

Maskininläring

2D1431

Örjan Ekeberg

Okt–Dec, 2006

- 1 Vad är maskininläring?
 - Definition av lärande
 - Tillämpningar
- 2 Praktisk information om kursen
 - Kursregistrering
 - Examination
 - Kursinnehåll
 - Laborationer
- 3 Exempel
 - Ett tänkt projekt

- 1 Vad är maskininläring?
 - Definition av lärande
 - Tillämpningar
- 2 Praktisk information om kursen
 - Kursregistrering
 - Examination
 - Kursinnehåll
 - Laborationer
- 3 Exempel
 - Ett tänkt projekt

Definition av lärande program

Centrala komponenter:

- Programmets *uppgift*, T
- Förbättring enligt ett *prestandamått*, P
- Utnyttjar *erfarenheten*, E

Definition av lärande program

Centrala komponenter:

- Programmets *uppgift*, T
- Förbättring enligt ett *prestandamått*, P
- Utnyttjar *erfarenheten*, E

Spela schack

- T : spela enligt reglerna
- P : antal vunna matcher
- E : träna mot sig själv

Definition av lärande program

Centrala komponenter:

- Programmets *uppgift*, T
- Förbättring enligt ett *prestandamått*, P
- Utnyttjar *erfarenheten*, E

Spela schack

- T : spela enligt reglerna
- P : antal vunna matcher
- E : träna mot sig själv

Läsa handskriven text

- T : tolka ord givna som bilder
- P : andel korrekt tolkade ord
- E : databas med tolkad skrift

Tillämpningar

Exempel på tillämpningar

- Taligenkänning
- Automatisk bilkörning
- Spel: Backgammon
- Autonoma robotar
- Spam-filter för e-post

Tillämpningar

Exempel på tillämpningar

- Taligenkänning
- Automatisk bilkörning
- Spel: Backgammon
- Autonoma robotar
- Spam-filter för e-post

Inläringens roll

Data mining Göra kunskap av data

Svårspecificerade uppgifter Robotik, tal, seende, spel

Adaptiva program Individanpassade program

- 1 Vad är maskininlärning?
 - Definition av lärande
 - Tillämpningar
- 2 Praktisk information om kursen
 - Kursregistrering
 - Examination
 - Kursinnehåll
 - Laborationer
- 3 Exempel
 - Ett tänkt projekt

Kursregistrering

- Centralt: via studievägledningen
- Lokalt: res checkin mi06

Kursinformation

<http://www.csc.kth.se/2d1431/mi06>

Examination

Obligatoriska moment i kursen

- Tentamen
- Labbar, 4 st

Examination

Obligatoriska moment i kursen

- Tentamen
- Labbar, 4 st

Bonussystem

- Lab redovisad i tid ger ett poäng till tentan.
- Max bonuspoäng=4 (6 motsvarar ett betygssteg på tentan).
- Bonuspoängen kan endast användas för betyghöjning, inte för att komma över godkänthöjningen.
- Bonuspoängen kan bara tillgodoräknas på ordinarie tentan och första omtentan.

Kursinnehåll

- Begreppsinlärning
- Beslutsträd
- Artificiella neuronnät
- Statistiska tekniker
- Exempelbaserade metoder
- Belöningsbaserad inlärning
- Genetiska algoritmer
- Lärbarhetsteori
- Inlärning av regler

Laborationer

- 1 Begreppsinlärning
- 2 Statistisk klassificering och Boosting
- 3 Belöningsbaserad inlärning
- 4 Genetiska algoritmer

Obs. Schemalagda labbtider är **redovisningstider**. Läs anvisningen och gör uppgifterna i god tid innan.

- 1 Vad är maskininlärning?
 - Definition av lärande
 - Tillämpningar
- 2 Praktisk information om kursen
 - Kursregistrering
 - Examination
 - Kursinnehåll
 - Laborationer
- 3 Exempel
 - Ett tänkt projekt

Tänkt projekt:

Ett program som lär sig spela Othello

Tänkt projekt:

Ett program som lär sig spela Othello

Vi måste specificera

- Uppgiften T
- Prestandamåttet P
- Erfarenheten E

Tänkt projekt:

Ett program som lär sig spela Othello

Vi måste specificera

- Uppgiften T
Välj drag enligt Othellos regler
- Prestandamåttet P
- Erfarenheten E

Tänkt projekt:

Ett program som lär sig spela Othello

Vi måste specificera

- Uppgiften T
Välj drag enligt Othellos regler
- Prestandamåttet P
Andel vunna matcher mot bra spelare
- Erfarenheten E

Tänkt projekt:

Ett program som lär sig spela Othello

Vi måste specificera

- Uppgiften T
Välj drag enligt Othellos regler
- Prestandamåttet P
Andel vunna matcher mot bra spelare
- Erfarenheten E
Spela mot sig själv och andra

- Varifrån kommer träningsexemplen?
- Vad ska göras, mer precist?
- Hur ska detta representeras internt?
- Vilken inlärningsalgoritm ska användas?

Träningsexempel

- Lärare eller erfarenhet?
- Direkt eller indirekt?
Talar någon om för dig vad som är bra drag?
- Credit assignment
Bra drag eller långsiktig strategi?
- Är träningsexemplen representativa?

Vad ska programmet lära sig?

Vad ska programmet lära sig?

$$P : b \mapsto d$$

b: en given ställning

d: vårt nästa drag

Vad ska programmet lära sig?

$$P : b \mapsto d$$

b: en given ställning

d: vårt nästa drag

Svårt!

Vad ska programmet lära sig?

$$P : b \mapsto d$$

b : en given ställning

d : vårt nästa drag

Svårt!

$$V : b \mapsto \mathcal{R}$$

b : ställningen efter vårt drag

\mathcal{R} : "värdet" på ställningen

Rekursiv definition av “värdet”

$V(b) = 100$ för en vinnande slutställning

$V(b) = -100$ för en förlorande slutställning

Rekursiv definition av “värdet”

$V(b) = 100$ för en vinnande slutställning

$V(b) = -100$ för en förlorande slutställning

$V(b) = V(b')$ där b' är den slutställning man når när man spelar optimalt från b .

Rekursiv definition av “värdet”

$V(b) = 100$ för en vinnande slutställning

$V(b) = -100$ för en förlorande slutställning

$V(b) = V(b')$ där b' är den slutställning man når när man spelar optimalt från b .

Inte rimlig att beräkna!

Dramatisk approximation:

$$V(b) \approx V(\text{succ}(b))$$

$\text{succ}(b)$: den ställning man kom till efter b

Dramatisk approximation:

$$V(b) \approx V(\text{succ}(b))$$

$\text{succ}(b)$: den ställning man kom till efter b

Mer approximation:

$$V_{\text{train}}(b) \leftarrow \hat{V}(\text{succ}(b))$$

$\hat{V}(\cdot)$: Vår skattning av funktionen $V(\cdot)$

$V_{\text{train}}(b)$: Nytt träningsexempel för att skatta $V(b)$

Dramatisk approximation:

$$V(b) \approx V(\text{succ}(b))$$

$\text{succ}(b)$: den ställning man kom till efter b

Mer approximation:

$$V_{\text{train}}(b) \leftarrow \hat{V}(\text{succ}(b))$$

$\hat{V}(\cdot)$: Vår skattning av funktionen $V(\cdot)$

$V_{\text{train}}(b)$: Nytt träningsexempel för att skatta $V(b)$

Under rätt förutsättningar kommer \hat{V} att konvergera mot V .

Hur ska \hat{V} representeras?

- Regler?
- Neuronnät?
- Valda features?

Hur ska \hat{V} representeras?

- Regler?
- Neuronnät?
- Valda features?

Ett rimligt förslag: *viktade features*

$$\hat{V}(b) = w_0 + w_1 a_1 + w_2 a_2 + w_3 a_3 + \dots$$

a_1 : Antal vita brickor

a_2 : Antal svarta brickor

a_3 : Antal vita hörnbrickor

a_4 : Antal svarta hörnbrickor

o.s.v.

Inlärning \equiv bestäm $w_i, \forall i$

Hur beräknas w_i från träningsexempel?

Hur beräknas w_i från träningsexempel?

LMS-regeln (*Least Mean Square*)

- 1 Beräkna felet vid en skattning

$$\delta(b) = V_{\text{train}}(b) - \hat{V}(b)$$

- 2 Ändra vikterna så att felet krymper

$$w_i \leftarrow w_i + \eta \delta(b) \cdot a_i$$

Kommer detta att fungera i praktiken?

Kommer detta att fungera i praktiken?
Troligen, men med begränsade prestanda.

Kommer detta att fungera i praktiken?
Troligen, men med begränsade prestanda.

Vad bör förbättras?

Kommer detta att fungera i praktiken?
Troligen, men med begränsade prestanda.

Vad bör förbättras?
I första hand representationen av värdefunktionen.