

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Aprendizaje Computacional

Eduardo Morales, Hugo Jair Escalante

INAOE

Contenido

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- 1 **Introducción**
- 2 **Aprendizaje**
- 3 **Aprendizaje Inductivo**
- 4 **Espacio de Versiones**

Objetivo General

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- La capacidad de aprender es uno de los atributos distintivos del ser humano
- El aprendizaje es una de las principales áreas de IA
- Generación y almacenamiento de datos + automatización procesos + avances en almacenamiento \Rightarrow muchos datos
- Interés comercial \Rightarrow desarrollo acelerado

Objetivo General

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- El área de aprendizaje en general trata de construir programas que mejoren su desempeño automáticamente con la experiencia.
- Los objetivos del curso son:
 - Dar un panorama general de lo que es aprendizaje computacional y
 - conocer a detalle las técnicas más importantes empleadas.

Temario

- 1 Introducción a aprendizaje computacional
- 2 Técnicas principales:
 - Árboles de decisión y regresión
 - Reglas de clasificación
 - Reglas de asociación
 - *Clustering*
 - Aprendizaje Semi-supervisado
 - Clasificación basada en instancias
 - Métodos Bayesianos
 - Redes neuronales
 - SVM (máquinas de soporte vectorial)
 - Ensamblajes de clasificadores
- 3 Temas relacionados:
 - Evaluación de algoritmos, validación, *overfitting*
 - LDA y PCA
 - Selección de atributos
- 4 Preámbulo de otras técnicas

Temas no vistos

- Aprendizaje evolutivo
- Aprendizaje por refuerzo y por refuerzo profundo
- Programación Lógica Inductiva
- Aprendizaje por transferencia
- Procesos Gaussianos
- Aprendizaje Multi-etiqueta
- AutoML
- Life-long learning
- ...

Evaluación

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

La evaluación del curso se hará con base en:

- Un examen (1/2)
- Proyecto final (1/2)

Referencias

- 1 T. Mitchell (1997) *Machine Learning*, McGraw–Hill.
- 2 I.H. Witten, E. Frank (2005) *Data Mining: practical machine learning tools and techniques* 2nd. Edition. Morgan Kaufmann
- 3 J. Han, M. Kamber (2001) *Data Mining: concepts and techniques*, Morgan Kaufmann.
- 4 E. Alpaydin (2010) *Introduction to Machine Learning*, MIT Press
- 5 C. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer.
- 6 P. Flach (2012). *Machine Learning: The art and science of algorithms that make sense of data*, Cambridge University Press.
- 7 I. Goodfellow, Y. Bengio, A. Courville (2015). *Deep Learning*. MIT Press.

Aprendizaje Computacional

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Posiblemente la característica más distintiva de la inteligencia humana
- Desde el comienzo de las computadoras se cuestionó si serían capaces de aprender
- La capacidad de aprendizaje abre una amplia gama de nuevas aplicaciones
- El entender como aprenden las máquinas nos puede ayudar a entender el aprendizaje humano

Aprendizaje Computacional

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

El aprendizaje humano es diverso e incluye:

- adquisición de conocimiento
- desarrollo de habilidades a través de instrucción y práctica
- organización de conocimiento
- descubrimiento de hechos
- ...

De la misma forma ML estudia y modela computacionalmente los procesos de aprendizaje en sus diversas manifestaciones

Se tiene más teoría y más aplicaciones lo que refleja maduración

Aprendizaje Computacional

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Para resolver problemas creamos programas/modelos que lo resuelvan
- Para algunos problemas es difícil formalizarlos, contar con expertos, es demasiada información, ... \Rightarrow ML
- ML genera automáticamente programas/modelos a partir de datos
- Esto abre una gran cantidad de posibles aplicaciones

Aprendizaje Computacional

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Aprendizaje: *Es el campo de estudio que le da a las computadoras la habilidad de aprender sin ser programadas explícitamente* [Samuel, 59]

Aprendizaje: *Cambios adaptivos en el sistema para hacer la misma tarea(s) de la misma población de una manera más eficiente y efectiva la próxima vez* [Simon, 83].

Aprendizaje: *Un programa de computadora se dice que aprende de experiencia E con respecto a una clase de tareas T y medida de desempeño D , si su desempeño en las tareas en T , medidas con D , mejora con experiencia E* [Mitchell, 97].

Perspectivas

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Los objetivos dependen de la perspectiva:

- Ingenieril (resolver tareas)
- Simulación cognitiva
- Análisis teórico

Programar una máquina lleva mucho tiempo, ML puede *suavizar* ese proceso

Aprendizaje en Sistemas Expertos

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Desde el punto de vista de sistemas basados en conocimiento...

*“...knowledge is currently acquired in a very painstaking way; individual computer scientists work with individual experts to explicate the expert’s heuristics – the problem of knowledge acquisition is the critical bottleneck in artificial intelligence.”
Feigenbaum and McCorduck*

Aprendizaje en Sistemas Expertos

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Desde el punto de vista de sistemas basados en conocimiento...

“Knowledge engineers have generally assumed that the procedural knowledge which drives an expert’s skilled behavior can be elicited by dialogue. The findings of the past decade in brain science, cognitive psychology and commercial software do not support this idea. Evidence is lacking that skills, having once reached the “automization” stage, can be de-automized by dialogue so as to make their inner workings accessible to introspective report.”
Donald Michie

Clasificaciones

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Varios autores clasifican a los sistemas de aprendizaje de diferentes formas:

- Por el esquema matemático subyacente
- Por la naturaleza de los datos
- Por las suposiciones sobre los modelos
- Por las tareas que resuelven

Esquema Matemático Subyacente

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- 1 Modelos geométricos: Los ejemplos definen un espacio de instancias sobre el cual se pueden construir modelos geométricos, e.g., calculando distancias, buscando hiperplanos, encontrando prototípos, etc.
- Normalmente los atributos son numéricos, por lo que se pueden utilizar conceptos geométricos como líneas, planos y distancias, y se pueden hacer transformaciones lineales y usar diferentes medidas de distancia
 - Algunos ejemplos con: Clasificadores lineales, vecinos más cercanos, k-means, SVMs, clasificación basada en prototipos, etc.

Esquema Matemático Subyacente

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

- 2 Modelos probabilistas: En aprendizaje queremos saber cuál es la mejor hipótesis (más probable) dados los datos
- Si $P(D)$ = probabilidad *a priori* de los datos (i.e., cuáles datos son más probables que otros) y $P(D | h)$ = probabilidad de los datos dada una hipótesis, lo que queremos estimar es: $P(h | D)$, la probabilidad posterior de h dados los datos.
 - Esto lo podemos estimar con Bayes.

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)}$$

Esquema Matemático Subyacente

- Para estimar la hipótesis más probable o MAP (*maximum a posteriori hypothesis*):

$$\begin{aligned} h_{MAP} &= \operatorname{argmax}_{h \in H} (P(h | D)) \\ &= \operatorname{argmax}_{h \in H} \left(\frac{P(D|h)P(h)}{P(D)} \right) \\ &\approx \operatorname{argmax}_{h \in H} (P(D | h)P(h)) \end{aligned}$$

Ya que $P(D)$ es una constante independiente de h .

- Si suponemos que las hipótesis son igualmente probables, nos queda la hipótesis de máxima verosimilitud o ML (*maximum likelihood*):

$$h_{ML} = \operatorname{argmax}_{h \in H} (P(D | h))$$

- Aquí se manejan conceptos de probabilidad a priori, máxima verosimilitud, teorema de Bayes, etc.

Esquema Matemático Subyacente

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- ③ Modelos lógicos: Modelos que pueden expresarse desde un punto de vista lógico, incluyendo conjunciones, disjunciones, negación, default, etc.
 - También se les conoce como declarativos, se prestan para dar explicaciones y manejan conceptos como completo y consistente
 - Por ejemplo, reglas de clasificación, árboles de decisión, ILP, patrones frecuentes, subgroup discovery, etc.

Naturaleza de los datos

Esta es una de las clasificaciones más usadas:

- 1 Supervisado: Se tienen datos (X) con una etiqueta (clase) asociada (Y) y se busca encontrar un modelo que dada una instancia de X prediga la etiqueta. Son tareas de clasificación y regresión y se usan conceptos como *overfitting*
- 2 No supervisado: En este caso no se tienen etiquetas asociadas y se busca encontrar una estructura inherente en los datos, organizandolos generalmente por similaridad
- 3 Aprendizaje por Refuerzo: Se busca aprender cómo mapear situaciones a acciones en un proceso iterativo de exploración en el ambiente. Se modelan usando procesos de decisión de Markov (MDPs)

Suposiciones sobre los modelos

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Paramétricos: El modelo resume los datos con un conjunto de parámetros finitos
- No paramétricos: No hacen en general suposiciones fuertes sobre la función o modelo que se quiere encontrar

Modelos Paramétricos

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Estos algoritmos involucran dos pasos:

- 1 Seleccionar la forma de la función
- 2 Aprender los valores de los coeficientes de la función a partir de los datos

Modelos Paramétricos

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Las funciones pueden ser muy variadas, por ejemplo:

- Funciones lineales
- Regresiones logísticas
- Perceptrones
- Naïve Bayes
- Redes neuronales sencillas
- ...

Modelos Paramétricos

Ventajas:

- **Simple:** Son más fáciles de entender y de interpretar sus resultados
- **Velocidad:** Se aprenden rápidamente
- **Datos:** Requieren en general menos datos

Desventajas:

- **Restrictivos:** Al seleccionar un tipo de función particular se restringe lo que se puede aprender
- **Complejidad limitada:** En general son adecuados para problemas más sencillos
- **Ajuste:** Es probable que el modelo seleccionado no ajuste adecuadamente la función subyacente

Modelos No Paramétricos

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- No hacen suposiciones acerca de la forma de la función, sino que se determina con los datos
- Algunos ejemplos son:
 - k-vecinos más cercanos
 - Árboles de decisión
 - SVM
 - Aprendizaje Bayesiano
 - ...

Modelos No Paramétricos

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Ventajas:

- Flexibilidad: Capaces de ajustar una gran cantidad de funciones
- Poder: No hacen grandes suposiciones sobre los modelos
- Desempeño: Pueden obtener mejores desempeños

Desventajas:

- Datos: Requieren una gran cantidad de datos
- Velocidad: En general se tardan en aprender
- Ajuste: Son propensos a realizar sobre-ajustes

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Descripción:** Normalmente es usada como análisis preliminar de los datos (resumen, características de los datos, casos extremos, etc.). Con esto, el usuario se *sensibiliza* con los datos y su estructura. Busca derivar descripciones concisas de características de los datos (e.g., medias, desviaciones estándares, etc.).

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Predicción:** Clasificación y Estimación.
 - **Clasificación:** Los datos son objetos caracterizados por atributos que pertenecen a diferentes clases (etiquetas discretas).
La meta es inducir un modelo para poder predecir una clase dados los valores de los atributos.
Se usan por ejemplo, árboles de decisión, reglas, SVM, etc.

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Predicción:** Clasificación y Estimación.
 - **Estimación o Regresión:** Las clases son continuas. La meta es inducir un modelo para poder predecir el valor de la clase dados los valores de los atributos. Se usan por ejemplo, árboles de regresión, regresión lineal, redes neuronales, LWR, etc.

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Segmentación:** separación de los datos en subgrupos o clases interesantes.

Las clases pueden ser exhaustivas y mutuamente exclusivas o jerárquicas y con traslapes.

Se puede utilizar con un clasificador

Se usan algoritmos de *clustering*, SOM (*self-organization maps*), EM (*expectation maximization*), *k-means*, etc.

Normalmente el usuario tiene una buena capacidad de formar las clases

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Análisis de dependencias:** El valor de un elemento puede usarse para predecir el valor de otro. La dependencia puede ser probabilística, puede definir una red de dependencias o puede ser funcional (leyes físicas).

También se ha enfocado a encontrar si existe una alta proporción de valores de algunos atributos que ocurren con cierta medida de confianza junto con valores de otros atributos.

Se pueden utilizar redes Bayesianas, redes causales, y reglas de asociación.

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Detección de desviaciones**, casos extremos o anomalías: Detectar los cambios más significativos en los datos con respecto a valores pasados o normales. Sirve para filtrar grandes volúmenes de datos que son menos probables de ser interesantes. El problema está en determinar cuándo una desviación es significativa para ser de interés.

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Aprendizaje de la mejor acción a tomar** a partir de experiencia: Esto involucra búsqueda y exploración del ambiente. Esto está relacionado principalmente con aprendizaje por refuerzo, pero también con técnicas como aprendizaje de macro-operadores, *chunking* y EBL.

Tareas de Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- **Optimización y búsqueda:** Existen una gran cantidad de algoritmos de búsqueda tanto determinística como aleatoria, individual como poblacional, local como global, que se utilizan principalmente para resolver algún problema de optimización. Aquí podemos incluir a los algoritmos genéticos, recocido simulado, *ant-colony*, técnicas de búsqueda local, enjambres, etc.

Técnicas Comunes

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Árboles de decisión y reglas de clasificación: realizan cortes sobre una variable (lo cual limita su expresividad, pero facilita su comprensión). Generalmente se usan técnicas heurísticas en su construcción (e.g., ID3, C4.5, CN2). Ver figura.

Técnicas Comunes

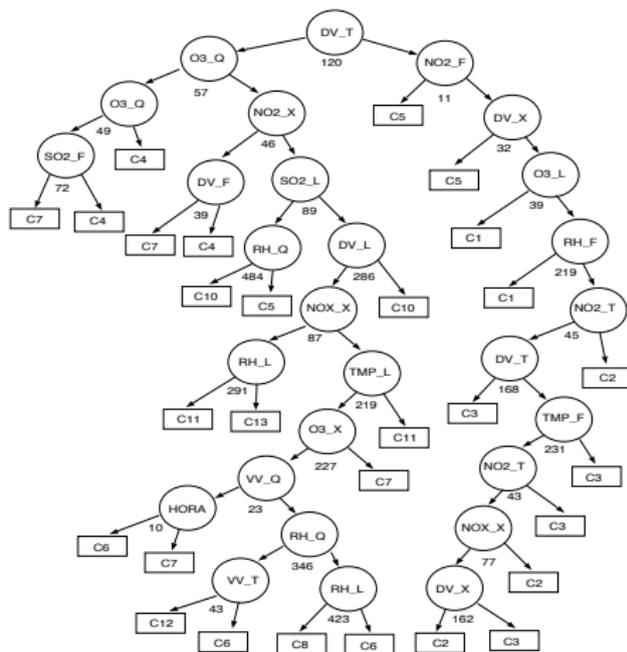


Figure: Predicción de Ozono en la Ciudad de México.

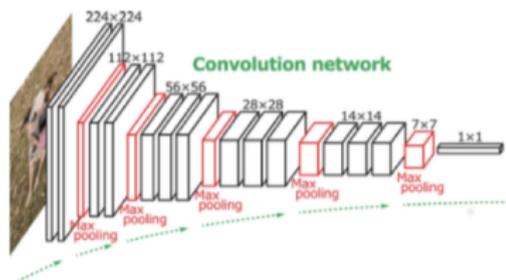
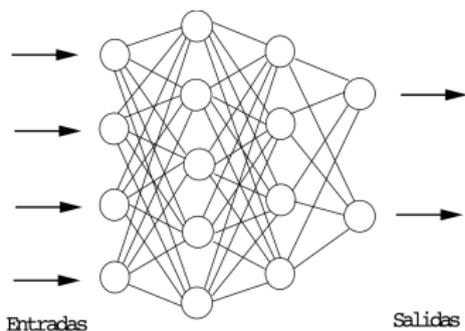
Técnicas Comunes

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

- Métodos de clasificación y regresiones no-lineales: tratan de ajustar combinaciones de funciones lineales y no-lineales, por ejemplo, redes neuronales (e.g., backpropagation), métodos de *splines* adaptativos, etc.



Técnicas Comunes

- Métodos basados en ejemplos prototípicos: se hacen aproximaciones con base en los ejemplos o casos más conocidos (*Exemplar-based learning* y *Case-based reasoning*). El problema es cómo determinar una medida de similitud adecuada.

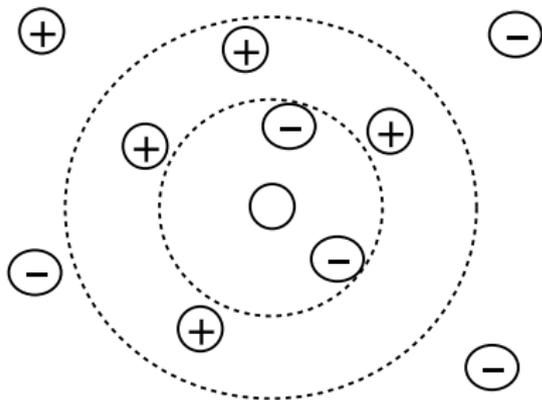


Figure: Aprendizaje basado en instancias.

Técnicas Comunes

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Modelos gráficos de dependencias probabilísticas: básicamente se utilizan redes bayesianas, en donde dado un modelo estructural y probabilístico, se encuentran los valores de ciertas variables dados valores de otras variables. Ver figura.

Técnicas Comunes

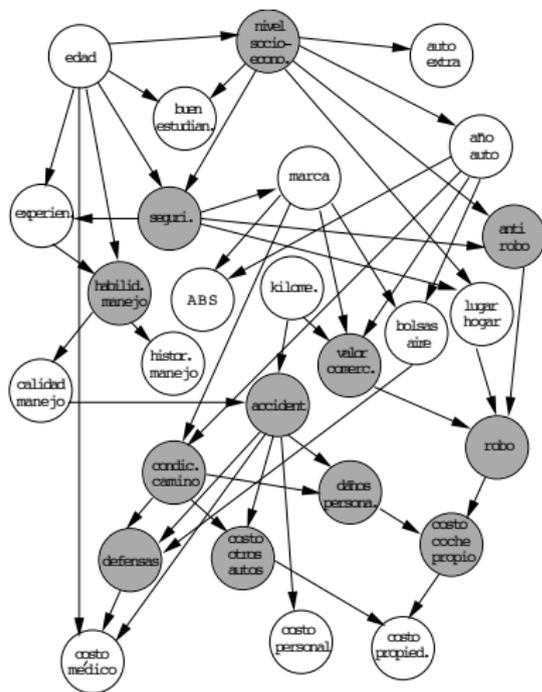


Figure: Red Bayesiana de seguros de coches.

Técnicas Comunes

- Modelos relacionales: Programación Lógica Inductiva (o ILP), en donde la búsqueda del modelo se basa en lógica y heurísticas.

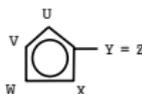
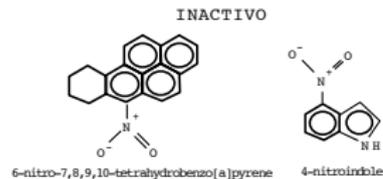
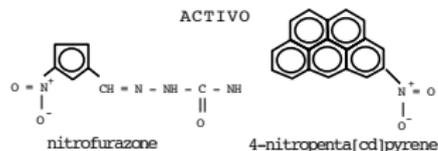


Figure: Predicción de mutagénesis.

Técnicas Comunes

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Reglas de Asociación: Reglas que relacionan un conjunto de pares atributo-valor con otros pares atributo-valor. Por ejemplo:

$$\text{edad}(X, 20 \dots 29) \wedge \text{ingresos}(X, 20K \dots 29K) \\ \Rightarrow \text{compra}(X, CD) \\ [\text{soporte} = 2\%, \text{confianza} = 60\%]$$

Técnicas Comunes

- Clustering: Agrupan datos cuya *distancia* multidimensional dentro de la clase es *pequeña* y entre clases es *grande*.

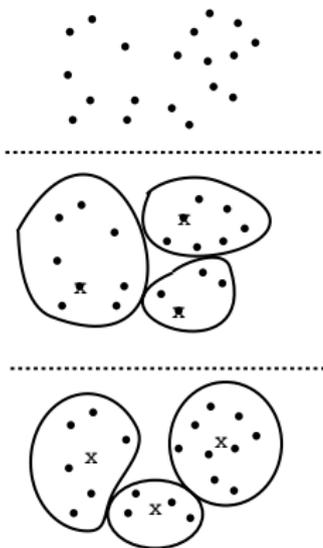


Figure: Ejemplo de Clustering.

Ejemplos

Algunos ejemplos de aplicaciones que usan aprendizaje:

- Sistemas de reconocimiento de voz (e.g., SPHINX, Lee 89),
- Manejo de vehículos autónomos (ALVINN, Pomerleau 89)
- Clasificación de nuevas estructuras de astronomía (SkyCat, Fayyad et al. 95)
- Aprendiendo a jugar Backgammon (TD-Gammon, Tesauro 92)
- Muchas aplicaciones de las que no estamos conscientes (sistema postal, tarjetas de crédito, recomendaciones de Amazon, sistemas de seguridad, uso de combustible en coches, ...)

Ejemplos Recientes

- Control en robots autónomos
- Control en vehículos/drones/... autónomos



- Identificación de edificios famosos (google googles)
- Agrupación de imágenes parecidas (ImageSwirl) y etiquetado automático



- Aprender caras y gatos usando 16,000 procesadores, red neuronal con más de mil millones de conexiones y usando 10 millones de imágenes.

Ejemplos Recientes

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Aprendizaje por Refuerzo con Deep Learning para jugar juegos (Atari, Go, Ajedrez, etc.) y en robótica
- Generación de fotos, videos, personas de manera artificial (*Deep fakes*)
- Traducción automática y generación de texto, LLMs
- Generación de música, pinturas
- Avances en ciencia, e.g., plagamiento de proteínas
- ...

Aplicaciones

Existe una gran cantidad de aplicaciones tales como:

- Astronomía
- Biología molecular
- Aspectos climatológicos
- Medicina
- Industria y manufactura
- Mercadotécnica
- Inversión en casas de bolsa y banca
- Detección de fraudes y comportamientos inusuales
- Análisis de canastas de mercado
- Aprendizaje de tareas en robótica
- ...

Áreas Relacionadas

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Inteligencia Artificial
- Métodos Bayesianos
- Teoría de complejidad computacional
- Teoría de control
- Teoría de información
- Filosofía
- Psicología y neurobiología
- Estadística

Consideraciones

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

Algunos puntos que aclarar cuando se quiere correr o desarrollar algún algoritmo de aprendizaje son:

- ¿Qué algoritmos existen para resolver cierta tarea? ¿cuándo y cómo usarlos? ¿qué propiedades tienen?
- ¿Cuántos datos o tiempo de entrenamiento necesito? ¿qué tanta confianza puedo tener en los resultados?
- ¿Cómo y cuándo usar conocimiento del dominio?
- ...

Se espera que algunas de estas preguntas las puedan resolver al final del curso.

Retos

- Volúmen de datos (giga, terabytes, ... ó cero, unidades, decenas)
- Alta dimensionalidad (y pocos datos)
- Sobreajuste (*overfitting*)
- Datos y conocimiento dinámicos
- Ruido, incertidumbre y datos incompletos y/o esparsos
- Relaciones complejas entre campos, jerarquías, etc.
- Interpretación y explicación de los resultados
- Incorporación de conocimiento del dominio
- Interacción activa del usuario
- Auto-aprendizaje y *Life-long learning*
- Causalidad y sentido común

Aprendizaje Inductivo

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- El aprendizaje inductivo puede verse como el proceso de aprender una función.
- Por ejemplo, en aprendizaje supervisado, al elemento de aprendizaje se le dá un valor correcto (o aproximadamente correcto) de una función a aprender para entradas particulares y cambia la representación de la función que está infiriendo, para tratar de aparear la información dada por la retroalimentación que ofrecen los ejemplos.

Aprendizaje

- Un ejemplo es un par $(\mathbf{x}, f(\mathbf{x}))$, donde \mathbf{x} es la entrada (que generalmente es un vector) y $f(\mathbf{x})$ la salida.
- El proceso de inferencia inductiva pura (o inducción) es: Dada una colección de ejemplos de f , regresar una función h tal que se aproxime a f . A la función h se le llama la *hipótesis*.
- En principio existen muchas posibilidades para escoger h , cualquier preferencia se llama *bias* o sesgo. Todos los algoritmos de aprendizaje exhiben algún tipo de sesgo.
- La selección de una representación para la función deseada es probablemente el factor más importante en el diseño de un sistema de aprendizaje.

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Desde un punto de vista más tradicional (hablando de representaciones simbólicas/reglas,...), podemos decir que una buena parte de ML está dedicada a inferir reglas a partir de ejemplos.
- Descripciones generales de clases de objetos, obtenidas a partir de un conjunto de ejemplos, pueden ser usadas para clasificar o predecir.
- En general, el interés no está en aprender conceptos de la forma en que lo hacen los humanos, sino aprender representaciones simbólicas de ellos.

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Angluin y Smith (83) listan cinco elementos que deben de especificarse para caracterizar un problema de inferencia inductiva:

- 1 La clase de reglas
- 2 El espacio de hipótesis
- 3 El conjunto de ejemplos y su presentación
- 4 La clase del método de inferencia
- 5 El criterio de éxito

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

1) La clase de reglas:

- La clase de reglas denota la clase de funciones o lenguaje bajo consideración.
- Por ejemplo, todas las expresiones regulares sobre un alfabeto específico, lenguajes libres de contexto, funciones recursivamente enumerables, programas en Prolog, etc.

Aprendizaje

2) El espacio de hipótesis:

- El espacio de hipótesis es el conjunto de descripciones tal que cada regla en la clase tiene por lo menos una descripción en el espacio de hipótesis.
- Diferentes espacios de hipótesis pueden usarse para la misma clase de reglas.
- El lenguaje de hipótesis debe de tener descripciones para todas las reglas en la clase, pero puede contener más.
- Por conveniencia, normalmente se supone que el lenguaje descrito por el espacio de hipótesis (i.e., el lenguaje de hipótesis) es el mismo que el de la clase de reglas

Aprendizaje

- 1 Lenguaje de Hipótesis: Sintáxis usada en la construcción de hipótesis
- 2 Espacio de Hipótesis: Conjunto de todas las posibles hipótesis dentro del lenguaje de hipótesis
 - El lenguaje de hipótesis determina el espacio de hipótesis del cual el método selecciona sus reglas e impone restricciones/preferencias en lo que se puede aprender y qué estrategias de razonamiento utilizar
 - Al escoger un lenguaje, debemos de considerar lo que queremos que el sistema realice, la información que se le debe de proporcionar y si lo va a resolver a tiempo
 - Existe, como en los métodos de inferencia de KR, un *balance fundamental* entre la expresividad y la eficiencia

Espacio de Hipótesis

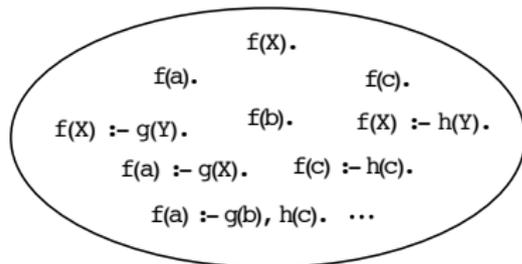
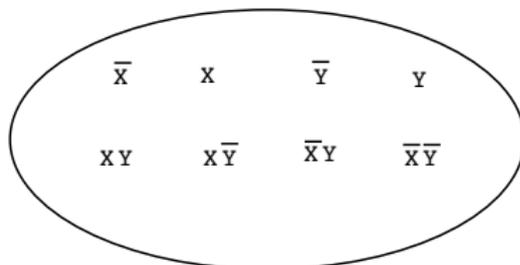


Figure: El espacio de hipótesis depende de la expresividad del lenguaje.

Espacio de Hipótesis

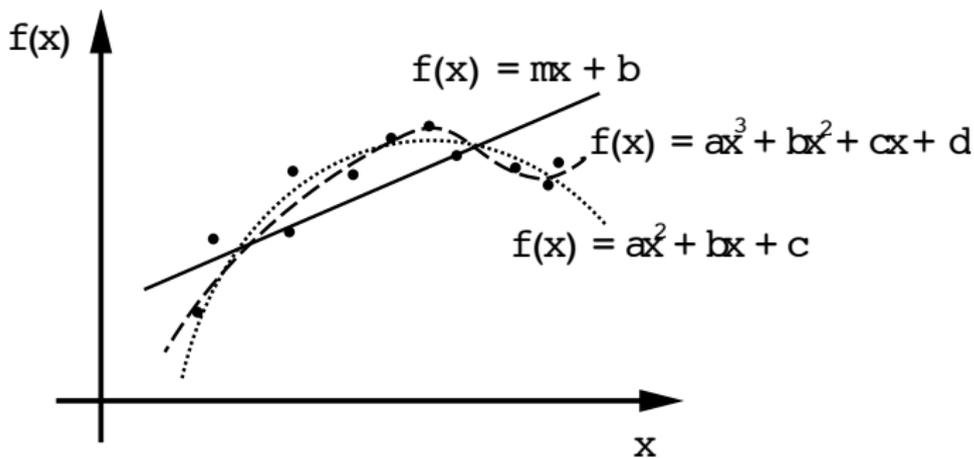


Figure: Qué tan bien se ajusta el modelo depende de la expresividad del lenguaje.

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- El lenguaje de hipótesis depende del área de aplicación. Una vez definido, una buena parte del tiempo se dedica a seleccionar cuidadosamente las estructuras de conocimiento adecuadas para la tarea de aprendizaje.
- Este tiempo se vuelve más crítico cuando el lenguaje de hipótesis restringe la expresividad y el conocimiento del dominio tiene que adaptarse al formalismo adoptado.
- El proceso de inducción puede verse como una búsqueda de hipótesis o reglas.

Aprendizaje como Búsqueda

- El espacio puede buscarse sistemáticamente, hasta encontrar la regla adecuada.
- Podemos tener una enumeración de descripciones, digamos d_1, \dots, d_n , tal que cada regla en el espacio de hipótesis tiene una o más descripciones en esta enumeración.
- Dada una colección de ejemplos, *identificación en el límite* recorre esta lista encontrando la primera descripción, digamos d_i , que es compatible con los ejemplos vistos y conjetura a d_i .
- Sin embargo es impráctico a menos que se tenga un número limitado de descripciones

Aprendizaje como Búsqueda

- Normalmente es crucial estructurar el espacio de hipótesis. Esto se puede hacer con un modelo de generalización.
- A grandes razgos una regla R_1 es más general que otra regla R_2 (o R_2 es más específica que R_1), si en cualquier mundo R_1 puede mostrar los mismos resultados que R_2 .
- Una estructuración permite cortar ramas durante la búsqueda sabiendo que especializaciones o generalizaciones de reglas hereden alguna propiedad.
- Las propiedades más comunes son: Incapacidad de cubrir un ejemplo conocido como verdadero o probar un ejemplo conocido como falso.

Aprendizaje

3) Conjunto de ejemplos y su presentación:

- Existen diferentes tipos de presentación de datos y sus efectos en el proceso de inferencia
- Los ejemplos pueden dar una retroalimentación directa o indirecta (e.g., al aprender a jugar un cierto juego, la retroalimentación se puede dar en cada jugada o al final del juego o después de un conjunto de jugadas que provocaron una pérdida de material, etc.)
- Aquí, surge el problema de *asignación de crédito* (cuál jugada es responsable del éxito o fracaso).
- Una presentación puede consistir en: (i) Sólo ejemplos positivos y (ii) positivos y negativos.

Aprendizaje

- Casi todos los algoritmos requieren presentaciones admisibles (para cada regla falsa consistente con los ejemplos existe un ejemplo negativo que la refuta)
Popper: Las teorías deben de ser refutables con hechos
- Los ejemplos se usan para probar y formar hipótesis (sobre un espacio de ejemplos)
- La selección puede ser por un oráculo, el medio ambiente, en forma aleatoria o por el sistema y se tiene que tener un *balance entre exploración y explotación*
- Una “buena” selección (posiblemente con conocimiento del dominio) de ejemplos puede mejorar el desempeño de un sistema (ver *Active Learning*)
- Se supone que la distribución que siguen los ejemplos es similar a la de ejemplos futuros.

Aprendizaje

4) Métodos de inferencia:

- Intuitivamente un método de inferencia es un proceso computacional de algún tipo que lee ejemplos y produce hipótesis del espacio de hipótesis.
- Existe una gran cantidad de métodos:
 - Realizar ajustes graduales con base en refuerzos sobre predicciones sucesivas (e.g., RL, NN, etc.).
 - Construcción incremental de hipótesis tratando de cubrir la mayor cantidad de ejemplos (e.g., reglas de clasificación, ILP) o con base en mejores particiones de ejemplos (e.g., TDIDT).
 - Guardar ejemplos prototípicos (e.g., CBR, IBL).
 - Buscar relaciones entre variables (e.g., BN).
 - Combinar o modificar hipótesis promisorias (e.g., GA).
 - ...

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

5) Criterio de éxito:

- Un componente importante dentro de la especificación de un problema de inferencia es el criterio de éxito (Identificación en el límite es uno de ellos).
- Recientemente Valiant, propuso un criterio de identificación correcta de una regla a partir de ejemplos usando un criterio estocástico.
- La idea es que después de un muestreo aleatorio de ejemplos positivos y negativos de una regla, un procedimiento de identificación debe de producir una regla que con “alta probabilidad” no sea “muy diferente” de la regla correcta (PAC)

PAC Learning

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

- Se basa en dos parámetros: ϵ y δ .
 - ϵ es una medida de *tolerancia* o un límite de la diferencia permitida entre la regla correcta y la hipótesis generada.
 - δ es una medida de *confianza*.
- Informalmente, un procedimiento de identificación se dice ser probablemente aproximadamente correcto o PAC si la diferencia entre la regla correcta y la hipótesis es menos que ϵ con probabilidad mayor a $1 - \delta$.

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

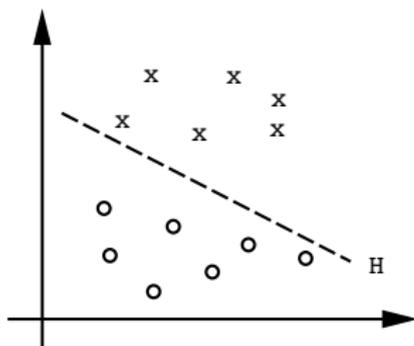
Espacio de
Versiones

En la práctica queremos ciertas garantías de la calidad de la hipótesis. Las más comunes son que sea completo y consistente (ver figura):

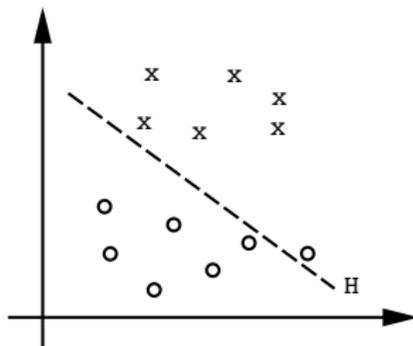
- Una hipótesis es *completa* si cubre todos los ejemplos positivos
- Una hipótesis es *consistente* si no cubre a ninguno de los ejemplos negativos

A veces el usuario determina el criterio de paro. Si el sistema genera sus propios ejemplos, éste lo determina.

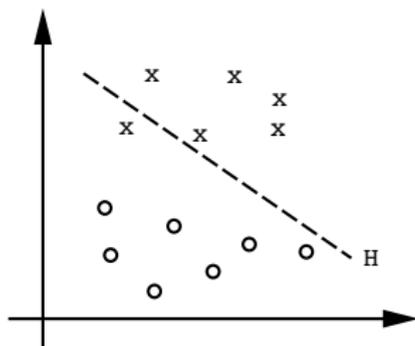
Completo y Consistente



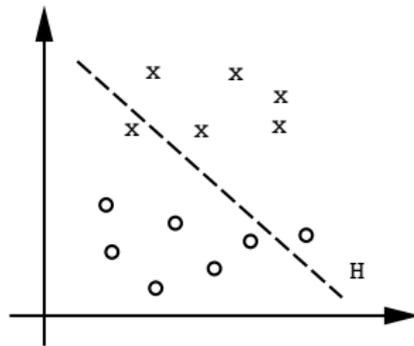
Completo y Consistente



Completo e Inconsistente



Incompleto y Consistente



Incompleto e Inconsistente

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Desde el punto de vista de lógica tratamos de encontrar una expresión lógica de un predicado meta (M) y que nos sirva para clasificar ejemplos correctamente.
- Cada hipótesis propone una expresión, y la llamaremos la definición candidata del predicado meta.
- Como lo mencionamos antes, el espacio de hipótesis H es el conjunto de todas las hipótesis $\{H_1, H_2, \dots, H_n\}$, que el algoritmo de aprendizaje está diseñado a producir.

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Cada hipótesis predice que un cierto conjunto de ejemplos. Aquellos que satisfacen su definición candidata, son ejemplos del predicado meta. A estos ejemplos también se les llama la *extensión* del predicado.
- En este sentido dos hipótesis son lógicamente equivalentes si tienen la misma extensión.
- Los ejemplos son objetos para los cuales el predicado meta puede o no satisfacerse.
- Una hipótesis es consistente lógicamente con los ejemplos si se cumple o no dependiendo si el ejemplo es positivo o negativo.

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Las condiciones por las cuales una hipótesis puede ser inconsistente con algún ejemplo son:

- Un ejemplo es un *negativo falso* para la hipótesis (i.e., la hipótesis dice que debe de ser negativo y en realidad es positivo)
- Un ejemplo es un *positivo falso* para la hipótesis (i.e., la hipótesis dice que debe de ser positivo y en realidad es negativo)

Si supones que el ejemplo es una observación correcta, un falso positivo o negativo implica que la hipótesis tiene que ser rechazada.

Aprendizaje

Desde un esquema de lógica, podemos caracterizar el aprendizaje inductivo eliminando gradualmente hipótesis que sean inconsistentes con los ejemplos

Ejemplo: cartas

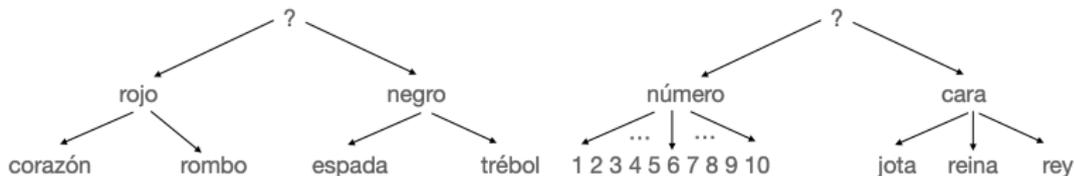
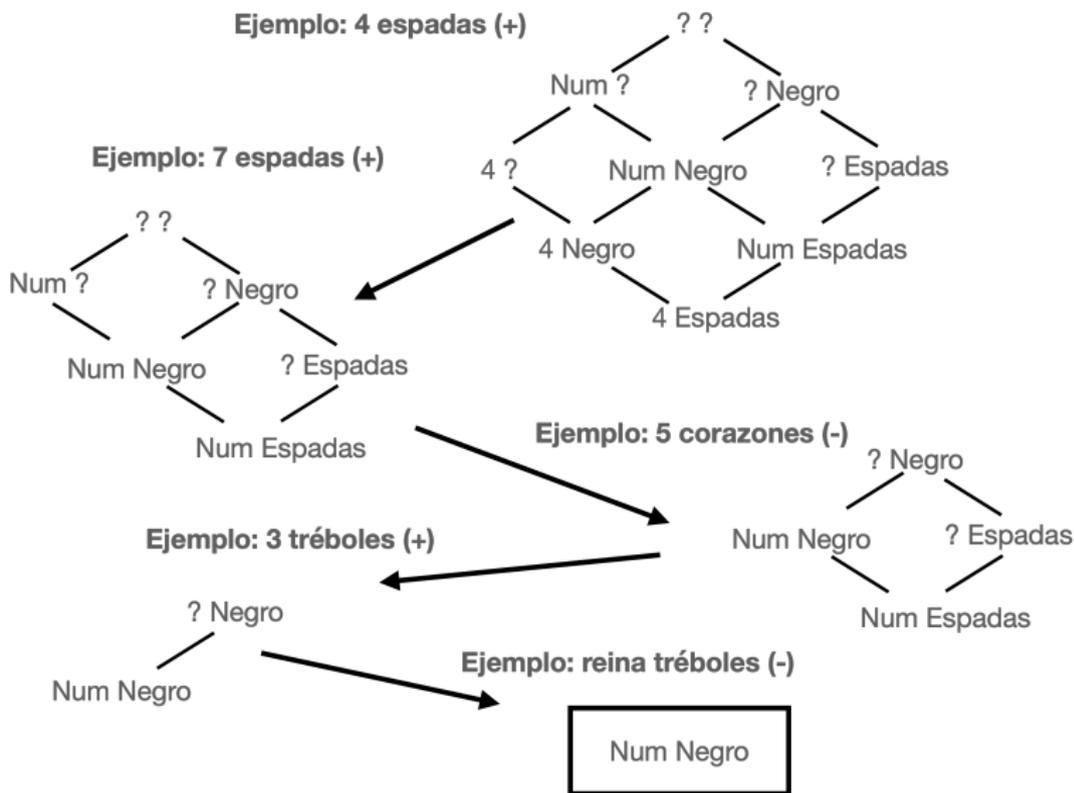


Figure: Espacio de hipótesis.

Sin embargo, el espacio es muy grande (e incluso infinito en muchos casos) haciendo su implantación directa impráctica (sino imposible).

Proceso de eliminación de hipótesis



Búsqueda de la Mejor Hipótesis Actual

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- La idea es mantener una sola hipótesis e ir la ajustando conforme nuevos ejemplos se consideran manteniendo consistencia.
- El algoritmo básico puede encontrarse descrito desde 1943 (John Stuart Mill).
- Si tenemos una hipótesis H_r y recibimos un negativo falso, entonces la extensión de la hipótesis debe aumentarse para incluirlo. A esto se le llama *generalización*.
- Si tenemos un positivo falso, entonces la extensión de la hipótesis debe reducirse para excluirlo. A esto se le llama *especialización*.

Aprendizaje

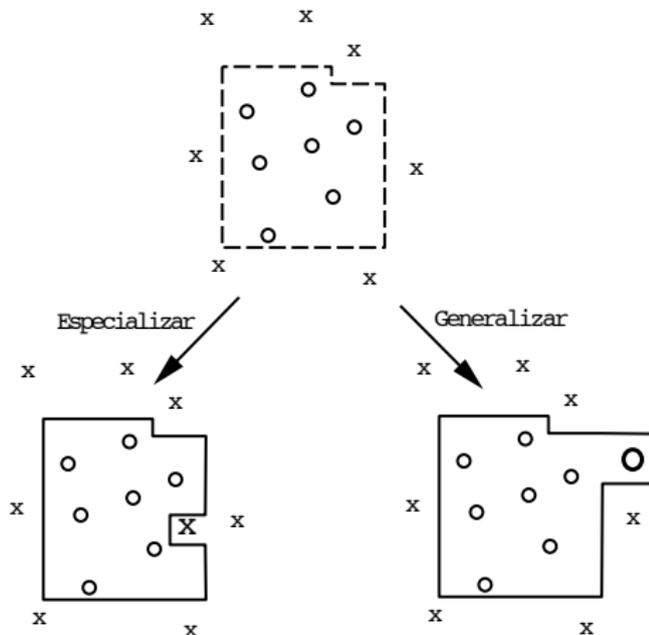


Figure: Proceso de especializar y generalizar.

Aprendizaje

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Definimos generalización y especialización como operaciones que cambian la *extensión* de una hipótesis
- Intuitivamente H_1 es más general que H_2 si la “cubre”
- Sintácticamente, una posible forma de generalizar es *eliminando condiciones* volviendo las definiciones más débiles y por lo tanto cubriendo un conjunto mayor de ejemplos o añadiendo disjunciones
- De forma dual, podemos especializar añadiendo condiciones o eliminando disjunciones.

Algoritmo de Mejor Hipótesis Actual

función *mejor-hipótesis-actual*(ejemplos)
regresa una hipótesis

$H \leftarrow$ cualquier hipótesis consistente con el primer ejemplo
 en *ejemplos*

para cada uno de los ejemplos restantes **hacer**

si e es positivo falso para H **entonces**

$H \leftarrow$ *selecciona* una especialización de H consistente
 con *ejemplos*

sino si e es negativo falso para H **entonces**

$H \leftarrow$ *selecciona* una generalización de H consistente
 con *ejemplos*

si no se puede construir una especialización /
 generalización consistente **entonces** *falla*

regresa H

Algoritmo de Mejor Hipótesis Actual

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- El algoritmo sigue básicamente una búsqueda en profundidad. Podemos empezar con una generalización o con una especialización consistente con los ejemplos
- Estas ideas se han usado en varios sistemas de aprendizaje, sin embargo, tiene algunos problemas:
 - ① Verificar todas las instancias anteriores cada vez que se hace una modificación
 - ② Es difícil encontrar buenas heurísticas de búsqueda y el hacer *backtracking* puede volverse “eterno” (se toman decisiones locales sin suficiente información)

Algoritmo de Mejor Hipótesis Actual

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Alternativamente, podemos buscar a lo ancho (i.e., mantener varias hipótesis a la vez).
- Si lo hacemos de específico a general, podemos tratar de mantener todas las generalizaciones más específicas que sean consistentes con las observaciones (o hacerlo de general a específico)

Algoritmo Específico a General BFS

$H \leftarrow$ el conjunto de generalizaciones más específicas consistentes con los ejemplos vistos en *ejemplos*
para cada uno de los ejemplos restantes **hacer**
si e es positivo falso para alguna hipótesis en H
entonces $H \leftarrow$ las hipótesis que no son consistentes con el ejemplo
sino si e es negativo falso para alguna hipótesis en H
entonces $H \leftarrow$ generaliza miembros de H , pero sólo al punto de aceptar el ejemplo
 Elimina de H cualquier elemento que sea
 (i) más general que otro elemento o
 (ii) aparea otros negativos
si no se puede construir una generalización consistente
entonces *falla*
regresa H

Introducción

Aprendizaje

Aprendizaje Inductivo

Espacio de Versiones

Algoritmo Específico a General BFS

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Los ejemplos positivos fuerzan las generalizaciones y los negativos eliminan generalizaciones. Sigue un proceso monótono de específico a general.
- Sin embargo, cada vez que generalizamos, seguimos teniendo que verificar consistencia con todos los ejemplos positivos.
- Una alternativa es mantener todas y sólo aquellas hipótesis que son consistentes con todos los datos.

Búsqueda con el Menor Compromiso

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Con cada instancia nueva, o no se hace nada, o se eliminan algunas hipótesis.
- Suponiendo que el espacio de hipótesis inicial tiene una respuesta correcta, la disjunción de hipótesis reducida, va a seguir teniendola. Al conjunto de hipótesis consistentes con los ejemplos se le llama *espacio de versiones* (*version space*).
- Una propiedad importante del algoritmo es que es *incremental* (nunca se tiene que regresar para examinar ejemplos viejos).

Búsqueda con el Menor Compromiso

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Problema obvio: Si el espacio es gigantesco, cómo podemos escribir la disjunción completa de hipótesis.
- El punto es que no la tenemos que escribir! Se puede hacer una analogía con números reales. Si queremos representar todos los números entre 1 y 2 $\Rightarrow [1, 2]$.
- Esto lo podemos hacer porque existe un ordenamiento.
- La generalización / especialización también nos da un orden, en este caso un orden parcial

Orden Parcial entre Hipótesis

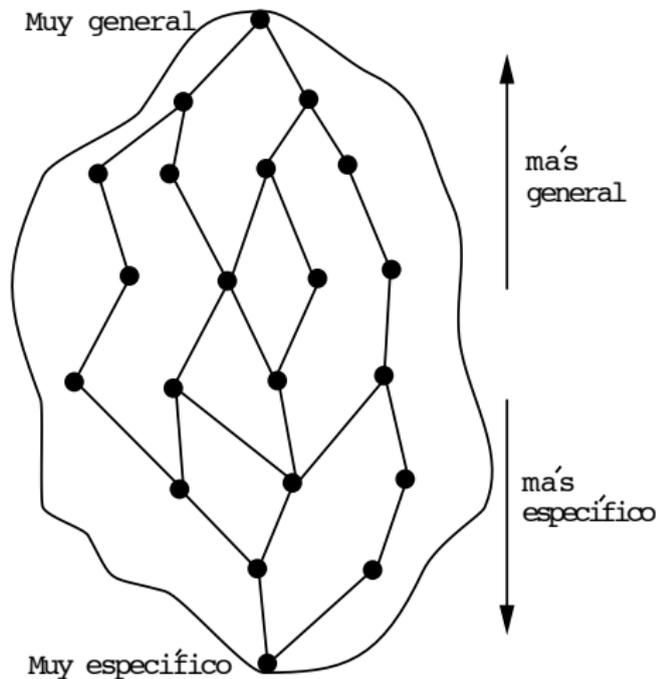


Figure: Orden parcial entre hipótesis.

Orden Parcial entre Hipótesis

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

- En este caso, las fronteras no son puntuales, sino conjuntos de hipótesis o conjuntos frontera (*boundary sets*).
- Lo bueno es que podemos representar todo el espacio de versiones usando sólo 2 conjuntos de frontera:
 - La frontera más general (el conjunto **G**)
 - La frontera más específica (el conjunto **S**)
- Todo lo que está entre **S** y **G** está garantizado a ser consistente con los ejemplos (el tamaño de **S** y **G** depende del lenguaje).

Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Resumiendo:

- El espacio de versiones actual es el conjunto de hipótesis consistente con todos los ejemplos vistos
- Cada elemento del conjunto **S** es consistente con todas las observaciones hasta el momento y no existen hipótesis consistentes que sean más específicas
- Cada elemento del conjunto **G** es consistente con todas las observaciones hasta el momento y no existen hipótesis consistentes que sean más generales

Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

El espacio de versiones inicial tiene que representar a todas las hipótesis. Esto se puede lograr haciendo $\mathbf{G} = True$ (contiene todo) y $\mathbf{S} = False$ (su extensión es vacía).

Se tienen que cumplir dos propiedades:

- Toda hipótesis consistente está entre \mathbf{S} y \mathbf{G}
- Toda hipótesis entre \mathbf{S} y \mathbf{G} es consistente

Lo *único* que queda es como actualizar \mathbf{S} y \mathbf{G} . Si S_i es una de las hipótesis en \mathbf{S} y G_j una en \mathbf{G}

Espacio de Versiones

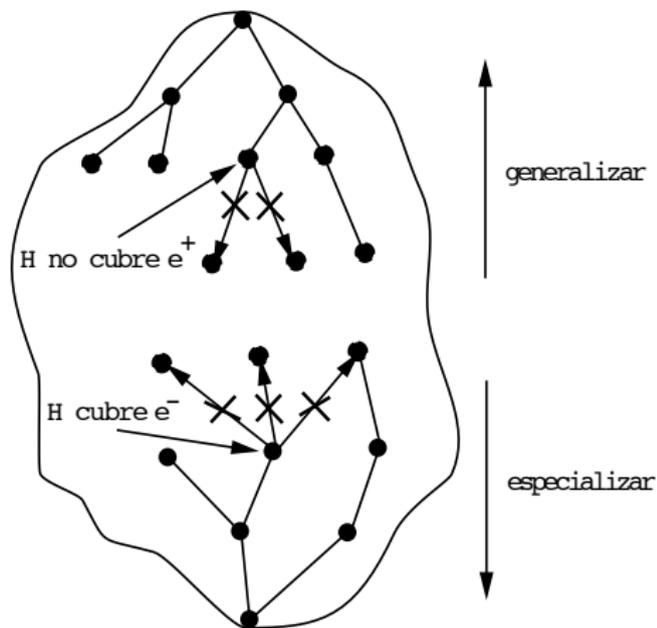


Figure: Actualización en el espacio de versiones

Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- 1 Positivo falso para S_i : S_i es muy general, pero por definición no existe una especialización de S_i consistente, por lo que la eliminamos
- 2 Negativo falso para S_i : S_i es muy específico y tenemos que substituirlo por su generalización inmediata
- 3 Positivo falso para G_i : G_i es muy general y tenemos que substituirlo por su especialización inmediata
- 4 Negativo falso para G_i : G_i es muy específico, pero por definición no existe una generalización de G_i consistente, por lo que la eliminamos

Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Continuamos con estas operaciones hasta que:

- 1 En el espacio de versiones queda una sola hipótesis
- 2 El espacio de versiones se *colapsa* (**S** y **G** se vuelven vacías), por lo que no hay una hipótesis consistente con los ejemplos
- 3 Se acabaron los ejemplos y tenemos varias hipótesis en el espacio de versiones, i.e., una disjunción de hipótesis (con un nuevo ejemplo, si todas las hipótesis están de acuerdo, clasificamos el ejemplo, sino, podemos tomar un *voto* mayoritario)

Algoritmo de Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

Inicializa los conjuntos **S** y **G** con *Falso* y *Verdadero*

Para cada ejemplo e_i subsecuente

si e_i es negativo **entonces**

- Mantén en **S** sólo las hipótesis que no cubren a e_i
- Especializa en **G** aquellas hipótesis que cubran a e_i , pero sólo al punto para no cubrirlo y que sigan siendo las más generales
- Elimina de **G** cualquier elemento más específico

sino si e_i es positivo **entonces**

- Mantén en **G** sólo las hipótesis que cubren a e_i
- Generaliza en **S** aquellas hipótesis que no cubran a e_i , pero sólo al punto para cubrirlo y que sigan siendo las más específicas
- Elimina de **S** cualquier elemento más general

Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

Ventajas:

- Detecta cuando acaba (cuando los ejemplos son suficientes)
- Cuando hay sólo algunos ejemplos, sigue dando resultados

Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
InductivoEspacio de
Versiones

Problemas:

- 1 Supone que podemos calcular la relación *más-general-que*
- 2 Si el dominio tiene ruido o insuficientes atributos para una clasificación exacta, el espacio de versiones se colapsa
- 3 Si permitimos una disjunción ilimitada en el espacio de hipótesis, el conjunto **S** va a tener una disjunción de los ejemplos positivos y **G** va a tener la negación de la disjunción de los ejemplos negativos

Espacio de Versiones

Introducción

Aprendizaje

Aprendizaje
Inductivo

Espacio de
Versiones

- Para el manejo de ruido no existe una solución general (pero vamos a ver varias).
- Para el caso de disjunciones ilimitadas, podemos usar una jerarquía de generalizaciones.
- El algoritmo se usó en Meta-Dendral y en LEX
- Puede servir para generar automáticamente ejemplos que dividan el espacio de búsqueda.
- Unos años después, se desarrolló una generalización del algoritmo que permite manejar ruido, valores faltantes e incorporar conocimiento del dominio usando conjuntos de fronteras que permiten cierta cantidad de ruido y métodos para fusionar elementos de las fronteras.