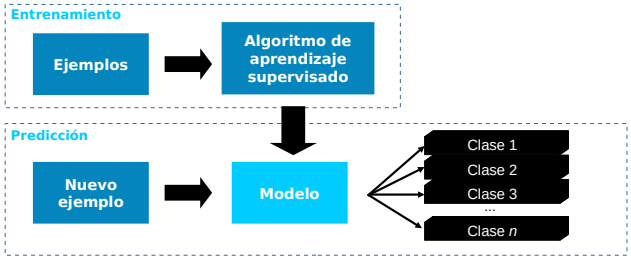
Definición

Aprendizaje Supervisado



Definición

Aprendizaje Supervisado

Algoritmos:

- Vecinos más cercanos
- Árboles de decisión
- Clasificación Bayesiana (Naïve Bayes)
- SVMs
- Redes neuronales
- Deep learning
- Ensembles (combinación de clasificadores base)

Algoritmos

k-NN

Aprendizaje basado en Instancias (Instance-based Learning o Memory-based Learning) es una familia de algoritmos de aprendizaje que, en lugar de realizar generalizaciones, compara nuevas instancias (ejemplos) con las instancias vistas en el entrenamiento, las cuales almacena en memoria

Algoritmos

k-NN

Asumiendo que cada ejemplo puede representarse como un punto en un espacio *n*-dimensional, para un ejemplo en particular los *k* ejemplos más cercanos en el espacio son sus **vecinos más cercanos**

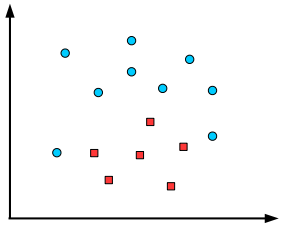
k- Nearest Neighbors: almacena todos los ejemplos disponibles y predice la clase de nuevos ejemplos en base a la **similitud** con ellos:

- Para **clasificación**: el ejemplo se asigna a la clase más frecuente entre sus *k* vecinos más cercanos (votación por mayoría)
- Para **regresión**: promedio de los valores de los *k* vecinos más cercanos

Algoritmos

k-NN

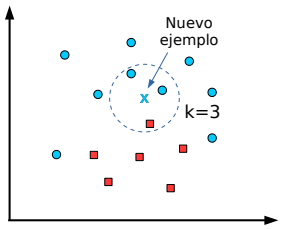
Entrenamiento



Algoritmos

k-NN

Predicción



Algoritmos

k-NN

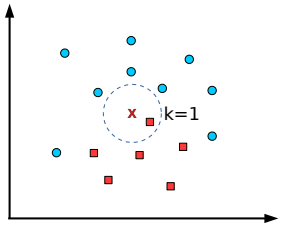
k-NN requiere calcular la similitud/distancia entre ejemplos:

- La elección de la medida depende del tipo de **características**
- Para valores reales es común la distancia Euclideana (las características tienen que estar en la misma escala o normalizadas)
- Para valores binarios se puede usar la distancia de Hamming, por ejemplo, que cuenta el número de discrepancias
- Cuando hay características de distintos tipos se pueden usar medidas mixtas (por ejemplo, Euclidiana y Hamming)
- Cada característica podría tener un peso asociado en la medida de similitud

Algoritmos

k-NN

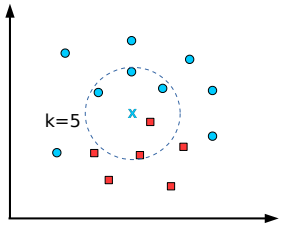
Predicción



Algoritmos

k-NN

Predicción

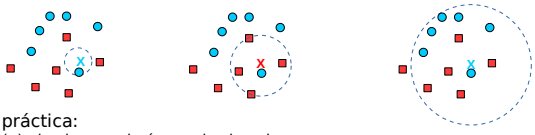


Algoritmos

k-NN

Cómo seleccionar el valor de *k*?

- Un valor de *k* demasiado pequeño es sensitivo a ruido
- Un *k* más grande puede funcionar bien
- Un *k* demasiado va a incluir la mayoría de ejemplos de otra clase



En la práctica:
k<sqrt(*n*), donde *n* es el número de ejemplos
k sea impar para clasificación binaria y no múltiplo del número de clases en otro caso

Algoritmos

k-NN

Entrenamiento

- Almacenar los ejemplos de entrenamiento

Predicción

- Calcular la distancia del nuevo ejemplo con todos los ejemplos de entrenamiento
- Identificar los *k* vecinos más cercanos
- Usar las clases de los *k* vecinos más cercanos para determinar la clase del nuevo ejemplo (voto por mayoría)

Algoritmos

k-NN

Pros

- Simple e intuitivo
- No requiere construir un modelo explícito, rápido para entrenamiento
  → **Lazy Learning**
- Buena precisión si el número de ejemplos es lo suficientemente grande
- Capaz de hacer una predicción aún con pocos ejemplos, si estos son representativos
- Incremental

Algoritmos

k-NN

Contras

- Requiere espacio de almacenamiento para todo el conjunto de entrenamiento
- Es sensible a ruido en los ejemplo y características, también a la presencia de *outliers*
- Lento para predicción, necesita comparar con todos los ejemplos de entrenamiento
- Desbalance en tiempos de entrenamiento/predicción
  → Requerimientos del dominio (por ejemplo, clasificación de *spam*)

Algoritmos

k-NN

Variaciones de *k*-NN

- Mejorar los tiempos de cómputo:
  - Mejorar las estructuras de datos para una búsqueda más rápida de los vecinos (indexación)
  - Puede ser suficiente con aproximar los vecinos más cercanos
- Reducir la necesidad de almacenamiento:
  - Mantener un subconjunto de los ejemplos de entrenamiento (representativos)

Algoritmos

Centroide más cercano

Clasificación basada en centroides:

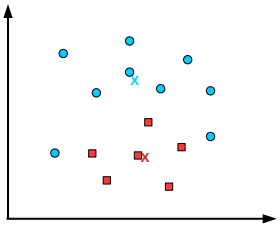
- Para cada clase se calcula una instancia **prototipo** sumando los ejemplos de entrenamiento de esa clase
- Durante la clasificación se asigna la clase del prototipo más cercano en base a una medida de distancia/similitud

Rocchio: inicialmente utilizado como un método de feedback de relevancia, se adaptó luego para clasificación de textos

Algoritmos

Centroide más cercano

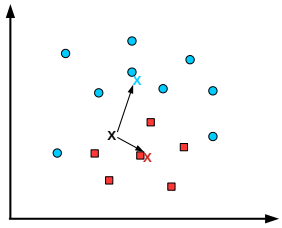
Entrenamiento



Algoritmos

Centroide más cercano

Predicción



Algoritmos

Centroide más cercano

Entrenamiento

- Construir un prototipo o centroide para cada clase c_i con todos los ejemplos de entrenamiento pertenecientes a c_i

Predicción

- Calcular la distancia del nuevo ejemplo con los centroides de todas las clases
- Asignar al ejemplo la clase del centroide más cercano

Algoritmos

Centroide más cercano

Pros

- Crea una representación simple para cada clase, el centroide
- La clasificación es basada en la distancia al centroide
- Muy usada en clasificación de textos
- Eficiente en tiempos de entrenamiento y clasificación
- Incremental

Contras

- Menos preciso que otros algoritmos
- Sensible a ruido y *outliers*

Algoritmos

Árboles de Decisión

Inducción de árboles de decisión:

El algoritmos de aprendizaje construye un árbol de decisión que representa la relación existente entre la clase y sus atributos

Es decir, se produce un proceso de generalización de forma que el árbol de decisión generado clasifica correctamente los ejemplos dados

Algoritmos

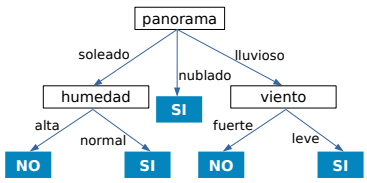
Árboles de Decisión

Los algoritmos de **inducción de árboles de decisión** generan estructuras en forma de árbol donde:

- Los nodos corresponden a atributos o características
- Las ramas saliendo de un nodo son evaluaciones sobre el valor de un nodo
- Las hojas son clases o categorías

Algoritmos

Árboles de Decisión



Algoritmos

Árboles de Decisión

El algoritmos básico de construcción de árboles de decisión (divide y conquista):

- El árbol de construye de manera top-down, particionando recursivamente
- Evaluar cada característica para determinar que tan buena es para dividir los ejemplos de entrenamiento
- Elegir el mejor atributo y crear una rama para cada uno de sus valores posibles
- Calcular los ejemplos en cada rama
- Asignar una clase si es posible o repetir los pasos anteriores con los restantes atributos

Algoritmos

Árboles de Decisión

Entrenamiento

- Construir el árbol de decisión

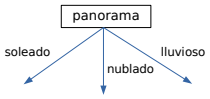
Predicción

- Usar el árbol para tomar decisiones de acuerdo a los valores de los atributos del ejemplo que se quiere clasificar
- Asignar al ejemplo la clase de la hoja del árbol alcanzada

Algoritmos

Árboles de Decisión

Día	Panorama	Temperatura	Humedad	Viento	Juega tenis?
1	Soleado	Caluroso	Alta	Leve	NO
2	Soleado	Caluroso	Alta	Fuerte	NO
3	Nublado	Caluroso	Alta	Leve	SI
4	Lluvioso	Templado	Alta	Leve	SI
5	Lluvioso	Frio	Normal	Leve	SI
6	Lluvioso	Frio	Normal	Fuerte	NO
7	Nublado	Frio	Normal	Fuerte	SI
8	Soleado	Templado	Alta	Leve	NO
9	Soleado	Frio	Normal	Leve	SI
10	Lluvioso	Templado	Normal	Leve	SI
11	Soleado	Templado	Normal	Fuerte	SI
12	Nublado	Templado	Alta	Fuerte	SI
13	Nublado	Caluroso	Normal	Leve	SI
14	Lluvioso	Templado	Alta	Fuerte	NO

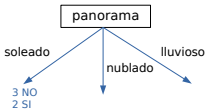


Entrenamiento

Algoritmos

Árboles de Decisión

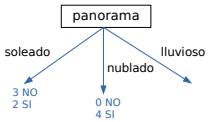
Día	Panorama	Temperatura	Humedad	Viento	Juega tenis?
1	Soleado	Caluroso	Alta	Leve	NO
2	Soleado	Caluroso	Alta	Fuerte	NO
3	Nublado	Caluroso	Alta	Leve	SI
4	Lluvioso	Templado	Alta	Leve	SI
5	Lluvioso	Frio	Normal	Leve	SI
6	Lluvioso	Frio	Normal	Fuerte	NO
7	Nublado	Frio	Normal	Fuerte	SI
8	Soleado	Templado	Alta	Leve	NO
9	Soleado	Frio	Normal	Leve	SI
10	Lluvioso	Templado	Normal	Leve	SI
11	Soleado	Templado	Normal	Fuerte	SI
12	Nublado	Templado	Alta	Fuerte	SI
13	Nublado	Caluroso	Normal	Leve	SI
14	Lluvioso	Templado	Alta	Fuerte	NO



Algoritmos

Árboles de Decisión

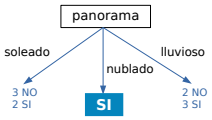
Día	Panorama	Temperatura	Humedad	Viento	Juega tenis?
1	Soleado	Caluroso	Alta	Leve	NO
2	Soleado	Caluroso	Alta	Fuerte	NO
3	Nublado	Caluroso	Alta	Leve	SI
4	Lluvioso	Templado	Alta	Leve	SI
5	Lluvioso	Frio	Normal	Leve	SI
6	Lluvioso	Frio	Normal	Fuerte	NO
7	Nublado	Frio	Normal	Fuerte	SI
8	Soleado	Templado	Alta	Leve	NO
9	Soleado	Frio	Normal	Leve	SI
10	Lluvioso	Templado	Normal	Leve	SI
11	Soleado	Templado	Normal	Fuerte	SI
12	Nublado	Templado	Alta	Fuerte	SI
13	Nublado	Caluroso	Normal	Leve	SI
14	Lluvioso	Templado	Alta	Fuerte	NO



Algoritmos

Árboles de Decisión

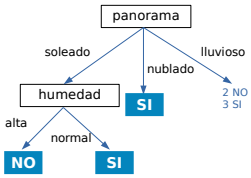
Día	Panorama	Temperatura	Humedad	Viento	Juega tenis?
1	Soleado	Caluroso	Alta	Leve	NO
2	Soleado	Caluroso	Alta	Fuerte	NO
3	Nublado	Caluroso	Alta	Leve	SI
4	Lluvioso	Templado	Alta	Leve	SI
5	Lluvioso	Frio	Normal	Leve	SI
6	Lluvioso	Frio	Normal	Fuerte	NO
7	Nublado	Frio	Normal	Fuerte	SI
8	Soleado	Templado	Alta	Leve	NO
9	Soleado	Frio	Normal	Leve	SI
10	Lluvioso	Templado	Normal	Leve	SI
11	Soleado	Templado	Normal	Fuerte	SI
12	Nublado	Templado	Alta	Fuerte	SI
13	Nublado	Caluroso	Normal	Leve	SI
14	Lluvioso	Templado	Alta	Fuerte	NO



Algoritmos

Árboles de Decisión

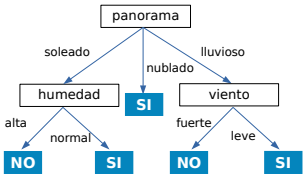
Día	Panorama	Temperatura	Humedad	Viento	Juega tenis?
1	Soleado	Caluroso	Alta	Leve	NO
2	Soleado	Caluroso	Alta	Fuerte	NO
3	Nublado	Caluroso	Alta	Leve	SI
4	Lluvioso	Templado	Alta	Leve	SI
5	Lluvioso	Frio	Normal	Leve	SI
6	Lluvioso	Frio	Normal	Fuerte	NO
7	Nublado	Frio	Normal	Fuerte	SI
8	Soleado	Templado	Alta	Leve	NO
9	Soleado	Frio	Normal	Leve	SI
10	Lluvioso	Templado	Normal	Leve	SI
11	Soleado	Templado	Normal	Fuerte	SI
12	Nublado	Templado	Alta	Fuerte	SI
13	Nublado	Caluroso	Normal	Leve	SI
14	Lluvioso	Templado	Alta	Fuerte	NO



Algoritmos

Árboles de Decisión

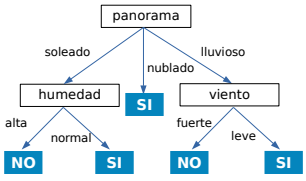
Día	Panorama	Temperatura	Humedad	Viento	Juega tenis?
1	Soleado	Caluroso	Alta	Leve	NO
2	Soleado	Caluroso	Alta	Fuerte	NO
3	Nublado	Caluroso	Alta	Leve	SI
4	Lluvioso	Templado	Alta	Leve	SI
5	Lluvioso	Frio	Normal	Leve	SI
6	Lluvioso	Frio	Normal	Fuerte	NO
7	Nublado	Frio	Normal	Fuerte	SI
8	Soleado	Templado	Alta	Leve	NO
9	Soleado	Frio	Normal	Leve	SI
10	Lluvioso	Templado	Normal	Leve	SI
11	Soleado	Templado	Normal	Fuerte	SI
12	Nublado	Templado	Alta	Fuerte	SI
13	Nublado	Caluroso	Normal	Leve	SI
14	Lluvioso	Templado	Alta	Fuerte	NO



Algoritmos

Árboles de Decisión

Día	Panorama	Temperatura	Humedad	Viento	Juega tenis?
1	Soleado	Caluroso	Alta	Leve	NO
2	Soleado	Caluroso	Alta	Fuerte	NO
3	Nublado	Caluroso	Alta	Leve	SI
4	Lluvioso	Templado	Alta	Leve	SI
5	Lluvioso	Frio	Normal	Leve	SI
6	Lluvioso	Frio	Normal	Fuerte	NO
7	Nublado	Frio	Normal	Fuerte	SI
8	Soleado	Templado	Alta	Leve	NO
9	Soleado	Frio	Normal	Leve	SI
10	Lluvioso	Templado	Normal	Leve	SI
11	Soleado	Templado	Normal	Fuerte	SI
12	Nublado	Templado	Alta	Fuerte	SI
13	Nublado	Caluroso	Normal	Leve	SI
14	Lluvioso	Templado	Alta	Fuerte	NO



Algoritmos

Árboles de Decisión

Qué atributo divide mejor los ejemplos?

- Una buena división da mayores certezas sobre la clasificación que antes de hacerla
- **Entropía**: es una medida que caracteriza la (in)pureza de una colección ejemplos

$$E(S) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

donde S un conjunto de ejemplos y p_i la proporción de ejemplos de S que pertenecen a la clase i

Algoritmos

Árboles de Decisión

Qué atributo divide mejor los ejemplos?

- **Information Gain**: la ganancia de información de un atributo A es la reducción esperada en la entropía de la colección S al particionar los ejemplos por tal atributo

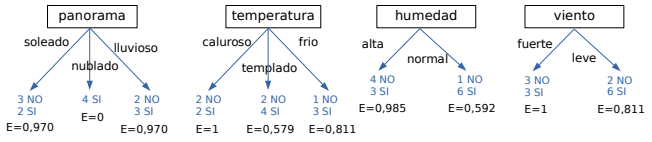
$$IG(S,A) = E(S) - \sum_{v \in values\ of\ A} \frac{|S_v|}{|S|} E(S_v)$$

Algoritmos

Árboles de Decisión

Entropía del dataset:

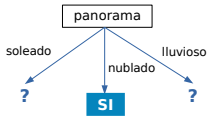
$$E(S) = -\left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right) - \left(\frac{9}{14}\right)\log_2\left(\frac{9}{14}\right) = 0{,}940$$



IG=0,247 **IG=0,174** **IG=0,151** **IG=0,048**

Algoritmos

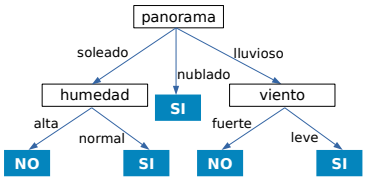
Árboles de Decisión



Algoritmos

Árboles de Decisión

Predicción



Variantes: ID3, C4.5/J48 (variables discretas y continuas), C5.0/See5 y CART (soporta regresión)

Algoritmos

Árboles de Decisión

Pros

- Fáciles de interpretar y útiles para explicación
- Es posible derivar reglas

Contras

- El tiempo de entrenamiento con muchas dimensiones es elevado
- Un ejemplo solo puede descender por una rama del árbol
- No considera interacciones entre los atributos
- Cada decisión se basa en un único atributo, no se recupera de errores en las ramas
- No incrementales
- Puede sufrir de *overfitting*

Algoritmos

Clasificación Bayesiana

Enfoque probabilístico: ve el aprendizaje supervisado desde un punto de vista probabilístico, la clasificación se fundamenta entonces en la teoría de probabilidades (teorema de Bayes).

Sean $X_1,\ldots,X_n$ un conjunto de atributos, la clase C y un ejemplo con valores observados para los atributos $x_1,\ldots,x_n$

La clasificación consiste en estimar la probabilidad **a posteriori**:

$$P(c_j|\mathbf{x})$$

de manera que la predicción es la clase c_j que maximice dicha probabilidad

Algoritmos

Clasificación Bayesiana

Objetivo: clasificar **x** en la clase con la mayor probabilidad a posteriori

$$c^* = \arg\max_{j=1\ldots m} P(c_j|\mathbf{x})$$

Cómo determinar $P(c_j|\mathbf{x})$ para cada clase c_j ?

$$P(c_j|\mathbf{x}) = \frac{P(c_j)P(\mathbf{x}|c_j)}{P(\mathbf{x})}$$

Teorema de Bayes

Igual para todas las clases, no es relevante

Regla de clasificación Bayesiana

$$c^* = \arg\max_{j=1\ldots m} P(c_j)P(\mathbf{x}|c_j)$$

Algoritmos

Clasificación Bayesiana

$$c^* = \arg\max_{j=1\ldots m} P(c_j)P(\mathbf{x}|c_j) \rightarrow P(x_1 \ldots x_n|c_j)$$

Es una probabilidad costosa de calcular para muchas dimensiones

Suposición de Independencia

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

Todos los atributos son condicionalmente independientes dada la clase

$$P(\mathbf{x}|c_j) = \prod_{i=1}^{n} P(X_i = x_i|c_j)$$

Regla de clasificación de Naïve Bayes

$$c^* = \arg\max_{j=1\ldots m} P(c_j) \prod_{i=1}^{n} P(X_i = x_i|c_j)$$

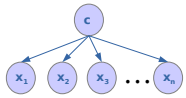
Observados en datos de Entrenamiento

Algoritmos

Naïve Bayes

Entrenamiento

- Estimar las probabilidades P(c_j) y P(X_i=x_i|c_j) los ejemplos de entrenamiento



Predicción

- Utilizar la regla de clasificación de Naïve Bayes
- Predice la probabilidad para cada clase c_j

Algoritmos

Naïve Bayes

Entrenamiento: dado un conjunto de datos con *N* ejemplos etiquetados, se estiman:

- Para cada clase:

$$\hat{P}(c_j) = \frac{N_j}{N}$$

N_j es el número de ejemplos de la clase c_j

- Para cada valor x_k del atributo X_i y para cada clase c_j:

Si X_i es discreto

$$\hat{P}(X_i = x_k|c_j) = \frac{N_{ijk}}{N_j}$$

N_{ijk} es el número de ejemplos de la clase c_j que tienen el valor x_k para el atributo X_i

Si X_i es continuo

$$\hat{P}(X_i = x_k|c_j) = g(x_k; \mu_{ij}, \sigma_{ij}) \qquad g(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

La media número μ_{ij} y la desviación estándar σ_{ij} se estiman de los ejemplos de entrenamiento

Algoritmos

Naïve Bayes

ID	Rembolso	Estado civil	Ingreso imponible	Evade
1	SI	Soltero	125K	NO
2	NO	Casado	100K	NO
3	NO	Soltero	70K	NO
4	SI	Casado	120K	NO
5	NO	Divorciado	95K	SI
6	NO	Casado	60K	NO
7	SI	Divorciado	220K	NO
8	NO	Soltero	85K	SI
9	NO	Casado	75K	NO
10	NO	Soltero	90K	SI

Para las clases:

P(NO)= 7/10
P(SI) = 3/10

$$P(c_j) = \frac{N_j}{N}$$

Entrenamiento

Algoritmos

Naïve Bayes

ID	Rembolso	Estado civil	Ingreso imponible	Evade
1	SI	Soltero	125K	NO
2	NO	Casado	100K	NO
3	NO	Soltero	70K	NO
4	SI	Casado	120K	NO
5	NO	Divorciado	95K	SI
6	NO	Casado	60K	NO
7	SI	Divorciado	220K	NO
8	NO	Soltero	85K	SI
9	NO	Casado	75K	NO
10	NO	Soltero	90K	SI

Para las clases:

$$P(c_j) = \frac{N_j}{N}$$

P(NO)= 7/10
P(SI) = 3/10

Para cada valor de atributo y clase:

P(estado=casado|NO)= 4/7

$$P(X_i = x_k | c_j) = \frac{N_{ijk}}{N_j}$$

Algoritmos

Naïve Bayes

ID	Rembolso	Estado civil	Ingreso imponible	Evade
1	SI	Soltero	125K	NO
2	NO	Casado	100K	NO
3	NO	Soltero	70K	NO
4	SI	Casado	120K	NO
5	NO	Divorciado	95K	SI
6	NO	Casado	60K	NO
7	SI	Divorciado	220K	NO
8	NO	Soltero	85K	SI
9	NO	Casado	75K	NO
10	NO	Soltero	90K	SI

Para las clases:

$$P(c_j) = \frac{N_j}{N}$$

P(NO)= 7/10
P(SI) = 3/10

Para cada valor de atributo y clase:

P(estado=casado|NO)= 4/7
P(rembolso=SI|SI) = 0
...

$$P(X_i = x_k | c_j) = \frac{N_{ijk}}{N_j}$$

Algoritmos

Naïve Bayes

ID	Rembolso	Estado civil	Ingreso imponible	Evade
1	SI	Soltero	125K	NO
2	NO	Casado	100K	NO
3	NO	Soltero	70K	NO
4	SI	Casado	120K	NO
5	NO	Divorciado	95K	SI
6	NO	Casado	60K	NO
7	SI	Divorciado	220K	NO
8	NO	Soltero	85K	SI
9	NO	Casado	75K	NO
10	NO	Soltero	90K	SI

Para cada par atributo y clase:

Ingreso, Evade=NO:

μ_{ij} = 110
σ_{ij} = 54.54

Definición

Naïve Bayes

ID	Rembolso	Estado civil	Ingreso imponible	Evade
1	SI	Soltero	125K	NO
2	NO	Casado	100K	NO
3	NO	Soltero	70K	NO
4	SI	Casado	120K	NO
5	NO	Divorciado	95K	SI
6	NO	Casado	60K	NO
7	SI	Divorciado	220K	NO
8	NO	Soltero	85K	SI
9	NO	Casado	75K	NO
10	NO	Soltero	90K	SI

	Rembolso		Estado civil			Ingreso
	SI	NO	soltero	casado	divorciado	
SI	0	1	2/3	0	1/3	μ_{ij} = 90 σ_{ij} = 5
NO	3/7	4/7	2/7	4/7	1/7	μ_{ij} = 110 σ_{ij} = 54.54

Entrenamiento

Algoritmos

Naïve Bayes

Predicción

P(Ingreso=120|evade=NO)=

$$\frac{1}{\sqrt{2\pi(54.54)}}\exp\left(-\frac{(120-110)^2}{2(54.54)^2}\right)=0.0072$$

Nuevo ejemplo:

$$X=(rembolso=NO, Divorciado, Ingreso=120\,K)$$

X|

Cuál es mayor P(evade=SI |X) o P(evade=NO |X)?

P(X|NO)= P(rembolso=NO|NO)*P(divorciado|NO)*P(120K|NO)
= 4/7 * 1/7 * 0.0072 = **0.00058**

P(X|SI)= P(rembolso=NO|SI)*P(divorciado|SI)*P(120K|SI)
= 1 * 1/3 * (1.2*10^{-9}) = **4*10^{-10}**

P(X|NO) > P(X|SI)

P(NO)= 0.00058 * 7 / 10 = **0.000406**
P(SI)= 4*10^{-10} * 3 / 7 **= 1.7 * 10^{-10}**

Algoritmos

Naïve Bayes

Pros

- Eficiente en tiempos de entrenamiento y clasificación
- Es robusto a atributos irrelevantes y *outliers*
- Trabaja con valores faltantes, ignorando la instancia en el cálculo de probabilidades
- Maneja bien combinación de atributos discretos y continuos
- Incremental

Contras

- Hace una suposición fuerte de independencia entre los atributos (poco realista en muchos casos, nunca es cierta en textos)

Algoritmos

SVM

Support Vector Machines (SVMs): las máquinas de vectores de soporte fueron propuestas por (Vapnik et al., 1992)

SVMs es un clasificador **binario** que encuentra un hiperplano para separar dos clases de datos (positivos y negativos)

Espacio de características inducido por un *kernel* para datos no linealmente separables

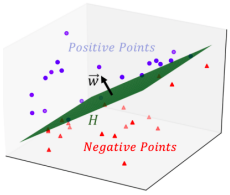
Algoritmos

SVM

Cada observación consiste en:

Atributos $x_i \in \mathbb{R}^n, i=1,\ldots,l$

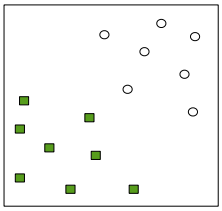
Clase $y_i \in \{+1,-1\}$



Existe un hiperplano *H* que separa los ejemplos positivos (+1) de los negativos (-1). Los puntos que están en el hiperplano satisfacen **w**·**x**+b=0, donde **w** es un vector ortogonal que define la orientación del hiperplano y b representa el desplazamiento desde el origen

Algoritmos

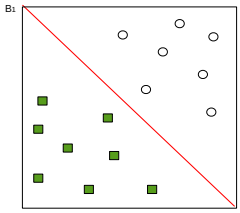
SVM



Objetivo: encontrar un hiperplano que separe los datos

Algoritmos

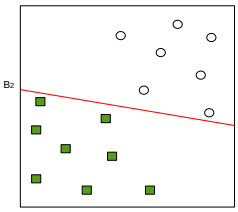
SVM



Una solución posible

Algoritmos

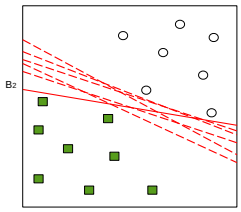
SVM



Otra solución posible

Algoritmos

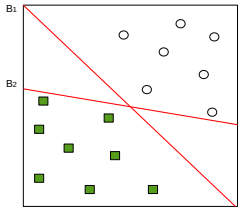
SVM



Otras soluciones posibles

Algoritmos

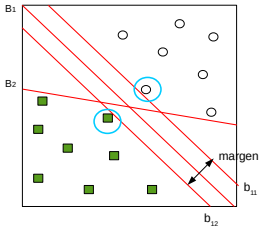
SVM



Cuál es mejor B1 o B2?
Cómo se define mejor?

Algoritmos

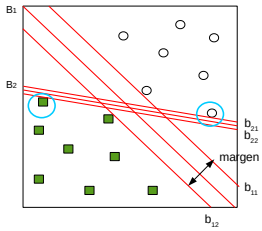
SVM



Encontrar el hiperplano que maximize el margen

Algoritmos

SVM

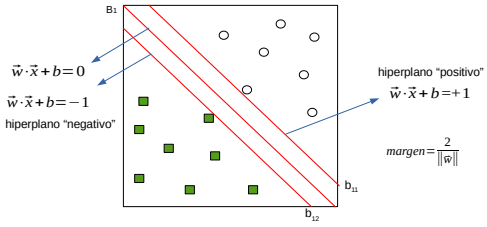


El **margen** se define como el ancho en que pueden desplazarse los límites antes de encontrar un punto

Encontrar el hiperplano que maximize el margen
→ B1 es mejor que B2

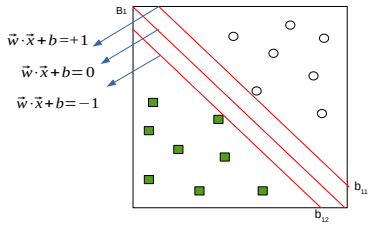
Algoritmos

SVM



Algoritmos

SVM



Objetivo: separar los ejemplos de entrenamiento de acuerdo a su etiqueta y dos hiperplanos:

$$f(\vec{x}) = \begin{cases} 1 & if\ \vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b \geq 1 \\ -1 & if\ \vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b \leq -1 \end{cases}$$

Algoritmos

SVM

Entrenamiento

- Encontrar el hiperplano con máximo margen implica resolver el problema de optimización con restricciones:

maximizar: $\frac{2}{\|\vec{\mathbf{w}}\|}$ sujeto a: $y_i = \begin{cases} 1 & if\ \vec{\mathbf{w}}\cdot\vec{x_i}+b \geq 1 \\ -1 & if\ \vec{\mathbf{w}}\cdot\vec{x_i}+b \leq -1 \end{cases}$

o

minimizar: $\frac{\|\vec{\mathbf{w}}\|^2}{2}$ $y_i(\vec{\mathbf{w}}\cdot\vec{x_i}+b) \geq 1$

Predicción

- Una vez encontrado *w* y *b*, para un ejemplo x_i:

$$f(\vec{x_i}) = \begin{cases} 1 & if\ \vec{\mathbf{w}}\cdot\vec{x_i}+b \geq 1 \\ -1 & if\ \vec{\mathbf{w}}\cdot\vec{x_i}+b \leq -1 \end{cases}$$

Algoritmos

SVM

Pros

- Buena precisión, especialmente en datos altamente dimensionales (por ejemplo, textos)
- Eficiente para predicción

Contras

- Clasificación binaria (one-vs-all o one-vs-one)
- Trabaja en el espacio de números reales, se necesita convertir valores discretos a numéricos
- Requiere ajuste de *kernel* y parámetros

Próxima clase

Algoritmos de Aprendizaje Supervisado

- Vecinos más cercanos
- Árboles de decisión
- Clasificación Bayesiana (Naïve Bayes)
- SVMs
- ...más Módulo 2 y 4

Evaluación del Aprendizaje

- Metodologías
- Métricas