Aprendizaje Inductivo

Descubrimiento de conocimiento a Partir de Datos

Dr. Marcelo G. Armentano
ISISTAN, Fac. de Cs. Exactas, UNICEN

Agenda

- Aprendizaje Inductivo
  - Concepto
- Clasificación
  - Árboles de Decisión
  - Clasificador Bayesiano
- Clustering
  - K-means
  - Clustering jerárquico

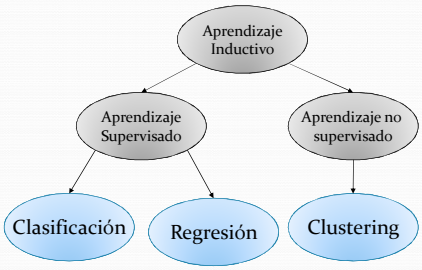
Inducción vs. Deducción

- Argumento deductivo
  - A ninguno de los alumnos le gusta matemáticas. Juan es un alumno → a Juan no le gusta matemáticas
- Argumento inductivo
  - A ninguno de los alumnos que fueron entrevistados les gusta matemáticas→ a ningún alumno le gusta matemáticas

Aprendizaje por inducción

- La descripción de un concepto, o clasificador, se induce a partir de un conjunto de instancias dadas del concepto (ejemplos)
- No puede garantizarse correctitud
- Es importante la interpretación humana

Jerarquía de aprendizaje



Clasificación

Agenda

- Aprendizaje Inductivo
  - Concepto
- Clasificación
  - Árboles de Decisión
  - Clasificador Bayesiano
- Clustering

Clasificación

- El objetivo de la clasificación de datos es organizar y categorizar los datos en clases diferentes
  - Se crea un modelo basándose en la distribución de los datos
  - El modelo es luego usado para clasificar nuevos datos
  - Dado el modelo, se puede predecir la clase de un nuevo dato
- Si se deduce un valor discreto → Clasificación
- Si se deduce un valor continuo → Regresión

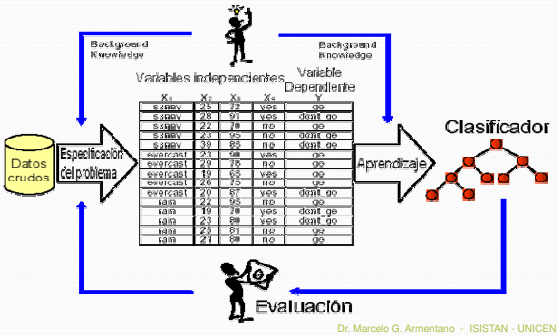
Preparación de los datos

- Transformación de datos
  - Discretización de datos continuos
  - Normalización a [-1..1] o [0..1]
  - Generalización
- Limpieza de datos
  - Suavizado para reducir el ruido y completar valores faltantes
- Análisis de relevancia (Feature Selection)
  - Selección de características para eliminar atributos redundantes e irrelevantes

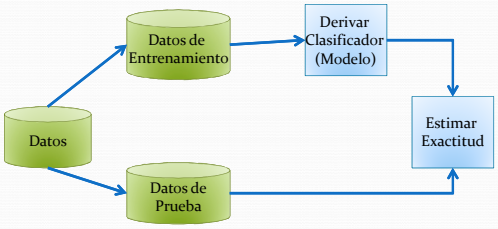
Aplicaciones

- Aprobación de créditos
- Diagnóstico médico
- Identificación de partes defectuosas en manufactura
- Detección de SPAM
- Etiquetado de emails
- Clasificación de documentos
- Clasificación de usuarios
- ...

Proceso de clasificación



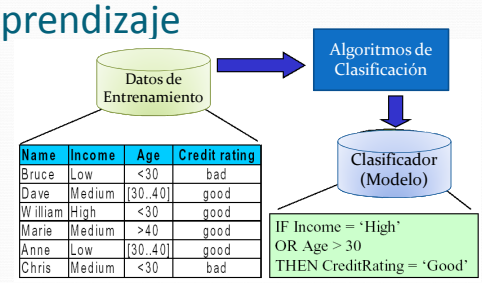
Terminología



Terminología

- Cada tupla se supone que pertenece a una clase predefinida, dada por uno de los atributos, llamada **etiqueta de clase**
- El conjunto de todas las tuplas usadas para la construcción del modelo se llama **conjunto de entrenamiento**
- El modelo se representa mediante alguna técnica. Por ejemplo:
  - Reglas de clasificación (sentencias IF-THEN)
  - Árbol de decisión
  - Fórmulas matemáticas

Aprendizaje

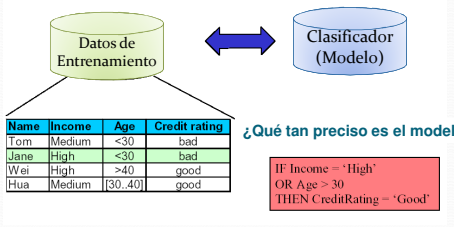


Name	Income	Age	Credit rating
Bruce	Low	<30	bad
Dave	Medium	[30..40]	good
William	High	<30	good
Marie	Medium	>40	good
Anne	Low	[30..40]	good
Chris	Medium	<30	bad

Evaluación del modelo

- Se estima la exactitud del modelo basándose en un conjunto de prueba
  - Se compara la etiqueta conocida de una muestra de prueba con el resultado de aplicar el modelo de clasificación
  - **Accuracy rate (precisión)** es el porcentaje de muestras del conjunto de test que son correctamente clasificadas por el modelo
  - El conjunto de test es independiente del conjunto de entrenamiento (método holdout)

Evaluación de Exactitud



Name	Income	Age	Credit rating
Tom	Medium	<30	bad
Jane	High	<30	bad
Wei	High	>40	good
Hua	Medium	[30..40]	good

Evaluación del modelo

- Holdout
  - Los datos se particionan aleatoriamente en 2 conjuntos independientes: training set (2/3 de los datos) y test set (1/3 de los datos)
- Random subsampling
  - Holdout k veces
- K-fold cross validation
  - Datos iniciales particionados en k subconjuntos mutuamente excluyentes de aproximadamente igual tamaño. Se hace training y testing k veces, se calcula la exactitud promediando los resultados.
- Stratisfied cross-validation
  - Los subconjuntos son armados de tal manera que la distribución de clase de los ejemplos en cada uno es aproximadamente igual a la que tienen los datos iniciales

Evaluación del modelo

- Tasa de Error

$$error(h) = \frac{\sum_{i=1}^{n} \| y_i \neq h(x_i) \|}{n}$$

- Precisión

$$precisión(h) = 1 - error(h)$$

Matriz de Confusión

Etiqueta de clase	Predicciones C1	Predicciones C2	...	Predicciones Ck
Verdaderos C1	**M(C1,C1)**	M(C1,C2)	...	M(C1,Ck)
Verdaderos C2	M(C2,C1)	**M(C2,C2)**	...	M(C2,Ck)
...	...	...	...	...
Verdaderos Ck	M(Ck,C1)	M(Ck,C2)	...	**M(Ck,Ck)**

$$M(C_i, C_j) = \sum_{\{\forall (x,y) \in T : y = C_i\}} \left\| h(x) = C_j \right\|$$

Clasificador ideal

- M(Ci, Ci) Casos correctamente clasificados
- M(Ci, Cj) i≠j Errores de clasificación

	C1	C2	...	Ck
C1	M(C1,C1)	0	...	0
C2	0	M(C2,C2)	...	0
...	...	...	...	0
Ck	0	0	...	M(Ck,Ck)

Evaluación del Modelo (Documentos)

- Precisión
  - De la cantidad de veces que se predijo una clase, cuántas fueron correctas?
- Recall
  - Se encontraron todos los ejemplos que pertenecen a la clase?

Precisión y recall

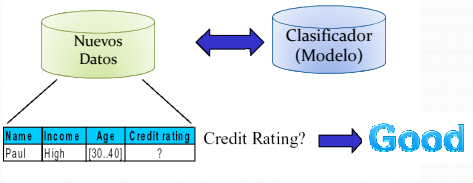
	Clase real	
Predicción	Verdaderos positivos (vp)	Falsos positivos (fp)
	Falsos negativos (fn)	Verdaderos negativos (vn)

$$precisión = \frac{vp}{vp + fp}$$

$$recall = \frac{vp}{vp + fn}$$

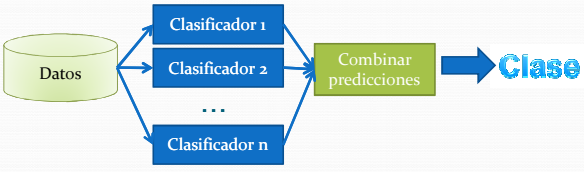
Uso del modelo - Clasificación

- El modelo se utiliza para clasificar nuevos objetos
  - Dar una etiqueta de clase a una nueva tupla
  - Predecir el valor de un atributo



Mejorar la precisión: Clasificadores compuestos

- **Bagging**: ej. consulto varios doctores y me quedo con la opinión mayoritaria (la que tenga más votos)
- **Boosting**: ej. pondero cada diagnóstico según la exactitud del médico (del clasificador)



Métodos de clasificación

- Árboles de decisión
- Redes Neuronales
- Clasificador Bayesiano
- Clasificación basada en asociación
- Vecino más cercano
- Razonamiento Basado en Casos
- Algoritmos Genéticos
- Modelos de Markov
- ...

Evaluación y comparación de métodos de clasificación

- Exactitud de predicción
  - Habilidad del modelo de predecir correctamente la etiqueta de clase de nuevos ejemplos
- Velocidad
  - Tiempo para construir el modelo
  - Tiempo para usar el modelo
- Robustez
  - Manejo de valores faltantes y ruido
- Escalabilidad
  - Eficiencia en grandes bases de datos
- Facilidad de interpretación
  - Nivel de entendimiento provisto por el modelo

Árboles de Decisión

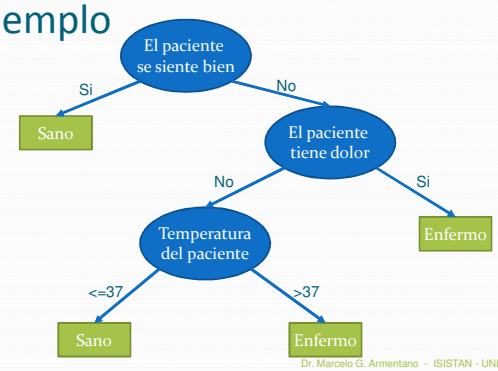
Agenda

- Aprendizaje Inductivo
- Clasificación
  - Árboles de Decisión
  - Clasificador Bayesiano
- Clustering

Árboles de Decisión

- Un árbol de decisión es una estructura de datos definida recursivamente como:
  - Un nodo hoja que contiene una clase
  - Un nodo de decisión que contiene una comprobación sobre algún atributo. Para cada resultado de esa comprobación existe un subárbol hijo, con la misma estructura descripta.

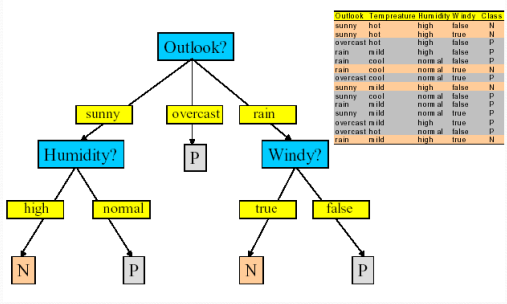
Ejemplo



Ejemplo: Datos de entrenamiento

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Ejemplo árbol de decisión

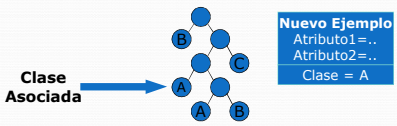


Utilización del árbol

- Directamente
  - Verificar el valor de un atributo de un ejemplo no conocido con el árbol
  - Se sigue el camino desde la raíz a la hoja que posea la etiqueta
- Indirectamente
  - El árbol de decisión se convierte en reglas de clasificación
  - Se crea una regla por cada camino de la raíz a las hojas
  - Las reglas IF-THEN son más fáciles de entender

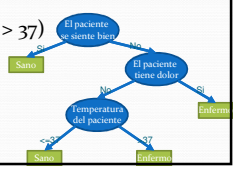
Clasificación de nuevos ejemplos

- Partir desde la raíz
- Avanzar por los nodos de decisión hasta alcanzar una hoja
- La clase del nuevo ejemplo es la clase que representa la hoja.



Equivalente en reglas

- **Si** El paciente se siente bien = Si **entonces**
  - Clase = Sano
- **Sino**
  - **Si** El paciente tiene dolor = No **entonces**
    - **Si** Temperatura del paciente <= 37 **entonces**
      - Clase = Sano
    - **Sino** (Temperatura del paciente > 37)
      - Clase = Enfermo
  - **Sino** (El paciente tiene dolor = Si)
    - **Clase = Enfermo**



Equivalente en reglas

- **Si** El paciente se siente bien = Si **entonces**
  - Clase = Sano
- **Si** El paciente se siente bien = No **and** El paciente tiene dolor = No **and** Temperatura del paciente <=37 **entonces**
  - **Clase = Sano**
- **Si** El paciente se siente bien = No **and** El paciente tiene dolor = No **and** Temperatura del paciente >37 **entonces**
  - **Clase = Enfermo**
- **Si** El paciente se siente bien = No **and** El paciente tiene dolor = Si **entonces**
  - **Clase = Enfermo**

Construcción del árbol de decisión

- La generación de árbol básica de arriba hacia abajo consiste de dos fases:
  - Construcción del árbol
    - Al inicio todos los ejemplos de entrenamiento están en la raíz
    - La partición de los ejemplos se realiza recursivamente basándose en la selección de atributos
  - Podado del árbol
    - Tiene por objetivo eliminar ramas del árbol que reflejen ruido en los datos de entrenamiento y lleven a errores cuando se clasifiquen los datos de test → mejorar la exactitud de clasificación

Construcción del árbol de decisión

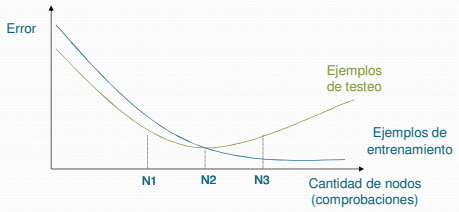
- T contiene uno o más ejemplos, todos pertenecientes a la misma clase Cj → Hoja (clase Cj)
- T no tiene ejemplos → Hoja (la clase debe determinarse a partir de la información anterior a T)
- T contiene ejemplos pertenecientes a varias clases → refinar T en subconjuntos de ejemplos que aparentan pertenecer a la misma clase
- Aplicar los pasos 1, 2 y 3 recursivamente para cada subconjunto de ejemplos de entrenamiento
- Podar el árbol

Construcción del árbol de decisión

- La recursión termina cuando:
  - Los ejemplos en el nodo corresponden a la misma clase
  - No quedan más atributos sobre los cuales separar (hoja con clase mayoritaria)
  - No hay ejemplos con el valor del atributo (para la rama)

Poda

- Es posible que el clasificador obtenido se ajuste demasiado a los ejemplos de entrenamiento
- *overfitting*



Overfitting

- Un árbol generado puede estar sobre-entrenado para los ejemplos de entrenamiento debido a ruido o tamaño pequeño del conjunto de entrenamiento. Resulta en baja exactitud de clasificación de nuevos ejemplos.
- Existen 2 enfoques para evitar overfitting
  - Parar antes (Pre-poda): detener el crecimiento del árbol antes que se produzca, decidiendo no particionar uno o más nodos
  - Podado: se construye todo el árbol, se permite overfitting y luego se poda el árbol. Se elige el árbol con menor tasa de error.

Podado del árbol

- Podar el árbol: transformar un subárbol en una hoja
  - Usar un conjunto de datos diferentes del conjunto de entrenamiento (conjunto de poda)
  - En un nodo del árbol, si la precisión sin particionar el nodo es más alta que la precisión particionando el nodo, reemplazar el subárbol por una hoja, etiquetándolo con la clase mayoritaria

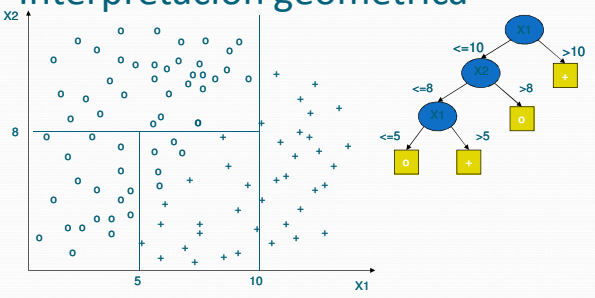
Interpretación geométrica

- Considerar cada ejemplo como un vector de m atributos
- Un AD se corresponderá con una división (por cada comprobación) de este espacio en regiones
- Cada región se corresponde a una clase

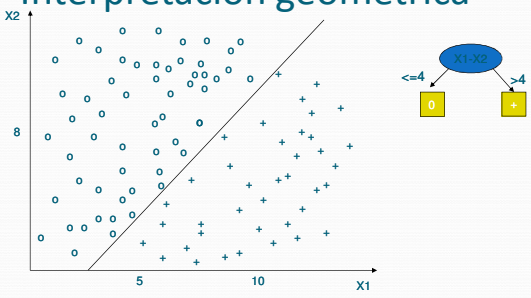
Interpretación geométrica

- Ejemplo
  - 2 atributos reales
  - 2 clases (+ y o)
  - Comprobaciones del tipo *Xi op Valor*
    - Xi ∈ {X_1, X_2}
    - Op es un operador en el conjunto {<=, >}
    - Valor es un número real

Interpretación geométrica



Interpretación geométrica



Interpretación geométrica

- Combinación lineal de atributos
  - c1 + c2*X1 + … + cm*Xm Op Valor
    - Ci: constantes
    - Xi: atributos continuos (enteros o reales)
    - Op: un operador del conjunto {<,<=,>,>=}
    - Valor: una constante

Ejemplo

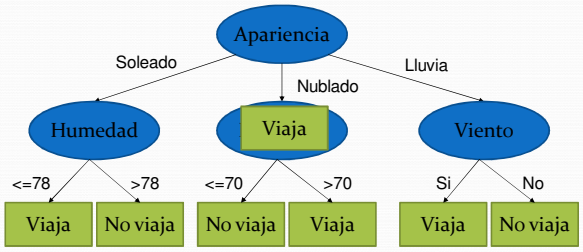
Ejemplo	Apariencia	Temperatura	Humedad	Viento	¿Viajar?
T1	Soleado	25	72	Si	Viajar
T2	Soleado	28	91	Si	No viajar
T3	Soleado	22	70	No	Viajar
T4	Soleado	23	95	No	No viajar
T5	Soleado	30	85	No	No viajar
T6	Nublado	23	90	Si	Viajar
T7	Nublado	29	78	No	Viajar
T8	Nublado	19	65	Si	No viajar
T9	Nublado	26	75	No	Viajar
T10	Nublado	20	87	Si	Viajar
T11	Lluvia	22	95	No	Viajar
T12	Lluvia	19	70	Si	No viajar
T13	Lluvia	23	80	Si	No viajar
T14	Lluvia	25	81	No	Viajar
T15	Lluvia	21	80	No	Viajar

Ejemplo

Comprobación	Ejemplo	Apariencia	Temperatura	Humedad	Viento	¿Viajar?
Si apariencia = Soleado	T1	Soleado	25	72	Si	Viajar
	T2	Soleado	28	91	Si	No viajar
	T3	Soleado	22	70	No	Viajar
	T4	Soleado	23	95	No	No viajar
	T5	Soleado	30	85	No	No viajar
Si apariencia = Nublado	T6	Nublado	23	90	Si	Viajar
	T7	Nublado	29	78	No	Viajar
	T8	Nublado	19	65	Si	No viajar
	T9	Nublado	26	75	No	Viajar
	T10	Nublado	20	87	Si	Viajar
Si apariencia = Lluvia	T11	Lluvia	22	95	No	Viajar
	T12	Lluvia	19	70	Si	No viajar
	T13	Lluvia	23	80	Si	No viajar
	T14	Lluvia	25	81	No	Viajar
	T15	Lluvia	21	80	No	Viajar

Comprobación	Ejemplo	Apariencia	Temp.	Humedad	Viento	¿Viajar?
Si apariencia = Soleado **y** humedad <=78	T1	Soleado	25	72	Si	Viajar
	T3	Soleado	22	70	No	Viajar
Si apariencia = Soleado **y** humedad >78	T2	Soleado	28	91	Si	No viajar
	T4	Soleado	23	95	No	No viajar
	T5	Soleado	30	85	No	No viajar
Si apariencia = Nublado **y** humedad > 70	T6	Nublado	23	90	Si	Viajar
	T7	Nublado	29	78	No	Viajar
	T10	Nublado	20	87	Si	Viajar
	T9	Nublado	26	75	No	Viajar
Si apariencia = Nublado **y** humedad <= 70	T8	Nublado	19	65	Si	No viajar
Si apariencia = Lluvia **y** Viento = No	T11	Lluvia	22	95	No	Viajar
	T14	Lluvia	25	81	No	Viajar
	T15	Lluvia	21	80	No	Viajar
Si apariencia = Lluvia **y** viento = Si	T12	Lluvia	19	70	Si	No viajar
	T13	Lluvia	23	80	Si	No viajar

Ejemplo



Comprobación	Ejemplo	Apariencia	Temp.	Humedad	Viento	¿Viajar?
Si apariencia = Soleado **y** humedad <=78	T1	Soleado	25	72	Si	Viajar
	T3	Soleado	22	70	No	Viajar
Si apariencia = Soleado **y** humedad >78	T2	Soleado	28	91	Si	No viajar
	T4	Soleado	23	95	No	No viajar
	T5	Soleado	30	85	No	No viajar
Si apariencia = Nublado	T6	Nublado	23	90	Si	Viajar
	T7	Nublado	29	78	No	Viajar
	T8	Nublado	19	65	Si	No viajar
	T9	Nublado	26	75	No	Viajar
	T10	Nublado	20	87	Si	Viajar
Si apariencia = Lluvia **y** Viento = No	T11	Lluvia	22	95	No	Viajar
	T14	Lluvia	25	81	No	Viajar
	T15	Lluvia	21	80	No	Viajar
Si apariencia = Lluvia **y** viento = Si	T12	Lluvia	19	70	Si	No viajar
	T13	Lluvia	23	80	Si	No viajar

Cuestiones principales en la construcción de árboles

- Criterio de separación
  - Usado para seleccionar el atributo para separar en un nodo del árbol durante la fase de generación del árbol
  - Diferentes algoritmos pueden usar distintas funciones: ganancia de información, gini, etc.
- Esquema de ramificado
  - Determinará la rama a la cual pertenece una muestra
  - Separación binaria (gini) versus separación múltiple (ganancia de información)
- Decisiones de terminación
  - Cuando detenerse y dejar de separar un nodo (medida de impureza)
- Reglas de etiquetado
  - Un nodo es etiquetado como la clase a la cual pertenecen la mayoría de los ejemplos que pertenecen a él

Elección del mejor atributo para clasificar

- Aleatoria
- Menos valores
- Más valores
- Ganancia de información (máxima)
- Índice Gini (Breiman, Friedman, Olshen, & Stone 1984)
- Razón de ganancia (Quinlan 1993)

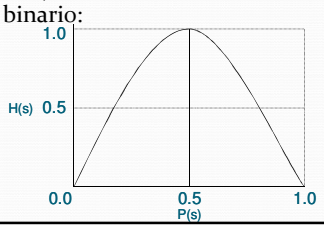
Entropía

- S={s1, s2, ..., sn} alfabeto de una fuente de información de memoria nula
- La cantidad media de información por símbolo del alfabeto S se denomina entropía de S, se representa por H(S)

$$H(S) = \sum_{i=1}^{n} p(s_i) \log_2 (\frac{1}{p(s_i)})$$

Entropía

- $0 \leq H(S) \leq \log_2(|S|)$
- H(S) es máxima cuando los símbolos son equiprobables
- H(S) es 0 cuando $p(s_i)=1$ y $p(s_i)=0$ para todo $j \neq i$
- Para un alfabeto binario:



Algoritmo ID3

- Se calcula la entropía del nodo N

  $H(E_i)=p^+\log_2(1/p^+)+p^-\log_2(1/p^-)$
- Se calcula el valor medio de la entropía del nivel siguiente, generado por el atributo X al que se está aplicando el test

$$H(X, E_i)=p(X=v_1/E_i)H(E_{i1})+...+ p(X=v_n/E_i)H(E_{in})$$

$V_1, v_2, \ldots, v_n$ → valores posibles del atributo X

$E_{ij} \subseteq E_i$ tal que $X=v_j$

Algoritmo ID3

- Se elige el atributo que maximice la ganancia

Ganancia(X)=H(Ei)-H(X,Ei)

Algoritmo ID3

- El atributo que tiene mayor ganancia de información se utiliza para particionar
- Minimizar la información que se necesita para clasificar ejemplos en las particiones resultantes
- Mide qué tan bien separa los ejemplos de entrenamiento de un conjunto dado de acuerdo a su clasificación objetivo (impurezas)
- Minimiza el número esperado de tests que se necesitan para clasificar un objeto y garantiza la construcción de un árbol simple.

ID3 - Ejemplo

Ejemplos	Estatura	Pelo	Ojos	Clase
e_1	Baja	Rubio	Azules	+
e_2	Alta	Rubio	Castaños	-
e_3	Alta	Pelirrojo	Azules	+
e_4	Baja	Moreno	Azules	-
e_5	Alta	Moreno	Azules	-
e_6	Alta	Rubio	Azules	+
e_7	Alta	Moreno	Castaños	-
e_8	Baja	Rubio	Castaños	-

ID3 Ejemplo

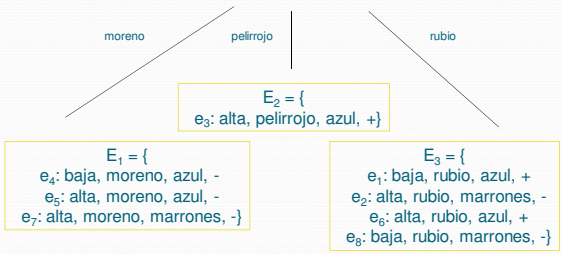
3 ejemplos en clase -; 5 ejemplos en clase +

- $H(E)= 3/8*\log_2(8/3) + 5/8*\log_2(8/5) = 0.954$

- E_1 = {Ejemplos de E que tienen pelo=moreno}
- E_2 = {Ejemplos de E que tienen pelo=pelirrojo}
- E_3 = {Ejemplos de E que tienen pelo=rubio}

ID3 - Ejemplo

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$



ID3 - Ejemplo

- H(pelo, E) = p(pelo=moreno/E)*H(E_1)+
  p(pelo=pelirrojo/E) *H(E_2)+
  p(pelo=rubio/E) *H(E_3)

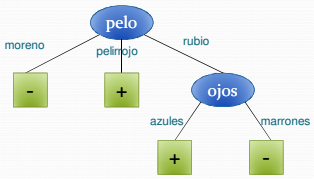
- H(E1)=p+*log2(1/p+)+p-*log2(1/p-) = 0*log2(1/0)+1*log2(1/1) = 0
- H(E2)=p+*log2(1/p+)+p-*log2(1/p-) = 1*log2(1/1)+0*log2(1/0) = 0
- H(E3)=p+*log2(1/p+)+p-*log2(1/p-) = 2/4*log2(4/2)+2/4*log2(4/2)
  = 0.5 + 0.5
  = 1

H(pelo,E) = 3/8 *0 + 1/8 *0 +4/8 *1 = 0.5

Ganancia(pelo) = 0.954 – 0.5 = 0.454

ID3 - Ejemplo

- Ganancia(pelo) = 0.454
- Ganancia(estatura)= 0.003
- Ganancia(ojos) = 0.347



C4.5 - Poda

- Poda:
  - C4.5 efectúa la poda después de haber construido el árbol
  - Estima el error de clasificación sobre ejemplos posteriores
  - Nivel de confianza (por defecto 25%)
  - Se compara el error estimado para un conjunto de hojas, hijas todas de un mismo nodo, y el error estimado para el padre en caso de podar sus hojas.
  - Decidida la poda de las hojas se repite la operación en un nivel superior, hasta que la poda no proceda.

Clasificador Bayesiano

Agenda

- Aprendizaje Inductivo
- Clasificación
  - Árboles de Decisión
  - Clasificador Bayesiano
- Clustering

Clasificador Bayesiano

- Es un clasificador estadístico basado en el **Teorema de Bayes**
- Usa aprendizaje probabilístico para calcular explícitamente las probabilidades de las hipótesis
- Un clasificador bayesiano simple o naive asume **independencia total entre atributos**
- Funciona bien con grandes conjuntos de datos y tiene alta precisión de clasificación
- El modelo es **incremental** es el sentido que cada ejemplo de entrenamiento puede aumentar o disminuir la probabilidad de que una hipótesis sea correcta. Conocimiento previo puede combinarse con datos observados

Clasificador Bayesiano

- El objetivo de un clasificador es identificar a qué clase pertenece un objeto, basándose en ciertos atributos.
- Consideremos un objeto X, cuyos atributos son $(a_1,a_2...a_n)$, y queremos clasificarlo dentro de un conjunto de clases S. Trataremos de encontrar una clase A que maximice $P(A| a_1,a_2...a_n)$.
- Podemos decir entonces:

"Es muy probable que X pertenezca a la clase A porque para un objeto, dada la condición de tener los atributos $(a_1,a_2...a_n)$, la probabilidad de que la clase sea A es máxima."

Ejemplo: Clasificación de correo electrónico

- Atributos: palabras que aparecen en el título del mail
- Clases: S = (spam, no spam)
- Ej: "Oferta increíble compre ya"

- Es spam o no? Lo determina el máximo de:
  - P(S = spam| {oferta, increible, compre, ya})
  - P(S = no spam | {oferta, increible, compre, ya})

Teorema de Bayes

X={redondo, rojo}
H=manzana

- Dado un objeto X con etiqueta de clase desconocida,
- H es la hipótesis de que X pertenece a una clase C

- La probabilidad a posteriori de la hipótesis H, P(H/X), sigue el **teorema de Bayes**

$$P(H/X) = \frac{P(X/H)\,P(H)}{P(X)}$$

  - P(H/X) probabilidad a posteriori
  - P(H), P(X) probabilidad a priori (datos)
  - P(X/H) probabilidad a posteriori (que un objeto sea redondo y rojo dado que es una manzana)

Clasificador Naive Bayes

- Supongamos que tenemos n clases $C_1, C_2,..., C_n$. Dado un ejemplo desconocido A, el clasificador predecirá que $A=(a_1,...,a_m)$ pertenece a la clase con la mayor probabilidad a posteriori:

$$A \in C_i \text{ si } P(C_i/A) > P(C_j/A) \text{ para } 1 \le j \le n,\ j \ne i$$

- Maximizar $P(A/C_i)\,P(C_i) / P(A)$ → Maximizar $P(A/C_i)\,P(C_i)$
- $P(C_i) = s_i/s$
- $P(A/C_i) = P(a_1,...,a_m /C_i)$

Suposición de Naive Bayes

- Siempre que $(a_1,a_2...a_n)$ sean condicionalmente independientes unas de otras.

$$P(a_1, a_2,..., a_n \mid C) = \prod_{i=1}^{n} P(a_i \mid C)$$

- Dado que $P(a_i|C)$ se obtiene fácilmente de los datos de entrenamiento, el clasificador tiene que encontrar una clase c_j que maximice la expresión:

$$\arg\max_{c_j \in C} \prod_{i=1}^{n} P(a_i \mid c_j)$$

Ejemplo

Tamaño	Precio	Peso	Color	compro?
Grande	Barato	Pesado	azul	Si
Grande	Caro	Medio	Azul	No
Chico	Caro	Medio	Verde	Si
Chico	Barato	Liviano	Azul	No
Chico	Caro	pesado	Verde	No

(chico, barato, pesado, azul), compro?

Ejemplo

P(L = Si | T = Chico, Pr = Barato, P = Pesado, C = Azul) y
P(L = No |T = Chico, Pr = Barato, P = Pesado, C = Azul)

Equivalente a:
(1): P(L = S) P(T = Chico | L = S) P(Pr = Barato|L = S) P(P = Pesado | L = S) P(C = Azul|L = S)
(2): P(L = N) P(T = Chico | L = N) P(Pr = Barato|L = N) P(P = Pesado | L = N) P(C = Azul|L = N)

Resultados:
(1): (2/5)(1/2)(1/2)(1/2)(1/2) = 0.025
(2): (3/5)(2/3)(1/2)(1/3)(2/3) = 0.044

Clustering

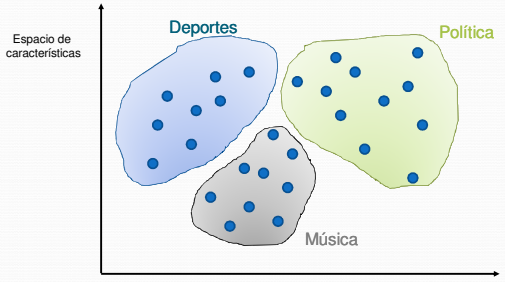
Agenda

- Aprendizaje Inductivo
- Clasificación
  - Árboles de Decisión
  - Clasificador Bayesiano
- Clustering

Clustering: Concepto

- Clustering es un proceso de aprendizaje no supervisado
  - Las **clases no están predefinidas** sino que deben ser descubiertas dentro de los ejemplos
- Primariamente es un **método descriptivo para interpretar un conjunto de datos**
- Particionar ejemplos de clases desconocidas en subconjuntos disjuntos de clusters tal que:
  - Ejemplos en un mismo cluster sean altamente similares entre sí
  - Ejemplos en diferentes clusters sean altamente disimiles entre sí

Clustering: Concepto



Supervisado vs. No supervisado

- **Clasificación supervisada** = clasificación
  - Conocemos las etiquetas de clase de las instancias y el número de clases
- **Clasificación No supervisada** = clustering
  - No conocemos las clases de los ejemplos y posiblemente tampoco en número de clases/clusters

Aplicaciones

- Reconocimiento de patrones
- Procesamiento de imágenes
- Investigación de mercado
- Web mining
  - Categorización de documentos
  - Clustering en Web logs
- Como preprocesamiento para otras técnicas de data mining

Características

- Un buen método de clustering debería identificar clusters que sean tanto compactos como separados entre sí. Es decir, que tengan:
  - Alta similaridad intra-cluster
  - Baja similaridad inter-cluster
- Un buen método debiera descubrir algunos o todos los patrones ocultos en los datos
- La calidad del método de clustering depende de la medida de similitud

Hard clustering vs. Soft clustering

- **Hard clustering**: Cada instancia pertenece a un único cluster
- **Soft clustering:** asigna probabilidades de pertenencia de una instancia a más de un cluster

Tipos de clustering

- Algoritmos basados en **particionamiento**: construyen varias particiones y las evalúan siguiendo algún criterio
- Algoritmos **jerárquicos**: crean una jerarquía que descompone el conjunto de datos usando algún criterio.
- Basados en **modelos**: se supone (hipótesis) un modelo para cada cluster y se trata de encontrar el modelo que mejor se adapte al cluster.

Alg. Basados en Particionamiento

- Construyen una partición del conjunto de datos *D* de *n* objetos en un conjunto de *k* clusters
- Dado un k, intentan encontrar una partición de k clusters que optimiza el criterio de particionamiento

K-Means

- Asume que las instancias son vectores de valores reales
- Los clusters se basan en centroides/medias:

$$\mu(c) = \frac{1}{|c|}\sum_{x \in c} x$$

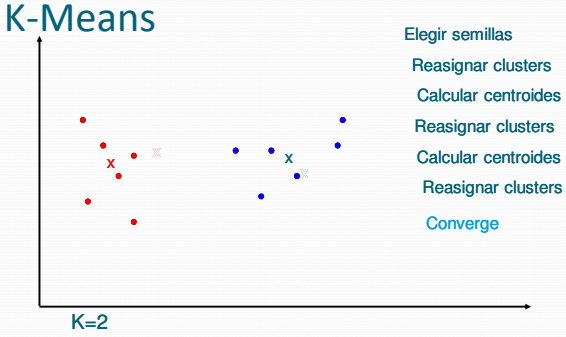
- Las instancias se reasignan a clusters en base a su distancia a los centroides

K-Means

Dado k (número de clusters) y el conjunto de datos n:

1. Arbitrariamente elegir k objetos como centros iniciales de cluster (semillas)
2. Repetir hasta que el algoritmo converja
   - (re)asignar cada objeto al cluster con el cual el objeto sea más similar
   - Actualizar los centroides

K-Means

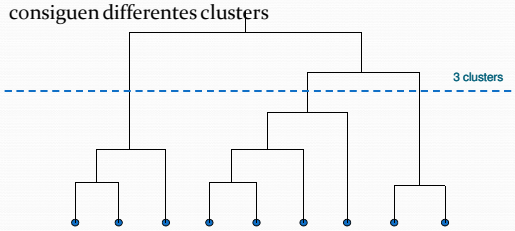


K-Means

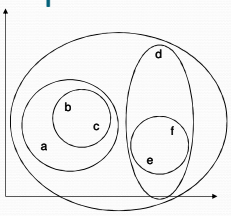
- Ventajas:
  - Entre los algoritmos de particionamiento es eficiente
  - Implementación sencilla
- Desventajas:
  - Necesito conocer *k* de antemano
  - Sensible a ruido
  - El resultado puede variar en base a las semillas elegidas al inicio
  - Algunas semillas pueden resultar en una tasa de convergencia menor
  - La selección de semillas se puede basar en heurísticas o resultados obtenidos por otros métodos
  - Puede caer en mínimos locales
  - No trata datos nominales (K-Modes)

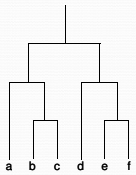
Clustering Jerárquico

- Un **dendograma** muestra como se mezclan los clusters de manera que cortando el dendograma en diferentes nivel se consiguen differentes clusters



Representaciones



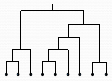


Clustering Jerárquico

- Construye un árbol binario o dendograma a partir de un conjunto de ejemplos:
  - ***Aglomerativo*** (*bottom-up*) métodos que comienzan con cada ejemplo en un cluster diferente y combinan iterativamente los clusters para formar clusters mayores (Ej. AGNES, Agglomerative Nesting [Kaufman90])
  - ***Divisivo*** (*top-down*) métodos que comienzan con todos los ejemplos en un mismo cluster y los separan en clusters más chicos (Ej. DIANA, Divisive Analysis [Kaufman90])

Clustering Jerárquico

- Clustering Aglomerativo:
  1. Comienza con todas las instancias en un cluster separado (cada instancia es un cluster)
  2. Hasta que quede un único clúster
     1. Entre todos los cluster existentes determinar los dos clusters c_i y c_j que son más similares
     2. Reemplazar c_i y c_j por un único cluster $c_i \cup c_j$



Clustering Jerárquico

- Asume una función de similitud que determina la **similitud de dos instancias**: *sim*(*x*,*y*)
  - Por ejemplo la similitud del coseno o coeficiente de correlación de Pearson
- Asume una función de similitud que determina la **similitud de dos clusters** conteniendo multiples instancias:
  - **Single Link**: similitud de los dos elementos del cluster más similares
  - **Complete Link**: similitud de los dos elementos del cluster menos similares
  - **Group Average**: promedio de similitudes entre los elementos del cluster

Similitud entre elementos

- Distancia Euclideana

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2}$$

Similitud entre elementos

- Similitud del coseno

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$C_{\text{cosine}}(\vec{x}, \vec{y}) = \frac{\frac{1}{N}\sum_{i=1}^{N} x_i \cdot y_i}{\|\vec{x}\| \cdot \|\vec{y}\|}$$

$$\vec{x} = -\vec{y} \qquad -1 \le C_{\text{cosine}}(\vec{x}, \vec{y}) \le +1 \qquad \vec{x} = \vec{y}$$

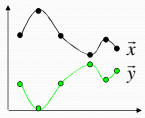
Similitud entre elementos

- Coeficiente de correlación de Pearson

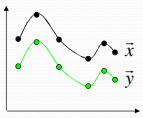
$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$C_{pearson}(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{N}(x_i - m_x)(y_i - m_y)}{\sqrt{[\sum_{i=1}^{N}(x_i - m_x)^2][\sum_{i=1}^{N}(y_i - m_y)^2]}}$$

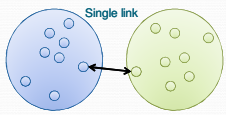
$$m_x = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad m_y = \frac{1}{N}\sum_{n=1}^{N} y_n$$



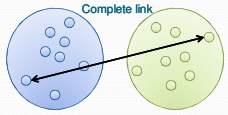
$$-1 \le C_{pearson}(\vec{x},\vec{y}) \le +1$$



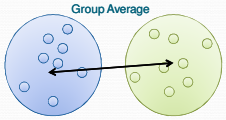
Similitud entre clusters



$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$



$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$



Clustering de documentos

Documentos iniciales

Matriz de Distancias

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

Clustering de documentos

Documentos iniciales

Matriz de Distancias

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

Clustering de documentos

Documentos iniciales

Matriz de Distancias

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

Clustering de documentos

Clusteres actuales

Matriz de Distancias

Dist	A	B	C	
AD		20	7	
B			10	25
C				3
D				

Clustering de documentos

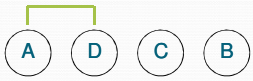
Clusteres actuales

Matriz de Distancias

Dist	AD	B	C	
AD		20	3	
B			10	
C				

Clustering de documentos

Clusteres actuales

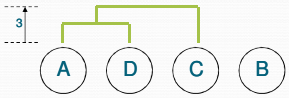


Matriz de Distancias

Dist	AD	B	C	
AD		20	3	
B			10	
C				

Clustering de documentos

Clusteres actuales

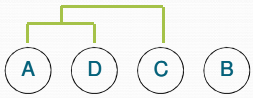


Matriz de Distancias

Dist	AD	B	C	
AD		20	3	
B			10	
C				

Clustering de documentos

Clusteres actuales

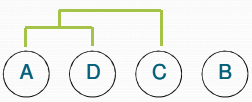


Matriz de Distancias

Dist	ADC	B		
ADC		10		
B				

Clustering de documentos

Clusteres actuales

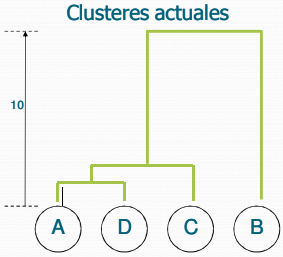


Matriz de Distancias

Dist	ADC	B		
ADC		10		
B				

Clustering de documentos

Clusteres actuales

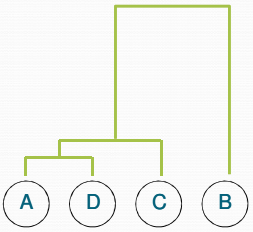


Matriz de Distancias

Dist	ADC	B		
ADC		10		
B				

Clustering de documentos

Clusteres actuales

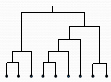


Matriz de Distancias

Dist	AD CB			
AD CB				

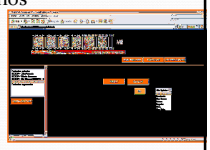
Clustering Jerárquico

- Ventajas:
  - No se necesita conocer el número de clusters k
  - Se puede explorar el dendograma en diferentes niveles, más rico para el análisis de los datos que el particionamiento
- Desventajas:
  - No puede recuperarse de decisiones incorrectas
  - Computacionalmente costoso

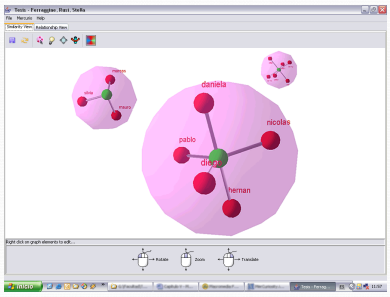


Ejemplo: Comunidades de usuarios en sistemas de recomendación

- **MovieRecommender**: sistema de recomendación de películas que utiliza filtrado colaborativo
- Sugiere películas basándose en la similitud de intereses del usuario actual con otros usuarios "cercanos"
- Análisis de diferentes algoritmos de clustering para armar comunidades de usuarios similares



Ejemplo: Comunidades de usuarios en sistemas de recomendación



Otros algoritmos de clustering

- Basados en modelo
  - Redes Neuronales
    - SOM: Self organizing maps [Kohonen 81]
  - Machine Learning
    - COBWEB (clustering conceptual) [Fisher 87]
  - Estadísticos
    - Gaussian Mixture Model [Raftery 93]
    - Autoclass (Bayesiano) [Cheeseman 96]

Evaluación

- Medidas de calidad internas: miden las propiedades internas de los clusters

$$\frac{1}{|S|^2}\sum sim(d',d)$$

Evaluación

- Medidas de calidad externa: qué tanto se asemejan los resultados de clustering con conocimiento previo del dominio
  - Entropía
  - F-Measure

Evaluación

- F-Measure: mide la efectividad del clustering jerárquico, dado un cluster j y una clase i

$$recall(i, j) = \frac{n_{ij}}{n_i} \qquad precision(i, j) = \frac{n_{ij}}{n_j}$$

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

$$F_\beta = \frac{(1+\beta^2) \cdot precision \cdot recall}{(\beta^2 \cdot precision + recall)}$$

¿Preguntas?