabajar con dichas mutaciones, se propone VHL-Hunter.

VHL-Hunter, es un servicio web de clasificación, el cual recibe una mutación puntual o una lista de ellas y evalúa si dicha mutación presenta una relevancia clínica. Un esquema representativo del flujo de trabajo de VHL-Hunter es como se expone en la Figura 1.2.

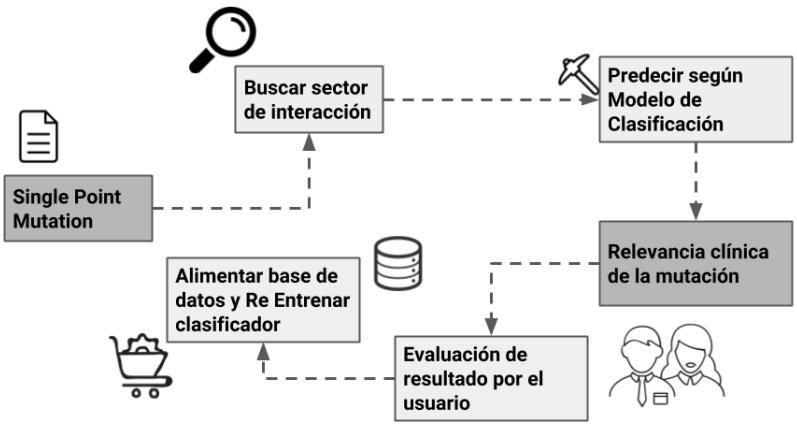

Figura 1.2: Flujo de Trabajo de VHL-Hunter.

Tal como se expone en la Figura 1.2, VHL-Hunter trabaja con mutuaciones puntuales como dato de entrada, el cual se compone por: Residuo original, Residuo Mutuado y la posición, en base a esto, la herramienta evalúa a qué sector de interacción pertenece el residuo y selecciona el modelo de clasificación a utilizar, se predice la relevancia clínica y se reporta el resultado al usuario.

El usuario tiene la opción de reportar si la clasificación generada es efectivamente correcta o no, según el conocimiento que éste disponga, dado esto, la mutuación puntual pasa a ser parte del sistema de almacenamiento persistente, se entrena nuevamente el modelo y se actualizan los datos asociados.

La selección del sector de interacción, hace referencia a información topológica que asocia la herramienta, en la cual se evalúan los sectores que interactúan con otras proteínas formando interacciones Proteína-Proteína. Se han reportado 13 sectores de interacción, razón por la cual, se utilizan como sectores independientes formando set de datos únicos, lo cual implica que son modelos de clasificación con algoritmos, parámetros y medidas de desempeño diferentes. Detalles los cuales serán explicados en la sección de Metodología.

Un punto importante a destacar es que, VHL-Hunter trabaja con información estructural y de estabilidad, en base a los resultados de predicción que entrega SDM [53] y adición de información filogenética, según las propensiones calculadas por MOSST [54], adicional a la información toplógica que implica el conocimiento de los sectores de interacción Proteína-Proteína que presenta VHL.

Finalmente, el hecho de permitir al usuario reportar si la mutación fue clasificada correctamente, implica que los modelos de clasificación se alimentarán con nueva información constantemente, adicionando nuevos elementos a la base de datos, y permitiendo las actualizaciones de los modelos, en base a la nueva data. Generando impactos en las medidas de desempeño que estos posean y en las técnicas de validación que se usen para dicha instancia.

1.2. Marco Teórico

1.2.1. Minería de Datos

Minería de datos es el proceso de descubrimiento de patrones en set de datos, involucrando métodos asociados a Machine Learning, Estadísticas y sistemas de bases de datos. [33]. La minería de datos es un subcampo interdisciplinario de la informática, el cual tiene por objetivo general extraer información (a través de métodos inteligentes) de un conjunto de datos y transformar la información en una estructura comprensible para su uso posterior. [34, 35]. La minería de datos es el paso de análisis del proceso de *descubrimiento*

de conocimiento en bases de datos, o KDD. [36]. Además del análisis en bruto de los datos, también incluye aspectos de manipulación de bases de datos y pre procesamiento de datos, evaluaciones de modelo e inferencia, métricas de interés, consideraciones de complejidad, post procesamiento de estructuras descubiertas, visualización y actualización de la información.

En la Figura 1.3, se exponen las principales ramas que componen la minería de datos y los diferentes procesos que se asocian a dichas ramas.

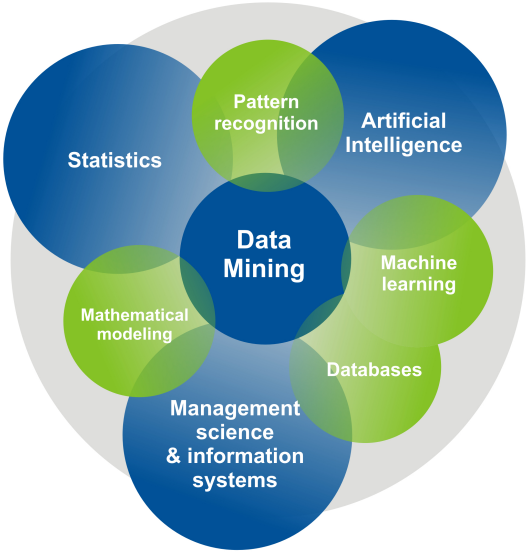

Figura 1.3: Componentes en la minería de datos

Son tres las principales áreas que abarca la minería de datos: Estadística, Inteligencia Artificial y Manipulación de sistemas de información, mientras que son distintos procesos los que interactúan entre estas ramas, tales como: Modelamiento Matemático, reconocimiento de patrones, Sistemas de almacenamiento persistente y machine learning.

Cada área en particular tiene un objetivo general y diversos objetivos específicos. Sin embargo, estas áreas interactúan entre sí, con el fin de poder extraer patrones de información que generen conocimientos a partir de la data de procesada.

La minería de datos se utiliza en diferentes campos, tales como: Genética, Evaluaciones proteómicas, Comercio, Sistemas de tránsito, Optimizaciones en procesos industriales, reconocimiento de patrones y rasgos cuantificables en enfermedades y más recientemente en áreas de dinámicas moleculares y parámetros para la generación de pipe lines automatizados de simulaciones cuánticas en sistemas químicos.

1.2.2. Aprendizaje de Máquinas

Aprendizaje de Máquina, es una rama de la inteligencia artificial que tiene por objetivo el desarrollo de técnicas que permitan a los computadores aprender, es decir, generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos.

La técnica es aplicada en: motores de búsqueda, diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, etc.

Tiene como resultado un modelo para resolver una tarea dada. Entre los que se distinguen:

- Modelos geométricos: basados en espacio de instancias, siendo de múltiples dimensiones.
- Modelos probabilísticos: basados en la detección de la distribución de probabilidades asociada a una función de los valores.
- Modelos lógicos: expresan las probabilidades en reglas organizadas en forma de árboles de decisión.

A continuación se abordarán temáticas relacionadas con el procesamiento de los datos, seguido a su vez de explicación de algoritmos de aprendizaje de máquinas, tanto supervisados como no supervisados, debido a que juegan un rol fundamental a la hora de dar cumplimiento a los objetivos de la memoria de título.

Pre Procesamiento de Datos

PCA

Análisis de Componentes Principales (PCA por sus iniciales en inglés), es una técnica estadística que permite la conversión de un conjunto de variables posiblemente correlacionadas a un conjunto de variables no correlacionadas linealmente, los cuales se denominan componentes principales, siendo la cantidad menor o igual que las variables originales, donde la principal característica de los componentes principales es que son ordenados en base a la varianza que entregan a los datos, así el primer componente principal aporta una mayor varianza que el segundo y así sucesivamente, basándose principalmente en el uso de vectores propios.

PCA se utiliza principalmente como una herramienta en el análisis de la exploración de datos los cuales tienen como objetivo generar modelos de predicción y sus resultados normalmente se exponen en puntuaciones que tienen estrecha relación con el aporte de varianza que estos entregan [3].

Intuitivamente es posible pensar el PCA como un elipsoide n-dimensional de datos, donde cada eje del elipsoide representa un componente principal, esto implica que si algún eje del elipsoide es pequeño, la varianza correspondiente a lo largo de éste también lo es, por lo que omitir dicho eje no implica una pérdida importante de información, esto último es denotado como la reducción de la dimensionalidad en base a los aportes a las varianzas que denotan cada componente.

Definición

Matemáticamente, es posible definir PCA como una transformación lineal ortogonal que transforma los datos a un nuevo sistema de coordenadas tal que la mayor varianza por alguna proyección de los datos pasa a situarse en la primera coordenada (llamado el primer componente principal), la segunda mayor varianza en la segunda coordenada, y así sucesivamente [4].

Se considera un conjunto de datos, $\mathbf{X}$, con una media empírica 0, donde cada una de las filas ($\mathbf{n}$) representan ejemplos y las columnas características o atributos ($\mathbf{p}$).

La transformación está definida por un set de vectores de dimensión $\mathbf{p}$ que poseen pesos denotados por $w_{(k)} = (w_1, ..., w_p)_{(k)}$, los cuales para cada vector x_i en X se operan para dar un vector con los componentes principales $t_{(i)} = (t_1, ..., t_k)_{(i)}$ el cual viene dado por $t_{k(i)} = x_i w_k$

De tal manera que las variables individuales de $\mathbf{t}$ considerado sobre el conjunto de datos sucesivamente heredan la varianza máxima posible de $\mathbf{x}$, con cada carga del vector $\mathbf{w}$.

El primer componente w_1 tiene que satisfacer las siguientes características:

- $w_{(1)} = \arg\max_{||w||=1}\{\sum_i (t_{(1)})_i^2\} = \arg\max_{||w||=1}\{\sum_i (x_{(i)} * w)^2\}$
- $w_{(1)} = \arg\max_{||w||=1}\{||Xw||^2\} = \arg\max_{||w||=1}\{w^T X^T X w\}$
- $w_{(1)} = \arg\max\{\frac{w^T X^T X w}{w^T w}\}$

Los k restantes componentes son encontrados efectuando la extracción de los primeros *k-1* componentes principales desde $\mathbf{x}$:

$\hat{x}_k = x - \sum_{\delta=1}^{k-1} X w_{(\delta)} w_{(\delta)}^T$

A su vez, para encontrar el vector de carga, es necesario extraer la varianza máxima del nuevo set de datos, tal que:

$w_{(k)} = \arg\max_{||w||=1}\{||\hat{x}_k w||^2\} = \arg\max\{\frac{w^T X^T \hat{X}_k^T \hat{x}_k w}{w^T w}\}$

La matriz de covarianza juega un rol fundamental en este análisis, cuyo valor entre dos componentes principales viene dado por:

$Q(PC_j, PC_k) \propto (Xw_{(j)})^T * (Xw_{(k)})$
$Q(PC_j, PC_k) = w_{(j)}^T X^T X w_{(k)}$
$Q(PC_j, PC_k) = w_{(j)}^T \lambda_{(k)} w_{(k)}$
$Q(PC_j, PC_k) = \lambda_{(k)} w_{(j)}^T w_{(k)}$

La principal característica que define al PCA es que es una técnica comúnmente utilizada para la reducción de la dimensionalidad, esto viene dado por la transformación que se genera, $T = xw$ donde cada vector $x_{(i)}$ existente en un espacio de coordenadas de variables p, es representado por un nuevo espacio en el cual las variables no se encuentran correlacionadas, sin embargo, si se utilizan L componentes principales para así utilizar los primeros L vectores de carga se obtiene una transformación truncada $T_L = XW_L$, de tal manera que la matriz T_L posee los n ejemplos originales. Sin embargo sólo posee L características que definen el set de datos, de tal manera que dicha transformación es posible expresarla como:

$t = W^T x$, donde $x \in R^p, t \in R^L$, para las cuales las columnas pxL de la matriz W forman una base ortogonal de las L características, de esta manera, al basarse en la construcción con sólo L columnas se maximiza la varianza original de los datos y se minimiza el error cuadrático tal que:

$||TW^T - T_L W_L^T||_2^2 = ||X - X_L||_2^2$

Normalmente esta reducción es usada para el manejo de set de datos de alta dimensionalidad [21].

Propiedades

Existen tres propiedades claras que facilitan la comprensión y la utilidad del PCA como medio de manejo de dimensionalidades, y transformación de datos a espacios con coordenadas similares no correlacionados linealmente, éstas son:

- Para cualquier entero $q, 1 \leq q \leq p$, se considera la transformación lineal $y = B'x$, donde y es un elemento del vector q y B es una matriz de (q x p) y la matriz de covarianza para y se puede denotar por $\sum_y = B' \sum B$, entonces la traza de $\sum_y$ es maximizada tomando $B = A_q$ donde A_q es la primera columna q de A.

- Considerando la transformación ortogonal $y = B'x$, la traza de $\sum_y$ es minimizada tomando $B = A_q^*$ donde A_q^* es la última columna q de A.
- Sea la descomposición espectral de $\sum = \lambda_1\alpha_1\alpha_1' + ... + \lambda_p\alpha_p\alpha_p'$, se tiene que $Var(x_j) = \sum_{k=1}^{p} \lambda_k\alpha_{kj}^2$.

En resumen, es posible denotar que todas las operaciones de los datos son centradas en cero, tal que $\frac{1}{N}\sum_{i=1}^{N} x_i = 0$, para lo cual se realizan tratamientos sobre la matriz de covarianza, la cual se puede exponer como: $C = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$, con el fin de obtener los vectores propios, denotados como **eigen descomposition**: $\lambda v = Cv$, los cuales pueden ser reescritos como $\lambda x_i^T = x_i^T Cv \ \forall i \in [1, N]$, lo cual genera la transformación lineal en vectores con menor dimensionalidad expuestos por sus aportes de varianzas en los datos, existentes en espacios con coordenadas similares, sin correlación lineal entre ellos.

Algoritmos de Aprendizaje en Minería de Datos

Los algoritmos de Aprendizaje pueden ser definidos como una serie de pasos que permitan cumplir una tarea dada, el uso del cual dependerá de las características que posea dicha tarea [6].

Los algoritmos de aprendizaje pueden clasificarse en dos grandes grupos:

- **Supervisados**: se cumple un rol de predicción, clasificación, asignación, etc. a un conjunto de elementos con características similares, por lo que los datos de entrada son conocidos.
- **No Supervisados**: su objetivo es agrupar en conjuntos con características similares los elementos de entrada dado los valores de estos atributos, en base a la asociación de patrones característicos que representen sus comportamientos.

A continuación se hace referencia a los diferentes tipos de algoritmos existentes, considerando la clasificación expuesta anteriormente.

Algoritmos de Aprendizaje Supervisado

Es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten en pares de objetos (normalmente vectores): una componente del par son los datos de entrada y el otro, los resultados deseados. La salida de la función puede ser un valor numérico (como en los problemas de regresión) o una etiqueta de clase (como en los de clasificación).

El objetivo del aprendizaje supervisado es crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente.

Existe una gran variedad de algoritmos de aprendizaje supervisado, dentro de los cuales destacan:

k Nearest Neighbors

Algoritmo de aprendizaje supervisado, el cual tiene por objetivo asociar un elemento a una clase en particular, dada la información de ejemplos de entrada que tengan asociadas características particulares, que puedan declararse como *vecinos* del nuevo ejemplo a clasificar, siendo **k** el número de vecinos que se está dispuesto a utilizar para aplicar la clasificación.

Con el fin de evaluar la cercanía de los ejemplos existentes contra el nuevo ejemplo a clasificar es necesario asociar ciertas medidas de distancia que permitan cuantificar esta característica, para así poder comparar esta distancia y evaluar la cercanía para asociarle una clase a este nuevo ejemplo.

Presenta algunos problemas, tales como: posibles errores al existir más de un elemento de distinta clase cercano al nuevo ejemplo a clasificar, sin embargo, dicho error estimado es reducido.

La distancia a emplear para evaluar la cercanía puede ser: Euclidiana, Manhattan, coseno, Mahalanobis[1], entre las principales.

La mejor elección de **k** depende fundamentalmente de los datos; generalmente, valores grandes de **k** reducen el efecto de ruido en la clasificación, pero crean límites entre clases parecidas.

Existen dos variaciones para la aplicación de KNN: aplicación basada en las distancias y aplicación basada en radios con respecto a puntos, la primera es mayormente usada, no obstante, en el caso de que los puntos no se encuentren uniformemente distribuidos es una mejor opción usar la segunda alternativa, siendo muy eficaz en problemas conocidos como *la maldición de la dimensionalidad*.

KNN utiliza el componente de peso, es decir, valores asignados a puntos específicos para determinar si un elemento a clasificar es de una clase o no, normalmente se utilizan pesos uniformes. Sin embargo, es posible asignar valores de tal manera que al momento de realizar la votación puntos más cercanos en base a distancias presenten más peso que otros.

[1]Se explican en detalle en la sección 1.2.2

Se han implementando diversos algoritmos a la hora de aplicar la técnica de KNN, los cuales tienen relación con el coste computacional que presentan, dentro de estos se encuentran: Brute Force, K-D Tree y Ball Tree [7].

Naive Bayes

Naive Bayes es un conjunto de algoritmos de aprendizaje supervisados basados en la aplicación del teorema de Bayes con la suposición *ingenua* de independencia entre cada par de características [8]. Dada una variable de clase y y un vector de característica dependientes de la forma $x_1, ..., x_n$, el teorema de Bayes establece la siguiente relación:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1,...x_n|y)}{P(x_1,...,x_n)}$$

Utilizando la suposición ingenua de independencia de características, se tiene que:

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y)$$

Para todo i, esta relación se simplifica a:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,...,x_n)}$$

Dado que $P(x_1, \ldots, x_n)$ es constante dada la entrada, se puede utilizar la siguiente regla de clasificación:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg \text{máx}_y \, P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

A pesar de sus supuestos aparentemente simplificados, los clasificadores de Naive Bayes han funcionado bastante bien en muchas situaciones del mundo real, la famosa clasificación de documentos y el filtrado de spam son ejemplos de ello. Requieren una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios. Pueden ser extremadamente rápido en comparación con métodos más sofisticados. El desacoplamiento de las distribuciones de las características condicionales de clase significa que cada distribución se puede estimar de forma independiente como una distribución unidimensional. Esto a su vez ayuda a aliviar los problemas derivados de la dimensionalidad.

Existen distintos tipos de clasificadores de Naive Bayes, diferenciándose entre sí en la función de distribución de probabilidad que utilizan, dentro de los que se encuentran:

- Gaussian Naive Bayes.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

- Multinomial Naive Bayes.

  La distribución se parametriza por el vector $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ para cada clase y, donde n es el número de características y θ_{y1} es la probabilidad $P(x_i \mid y)$ de que la característica i aparezca en una muestra que pertenece a la clase y.

  Cada θ_y es estimado por:

$$\hat{\theta}_{yi} = \frac{N_{yi}+\alpha}{N_y+\alpha n}$$

  Donde $N_{yi} = \sum_{x \in T} x_i$ es el número de veces que aparece la característica i en la muestra de clase y en el set de entrenamiento T y $N_y = \sum_{i=1}^{|T|} N_{yi}$ representa el total de todas las características para la clase.

- Bernoulli Naive Bayes.

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

Árboles de Decisión

Se define árbol de decisión como un modelo de predicción utilizado en el ámbito de la inteligencia artificial, en el cual dado un conjunto de datos se fabrican diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

El aprendizaje basado en árboles de decisión utiliza un árbol como un modelo predictivo que mapea las observaciones de las características que presenta un elemento [9].

En estas estructuras de árbol, las hojas representan etiquetas de conjuntos ya clasificados, los nodos, a su vez, nombres o identificadores de los atributos y las ramas representan posibles valores para dichos atributos.

Aprendizaje basado en árboles de decisión es un método comúnmente utilizado en la minería de datos, cuyo objetivo consiste en desarrollar un modelo de predicción para el valor de una variable de destino en función de diversas variables de entrada.

Un árbol de decisión es una representación simple para clasificar ejemplos, el aprendizaje basado en esta metodología es una de las técnicas más eficientes para la

clasificación supervisada. Donde cada ejemplo consta de atributos con valores discretos dentro de un dominio de conjunto finito, y existe un sólo término final denominado clasificación.

En un árbol de decisión, cada elemento del dominio de la clasificación se llama clase, cada nodo interno (no hoja) está etiquetado con una función de entrada, las ramas procedentes de un nodo etiquetado con una característica están asociados con cada uno de los posibles valores de la característica. Cada hoja del árbol se marca con una clase o una distribución de probabilidad sobre las clases.

Un árbol puede ser entrenado mediante el fraccionamiento del conjunto inicial en subconjuntos basados en una prueba de valor de atributo. Este proceso se repite en cada subconjunto derivado de una manera recursiva llamada particionamiento recursivo. La recursividad termina cuando el subconjunto en un nodo tienen todos el mismo valor de la variable objetivo, o cuando la partición ya no agrega valor a las predicciones.

Para cada división es necesario el uso de una función que entregue una medida de impureza en cada división, esto con el objetivo de seleccionar la mejor partición para un atributo dado, la elección de dicho atributo se basa en el objetivo de separar de mejor manera los ejemplos.

La selección de los atributos se basa en qué atributo al momento de clasificar genera nodos más puros, para ello se utiliza una función de ganancia de información, la cual representa la ganancia otenida a partir de una división de los ejemplos de entrenamiento. Dicha función puede ser expuesta como sigue:

$$\Phi(D,t) = I(t) - \sum_{i=1}^{l} I(t_i)P_i \tag{1.1}$$

Donde:

- $I(t)$ representa la Medida de Impureza asociada al nodo **t**, desde el cual se comenzará a realizar la partición o nodo padre.
- $\sum_{i=1}^{l} I(t_i)$ representa la suma ponderada de las impurezas de los nodos hijos t_i generados a partir de una división **D**.
- P_i representa la proporción de ejemplos que siguen la rama **i** asociada a la división **D**.

Dentro de las medidas de impureza, existen:

- Gini Index = $\sum\ p_i\ x\ (1 - p_i)$
- Entropía = $-\sum\ p_i\ x\ \log_2(p_i)$

Siendo la más utilizada la medida de Entropia.

En ambas, p_i corresponde a la proporción de ejemplos asociados a cada una de las clases, presentes en el nodo evaluado.

El algoritmo para el cual se entrena y se clasifica un nuevo ejemplo es el que sigue:

- Partir desde un nodo inicial o padre.
- Seleccionar el mejor atributo que divide de una manera óptima los ejemplos, lo cual se observa por medio de la función de ganancia de información.
- Se clasifica los ejemplos del conjunto de entrenamiento de un nodo entre sus descendientes.
- El proceso finaliza si los ejemplos del conjunto de entrenamiento quedan perfectamente clasificados, esto ocurre en dos casos: todos los ejemplos pertenecen a una misma clase o se llega a una hoja.
- En el caso de no cumplirse lo del punto anterior, se itera para cada rama de manera recursiva, utilizando sólo los ejemplos que llegan a esa rama.

SVM

Support Vector Machine (SVM), también conocido como redes de soporte vectorial, son modelos de aprendizaje supervisados asociados al análisis de los datos utilizados para la clasificación. Dado un conjunto de ejemplos de entrenamiento, cada uno marcado como perteneciente a una u otra de las dos categorías, un algoritmo de entrenamiento de SVM construye un modelo que asigna nuevos ejemplos a una categoría u otra, convirtiéndolo en un clasificador binario lineal no probabilístico. Un modelo SVM es una representación de los ejemplos como puntos en el espacio, mapeados de modo que los ejemplos de las categorías separadas se dividan por un espacio claro que es tan amplio como sea posible. Nuevos ejemplos son entonces mapeados en ese mismo espacio y predicen si pertenecen a una categoría en base a qué lado del espacio son asignados [10].

SVM puede realizar eficientemente una clasificación no lineal utilizando funciones kernel, con el fin de generar transformaciones de espacio dimensional de los datos, para mapear implícitamente sus entradas en espacios característicos de alta dimensión.

Las ventajas de máquinas de soporte vectorial son:

- Efectivo en espacios de dimensiones altas.
- Efectivo aún en casos donde el número de dimensiones es mayor que el número de muestras.
- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamada vectores de soporte), por lo que también es memoria eficiente.
- Versátil: diferentes funciones del núcleo pueden ser especificadas para la función de decisión. Se proporcionan núcleos comunes, pero también es posible especificar núcleos personalizados.

Las desventajas de las máquinas de soporte vectorial incluyen:

- Si el número de características es mucho mayor que el número de muestras, es probable que el método tenga un mal desempeño.
- SVMs no proporciona directamente estimaciones de probabilidad, estos se calculan utilizando cinco veces una costosa validación cruzada.

Existen diversas variaciones de SVM, tales como: SVC, NuSVC y LinearSVC, los cuales son capaces de realizar una clasificación multiclase[2] en un conjunto de datos, es decir, ya no depender de un clasificador único para dos clases.

SVC presenta una aplicación basada en libsvm[3]. La complejidad del tiempo de ajuste se hace cuadrática con el número de muestras, lo que dificulta escalar a conjunto de datos con tamaño mayor a 10000 [11]. El apoyo multiclase es manejado según un esquema de uno vs uno.

Por otro lado NuSVC presenta características similares a SVC, pero, utiliza un parámetro para controlar el número de vectores de soporte. La aplicación se basa en libsvm.

LinearSVC es similar a SVC pero, se utiliza una función de kernel lineal, además es implementado en términos de liblinear en lugar de libsvm, por lo que tiene más flexibilidad en la elección de las penalizaciones y las funciones de pérdida y debería escalar mejor a un gran número de muestras. Esta clase soporta entradas densas y escasas y el soporte de multiclase se maneja de acuerdo con un esquema de uno contra el resto.

Cada uno de los clasificadores expuestos en los puntos anteriores toman como entrada el set de entramiento y las etiquetas asociadas a las clases, con el fin de generar tanto el testeo

[2]Implica la existencia de un número de clases mayor a dos

[3]Librería implementada para el desarrollo de máquina soporte de vectores, https://www.csie.ntu.edu.tw/ cjlin/libsvm/

como la validación del modelo, previo etapa de entrenamiento, la principal característica es que se utilizan vectores de apoyo para el set de entrenamiento, los que son denominados vectores de soporte, normalmente se utilizan funciones kernel para la obtención de estos vectores de soporte.

SVC y NuSVC implementan el enfoque *uno contra uno* para la clasificación multiclase. Si existen n clases, se construyen $\frac{n*(n-1)}{2}$ clasificadores, de los cuales cada uno forma un set datos de dos clases; por otro lado, LinearSVC implementa una estrategia multi-clase *uno contra el resto*, formando así modelos de n clases, los cuales son entrenados n veces. Si sólo hay dos clases, sólo se entrena un modelo.

Los algoritmos SVM están asociados a diversos problemas, sin embargo el principal radica en el desbalance de clases, ya sea por el número que presentan o por el peso asociado a éstas, tal como se expone en la Figura 1.4:

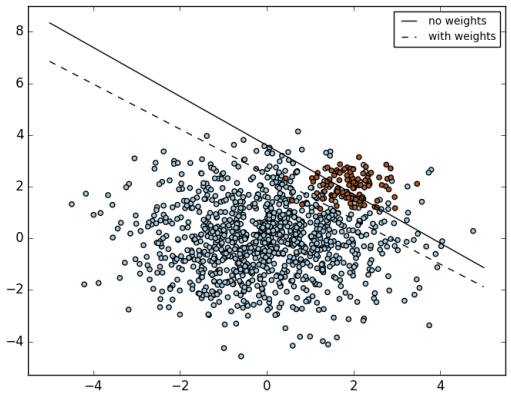

Figura 1.4: Muestra de desbalance de clases en SVM.

Complejidad

Las máquinas de soporte vectorial son herramientas poderosas, pero sus requerimientos de computación y almacenamiento aumentan rápidamente con el número de vectores de entrenamiento. El kernel de un SVM es un problema de programación cuadrática (QP), separando los vectores de soporte del resto de los datos de entrenamiento. Es posible eslacar esta solución entre $O(n_{features} \times n_{samples}^2)$ y $O(n_{features} \times n_{samples}^3)$ [9].

Formulación Matemática

SVM construye un hiperplano o conjunto de hiperplanos en un espacio dimensional

alto o infinito, que puede usarse para clasificación, regresión u otras tareas. Intuitivamente, se logra una buena separación por el hiperplano que tiene la mayor distancia a los puntos de datos de entrenamiento más próximos de cualquier clase (llamado margen funcional), ya que en general, cuanto mayor es el margen, menor es el error de generalización del clasificador, tal como se expone en la Figura 1.5:

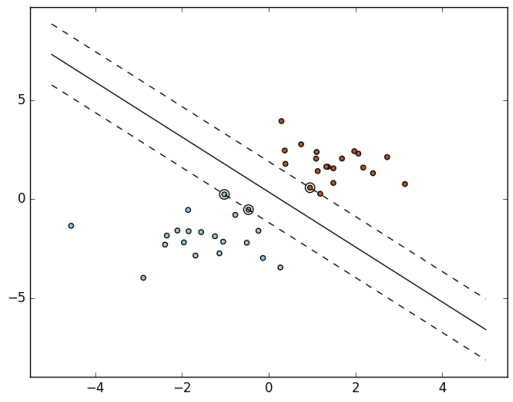

Figura 1.5: Esquema de hiperplanos en SVM.

Se expone la formulación matemática para cada uno de los clasificadores expuestos anteriormente:

SVC

Dado los vectores de entrenamiento $x_i \in R^p$, i=1,..., n, en dos clases, y un vector, $y \in \{1, -1\}^n$, SVC resuelve el siguiente problema primario:

$$\text{mín}_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$

$$para\ la\ clase\ y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,\ \zeta_i \geq 0, i = 1, ..., n$$

Su doble es

$$\text{mín}_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

$$para\ la\ clase\ y^T \alpha = 0\ 0 \leq \alpha_i \leq C, i = 1, ..., n$$

Donde e es el vector de todos los unos, $C > 0$ es el límite superior, Q es una matriz de $n\ x\ n$ definida semipositiva, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, donde $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ es el kernel. Los vectores de entrenamiento son implícitamente mapeados en un espacio dimensional mayor (tal vez infinito) por la función ϕ.

La función de decisión es:

$$sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \rho)$$

NuSVC

Se introduce el parámetro ν el cual controla el número de vectores de soporte y errores de entrenamiento. El parámetro $\nu \in (0, 1]$ es un límite superior en la fracción de errores de entrenamiento y un límite inferior de la fracción de vectores de soporte.

SVR

Dados los vectores de entrenamiento $x_i \in R^n$ ε-SVR resuelve el siguiente problema primario:

$$\text{mín}_{w,b,\zeta,\zeta^*} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\zeta_i + \zeta_i^*)$$

$$paralaclasey_i - w^T \phi(x_i) - b \leq \varepsilon + \zeta_i,$$

$$w^T \phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*,\ \zeta_i, \zeta_i^* \geq 0, i = 1, ..., n$$

Donde e es el vector para todos, $C > 0$ es el límite superior, Q es una matriz de n x n definida semipositiva, $Q_{ij} \equiv K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ es el kernel. Aquí los vectores de entrenamiento son implícitamente mapeados en un espacio dimensional mayor (tal vez infinito) por la función ϕ.

La función de decisión es:

$$\sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + \rho$$

Medidas de Evaluación de Modelos

Medir el desempeño del modelo predictivo es importante a la hora de evalua qué tan efectivo es el entrenamiento y la clasificación que se genera, existen medidas que sólo se basan en la cantidad de aciertos o errores que comente el clasificador, otras que implican la eficiencia del modelo y otras que se basan en la precisión, se define brevemente algunas de las performance utilizadas a la hora de evaluar modelos de aprendizajes supervisados:

- **Tasa de Verdaderos Positivos**: corresponde a la medida asociada a las correctas clasificaciones versus el total de clasificaciones realizadas, es decir, cuántos predicciones efectivas se obtuvieron con respecto a una clase.

- **Tasa de Falsos Positivos**: corresponde a la medida asociada a las clasificaciones mal efectuadas, es decir, cuántas predicciones erradas existen con respecto a una clase.
- **Eficiencia Global**: corresponde al total de predicciones correctamente efectuadas con respecto al total de ejemplos existentes en la muestra.
- **Especificidad**: corresponde a la probabilidad asociada a clasificar de manera negativa un ejemplo positivo.
- **Precisión**: corresponde a la probabilidad asociada a predecir un ejemplo de manera correcta, con respecto a la predicción.
- **Alcance**: corresponde a la probabilidad asociada a predecir un ejemplo de manera correcta, con respecto a la realidad.
- **Medida F (F-measure)**: es una medida de combinación entre la precisión y el alcance, utilizada como media entre los dos evaluadores.

Métodos de Validación de Aprendizajes Supervisados

Validación Cruzada

La validación cruzada, a veces llamada estimación de la rotación, es una técnica de validación del modelo para evaluar cómo los resultados de un análisis estadístico se generalizarán a un conjunto de datos independiente. Se utiliza principalmente en entornos donde la meta es la predicción, y se quiere estimar la precisión con la que un modelo predictivo se llevará a cabo en la práctica. En un problema de predicción, a un modelo se le suele asignar un conjunto de datos, de los datos conocidos sobre los que se ejecuta el entrenamiento (conjunto de datos de formación) y un conjunto de datos desconocidos (o primeros datos) contra los que se prueba el modelo. El objetivo de la validación cruzada es definir un conjunto de datos para *probar* el modelo en la fase de entrenamiento (es decir, el conjunto de datos de validación), con el fin de limitar problemas como sobre ajuste.

La idea es dividir el set de datos totales abarcando un set de entrenamiento y un set de validación, lo cual se puede explicar en la Figura 1.6:

Una ronda de validación cruzada implica dividir una muestra de datos en subconjuntos complementarios, realizar el análisis en un subconjunto (denominado conjunto de entrenamiento) y validar el análisis en el otro subconjunto (denominado conjunto de validación o conjunto de pruebas). Para reducir la variabilidad, varias rondas de validación

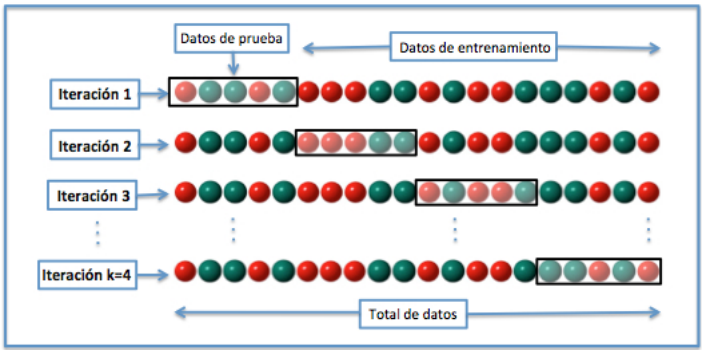

Figura 1.6: Esquema representativo de validación cruzada.

cruzada se realizan utilizando diferentes particiones, y los resultados de validación se promedian durante las rondas, siendo las más utilizadas *10-fold cross validation*.

Leave one out (Dejar uno)

Es un tipo especial de validación cruzada, en donde se tiene una muestra con n ejemplos en la etapa de entrenamiento se subdivide dicho set de datos considerando $n-1$ elementos, de tal manera que 1 no se considera, la idea en particular radica en entrenar con los $n-1$ ejemplos y validar o testear con el ejemplo restante, esto se itera n veces, tal como se expone en 1.7, implicando una mayor cantidad de iteraciones que validación cruzada, provocando además un mayor coste computacional.

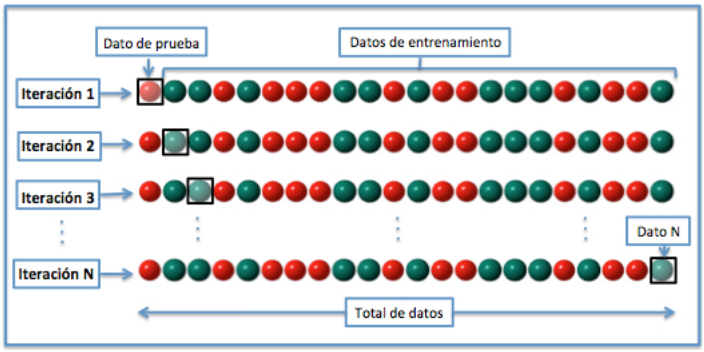

Figura 1.7: Esquema representativo de Leave One.

Principales problemas en Aprendizaje Supervisado

Dentro de los principales problemas que pueden presentar los modelos de aprendizaje supervisado se encuentran las situaciones en las que la cantidad de atributos que puede presentar un set de datos es muy mayor con respecto a la cantidad de ejemplos que se

posee, es decir si existen n ejemplos y la cantidad de atributos es n x n es posible que ocurra dicha problemática, para solucionar este problema, existen técnicas como PCA que permiten la reducción de atributos en base al aporte que entrega a la varianza total de la muestra. Otro posible problema que se puede denotar es el sobreajuste, esto quiere decir, que el modelo es extremadamente complejo, por lo que éste se ajusta muy bien al set de entrenamiento, no obstante a la hora de probar con nuevos set de datos no representa la performance obtenida.

Algoritmos de Aprendizaje No Supervisado

Es un método de Aprendizaje Automático donde un modelo es ajustado a las observaciones. Se caracteriza por el hecho de que no hay un conocimiento a priori. En el aprendizaje no supervisado, un conjunto de datos de objetos de entrada es tratado como un conjunto de variables aleatorias, siendo construido un modelo de densidad para el conjunto de datos.

Existen diversas formas de ver los algoritmos supervisados, ya sea en forma de reglas de asociación o como algoritmos propiamente tal, siendo explicadas ambas en los siguientes puntos.

Reglas de Asociación

Las reglas de asociación son un tipo de modelo asociado a aprendizaje no supervisado, el cual tiene por objetivo encontrar patrones de comportamiento de ejemplos, a partir de relaciones o asociaciones entre los atributos que caracterizan a estos, lo cual depende o está en función de la aparición de ciertos valores de uno, dos o más atributos.

Las reglas son del tipo *Si ocurre este evento y este atributo posee este valor, ocurre este acontecimiento*. Tiene una similitud a las reglas de decisión, siendo las reglas asociadas a atributos del tipo nominal, normalmente las reglas expresan combinaciones de valores de los atributos que suceden más frecuentemente.

Dentro de las aplicaciones se exponen:

- Asociar qué productos la gente compra habitualmente.
- Búsqueda de patrones en Internet.
- Bioinformática.
- En general, tareas asociadas a grandes volúmenes de datos.

Clusterización

La clusterización (clustering) se define como un método de aprendizaje no supervisado, en el cual se cuenta con un conjunto de datos que representan a una muestra y en base a dicha muestra de datos, se trata de obtener grupos de objetos, denominados clusters.

Los clusters deben cumplir con dos características fundamentales:

- Los objetos que pertenezcan a un mismo clúster deben ser bastante homogéneos entre ellos.
- Entre los clústers debe existir un alto grado de heterogeneidad.

En la clusterización se trata, fundamentalmente, de resolver el siguiente problema: Dado un conjunto de N individuos, caracterizados por la información de n variables X_j con j entre $1, .., n$, se plantea el reto de ser capaces de agruparlos de manera que los individuos pertenecientes a un grupo (cluster) (y siempre con respecto a la información disponible) sean tan similares entre sí como sea posible, siendo los distintos grupos entre ellos tan disimilares como sea posible.

Básicamente, el análisis constará de un algoritmo de clusterización que permitirá la obtención de una o varias particiones, de acuerdo con los criterios establecidos.

Criterios de Similitud

Tal como se ha mencionado anteriormente, el hecho de tener elementos pertenecientes a un mismo grupo bastante similares entre ellos y divergentes entre distintos clúster, es la característica primordial a tratar, esto último radica en la importancia de las variables que componen a un elemento en particular y en los valores que tomen éstas.

Por lo tanto se debe determinar que tan similares o diferentes son los valores que tomen las variables con respecto al elemento al cual pertenece.

Para medir lo similar o disimilar que son los individuos existe una enorme cantidad de índices de similaridad y de disimilaridad o divergencia. Todos ellos tienen propiedades y utilidades distintas y habrá que ser consciente de ellas para su correcta aplicación al momento de hacer uso de una de ellas.

Los índices expuestos normalmente pueden ser clasificados tal como sigue:

- Indicadores basados en la distancia, para lo cual se considera a los individuos como vectores en el espacio de las variables, en este sentido un elevado valor de la distancia entre dos individuos indicará un alto grado de disimilaridad entre ellos.

- Indicadores basados en coeficientes de correlación; la correlación permite indicar cuál es la fuerza y la dirección de una relación lineal entre dos variables. Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores de la otra, esto es: si se tiene dos variables (A y B) existe correlación si al aumentar los valores de A lo hacen también los de B y viceversa.

- Indicadores basados en tablas de datos de posesión o no de una serie de atributos, es decir teniendo dos vectores A y B que representan a dos individuos de una población se hace una comparación de los elementos existentes en A y no en B, los elementos existentes en B y no en A además de la intersección entre ellos, es decir, los elementos existentes en A y en B.

Criterios basados en distancias

La distancia o disimilaridad entre dos individuos i y j corresponde a una medida, indicada por $D_{(i,j)}$, que mide el grado de semejanza, entre ambos individuos, en relación a un cierto número de características cuantitativas y/o cualitativas. El valor de $D_{(i,j)}$ es siempre un valor positivo, y cuanto mayor sea este valor, mayor será la diferencia entre los individuos i y j.

Las propiedades que debe cumplir un indicador de distancia son:

- No negatividad, es decir, $D_{(i,j)} > 0$.
- $D_{(i,j)} = 0 \Leftrightarrow i = j$.
- Simetría, es decir, $D_{(i,j)} = D_{(j,i)}$.

Algunos de los tipos de distancias que pueden ser calculados son expuestos a continuación.

Distancia Euclidiana

La distancia euclidiana es la más conocida y más sencilla de comprender, pues su definición coincide con el concepto más común de distancia, es una recta que une dos puntos.

Su expresión es:

$$D_{(X,Y)} = \sqrt{\sum_{i=1}^{l}(X_i - Y_j)^2}$$

La distancia euclidiana tiene dos graves inconvenientes:

- Es una distancia sensible a las unidades de medida de las variables: las diferencias entre los valores de variables medidas con valores altos contribuirán en mucha mayor medida que las diferencias entre los valores de las variables con valores bajos. Como consecuencia de ello, los cambios de escala determinarán, también, cambios en la distancia entre los individuos.
- Si las variables utilizadas están correlacionadas, estas variables darán una información, en gran medida redundante. Parte de las diferencias entre los valores individuales de algunas variables podrían explicarse por las diferencias en otras variables. Como consecuencia de ello la distancia euclidiana inflará la disimilaridad o divergencia entre los individuos.

La distancia euclidiana será, en consecuencia, recomendable cuando las variables sean homogéneas y estén medidas en unidades similares y/o cuando se desconozca la matriz de varianzas.

Distancia Manhattan

Es muy similar a la distancia euclidiana, la diferencia radica en que se aplican distancia en zig-zag de los datos, se representa por: $D_{(X,Y)} = \sum_{i=1} n|X_i - Y_i|$.

Distancia del Coseno

Considera cada ejemplo como un vector de n dimensiones, para los cuales estima el coseno del ángulo que forman, se representa por: $D_{(X,Y)} = \arccos(\frac{X^T Y}{||X||\ ||Y||})$

Distancia Mahalanobis

Esta distancia presenta propiedades que solucionan los inconvenientes de la aplicación de la distancia euclideana: no varía a cambios de escala, por lo que no depende de las unidades de medida, además se consideran las correlaciones entre las variables, debido al uso de una matriz de covarianza entre los datos y se corrige el efecto de la redundancia. Se destaca además que no asume independencia entre los datos.

La distancia Mahalanobis se representa como: $D_{(X,Y)} = \sqrt{(X-Y)^T S^{-1}(X-Y)}$.

Criterios basados en similaridades

Existen indicadores que permiten medir el grado de homogeneidad entre los individuos, diferentes a las distancias expuestas en el punto anterior, conocidos como indicadores de

similitud.

Los indicadores de similitud indican que a medida que aumente su valor, la similitud entre los individuos será mayor.

La gran mayoría de los indicadores de similitud son basados en coeficientes de correlación o de asociación.

El Coeficiente de Correlación de Pearson utiliza preferentemente datos cuantitativos además del algoritmo de distancias mínimas, por otro lado, los Coeficientes de Correlación por rangos de Kendall y Spearman utilizan variables ordinales [12].

Existen otros criterios, tales como el índice binario y el índice de Tanimoto.

Algoritmos de Clustering

Existen diversos algoritmos de clustering, cada uno con características que los diferencian, los cuales, pueden ser aplicados a diversos casos, dependiendo de las características de los datos de entrada, es decir, de la geometría de estos datos, sin embargo, esta representación se basa principalmente en el uso de matrices, donde cada fila representa un ejemplo y cada columna el valor de un atributo o rasgo cualitativo para dicho ejemplo.

k-Medias (k-Means)

El algoritmo k-medias, trata la separación de muestras en n grupos de igual varianza, minimizando el criterio conocido como inercia, lo que se traduce en la suma de los cuadrados dentro de los clúster [13].

La principal característica y deficiencia a la vez, es que se requiere que el número de grupos sea entregado, es decir, se debe entregar el valor de k, así, si se selecciona un valor de $k = 3$, serán tres grupos los que se encontrarán.

Este algoritmo divide un set de N ejemplos X en K particiones distintas denominadas clúster C, cada uno de ellos descrito por la media μ_j de las muestras en el clúster. Esta media es llamada centroide, por lo que en general, k-medias elige sus centroides de tal manera que el principio de inercia sea reducido al mínimo, es decir, que la suma de los cuadrados de los integrantes de un mismo grupo sea mínima, a través de:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_j - \mu_i||^2)$$

Sin embargo, el principio de inercia, o la suma de los cuadrados mínimos entre los integrantes de los clustering, puede sufrir varios inconvenientes:

- Se hace la suposición de que las agrupaciones son convexas e isotrópicas, lo cual no se da siempre, razón por la que responde mal ante a clusters que posean forma alargadas o con formas irregulares.

- No es una métrica normalizada, es decir, se sabe que los valores más bajos son mejores y el cero es óptimo, sin embargo, en espacios de muy de alta dimensionalidad, las distancias euclidianas tienden a ser infladas, lo que se conoce como "la maldición de la dimensionalidad", razón por la cual, son utilizados algoritmos de reducción de la dimensionalidad, tal como PCA.

Ambos puntos, son posibles observarlos en la Figura 1.8, en la cual se exponen, problemas con varianzas distintas, diferencias asociadas al tamaño de los clúster, anisotropía[4] de los datos, etc.

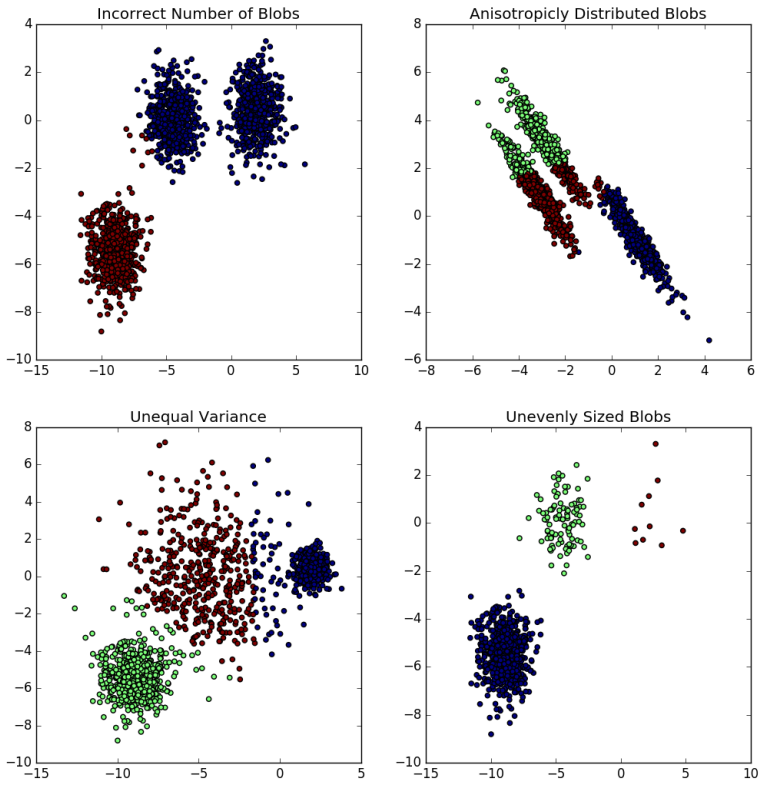

Figura 1.8: Posibles inconvenientes con los datos, donde k-medias no funciona correctamente

k-medias es referido a menudo como el algoritmo de Lloyd. En términos básicos, el algoritmo tiene tres pasos. El primer paso consiste en elegir los centroides iniciales, con

[4]Las variables varían en base a las direcciones en las que se examinan

el método más básico para elegir k muestras del conjunto de datos X. Después de la inicialización, k-medias consta de un bucle entre los otros dos pasos.

El primer paso asigna cada muestra a su centroide más cercano. En el segundo se crean nuevos centroides tomando el valor medio de todas las muestras asignadas a cada centroide anterior. La diferencia entre la media anterior y la actual (diferencia entre centroides) se calcula y se itera estas acciones hasta que este valor sea inferior a un umbral. En otras palabras, se repite hasta que los centroides no se mueven de manera significativa.

Una alternativa a k-medias , es conocida como el algoritmo **Mini lotes k-medias (Mini Batch K-means)** , cuya diferencia principal es que utiliza *mini segmentos* con el fin de optimizar el tiempo de cómputo, sin embargo, ambos cumplen con el mismo objetivo.

Estos "mini segmentos"son subconjuntos de los datos de entrada, seleccionados al azar en cada iteración de entrenamiento. Reducen drásticamente la cantidad de cálculo requerido para converger a una solución local.

El algoritmo itera entre dos pasos principales, similar a k-medias . En la primera etapa, las muestras b son elegidas aleatoriamente del conjunto de datos, para formar los mini segmentos. Se continúa con la asignación de estos al centroide más cercano. En el segundo paso, los centroides se actualizan. La principal diferencia, radica en cómo se efectúa la actualización de los centroides, debido a que esto se hace en función de cada muestra. Estos pasos se llevan a cabo hasta la convergencia o se alcance un número predeterminado de iteraciones.

Affinity Propagation (Propagación por Afinidad)

AffinityPropagation crea grupos mediante el envío de mensajes entre pares de muestras hasta la convergencia. Un conjunto de datos es descrito por el uso de un pequeño número de ejemplares, que se identifican como las más representativas de otras muestras. Los mensajes enviados entre pares representan la idoneidad para una muestra a ser el ejemplo de la otra, la cual se actualiza en respuesta a los valores de otros pares. Esta actualización ocurre de forma iterativa hasta la convergencia, momento en el que se eligen los ejemplares finales, y por lo tanto se da el agrupamiento final [14].

Se elige el número de grupos en base a los datos proporcionados. Para este propósito, los dos parámetros importantes son la preferencia, que controla el número de ejemplares que se utilizan, y el factor de amortiguamiento.

El principal inconveniente que presenta este algoritmo viene dado por la complejidad que posee, el cual se representa por $O(N^2T)$, donde N es el número de muestras y T

es el número de operaciones necesarias para convergir, razón por la cual, el uso de este algoritmo es para set de datos con pequeña cantidad de ejemplos.

Con respecto a la descripción del algoritmo, es posible mencionar que los mensajes enviados a los grupos, pertenecen a dos categorías, la primera es la responsabilidad $r(i,k)$ la cual consiste en la evidencia acumulada que denota que la muestra k podría ser un ejemplar para la muestra i. La segunda es la disponibilidad $a(i,k)$, la cual se define como la evidencia existente para que la muestra i pueda escoger a la muestra k para ser su ejemplar, además considera todos los valores de las otras muestras de k que podrían ser ejemplares. De esta manera, los ejemplares son escogidos por las muestras si:

- Existe una similaridad bastante alta con respecto a las muestras.
- Si es elegido por muchas muestras y resulta ser representativo de sí mismos.

En forma matemática es posible definir la responsabilidad como:

$$r(i,k) \leftarrow s(i,k) - max[a(i,\acute{k}) + s(i,\acute{k}) \forall \acute{k} \neq k]$$

Donde $s(i,k)$ es la similaridad entre las muestras i y k.
A su vez, la disponibilidad, es posible definirla como:

$$a(i,k) \leftarrow min[0, r(k,k) + \sum_{\acute{i} \ s.t. \ \acute{i} \notin \{i,k\}} r(\acute{i},k)]$$

Variación Media (Mean Shift)

El algoritmo de Mean Shift tiene como objetivo descubrir manchas (blobs) en una densidad uniforme de las muestras. Es un algoritmo basado en centroides, que funciona mediante la actualización de los candidatos para centroides para ser la media de los puntos dentro de una región determinada. Estos candidatos se filtran en una etapa de post-procesamiento para eliminar duplicados y así formar el conjunto final de centroides.

Dado un candidato x_i para la iteración t, el candidato a centroide es actualizado en base a la ecuación:

$$x_i^{t+1} = x_i^t + m(x_i^t)$$

Donde $N(x_i)$ es la vecindad de las muestras dentro de una distancia dada alrededor x_i y m es el vector de desplazamiento medio, que se calcula para cada centroide que apunta hacia una región del aumento máximo en la densidad de puntos. Ésta se calcula utilizando la siguiente ecuación, en la que la actualización de manera efectiva denota a un centroide ser la media de las muestras dentro de su vecindad:

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}$$

El algoritmo ajusta automáticamente el número de grupos, dependiendo de un parámetro bandwidth, el cual se asocia al tamaño de la región en la que se debe buscar los centroides. El algoritmo no es altamente escalable, ya que requiere múltiples búsquedas de vecinos más cercanos durante la ejecución del algoritmo. El algoritmo se garantiza a converger, sin embargo, se detendrá la iteración cuando el cambio en centroides es pequeño.

Clustering Jerarquizado

La agrupación jerárquica o HCA (por sus siglas en inglés) es un método de análisis de conglomerados, que busca construir una jerarquía de agrupaciones. Las estrategias para la agrupación son posible dividirlas en dos:

- **Aglomerativa**: consiste en un enfoque de "abajo hacia arriba": cada observación se inicia en su propio clúster, y pares de grupos se fusionan a medida que se asciende en la jerarquía.
- **Divisiva**: consiste en un enfoque "de arriba hacia abajo": todas las observaciones se inician en un único clúster, y las divisiones se realizan de forma recursiva conforme se desciende en la jerarquía.

Siendo generalmente expuestos los resultados en forma de dendrograma, por otro lado, la complejidad del algoritmo, en el caso general es $O(n^2 \log(n))$, lo cual presenta problemas para set de datos extensos [15].

Con el fin de decidir qué grupos se deben combinar (por aglomeración), o cuando un grupo se debe dividir (por división), se requiere una medida de disimilitud entre los conjuntos de observaciones. En la mayoría de los métodos de agrupación jerárquica, esto se logra mediante el uso de una métrica apropiada (una medida de la distancia entre pares de observaciones), y un criterio de vinculación que especifica la disimilitud de conjuntos como una función de las distancias por pares de observaciones en los conjuntos.

Las métricas asociadas pueden ser las siguientes [8]:

- **Distancia Euclidiana**: $\|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$.
- **Distancia Euclidiana cuadrática**: $\|a - b\|_2^2 = \sum_i (a_i - b_i)^2$.
- **Distancia Manhattan**: $\|a - b\|_1 = \sum_i |a_i - b_i|$.
- **Distancia Máxima**: $\|a - b\|_\infty = \text{máx}_i |a_i - b_i|$

- **Distancia de Mahalonobis**: $\sqrt{(a-b)^\top S^{-1}(a-b)}$, donde S es la matríz de covarianza.

El criterio de linkage determina la distancia entre conjuntos de observaciones como una función de las distancias por pares entre observaciones.

Algunos criterios de linkage de uso común entre los dos conjuntos de observaciones A y B son:

- **Acoplamiento máximo (complete linkage clustering)**: $\max\{\, d(a,b) : a \in A,\, b \in B \,\}$.
- **Acoplemiento mínimo (single-linkage clustering)**: $\min\{\, d(a,b) : a \in A,\, b \in B \,\}$.
- **Acoplamiento por la Media (average linkage clustering, UPGMA)**: $\frac{1}{||A||B||} \sum_{a\in A} \sum_{b\in B} d(a,b)$.
- **Acoplamiento por los centroides (Centroid linkage clustering, UPGMC)**: $\|c_s - c_t\|$ donde c_s y c_t son los centroides de los clusters s y t, respectivamente.
- **Mínimo energía (Minimum energy clustering)**: $\frac{2}{nm} \sum_{i,j=1}^{n,m} \|a_i - b_j\|_2 - \frac{1}{n^2} \sum_{i,j=1}^{n} \|a_i - a_j\|_2 - \frac{1}{m^2} \sum_{i,j=1}^{m} \|b_i - b_j\|_2$.

Un aspecto interesante de este algoritmo es que pueden ser añadidas las limitaciones de conectividad, es decir, sólo grupos adyacentes pueden fusionarse entre sí, esto es, a través de una matriz de conectividad que define para cada muestra, las muestras de vecinos después de una estructura dada de los datos. Estas restricciones son útiles para imponer una cierta estructura local, así como para hacer que el algoritmo sea más rápido, especialmente cuando el número de las muestras es alta.

DBSCAN

El algoritmo DBSCAN ve agrupaciones como áreas de alta densidad separadas por zonas de baja densidad. Debido a esta visión bastante genérica, las agrupaciones que se encuentran pueden ser de cualquier forma, en lugar de k-medias que supone que los grupos tienen la forma convexa [16].

El componente central de DBSCAN es el concepto de muestras de núcleo, las cuales son las muestras que se encuentran en áreas de alta densidad. Por lo tanto, un clúster es un conjunto de muestras de núcleos, cada uno cerca del otro (medido por alguna medida de

distancia) y un conjunto de muestras no básicas que se encuentran cerca de una muestra básica. Hay dos parámetros necesarios para el algoritmo, min samples y EPS, los cuales definen formalmente la densidad deseada.

Más formalmente, se define una muestra del núcleo como una muestra del conjunto de datos de tal manera que existe una cantidad de muestra mínimas y a su vez otras muestras dentro de una distancia de EPS, que se definen como vecinos de la muestra del núcleo. Esto dice que la muestra de núcleo se encuentra en un área densa del espacio vectorial. Un clúster es un conjunto de muestras de núcleo que se puede construir mediante la adopción de forma recursiva de una muestra básica, la búsqueda de todos sus vecinos que son muestras de la base, la búsqueda de la totalidad de sus vecinos que son muestras de núcleo, y así sucesivamente. Un grupo también tiene un conjunto de muestras no básicas, que son las muestras que son vecinos de una muestra básica de la agrupación, pero no son en sí mismos muestras de núcleos. Intuitivamente, estas muestras están al margen de un clúster.

Cualquier muestra de núcleo es parte de un clúster, por definición. Cualquier muestra que no es una muestra del núcleo, y está al menos una distanca eps de cualquier muestra del núcleo, se considera un valor atípico por el algoritmo.

Normalmente, los resultados del algoritmo, pueden representarse tal como se expone en la Figura 1.9, el color indica la pertenencia al clúster, con grandes círculos que indican muestras de núcleos encontrados por el algoritmo, círculos más pequeños son muestras no básicas que todavía son parte de un clúster. Por otra parte, los valores atípicos se indican con puntos negros.

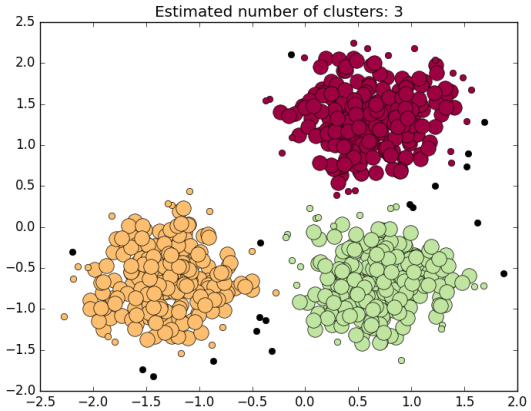

Figura 1.9: Representación de resultados al aplicar la clusterización por DBSCAN

Birch

Birch construye un árbol llamado Characteristic Feature Tree (CFT) de los datos correspondientes. Los datos están esencialmente con pérdida de información, comprimidos en un conjunto de nodos de rasgo característico denominados CF nodos. Estos tienen una serie de subgrupos llamados subclusters de rasgo característico, ubicados en los nodos CF no terminales los cuales pueden tener CF nodos como hijos.

Los subgrupos CF puende contener la información necesaria para la agrupación que evite la necesidad de mantener los datos de entrada enteros en la memoria. Esta información incluye:

- Número de muestras en un subgrupo.
- Suma lineal, representada por un vector n-dimensional que sostiene la suma de todas las muestras.
- Suma al cuadrado, representada por la suma cuadrática de la norma L2 de todas las muestras.
- Centroides, para evitar un nuevo cálculo de sumas lineales con respecto al número de muestras.
- Norma al cuadrado de los centroides.

El algoritmo de Birch tiene dos parámetros, el umbral y el factor de branching. El factor de branching limita el número de subgrupos en un nodo y el umbral limita la distancia entre la muestra de entrada y los subclusters existentes.

Este algoritmo puede ser visto como un método de instancia o reducción de datos, ya que reduce los datos de entrada a un conjunto de subclusters que se obtienen directamente de las hojas de la CFT. Estos datos reducidos pueden ser procesados por la alimentación en un clúster global. Este clúster global puede ser establecido por n clusters. Si este parámetro se establece como valor 0 o ninguno, los subgrupos de las hojas se leen directamente, de lo contrario un paso global de la agrupación etiqueta estos subgrupos en grupos globales y las muestras se asignan a la etiqueta global del subgrupo más cercano.

Una descripción del algoritmo, es posible realizarla en los siguientes puntos:

- Una nueva muestra se inserta en la raíz del árbol CF que es un nodo CF. A continuación, se fusiona con el subgrupo de la raíz, el que tiene el radio más pequeño después de la fusión, limitada por el umbral de ramificación y condiciones de los

factores. Si el subcluster tiene algún nodo hijo, entonces esto se realiza repetidamente hasta que llega a una hoja. Después de encontrar el subcluster más cercano en la hoja, las propiedades de este subgrupo y los subclusters padres se actualizan de forma recursiva.

- Si el radio del subcluster obtenido mediante la fusión de la nueva muestra y el subgrupo más cercano es mayor que el cuadrado del umbral y si el número de subclusters es mayor que el factor de ramificación, a continuación, un espacio se asigna temporalmente a esta nueva muestra. Los dos subgrupos más lejanos se toman y de los subgrupos se dividen en dos grupos sobre la base de la distancia entre estos subgrupos.

- Si este nodo de división tiene un subgrupo de los padres y no hay espacio para un nuevo subgrupo, entonces el padre se divide en dos. Si no hay espacio, entonces este nodo se divide de nuevo en dos y el proceso se continúa de forma recursiva, hasta que llega a la raíz

Mixture Model

Los métodos de clustering basado en modelos tratan de optimizar el conjunto de datos a un modelo matemático. En general estos métodos se basan en la suposición que los datos han sido generados por una mezcla de distribuciones de probabilidad. Dentro de los más utilizados se encuentran **Gaussian Mixture** y **Expectation-Maximization**.

Expectation Maximization, supone que los datos emergen de una mecla de distribuciones, donde cada distribución se denomina como *component distribution*, razón por la cual, los datos pueden agruparse usando un modelo de mezcla de densidades de k distribuciones de probabilidades, sin embargo, el problema reside en estimar los parámetros de estas distribuciones para proveer del mejor ajuste posible a los datos.

El algoritmo Expectation Maximization puede ser considerado como una extensión de k-medias , esto es debido a que: Si k-means asigna cada objeto a un clúster en función de su media, EM asigna cada objeto a un clúster en función de un peso que representa la probabilidad de pertenencia al clúster. Esto requiere que se defina una distribución de probabilidad para los clusters.

Por otro lado, un modelo de mezcla gaussiano como Gaussian Mixture Model (GMM) es una función de densidad de probabilidad representada por una suma de componentes gausianas, GMMs son usadas como modelos paramétricos de la distribución de probabilidad de medidas continuas, donde los parámetros de GMM son estimados

usando iterativamente el algoritmo Expectation-Maximization.

Un GMM es una suma con pesos de densidades gaussianas:

$$p(\vec{x}) = \sum_{i=1}^{M} w_i \times N(\vec{x}|\mu_i, \sum_i)$$

Donde $\vec{x}$ es un vector D-dimensional de datos, w_i son los pesos con $\sum_{i=1}^{M} w_i = 1$, y $N(\vec{x}|\mu_i, \sum_i)$ es la densidad gaussiana, por lo tanto, la caracterización se completa con la media, la matriz de covarianza y el peso de cada componente gaussiana.

Una de las limitantes es que el número de componentes gausianos tiene que ser fijado al principio del algoritmo.

El GMM consta de los siguientes pasos:

1. **Iniziacilización**: para cada clase, un vector compuesto de la media y la matriz de covarianza es construido. Este vector representa las características de la distribución gaussiana usada para caracterizar las entidades del conjunto de datos. Inicialmente estos valores son generados aleatoreamente, posteriormente el algoritmo EM trata de aproximar los valores del vector de la distribución real de los datos.

2. Se estima la probabilidad de cada elemento de pertenecer a un clúster.

3. Se estiman los parámetros de la distribución de probabilidad para el próximo ciclo, primero se calcula la media de la clase a través de la media de todos los puntos en función del grado de relevancia de cada punto, continuando con el cálculo de la matriz de covarianza.

4. **Convergencia**: Después de cada ciclo se ejecuta un test de convergencia para verificar cuánto ha cambiado el vector de parámetros y si la diferencia es menor que un umbral de tolerancia el algoritmo se detiene, no obstante es posible detener el algoritmo debido al alcance de un número máximo de ciclos.

Una representación visual del modelo es posible observarla en la Figura 1.10, en ella se aprecia cómo a medida que se va iterando el algoritmo se generan los cambios y las *separaciones* en grupos de clúster definidos.

Cuadro Resumen

En la Tabla 1.1 se expone un resumen de las características de cada algoritmo expuesto, la escalabilidad que poseen, las distancias que ocupan, los casos de uso y los parámetros que poseen.

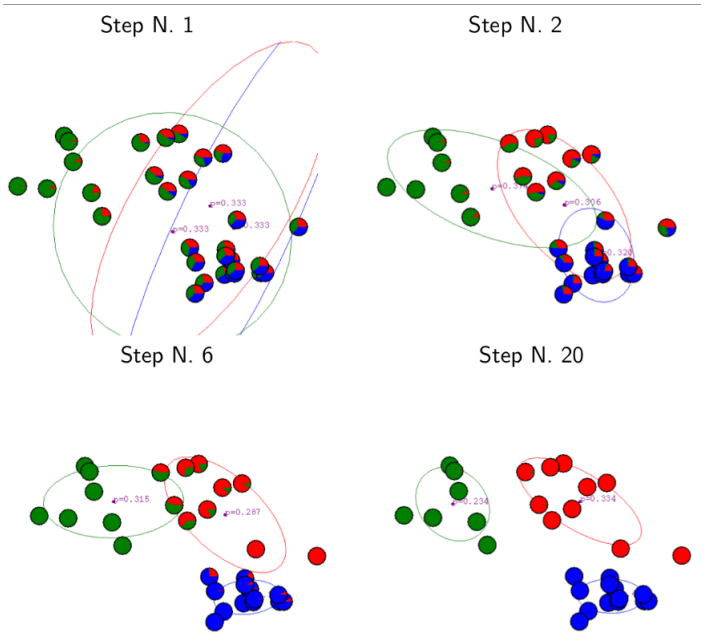

Figura 1.10: Esquema representativo de cambios durante las iteraciones en GMM

Tabla resumen de Algoritmos de Aprendiza No Supervisado				
Algoritmo	**Parámetros**	**Escalabilidad**	**Usos**	**Métrica usada**

K-Means	Número de clúster	Muchas muestras, mediana cantidad de clúster.	De propósito general, la geometría plana, no demasiados grupos	Distancia entre puntos
Affinity propagation	preferencia	No escalable con n ejemplos	Muchos clúster, tamañode clúster desigual, geometría no plana	Distancia gráfica

Mean-shift	bandwidth	No escalable con n ejemplos	Muchos clúster, tamañode clúster desigual, geometría no plana	Distancia entre puntos
Ward hierarchical clustering	Número de clúster	Mucha cantidad de ejemplos y de clusters	Cualquier clúster, es posible conección de constraints	Distantia entre puntos

Agglomerative clustering	Número de clúster, tipo de unión, distancia	Mucha cantidad de ejemplos y de clusters	Muchos clusters, posiblemente restricciones de conectividad, distancias no euclidianas	Cualquier distancia pairwise
DBSCAN	tamaño vecino	Mucha cantidad de ejemplos, mediana cantidad de clúster	Geometría no plana, tamaños de clusters distintos	Distancia entre puntos vecinos

Gaussian mixtures	variado	No escalable	Geometría plana, bueno para la estimación de la densidad	Distancia Mahalanobis para los centros
Birch	branching, umbral	Alto número de clúster y ejemplos	Largo set de datos, eliminación valores atípicos, reducción de datos	distancia euclidiana entre puntos

Tabla 1.1: Cuadro resumen de algoritmos de aprendizaje supervizado

Evaluación del desempeño de un clustering

Evaluar el desempeño de un algoritmo de clustering no es tan trivial como contar el número de errores o la precisión y la recuperación de un algoritmo de clasificación supervisada. En particular, cualquier métrica de evaluación no debe tomar los valores absolutos de las etiquetas de clúster en cuenta, sino más bien si estas agrupaciones definen

separaciones de los datos, de tal manera que los miembros que pertenecen a la misma clase son más similares que los miembros de diferentes clases de acuerdo con alguna similitud métrica.

Existen diversas medidas de similitud con el fin de evaluar el clustering, las cuales se explican a continuación:

Índice Rand ajustado (Adjusted Rand index)

Dado el conocimiento de las clases asignadas como verdaderas (etiquetas verdaderas) y las etiquetas obtenidas por el algoritmo de clustering (etiquetas predichas) el adjuster rand index es una función que mide la similaridad de las dos asignaciones, ignorando permutaciones, posee valores entre -1 y 1, siendo 1 el valor perfecto. Sin embargo, es imperante para evaluar el desempeño, conocer las etiquetas verdaderas de los datos.

Matemáticamente es posible definirlo como:

Sea C una asignación de clase real y dada la agrupación K, se define a y b como:

- a, el número de pares de elementos que están en el mismo set en C y en el mismo set en K.
- b, el número de pares de elementos que están en diferentes set en C y en diferentes set en K.

El valor del rand index no ajustado viene dado por:

$$RI = \frac{a+b}{C_2^{n_{samples}}}$$

Donde $C_2^{n_{samples}}$ es el número total de posibles pares en el set de datos.

Sin embargo, la puntuación de RI no garantiza que las asignaciones de etiquetas al azar conseguirán un valor cercano a cero, para contrarrestar este efecto se puede descartar la espera RI $E[RI]$ de etiquetas al azar mediante la definición del adjusted rand index:

$$ARI = \frac{RI - E[RI]}{\text{máx}(RI) - E[RI]}$$

Información mutua basada en scores

Dado el conocimiento de las etiquetas de las clases reales y las asignaciones obtenidas de algoritmos de agrupación de las mismas muestras, el mutual information es una función que mide el *acuerdo* de las dos asignaciones, ignorando las permutaciones. Existen dos versiones normalizadas diferentes de esta medida:

Normalized Mutual Information, NMI (Información mutua normalizada) y **Adjusted Mutual Information, AMI (Información mutua ajustada)**. NMI es a menudo usado en la literatura mientras que AMI fue porpuesto más recientemente.

Matemáticamente, es posible definir esta forma de evaluación tal que: se asume dos etiquetas asignadas (de los mismos N objetos), U y V, su entropía es la cantidad de incertidumbre para un conjunto de particiones definido por:

$$H(U) = \sum_{i=1}^{|U|} P(i) \log(P(i))$$

donde $P(i) = |U_i|/N$ es la probabilidad que un objecto seleccionado aleatoriamente de la clase U sea asignado a la clase U_i, de igual manera para V:

$$H(V) = \sum_{j=1}^{|V|} P'(j) \log(P'(j))$$

Con $P'(j) = |V_j|/N$ el mutual information (MI) entre U y V es calculado por:

$$MI(U,V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i,j) \log(\frac{P(i,j)}{P(i)P'(j)})$$

donde $P(i,j) = |U_i \cap V_j|/N$ es la probabilidad de que un objeto seleccionado aleatoriamente sea asignado a ambas clases U_i y V_j.

El valor normalizado del mutual information es definido como:

$$NMI(U,V) = \frac{MI(U,V)}{\sqrt{H(U)H(V)}}$$

Este valor del mutual information y también la variante normalizada no se ajusta al azar y tiende a aumentar a medida que aumenta el número de diferentes etiquetas (clusters), independientemente de la cantidad real de *mutual information* entre las asignaciones de etiquetas.

El valor esperado para el mutual information puede ser calculado usando la ecuación descrita por Vinh, Epps, and Bailey, (2009) [17]. En esta ecuación, $a_i = |U_i|$ (el número de elementos en U_i) y $b_j = |V_j|$ (el número de elementos en V_j).

$$E[MI(U,V)] = \sum_{i=1}^{|} U| \sum_{j=1}^{|} V| \sum_{n_{ij}=(a_i+b_j-N)^+}^{\text{mín}(a_i,b_j)} \frac{n_{ij}}{N} \log(\frac{N.n_{ij}}{a_i b_j}) \frac{a_i! b_j! (N-a_i)! (N-b_j)!}{N! n_{ij}! (a_i-n_{ij})! (b_j-n_{ij})! (N-a_i-b_j+n_{ij})!}$$

Usando el valor esperado, el adjusted mutual information puede ser calculado usando una forma similar al ARI:

$$AMI = \frac{MI - E[MI]}{\text{máx}(H(U),H(V)) - E[MI]}$$

Homogeneidad, Completación y medida V (V-measure)

Para el caso en el que se conozca a ciencia cierta las etiquetas reales de las clases, es posible definir medidas de evaluación basándose en la entropía existente. En particular Rosenberg y Hirschberg (2007) [**?**] definen los siguientes dos objetivos deseables para cualquier asignación de clusters:

- **homogeneidad**: cada clúster contiene sólo miembros de una única clase.
- **Totalidad (completeness)**: todos los miembros de una clase son asignados a un mismo clúster.

Los valores de estos score abarcan los rangos entre 0 y 1, siendo 1 el score perfecto.
Es posible definir la homogeneidad y el completeness como:

$$h = 1 - \frac{H(C|K)}{H(C)}$$
$$c = 1 - \frac{H(K|C)}{H(K)}$$

donde $H(C|K)$ es la entropía condicional de las clases dada la asignación del clúster y es definida por:

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n}\cdot\log(\frac{n_{c,k}}{n_k})$$

y $H(C)$ es la entropía de la clase y cuyo valor viene dado por:

$$H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n}\cdot\log(\frac{n_c}{n})$$

con n siendo el número total de muestras, n_c y n_k el número de muestras respectivamente pertenecientes a la clase c y al clúster k, y finalmente $n_{c,k}$ el número de muestra de la clase c asignados al clúster k.

Rosenberg y Hirschberg también definieron un **V-measure** como el score medio de la homogeneidad y completeness:

$$v = 2\cdot\frac{h\cdot c}{h+c}$$

Coeficiente de silueta (Silhouette Coefficient)

Este coeficiente es posible utilizarlo cuando se desconocen las reales etiquetas de los ejemplos, una puntuación alta (por sobre 0.75) denota un modelo con grupos bien definidos. Este coeficiente se define para cada muestra y posee dos score [18]:

- **a**: la distancia media entre un ejemplo y todos los otros puntos en la misma clase.
- **b**: la distancia media entre un ejemplo y todos los otros puntos en el siguiente clúster vecino.

El coeficiente para una única muestra, viene dado por:

$$s = \frac{b-a}{max(a,b)}$$

Calinski-Harabaz Index

Este índice es utilizado cuando las etiquetas son desconocidas, donde un mayor valor de éste implica un modelo mejor definido [19].

Para k clusters, el Calinski-Harabaz index s se da como la razón de la dispersión entre clusters y la dispersión dentro del grupo:

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

Donde B_K es la matriz de dispersión entre grupos y W_K es la matriz de dispersión dentro del clúster definida por:

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$
$$B_k = \sum_q n_q (c_q - c)(c_q - c)^T$$

Con N como el de puntos en el set de datos, C_q el set de puntos en el cluster q, c_q el centro del clúster q, c el centro de E, n_q el número de puntos en el clúster q.

Otros Métodos de Aprendizajes en Minería de Datos

Redes Neuronales

Redes neuronales es posible definirlas como una serie de modelos de aprendizaje que se basan en la forma de trabajo de las redes neuronales biológicas, es decir, se usa el concepto de *neurona* para estimar una función aproximada, la cual dependerá de un largo número de inputs, generalmente desconocidos.

En la imagen 1.11 se aprecia un sistema de red neuronal, en la cual se observa un sistema interconectado por neuronas, las cuales intercambian información en forma de mensaje entre ellas, además cada interconexión tiene un peso, el cual es un valor numérico, que puede ser obtenido en base a la experiencia, en resumen, una red neuronal es un conjunto de entradas y salidas regidas por capas intermedias que permiten evaluar la salida, dichas capas operan entre sí en base a funciones matemáticas y brindan un peso a la conexión, finalmente cada capa es usada para diseñar un modelo de aprendizaje supervisado o no.

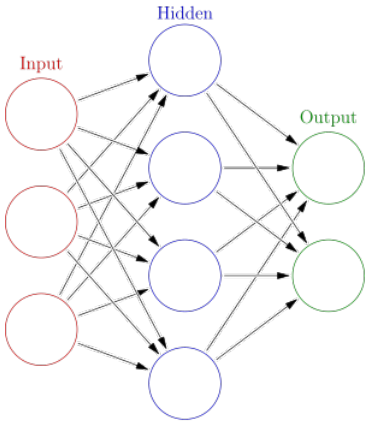

Figura 1.11: Representación esquemática de una Red Neuronal

Deep Learning

Deep Learning es una herramienta de Machine Learning la cual tiene por objetivo modelar abstracciones de alto nivel en los datos por medio del uso de múltiples capas de procesamiento, ya sea a través del uso de estructuras complejas a través de múltiples transformaciones no lineales [20][22][23].

La investigación en esta área tiene por objetivo generar mejores representaciones y crear modelos para aprender de estas representaciones a partir de datos no marcados a gran escala. En geneal las representaciones obtenidas se inspiran en los avances en la neurociencia y se basa libremente en la interpretación de los patrones de procesamiento y comunicación de información en un sistema nervioso, como la codificación neural que intenta definir una relación entre varios estímulos y respuestas neuronales asociadas en el cerebro[19-20-21].

Deep learning posee diversas arquitecturas, tales como: deep learning network, matrices de convoluciones, redes neuronales recurrentes, etc. las cuales han sido utilizadas en visión artificial para el reconocimiento de patrones, aprendizaje de escritura, etc. Deep Learning es una herramienta de Machine Learning la cual tiene por objetivo modelar abstracciones

alto nivel en los datos por medio del uso de múltiples capas de procesamiento, ya sea a través del uso de estructuras complejas a través de múltiples transformaciones no lineales [24][25][26].

Algoritmos Genéticos

Los algoritmos genéticos se basan en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados [27] [28].

Los algoritmos genéticos se enmarcan dentro de los algoritmos evolutivos, que incluyen también las estrategias evolutivas, la programación evolutiva y la programación genética.

Los algoritmos genéticos funcionan entre el conjunto de soluciones de un problema llamado fenotipo, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, generalmente binaria, llamada cromosoma. Los símbolos que forman la cadena son llamados los genes.

Cuando la representación de los cromosomas se hace con cadenas de dígitos binarios se le conoce como genotipo. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud. Las siguientes generaciones (nuevos cromosomas), son generadas aplicando los operadores genéticos repetidamente, siendo estos los operadores de selección, cruzamiento, mutación y reemplazo.

Un algoritmo genético puede presentar diversas variaciones, dependiendo de cómo se aplican los operadores genéticos (cruzamiento, mutación), de cómo se realiza la selección y de cómo se decide el reemplazo de los individuos para formar la nueva población. En general, el pseudocódigo consiste de los siguientes pasos:

- **Inicialización:**

  Se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones del problema. En caso de no hacerlo aleatoriamente, es importante garantizar que dentro de la población inicial, se tenga la diversidad estructural de estas soluciones para tener una representación de la mayor parte de la población posible o al menos evitar la convergencia prematura.

- **Evaluación:**

A cada uno de los cromosomas de esta población se aplicará la función de aptitud para saber cómo *buena* es la solución que se está codificando.

- **Condición de término:**

  El Algoritmo Genético se deberá detener cuando se alcance la solución óptima, pero ésta generalmente se desconoce, por lo que se deben utilizar otros criterios de detención. Normalmente se usan dos criterios: correr el Algoritmo Genético un número máximo de iteraciones (generaciones) o detenerlo cuando no haya cambios en la población. Mientras no se cumpla la condición de término se hace lo siguiente:

  - **Selección:** Después de saber la aptitud de cada cromosoma se procede a elegir los cromosomas que serán cruzados en la siguiente generación. Los cromosomas con mejor aptitud tienen mayor probabilidad de ser seleccionados.
  - **Recombinación o Cruzamiento:** La recombinación es el principal operador genético, representa la reproducción sexual, opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos cromosomas padres.
  - **Mutación:** modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual.
  - **Reemplazo:** una vez aplicados los operadores genéticos, se seleccionan los mejores individuos para conformar la población de la generación siguiente.

1.2.3. Sistemas de Información y Modelo Vista Controlador

Los Sistemas de información son la base de toda dispocisión de información, ya sea a través de una herramienta o por medio de una plataforma web, en general consta con un sistema de almacenamiento persistente, un conjunto de herramientas que permiten extraer y manejar la información con respecto a dicho almacenamiento y módulos de consulta dispuestos en herramientas de visualización de los datos. Normalmente su desarrollo se basa en el conjunto de metodologías de ingeniería en software a través de las cuales es analizado, diseñado e implementado el sistema, mediante el desarrollo de diversos artefactos de software, tales como: casos de uso, diagramas de clases, arquitectura de software, etc. Dentro de las arquitecturas más utilizadas se encuentran: arquitectura 3 capas, cliente-servidor, modelo Vista Controlador, siendo esta última la que se utilizará para el desarrollo de la herramienta computacional.

El modelo vista controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son: el modelo, la vista y el controlador.

Por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.[29] [30].

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento [31] [32]

Se puede definir los componentes del modelo como sigue:

- **Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la *vista* aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al *modelo* a través del *controlador*.

- **Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al *modelo* cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su *vista* asociada si se solicita un cambio en la forma en que se presenta el *modelo* (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el *controlador* hace de intermediario entre la *vista* y el *modelo*.

- **Vista:** Presenta el *modelo* (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho *modelo* la información que debe representar como salida.

Bibliografía

[1] Andres Irback, Simon Mitternacht, Sandipan Mohanty, *An effective all-atom potencial for proteins.*

[2] M. L. Hekkelman, T. A. H. te Beek, S. R, Pettifer, D. Thorne, T. K. Attwood, G. Vriend, *WIWS: a protein structure bioinformatics Web service collection.*

[3] Abdi. H., And Williams, L.J. (2010). "Principal component analysis". Wiley Interdisciplinary Reviews: Computational Statistics. 2 (4): 433-459. doi:10.1002/wics.101.

[4] Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4.

[5] Bengio, Y.; et al. (2013). "Representation Learning: A Review and New Perspectives"(PDF). Pattern Analysis and Machine Intelligence. 35 (8): 1798-1828. doi:10.1109/TPAMI.2013.50.

[6] S. Kotsiantis, supervisado Aprendizaje Automático: Una Revisión de la Clasificación de las técnicas de Informática Diario 31 (2007) 249-268.

[7] "Five balltree construction algorithms", Omohundro, S.M., International Computer Science Institute Technical Report (1989).

[8] "The DISTANCE Procedure: Proximity Measures". SAS/STAT 9.2 Users Guide. SAS Institute. Retrieved 2009-04-26.

[9] "Automatic Capacity Tuning of Very Large VC-dimension Classifiers", I. Guyon, B. Boser, V. Vapnik - Advances in neural information processing 1993.

[10] Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning. 20 (3): 273-297. doi:10.1007/BF00994018.

[11] Pedregosa, F. Varoquaux, G. Gramfort, A. Michel, V. (2010-2016). Scikit-learn: Support Vector Machines. INRIA. Recuperado de http://scikit-learn.org/stable/modules/svm.html/kernel-functions.

[12] Lebart,L.;Morineau,A. y Fenelon,J.P.: "Traitement des Donées Statistiques"Dunod. 1979.

[13] "k-means++: The advantages of careful seeding.ᴬrthur, David, and Sergei Vassilvitskii, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics (2007).

[14] Brendan J. Frey; Delbert Dueck (2007). "Clustering by passing messages between data points". Science. 315 (5814): 972-976. doi:10.1126/science.1136800. PMID 17218491.

[15] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009). 14.3.12 Hierarchical clustering (PDF). The Elements of Statistical Learning (2nd edición). Nueva York: Springer. pp. 520-528. ISBN 0-387-84857-6. Consultado el 20 de diciembre de 2016.

[16] "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.ᴱˢter, M., H. P. Kriegel, J. Sander, and X. Xu, In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231. 1996.

[17] Vinh, Epps, and Bailey, (2009). "Information theoretic measures for clusterings comparison". Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09. doi:10.1145/1553374.1553511. ISBN 9781605585161.

[18] Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics 20: 53-65. doi:10.1016/0377-0427(87)90125-7.

[19] Calinski, T., & Harabasz, J. (1974). "A dendrite method for cluster analysis". Communications in Statistics-theory and Methods 3: 1-27. doi:10.1080/03610926.2011.560741.

[20] Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8): 1798-1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50.

[21] Bengio, Y.; et al. (2013). "Representation Learning: A Review and New Perspectives"(PDF). Pattern Analysis and Machine Intelligence. 35 (8): 1798-1828. doi:10.1109/TPAMI.2013.50.

[22] Bengio, Yoshua (2009). "Learning Deep Architectures for AI"(PDF). Foundations and Trends in Machine Learning 2 (1): 1-127. doi:10.1561/2200000006.

[23] Deng, L.; Yu, D. (2014). "Deep Learning: Methods and Applications"(PDF). Foundations and Trends in Signal Processing 7: 3-4. doi:10.1561/2000000039.

[24] Deep Machine Learning - A New Frontier in Artificial Intelligence Research - a survey paper by Itamar Arel, Derek C. Rose, and Thomas P. Karnowski. IEEE Computational Intelligence Magazine, 2013.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016). Deep Learning. MIT Press. Online.

[26] Olshausen, B. A. (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". Nature 381 (6583): 607-609. doi:10.1038/381607a0.

[27] J. H. Holland. University of Michigan Press, Ann Arbor. 1975. Adaptation in Natural and Artificial Systems.

[28] D. E. Goldberg. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 1989. Genetic Algorithms in Search, Optimization and Machine Learning.

[29] "More deeply, the framework exists to separate the representation of information from user interaction."The DCI Architecture: A New Vision of Object-Oriented Programming-Trygve Reenskaug and James Coplien, March 20, 2009.

[30] "... the user input, the modeling of the external world, and the visual feedback to the user are explicitly separated and handled by three types of object.", Applications.

[31] Programming in Smalltalk-80(TM):How to use Model-View-Controller (MVC).

[32] Simple Example of MVC (Model View Controller) Design Pattern for Abstraction

[33] *Data Mining Curriculum*. ACM SIGKDD. 2006-04-30. Retrieved 2014-01-27.

[34] Clifton, Christopher (2010). *Encyclopedia Britannica: Definition of Data Mining*. Retrieved 2010-12-09.

[35] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Retrieved 2012-08-07.

[36] Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic (1996). *From Data Mining to Knowledge Discovery in Databases* (PDF). Retrieved 17 December 2008.

[37] inehan WM, Zbar B, Klausner R. Renal carcinoma. In: Scriver CR, Beaudet AL, Sly WS, Valle D, eds. The metabolic and molecular basis of inherited disease. 8th edn. New York: McGraw-Hill, 2001:907-29.

[38] inehan WM, Zbar B, Klausner R. Renal Carcinoma. In: Vogelstein B, Kinzler K, eds. The genetic basis of human cancer. 2 edn. New York: McGraw-Hill, 2002.

[39] Linehan WM, Lerman MI, Zbar B. Identification of the von Hippel-Lindau (VHL) gene. Its role in renal cancer. JAMA 1995; 273:564-70.

[40] Clifford SC, Maher ER. Von Hippel-Lindau disease: clinical and molecular perspectives. Adv Cancer Res2001;82: 85-105.

[41] Maher ER, Kaelin WG Jr. von Hippel-Lindau disease. Medicine (Baltimore) 1997; 76: 381-91.

[42] Kaelin WG Jr. Molecular basis of the VHL hereditary cancer syndrome. Nat Rev Cancer 2002; 2:673-82.

[43] Russell R Lonser, Gladys M Glenn, McClellan Walther, Emily Y Chew, Steven K Libutti, W Marston Linehan, Edward H Oldfield, *von Hippel-Lindau disease*, Lancet 2003; 361: 2059-67.

[44] Maher ER, Kaelin WG Jr, *von Hippel-Lindau disease*, Medicine [01 Nov 1997, 76(6):381-391], (PMID:9413424), DOI: 10.1097/00005792-199711000-00001.

[45] Gossage, L., Pires, D. E. V., Olivera-Nappa, A., Asenjo, J., Bycroft, M., Blundell, T. L., and Eisen, T. (2014). *An integrated computational approach can classify VHL missense mutations according to risk of clear cell renal carcinoma.* Human Molecular Genetics, 23(22), 5976-5988. ISSN:0964-6906.

[46] Christophe Béroud, Dominique Joly, Catherine Gallou, Frédéric Staroz, Marie Therése Orfan elli and Claudine Junien, *Software and database for the analysis of mutations in the VHL gene*, Nucleic Acids Research, 1998, Vol. 26, No. 1.

[47] Duan DR, Pause A, Burgess WH, Aso T, Chen DY, Garrett KP, Conaway RC, Conaway JW, Linehan WM, Klausner RD. *Inhibition of transcription elongation by the VHL tumor suppressor protein.* Science. 1995 Sep 8;269(5229):1402-6.

[48] Kishida T, Stackhouse TM, Chen F, Lerman MI, Zbar B. *Cellular proteins that bind the von Hippel-Lindau disease gene product: mapping of binding domains and the effect of missense mutations.*, Cancer Res. 1995 Oct 15;55(20):4544-8.

[49] Schoenfeld AR, Davidowitz EJ, Burk RD. *Elongin BC complex prevents degradation of von Hippel-Lindau tumor suppressor gene products.* Proc Natl Acad Sci U S A. 2000 Jul 18;97(15):8507-12.

[50] Lamiell JM, Salazar FG, Hsia YE. von Hippel-Lindau disease affecting 43 members of a single kindred. Medicine 1989; 68:1-29.

[51] Escourolle R, Poirer J. Manual of Neuropathology. 2nd edn.Philadelphia: WB Saunders, 1978: 49-51.

[52] Melmon KL, Rosen SW. Lindau s disease. Am J Med 1964; 36: 595-617.

[53] Catherine L. Worth Robert Preissner Tom L. Blundell, *SDM-a server for predicting effects of mutations on protein stability and malfunction*, Nucleic Acids Research, Volume 39, Issue suppl_2, 1 July 2011, Pages W215-W222, https://doi.org/10.1093/nar/gkr363

[54] Alvaro Olivera-Nappa, Barbara A Andrews and Juan A Asenjo, *Mutagenesis Objective Search and Selection Tool (MOSST): an algorithm to predict structure-function related mutations in proteins*, BMC Bioinformatics201112:122.