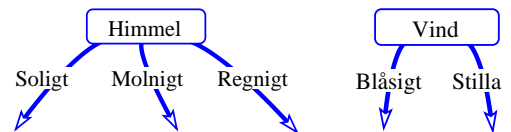


# Beslutsträd

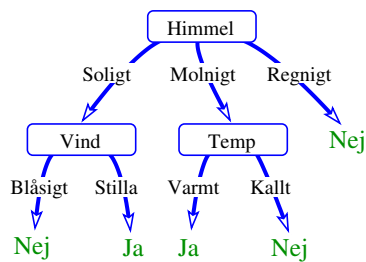
- 1 Beslutsträd
  - Användning
  - Inläring
- 2 Oförutsägbarhet
  - Entropimåttet
  - Entropi för datamängder
  - Information Gain
- 3 Bias
  - Bias
  - Occam's princip
  - Överträning
- 4 Förbättringar

- 1 Beslutsträd
  - Användning
  - Inläring
- 2 Oförutsägbarhet
  - Entropimåttet
  - Entropi för datamängder
  - Information Gain
- 3 Bias
  - Bias
  - Occam's princip
  - Överträning
- 4 Förbättringar

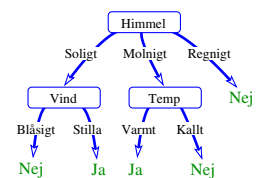
Grundidé: Testa ett attribut i taget



Hela analysstrategin kan betraktas som ett träd.



Svaren (kategoriseringen) beskrivs av löven i trädet



Vad representerar trädet?

$$(Soligt \wedge Stilla) \vee (Molnigt \wedge Varmt)$$

Fungerar som en *disjunktion av konjunktioner*

Normalform för boolska funktioner

Godtyckliga kategoriseringar kan göras!

Hur kan man bygga träden automatiskt?

- 1 Välj ett attribut att fråga om
- 2 Grenar som med entydig klassning är klara
- 3 Andra grenar byggs vidare rekursivt

Central fråga: Hur väljer vi attribut?

Girig idé:

Välj i varje läge det attribut som *säger mest* om svaret

- 1 Beslutsträd
  - Användning
  - Inläring
- 2 Oförutsägbarhet
  - Entropimättet
  - Entropi för datamängder
  - Information Gain
- 3 Bias
  - Bias
  - Occam's princip
  - Överträning
- 4 Förbättringar

## Entropi

Entropi — mått på **oförutsägbarheten**

$$\text{Entropi} = \sum_i -p_i \log_2 p_i$$

$p_i$  sannolikheten för händelsen  $i$

## Entropi

Exempel: singla slant

$$p_{\text{krona}} = 0.5; \quad p_{\text{klave}} = 0.5$$

$$\begin{aligned} \text{Entropin} &= \sum_i -p_i \log_2 p_i = \\ &= -0.5 \log_2 0.5 + -0.5 \log_2 0.5 = -0.5 \underbrace{\log_2 0.5}_{-1} + -0.5 \underbrace{\log_2 0.5}_{-1} \\ &= 1 \end{aligned}$$

Utfallet av en slantsingling innehåller **1 bit** information

## Entropi

Exempel: kasta tärning

$$p_1 = \frac{1}{6}; \quad p_2 = \frac{1}{6}; \dots \quad p_6 = \frac{1}{6}$$

$$\begin{aligned} \text{Entropin} &= \sum_i -p_i \log_2 p_i = \\ &= 6 \times -\frac{1}{6} \log_2 \frac{1}{6} = \\ &= -\log_2 \frac{1}{6} = \log_2 6 \approx 2.58 \end{aligned}$$

Utfallet av ett tärningskast innehåller **2.58 bit** information

## Entropi

Exempel: kasta en **falsk tärning**

$$p_1 = 0.1; \dots \quad p_5 = 0.1; \quad p_6 = 0.5$$

$$\begin{aligned} \text{Entropin} &= \sum_i -p_i \log_2 p_i = \\ &= -5 \cdot 0.1 \log_2 0.1 - 0.5 \log_2 0.5 = \\ &\approx 2.16 \end{aligned}$$

En riktig tärning är **mer oförutsägbar** (2.58 bit) än en falsk (2.16 bit)



- 1 Beslutsträd
  - Användning
  - Inläring
- 2 Oförutsägbarhet
  - Entropimåttet
  - Entropi för datamängder
  - Information Gain
- 3 Bias
  - Bias
  - Occam's princip
  - Överträning
- 4 Förbättringar

Vilken Bias har denna inlärningsalgoritm?

- **Restriction Bias?**  
Nej, alla hypoteser kan erhållas
- **Preference Bias?**  
Ja, vissa typer av träd hittas före andra

Vilka hypoteser (här: träd) prioriteras?

- Grunda träd
- "Viktiga frågor" tidigt

Hur ska man veta vilka hypoteser som ska föredras när flera stämmer med exemplen?

**Occam's princip** (*Occam's razor*, "Occam's rakkniv")

William från Ockham, Teolog och Filosof (1285–1349)

*"Entia non sunt multiplicanda praeter necessitatem"*

fritt översatt:

"Man bör inte anta fler företeelser än vad som är nödvändigt för att förklara fenomenen"

Om fler hypoteser kan förklara data,  
välj då den enklaste

Varför är enkla hypoteser bättre?

Troligare att verkligheten som genererat exemplen har en enkel genererande mekanism.

Enkla hypoteser generaliserar normalt bättre.

**Överträning, overfitting**

När hypoteserna är för specialiserade för de aktuella träningsexemplen.

Bra på träningsdata men generaliserar dåligt

När inträffar detta?

- Icke-representativt sample
- Brus bland exemplen

Vad kan man göra åt det?

Välj en enklare hypotes och acceptera fel även för träningsexemplen

Möjliga förbättringar av beslutsträden

- Undvik överträning
  - Begränsa trädets höjd
  - Beskärning (*Pruning*)
- Attribut med graderade värden
- Saknade attributvärden
- Olika kostnad för olika attribut