

Inläring av Regler

- 1 Regler
 - Predikat
 - Horn-satser
- 2 Inläring av Predikat
 - Beslutsträd
 - Sekvensiell täckning
- 3 Induktiv Logikprogrammering
 - Top-down
 - Bottom-up

- 1 Regler
 - Predikat
 - Horn-satser
- 2 Inläring av Predikat
 - Beslutsträd
 - Sekvensiell täckning
- 3 Induktiv Logikprogrammering
 - Top-down
 - Bottom-up

Predikat

Predikat — Logiska uttryck

Predikat

Predikat — Logiska uttryck

Beroende av värdet av ett antal attribut

Predikat

Predikat — Logiska uttryck

Beroende av värdet av ett antal attribut

$\text{Regnigt} \wedge \text{Kallt}$

Horn-satser

Horn-satser — Första ordningens logik

Horn-satser

Horn-satser — Första ordningens logik

Kan innehålla **variabler**

Horn-satser

Horn-satser — Första ordningens logik

Kan innehålla **variabler**

$$\text{syskon}(x, y) : \text{förälder}(z, x) \wedge \text{förälder}(z, y)$$

Horn-satser

Horn-satser — Första ordningens logik

Kan innehålla **variabler**

$\text{syskon}(x, y) : \text{förälder}(z, x) \wedge \text{förälder}(z, y)$

Betydligt mer uttrycks kraft än predikatlogik

Grunden för t.ex. *Prolog*

Varför vill man lära regler?

Varför vill man lära regler?

- Kan översättas till ”vanligt språk”

Varför vill man lära regler?

- Kan översättas till ”vanligt språk”
- Förklarande

Varför vill man lära regler?

- Kan översättas till ”vanligt språk”
- Förklarande

Nackdelar?

Varför vill man lära regler?

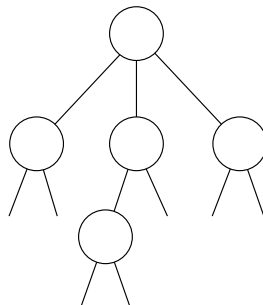
- Kan översättas till ”vanligt språk”
- Förklarande

Nackdelar?

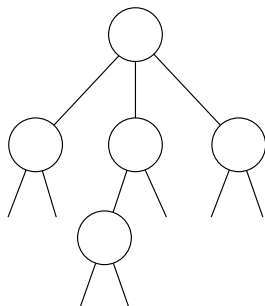
- Svårt att hantera brus och osäkerhet

- 1 Regler
 - Predikat
 - Horn-satser
- 2 Inläring av Predikat
 - Beslutsträd
 - Sekvensiell täckning
- 3 Induktiv Logikprogrammering
 - Top-down
 - Bottom-up

Beslutsträd



Beslutsträd



Disjunktion av konjunktioner

Sekvensiell täckning

(Sequential Covering)

Sekvensiell täckning

(Sequential Covering)

Idén bakom *Sekvensiell täckning*

Försök inte förklara allting på en gång!

Sekvensiell täckning

(Sequential Covering)

Idén bakom *Sekvensiell täckning*

Försök inte förklara allting på en gång!

- 1 Konstruera **en** regel

Sekvensiell täckning

(Sequential Covering)

Idén bakom *Sekvensiell täckning*

Försök inte förklara allting på en gång!

- 1 Konstruera **en** regel som
 - Matchar många positiva exempel
 - Inte matchar några negativa exempel

Sekvensiell täckning

(Sequential Covering)

Idén bakom *Sekvensiell täckning*

Försök inte förklara allting på en gång!

- 1 Konstruera **en** regel som
 - Matchar många positiva exempel
 - Inte matchar några negativa exempel
- 2 Tag bort de positiva exempel som matchades

Sekvensiell täckning

(Sequential Covering)

Idén bakom *Sekvensiell täckning*

Försök inte förklara allting på en gång!

- 1 Konstruera **en** regel som
 - Matchar många positiva exempel
 - Inte matchar några negativa exempel
- 2 Tag bort de positiva exempel som matchades
- 3 Upprepa tills alla positiva exempel matchats

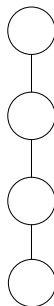
Hur konstruerar man en bra regel?

Hur konstruerar man en bra regel?

Girig sökning — Välj succesivt attribut som matchar så många positiva exempel som möjligt

Hur konstruerar man en bra regel?

Girig sökning — Välj succesivt attribut som matchar så många positiva exempel som möjligt



Beam search

Beam search

Alternativ till girig sökning

Beam search

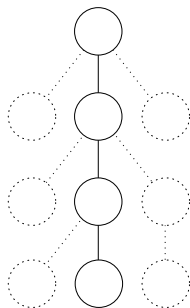
Alternativ till girig sökning

Flera alternativa sökvägar hålls aktuella

Beam search

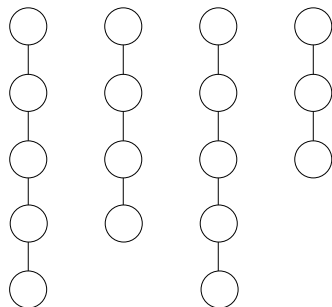
Alternativ till girig sökning

Flera alternativa sökvägar hålls aktuella



Resultaten från sökningen

Resultaten från sökningen: flera regler



- 1 Regler
 - Predikat
 - Horn-satser
- 2 Inlärnning av Predikat
 - Beslutsträd
 - Sekvensiell täckning
- 3 Induktiv Logikprogrammering
 - Top-down
 - Bottom-up

Induktiv Logikprogrammering

ILP — Inductive Logic Programming

Induktiv Logikprogrammering

ILP — Inductive Logic Programming

Automatisk konstruktion av Horn-satser från tränings exempel

Induktiv Logikprogrammering

ILP — Inductive Logic Programming

Automatisk konstruktion av Horn-satser från tränings exempel

Horn-satser

- Uttrycksfullare än predikat
- Variabler i beskrivningarna

Induktiv Logikprogrammering

ILP — Inductive Logic Programming

Automatisk konstruktion av Horn-satser från tränings exempel

Horn-satser

- Uttrycksfullare än predikat
- Variabler i beskrivningarna

Exempel

$$\begin{aligned} \text{farfar}(x, y) : & \text{ förälder}(x, z) \\ & \wedge \text{ man}(x) \\ & \wedge \text{ förälder}(z, y) \\ & \wedge \text{ man}(z) \end{aligned}$$

Induktiv Logikprogrammering

Hur kan ett program automatiskt hitta regeln

$\text{farfar}(x, y) : \text{förälder}(x, z) \wedge \text{man}(x) \wedge \text{förälder}(z, y) \wedge \text{man}(z)$

från exempel av typen

$\text{farfar}(\textit{Sven}, \textit{Pär}) \mapsto \mathbf{False}$

$\text{farfar}(\textit{Pär}, \textit{Sven}) \mapsto \mathbf{True}$

$\text{farfar}(\textit{Lisa}, \textit{Sven}) \mapsto \mathbf{False}$

- Top-down

- Bottom-up

- Top-down
Utnyttja sekvensiell täckning
Starta från en generell regel och specialisera
- Bottom-up

- **Top-down**
Utnyttja sekvensiell täckning
Starta från en generell regel och specialisera
- **Bottom-up**
Starta från ett exempel och konstruera förklarande regler

Top-down teknik för att hitta **en regel**

Top-down teknik för att hitta **en regel**

- Stara generellt

Top-down teknik för att hitta **en regel**

- Stara generellt

$\text{farfar}(x, y) : \mathbf{True}$

Top-down teknik för att hitta **en regel**

- Stara generellt

$\text{farfar}(x, y) : \mathbf{True}$

- Generera möjliga specialiseringar

Top-down teknik för att hitta **en regel**

- Stara generellt

$\text{farfar}(x, y) : \mathbf{True}$

- Generera möjliga specialiseringar

$\text{farfar}(x, y) : \text{förälder}(x, y)$

$\text{farfar}(x, y) : \text{förälder}(x, z)$

$\text{farfar}(x, y) : \text{kvinna}(x)$

$\text{farfar}(x, y) : \text{man}(x)$

Top-down teknik för att hitta **en regel**

- Stara generellt

$\text{farfar}(x, y) : \mathbf{True}$

- Generera möjliga specialiseringar

$\text{farfar}(x, y) : \text{förälder}(x, y)$

$\text{farfar}(x, y) : \text{förälder}(x, z)$

$\text{farfar}(x, y) : \text{kvinna}(x)$

$\text{farfar}(x, y) : \text{man}(x)$

- Behåll den som bäst beskriver data

Top-down teknik för att hitta **en regel**

- Stara generellt

$\text{farfar}(x, y) : \mathbf{True}$

- Generera möjliga specialiseringar

$\text{farfar}(x, y) : \text{förälder}(x, y)$

$\text{farfar}(x, y) : \text{förälder}(x, z)$

$\text{farfar}(x, y) : \text{kvinna}(x)$

$\text{farfar}(x, y) : \text{man}(x)$

- Behåll den som bäst beskriver data
- Om regeln matchar negativa exempel **specialiseras den**

Hur specialiserar man en regel?

Hur specialiserar man en regel?

- Lägg på en konjunktionsdel

Hur specialiserar man en regel?

- Lägg på en konjunktionsdel

$$\text{farfar}(x, y) : \text{förälder}(x, z) \wedge \text{man}(x)$$
$$\text{farfar}(x, y) : \text{förälder}(x, z) \wedge \text{man}(z)$$

Hur specialiserar man en regel?

- Lägg på en konjunktionsdel

$$\text{farfar}(x, y) : \text{förälder}(x, z) \wedge \text{man}(x)$$
$$\text{farfar}(x, y) : \text{förälder}(x, z) \wedge \text{man}(z)$$

- Välj den bästa

Hur specialiserar man en regel?

- Lägg på en konjunktionsdel

$$\text{farfar}(x, y) : \text{förälder}(x, z) \wedge \text{man}(x)$$
$$\text{farfar}(x, y) : \text{förälder}(x, z) \wedge \text{man}(z)$$

- Välj den bästa

Detta upprepas tills inga negativa exempel matchas

Hur konstruerar man de nya faktorerna i regeln?

Hur konstruerar man de nya faktorerna i regeln?

- Prova **alla kända regler** med "gamla" variabler

Hur konstruerar man de nya faktorerna i regeln?

- Prova **alla kända regler** med ”gamla” variabler

farfar(x, y) : förälder(x, y)
 förälder(y, x)
 man(x)
 man(y)

Hur konstruerar man de nya faktorerna i regeln?

- Prova **alla kända regler** med "gamla" variabler
- Sätt in **nya variabler**, utom på en plats

Hur konstruerar man de nya faktorerna i regeln?

- Prova **alla kända regler** med ”gamla” variabler
- Sätt in **nya variabler**, utom på en plats

farfar(x, y) : förälder(x, z)
 förälder(z, x)
 förälder(y, z)
 förälder(z, y)

Hur konstruerar man de nya faktorerna i regeln?

- Prova **alla kända regler** med ”gamla” variabler
- Sätt in **nya variabler**, utom på en plats
- Använd **likhet** och olikhet

Hur konstruerar man de nya faktorerna i regeln?

- Prova **alla kända regler** med ”gamla” variabler
- Sätt in **nya variabler**, utom på en plats
- Använd **likhet** och olikhet

farfar(x, y) : $x = y$

$x \neq y$

$x < y$

$x > y$

Hur konstruerar man de nya faktorerna i regeln?

- Prova **alla kända regler** med ”gamla” variabler
- Sätt in **nya variabler**, utom på en plats
- Använd **likhet** och olikhet
- Använd **typinformation** för beskärning

Bottom-up teknik

Bottom-up teknik

Inverse Resolution

Bottom-up teknik

Inverse Resolution

Använd metoder för automatisk härledning ”baklänges”

Bottom-up teknik

Inverse Resolution

Använd metoder för automatisk härledning ”baklänges”

- Starta från **ett exempel**

Bottom-up teknik

Inverse Resolution

Använd metoder för automatisk härledning ”baklänges”

- Starta från **ett exempel**
- Bygg en förklarande regel

Bottom-up teknik

Inverse Resolution

Använd metoder för automatisk härledning ”baklänges”

- Starta från **ett exempel**
- Bygg en förklarande regel
- Tag bort förklarade data

Bottom-up teknik

Inverse Resolution

Använd metoder för automatisk härledning ”baklänges”

- Starta från **ett exempel**
- Bygg en förklarande regel
- Tag bort förklarade data
- Upprepa