IN-T6-7-8-9.pdf

patrivc

Apuntes Variados

4º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

TEMA 6: CLUSTERING

¿Qué es un cluster?

Un cluster es un grupo o conjunto de objetos. Similar a cualquier otro cluster incluido en el mismo grupo y distinto a los objetos incluidos en otros grupos.

¿Qué es el clustering?

Es el análisis del cluster. Es segmentar una población heterogénea en un número de subgrupos o Clusters. Puede verse como clasificación no supervisada, las clases no están predefinidas.

¿A qué problemas nos enfrentamos con el clustering?

A la dificultad del manejo de outliers (ya que se pueden ver como Clusters solitarios o pueden estar forzados a integrarse en un único cluster). Si se realiza en BBDD dinámicas implica que la pertenencia a *clusters* varía en el tiempo, los resultados del clustering son dinámicos. Interpretar el significado de cada *cluster* puede ser difícil. No hay una única solución para un problema de clustering y no es fácil determinar el número de Clusters.

¿Qué es el aprendizaje supervisado y no supervisado?

- **Supervisado:** aprende, a partir de un conjunto de instancias pre-etiquetadas, un método para predecir la clase a que pertenece una nueva instancia

- No supervisado: encuentra un agrupamiento de instancias "natural" dado un conjunto de instancias no etiquetadas

¿Qué es la bondad en un análisis de cluster?

Un buen método de clustering debe producir clusters en los que: se maximize la similaridad intra-cluster y se minimize la similaridad inter-cluster.

La calidad del clustering resultante depende tanto de la medida de similaridad (normalmente es una función de distancia d(i,j)) utilizada como de su implementación.

Las ***funciones de distancia*** son muy sensibles al tipo de variables usadas, así su definición puede cambiar según el tipo: medidas por intervalos, booleanas, categóricas (nominales), ordinales, etc.

El caso más simple de las ***funciones de distancia*** para los valores numéricos es: si tenemos un atributo numérico A → Distancia(X,Y) = A(X) – A(Y) o también Distancia(X,Y) = Distancia euclídea entre X,Y.

Para los atributo nominales: la distancia se fija a 1 si los valores son diferentes, a 0 si son iguales

Es posible dar peso a ciertas variables dependiendo de distintos criterios. En general, es complicado dar definiciones para términos como "suficientemente similar", así que algunas respuestas serán subjetivas y dependientes de umbrales

Propiedades deseables en un método de clustering en minería de datos

Encontramos varias, entre ellas: escalabilidad, capacidad para tratar distintos tipos de variables, capacidad para descubrir *clusters* con formas arbitrarias, capacidad para tratar datos con ruido y *outliers, r*esultados interpretables, …

¿Qué propiedad debe verificar un cluster?

La propiedad más importante que debe verificar un *cluster* es que haya más cercanía entre las instancias que están dentro del *cluster* que respecto a las que están fuera del mismo (similitud entre instancias). Debe responder a las preguntas ¿Qué es la similitud?¿Cómo medir la similitud entre instancias?

¿Qué tipos de clustering encontramos?

Los tipos de clustering que encontramos son: particiones y jerárquico.

- **Algoritmos de particionamiento:** Construir distintas particiones y evaluarlas de acuerdo a algún criterio

- **Algoritmos jerárquicos:** Crear una descomposición jerárquica del conjunto de datos (objetos) usando algún criterio

- **Otros:** basados en densidad, utilizan funciones de conectividad y densidad, basados en rejillas, utilizan una estructura de granularidad de múltiples niveles y basados en modelos. Se supone un modelo para cada uno de los clusters y la idea es encontrar el modelo que mejor ajuste

¿Cuáles son los métodos basados en particionamiento?

Los métodos basados en particionamiento construyen una partición de la base de datos D formada por n objetos en un conjunto de k clusters. Encontramos:

- **k-means (k medias):** cada cluster se representa por el centro del cluster

- **k-medoids o PAM (particionamiento alrededor de los medoides):** cada cluster se representa por uno de los objetos incluidos en el cluster

¿En qué se basa el algoritmo K-means?

Es un algoritmo iterativo en el que las instancias se van moviendo entre *clusters* hasta que se alcanza el conjunto de *clusters* deseado. Necesita como parámetro de entrada el número de *clusters* deseado.

Ventajas:

1. Es relativamente eficiente: O(tkn) donde n es el numero de objetos, k el numero de clusters y t el numero de iteraciones. Normalmente: k, t << n
2. Con frecuencia finaliza en un óptimo local, dependiendo de la elección inicial de los centros de clusters.
3. Reiniciliza las semillas
4. Utiliza técnicas de búsqueda más potente como algoritmos genéticos o enfriamiento estocástico

Desventajas:

1. Requiere especificar el número de clusters (k) en la entrada, lo que puede ser difícil en
   algunos casos.
2. Solo es aplicable cuando el concepto de media es definible
3. Es débil ante datos ruidosos y/o con outliers
4. Solo genera clusters convexos

¿En qué se basa el algoritmo Mean Shift?

En lugar de fijar k, el algoritmo MeanShift fija un radio (bandwidth) y va desplazando centroides hasta las regiones más densas. Se pueden usar kernels gausianos para ponderar los objetos. El radio se puede estimar con k-NN.

¿En qué se basa el algoritmo DBSCAN?

A partir de un punto, va buscando otros puntos en su vecindad y uniéndolos al clúster hasta que no se alcancen más puntos. Puede encontrar cluster con distintas formas y es robusto a outliers. Parámetros: eps (radio) y minPts (tamaño mínimo). eps se puede estimar por k-distancia con k=minPts.

¿En qué se basa el algoritmo BIRCH (clustering incremental)?

C*lustering* incremental: agrupa conforme se van recibiendo objetos. Mantiene características de los *clusters* (CF={numero elementos, suma lineal, suma cuadrática}) para resumir los objetos que van llegando.

Cada vez que llega un objeto, desciende por el árbol escogiéndose en cada nodo el CF más cercano. Cuando llega a una hoja, si puede ser absorbido por algún CF existente (porque esté en su radio menor que T) se agrega, si no, se crea un CF nuevo si hay menos de L.

Si se alcanza el máximo L, se divide la hoja en dos con el par de CF más lejanos de la hoja anterior. La agregación y división se propaga recursivamente hacia arriba

¿Qué medidas de rendimiento hay?

- Coeficiente silhouette: mide cómo de similares son los objetos de un mismo clúster (cohesión) comparado con otros clusters (separación). La media de todos los *s(i)* es el coeficiente

silhoutte que mide la calidad global del agrupamiento $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- Calinski-Harabasz: razón entre la dispersión intra-clusters y la dispersión inter- clusters. Cuanto mayor es el valor, mejor es el agrupamiento. N es el número de objetos y |P|=k es el número de clusters

$$\mathrm{CH}(P) = \frac{(N - |P|)\,\mathrm{inter}_{CH}(P)}{(|P| - 1)\,\mathrm{intra}_{CH}(P)}$$

$$\mathrm{inter}_{CH}(P) = \sum_{C \in P} |C|\,\mathrm{d}(\bar{C}, \bar{X}) \text{ e } \mathrm{intra}_{CH}(P) = \sum_{C \in P} \sum_{x \in C} \mathrm{d}(x, \bar{C})$$

¿Qué son los métodos jerárquicos?
La salida es una jerarquía entre clusters. Dependiendo del nivel de corte obtendremos un clustering distinto. No requiere como parámetro el número de clusters. Encontramos:

- **Métodos aglomerativos:** se basan en medir la distancia entre clusters. En cada paso se fusionan los dos clusters más cercanos. Los outliers se agrupan en el nivel alto de la jerarquía.
- **Métodos divisivos:** comienzan con un único clúster (toda la BD) y en cada paso se selecciona un clúster y se subdivide. Se debe dar una condición de parada, o en su defecto se detiene el proceso cuando cada clúster contiene un único objeto. Podemos distinguir dos variables: unidimensional (solo se considera una variable para la partición) y multidimensional (se consideran todas las variables).

TEMA 7: PATRONES FRECUENTES Y REGLAS DE ASOCIACION

¿Qué es el descubrimiento de asociaciones?
Es la búsqueda de patrones frecuentes, asociaciones, correlaciones o estructuras causales entre conjuntos de artículos u objetos a partir de bases de datos transaccionales, relacionales y otros conjuntos de datos. Búsqueda de secuencias o patrones temporales.
Ejemplos de aplicaciones donde se usa: análisis de la cesta de la compra, diseño de catálogos

¿Qué es el Market Basket Analysis (analisis de la cesta de la compra)?
Se divide en:
- Análisis de clientes: se utiliza información sobre lo que ha comprado un cliente para ofrecernos una aproximación sobre quién es y por qué hace ciertas compras
- **Análisis de productos:** aporta información sobre qué productos tienden a ser comprados juntos
Ejemplo: Los viernes por la tarde, con frecuencia, quienes compran pañales, compran también cerveza.

¿Qué son las reglas de asociación?
Debido a la claridad y utilidad de los resultados de la investigación en torno al área del análisis de cestas de la compra, se forman las reglas de asociación. El objetivo de estas es: que dada una base de datos de transacciones, donde cada transacción es una lista de artículos, encontrar todas las reglas que co-relacionen la presencia de un conjunto de artículos con otro conjunto de artículos.
La idea es obtener reglas del tipo: “Antecedente => Consecuente [soporte, confianza]” en el ejemplo de pañales y cerveza es: compra(x, “pañales”) => compra(x, “cerveza”) [0.5%, 60%]

Conceptos básicos
- **Transacción**. *Formato relacional:* < Tid, item >, < 1, item1 >, <1, item2 >, <2, item3 >. *Formato compacto* , < 1,{item1, item2} >, <2,{item3} >
- **Item (o artículo)**: elemento individual
- **Itemset (o conjunto):** conjunto de ítems/artículos
- **Soporte del conjunto I:** no de transacciones conteniendo I
- **Soporte mínimo m_s:** umbral del soporte
- **Conjunto frecuente:** con soporte >= m_s
  Los conjuntos frecuentes representan conjuntos de artículos que están correlacionados positivamente.

Tipos de reglas de asociación:
- **Asociaciones Booleanas vs Cuantitativas** dependiendo del tipo de los valores que se manejan
    compra(x, “SQLServer”) ^ compra (x, “Libro de MD”) → compra(x,"DBMiner") [0.2%,60%]
    Edad (x, ‘30..39’) ^ ingresos (x, ’42K..48K’) → compra (x, ‘PC’) [1%,75%]
- **Asociaciones unidimensionales vs multidimensionales** A->B, | | , A&B&...&N->D
- **Análisis con distintos nivles de abstracción:**
    Edad(x,'30..39')→ compra (x, cerveza)
    Edad (x,’30..39’) → compra (x, cerveza alemana)
- **Posibles Extensiones:** Correlaciones, análisis de causalidad. Asociación no implica necesariamente correlación o causalidad.

¿Qué es el soporte (s) y la confianza (c)?
Son medidas para encontrar todas las reglas X & Y ⇒ Z con un mínimo de soporte y confianza.
El **soporte** es la probabilidad de que una transacción contenga {X & Y & Z}. La **confianza** es la probabilidad condicional P(Z|X&Y).
El soporte minimo (s) es aquel en el que tenemos un alto nivel de pocas reglas validas que ocurren con frecuencia y un bajo nivel de muchas reglas validas que ocurren con poca frecuencia.
La confianza mínima, tenemos un alto nivel de pocas reglas, pero muy fiables y un bajo nivel de muchas reglas, pero muchas de ellas inciertas.

¿Cuál es el proceso de extracción?
Al tratar con bases de datos grandes, el proceso se descompone en dos pasos:
1. Encontrar conjuntos de artículos frecuentes: Mayor ocurrencia que el soporte mínimo fijado
2. Generar reglas de asociación “fuerte” a partir de los conjuntos de artículos frecuentes: Deben satisfacer el mínimo fijado tanto para soporte como para confianza

¿En qué se basa el Algoritmo Apriori?
Encuentra las asociaciones más frecuentes. Itera sobre la base de datos hasta que las asociaciones obtenidas no tienen el soporte mínimo. Método simple pero robusto. Tiene una salida intuitiva.
El algoritmo busca iterativamente conjuntos frecuentes con cardinalidad 1 hasta k (k-conjunto), y después. Usa los conjuntos frecuentes para generar las reglas de asociación. En el paso clave del descubrimiento de conjuntos frecuentes, se basa en el principio “a priori”: cualquier subconjunto de un conjunto de artículos frecuente debe ser frecuente
Ejemplo: si {AB} es un conjunto frecuente, entonces tanto {A} como {B} deberían ser frecuentes
Esto permite definir el principio de poda en Apriori: Dado un conjunto “infrecuente”, no hay necesidad de generar sus superconjuntos (cualquier conjunto que contenga al subconjunto infrecuente, no es frecuente)

¿Qué medidas de interés encontramos?
- Medidas objetivas: soporte y confianza
- Medidas subjetivas: una regla (patrón) es interesante si es inesperada (sorprendente para el usuario) y/o útil (el usuario puede hacer algo con ella)
- **Interés** (correlación, empuje)
    • lift(A→B) = P(B|A)/P(B) = P(A y B)/(P(A)·P(B))
    • Toma *P(A)* y *P(B)* en consideración
    • *P(A* y *B) = P(A)·P(B)* si A y B son independientes (*lift* = 1)
    • *A* y *B* negativamente correlacionadas si *lift* es menor que 1; *A* y *B* positivamente correlacionadas si *lift* es mayor que 1
- **Lift**: relación entre el apoyo del observador y el apoyo si A y C fueran estadísticamente independientes. Si Lift < 1, correlación negativa. Si Lift > 1, correlación positiva. Si Lift = 1, no hay correlación.

TEMA 8: DEEP LEARNING

¿Qué significa Deep Learning?

Significa utilizar una red neuronal con varias capas de nodos de entrada y salida. La serie de capas entre entrada y salida realizan una identificación de características y procesamiento en una seria de etapas (tal y como pueden hacer nuestros cerebros).

¿Qué es una RNA convencional?

En una red neuronal artificial (RNA) estándar como MLP (multilayer perceptron), para cada ejemplo de entrenamiento, se propaga la entrada por la red para obtener una salida. Esta se compara con la salida esperada y se va retropropagando el error para ir ajustando suavemente los pesos de cada capa desde la última hasta la primera.

Para que funcione, necesita recibir miles y miles de datos y realizar miles y miles de pequeños ajustes en los pesos.

¿Qué pasa con la invariancia de posición?

Nuestros detectores de unidades de ejemplo estaban vinculados a partes específicas de la imagen. Las capas sucesivas pueden aprender características de nivel superior.

¿Tiene sentido emplear múltiples capas?

Si. Las arquitecturas de redes neuronales de muchas capas pueden ser capaces de aprender las verdaderas características subyacentes y la "lógica de características" para así generalizar mejor.

¿Cuál es la nueva forma de entrenar un RNA multicapa con aprendizaje profundo?

Pues entrena primero la primera capa, luego la segunda, luego la tercera,.., etc.

Cada una de las capas de salida esta entrenada para ser un codificador automático (auto-encoder). Este se ve obligado a aprender buenas características (reducción de la dimensionalidad) que describen lo que proviene de la capa anterior.

¿Qué es un auto-encoder?

Un auto-encoder esta entrenado con un algoritmo de ajuste de peso absolutamente estándar para reproducir la entrada. Al hacer que esto suceda con muchas menos unidades que las entradas, obliga a las unidades de la capa oculta a convertirse en buenos detectores de características.

¿Qué es una Red Neuronal Convolucional (CNN)?

Una CNN aprende automáticamente los valores de sus filtros en función a la tarea que desea realizar.

Las capas se van apilando y la salida de uno se convierte en la entrada de otro. Las capas pueden repetirse (apilamiento profundo). Juntando todo tenemos que: un conjunto de píxeles puede convertirse en un conjunto de votos. El problema es que requiere una gran cantidad de datos para obtener mejores precisiones.

¿Cómo trabaja la CNN?

Con pasos convolucionales. La red neuronal convolucional utiliza tres ideas básicas: campos receptivos locales, pesos compartidos y agrupación.

Descripción de los pasos convolucionales

La **zancada (stride)** es el tamaño del paso que el filtro de convolución mueve cada vez. Suele ser 1; el filtro se desliza pixel a pixel. A mayor tamaño de zancada, el filtro se desliza sobre la entrada con un intervalo mayor y por tanto, tiene menos superposición entre las celdas.

El tamaño del mapa de entidades siempre es más pequeño que la entrada, por lo que debemos hacer algo para evitar esto. Aqui es donde entra el **relleno (padding)**: agregamos una capa de pixeles de valor cero para rodear la entrada y evitar que el mapa de entidades encoja. Esto mejora el rendimiento y aseguramos que el tamaño del núcleo y la zancada coincidan con la entrada.

Después de una capa de convolución, es común agregar una **capa de agrupación (pooling)** entre las capas de CNN. Su función es reducir continuamente la dimensionalidad para reducir el número de parámetros y el cálculo en la red. Esto acorta el tiempo de entrenamiento y controla el sobreajuste. El tipo más frecuente es '***max pooling'***, que toma el valor máximo en cada ventana. Esto disminuye el tamaño del mapa de entidades y mantiene la información significativa.

¿Qué son los mapas de características?

Son la salida de cada convolución. En la primera capa (en las arquitecturas CNN), suelen actuar como detectores de bordes y formas simples. Conservan la mayor parte de la información presente en la imagen.

Los mapas de características más profundos contienen menos información sobre la imagen y más sobre la clase de la imagen. Esto es porque codifican conceptos de alto nivel como "nariz de gato" y "oreja de perro".

A medida que profundizamos se vuelven más dispersos, lo que significa que los filtros detectan menos funciones (ya que buscamos cosas más complejas).

Limitaciones de los mapas de características

- Calidad de los datos (gran cantidad de datos)
- Requiere una gran cantidad de datos para obtener mejores precisiones.

Las soluciones a esto son:

- Preprocesamiento y aumento de datos: aumentar el volumen del conjunto de datos de entrenamiento artificialmente usando transformaciones.
- Transferir el aprendizaje

TEMA 9: PROBLEMAS SINGULARES

Definición del problemas de las clases desbalanceadas/equilibradas

Las áreas reales de aplicación en ingeniería se caracterizan por tener una distribución de ejemplos muy diferente entre sus clases. Debido al problema o a las limitaciones durante el proceso de recolección de datos, la clase positiva a menudo representa el concepto de mayor interés para el problema, mientras que la clase negativa representa contraejemplos.

Problema de conjuntos de datos desequilibrados: establece un handicap para la correcta identificación de los diferentes conceptos a aprender

¿Por qué puede ser difícil aprender de conjuntos de datos desequilibrados?

Porque el proceso de búsqueda esta guiado por errores de datos globales. Se ignoran los grupos pequeños, los grandes se clasifican con precisión. Las reglas de clasificación sobre la clase positiva están altamente especializadas.

Encontramos distribuciones de clases sesgadas, características intrínsecas de los datos y las clases mayoritarias se superponen a las clases minoritarias.

El problema de las clases desequilibradas es que los datos están sesgados hacia la clase mayoritaria.

¿Cómo podemos evaluar un algoritmo en dominios desequilibrados?

Con una matriz de confusión para un problema de dos clases. No tiene en cuenta las "Tasas Individuales", que son muy importantes en problemas de desequilibrio

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

$TPR = \frac{TP}{TP+FN}$ → Sensitivity

$TNR = \frac{TN}{TN+FP}$ → Specify

$FNR = \frac{FN}{TP+FN}$

$FPR = \frac{FP}{TN+FP}$

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

$Precision = \frac{TP}{TP+FP}$

Classical evaluation:

Error Rate: (FP + FN)/N
Accuracy Rate: (TP + TN) /N

¿Qué es la F-Measure?

La media F1 es una media armónica entre la precisión y la recuperación:

▪**Precisión**: número de resultados positivos correctos dividido por el número de todos los resultados positivos

▪ **Recuperación/sensibilidad**: número de resultados positivos correctos dividido por el número de resultados positivos que deberían haberse devuelto.

$$F_1 = 2 \cdot \frac{1}{(\frac{1}{recall} + \frac{1}{precision})} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Fórmula general

$$F_b = (1 + b^2) \cdot \frac{precision \cdot recall}{(b^2 \cdot precision) + recall}$$

¿Cómo abordar conjuntos de datos desequilibrados?

Mediante el preprocesamiento de algoritmos (en concreto estos dos enfoques son híbridos):

- **Over-sampling**: Random Focused. Mantener ejemplos influyentes y reequilibrar el conjunto de entrenamiento

- **Under-sampling:** Random Focused. Reequilibrar el conjunto de entrenamiento, eliminar instancias ruidosas en los límites de decisión y reducir el conjunto de entrenamiento

También con el **remuestreo** del conjunto de datos original: es el proceso de manipular la distribución de los ejemplos de entrenamiento en un esfuerzo por mejorar el rendimiento de los clasificadores. No hay garantía de que los ejemplos de entrenamiento ocurran en su distribución óptima en problemas prácticos. La idea del remuestreo es “agregar o eliminar ejemplos con la esperanza de alcanzar la distribución óptima de los ejemplos de entrenamiento” y, por lo tanto, mejorar la capacidad potencial de los clasificadores.

Dentro del UnderSampling encontramos:

Tomek Links: tenemos Ei, Ej pertenecen a diferentes clases, d(Ei, Ej) es la distancia entre ellos. Un par (Ei, Ej) se denomina enlace Tomek si no hay un ejemplo El, tal que d(Ei, El) < d(Ei, Ej) o d(Ej, El) < d(Ei, Ej). Hay que liminar ejemplos de la clase mayoritaria que pertenece a los enlaces de Tomek.

CNN (vecinos más cercanos condensados): sirve para eliminar tanto el ruido como los límites ejemplos. El algoritmo es el siguiente:

1. Sea E el conjunto de entrenamiento original
2. Sea E’ que contenga todos los ejemplos minoritarios de S y un ejemplo mayoritario seleccionado al azar
3. Clasifica E con la regla 1-NN usando los ejemplos en E’
4. Mover todos los ejemplos mal clasificados de E a E’
5. Si E’ ha cambiado, vaya a 3. De lo contrario, deténgase.

Dentro del OverSampling encontramos:

SMOTE: en lugar de replicar, inventemos algunos nuevas instancias. Se generan nuevos ejemplos de las clases minoritarias y se realiza la interpolación entre varias instancias de clases minoritarias que yacen juntas. Para cada muestra minoritaria: encontramos sus k vecinos minoritarios más cercanos, seleccionamos aleatoriamente j vecinos y se generan aleatoriamente muestras sintéticas a lo largo las líneas que unen la muestra minoritaria seleccionada y sus vecinos j.

Las muestras sintéticas se generan de la siguiente manera: se toma la diferencia entre el vector de características (muestra) bajo consideración y su vecino más cercano. Multiplicar esta diferencia por un número aleatorio entre 0 y 1. Se agrega al vector de características bajo consideración.

¿Qué es el sobremuestreo aleatorio (con reemplazo) de la clase minoritaria?

Tiene el efecto de hacer que la región de decisión para la clase minoritaria sea muy específica. En un árbol de decisión, causaría una nueva división y, a menudo, daría lugar a un sobreajuste.

¿Qué es el sobremuestreo informado de SMOTE?

Generaliza la región de decisión para la clase minoritaria. Se aprenden regiones más grandes y menos específicas. Presta atención a las muestras de clases minoritarias sin causar un sobreajuste.

¿Qué es la sobregeneralización?

SMOTE es intrínsecamente peligroso ya que generaliza ciegamente el área minoritaria hace caso omiso de la clase mayoritaria. Esta estrategia es particularmente problemática en el caso de distribuciones de clase altamente sesgadas: ya que la clase minoritaria es muy escasa con respecto a la mayoritaria y da como resultado una mayor probabilidad de mezclas de clases.

¿Qué problemas tenemos con SMOTE?

Problema con Smote: podría introducir los ejemplos de la clase minoritaria artificial demasiado profundamente en el espacio de la clase mayoritaria.

Smote + Tomek: en lugar de eliminar solo los ejemplos de la clase mayoritaria que forman los enlaces Tomek, se eliminan los ejemplos de ambas clases.
Smote + ENN: ENN elimina cualquier ejemplo cuya etiqueta de clase difiera de la clase de al menos dos de sus vecinos. ENN elimina más ejemplos que los enlaces de Tomek. ENN eliminar ejemplos de ambas clases

¿Qué es el aprendizaje semisupervisado?

SSL es un paradigma de aprendizaje relacionado con el diseño de modelos en presencia de datos etiquetados y no etiquetados. Esencialmente, los métodos SSL utilizan muestras no etiquetadas para modificar o volver a priorizar la hipótesis obtenida solo a partir de muestras etiquetadas.

• SSL es una extensión del aprendizaje supervisado y no supervisado al incluir información adicional típica del otro paradigma de aprendizaje

• Una metodología exitosa para abordar el problema de SSC se basa en algoritmos tradicionales de clasificación supervisada. Estas técnicas tienen como objetivo obtener uno (o varios) conjuntos etiquetados ampliados, en función de sus predicciones más confiables, para clasificar datos no etiquetados. Denotamos estos algoritmos como técnicas autoetiquetadas