TEMA 2 APRENDIZAJE INDUCTIVO

Francisco José Ribadas Pena, Santiago Fernández Lanza

Modelos de Razonamiento y Aprendizaje
5º Informática
ribadas@uvigo.es, sflanza@uvigo.es

28 de enero de 2013

2.1 Introducción

- Sigue el esquema del razonamiento inductivo:

```
Conjunto Premisas (conocimiento particular - ejemplos)
-------------------------------------------------------
Conclusión (conocimiento general - hipótesis)

(la conclusión se sigue de las premisas con una probabilidad)
```

- Identifica conocimiento general *(hipótesis)* a partir de conocimiento particular *(ejemplos)*
- Puede verse como el proceso de aprender una función $\mathbf{c}(\mathbf{x})$ a partir de un conjunto de ejemplos.
  - En aprendizaje supervisado: Un ejemplo es un par $[\mathbf{e}, \mathbf{c}(\mathbf{e})]$
  - **Objetivo:** dado un conj. de ejemplos encontrar una función $\mathbf{h}$ *(hipótesis)* que aproxime a $\mathbf{c}$
  - Caso más simple: funciones booleanas (SALIDA: verdadero, falso)
- Propiedades de las hipótesis:
  - **Hipotesis Completa:** cubre todos los ejemplos positivos
  - **Hipotesis Correcta:** no cubre ningún ejemplo negativo
- **Hipótesis de partida en el aprendizaje inductivo:**
  Si una hipótesis describe bien el concepto meta de acuerdo a un número suficientemente grande de ejemplos de entrenamiento, también lo describirá bien ante ejemplos futuros nunca vistos
  - Importante: capacidad de generalizar
- Capacidad de aprendizaje viene determinada por:
  1. Lenguaje de representención de las hipótesis (lenguaje de hipótesis)
     - Determina el espacio de hipótesis (*language bias*)
  2. Procedimiento (algoritmo de búsqueda) para inferir hipótesis a partir de ejemplos (*search bias*)
- Ambos aspectos determinan preferencias en el aprendizaje (sesgos o *bias*)

- **Espacio de Hipótesis:** Conjunto de todas las posibles representaciones de hipótesis que se pueden formular empleando el lenguaje de hipótesis. $\mathbf{H} = \{H_1, H_2, ..., H_n\}$
  - Lenguaje determina espacio de hipótesis $\Rightarrow$ impone restricciones (sesgo)
  - Contraposición Expresividad vs. Eficiencia
  - Depende del área de aplicación, del método de aprendizaje, etc...
- Inducción puede verse como un proceso de búsqueda en un espacio de hipótesis.
  - Búsqueda sistemática: general, pero demasiado costosa (espacio búsqueda muy grande)
  - Necesidad estructurar espacio de hipótesis $\Rightarrow$ uso de la relación de generalización

2.2 Métodos generales

- Técnicas generales de aprendizaje inductivo.
  - Aplicables en cualquier problema
  - No eficientes en problemas de tamaño real
- **Extensión de una hipótesis:** conjunto de ejemplos que dicha hipótesis predice correctamente
  - Dos hipótesis son equivalentes si tienen idéntica extensión
  - Hipótesis H_i es consistente con un ejemplo si lo predice correctamente
  - Hipótesis inconsistentes (deben ser rechazadas):
    - *Falso positivo:* H_i predice positivo cuando es negativo
    - *Falso negativo:* H_i predice negativo cuando es positivo
- **Relación de generalización:** ''Una hipótesis $\mathbf{H_1}$ es *más general* que otra $\mathbf{H_2}$ ($\mathbf{H_1} \geq \mathbf{H_2}$), si cualquier ejemplo que sea consistente con $\mathbf{H_2}$ lo es también con $\mathbf{H_1}$''
  - Relación de orden parcial
  - Permite ordenar las hipótesis

2.2.1 Búsqueda mejor hipótesis

- **IDEA:** Mantener una ÚNICA hipótesis e ir ajustándola a medida que aparecen nuevos ejemplos.
- **ALGORITMO:**

  Dada la hipótesis actual H_i.

  Para cada nuevo ejemplo de entrenamiento e, se construye H_{i+1}:

  1. Si $H_i(e)$ es *falso negativo*:
     - aumentar la extensión de H_i para incluirlo (generalización)
  2. Si $H_i(e)$ es *falso positivo*:
     - reducir la extensión de H_i para eliminarlo (especialización)
  3. Si predicción correcta: no cambiar

- **Generalización/especialización**: operaciones que cambian la *extensión* de una hipótesis.
  - Dependen del lenguaje de representación de hipótesis
  - *Ejemplo:* Usando representaciones lógicas (conjunción de condiciones):
    - **Generalización:** eliminar condiciones
    - **Especialización:** añadir condiciones
- PROBLEMAS:
  - Necesidad de verificar la corrección de la nueva hipótesis con respecto a TODOS los ejemplos anteriores en cada iteración (mantener consistencia)
  - Difícil encontrar heurísticas que guíen la búsqueda

2.2.2 Búsqueda por el menor compromiso (espacio de versiones)

- **IDEA:** Mantener el conjunto de TODAS las hipótesis consistentes con los ejemplos estudiados hasta el momento.
  - Con cada nuevo ejemplo, se irán eliminando hipótesis no consistentes
- **Espacio de versiones:** Conjunto de todas las hipótesis consitentes con los ejemplos recibidos hasta el momento
- **Representación del conjunto de hipótesis:**
  - *1ª Aprox.:*Disyunción de todas las posibles hipótesis consistentes
  - *Problema:* Espacio muy grande $\Rightarrow$ usar representación más compacta
  - *Solución:* Usar un ordenamiento (relación de generalización)
    - Analogía con nº reales : [1,2] representa todos los reales entre 1 y 2
    - Manejar sólo dos subconjuntos de hipótesis (fronteras)
  - **Conjuntos frontera:**
    - *Conjunto* $\mathbf{G}$: Frontera hipótesis más generales
      - ◇ Toda hipótesis de $\mathbf{G}$ es consistente con los ejemplos vistos y no hay hipótesis consistentes más generales.
      - ◇ Resume toda la información conocida sobre ejemplos negativos (los delimita)
    - *Conjunto* $\mathbf{S}$: Frontera hipótesis más específicas
      - ◇ Toda hipótesis de $\mathbf{S}$ es consistente con los ejemplos vistos y no hay hipótesis consistentes más específicas.
      - ◇ Resume toda la información conocida sobre ejemplos positivos (los delimita)
    - Toda hipótesis entre $\mathbf{G}$ y $\mathbf{S}$ es consistente

- **Funcionamiento:** Con cada nuevo ejemplo, se eliminan las hipótesis no consistentes
  - $\mathbf{G}$ se hace más específico y $\mathbf{S}$ se hace más general
  - Ejemplos positivos: mueven $\mathbf{S}$ hacia $\mathbf{G}$ (hacia arriba)
  - Ejemplos negativos: mueven $\mathbf{G}$ hacia $\mathbf{S}$ (hacia abajo)
- **Espacio de versiones inicial:** Debe representar a todas las posibles hipótesis consistentes.
  - $\mathbf{G}$: mayor conjunto posible de hipótesis consistentes
    $\rightarrow$ Usando representación lógica: $\mathbf{G}$ = verdadero (contiene todo)
  - $\mathbf{S}$: menor conjunto posible de hipótesis consistentes
    $\rightarrow$ Usando representación lógica: $\mathbf{S}$ = falso (no contiene nada)

- **ALGORITMO:** Actualización de $\mathbf{G}$ y $\mathbf{S}$

  Para cada nuevo nuevo ejemplo e:

  1. Si e es un ejemplo positivo
     a) Eliminar de $\mathbf{G}$ todas las hipótesis inconsistentes con e (falso negativo)
     b) Para cada s_i de $\mathbf{S}$ inconsistente (falso negativo)
        1) Sustituir s_i por sus generalizaciones más inmediatas
           - que sean consistentes con e
           - que exista en $\mathbf{G}$ hipótesis más generales que ellas
        2) Eliminar las hipótesis nuevas que sean más generales *(menos específicas)* que otras ya presentes en $\mathbf{S}$
  2. Si e es un ejemplo negativo
     a) Eliminar de $\mathbf{S}$ todas las hipótesis inconsistentes con e (falso positivo)
     b) Para cada g_i de $\mathbf{G}$ inconsistente (falso positivo)
        1) Sustituir g_i por sus especializaciones más inmediatas
           - que sean consistentes con e
           - que exista en $\mathbf{S}$ hipótesis más específicas que ellas
        2) Eliminar las hipótesis nuevas que sean más específicas *(menos generales)* que otras ya presentes en $\mathbf{G}$

- Finalización:
  - Si $\mathbf{G}$ y $\mathbf{S}$ convergen $\Rightarrow$ encontrada la única hipótesis en el lenguaje usado consistente con los datos
  - Si $\mathbf{G}$ o $\mathbf{S}$ se vuelven vacíos $\Rightarrow$ no hay en el lenguaje usado hipótesis consistentes con los datos
  - Si se acaban los ejemplos sin llegar a estos casos $\Rightarrow$ tenemos una disyunción de posibles hipótesis $\Rightarrow$ Combinar sus predicciones (votación, ponderación...)

- **Problemas:**
  - Calcular relación *más general* y definir los operadores de especialización/generalización
    - Coste computacional depende de la expresividad del lenguaje (coste transformaciones)
  - Sensible a presencia de ruido (ejemplos incorrectos) $\Rightarrow$ el ruido hará que se seleccionen hipótesis incorrectas
  - Sensible a carencia de número de ejemplos $\Rightarrow$ espacio de versiones se colapsará (vacío)
  - Si no se restringe la representación de $\mathbf{S}$ y $\mathbf{G}$ $\Rightarrow$ degenera y no generaliza
    - $\mathbf{S}$: disyunción de ejemplos positivos
    - $\mathbf{G}$: negación de disyunción de ejemplos negativos
- **Ventajas:**
  - Modelo de algoritmo general para aprendizaje inductivo
  - Aplicable en problemas "adecuados"
  - Sirve de punto de partida general para otros métodos más específicos/eficientes