

# Aprendizaje Supervisado Simbólico (inducir un modelo a partir de un conjunto de ejemplos de entrenamiento etiquetados) Clasificadores binarios, atributos nominales

	Aprendizaje de Conceptos	Aprendizaje de Reglas	
	Lógica Proposicional(reglas y ejemplos descritos en términos de atributo-valor)		Lógica Predicados (reglas y ejemplos descritos en términos de predicados y relaciones)
	Inductivos		Mixto (inducción ejemplos, deducción dominio)
	<b>EV</b> (Búsqueda bidireccional)	<b>A<sup>Q</sup></b> (dos búsquedas anidadas)	<b>FOIL</b> (dos búsquedas anidadas)
Entradas y salidas	1) los ej., representados simbólicamente, se caracterizan por El valor de sus atributos y su clase 2) incremental 3) no. 4) externa 5) Es un conjunto de hipótesis consistentes con todos los ejemplos presentados, máximamente generales y máximamente específicos.	1) Ej. vienen en función de los valores de los atributos, y su clase Función LEF, lista de criterios de preferencia de reglas 2) de un solo paso, pese a q e+ se tratan incrementalmente necesitamos todos los e- 3) No 4) Externa 5) Es un recubrimiento, o conjunto de reglas que clasifican a todas las instancias positivas y a ninguna negativa	1. Ejemplos etiquetados y teoría del dominio De forma simbólica en lógica de predicados. Hechos (definición explícita o extensional) 2. Un solo paso 3. Si, suposición de mundo cerrado.(cuando no dan e-) 4. externa al sistema. 5. regla de Hron en el mismo lenguaje que la teoría del dominio, (definición implícita o intensional)
Bias	1. <b>No</b> sujeto a restricciones o bias de preferencia en espacio de búsqueda 2. Restricción en el proceso de búsqueda por el propio lenguaje de descripción de hipótesis ( <b>LDC</b> )	1- La heurística le viene, por un lado, por la función <b>LEF</b> proporcionada por el usuario 2- Por el <b>lenguaje</b> elegido (ej: no tratar con atrib continuos)	Por ser el espacio de problemas mucho mayor, tiene restricciones más fuertes en las heurísticas de búsqueda y restricciones sobre <b>lenguaje</b> a emplear. Similar a ID3. Supone <b>mundo cerrado</b> para generar ejemplos negativos si no los suministra la entrada <b>Literal de máxima ganancia</b> $G(L) = k \times [I(n^+, n^-) - I(nL^+, nL^-)]$
Ruido	<b>En entradas.</b> No trata con ejemplos con ruido. Se supone que las generalizaciones producidas abarcan todos los e+ y excluyen todos los e-. <b>En estructura.</b> La salida del EV <b>no</b> es tolerante a fallos.	<b>En entradas.</b> No trata con ejemplos con ruido, al tener que conseguir que los complejos de la estrella no describan ningún e-, así como exigir que el recubrimiento describa a todos los e+ <b>En estructura.</b> Tolerante, si se elimina 1 regla aún es posible clasificar con el resto.	<b>En entrada:</b> si puede existir. Condición de parada es que la ganancia que se obtiene esté por debajo de umbral. <b>En estructura:</b> si, Sensible a eliminar elementos aprendidos, pero tolerante a fallos
Control tarea aprendida	<b>CUV:</b> No realiza ningún procedimiento explícito de crítica o evaluación de las hip. aprendidas <b>UAP:</b> El sistema tiene capacidad de distinguir, al converger a una misma hipótesis los conjuntos S y G, cuando el concepto ha sido aprendido. A lo largo del propio proceso de aprendizaje, el sistema también puede utilizar el EV como si el concepto estuviera totalmente aprendido.	<b>CUV:</b> no realiza una crítica de lo aprendido <b>UAP:</b> la forma de utilizar lo aprendido es llamar a un procedimiento clasificar con el ejemplo que se desee clasificar y el recubrimiento (devuelto por el proceso de entrenamiento por la función estrella). La función clasificar, devolverá la clase positivo si se cumplen las condiciones de alguna regla y negativo en caso contrario).	<b>CUV:</b> no realiza crítica de lo aprendido <b>UAP:</b> consiste en clasificar un determinado ejemplo de entrada por medio del programa PROLOG aprendido.
Complejidad	<b>Espacio:</b> O (s+g) <b>Tiempo:</b> O (n) polinomial respecto al ejemplos de entrada g= tamaño max alcanzado por conj G s= tamaño max alcanzado por conj S n=número de ejemplos de entrada	<b>Espacio:</b> peor caso una regla por atributo $ A  \times  p $ A= conj de atrib p= inst + <b>Tiempo:</b> para cada e+ Depende numero atributos y número de e-	<b>Espacio:</b> C (el conjunto de cláusulas aprendidas. Máximo nº de ej + <b>Tiempo:</b> exponencial en el nº de arg de los predicados
fiabilidad	Si ej. no tienen ruido, EV encuentra el conj de hipótesis q. resume el conj total de hip. Consistentes con ej. tratados. Con nº suficiente de ej, el proc encuentra una única hip. o concepto aprendido (caso de que S=G)	Si ej. no tienen ruido, A <sub>q</sub> encuentra el recubrimiento (conjunto de reglas) que describen correctamente a todos los ej.	Descripción correcta del concepto meta en función de los ej. de entrada y de la representación, que en este caso afecta más.
>Dependencia Conocimiento del dominio	Se refleja en la definición de los atributos y sus valores, LDC	Se refleja en la definición de los atributos y sus valores, LDC	Si, dependen de la teoría del dominio suministrada como entrada

1=Representación de las entradas 2= Tratamiento de las entradas 3= Preprocesamiento de las entradas 4= Fuente de las entradas 5= Representación de las salidas

CUV = Crítica/utilidad/valoración

UAP = Utilización de lo aprendido

## Aprendizaje Supervisado inductivo Mixto (inducir un modelo a partir de un conjunto de ejemplos de entrenamiento etiquetados) Clasificadores, atributos simbólicos y numéricos

	<b>Arboles Decisión ID3-IC4-5</b> (clases discretas y predefinidas)	<b>Arboles Regresión M5</b> (predice valor de clase, clases continuas) (tres búsquedas consecutivas)	<b>Técnicas aprendizaje vago</b> (aprendizaje basado en instancias)	<b>Clasificadores bayesianos</b> (clases discretas y predefinidas)
Entradas y salidas	1. Igual que alg inductivos, ej. Vienen en función de los valores de los atributos (en v. simple, <b>no continuos</b> ). Ejemplos >>clases, al ser proceso estadístico 2. de un solo paso. 3. no. 4. externa. 5.un árbol de decisión transformable a conj de reglas	1. Ej. vienen en función de los valores de los atributos y valor de clase, umbrales $\omega$ y $\tau$ 2.en un solo paso. 3.no. 4.externa. 5.un árbol de regresión que tiene en cada nodo intermedio una pregunta sobre el valor de un atributo y en cada nodo hoja un mod. Lineal para calcular la clase numérica.	1. Matriz (filas son ejemplos, columnas atributos, y posiciones matriz representan valor del atrib para ej. ,medida de similitud y K 2.incremental. 3.no. 4.externa. 5.Conj de ej. que pueden estar almacenados en diferentes estructuras de datos, como listas ordenadas, matrices o tablas hash.	1. Atrib simbólicas o numéricos. 2.en un solo paso, aunque es fácil realizar incremental. 3.no 4.externa 5. Matrices numéricas : $MC$ , $MAD$ y $MAC$
Bias	<b>Ganancia de Información (Diferencia de entropía entre el padre y sus sucesores)</b> . Prefiere árboles que tengas atributos con mayor información más cerca de la raíz. Favorece atributos con muchos valores	<b>maximizar la reducción del error (minimizar <math>\sigma</math> hijos)</b> De búsqueda(que atrib es el mejor) y de lenguaje (representa modelos lineales solamente)		Debido a la utilización del teorema de Bayes, que supone que los valores de los atributos son independientes con respecto a la clase. captura de manera probabilística la Información semántica que podrían estar almacenando los atributos de valores simbólicos.
Ruido	<b>En entradas</b> : SI, no necesario q todos ej. Sean de la misma clase para parar <b>En estructura</b> : tolerante a fallos. Si se pierde algún subárbol el resto puede seguir clasificando.	<b>En entrada</b> : se permite, suavizado por análisis de errores <b>En estructura</b> : tolerante a fallos (dispone en cada nodo intermedio de mod. Lineal, puede perderse parte AR)	<b>En entradas</b> : se suaviza mediante función de similitud <b>En estructura</b> : tolerante a fallos, aunque se pierda algún elemento	<b>En entrada</b> : estadística suaviza ruido <b>En estructura</b> : puede afectar a clasificación final, pero al estar multiplicado por otros términos numéricos, resultado posib. similar.
Control tarea aprendida	<b>CUV</b> : no realiza crítica de lo aprendido <b>UAP</b> : consiste en llamar a un proc clasificar con ej. que se desee y el árbol de decisión	<b>CUV</b> : valoración interna a través del cálculo de error, con conjunto de test, 2 y 3 búsquedas. <b>UAP</b> : consiste en llamar a un proc clasificar con ej. que se desee clasificar y árbol de regresión	<b>CUV</b> : no existe <b>UAP</b> : <b>Clasificación</b> de ejemplos posteriores en función de similitud entre éste y los almacenados. Suele ser inverso de función euclídea.	<b>CUV</b> : critica surge cuando aparece un nuevo ej., dado que hay que recalcular las matrices de probabilidad. <b>UAP</b> : para clasificar se utiliza teorema de Bayes
Complejidad	<b>Espacio</b> : el ocupado por el árbol de decisión <b>Tiempo</b> : linealmente con $n^o$ de ej. de entrenamiento y exp. con $n^o$ atrib	<b>NP(exponencial), si no fuera por las bias</b> <b>Espacio</b> : el ocupado por el árbol de regresión <b>Tiempo</b> : depende del $n^o$ de atrib y del $n^o$ de ej. Lineal $n^o$ de nodos, exp. con $n^o$ atrib	<b>Espacio</b> : número de ej. de entrada <b>Tiempo</b> : depende de estructura, pero normalmente menor que complejidad polinómica	<b>Espacio</b> : tres matrices <b>Tiempo</b> : proporcional a Matrices, pero cálculo de media y varianza proporcional a ejemplos
fiabilidad	Sin ruido, encuentra árbol de decisión que describe ej. con ruido, depende de lo significativos que sean los ej.	Fiable si ej. de entrenamiento representativos y valores convenientes de v y w.	Depende de la medida de similitud y de ejemplos suministrados en la entrada	Fiable igual que métodos inductivos, siempre que ej. de entrenamiento sean representativos de los futuros
>Dependencia Conocimiento del dominio	Se refleja en la definición de los atributos y sus valores	Se refleja en la definición de los atributos y sus valores	Representado en la medida de la similitud.	Se refleja en la definición de los atributos y sus valores

1=Representación de las entradas 2= Tratamiento de las entradas 3= Preprocesamiento de las entradas 4= Fuente de las entradas 5= Representación de las salidas

CUV = Crítica/utilidad/valoración

UAP = Utilización de lo aprendido

## Aprendizaje Supervisado NO Simbólico (inducir un modelo a partir de un conjunto de ejemplos de entrenamiento etiquetados) Clasificadores, atributos numéricos

	<b>Redes de Neuronas Artificiales (RNA)</b> <b>perceptrón multicapa</b> aprendizaje por retropropagación del error (excelente aproximador de funciones no lineales, no son un algoritmo, sino un paradigma)	<b>Máquinas de Vectores Soporte</b> permiten resolver tanto tareas de clasificación como de regresión
Entradas y salidas	<ol style="list-style-type: none"> <li>1. Pares de vectores numéricos, ejemplos etiquetados con atributos numéricos continuos. -Otros parámetros entrada: Regla aprendizaje, Tasa aprendizaje <math>\mu</math>, iteraciones o umbral, función activación, arquitectura, pesos iniciales(aleatorios)</li> <li>2. Incremental iterativo</li> <li>3. Normalización de atributos</li> <li>4. Externa</li> <li>5. Vector numérico, misma dimensión q vector de salida deseada</li> </ol>	<ol style="list-style-type: none"> <li>1. Conjunto de ejemplos etiquetados</li> <li>2. En un solo paso</li> <li>3. No</li> <li>4. Externa</li> <li>5. coeficientes reales <math>w</math> y <math>b</math> que definen un hiperplano de separación de margen máximo(una dimensión menor q la de los ejemplos de entrada), o una función de regresión.</li> </ol>
Bias	Función de error(minimización del error de aprendizaje) , el heurístico de búsqueda(regla de aprendizaje, controlada por tasa aprendizaje variable y momento constante) descenso de gradiente	- minimizar lo que se conoce como riesgo estructural, maximizar MARGEN DEL HIPERPLANO y minimizar variables de holgura(ejemplos no separables) - vectores soporte - transformación problema primal a dual - funciones kernel
Ruido	<b>En entradas: SI</b> , elevada resistencia, solo intenta aproximar las salidas, no valor exacto de ellas. Pueden aprender a reconocer patrones con ruido, distorsionados, o incompletos. (La triple i: información incompleta, inexacta e inconsistente) <b>En estructura:</b> tolerante a fallos, a la disfunción de elementos concretos. Pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red. Codificación de la información distribuida, representación información redundante, funcionamiento correcto ante destrucción parcial.	<b>En entradas: SI</b> , admiten margen de error en clasificación, variable de holgura, SVM margen blando <b>En estructura: NO</b> , si varía alguno de los parámetros de salida el hiperplanos y la función de regresión no clasificarán bien
Control tarea aprendida	<b>CUV:</b> critica mediante evaluación del error, error de aprendizaje y error de validación. <b>UAP:</b> pesos fijos, se le presenta un vector de entrada y responde con un vector de salida	<b>CUV:</b> critica mediante margen del hiperplanos y variables de holgura. <b>UAP:</b> fijados los parámetros del hiperplanos o la función de regresión, se le presenta un vector de entrada y responde con la clase o valor de clase(regresión) predicho
Complejidad	<b>Espacio:</b> constante, depende arquitectura de la red, matriz de pesos <b>Tiempo:</b> depende número de ejemplos y de iteraciones hasta convergencia o condición de parada	<b>Espacio:</b> constante, parámetros del hiperplanos o de la función de regresión. <b>Tiempo:</b> depende número de vectores soporte no de ejemplos, y de la separabilidad o no de los ejemplos
Fiabilidad	Capaz de aproximar con la precisión q se desee si se elige arquitectura correcta y se realizan suficientes iteraciones <b>FIABILIDAD</b> determinada por el error de validación, respuesta de la red tiene un error asociado.	Muy fiables debido a su tolerancia al ruido de entrada
>Dependencia Conocimiento del dominio	NO, Independiente, puede aplicarse para la selección de los ejemplos de entrenamiento	En la representación de los ejemplos y soluciones, así como en la elección del tipo de máquina, kernelizada o no.

1=Representación de las entradas 2= Tratamiento de las entradas 3= Preprocesamiento de las entradas 4= Fuente de las entradas 5= Representación de las salidas

CUV = Crítica/utilidad/valoración

UAP = Utilización de lo aprendido

## Aprendizaje NO Supervisado (conjunto de ejemplos de entrenamiento NO etiquetados)

	<b>Agrupamiento (Clustering)</b> (mixto, atributos numéricos y simbólicos)		(No simbólico, atributos numéricos)se puede usar en clasificación y en agrupamiento
	<b>Estrategia Aglomerativa</b>	<b>k-medias</b>	<b>Redes Neuronales auto-organizadas</b>
Entradas y salidas	Entradas: conjunto ejemplos y medida de similitud 1)Matriz, filas ejemplos, columnas atributos(normalmente numéricos) 2) en un solo paso 3) normalización de atributos 4)externa 5) una jerarquía de clases que incluya, cada una de ellas, a un subconjunto del conjunto de ejemplos, <b>dendograma</b> , cada nodo representa una clase de observaciones.	Entradas: conjunto ejemplos, nº de clases y medida de similitud 1)Matriz, filas ejemplos, columnas atributos(normalmente numéricos) 2) en un solo paso 3) normalización de atributos 4)externa 5)Conjunto de clases, cada una con su lista de ejemplos y su prototipo (centroide)	Entradas: conjunto ejemplos, arquitectura de la red, pesos iniciales, tasa de aprendizaje 1)vectores numéricos de dimensión fija, ejemplos 2) incremental-iterativa 3) no es necesario, función de activación normaliza 4)externa 5)Matriz de valores numéricos, pesos, representan la posición de los prototipos de la regla de clasificación
Bias	Función de similitud	-Nº de clases a generar, limita el espacio de hipótesis - función de similitud	-De lenguaje -Regla de Hebb, reforzar conexiones entre células qu reaccionan de forma similar -Función de similitud -Nº de categorías a generar, limita el espacio de hipótesis
Ruido	<b>En entradas.</b> No se trata, se desconoce la clase de los ejemplos. Le afecta menos q a los supervisados <b>En estructura.</b> Si se elimina o modifica una clase del dendograma puede seguir clasificando en clases más generales o más específicas	<b>En entradas.</b> No lo trata, pero al incluir ej. En la clase más parecida se suaviza <b>En estructura.</b> Si se elimina una clase seguirá clasificando, pero los futuros ejemplos de la clase eliminada serán asignados a la clase mas parecida de las restantes	<b>En entradas</b> Tolerante al ruido tanto en aprendizaje como en clasificación <b>En estructura.</b> Si se elimina una célula de la capa de competición se pierde una clase de salida, clasificara los ejemplos en la clase más parecida. La pérdida de células de la capa de entrada afecta dependiendo de lo critico del atributo. La desaparición de conexiones cierta degradación en la clasificación.
Control tarea aprendida	<b>CUV:</b> No existe, al no estar etiquetados los ejemplos no se puede valorar <b>UAP:</b> dado un nuevo ejemplo se calcula desde la raíz del dendograma su similitud con los centroides de las clases sucesoras, devolviendo subárbol de clases más similares desde raíz a la mas similar	<b>CUV:</b> No existe, al no estar etiquetados los ejemplos no se puede valorar <b>UAP:</b> con un nuevo ejemplo a clasificar se llama a la función <b>clase-mas-parecida</b> con argumentos el ejemplo, lista de clases y su centroide, y la función de similitud	<b>CUV:</b> No existe, al no estar etiquetados los ejemplos no se puede valorar <b>UAP:</b> Se puede utilizar tanto para clasificar como para agrupar
Complejidad	<b>Espacio:</b> el utilizado por el dendograma $O(2 E A)$ número de nodos y cada nodo atributos <b>Tiempo:</b> lineal respecto a nº ejemplos y cuadrático respecto a número de atributos $O( E A^2)$	<b>Espacio:</b> solo guarda los centroides, proporcional a número de clases <b>Tiempo:</b> Depende de la elección de las semillas aleatorias, según lo acertadas q resulte se requerirán mas o menos iteraciones , en cada una cada ejemplo se asigna a una clase, comparando los atributos , $I \times N \times K \times A$	<b>Espacio:</b> Matriz de pesos, células de entrada x células competición <b>Tiempo:</b> Depende del esquema de aprendizaje
Fiabilidad	Fiable si número de ejemplos suficiente y representativo de las clases	Depende de donde se coloquen los centroides iniciales, pero en general se comporta muy bien incluso en presencia de ruido	Depende del conjunto de ejemplos
>Dependencia Conocimiento del dominio	Si se tiene se pueden representar mediante la medida de similitud aproximándola más q la general, así como en la elección de los atributos representativos y la elección de los ejemplos de entrenamiento	Si se tiene se pueden representar mediante la medida de similitud aproximándola más q la general, así como en la elección de los atributos representativos y la elección de los ejemplos de entrenamiento. También se puede aplicar en la elección de las semillas iniciales, de las cuales dependerá el resultado obtenido, así como en la elección del número de clases.	Independiente del conocimiento del dominio

1=Representación de las entradas 2= Tratamiento de las entradas 3= Preprocesamiento de las entradas 4= Fuente de las entradas 5= Representación de las salidas

CUV = Crítica/utilidad/valoración

UAP = Utilización de lo aprendido

## Aprendizaje Por Refuerzo (Valor de clase retardado en el tiempo, recompensa o castigo)

	<b>Algoritmo Q-learning</b> (Objetivo: adquirir conocimiento, <i>política de acción</i> )
Entradas y salidas	<p>Entradas: conjunto estados, conjunto acciones, conjunto posibles refuerzos a recibir del entorno, conjunto de tuplas (instancias de aprendizaje) de la forma <math>(s,a,s',r)</math></p> <ol style="list-style-type: none"> <li>tuplas (instancias de aprendizaje) de la forma <math>(s,a,s',r)</math>, estado actual(discreto), acción aplicada(discreto), nuevo estado(discreto) y refuerzo(continuo)</li> <li>incremental e iterado</li> <li>no</li> <li>externa</li> <li>Matriz numérica, tabla, de los refuerzos esperados en el tiempo para cada estado y cada acción.</li> </ol>
Bias	-De representación, asume estados y acciones finitos para acotar el espacio de hipótesis, si son continuos se discretizan
Ruido	<p><b>En entradas:</b> la formula no determinista lo suaviza,</p> <p><b>En estructura:</b> modificación de alguna posición de la matriz no afecta demasiado al resultado final</p>
Control tarea aprendida	<p><b>CUV:</b> critica surge cuando aparece un nuevo ejemplo, hay q recalcular la matriz de refuerzos.</p> <p><b>UAP:</b> estando en un estado <math>s</math>, consulta matriz de refuerzo para determinar la acción q proporcionara un mayor refuerzo esperado en el futuro.</p>
Complejidad	<p><b>Espacio:</b> el ocupado por la matriz de refuerzos, <math> S  \times  A </math></p> <p><b>Tiempo:</b> proporcional a los ejemplos de entrenamiento y al número de ciclos q se desee ejecutar.</p>
fiabilidad	Converge al optimo si se dan las circunstancias apropiadas.
>Dependencia Conocimiento del dominio	En la representación de los estados y las acciones así como en la definición del refuerzo.

1=Representación de las entradas 2= Tratamiento de las entradas 3= Preprocesamiento de las entradas 4= Fuente de las entradas 5= Representación de las salidas

CUV = Crítica/utilidad/valoración

UAP = Utilización de lo aprendido