



Mesterséges
Intelligencia

Csató Lehel

Mesterséges Intelligencia

Csató Lehel

Matematika-Informatika Tanszék
Babeș–Bolyai Tudományegyetem, Kolozsvár

2007/2008



Az Előadások Témái

Mesterséges
Intelligencia

9

Csató Lehel

Tanulás

Indukció

Cross-validation

Döntési Fák

Netlab

- Bevezető: mi a *mesterséges* intelligencia ...
- „Tudás”–reprezentáció
- Gráfkeresési stratégiák
- Szemantikus hálók / Keretrendszerek
- Játékok modellezése
- Bizonytalanság kezelése
- Grafikus modellek
- **Tanuló rendszerek**
- Szimulált kifűtés, Genetikus algoritmusok
- Neurális hálók
- Gépi tanulás
- Nemparametrikus módszerek



Mesterséges
Intelligencia

9

Csató Lehel

Tanulás

Indukció

Cross-validation

Döntési Fák

Netlab

Előadások honlapja

<http://www.cs.ubbcluj.ro/~csatol/mestint>

Vizsga

Szóbeli (60%) + Gyakorlat (40%)
(v) Előadás (60%)

Laborgyakorlatok:

- | | | |
|---|---|----------|
| 1 | Clean vagy Prolog - dedukciós algoritmus | 30% |
| 2 | C / C++ / C# / ... - genetikus algoritmus | 10% vagy |
| 3 | Matlab - Neurális hálózatok vagy SVM | 10% |

A gépi fordítás – annak ellenére, hogy régóta vizsgált tudományág – nem versenyezhet a tolmácsokkal. Még akkor sem, ha az egyértelmű szókészlettel rendelkező, kifejezetten formális szövegeket produkáló területeken – például a repülőgép-gyártásban – figyelemreméltó eredményeket érnek el.

Korpusz

- Nagymennyiségű strukturált, elektronikusan tárolt szövegbázis;
- Egy adott nyelvet reprezentál, akár több tízmillió szóból is állhat;
- Általában alterületen használják: statisztikai elemzésekre, illetve nyelvi törvények érvényességének vizsgálatára.
- A szövegek lehetnek egyetlen (monolingual) vagy két (több) nyelvűek (multilingual). A rendszer így tanulja meg, hogy egy nyelv szavai és kifejezései miként kapcsolódnak egy másikhoz.

Bebizonyosodott, hogy **METISII** a hosszú évtizedek fejlesztésének eredményeként létrejött piacvezető – szabályalapú – rendszerrel is képes felvenni a versenyt.





„Tanuló” rendszerek

pp. 525 – Russell & Norvig, 1995

Agents that can improve their behaviour through diligent study of their own experiences.

Mesterséges
Intelligencia

9

Csató Lehel

Tanulás

Indukció

Cross-validation

Döntési Fák

Netlab

- Algoritmus változtatható állapottérrel, mely működése során az újabb – ugyanolyan típusú – feladatokat **jobban** oldja meg.
- Példák: genetikus algoritmus, neurális hálók, stb.
- „Machine Learning” – „Gépi tanulás”.
- Induktív rendszerek – inferencia.



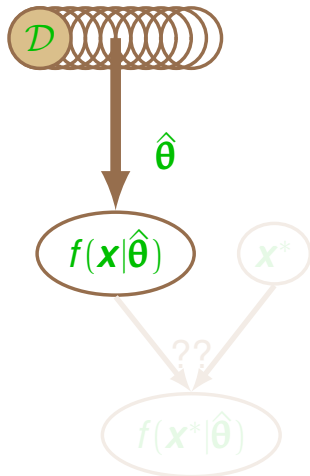
Induktív megoldás

- Modell feltételezése:
 $f(\mathbf{z}, \boldsymbol{\theta})$,
- Adatok halmaza:
 $\mathcal{D} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$
- Illesztő – hiba –
függvény: $L(f(\mathbf{z}))$
- „optimális modell”: $\boldsymbol{\theta}^*$
- **Predikció:** $f(\mathbf{z}, \boldsymbol{\theta}^*)$.

Deduktív megoldás

- Nincs modell,
- Adatok:
 $\mathcal{D} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$
- Illesztési művelet egy \mathbf{z}
új mintára,
- használunk minden
adatot.

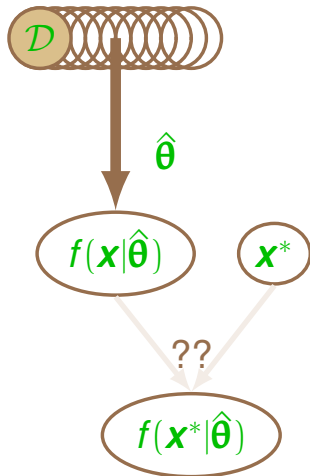
Induktív megoldás



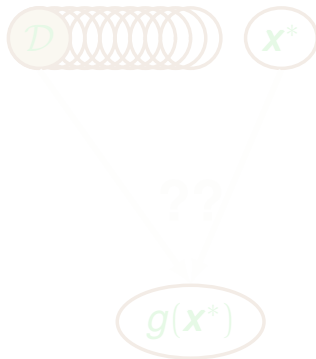
Deduktív megoldás



Induktív megoldás

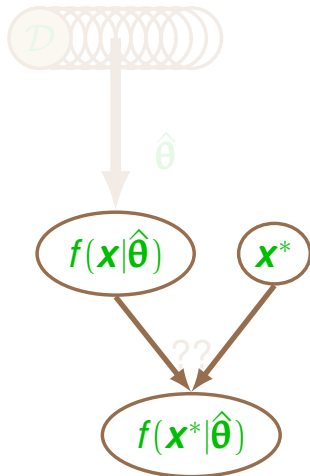


Deduktív megoldás

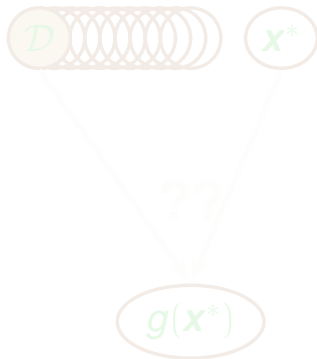




Induktív megoldás



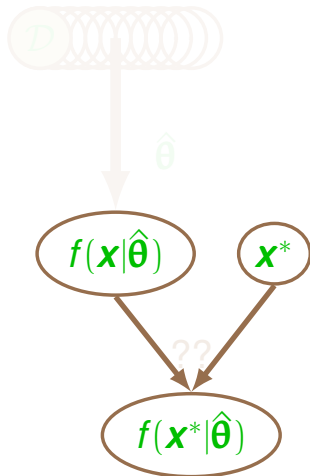
Deduktív megoldás



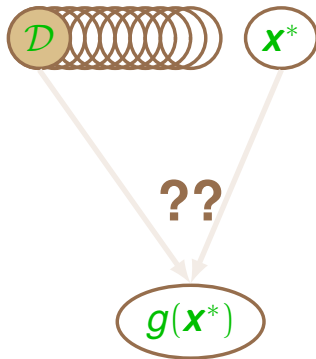
„Levezetjük” a g függvény
 x^* -hoz tartozó értékét.



Induktív megoldás

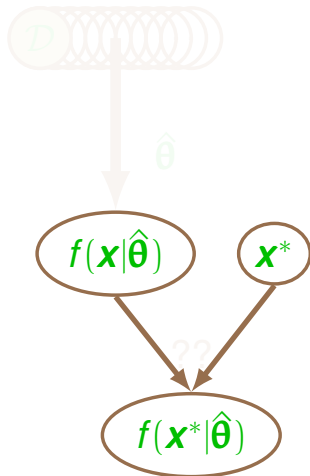


Deduktív megoldás

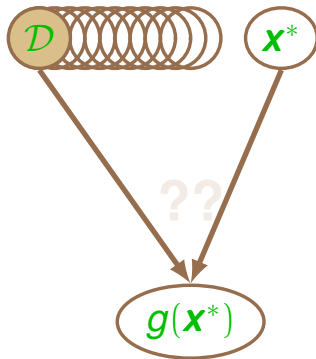


„Levezetjük” a g függvény
 x^* -hoz tartozó értékét.

Induktív megoldás



Deduktív megoldás



„Levezetjük” a g függvény
 x^* -hoz tartozó értékét.



„Becslés” egy modell ismeretében.

- Modell-osztály;
- Modellek közötti preferencia $\Rightarrow \hat{\theta}$;
- Predikció a modell alapján;
- PI: lineáris becslések, neurális hálók;
- Döntési/regressziós fák;
- Rejtett változós modellek

- A cél **egy ismert** adatra egy választ fogalmazni;
- A válasz: kategoriális, folytonos ...
- K-nn: **K** legközelebbi szomszéd alapján történő közelítés;
- Rezolúció: egy ismert predikátumról eldönteni, hogy igaz vagy hamis.

A továbbiakban **Induktív módszerekkel** foglalkozunk.



adatokból „információt” vonnak ki;⁶

Inductive learning methods

Systematically produce intensional concept descriptions from extensional concept descriptions.

I.e, **from the specific knowledge** provided by domain examples, ~ obtain **general domain knowledge**.

Információ induktív rendszereknél

Egy indukció folyamán az adatokból „nyert” információt a modell paraméterei tárolják.

Egy $\mathcal{F} \stackrel{\text{def}}{=} \{f(\mathbf{x}|\boldsymbol{\theta}), \boldsymbol{\theta} \in \Omega\}$ modell esetén a \mathcal{D} adatokat az optimális $\hat{\boldsymbol{\theta}}$ paraméter helyettesíti.

⁶Hogyan definiáljuk az információ fogalmát?



„Információ” - $\hat{\theta}$:

- egy $f(\mathbf{x}|\theta)$ függvényt keresünk.
- $\hat{\theta}$ az optimális függvény „koordinátája”.
- **Nem ismerjük az adatokat generáló függvényt**

Melyik paraméter jobb?

Tesztelési módszer:

- az adatokat kettéosztjuk: tanulási- illetve **teszt**-adathalmaz;
- az optimális $\hat{\theta}$ meghatározásához **csak** a tanuló-adatokat használjuk;
- Tesztelés: **hiba mérése a teszt-adathalmazon.**



Cross-validation:

- módszer, mely méri a tanulási folyamat eredményességét;
- olyan esetekben használatos, ahol nincs modell illetve külön teszt-adat;

Módszer:

- a **teljes** adat felosztása: K részre;
- tanulási-, illetve teszt-adatok **definiálása**:
 - j -dik rész a **teszt-adat**;
 - $\{1..K\} \setminus j$ tanulási adatok.
- minden j -re mérjük a teszt-hibákat. (összegezzük / átlagoljuk)



Cross-validation

Előnye:

- nem függ a hibafüggvénytől illetve az adatok típusától (diszkrét, folytonos, strukturált);
- könnyen kódolható.

Hátránya:

- Nagy adathalmazra sokáig fut;
- **Nem alkalmazható paraméterek becslésére.**



Döntési/Regressziós fák

Mesterséges
Intelligencia

9

Csató Lehel

Tanulás

Indukció

Cross-validation

Döntési Fák

Netlab

Regresszió:

$$f : \Omega \rightarrow \mathbb{R}$$

Osztályozás:

$$f : \Omega \rightarrow \{\text{Igen, Nem}\}$$

Definíció:

- fa formájában tárol egy függvényt;
- leszarmazottak nélküli csúcs - levél; másképp belső csúcs;
- minden **belső csúcs**hoz van egy predikátum rendelve; Például:
 - $\text{Kor} < 20$,
 - $\text{Fogl} \in \{\text{Diak}, \text{Tanar}\}$.

- szabályok megállapítására szolgál;
- általánosítások – extrapolálás;
- Nem fontos az adatok diszkrét/folytonos léte;



- sorrend-függő az eredmény;
- szomszédság/topológia meghatározása: modellezési kérdés.
- Sok adat \Rightarrow **nagy fa**, redundáns modellezés.

Logikai bemenő adatok esetén:

- Bináris fa természetes;

Folytonos bemenő adatok esetén:

- El kell dönteni egy-egy ág érvényességét; \Leftrightarrow
- Diszkrétizálni kell a teret; \Leftrightarrow
- Modellezési feladat



Döntési fa építése

Mesterséges
Intelligencia

9

Csató Lehel

Tanulás

Indukció

Cross-validation

Döntési Fák

Netlab

Döntési fa építése

Rekurzívan:

- Vizsgáljuk meg az adatokat és állapítsuk meg a **legjobb vágást**;
- Osszuk ketté az adatokat a vágás szerint;

? Kérdések ?

- Megállás;
- Legjobb vágás keresése;

Vágások minősége: CART (Classification and Regression Trees) algoritmus.



*Ian T. Nabney: NETLAB
Algorithms for Pattern Recognition
Springer 2002*

<http://www.ncrg.aston.ac.uk/netlab>

Matlab függvény-gyűjtemény

- Ingyenesen letölthető; egyszerűen kezelhető;
- **Nagyon** sok modell és algoritmus tesztelhető –
anélkül, hogy a modelleket implementálni kellene;

Implementált modellek és algoritmusok: (módosítható demó-k)

- Regresszió
- Osztályozás
- Klaszterezés
- Bayes-módszerek
- Optimalizálás
- Mintavételezés



Netlab használata:

- Installálás: letöltés és: `addpath('_netlab_path_')`
- Demók indítása: `demnlab`
- Programozás a demók módosításával, pl. `demmlp1`

Példa:

%data generation.

`ndata = 20;`

`noise = 0.2;`

`x = {0:1/(ndata - 1):1}';`

`t = sin(2*pi*x) + noise*randn(ndata, 1);`

% Set up network parameters.

`nin = 1; % Number of inputs.`

`nhidden = 3; % Number of hidden units.`

`nout = 1; % Number of outputs.`

`alpha = 0.01; % weight-decay prior.`

% Initialisation

`net = mlp(nin, nhidden, nout, 'linear', alpha);`

% Option vector intialisation

`options = zeros(1,18);`

`options(1) = 1; % display error values.`

`options(14) = 100; % Number of training cycles.`

% Train using scaled conjugate gradients.

`{net, options} = netopt(net, options, x, t, 'scg');`

% Plot for TEST data

`plotvals = {0:0.01:1}';`

`y = mlpfwd(net, plotvals);`

A **Netlab** programmal egyszerűen megoldható a laborfeladat.