UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

título del TFG

Presentado por Mario Sanz Pérez
en Universidad de Burgos — 18 de enero de 2024
Tutor: Álvar Arnaiz González

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Mario Sanz Pérez, con DNI 71482918E, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 18 de enero de 2024

Vº. Bº. del Tutor:

D. Álvar Arnaiz González

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general iii

Índice de figuras iv

Índice de tablas v

1. Introducción 1

2. Objetivos del proyecto 3

3. Conceptos teóricos 5
- 3.1. Aprendizaje automático 5
- 3.2. Aprendizaje semi-supervisado 9
- 3.3. *Random Forest* 15
- 3.4. *Adaboost* 16
- 3.5. Diseño web 16

4. Técnicas y herramientas 17

5. Aspectos relevantes del desarrollo del proyecto 19

6. Trabajos relacionados 21

7. Conclusiones y Líneas de trabajo futuras 23

Bibliografía 25

Índice de figuras

3.1. Reducción de dimensionalidad 8
3.2. *Smoothness assumption* y *Low-density assumption* [8] 10
3.3. *Manifold assumption* [4] 10
3.4. *Cluster assumption* [4] 11
3.5. Clasificación de los diferentes algoritmos que pretenden incorporar datos no etiquetados a métodos de clasificación. Basado en [8] 12

Índice de tablas

3.1. Comparación aprendizaje supervisado y no supervisado [7]. . . 8

1. Introducción

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.

2. Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

3. Conceptos teóricos

En esta sección se resumirán los conceptos teóricos básicos y necesarios para comprender el trabajo. Principalmente se hablará de aprendizaje automático y luego se profundizará en el aprendizaje semi-supervisado.

3.1. Aprendizaje automático

El aprendizaje automático (*Machine Learning* en inglés) es el campo de la inteligencia artificial (IA) que se centra en el uso de datos y en el desarrollo de algoritmos para imitar la manera de aprender de los humanos [3]. La esencia radica en la capacidad de los sistemas informáticos para aprender de datos y realizar tareas sin intervención humana directa, si no descubriendo patrones y tendencias en los mismos. A estos sistemas se les conoce como **modelos**, los cuales pueden mejorar su rendimiento y adaptarse a nuevas situaciones basándose en la experiencia pasada.

Según [1], existen cuatro etapas principales en el desarrollo de un modelo. El primer paso consiste en seleccionar y preparar el conjunto de datos (*dataset*) que utilizará el modelo para aprender a resolver el problema para el que se ha diseñado. En el segundo paso se selecciona el algoritmo para ejecutar sobre el *dataset*. Este dependerá del tamaño y el tipo de los datos de entrada y del tipo de problema que se está resolviendo. El tercer paso consiste en entrenar el algoritmo hasta que la mayoría de los resultados sean los esperados. El cuarto y último paso trata de usar el modelo sobre nuevos datos y hacer una evaluación para una posible mejora.

Según los datos que se seleccionen en el primer paso, podemos tener dos ramas distintas en el aprendizaje automático [5]:

- **Predictiva**: también caracterizada por utilizar el aprendizaje supervisado, es decir, datos de entrada etiquetados.
- **Descriptiva**: al contrario, utiliza el aprendizaje no supervisado, con datos de entrada no etiquetados.

Aprendizaje supervisado

El aprendizaje supervisado es un tipo de aprendizaje automático en el que los modelos son entrenados utilizando conjuntos de datos etiquetados, en los que se basarán las decisiones y predicciones. Los conjuntos de datos contienen ejemplos emparejados de variables de entrada (o características) y de salida (o etiquetas). La esencia de este tipo de aprendizaje se basa en la capacidad del modelo para aprender la relación funcional entre las entradas y las salidas, permitiéndole hacer predicciones precisas sobre nuevos datos no vistos [2]. De ahí su clasificación como «predictiva» en la sección anterior. Se puede clasificar este tipo de aprendizaje en dos tipos:

- **Clasificación**: los modelos asignan categorías o clases a las entradas no etiquetadas. Dentro de este tipo se puede encontrar la clasificación binaria y la multi-clase. La primera se ve en un caso como la clasificación de correos electrónicos marcadas como spam o no spam (solo una etiqueta). Y la segunda se puede ver en cualquier ejemplo en el que haya mas de dos clases, como al establecer si un paciente tiene alto, medio o bajo riesgo de muerte ante una operación.
- **Regresión**: es similar a la clasificación, pero en vez de asignar un valor discreto, ahora es un valor continuo. Un ámbito común en el que se suele dar es en la economía, con la predicción de acciones o ventas.

También es importante comentar las principales fases que forman este aprendizaje y los posibles problemas o desafíos que pueden surgir, ya que pueden servir para tener en cuenta en los algoritmos concretos a implementar. En la mayoría de algoritmos que utilizan datos etiquetados, estos se dividen en tres conjuntos: entrenamiento, validación y prueba. El conjunto de entrenamiento se utiliza para ajustar los parámetros del modelo, el conjunto de validación para ajustar los hiperparámetros y prevenir el sobreajuste y el conjunto de prueba apra evaluar el rendimiento final. El sobreajuste o ***overfitting*** es uno de los principales problemas del aprendizaje automático y ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento, es decir, los memoriza en vez de generalizar.

Aprendizaje no supervisado

Para explicar este apredizaje se usará el artículo [7]. El aprendizaje no supervisado hace referencia a los tipos de problemas en los que se utiliza un modelo para caracterizar o extraer relaciones en los datos. A diferencia del aprendizaje supervisado, estos algoritmos descubren la estructura implícita de un conjunto de datos utilizando únicamente características de entrada y no clases o categorías. Ya que no existen etiquetas en los datos, los métodos no supervisados se utilizan normalmente para crear una representación concisa de los datos, posibilitando la generación de contenido creativo a partir de ellos. Por ejemplo, si tenemos una gran cantidad de fotografías sin clasificar, un modelo no supervisado encontraría relaciones entre las caracteristicas para poder organizar automáticamente las imágenes en grupos. Se pueden clasificar en tres diferentes categorías:

- **Clustering**: segmentación o agrupamiento. Consiste en la identificación de grupos o *clusters* en función de sus similitudes y diferencias. Dentro de este tipo, se puede diferenciar un agrupamiento exclusivo, donde los datos pertenencen a un unico grupo, y un agrupamiento superpuesto, donde los datos pueden perteneces a varias agrupaciones. El ejemplo de las fotografías entra dentro de esta categoría.
- **Reglas de asociación**: utiliza una medida de interés para obtener un conjunto de reglas sólidas que permitan descubrir asociaciones interesantes entre las características de un conjunto de datos. La principal aplicación es el «análisis de cestas de compra», que se usa para determinar los patrones de compra de los clientes en funcion de las relaciones entre productos.
- **Reducción de dimensionalidad**: estos algoritmos buscan reducir la complejidad de un conjunto de datos de alta dimensión a espacios de baja dimensión sin perder propiedades fundamentales de los datos originales. Este tipo de algoritmos se utiliza en la fase de análisis de datos, facilitando la representación gráfica. Se puede ver un ejemplo a continuación.

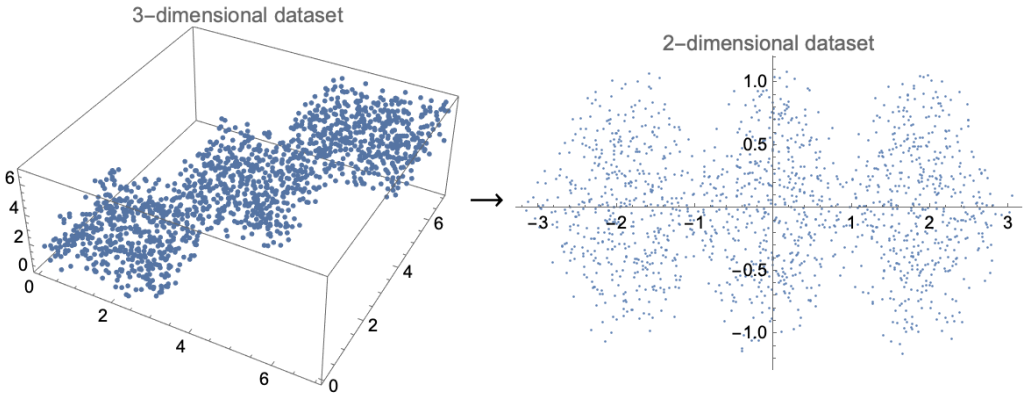

Figura 3.1: Reducción de dimensionalidad

En la siguiente tabla se resumen las principales diferencias entre aprendizaje supervisado y no supervisado:

	Supervisado	**No supervisado**
Objetivo	Aproximar una función que asigna entradas a salidas a partir de un conjunto de datos clasificados.	Crear una representación concisa de los datos, posibilitando la generación de contenido creativo a partir de ellos.
Complejidad	Complejidad simple.	Complejidad computacional mayor.
Entrada	Se conoce el número de clases (datos etiquetados).	No se conoce el número de clases (datos no etiquetados).
Salida	Genera un valor de salida esperado.	No se tienen valores de salida asociados
Tipos	Clasificación, Regresión	Clustering, Reglas de asociación, Reducción de dimensionalidad

Tabla 3.1: Comparación aprendizaje supervisado y no supervisado [7].

3.2. Aprendizaje semi-supervisado

Como el nombre sugiere, el aprendizaje semi-supervisado se encuentra entre los dos tipos vistos anteriormente. Los algoritmos dentro de esta estrategia se basan en extender cualquiera de los aprendizajes, supervisado o no supervisado, para añadir información adicional que el otro no proporciona [9].

Los métodos de clasificación semi-supervisada intentan utilizar puntos de datos no etiquetados para generar un modelo cuyo rendimiento supere el de los modelos obtenidos al utilizar solo datos etiquetados [8].
Por ejemplo, imaginemos que se esta trabajando en la clasificación de imágenes médicas para identificar diferentes tipos de enfermedades. En este caso, consideramos específicamente la detección temprana de ciertos tipos de cáncer a partir de imágenes de tomografías. En un enfoque supervisado, podríamos entrenar un modelo utilizando un conjunto de datos etiquetado que incluye imágenes con diagnósticos de cáncer y sin cáncer. Sin embargo, la obtención de un gran conjunto de datos etiquetado puede ser costosa y consume tiempo. En un escenario de aprendizaje semi-supervisado, además de los datos etiquetados, podríamos tener un conjunto de datos mucho más grande que incluye imágenes no etiquetadas. Algunas de estas pueden contener señales sutiles o características asociadas con el cáncer que no han sido previamente etiquetadas.
El modelo de aprendizaje semi-supervisado podría analizar estas imágenes no etiquetadas y descubrir patrones que podrian indicar la presencia temprana de cáncer. Por ejemplo, podría aprender a reconocer características microscopicas especificas de las imagenes que no son evidentes para el ojo humano. Cuando se encuentra con nuevas imágenes no etiquetadas que comparten estas características, el modelo podría clasificarlas como indicativas de la presencia de cáncer, incluso si no ha visto exactamente esas características en el conjunto de datos etiquetado. Existe una condición necesaria en el aprendizaje semi-supervisado: la distribución marginal subyacente $p(x)$ sobre el espacio de entrada debe contener información acerca de la distribución posterior $p(x|y)$ [8]. Es decir, la naturaleza de los datos no etiquetados debe contener información útil para inferir las etiquetas correspondientes. Esta suposición es básica y en la mayoría de los ejemplos se cumple. Aún asi, como la manera de interactuar entre $p(x)$ y $p(x|y)$ no es siempre la misma, se pueden tomar malas decisiones que conllevarían un rendimiento cada vez peor. Por esta razón, existen tres principales suposiciones que todo algoritmo semi-supervisado debe cumplir para funcionar correctamente.

- ***Smoothness assumption***: traducida como suposición de suavidad, consiste en que para dos puntos x_1 y x_2 que están cerca en una región densa, entonces sus correspodientes salidas (o etiquetas) y_1 y y_2 deben ser las mismas. Esto es útil sobretodo con datos no eitquetados, ya que por la propiedad transitiva, dos puntos que no estén relativamente cerca, pueden ser de la misma clase.

- ***Low-density assumption***: esta suposición está definida sobre la distribución de datos de entrada $p(x)$ y dice que el límite de decisión en la clasificación debe pasar antes por un área de poca densidad que por una de mayor densidad. Esto se puede observar en la figura 3.2.

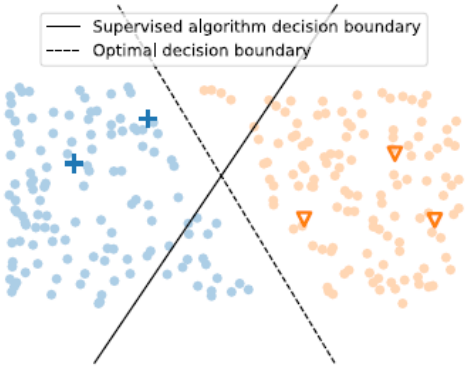

Figura 3.2: *Smoothness assumption* y *Low-density assumption* [8]

- ***Manifold assumption***: esta suposición afirma que los datos utilizados se encuentran en un *manifold* de baja dimensión incrustado en un espacio de mayor dimensión. En otras palabras, los datos, en lugar de proceder de cualquier parte del espacio, deben proceder de estos *manifolds* de dimensiones más bajas [4].

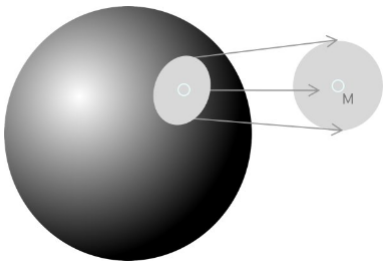

Figura 3.3: *Manifold assumption* [4]

En algunas ocasiones, aparece una cuarta suposición: ***cluster assumption***. Esta indica que dos datos que pertenecen a un mismo *cluster*, pertenecen también a la misma clase. Se tomará esta suposición como una generalización de las tres anteriores [8].

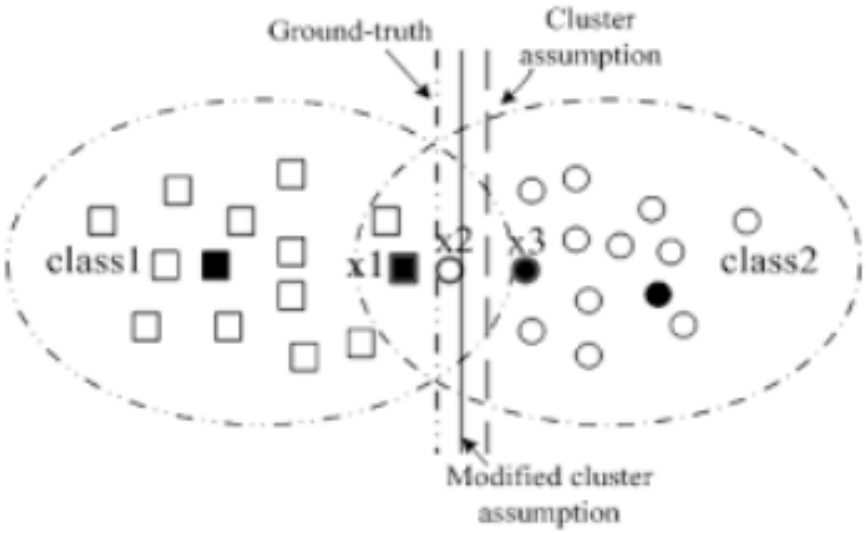

Figura 3.4: *Cluster assumption* [4]

.

No hay una clasificación oficial de algoritmos de aprendizaje semi-supervisado, pero si se pueden encontrar aproximaciones teniendo en cuenta las suposiciones en las que estan basadas los algoritmos y en como se relacionan con los algoritmos supervisados y no supervisados.

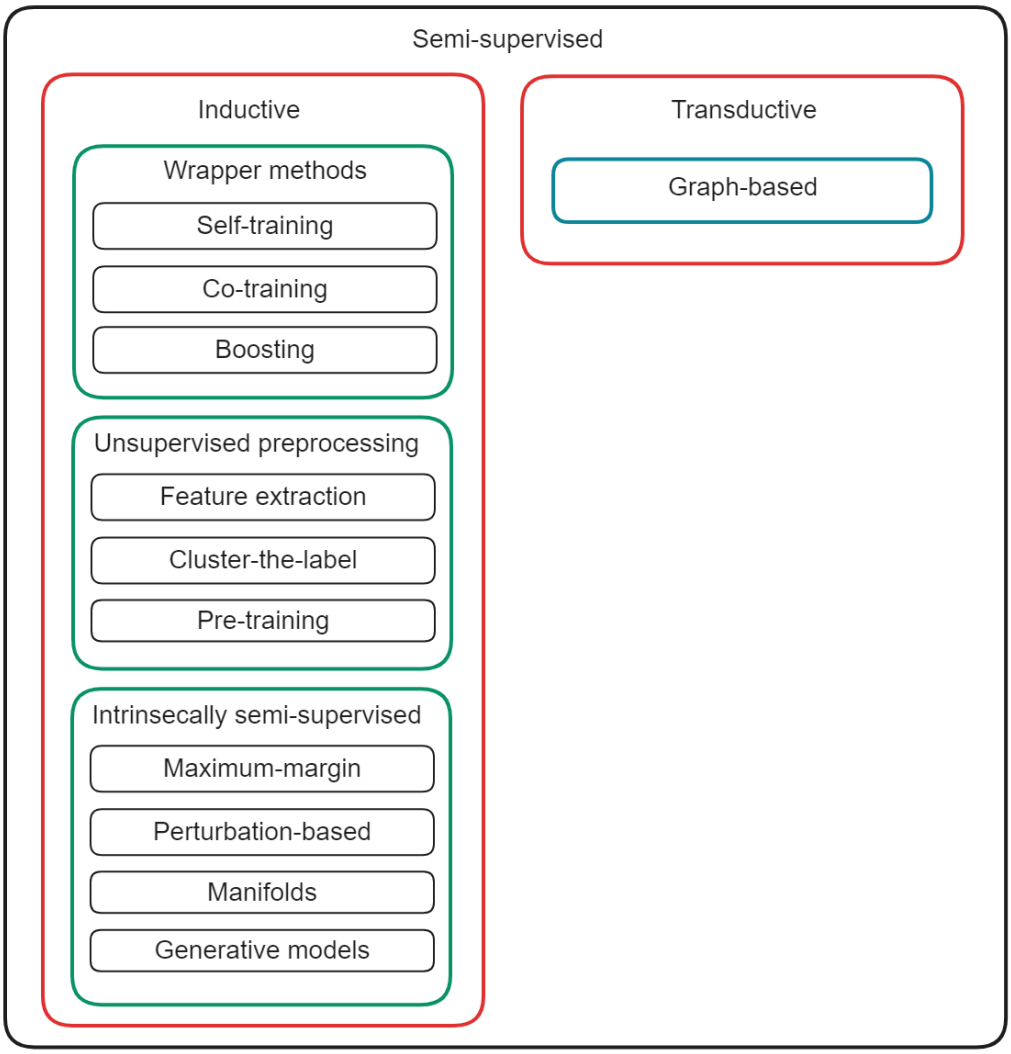

Figura 3.5: Clasificación de los diferentes algoritmos que pretenden incorporar datos no etiquetados a métodos de clasificación. Basado en [8]

.

Métodos inductivos

Los métodos inductivos pretenden construir un clasificador que pueda generar predicciones para cualquier objeto del espacio de entrada. En el entrenamiento de este clasificador o modelo se pueden utilizar datos no etiquetados, pero las predicciones cuando hay varios son independientes entre sí una vez finalizado el entrenamiento.

- ***Wrapper methods***: Estos métodos entrenan inicialmente clasificadores con datos etiquetados y luego utilizan las predicciones para generar datos adicionales etiquetados. Los clasificadores se vuelven a entrenar con estos datos pseudo-etiquetados.

- ***Unsupervised preprocessing***: Estos métodos extraen características útiles, pre-agrupan datos o determinan parámetros iniciales de aprendizaje de manera no supervisada, pero solo se aplican a datos originalmente etiquetados. Mejoran el rendimiento de clasificadores supervisados al utilizar información de datos no etiquetados durante la etapa de preprocesamiento.

- ***Intrinsecally semi-supervised***: Incorporan directamente datos no etiquetados en la función objetivo o procedimiento de optimización. Son extensiones de métodos supervisados al entorno semi-supervisado. Maximizan la información obtenida de datos no etiquetados durante el proceso de aprendizaje.

Métodos transductivos

A diferencia de los métodos inductivos, los métodos transductivos no construyen un clasificador para todo el espacio de entrada. En su lugar, su poder predictivo se limita exactamente a los objetos que encuentra durante la fase de entrenamiento. Por lo tanto, los métodos transductivos no tienen fases de entrenamiento y prueba distintas [8]. El aprendizaje transductivo puede ahorrar tiempo y es preferible cuando el objetivo se orienta a mejorar nuestro conocimiento sobre el conjunto de datos sin etiquetar. Sin embargo, como este escenario implica que el conocimiento del conjunto de datos etiquetados puede mejorar nuestro conocimiento del conjunto de datos sin etiquetar, no es ideal ni utilizable para procesos causales [4]. Los métodos transductivos suelen definir un grafo sobre todos los puntos de datos, etiquetados y no etiquetados, codificando la similitud entre pares de puntos de datos con aristas posiblemente ponderadas. Estos grafos se dividen en tres pasos: creación del grafo, ponderación del grafo (matriz de pesos) e inferencia (se refiere al proceso de asignar etiquetas o categorías a los nodos no etiquetados en un grafo utilizando la información de los nodos etiquetados y la estructura del grafo).

Construccion de grafos

El primer paso es la construcción de la matriz de adyacencia, que indica la presencia de aristas entre pares de nodos. Existen tres métodos posibles de construcción de grafos:

- ε neighbourhood: conecta cada nodo a todos los nodos a los que la distancia es como máximo ε. La medida de distancia normal suele ser euclídea. La estructura depende en gran medida de la elección de ε y de la medida de distancia.
- K - nearest neighbours: cada nodo se conecta a sus k vecinos más cercanos según alguna medida de distancia. Ocurre un problema que tiene dos soluciones: symmetric k nearest neighbours, que construye una arista si i o j estan en el "vecindario" de k y mutual k-nearest neighbours que la construye si ambos están en la k vecindad del otro.
- b-matching: el anterior método suele originar grafos en los que cada nodo no tiene k vecinos, lo que está demostrado que afecta al rendimiento del clasificador. El objetivo del b-matching es encontrar el subconjunto de aristas en el grafo completo de forma que cada nodo tenga grado b y la suma de los pesos de las aristas sea máxima.

El segundo paso de la construcción del grafo es ponderarlo (dar pesos a las aristas). En primer lugar, se construye una matriz de adyacencia completa utilizando una función k y después se obtiene la matriz de pesos W mediante sparsification (eliminar aristas de la matriz). Uno de los métodos más populares es el de ponderación de bordes gausiano. Otro es el de linear neighbourhood propagation (LNP) que se basa en que cualquier punto de datos pueda aproximarse como una combinación lineal de sus vecinos. El algoritmo LNP asume una estructura del grafo conocida y fija. Sin embargo, en lugar de fijar la estructura del grafo, también se puede inferir simultáneamente la estructura del grafo y los pesos de las aristas reconstruyendo linealmente los nodos basándose en todos los demás nodos.

Fase de inferencia

Existen diferentes maneras de llevar a cabo esta fase:

- Asignación de etiquetas duras: grafo min-cut: se añade un único nodo fuente v+, conectado con peso infinito a los puntos de datos positivos y un único nodo v- conectado con peso infinito a los puntos de

datos negativos. Por lo tanto, determinar el corte mínimo consiste en encontrar un conjunto de aristas con un peso combinado mínimo que, cuando se eliminan, dan como resultado un grafo sin rutas desde el nodo de origen hasta el noso de destino. Los nodos conectados a v+ se etiquetan como positivos y los conectados a v- como negativos. Este enfoque puede conducir a que casi todos los datos se etiqueten con la misma.

- Asignación probabilística de etiquetas: campos aleatorios de Markov: el anterior método solo produce etiquetas de clase y no probabilidades, lo que es una gran desventaja. La idea principal detrás de los CRF es calcular la probabilidad conjunta de todas las etiquetas dadas las observaciones y las relaciones entre etiquetas vecinas. Esto se hace utilizando una función de energía que mide cuán compatibles son las etiquetas y las características observadas. Luego, se utiliza una distribución exponencial para traducir la función de energía en una distribución de probabilidad.

- Asignación probabilística eficiente de etiquetas: campos aleatorios gaussianos: La principal mejora de los GRFs con respecto a los MRFs es que los GRFs permiten una estimación más eficiente de las probabilidades de asignación de etiquetas.

- Local and global consistency: La última propuesta no maneja bien el ruido de las etiquetas ya que las etiquetas verdaderas se fijan a los puntos de datos etiquetados. En segundo lugar, en los grafos irregulares, la influencia de los nodos con un grado alto es relativamente grande. La solución es el método LGC. El primer problema se resuelve penalizando el error cuadratico entre la etiqueta verdadera y la etiqueta estimada. El segundo problema se resuelve regularizando el termino de penalización para los puntos de datos no etiquetados mediante los grados de los nodos.

Ensembles

[6] Boosting, bagging

3.3. *Random Forest*

/// Ejemplo de estructura

3.4. *Adaboost*

3.5. Diseño web

4. Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

5. Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

6. Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

7. Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] DataScientest. Machine learning: definicion, funcionamiento, usos. https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos, 2021. [Internet; descargado 16-enero-2024].

[2] emeritus. What is supervised learning in machine learning? a comprehensive guide. https://emeritus.org/blog/ai-and-ml-supervised-learning/, 2023. [Internet; descargado 17-enero-2024].

[3] ibm. What is machine learning? https://www.ibm.com/topics/machine-learning, 2019. [Internet; descargado 15-enero-2024].

[4] Vidushi Meel. What is semi-supervised machine learning? a gentle introduction.

[5] César García Osorio and José Francisco Diez Pastor. *Aprendizaje automático. Introducción y problemas tipo*. Sistemas Inteligentes, 2022.

[6] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45, 2006.

[7] Kurtis Pykes. Introduction to unsupervised learning. https://www.datacamp.com/blog/introduction-to-unsupervised-learning, 2024. [Internet; descargado 17-enero-2024].

[8] Jesper E. van Engelen and Holger H. Hoos. A survey on semi supervised learning. *Machine Learning*, 109:373–400, 2020.

[9] Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan and Claypool, 2009.