
Universidad
Tecmilenio®

Universidad Tecmilenio

Aprendizaje Automático No Supervisado

Máquinas de vectores de soporte

Introducción

Las máquinas de vectores de soporte (SVM) tienen sus raíces en los métodos estadísticos pertenecientes al campo del reconocimiento de patrones. Este método fue originalmente diseñado para resolver problemas de clasificación binaria, pero, con el paso del tiempo, su aplicación se ha extendido a situaciones donde se requiere de una clasificación múltiple o un resultado numérico obtenido por regresión.

El entrenamiento de los modelos basados en SVM fue diseñado para sacar el máximo provecho de los datos matemáticos en los problemas binarios, por lo cual es asociado comúnmente con el proceso de maximizar la precisión para estas dos clases en particular.

En la actualidad continúa siendo un método muy utilizado para la detección de anomalías, la resolución de problemas de regresión, la categorización de texto, el análisis de series de tiempo o las aplicaciones de visión por computadora (siendo el método que mejor desempeño ha demostrado cuando se tiene una cantidad reducida de muestras para describir un problema).

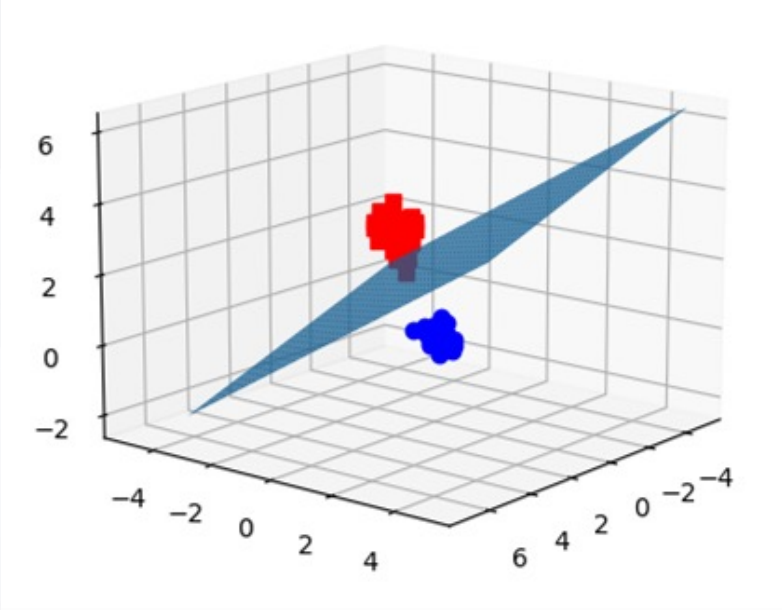
En este tema se dará un recorrido por la teoría en la que se sustentan las máquinas de vectores de soporte, sus diversas aplicaciones y la forma en que pueden ser implementadas.

Explicación

Teoría de las máquinas de soporte de vectores

Las máquinas de soporte vectorial tienen sus fundamentos en la utilización del concepto de hiperplano para clasificar conjuntos de datos según sus características. La implementación más simple de un hiperplano para realizar esta tarea es conocida como **clasificador de margen máximo** (Maximal Margin Classifier), el cual se utiliza principalmente para efectuar la clasificación cuando los datos son linealmente separables.

A saber, para comprender las bases de este método se requiere dominar la matemática relacionada con el álgebra lineal y la optimización de parámetros.

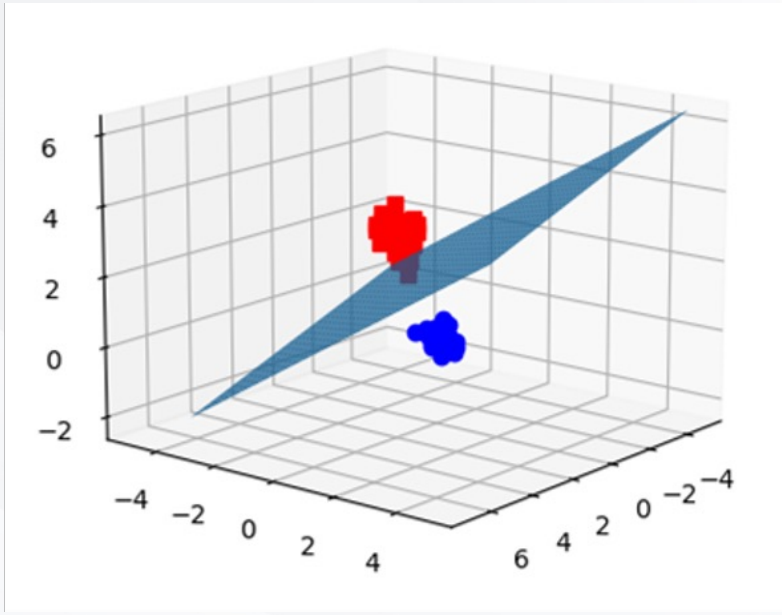


Explicación

Un hiperplano se define como un subespacio vectorial dimensional que no necesariamente pasa por el origen y que forma parte de un espacio de p+1 dimensiones.

Aunque esta descripción pudiera parecer muy compleja, es bastante fácil de comprender si asumimos, por ejemplo, que en un espacio de dos dimensiones un hiperplano sería una recta (subespacio de una dimensión) y que, análogamente, en un espacio tridimensional, un hiperplano es un plano convencional (subespacio de dos dimensiones).

A medida que se aumenta la dimensión, se hace más complejo de visualizar, no obstante, el principio base se mantiene igual.



Esta figura muestra un ejemplo de hiperplano de un espacio tridimensional.

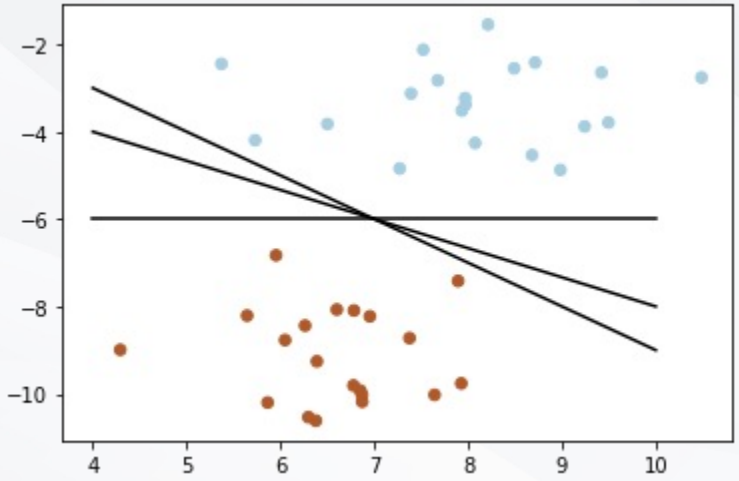
Explicación

Casos linealmente separables

Una situación que tiene una cantidad de observaciones, en donde cada una de ellas se defina por un número de predictores (cuya variable resultante posea dos categorías claramente definidas), será muy cómodo el empleo de hiperplanos para generar un sistema de clasificación que prediga al grupo al cual pertenece una observación específica, en función de sus predictores.

Entonces, un escenario con las características antes descritas describe un caso en donde las observaciones son linealmente separables.

Uno de los inconvenientes de la definición de hiperplano (para casos linealmente separables), consiste en la cantidad infinita de posibles hiperplanos que cumplen con la función de clasificación (ver la figura). Debido a esta situación se hace necesario definir un método general que permita seleccionar solamente uno de ellos como el clasificador óptimo.



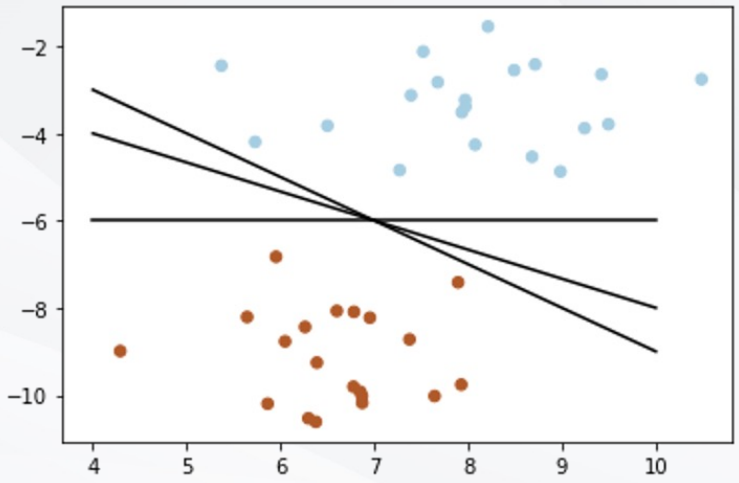
Explicación

Casos linealmente separables

La solución a este problema radica en la selección (como clasificador óptimo) del hiperplano que se encuentra más distante de todas las observaciones de entrenamiento. A este elemento en particular se le conoce como **hiperplano óptimo de separación** (*maximal margin hyperplane*).

Una de las primeras propuestas que se consideró para determinar cuál era el hiperplano óptimo, consistía en calcular la distancia perpendicular de cada observación con un hiperplano válido.

Si consideramos como margen a la menor de estas distancias, entonces el criterio para la selección del hiperplano óptimo de separación consiste en determinar cuál de todos esos posibles candidatos es el que garantiza un mayor margen.



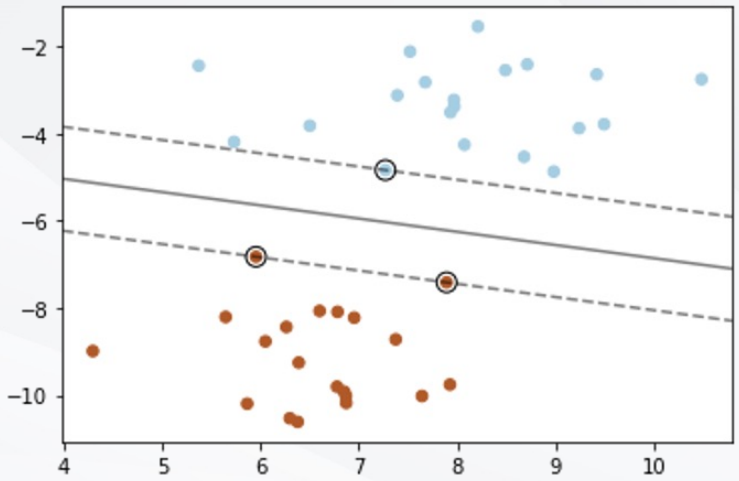
Explicación

Casos linealmente separables

De manera gráfica, el ***maximal margin hyperplane*** (MMH) se representa de la forma mostrada en la figura. En esta representación se muestra el hiperplano (línea negra continua) y su margen (las dos líneas discontinuas). Las tres observaciones equidistantes respecto al **hiperplano óptimo** (que se encuentran a lo largo de las líneas discontinuas) se les conoce como vectores soporte, ya que son los vectores del espacio/dimensional que lo soportan o definen.

Tal aproximación continúa presentando dos grandes limitaciones:

- Debido a que el hiperplano tiene que dividir perfectamente las observaciones, es muy sensible a variaciones en los datos de entrenamiento.
- Problemas de sobreajuste.



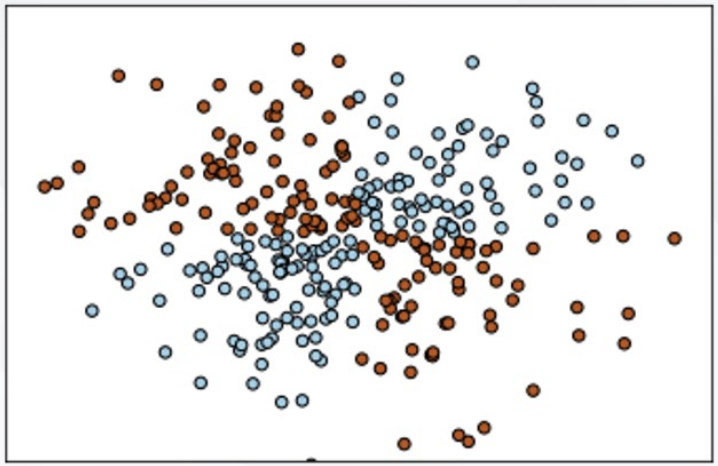
Al clasificador construido sobre el principio de hiperplano óptimo se le conoce como **clasificador de margen máximo** (*maximal margin classifier*).

Explicación

No-linealidad y uso de kernels

El **hiperplano óptimo** que se describió anteriormente es una forma muy simple y natural de clasificación, siempre y cuando las observaciones estén linealmente separadas. En la gran mayoría de casos de la vida real, los datos no cumplen esta condición de manera perfecta, por lo que no existe ningún hiperplano que los pueda separar (ver la figura).

Para atender este tipo de situaciones es posible extender el concepto de MMH, siendo posible encontrar un hiperplano que también divida las clases, pero que permita un cierto margen de error en la clasificación. A este tipo de especial de hiperplano se le denomina como **clasificador de vector de soporte**.



Explicación

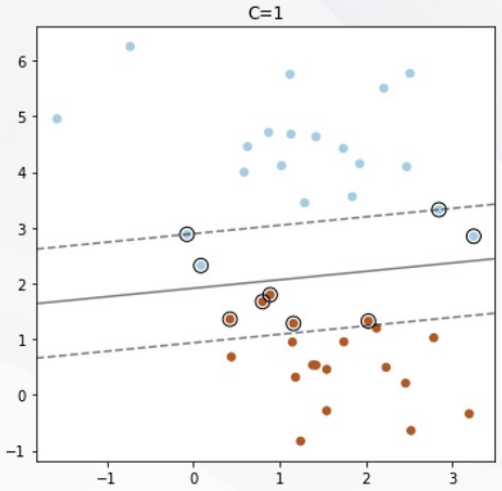
Clasificador de vector de soporte

Los clasificadores de vector de soporte, también conocidos como ***soft margin classifiers*** *o* ***support vector classifiers***, basan igualmente su función en la utilización de hiperplanos, pero, aunque no garantizan la perfecta separación de las diversas clases como el caso de MMH, son mucho más robustos y con mayor capacidad predictiva ante nuevas observaciones.

Para lograr este objetivo, en vez de buscar el margen de clasificación más ancho posible, hay que permitir que las observaciones tengan un cierto grado de flexibilidad, con respecto al margen e incluso al propio hiperplano.

En la figura se muestra un clasificador de vector de soporte, ajustado a un conjunto de observaciones.

La línea continua representa el hiperplano y las líneas discontinuas el margen correspondiente a cada lado. Como se puede apreciar, algunas de las muestras están dentro del margen, lo cual no incluye un error de clasificación.



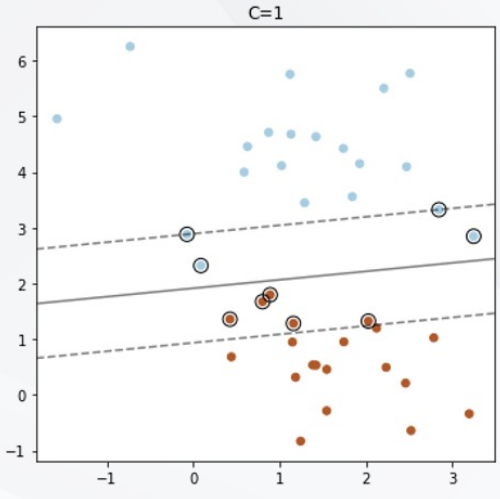
Explicación

Relevancia del hiperparámetro C

C es la forma mediante la cual se puede controlar la manera en que los clasificadores de vector de soporte manejan los errores, lo cual incluye el número y la severidad de las violaciones permitidas sobre el margen y el propio hiperplano, mismas que se pueden tolerar durante el proceso de ajuste. En este caso se definen dos condiciones principales:

- C ~ ∞: esto significa que no se permite ninguna violación del margen, por tanto, el resultado es equivalente al clasificador de margen máximo (considerando que esta solución solo es posible si las clases son perfectamente separables).

- C ≈ 0: en este caso, se penalizan menos los errores, por lo que más observaciones pueden estar en el lado incorrecto del margen o incluso del hiperplano.

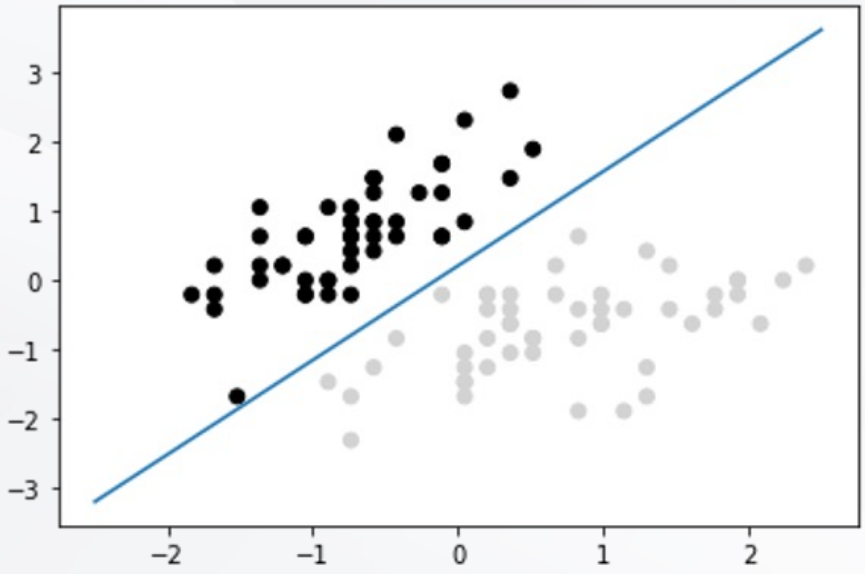


De forma general, se puede considerar a C como el hiperparámetro encargado de controlar el balance entre el sesgo (*bias*) y la varianza del modelo. En la práctica, la obtención de su valor óptimo se realiza mediante el proceso de validación cruzada.

Explicación

La librería Scikit Learn de Python permite implementar un clasificador de vector de soporte para resolver una problemática, por lo que es una tarea relativamente sencilla.

Un ejemplo muy cómodo para ilustrar este proceso es a partir del conjunto de datos de las características de la flor de iris, en donde a partir de la información de la longitud del pétalo y sépalo de la flor, se tienen tres categorías fácilmente distinguibles. La figura muestra de manera gráfica el hiperplano generado mediante el algoritmo de un clasificador de vector de soporte, a partir de los datos de entrenamiento.



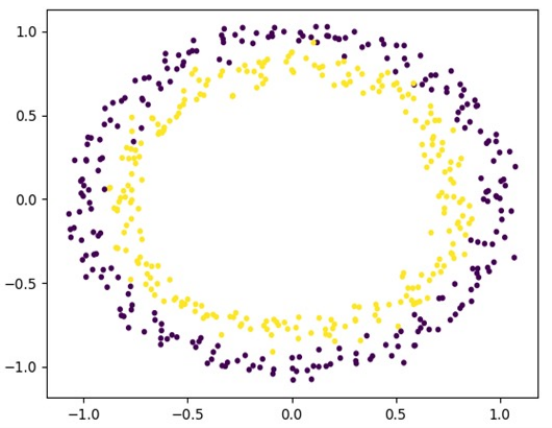
Explicación

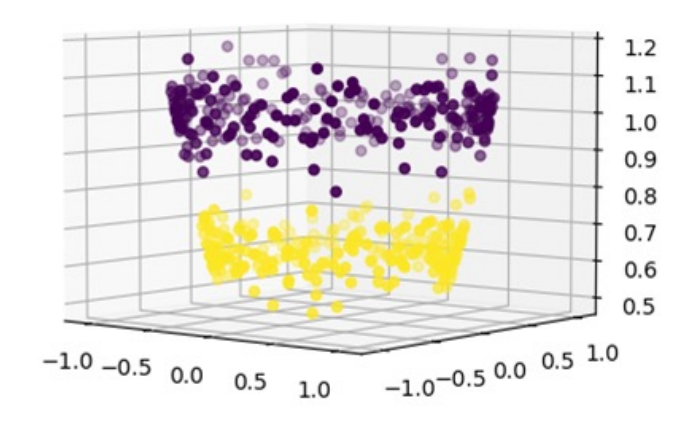
Máquinas de vector de soporte

El clasificador de soporte vectorial (descrito anteriormente) logra buenos resultados aceptables cuando el límite de separación entre clases es aproximadamente lineal.

En caso contrario, su capacidad de clasificación acertada disminuye considerablemente. Una estrategia muy útil para afrontar situaciones en donde la separación de los grupos se comporta de manera no lineal, radica en realizar una expansión de las dimensiones del espacio original.

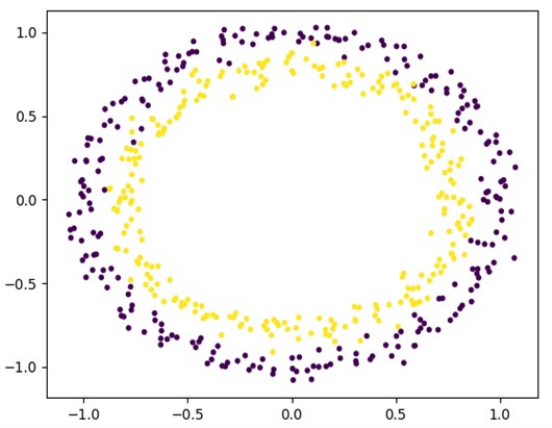
El hecho de que los grupos no sean linealmente separables (en el espacio original) no representa una limitación para que lo sean en un espacio de mayores dimensiones. Ambas figuras muestran ejemplos de estas distribuciones.

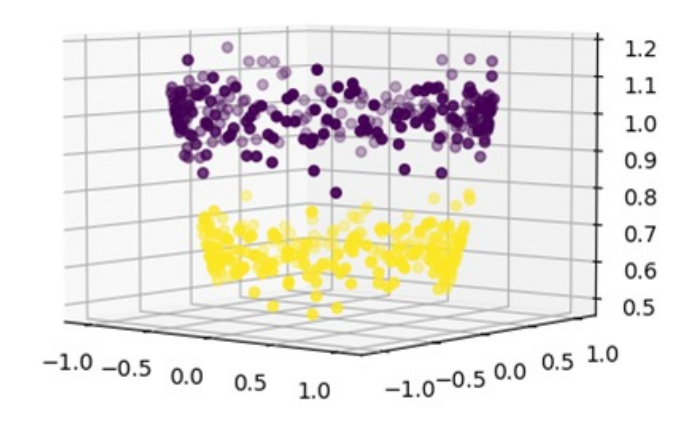




Explicación

Las **máquinas de vector de soporte** (SVM) poseen la capacidad de realizar esta transformación, por lo que se pueden considerar como una extensión del **clasificador de vector de soporte**, ya que incrementan la dimensión de los datos. Los límites de separación lineales generados en el espacio aumentado se convierten en límites de separación no lineales al proyectarlos en el espacio original.





Por otra parte, un **kernel** es una función que entrega como resultado el producto escalar entre dos vectores en un nuevo espacio dimensional, pero distinto del original en donde se encuentran dichos vectores.

- Kernel lineal: $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x} \cdot \boldsymbol{x}'$
- Kernel polinómico: $K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + c)^d$
- Kernel Gaussiano (RBF): $K(\boldsymbol{x}, \boldsymbol{x}') = exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$

Actividad

Después de haber estudiado el tema, aborda las siguientes cuestiones:

- ¿A qué otro tipo de conjunto de datos podrías aplicar el método de clasificación de vector de soporte?
- Investiga un ejemplo de aplicación que trabaje con un modelo de máquinas de vector de soporte.

Cierre

En la práctica, los datos no cumplen con la condición de ser linealmente separables, por lo cual se recurre a los clasificadores de vector de soporte que, a pesar de no garantizar la división perfecta de las clases, son mucho más robustos y con mayor capacidad predictiva ante nuevas observaciones. La flexibilidad de estos clasificadores se regula mediante el hiperparámetro, el cual se encarga de controlar el balance entre el sesgo y la varianza del modelo.

El hecho de que los grupos no sean linealmente separables en el espacio original no representa una limitación para que lo sean en un espacio de mayores dimensiones. Por ende, la ventaja de las máquinas de vector de soporte radica en que poseen la capacidad de realizar esta transformación, por lo que se pueden considerar como una extensión mejorada del clasificador de vector de soporte.

Por su parte, un kernel es una función matemática que entrega como resultado un producto escalar entre dos vectores en un nuevo espacio dimensional. Por lo tanto, dependiendo del tipo de problema que se esté tratando de resolver, se puede elegir entre los diversos tipos de kernels que existen, con el fin de mejorar la efectividad de la clasificación. Este recurso se conoce como truco del kernel, lo cual fue una de las ideas más revolucionarias en la aplicación de la inteligencia artificial y el aprendizaje automático en los últimos 20 años.

Universidad Tecmilenio

Aprendizaje Automático No Supervisado

Modelos probabilísticos

Introducción

Un gran grupo de modelos de aprendizaje automático se basan en la utilización de métodos algebraicos, gráficos o de cálculo para realizar sus predicciones. Sin embargo, existe otra enorme familia de modelos que entregan sus resultados haciendo uso de la teoría de las probabilidades.

Los modelos probabilísticos tienen una gran influencia en los fundamentos teóricos de Thomas Bayes, aplicados en la inteligencia artificial y el aprendizaje automático.

En este tema se estudiarán de forma particular los métodos probabilísticos, los cuales realizan sus operaciones a partir de la asignación de algún valor de incertidumbre a las variables desconocidas, así como alguna forma de certeza a las variables conocidas.

Los modelos probabilísticos se clasifican principalmente en dos tipos: generativos y discriminantes. Es importante destacar que la mayoría de los enfoques no probabilísticos pertenecen a la clase del segundo tipo. Por tanto, en el desarrollo de este tema, se verán los detalles específicos de cada uno de ellos.

Explicación

Para comprender más intuitivamente a los modelos probabilísticos, es necesario introducir algunos conceptos, por ejemplo, la entidad desconocida, llamada estado del sistema, coexiste junto con la entrada y la salida de este.

Por su parte, la acción de la entrada puede introducir algunos cambios en el estado del sistema, en donde ese cambio, junto con los mismos valores de entrada, se conjugan para influir en los valores de salida.

A partir de estos elementos se pueden definir a los modelos discriminativos como aquellos que intentan predecir los cambios en la salida, basándose únicamente en cambios en la entrada. Por su parte, los métodos generativos son aquellos que intentan modelar los cambios en la salida, en función de los cambios en la entrada y en el estado del sistema.

Explicación

Los **enfoques probabilísticos** (discriminantes y generativos) también se dividen con base en dos escuelas de pensamiento:

- Enfoque frecuentista (estimación de máxima verosimilitud).
- Enfoque bayesiano.

El enfoque de **estimación de máxima verosimilitud** o **MLE** aborda los problemas a partir de su valor nominal, parametrizando la información utilizando variables. Por lo tanto, los valores de estas variables (que maximizan la probabilidad de las observaciones) conducen a la solución del problema.

Explicación

El método **MLE** define la función de verosimilitud denotada como L(y | θ), la cual normalmente es la probabilidad conjunta de los parámetros y las variables observadas: L (y | θ) = P (y, θ). El objetivo es encontrar los valores óptimos para θ que maximice la función de verosimilitud dada por:

$$\theta^{MLE} = \arg\max_{\theta \in \Theta} \{L\ (y \mid \theta)\}$$

La cual también puede ser expresada como:

$$\theta^{MLE} = \arg\max_{\theta \in \Theta} \{P\ (y, \theta)\}$$

Este es un enfoque puramente frecuentista, es decir, que solo depende de los datos.

Explicación

El **enfoque bayesiano** ve el problema de una manera diferente. En este caso, todas las incógnitas se modelan como variables aleatorias con distribuciones de probabilidad previas conocidas. Por ende, si se denota la probabilidad previa condicional de observar la salida ***y*** para el vector de parámetros θ como $P(y \mid \theta)$, en donde las probabilidades marginales de estas variables son $P(y)$ y $P(\theta)$. Por consiguiente, la probabilidad conjunta de las variables se puede escribir en términos de probabilidades condicionales y marginales como:

$$P(y,\theta) = P(y \mid \theta) \cdot P(\theta)$$

La misma probabilidad conjunta se puede representar también de la siguiente forma:

$$P(y,\theta) = P(\theta \mid y) \cdot P(y)$$

La misma probabilidad conjunta se puede representar también de la siguiente forma:

$$P(\theta \mid y) \cdot P(y) = P(y \mid \theta) \cdot P(\theta)$$

Ahora, acomodando los términos, tenemos que:

$$P(\theta \mid y) = \frac{P(y \mid \theta) \cdot P(\theta)}{P(y)}$$

Explicación

La ecuación obtenida es el **teorema de Bayes** y es la base de todo el marco bayesiano. De forma general, este teorema da la relación entre la probabilidad *a posteriori* y la probabilidad *a priori* de una manera simple y elegante. Cada término en la ecuación anterior se le da un nombre específico: a $P(\theta)$ se le conoce como probabilidad *a priori*, en donde $P(y \mid \theta)$ es la verosimilitud; a la $P(y)$ se le denomina como evidencia, mientras que la $P(\theta \mid y)$ es la probabilidad posterior 0.

La **estimación bayesiana** se basa en maximizar la parte posterior. Por lo tanto, el problema de optimización basado en el teorema de Bayes se puede plantear de la siguiente manera:

$$\theta^{Bayes} = \arg\max_{\theta \in \Theta} \{P(\theta \mid y)\}$$

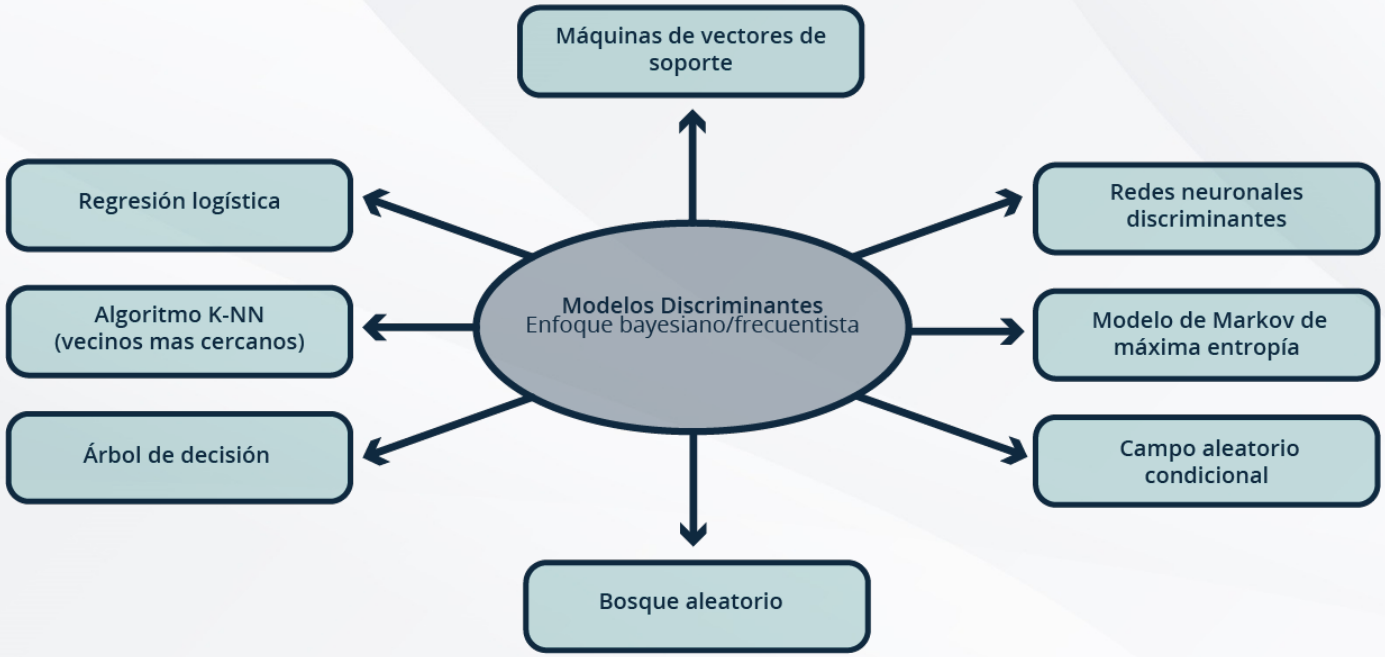
De forma expandida:

$$\theta^{Bayes} = \arg\max_{\theta \in \Theta} \left\{ \frac{P(y \mid \theta) \cdot P(\theta)}{P(y)} \right\}$$

Comparando esta ecuación con la expresión frecuentista, es fácil distinguir que el enfoque bayesiano agrega más información en forma de probabilidad *a priori*. En situaciones específicas, donde esta información está disponible, el enfoque bayesiano se convierte claramente en el preferido.

Explicación

Los **modelos discriminantes** aprenden a partir de la probabilidad condicional P(Y|X), mapeando directamente la variable no observada X (objetivo) en una etiqueta de clase Y. En algunas bibliografías también se les denomina como modelos condicionales, pero en ciertas ocasiones se puede distinguir una ligera diferencia entre ambas aproximaciones.



Explicación

Los **modelos generativos**, como su nombre lo indica, intentan comprender cómo son generados los datos que están analizando. Se encargan de buscar patrones en los elementos de entrenamiento para que, en caso de ser necesario, sean capaces de construir datos nuevos muy parecidos a los originales. De una manera más formal, se puede plantear que, dada una variable observable X y una variable objetivo Y , un modelo generativo es un modelo estadístico de la distribución de probabilidad conjunta P(X,Y).

Se pueden clasificar en dos grupos principales:

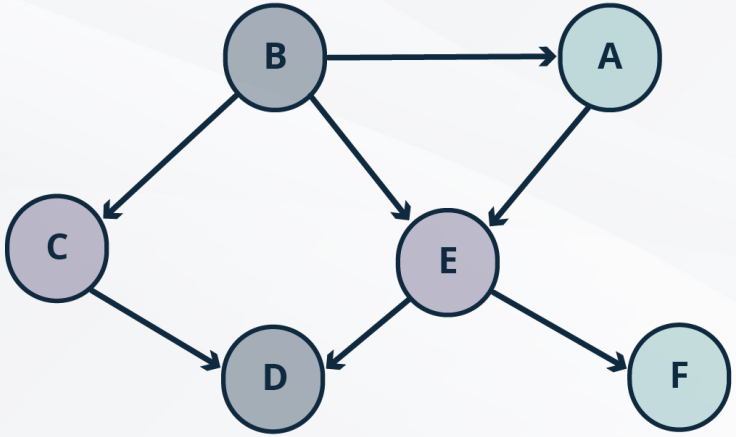
- Modelos clásicos.
- Modelos basados en aprendizaje profundo.

Explicación

Con el surgimiento del aprendizaje profundo se ha desarrollado una gran cantidad de métodos generativos, denominados como **modelos generativos profundos (DGM)**. El gran impacto de estas propuestas radica en la combinación de los modelos generativos clásicos y las redes neuronales profundas.

Por su parte, entre los **modelos generativos clásicos** o más comunes se pueden encontrar los siguientes:

- **Modelos mixtos:** uno de los aspectos fundamentales de los modelos generativos es comprender la composición de las entradas, y cómo es que se generaron dichos datos en primer lugar.
- **Redes bayesianas:** las redes bayesianas representan gráficos acíclicos dirigidos (ver la figura), donde cada nodo representa una variable observable o un estado, mientras que los extremos representan las dependencias condicionales entre los nodos.



Explicación

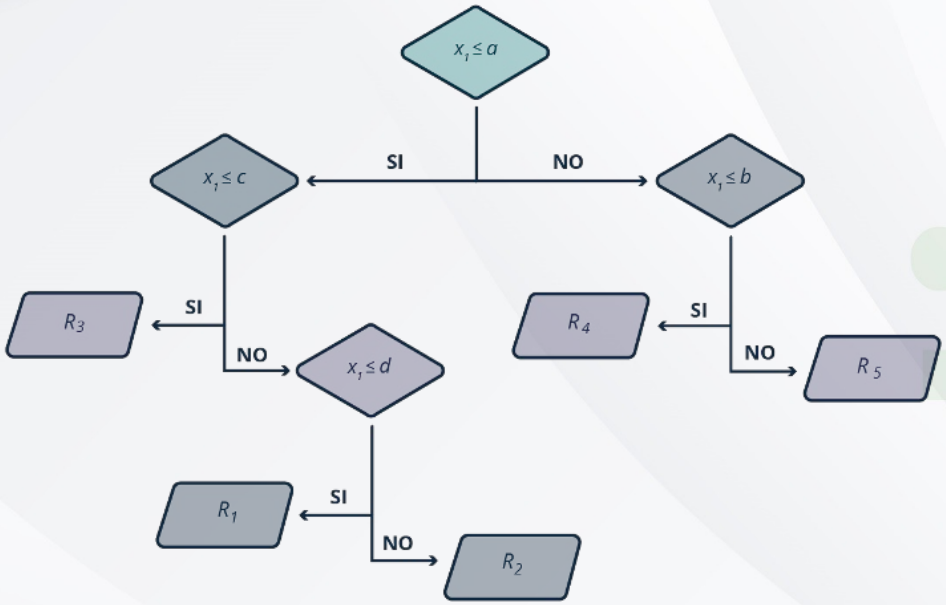
Si definimos el problema de optimización, en función de minimizar el error cuadrático medio:

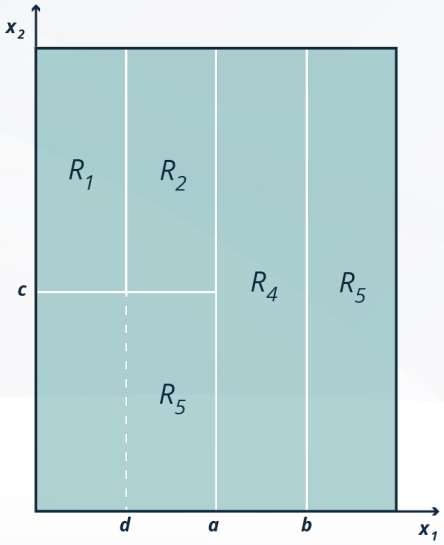
$$\sum_{i=1}^{i=p}(y_i - t(x_i)^2)$$

Entonces, el cálculo que mostraría el valor de la estimación dada por r_k viene dado por:

$$r_k = prom(y_i | x_i \in R_k)$$

Resolver el problema de encontrar las regiones óptimas globales para minimizar el error cuadrático medio es un problema muy complejo, por lo que no se puede resolver de manera general con los métodos de tiempo finito.





Explicación

El algoritmo Naive Bayes es un algoritmo de aprendizaje supervisado que constituye un modelo generativo clásico. Tiene su fundamentación sobre el teorema de Bayes y se utiliza comúnmente para resolver problemas de clasificación. A saber, ha tenido mucho éxito resolviendo problemáticas de clasificación de texto, es decir, cuando el conjunto de datos de entrenamiento posee una alta dimensión.

Debido a la variedad de modelos de Naive Bayes que existen, se ha llegado a considerar como todo un marco de trabajo. A continuación, se presentan tres de los más conocidos:

• **Naive Bayes gaussiano:** el modelo gaussiano se distingue por considerar que las características de los datos de entrada describen una distribución normal.

• **Naive Bayes multinomial:** este modelo se utiliza principalmente cuando los datos se distribuyen multinomialmente, por lo que es muy adecuado para la clasificación con características discretas..

• **Naive Bayes Bernoulli:** el clasificador de Bernoulli funciona de manera similar al clasificador multinomial, pero en este caso las variables predictoras son los valores booleanos autónomos.

Explicación

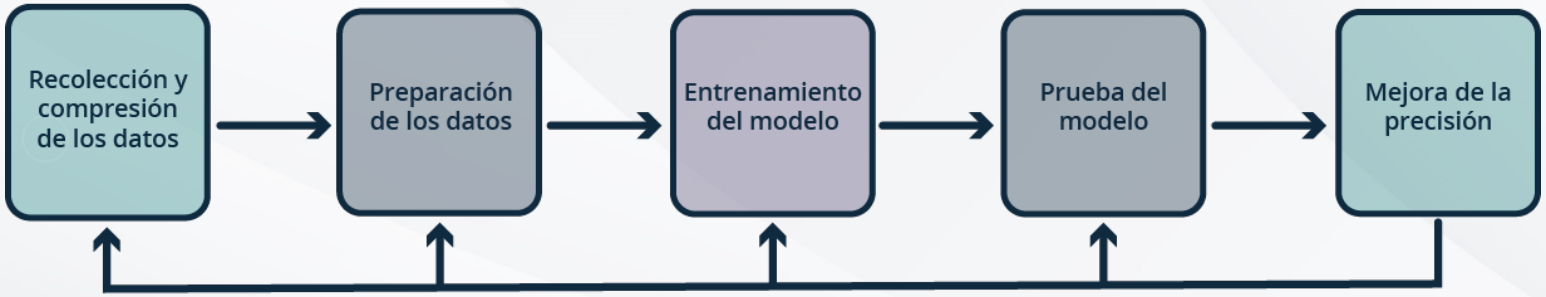
Los modelos de Naive Bayes son muy robustos y versátiles, pero también tienen sus puntos débiles. En la siguiente tabla se muestra un resumen de las fortalezas y debilidades que se necesitan considerar sobre esta familia de algoritmos para utilizarlos correctamente.

Fortalezas	Debilidades
• En problemas de clasificación binaria o multiclase, constituyen una manera fácil y rápida de realizar la predicción. • Cuando haya casos en los que se pueda asumir la independencia de los datos de entrada, este tipo de algoritmo se comporta mejor que otros modelos de clasificación, incluso en la ausencia de grandes cantidades de valores para el entrenamiento. • Al considerar de forma independiente la distribución de las características condicionales que posee cada clase, estas pueden ser estimadas como si tuvieran una sola dimensión. Esto es una gran ventaja sobre los problemas derivados de la alta dimensionalidad, mejorando el rendimiento de manera sustancial.	• Aunque son unos clasificadores bastante buenos, también son reconocidos por ser pobres estimadores. Esto implica que no se deben considerar rigurosamente los valores de las probabilidades que se obtienen con ellos. • Debido a la inocente presunción de independencia en los datos, estos casi nunca reflejan las verdaderas características que poseen en el mundo real. • Cuando el conjunto de datos de prueba tiene una característica que no ha sido considerada en el conjunto de entrenamiento, el modelo le asignará una probabilidad igual a cero, por lo que será inútil para realizar predicciones. Aplicar alguna técnica de suavizado es uno de los métodos que se utiliza para evitar esta situación.

Explicación

Construir un clasificador de Naive Bayes es un proceso muy sencillo con el apoyo de Python y la librería Scikit-learn, la cual incluye dentro de sus recursos varios conjuntos de datos de prueba para realizar algunos experimentos interesantes.

El programa se realiza siguiendo el flujo de trabajo para diseñar de una aplicación de aprendizaje automático. Por ejemplo, en la siguiente figura se muestra un clasificador de Naive Bayses:



El criterio para seleccionar entre los diversos modelos probabilísticos dependerá de los detalles específicos de cada aplicación, lo cual decidirá el enfoque más adecuado para su caso en particular.

Actividad

Piensa en otro problema en donde puedas aplicar un modelo probabilístico visto durante el tema.

Revisa el algoritmo del modelo e idea un plan para aplicarlo al problema que escogiste, con el fin de encontrar una solución.

Cierre

En este tema aprendiste sobre los modelos probabilísticos y sus dos principales tipos: los discriminantes y los generativos, donde los primeros se encargan de modelar la probabilidad condicional y los segundos la probabilidad conjunta. Cada uno de estos se puede abordar desde un enfoque frecuentista, a partir de la estimación de máxima verosimilitud (MLE), o bien, desde un enfoque bayesiano con la estimación de probabilidad máxima (MAP).

Los modelos discriminantes estiman directamente las probabilidades posteriores de un evento, mostrando un mejor rendimiento en relación a otros similares, ya que centran la utilización de los recursos computacionales en una tarea determinada.

Por su parte, los modelos generativos se encargan de buscar patrones en los elementos de entrenamiento, con el fin de que sean capaces, en caso de ser necesario, de construir datos nuevos similares a los originales.
Para seleccionar entre los modelos probabilísticos es importante considerar que los modelos generativos requieren una gran cantidad de suposiciones para resolver un problema. Por ende, en cuestiones de precisión, modelar las funciones de densidad condicional usando un modelo discriminante puede ser la mejor opción.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, estas se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.
