Mesterséges Intelligencia

Csató Lehel

Matematika-Informatika Tanszék
Babeş–Bolyai Tudományegyetem, Kolozsvár

2010/2011

Az Előadások Témái

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

Admin ... ... trívia

`http://www.cs.ubbcluj.ro/~csatol/mestint`

Vizsga

Szóbeli (60%) + Gyakorlat (40%)

Laborgyakorlatok:

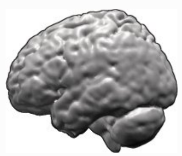
1. Gráfok ábrázolása - dedukciós algoritmus **18%**
2. Játékelmélet **10%**
3. Matlab - tanulási algoritmus **12%**
4. Opcionális feladatok - max. 3/személy **sok%**

Bemutatók (5–20 pont)

Alkalmazások bemutatása, melyek adaptív, gépi tanulásos vagy *mestint* módszereket alkalmaznak.

A „mesterséges intelligencia”

Nincs pontos definíció.



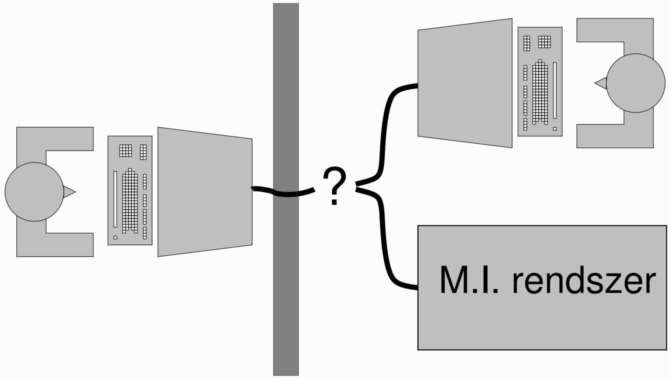
„Elvárások”

- intelligens viselkedés
- racionális viselkedés
- gondolkodó rendszer
- cselekvő rendszer

Cog.Bot.Lab – München
J. Schmidthuber

Turing-teszt

Egy megfigyelő tesztel egy rendszert, melyről nem tudja, hogy
ember **vagy** gép.
Feladat, hogy a feltett kérdések nyomán találjuk ki, hogy a rendszert
gép **vagy** ember vezérli.



Kérdésfelvetés:

Alan Turing — Képessé tehető – programozható – a számítógép a gondolkodás műveletére?

Neumann János — A fogalmak eléggé **pontos** specifikálása esetén a gép „intelligens” lesz.

Bevezető fogalmak

Mesterséges Intelligencia – „A.I.”

Emberhez hasonlóan gondolkodó rendszerek Bellman: döntéshozatal, problémamegoldás, tanulás automatizálása.	Racionálisan gondolkodó rendszerek Charniak: Mentális képességek tanulmányozása.
Emberhez hasonlóan cselekvő rendszerek Rich: Végeztetni dolgokat, melyeket az emberek jobban tudnak.	Racionálisan cselekvő rendszerek Schalkoff: Utánozni és magyarázni az intelligens viselkedést.

Russell, 1996

Az M.I. fejlődése

- 1956 – első M.I. konferencia Darthmouth-ban.
- „Alapítók”:
  - Minsky (Logo, Neurális háló),
  - McCarthy (Lisp),
  - Shannon (információ-elmélet)

Fejlődési területek:

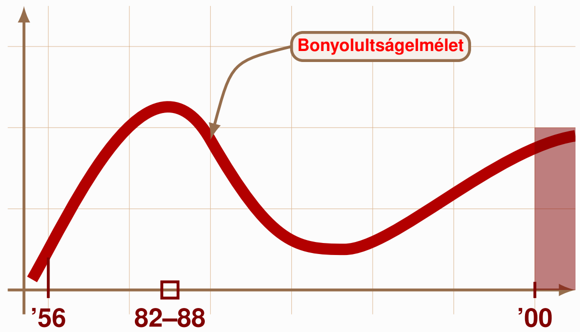
- szimbolikus M.I.
  - szakértői rendszerek
  - dedukciós algoritmusok
- „konnekcionista” megközelítések
  - neurális hálók
  - Boltzmann gépek
  - evolúciós algoritmusok

Ezekkel párhuzamosan:

- kognitív tudományok - cognitive neuroscience (CNS)
- Fuzzy algoritmusok

M.I. fejlődésgrafikon

Fejlődési grafikon
cikkek száma,
konferenciák
látogatottsága ...



- kezdetek - elméleti háttér: dedukciós algoritmusok, feladatok meghatározása,
- '80-as években – nagyon nagy a támogatottsága,
- később az érdeklődés csökkent, de
- 1997-ben a DEEP BLUE nyer a sakk-világbajnok ellen

M.I. fejlődése évszámokban

- '50 Turing: „Computing Machinery and Intelligence”;
- '56 Dartmouth: „Mesterséges Intelligencia”;
- '52–'69 „Look, Ma, no hands!”[1];
- '50– Sakk – Samuel, logika – Newell & Simon, „Geometry Engine” – Gelernter;
- '65 logikai következtető algoritmus – Robinson;
- '66–'73 Bonyolultságelmélet – csökkenő támogatottság;
- '69–'79 Tudásalapú rendszerek;
- '80– M.I. ipari ágazat;
- '86– Neurális háló modellek újra népszerűek;
- '87– M.I. tudományág;

[1] *vicc utolsó előtti sora. Utolsó: „Look, Ma, no teeth!”*

M.I. jelen időben I

„Modern AI focuses on practical engineering tasks”

‡ Egy pragmatikus megközelítés.

Tudományterületek, melyek kiváltak:

- Felismerő rendszerek: minta-, beszéd-, OCR;
- Szakértői rendszerek;
- Gépi fordítás;
- Robotika;
- Játékelmélet;
- Dedukciós algoritmusok – Maple, Mathematica a Fermat–tétel bizonyítása, stb.

‡ `http://wikipedia.org`

M.I. – jelen időben II

Cinikusan: M.I. feladat

= egyelőre jó megoldás nem ismert;

= bizonyított, hogy a megoldáshoz nagyon hosszú idő kell.

Sikerek:

- Deep Blue, 1997 – Gary Kasparov-ot legyőzi (rendszerek, melyek legyőzik a Deep Blue-t),
- Robbins sejtés bizonyítása,
- tervezés – ütemezés az 1991-es iraki háborúban: 50.000 egység koordinálása,
- `Proverb` – keresztrejtvények megfejtése.

Könyvészet I

S.J. Russell, P. Norvig
Mesterséges intelligencia modern közelítésben. (második kiadás) Panem, 2005.

I. Futó (szerk)
Mesterséges intelligencia. Aula, 1999.

S.J. Russell, P. Norvig
Artificial Intelligence: a Modern Approach. Prentice Hall, 1995.

T.M. Mitchell
Machine Learning. McGraw-Hill, 1997.

C.M. Bishop
Pattern Recognition and Machine Learning. Springer Verlag, 2006.

Könyvészet II

- C.M. Bishop
  Neural Networks for Pattern Recognition. Oxford University Press, 1995.
- M.A. Arbib
  The Handbook of Brain Theory and Neural Networks. The MIT Press, 2003.

Könyvészet olvasása kötelező

Az előadás anyaga **csupán** útmutató a tanuláshoz és segít a jegyzetelés megkönnyítésében.

Könyvészet olvasása kötelező II

A vetített anyag nem elégséges a vizsgához.

M.I. algoritmusok gyakorlatban I

Index - '05 okt. 2 – **Nyelveket tanul a szoftver**
Automatic DIstillation Of Structures (ADIOS)

„Cél az agyban lévő szintaktikus és szemantikus ismeretek ... számítógépes modellezése.”

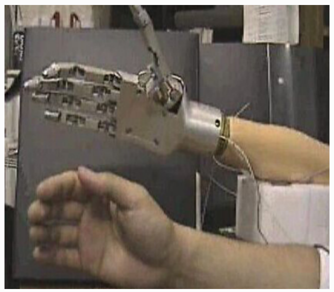
„A rendszer nyers adatokból (szöveg, beszéd, aminósav, hangjegy) SZABÁLYOKAT határoz meg.”

http://adios.tau.ac.il

M.I. algoritmusok gyakorlatban II

www.agent.ai – '05 szept. 19 – **Pontosan utánoz a műkéz**

A Southampton-i Egyetem mesterséges végtagja



A „Remedi-Hand” (Rehabilitation and Medical Research Trust) parányi feldolgozó egységen keresztül kapcsolódik a karizmokkal.
A készüléket a csuklót mozgató izmok összehúzódásai vezérlik.

M.I. algoritmusok gyakorlatban III

www.agent.ai – '05 szept. 16 – **Tökéletes ujjlenyomatok**

Genetikus algoritmusok a bűnüldözésben

Ujjlenyomatokról készült képek tömörítésekor nagyon óvatosan kell eljárni: a legcsekélyebb torzulás is hasznavehetetlenné teheti az ujjlemyonat képét.

M.I. algoritmusok gyakorlatban IV

yahoo.com – ’05 okt. 9 – **Stanford Volkswagen Wins $2M Robot Race**

http://robots.stanford.edu/

Defense Advanced Research Projects Agency, DARPA director Dr. Tony Tether, sets a medal on Stanford Racing Team’s Stanley #03.

Sebastian Thrun

Let’s Talk! The computer can translate V

- mouthing words in Mandarin
- 11 electrodes attached on face and neck
- computer program to figure out what he was saying

http://www.post-gazette.com/pg/05301/596293.stm

Fontos kérdések az implementáció folyamán:

Milyen modellek, rendszerek, algoritmusok voltak használva.

Feladat ?házi?

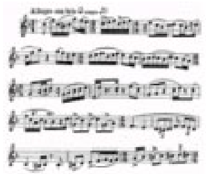
Találjatok a fentiekhez hasonló példákat, ahol a „mesterséges intelligens” eszközök sikeresek voltak.

+5 pont - kis bemutató

„Tudás”reprezentáció

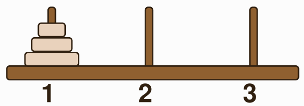
Ismeretek számítógépes formában való tárolása





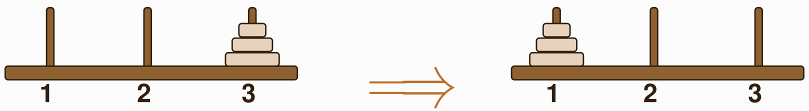

vagy:

Hanoi tornyok feladata



„Tudás”reprezentáció

Hanoi tornyok feladata



kezdeti

vég

Állapotok

$(3,3,3) \Longrightarrow (1,1,1)$

Szabály: nem helyezhető egy korong egy nála kisebb korong tetejére.

Szabály $\Longrightarrow$ **Állapottér**

„Állapottér”

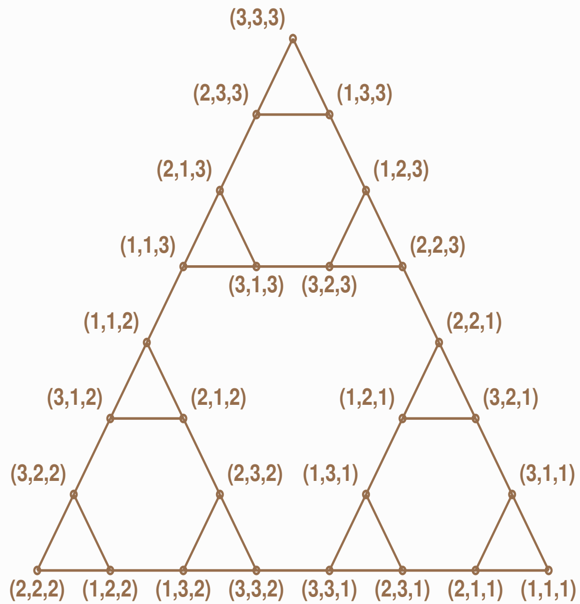
Állapottér: szabályos lépések sorozata.

Ábrázolási mód: irányítatlan gráf, minden lépés megfordítható.

Csúcs – állapot

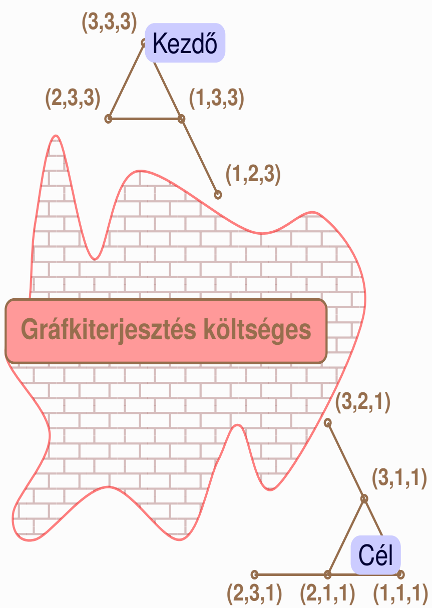
Él – lépés

Gráf



„Feladat”

Feladat:
a kezdeti állapotból:
$(3,3,3)$
a cél-állapotba
$(1,1,1)$
eljutni.



Megoldáskeresés az állapottérben

Hegymászó módszer

Heurisztika: az állapotokhoz rendel egy numerikus függvényt, mely maximum a kezdeti állapotban és minimum a vég állapotban.

$$\mathrm{Val}(\mathrm{CS}) = \sum_k \mathrm{Poz}_k$$

$$\begin{aligned} \text{kezdeti} &= 9 \\ \text{vég} &= 3 \end{aligned}$$

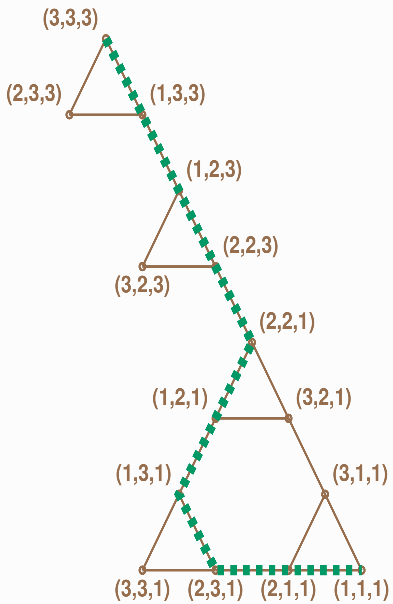
Megoldás a hegymászó módszerrel

Hegymászó módszer (Hill climbing)

Hegymászó:

- *ÁLLAPOT* $\Leftarrow$ *kezdőállapot*
- *Amíg ÁLLAPOT* $\neq$ *CÉLÁLLAPOT*
  - *Válassz ÚJ_ÁLLAPOT-ot*
  - *ÁLLAPOT* $\Leftarrow$ *ÚJ_ÁLLAPOT*

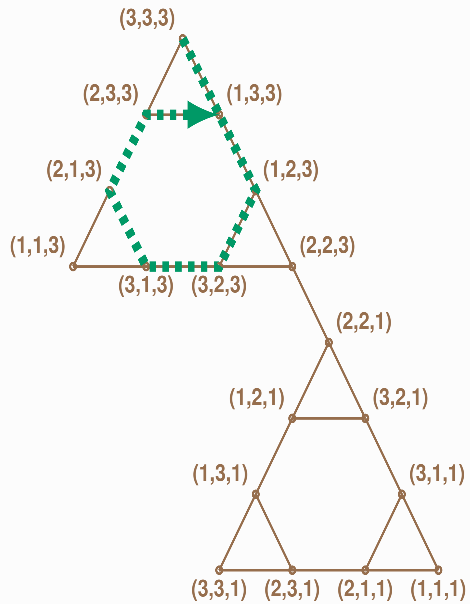
A következő lépés:
A $\mathrm{Val(CS)}$ legkisebb – szülőtől különböző – csúcs.



Hegymászó módszer

Jellemzők:

- Heurisztika nem bizonyítható a konvergencia,
- Nem kerüli el a ciklusokat,
- függ a paraméterezéstől: például a $(2,2,2)$-be nem írható algoritmus.



Backtracking I

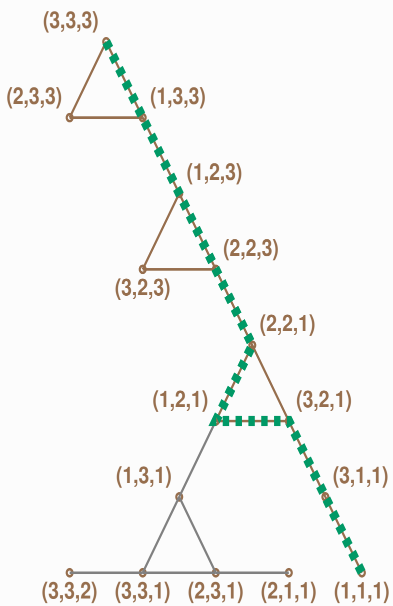
Visszalépéses keresés

Visszalép:

- *ÚT* $\longleftarrow$ *kezdőállapot*
- *Amíg ÚT–vég nem CÉL*
  - *Válassz SZ az út végére alkalmazható műveletek v. visszalép*
  - *ÚT* $\longleftarrow$ *SZ(ÚT)*

A választásnál lehet a definiált célfüggvényt használni.

SZ $=$ szabály

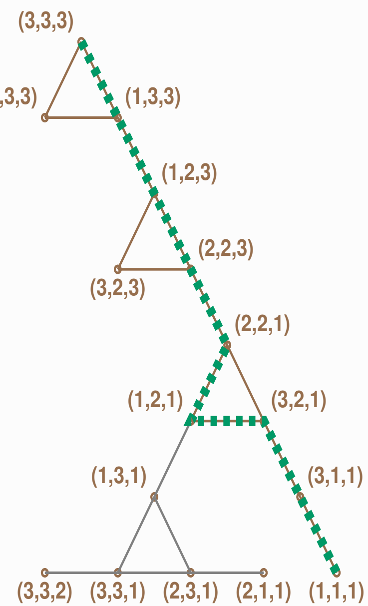


Backtracking Összefoglaló

Visszalépéses keresés

Fontos:

- a heurisztika $\longrightarrow$ hatékonyság,
- maximális úthossz korlátozás,
- jobb megoldás de nem optimális.



Gráfkeresés I

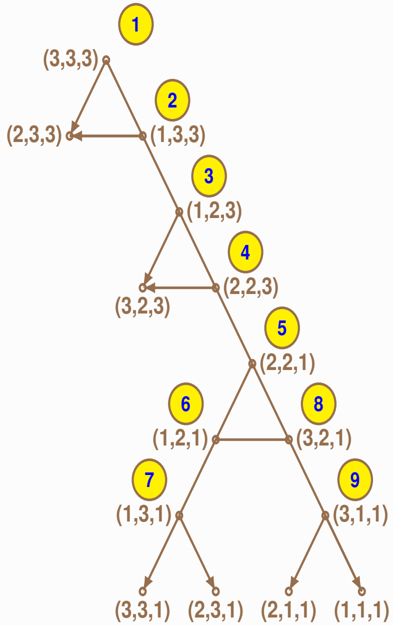
Keresés gráfban

Algoritmus:

- *GRÁF* $\longleftarrow$ *kezdőállapot*
- *Amíg GRÁF* $\not\ni$ *CÉL*
  - *Válassz SZ GRÁF-ra alkalmazható műveletek*
  - *GRÁF* $\longleftarrow$ *SZ(GRÁF)*

A választásnál lehet a definiált célfüggvényt használni.

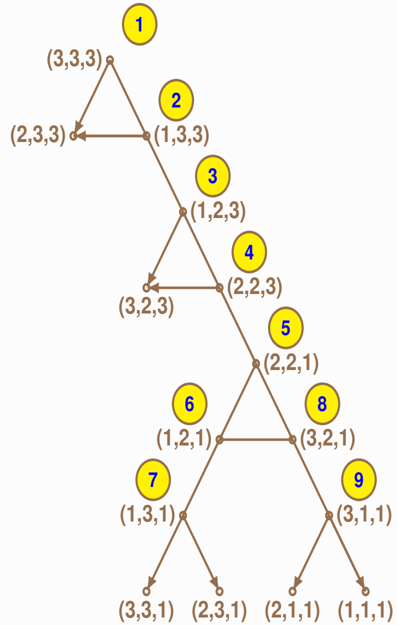
SZ $=$ szabály



Gráfkeresés II

Keresés gráfban

- Felépíti a gráfot,
- A legköltségesebb,
- nem találja meg a legrövidebb utat;
  Elérhetjük a célcsúcsot úgy is, hogy olyan csúcso(ka)t hagyunk ki, melyek a legrövidebb út részei lennének.



Feladat dekompozíció I

Rekurzív függvényhívás iskolapéldája

Jelölje: $\langle n, i, j, k\rangle$ a műveletet, melyben

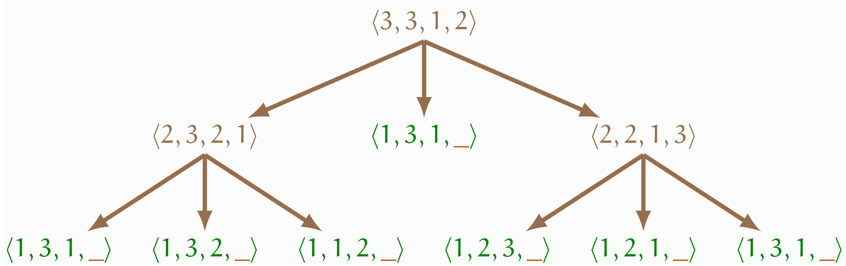
- a legfelső n korongot
- az i-edik rúdról
- a j-edik rúdra helyezzük
- a k-adik rúd segítségével

A feladat dekomponálható:

$$\langle n, i, j, k\rangle \longrightarrow \langle n-1, i, k, j\rangle\,\langle 1, i, j, k\rangle\,\langle n-1, k, j, i\rangle$$
ha $n > 1$

$n = 1$ – nem kell tovább bontani a feladatot

Feladat dekompozíció II



ÉS/VAGY gráf:

- csúcs $=$ probléma
- köteg - a részfeladatok, melyeket meg kell oldani a feladat megoldásához.
- Itt nincs **VAGY** csúcs.
- megoldás $=$ részgráf, melyben minden részprobléma csupa megoldható problémára vezethető vissza.

Predikátumkalkulus I

Szabályalapú következtetés
$t(i,j)$ – legfelső korong $i \Rightarrow j$ mozgatása.

A feladat megoldása mozgatások sorozata $\Rightarrow$ lista

Lista: $a.b.c.nil$

Lista axiómái: $A(\cdot,\cdot,\cdot)$ – append

$$(1) \quad A(nil, r, r)$$
$$(2) \quad A(u, v, w) \quad \rightarrow \quad A(s.u, v, s.w)$$

(1) – Üres lista nem változtat az eredményen
(2) – Ha w az u és v összetétele, ez érvényes egy s előtaggal is.

Predikátumkalkulus II

Hanoi tornyai axiómái:

$$\begin{array}{ll} (3) & H(1,i,j,k,t(i,j).nil) \\ (4) & H(n-1,i,k,j,y) \wedge H(1,i,j,k,t(i,j).nil) \wedge \\ & H(n-1,k,j,i,z) \wedge A(y,t(i,j).z,x) \\ & \rightarrow H(n,i,j,k,x) \end{array}$$

(3) – 1 *elemet átteszünk:* $t(i,j)$
(4) – n *elem áttételéhez előbb* $n-1$ *elemet mozgatunk* y *sorozattal, egy elemet* $t(i,j)$*-vel, majd* $n-1$*-et vissza.*

Kérdés: $(5) \quad (\exists x)\ H(2,1,2,3,x)$

(5) – *azon mozgatások, melyek megvalósítják* 2 *korong mozgatását.*

Predikátumkalkulus III

Algoritmus:

- *GRÁF* $\Longleftarrow$ *célállítás*
- *Amíg GRÁF-ban nincs ellentmondásmentes levezetés*
- *Válassz SZ a GRÁF-hoz alkalmazható illesztések vagy visszalépés*
- *GRÁF* $\Longleftarrow$ *SZ(GRÁF)*

Egy **ÉS/VAGY** gráfot járunk be és keresünk egy gráfot, mely tényekben végződik és nem ellentmondóak az illesztések.

Rezolúció I

Két elemű Hanoi-toronyra a kérdés:

$$(\exists x)\ H(2,1,2,3,x)$$

Rezolúció: bizonyítani, hogy az axiómákból következik a célállítás.
Módszer: Tagadjuk a kijelentést és bizonyítjuk, hogy ez utóbbi hamis.

$$\overline{A \rightarrow B} \quad \Leftrightarrow \quad \overline{\overline{A} \vee B} \quad \Leftrightarrow \quad A \wedge \overline{B}$$

$$\Longleftrightarrow (1) \wedge (2) \wedge (3) \wedge (4) \wedge \overline{(5)} \quad \text{kielégíthetetlen}$$

$$\overline{(5)} \quad = \quad (\forall x)\ \overline{H(2,1,2,3,x)}$$

Rezolúció II

Bizonyítás: ellentmondásos axiómarendszer:

$$(1)\quad (\forall r)\quad A(nil, r, r)$$

$$(2)\quad (\forall \ldots)\quad \overline{A(u, v, w)} \vee A(s.u, v, s.w)$$

$$(3)\quad (\forall \ldots)\quad H(1, i, j, k, t(i, j).nil)$$

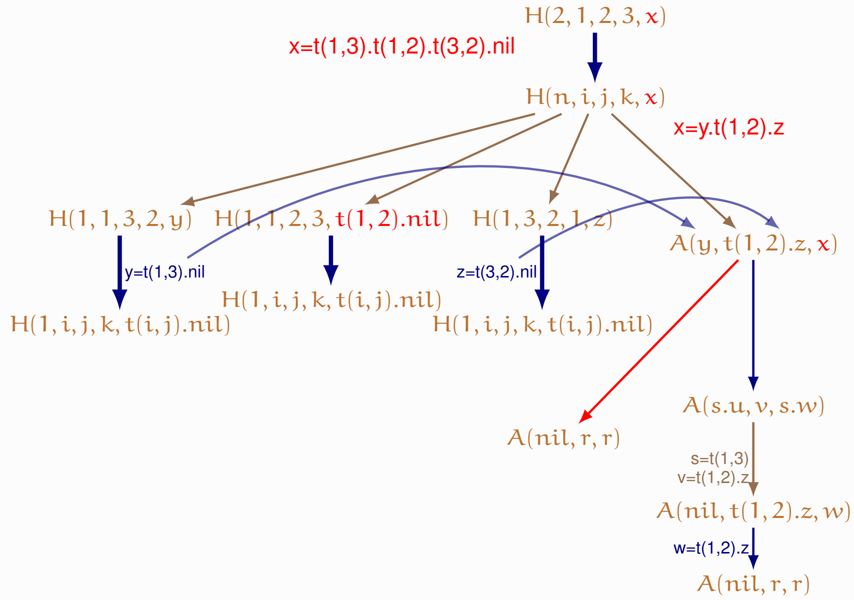
$$(4)\quad (\forall \ldots)\quad \overline{H(n-1, i, k, j, y)} \vee \overline{H(1, i, j, k, t(i, j).nil)} \vee \overline{H(n-1, k, j, i, z)} \vee \overline{A(y, t(i, j).z, x)} \vee H(n, i, j, k, x)$$

$$(5)\quad (\forall x)\quad \overline{H(2, 1, 2, 3, x)}$$

Cáfolati gráf: létezik út, melyre fennáll az $(1) - (4)$ és $\overline{(5)}$.

Figyeljük meg az univerzális kvantorokat!

Rezolúció Példa



Keresőrendszerek I

Keresőrendszerek (Production systems)

Különválasztják a

- feladat adatait ;
- az adatokon értelmezett műveleteket ;
- a vezérlést, mely a műveleteket algoritmussá szervezi.

Keresőrendszer: (Adat,Szabály,Vezérlés)

Keresőrendszerek II

Általános stratégia:

- **ADAT** $\leftarrow$ Kezdeti adatbázis
- AMÍG **ADAT** nem terminális
- Válassz **SZ** az ADAT-ra alkalmazható szabályok közül,
- **ADAT** $\leftarrow$ SZ(**ADAT**)

Keresési stratégia: az alkalmazható szabályok közül egyet kiválaszt.

Keresési stratégia:

- előrehaladó – *hegymászó, visszalépés, gráf*
- visszafelé haladó – *szabályalapú*
- kétirányú – bidirectional

Ismeretábrázolás

Ismeretek osztályozása:

- deklaratív ismeret — *állapot,részprobléma,axiómák*
- procedurális ismeret — *művelet, dekompozíció*
- vezérlési ismeret — VAL *függvény*

Közös vonás: gráf $\Longrightarrow$ **gráfreprezentáció**.

ADAT = a reprezentációs gráf egy részgráfja.
= „Ablak”, melyet a szabályok módosítanak,
egy csúcs, egy részgráf.

Keresési stratégiák

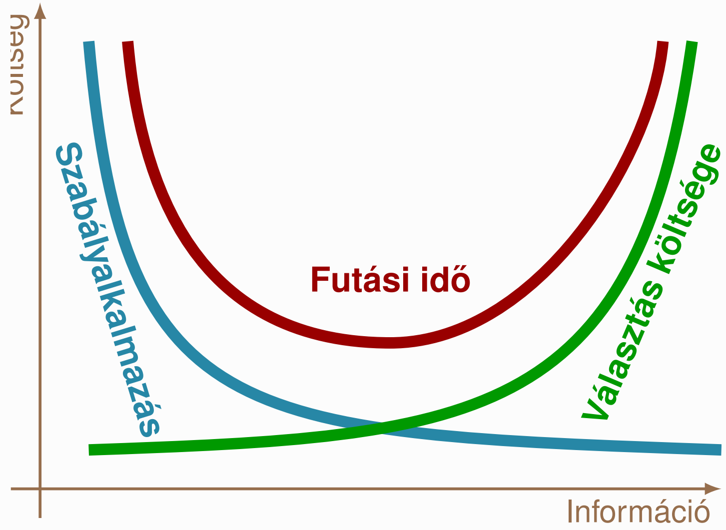
Gráfkeresési stratégiák:

- Elsődleges stratégia
  - nem-módosítható stratégia (hegymászó, rezolúció)
  - módosítható stratégia (szabályok választása)
- másodlagos stratégia – figyelembe veszi az adott reprezentációt.

Módosítható stratégiák:

- visszalépéses keresés – BackTracking
- gráfkereső – GraphSearch

A heurisztika szerepe

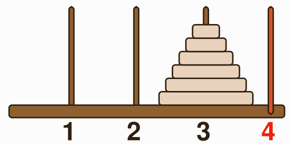


No free lunch.

Nehéz a futási időt optimizálni.
Közelítő megoldások „javasoltak”.

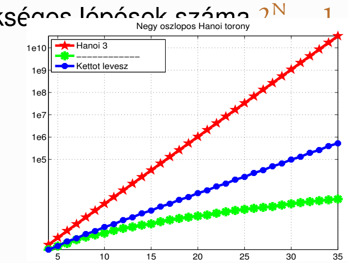
Négyes hanoi torony — Opc. feladat

A háromoszlopos hanoi toronynál az első oszlop korongjait kell egyenként áthelyezni a második oszlopra úgy, hogy mindhárom oszlopon a korongok lentről felfele csökkenő sorrendben legyenek.



Három oszlop esetében N korong áthelyezéséhez szükséges lépések száma $2^N - 1$

A feladatot módosítjuk úgy, hogy egy negyedik oszlopra is pakolhatunk. Ekkor a lépések száma csökken.



Feladat

Írjunk programot mely a négy-oszlopos Hanoi tornyokat kevés lépésszámmal oldja meg.

(5 pont)

Opcionális feladat

Hogyan írjunk jól angolul? I

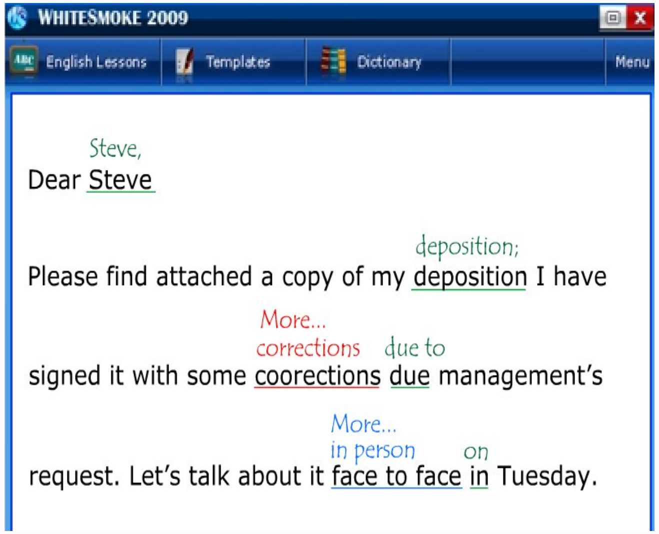
A WhiteSmoke "szövegértő" szoftvere.

„Előfordul, hogy jól beszélünk angolul, ám fontos leveleinkbe becsúsznak hibák és a címzett az eredeti szándéktól különbözőnek olvashatja mondandónkat. Egy izraeli szoftver a helyesíráson és a nyelvtanon túlmutató megoldást kínál."

Míg például a Word helyesírási és nyelvtani ellenőrzője csak e két területen hatékony, addig jelen szoftver lényegesen többet tud: a szöveget mesterséges intelligencia segítségével fürkészi át, majd azt pontosabbá, egyértelműbbé és folyékonyabbá tevő javaslatokkal áll elő. (azaz ?kozmetikáz?)

agent.ai

Hogyan írjunk jól angolul? II



Gráfkereső stratégiák

Egy korai M.I. terület - külön tudományággá fejlődött

Nagyon sok feladatot lehet gráfokkal reprezentálni: a **gráfreprezentáció** az algoritmusok keresési tere.

1. irányított gráfok
2. ÉS/VAGY gráfok

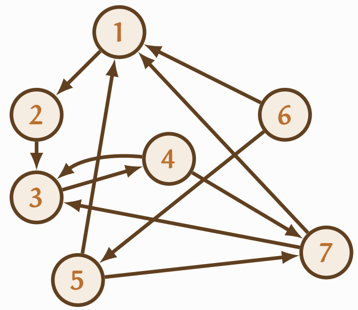
Példák:

- Hanoi tornyok reverzibilis lépések – irányítatlan gráf
- Irányított gráf?

Irányított GRÁFOK

Jelölés:

N – csúcsok (nodes)
A – élek $A \subset N \times N$ (adjacency)
szülő – 1 a 2-nek
utód – ...
$c(n, m)$ – költség



Tulajdonságok:

σ-tulajdonság: $\exists\sigma \ \forall n \quad |\{m|(n,m) \in A\}| \leq \sigma$

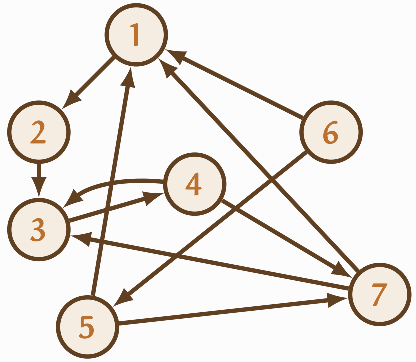
δ-tulajdonság: $\exists\delta > 0 \ \forall(n,m) \in A \quad c(n,m) \geq \delta$

hyper

Gráfok ábrázolása

δ-gráfok $=$ a δ és σ tulajdonsággal rendelkező gráfok.

$\times$	1	2	3	4	5	6	7
1	.	1	.	.	.	.	.
2	.	.	1	.	.	.	.
3	.	.	.	1	.	.	.
4	.	.	1	.	.	.	1
5	1	.	.	.	.	.	1
6	1	.	.	.	1	.	.
7	1	.	1	.	.	.	.



Konvenció: amennyiben nem specifikáljuk, az élek bejárásának a költsége $\mathbf{1}$.

Irányított utak

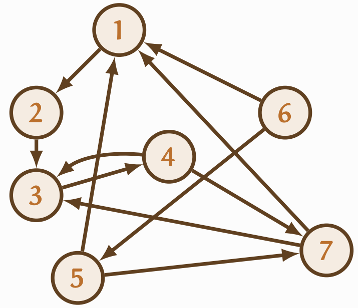
Irányított út – út: az n-ből az m-be

Ha $\exists\ n_1, \dots, n_k$ úgy hogy $\{(n, n_1), \dots, (n_k, m)\} \in A$.

Út: $\alpha = (n = n_0, n_1, \dots, n_k = m)$

Út költsége

$$c^\alpha(n, m) = \sum_{j=1}^{k} c(n_{j-1}, n_j)$$



Példa: $\alpha = (6, 5, 7, 3, 4, 3, 4, 7, 5, 1, 2)$ költsége $\mathbf{10}$.

Optimális út

Optimális költség: az n-ből az m-be

$$c^*(n, m) = \min_{\alpha \in \{n \to m\}} c^\alpha(n, m)$$

Optimális út: az n-ből az m-be

$$\alpha^*(n, m) = \arg \min_{\alpha \in \{n \to m\}} c^\alpha(n, m)$$

?

- Létezik mindig – optimális – út? nem. Ekkor az út hossza ∞
- Amennyiben igen, egyedi? nem - δ-gráf

Irányított Fa

Irányított fa: gráf, melyben egy kitüntetett csúcsból - a gyökérből – minden más csúcsba csak **egy** út vezet.

A gyökérbe nem vezet él.

Levél – csúcs, melyből nem vezet ki él.

Tulajdonságok:

- Bejárása egyszerű;
- Nem minden feladat ábrázolható faként.

ÉS/VAGY gráfok

ÉS/VAGY gráfok

Olyan irányított hipergráfok, melyekben egy hiperél egy csúcsból egy csúcshalmazba vezet.

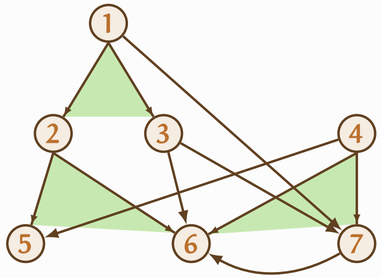
$$R(N, A) \quad \text{ahol} \quad A \subseteq \{(n, M) \in N \times 2^N \mid 0 \neq |M| \leq \infty\}$$

Hiperélek:
$(1,\{2,3\})$ $(1,\{4\})$
$(2,\{5,6\})$ $(3,\{7\})$
$(4,\{6,7\})$ $(7,\{6\})$

Élköltség: $c(n, M)$

Kérdés: σ és δ tulajdonságok

▸ G



Hiperutak ÉS/VAGY gráfokban

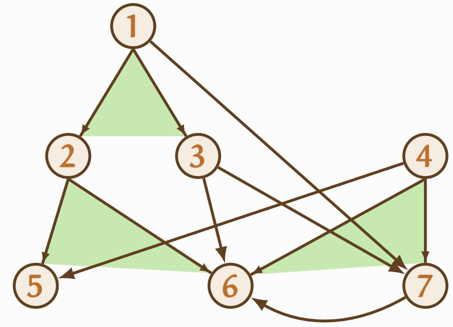
Irányított hiperút (n, M) között

Részgráf, melyben mindegyik csúcsból legfeljebb egy hiperél indul ki. M-ből nem indul hiperél.

Hiperutak $1 \to \{5,6\}$:

$(1,\{2,3\}), (2,\{5,6\})$

$(1,\{2,3\}), (3,\{6\}),$
$(1,\{4\}), (4,\{5\})$



ÉS/VAGY gráfok átalakítása

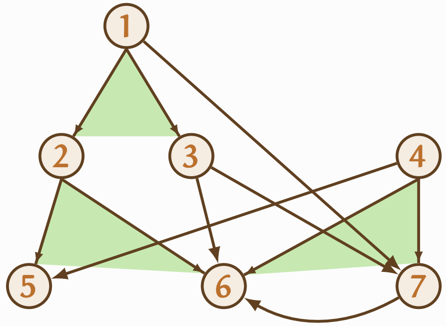
ÉS/VAGY gráfok kezelése nehézkes.

Átalakíthatóak irányított gráfokká.

Új csúcsok bevezetése: utódcsúcs = átalakítandó hiperél utódainak halmaza.

A fenti műveletet kiterjesszük a kezdőcsúcstól a célig.

Feladat: az $1 \to \{5,6\}$ egy hiperútjának megfelelő gráfátalakítás.



8-as kirakós játék I

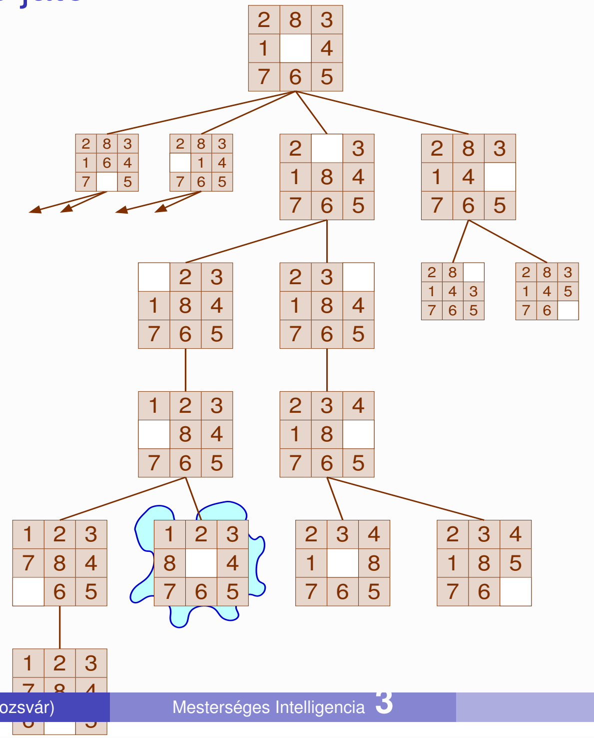
2	8	3
1		4
7	6	5

→

1	2	3
8		4
7	6	5

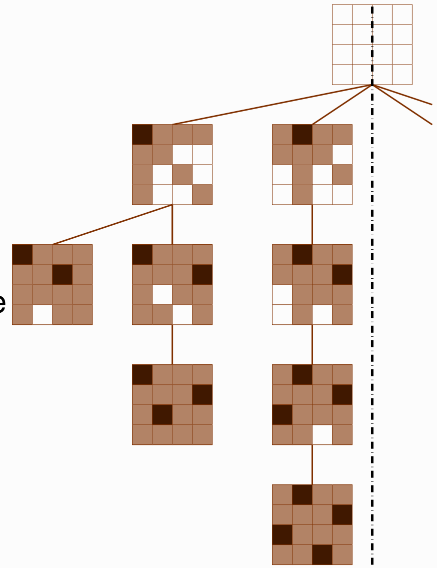
- **Kódolás:** 9 hely – $9! = 362880$ lehetőség.
- **Üres hely mozgatása** – meghatároz egy állapotgráfot.

8-as kirakós játék II



4 királynő

- 4×4-es táblán 4 királynőt elhelyezni.
- **Állapottér:** sakk–állások $1 - 4$ királynővel.
- **Művelet:** egy királynő egy mezőre helyezése.
- **Kezdőállapot:** üres sakktábla.
- **Célállapot:** 4 királynőt tartalmazó sakktábla.



Gráfkereső algoritmusok I

Nem-módosítható keresések:
Egy lépést – szabályt – nem lehet visszavonni.

1. Hegymászó algoritmus (hill-climbing)
   - kritérium-függvény, mely „vezérli” az algoritmust.
   - nem-determinisztikus
   - gond a lokális minimum jelenléte
2. Kommutatív rendszerek (commutative systems)
   - a D-re alkalmazható szabályok alkalmazhatóak a D leszármazottjaira is.
   - a D-ből előállított adatbázis független a műveletek sorrendjétől – felcserélhető.
   - ha a D kielégíti a terminálási feltételt, akkor annak minden leszármazottja is.

Nincs bonyolult stratégia. Heurisztika $\Longrightarrow$ hatékonyság.

Visszalépés I

- Egy utat tart nyilván a reprezentációs gráfból.
- Induló érték: start-csúcs.

Vezérlési stratégia – visszalépés alkalmazása ha:

1. nincs több él – **zsákutca**;
2. nincs több „jó út” – **vágás**;
3. minden továbbvezető útról visszaléptünk – **torkolat**;
4. egy már bejárt csúcshoz jutunk – **kör**;
5. túl hosszú a bejárt út – **mélységi korlát**.

Visszalépés II

Visszalépés ha
1. nincs több él – **zsákutca**;
2. nincs több „jó út” – **vágás**;
3. minden továbbvezető útról visszaléptünk – **torkolat**;
4. egy már bejárt csúcshoz jutunk – **kör**;
5. túl hosszú a bejárt út – **mélységi korlát**.

Tétel

A visszalépéses algoritmus az (1) és (2) feltételekkel terminál véges és körmentes gráfokon.

Visszalépés II

Visszalépés ha
1. nincs több él – **zsákutca**;
2. nincs több „jó út” – **vágás**;
3. minden továbbvezető útról visszaléptünk – **torkolat**;
4. egy már bejárt csúcshoz jutunk – **kör**;
5. túl hosszú a bejárt út – **mélységi korlát**.

Tétel

A visszalépéses algoritmus az (1)–(5) feltételekkel mindig terminál. Ha létezik a mélységi korlátnál nem hosszabb megoldás, megtalálja azt.

Bűvös négyzetek I 1. laborfeladat

Feladat: ábrázoljuk a bűvös négyzetek keresését gráf-kiterjesztési feladatként:

- építsük fel a feladat állapotterét; defniáljunk egy gráfot a helyes megoldásokat eredményező kitöltések folyamataként;
- definiáljunk egy gráfkiterjesztési procedúrát;
- keressük meg az összes lehetséges megoldást gráfkereső (?backtracking?) módszerrel.

Bűvös négyzet: az az $N \times N$-es négyzet, melyben az elemek száma megegyezik sorok és oszlopok szerint.

$$S_{sor} = \frac{1}{N}\sum_{n=1}^{N^2} n = \frac{1}{N}\cdot\frac{N^2(N^2+1)}{2} = \frac{N(N^2+1)}{2}$$

Bűvös négyzetek II 1. laborfeladat

Követelmények:

- Dokumentáció, mely tartalmazza a
  1. paraméterterét a feladatnak,
  2. a gráfkiterjesztés lépéseit,
  3. a gráfbejárás sorrendjét.
- Program, mely az N szám ismeretében kiírja (pl. egy TXT állományba) az összes megoldást valamint kiírja a képernyőre a megoldások számát.

A bemutatás személyesen történik – valamely futtatási környezetben – úgy, hogy a programban módosítani lehessen paramétereket.

(8 pont)

Kötelező feladat

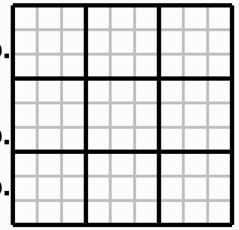
Bűvös négyzetek II — Példa

Example:

93	108	123	138	153	168	1	16	31	46	61	76	91
107	122	137	152	167	13	15	30	45	60	75	90	92
121	136	151	166	12	14	29	44	59	74	89	104	106
135	150	165	11	26	28	43	58	73	88	103	105	120
149	164	10	25	27	42	57	72	87	102	117	119	134
163	9	24	39	41	56	71	86	101	116	118	133	148
8	23	38	40	55	70	85	100	115	130	132	147	162
22	37	52	54	69	84	99	114	129	131	146	161	7
36	51	53	68	83	98	113	128	143	145	160	6	21
50	65	67	82	97	112	127	142	144	159	5	20	35
64	66	81	96	111	126	141	156	158	4	19	34	49
78	80	95	110	125	140	155	157	3	18	33	48	63
79	94	109	124	139	154	169	2	17	32	47	62	77

Sudoku — 2. laborfeladat

A sudoku-ban *számokat* helyezünk el egy $n^2 \times n^2$ méretű négyzetrácsban. Az $\{1, \ldots, n^2\}$ számokat úgy helyezzük el n^2-szer úgy, hogy egy oszlopban, egy sorban és minden kisebb négyzetben egy szám **egyszer** szerepeljen (lásd ábra).



- Az $n = 2$-re mi a paraméter–tér? **1p.**
- Jelenítsük meg „szépen” a megoldásokat az $n = 2$ és $n = 3$ esetekre. **1p.**
- Az $n = 2$ esetre generáljuk az összes megoldást. **1p.**
  - Találjuk meg egy részlegesen kitöltött feladat kitöltött változatát. **3p.**
  - × **Generáljunk** egy sudoku rejtvényt: egy részlegesen kitöltött feladat, melynek **csak egy** megoldása van. **4p.**

(10 pont)

Kötelező feladat

Bioinformatika

A számítástudomány és a molekuláris biológia között[2]

David Haussler & Judea Pearl

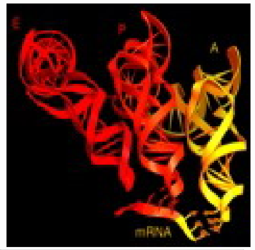
Kifejlesztették az emberi genomot feltérképező programokat. A lehetőséget a számítógép- technológia és a biokémia fejlődése biztosította. A genom biológiai összetevőinek felderítését és elemzését a tudós által kidolgozott **valószínűségi megközelítés** alapozta meg.

Az emberi génállomány mintegy hárommilliárd alappárt képez: a kettős spirál-alakú DNS négy alap-nukleotidból **(A, C, G, T)** épül fel; Minden egyes nukleotid része egy párnak. A mennyiség nagyon nagy, a munka csak számítógépes módszerekkel végezhető el.

agent.ai

[2] Pearl J (2000):**Causality**: Models, Reasoning, and Inference, The CUP Press

Bioinformatika



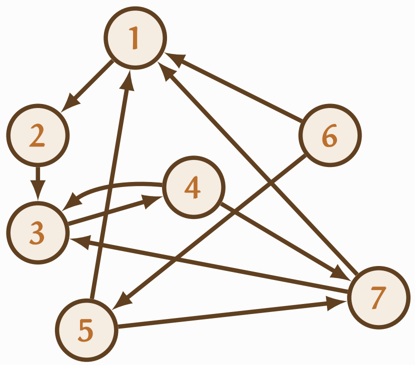
2001. júliusában közölték a módszer vázlatait, majd az emberi és egyéb organizmusok (egér, patkány, stb.) génszekvenciáit elemző, jegyzetekkel ellátott interaktív webalapú keresőket fejlesztettek.
Tudományos fórumot teremtettek, míg programjaikat gyakran használják különböző biomedikális kutatásoknál, kísérleteknél.

A gének evolúciója

A CBSE kutatásai az interdiszciplináris megközelítés jegyében folynak. Biológia, információs és nanotechnológia fúziójára, minél kisebb szerkezetek létrehozására törekednek.

Az emberi genom evolúciójának jobb megértése az egyik fo"irány: a cél érdekében permanensen fejlesztik az új statisztikai és algoritmikus módszereket.

Gráfkeresés



Feladatok:

- egy megoldás megtalálása
- minimális út keresése

Módszerek:

- Heurisztika

„Alapalgoritmus*” I

(* – Futó: Mesterséges Intelligencia, 73.o)

Visszalépés – backtracking – hátránya, hogy nem találja meg az optimális utat.

Gráfkereső algoritmus:

- a startcsúcsból indul s
- feltárja a reprezentációs gráfot G
  1. kiválaszt egy csúcsot, melynek utódai nem ismertek, $n \in \mathrm{NYILT}$
  2. kiterjeszti a választott csúcsot $G \leftarrow G \bigcup \Gamma(n)$

$$\mathrm{NYILT} \leftarrow \mathrm{NYILT} \setminus \{n\}$$
$$\mathrm{NYILT} \leftarrow \mathrm{NYILT} \bigcup \Gamma(n)$$

- addig keres, amíg egy célcsúcsot nem talál és van kiterjeszthető csúcs.

Alapalgoritmus II

Kommutatív rendszer – a kiterjesztések bármilyen sorrendben végrehajthatók.

$\Longrightarrow$

- A vezérlési stratégia nem informatív,
- Másodlagos stratégia $\Rightarrow$ továbblépés.

Módosítás:

$$n \leftarrow \underset{m \in \mathrm{NYILT}}{\operatorname{argmin}} f(m)$$

ahol $f : \mathrm{NYILT} \rightarrow \mathbb{R}$ kiértékelő függvény, amely

- egy csúcs „jósága”,
- pl. az s-ből m-be vivő legkisebb út hossza.
- $\Rightarrow$ dinamikus függvény. (felületes def.)

Alapalgoritmus III

Bevezetjük:

- kitüntetett **szülő csúcsot** (parent) $p : G \rightarrow G$
  (az s-ből egy utat specifikál). $p(s) = nil$
- **költségfüggvényt:** $g : G \rightarrow \mathbb{R}$
  $g(m)$ az s-ből m-be vivő út költsége.

Az n csúcs minden k utód-csúcsára $\forall k \in \Gamma(n)$:

- Ha k új csúcs, vagy $k \notin G$
- Ha nem új és $g(k) > g(n) + c(n, k)$, akkor $k \in G$

$$p(k) \leftarrow n$$
$$g(k) \leftarrow g(n) + c(n, k)$$

Alapalgoritmus IV

Probléma:

- ha k zárt,
- korábban megkerestük utódjait
- és rövidebb utat találtunk,

$\Rightarrow$

- a g,p függvények nem helyesek a k utódain;
- a feszítőfa nem optimális.

Megoldások:

1. k összes leszármazottját újraértékeljük;
2. a k csúcsot visszatesszük a NYILT halmazba.
   Hátrány:
   - Nagyobb futási idő;
   - p nem mindig optimális.
3. Olyan f választása, mely garantálja, hogy nem lesz ilyen k.

Általános algoritmus

- $G \leftarrow \{s\}$, $\mathrm{NYILT} \leftarrow \{s\}$, $g(s) \leftarrow 0$, $p(s) \leftarrow nil$.
- `While` nem `ures`(NYILT),
- $n \leftarrow \arg\min_{m \in \mathrm{NYILT}} f(m)$
- `If cél(n) then` kilép.
- $\mathrm{NYILT} \leftarrow \mathrm{NYILT} \setminus \{n\}$
- `For` $\forall k \in \Gamma(n)$
- `If` $k \notin G$ `or` $g(k) > g(n) + c(n,k)$

  $p(k) \leftarrow n$

  $g(k) \leftarrow g(n) + c(n,k)$

  $\mathrm{NYILT} \leftarrow \mathrm{NYILT} \cup \{k\}$
- `endfor`
- $G \leftarrow G \cup \Gamma(n)$
- `endwhile`

Az általános algoritmus tulajdonságai

Tulajdonságok

Az általános gráfkereső algoritmus

- egy csúcsot véges sokszor terjeszt ki;
- véges gráfban mindig terminál;
- mindegyik $n \in \mathrm{NYILT}$ csúcs kiterjesztése előtt $\forall s \xrightarrow{*} n$ van m csúcs az optimális úton, mely
  1. $m \in \mathrm{NYILT}$,
  2. $g(m) = g^*(m)$,
  3. minden m-et előző csúcs az úton zárt;
- Egy véges gráfban, ha létezik megoldás, akkor az algoritmus egy célcsúcs megtalálásával terminál.
- Csökkenő kiértékelőfüggvény használata mellett optimális és konzisztens a feszítőfa.

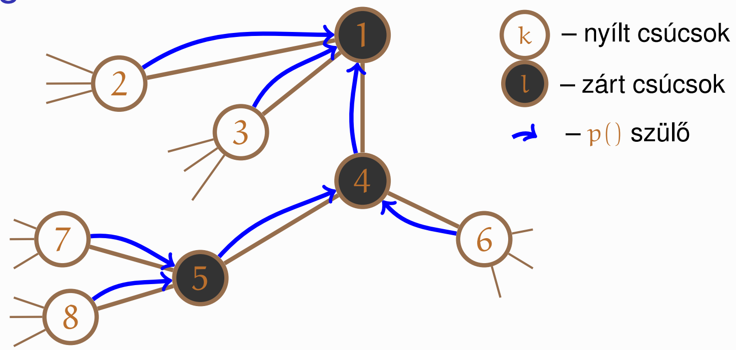
Bizonyítás

Nevezetes gráfkereső algoritmusok

Futó: Mesterséges Intelligencia – pp. 83

- Nem-informált gráfkeresések
  1. Mélységi keresés
  2. Szélességi keresés
  3. Egyenletes keresés
- Heurisztikus keresések
  1. Előretekintő keresés
  2. A algoritmus
  3. A^* algoritmus
  4. A^c algoritmus

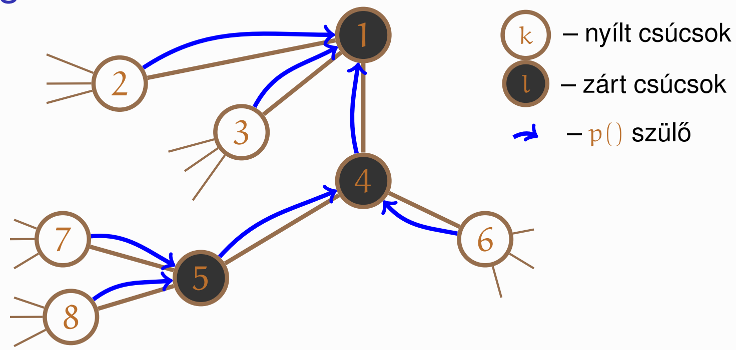
Példa gráfkeresésre



A harmadik iteráció végén tehát

- a zárt csúcsok halmaza $\{1, 4, 5\}$;
- a nyílt csúcsok halmaza $\{2, 3, 6, 7, 8\}$;
- a szülő-függvény $(z, p(z))$ párok: $\{(2,1), (3,1), (4,1), (5,4), (6,4), (7,5), (8,5)\}$

Példa gráfkeresésre



A következő csúcs kiválasztása a nyílt halmazból **bármilyen** kritérium alapján történhet. A kritérium alapja az $f(\cdot)$ függvény.

A függvény megválasztásával különböző keresési stratégiákhoz jutunk.

Mélységi keresés

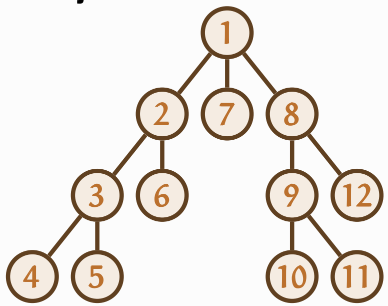
- Mindig a legmélyebben fekvő nyílt csúcsot választjuk,
- Ha minden él költsége ugyanannyi (pl. $c(m, n) = 1$), akkor a kiértékelő függvény:

$$f(n) = -g(n) \qquad \forall n \in NYILT,$$

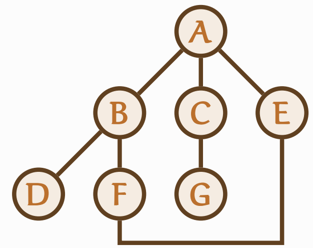
- Szükséges (? – mikor) a mélységi korlát bevezetése,
- Az algoritmus nem mindig talál megoldást,
- Iteratív növelése a mélységi korlátnak – megoldást talál (?optimális?).

Mélységi keresés Példák

Kiterjesztési sorrend:



Ellenpélda:

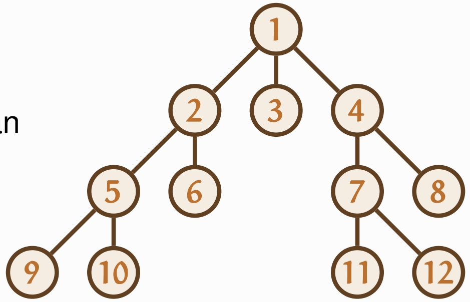


- Ha megjegyezzük a csúcsokat: A, B, D, F, E, C, G;
- Ha nem: $A, B, F, E, A, B, \ldots$

http://en.wikipedia.org/wiki/Depth_first_search

Szélességi keresés

- A mélységi keresés ellentettje,
- Mindig a legmagasabban fekvő nyílt csúcsot választjuk,



- Ha minden él költsége ugyanannyi (pl. $c(m,n) = 1$), akkor a kiértékelő függvény:

$$f(n) = g(n) \qquad \forall n \in \text{NYILT}$$

- Az algoritmus mindig talál megoldást – amennyiben ez létezik.

Egyenletes keresés

- Súlyozott változata a szélességi keresésnek,
- A kiértékelő függvény (általános $c(m,n)$ esetén):

$$f(n) = g(n) \qquad \forall n \in \text{NYILT}$$

- Az algoritmus mindig talál megoldást – amennyiben ez létezik,
- Dijkstra algoritmusa (1959).

Előretekintő keresés

- Heurisztikus kereső algoritmus,
- A kiértékelő függvény csak a heurisztika:

$$f(n) = h(n) \qquad \forall n \in \text{NYILT}$$

- $h(n)$ – heurisztikus függvény.
- pl. $f(n) = W(n)$ – a 8–as kirakójátékban;
- A keresőgráf kisebb, mint az egyenletes keresés esetében;
- A keresőgráf nem optimális;
- Erősen függ a választott keresőfüggvénytől.

„A” algoritmus

Futó: Mesterséges Intelligencia – pp. 90

„. . . ötvözi az egyenletes keresés óvatosságát az előretekintő keresés célratörésével, egyesítve előnyös tulajdonságaikat.”

- Kiértékelő függvény:

$$f(n) = g(n) + h(n) \qquad \forall n \in \text{NYILT}$$

ahol $h(n) > 0$.

- $h(n)$ „becsüli” az n-ből a cél–csúcsba vivő optimális út költségét.

„A” algoritmus tulajdonságai

Tulajdonságok

- $f^*(n) = g^*(n) + h^*(n)$ - a startból a célba az n-en keresztül vivő optimális út költségének a becslése.
- Ha az A algoritmus nem terminál, akkor minden $NYILT$ halmazba került csúcs véges sok lépés után kiterjesztésre kerül.
- Az A algoritmus mindig talál megoldást feltéve, hogy létezik megoldás.

„A*” algoritmus

- A algoritmus kiértékelő függvénye, ahol
- A heurisztikus függvény bármely csúcsban alulról becsüli a a célba vezető optimális út költségét, azaz

$$h(n) < h^*(n) \qquad \forall n \in G$$

- A fenti kritérium a heurisztika **megengedhetősége**.

Egy gráfkereső algoritmus megengedhető, ha megoldás létezése esetén megtalálja az optimális megoldást.

A^* tulajdonságai

- Bármely kiterjesztésre választott csúcsra $f(n) \leq f^*(n)$.
- Mindig optimális megoldással terminál, feltéve ha az létezik.

„A^c” algoritmus

- **Korlátozás a heurisztikus függvényre**: $h(n)$ kielégíti a monoton megszorítás (monotone restriction) feltételét, ha

$$h(n) - h(m) \leq c(n, m) \qquad \forall (n, m) \in A$$

- Egy n csúcs nem kedvezőtlen, ha egy utód m csúcs nagyon kedvező.
- A^c **algoritmus**: az olyan A algoritmus, ahol $h(n)$ monoton megszorításos és $h(t) = 0$ minden terminális csúcsra.

„A^c” algoritmus tulajdonságai

A^c tulajdonságai

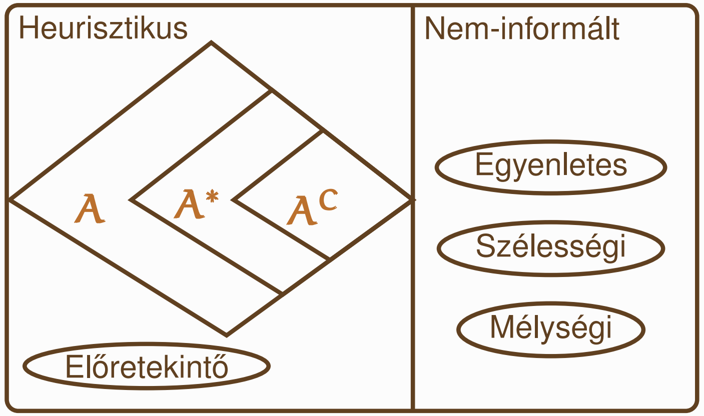
- Ha teljesül a monoton megszorítás, akkor egy n csúcsba vezető optimális út mentén a $g + h$ növekvő.
- Bármely n kiterjesztésre választott csúcshoz az optimális út van megjelölve:

$$g(n) = g^*(n)$$

- Mindig optimális megoldással terminál, feltéve ha az létezik.

Gráfkereső összefoglaló

- a heurisztika nagyon fontos – egy algoritmus alkalmazhatósága a választott heurisztikán áll vagy bukik.



Opcionális feladatok

- Gyakorló feladatok gráfokkal
  - Az A^* algoritmust használva jussunk el egy $I_1I_2I_3$ számból egy $J_1J_2J_3$ számba.
  - Keressük meg az $\{1, \ldots, N\}$ halmaz k részbe való felosztását.
  - Fejtsük meg a $SEND + MORE = MONEY$ feladatot.

(10 pont)

- Írjunk programot, mely megoldja a háromszög-kirakós játékot.

(12 pont)

- Rubik-kígyó megoldása.

(10 pont)

`http://.../feladatok/labor.pdf`

Blogbányászat

Ok-okozati viszonyokat tanul a gép
Amerikai kutatók[3] blogok elemzésére tanítják rendszerüket, mely történetmesélésre összpontosítva, nyelvi jegyek alapján szelektál közülük. A gyűjtött adatokból a kialakuló trendekre és viselkedésformákra következtet a rendszer.

A „tanítás” menete:

1. Blog-bejegyzéseket osztályoztak manuálisan a történet / nem történet osztályokba. **Eredmény:** a narratívák azonosítása.
2. A történetek elemei között az oksági kapcsolatok keresése. Pl: késő volt, lefeküdtem. **Eredmény:** tények + oksági kapcsolatok.

A rendszer sosem unatkozik
Távlati cél egy rendszer kidolgozása, mely napi rendszerességgel gyűjt és rendszerez adatokat.Ez fontos, mert más forrásokból hozzáférhetetlen, működése hasonló Google sertésinfluenza-követő programjához.
Mire jó a blogbányászat?
A blog-ok általában azonnali reagálást jelentenek, ezért garantált a gyűjtött „bányászott” információ aktualitása. Az élet legkülönbözőbb területeit érintik: filmek, könyvek, termékek, nemzetiségi-, vallási ellentétek, kábítószer-kereskedelem...

[3] **Andrew Gordon** Kreatív Technológiák Intézete Link

Szemantikus hálók

Futó: Mesterséges Intelligencia, pp. 186

Szemantikus háló:

- az emberi információtárolás és keresés modellezése (Quillian & Collins);
- gyakori név az asszociatív háló.
- kognitív pszichológiai kísérletek az „alapjai”;

Tulajdonságok:

- objektumokhoz tulajdonságokat rendelünk;
- hierarchia az objektumok szintjén $\Rightarrow$ absztrakció
- a tulajdonságok a legfelsőbb szinten asszociálódnak.

Quillian és Collins kísérlete:

Kísérlet: kérdések a madarakról és a reakcióidők mérése.

Kérdések:

1. Tud-e a kanári énekelni? — 1.3mp
2. Tud-e a kanári repülni? — 1.4mp
3. Van-e a kanárinak bőre? — 1.5mp

Hosszabb asszociációs lánc az utolsó kérdésnél.

Magyarázat: egy szemantikus hálóban a bőre és az énekel tulajdonságok nem egyforma távolságra vannak a kanári-tól.

Szemantikus hálók

Futó: Mesterséges Intelligencia, pp. 186

Szemantikus háló:

- az emberi információtárolás és keresés modellezése (Quillian & Collins);
- gyakori név az asszociatív háló.
- kognitív pszichológiai kísérletek az „alapjai”;

Tulajdonságok:

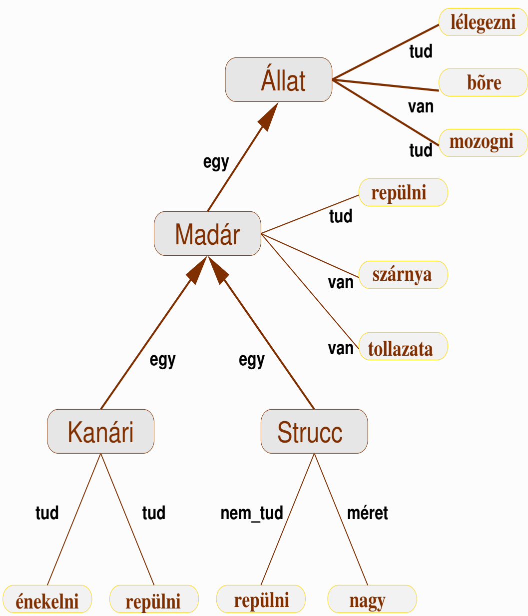
- objektumokhoz tulajdonságokat rendelünk;
- hierarchia az objektumok szintjén ⇒ absztrakció
- a tulajdonságok a legfelsőbb szinten asszociálódnak.

Kanári szemantikus háló

Szemantikus háló:

Irányított gráf, ahol

- **Csúcsok:** objektumok, objektumosztályok és tulajdonságok értékei;
- **Élek:** a csúcsok közötti kapcsolat neve.



Wordnet — Nagy szemantikus háló

Noun

- S: (n) canary, **canary bird** (any of several small Old World finches)
  - S: (n) finch (any of numerous small songbirds with short stout bills adapted for crushing seeds)
    - S: (n) oscine, oscine bird (passerine bird having specialized vocal apparatus)
      - S: (n) passerine, passeriform bird (perching birds mostly small and living near the ground with feet having 4 toes arranged to allow for gripping the perch; most are songbirds; hatchlings are helpless)
        - S: (n) bird (warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings)
          - S: (n) vertebrate, craniate (animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium)
            - S: (n) chordate (any animal of the phylum Chordata having a notochord or spinal column)
              - S: (n) animal, animate being, beast, brute, creature, fauna (a living organism characterized by voluntary movement)
                - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                  - S: (n) living thing, animate thing (a living (or once living) entity)
                    - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity)
                      - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow)
                        - S: (n) physical entity (an entity that has physical existence)
                          - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

http://wordnetweb.princeton.edu

Feladatmegoldás szemantikus hálókkal

Feladat: lekérdezés megválaszolása adott tárgyköri tudással.

Tárgyköri tudás: egy taxonomikus hierarchia – azaz egymásba ágyazott objektumok halmaza – számítógépes reprezentációja.

Adatbázis

Lekérdezés: egy célháló illesztése a szemantikus hálóba.

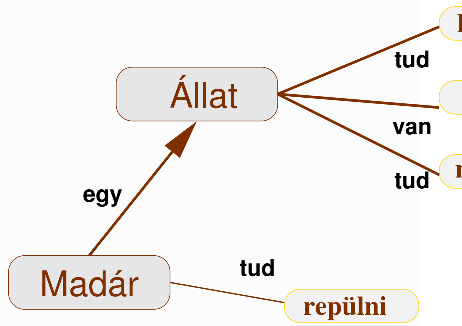
Illesztés

Milyen feladatokra megfelelő

Klasszikus logika nyelvén:

$$\forall x (x \in MADARAK \Rightarrow x \in REPUL)$$

Kivételek kezelése (strucc) nehézkes.



Melyik alkalmazás modellezhető szemantikus hálóval:

- játékok,
- vízelemző rendszerek,
- rendszerkonfigurálás

osztályozás, nyelvelemzés

?.

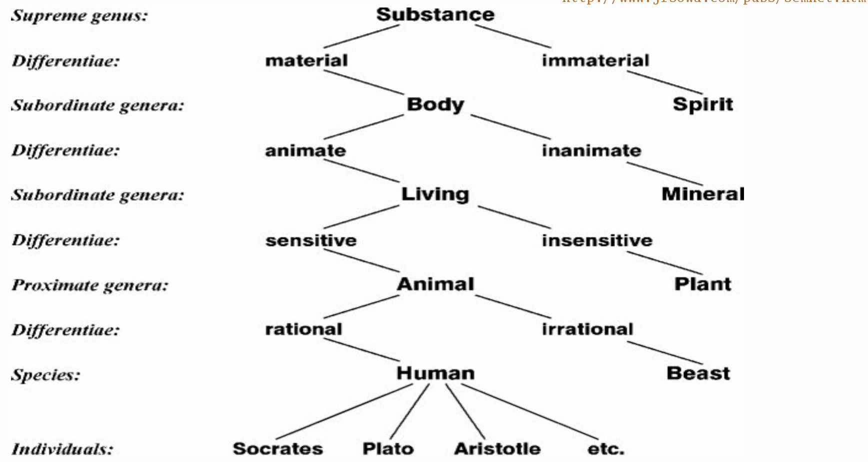
Összefoglaló – szemantikus hálók

- Fogalmak és kapcsolataik modellezése.
- Asszociatív memóriák.
- Információk egyszerű reprezentációja – !smiley! programozási paradigma jött létre.
- Ezt használjuk információ reprezentálására?

Történelem Keretrendszerek

Definciós hálók Kitérő

http://www.jfsowa.com/pubs/semnet.htm



Porfirius (i.sz. 300 körül) - magyarázata *Arisztotelész* „Kategóriá"jához.

Típus és **különbözőség** szerint rajzolt egy **definiciós hálót**, ahol alá- és fölérendelt kategóriákat különböztetett meg.

Keretrendszerek

Keretalapú Ismeretreprezentáció Fútó. pp.198

- 1975 Minsky - látás egy pszichológiai modelljének a leírása.
- A tanulmányozott világ fizikai vagy fogalmi entitásainak egy strukturált szimbolikus modellje.
- hasonlít a szemantikus hálókhoz – annak továbbfejlesztése.
- Új elem a procedurális reprezentáció.

Majdnem OOP - a különbség, hogy az OOP keretrendszer célja a kódolás és nem a tudásreprezentáció.

Level 5 Object

FRAME-es reprezentáció

frame *Személy*
instance-of: Class
azonosító: személyi
vezetéknév:
keresztnév:
end

frame *Főiskolás*
is-a: Személy
közös-cím: 'hallg@foisk.hu'
levél-cím:
end

frame *Kosarazó*
is-a: Személy
havi-juttatás:
end

frame *Kosár-center*
is-a: Kosárlabdázó
end

frame *Kosárcsapat*
instance-of: Class
edző: Személy
játékosok: collection-of Személy
end

frame *Főiskolás-kosárcsapat*
instance-of: Kosárcsapat
edző: Oktató
játékosok: collection-of Főiskolás

frame *Szöcskék*
instance-of: Főiskolás-kosárcsapat
játékosok: Péter, Tamás, ...
end

frame *Péter*
instance-of: Főiskolás
instane-of: Kosár-center
levél-cím: 'peter@foisk.hu'
magasság: 193
havi-juttatás: 10000
end

Démonok

Démonok

- Eljárások, melyeket osztályokhoz illetve azok attribútumaihoz lehet hozzárendelni.
- „paraméterezés”: mikor lépjenek működésbe (milyen esemény bekövetkeztekor).

when-needed-demon
when-changed-demon
when-deleted-demon
when-added-demon

Frame-rendszer működése: rendszer összes démonának együttes működése

(pl. útkereszteződés működtetése).

FRAME-ek tulajdonságai

- Egy keret vagy osztály vagy példány. Különbség az is-a illetve az instance-of között.
- Többszörös öröklődés - amikor egy osztály lehet több osztálynak az utóda.
- Példányok a hierarchia alján - nem lehet tovább példányosítani.

Mi történik egy hiányos osztályleírás esetén?
A rendszer a megszorítások alapján

- kiegészíti
- vagy nem

a hiányzó információkat (pl. Péter nem főiskolás).

Játékelmélet

Futó, pp. 215

... problems arising when we try to plan ahead in presence of hostile agents ...

Russell&Norvig, pp. 122

- Babbage (1846) - gépet tervez, mely Tic-tac-toe-t játszik,
- Leonardo Torres y Quevedo (1890) - sakk végjáték,
- von Neumann & Morgenstern (1944) - *Theory of games and Economic Behaviour*
- Shannon és Turing (1950) - sakkprogram, mert
  1. „intelligencia” szükséges a játékhoz,
  2. egyszerű szabályok,
  3. teljes informáltság,
- McCarthy (1956) - vágások.

Játékok és keresés

„Ismeretlen” ellenfél:

- Nem ismerjük a lépéseit,
- Feltételezzük, hogy nyerni akar.

Válasz: egy stratégia, mely az ellenfél minden lehetséges lépését figyelembe veszi.

Sakk-program esetében: nincs lehetőség az összes lehetőség vizsgálatára $\Rightarrow$ szükségesek a közelítések.

Mit közelítünk?

Játékok típusai

	determinisztikus	valószínűségi
Teljes információs	sakk, go	Dáma, Monopoly
Részleges információs	battleship	bridge, póker

?? Kupacos játék - Maya.

Kétszemélyes játékok I

Kétszemélyes teljes információs játékok:

- két játékos lép felváltva, adott szabályok szerint;
- a játékosok minden információval rendelkeznek;
- minden állapotban véges számú lépés létezik;
- véges a játszma ideje;
- az egyik játékos mindig nyer (esetenként lehetséges döntetlen...)

Ezzel a játékosztállyal foglalkozunk.

Kétszemélyes játékok II

Formális definíció:

- Két játékos, legyen MAX illetve MIN;
- MAX kezd;
- ismert kezdőállapot;
- műveletek, melyek leírják a lehetséges lépéseket;
- játék végének a tesztje;
- nyereség-függvény;

Stratégia: szabály az egyik – MAX – játékos optimális lépéseinek megadására.

Nim játék

Nim = „nip” + „muster”

Játék:

- gyufaszálak több sorban; m
- adott gyufaszál minden sorban; $[n_1, \ldots, n_m]$
- lépés = egy sorból i
  valahány gyufaszál elvétele; $0 \leq n_i' < n_i$
- játék vége: elfogynak a gyufaszálak; $\forall i;\ n_i = 0$
- veszít az a játékos, mely már nem tud gyufaszálat felvenni.

Változatok: vesztes az utolsó gyufaszálat felvevő; nem lehet tetszőleges számú elemet felvenni, *etc.*

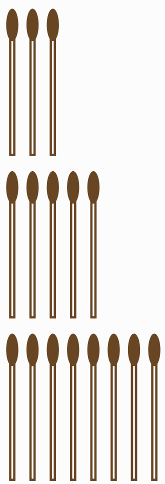
Nim játék

A játék állásai:

- **Nyerő** – ha a játékos tud úgy lépni, hogy az ellenfél lépéseitől függetlenül nyer;
- **Vesztő** – ha nincs nyerő lépés.

Nyerő stratégia:

$$\begin{array}{cccccc} 0 & 0 & 1 & 1 & = & 3 \\ 0 & 1 & 0 & 1 & = & 5 \\ 1 & 0 & 0 & 0 & = & 8 \\ \hline 1 & 1 & 1 & 0 & & (\mathrm{XOR}) \end{array}$$



Állítás: azon állások vesztők, melyekre az XOR csupa nullát eredményez.

Nim játék tulajdonságai

Állítás: azon állások vesztők, melyekre az XOR csupa nullát eredményez.

1. Lemma

Ha egy állásban az XOR nem csupa nullát eredményez, akkor van lépés, mely az XOR szerint nullát eredményez.

2. Lemma

Ha egy állapotban az XOR csupa nulla, akkor nincs lépés, mely eredményeként az XOR nulla lesz.

Nyerő stratégia

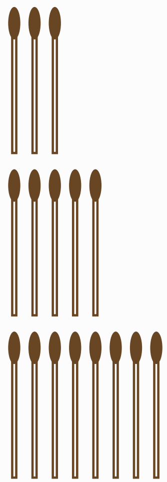
Ha XOR nem nulla, akkor le tudjuk nullázni és az ellenfél nem tud olyat lépni, hogy számunkra megint nulla legyen.

$\Longrightarrow$ nyerő stratégia.

Nim játék példa

Kezdeti állapot:

$$\begin{array}{cccccc} 0 & 0 & 1 & 1 & = & 3 \\ 0 & 1 & 0 & 1 & = & 5 \\ 1 & 0 & 0 & 0 & = & 8 \\ \hline 1 & 1 & 1 & 0 & & (\mathrm{XOR}) \end{array}$$



Válasszuk az utolsó (8 elemes) sort.
Ahhoz, hogy mindenhol 0 legyen

$$0 \quad 1 \quad 1 \quad 0 \; = \; 6$$

kell maradjon, tehát 2 elemet kell elvenni.

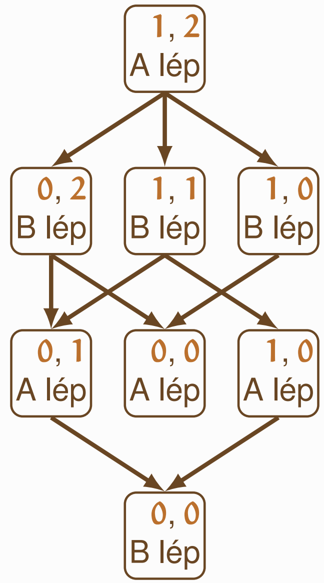
Az új pozíció vesztes.

Nim játék gráfja

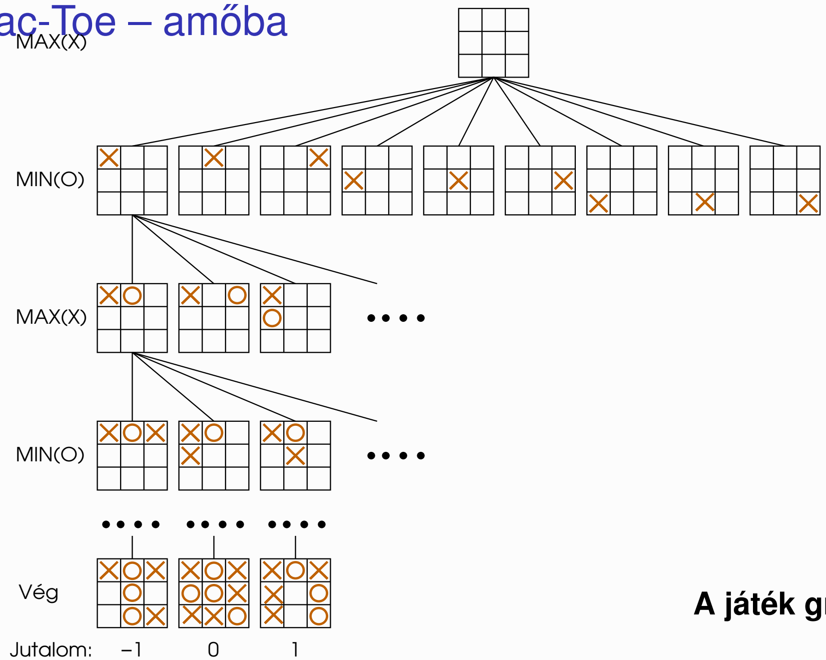
- játék gráfja véges mélységű;

Nyerő stratégia:

- mindig van legalább egy olyan lépés, melyből győzni tud;
- függetlenül attól, hogy az ellenfél mit lép;



Tic-Tac-Toe – amőba



A játék gráfja

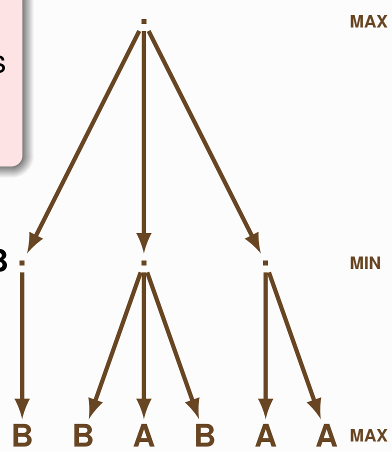
Nyerő stratégia létezése

Tétel

Egy teljes információjú kétszemélyes játék esetén mindig létezik egy játékos számára nyerő stratégia (ha nincs döntetlen).

Bizonyítás:
Az élek címkézése lentről felfelé. Ha **B** lép,

- $\exists$ ág, mely **B**-vel van címkézve, akkor **B**,
- ellenkező esetben **A**.



Majdnem NIM   Példa

Kuglizás – ahol az összes bábú egy sorban van, tehát **csak** egy vagy két – *egymásmelletti* – bábút tudunk leütni.
Vesztes, akinek először nem marad bábúja.

Feladat:

- Határozzuk meg a játék állapotterét $k = 3$ egymásmelletti bábúra;
- Határozzuk meg, hogy az kezdő játékos nyer vagy veszít.
- Számítsuk ki, hogy a kezdő játékos nyertes-e $k = 5$ egymás-melletti bábú esetén.
- Írjunk programot, mely meghatározza, hogy a kezdő játékos nyer-e tetszőleges konfigurációnál.

Stratégiák keresése

- Stratégia nem kereshető mert a teljes gráf nem fér el a memóriában;
- túl sok állapot.
- Sakk:
  - ≈ 45 lépés tehát 90 mélységű fa,
  - ≈ 35 lehetőség;
  - $35^{90} = 10^{139}$ levél. 10^{80} összes elektron
- Deep Blue – $32\text{CPU} \times 8$ dedikált sakk-processzor (13-30 mélységig; 30 milliárd lépés/perc)

Minimax algoritmus

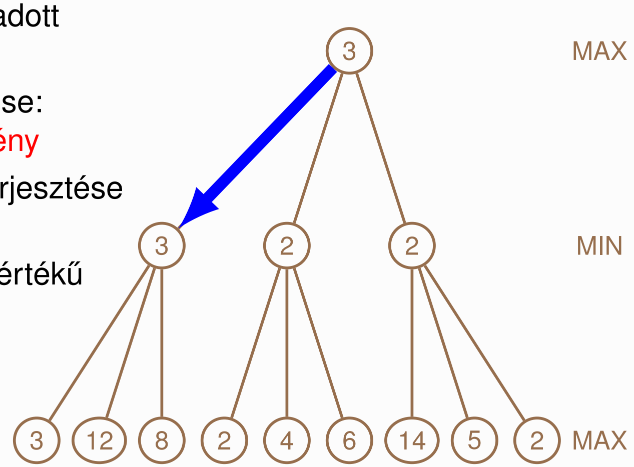
Minimax algoritmus:

- bonyolultabb játékok esetén használják;
- nem építhető meg a teljes játékfa;
- nem talál biztosan nyerő stratégiát;
- „erős” vagy „elég jó” lépés;

Közelítéseket adunk a nyerő/vesztes értékelés helyett.

Minimax algoritmus játékfákon

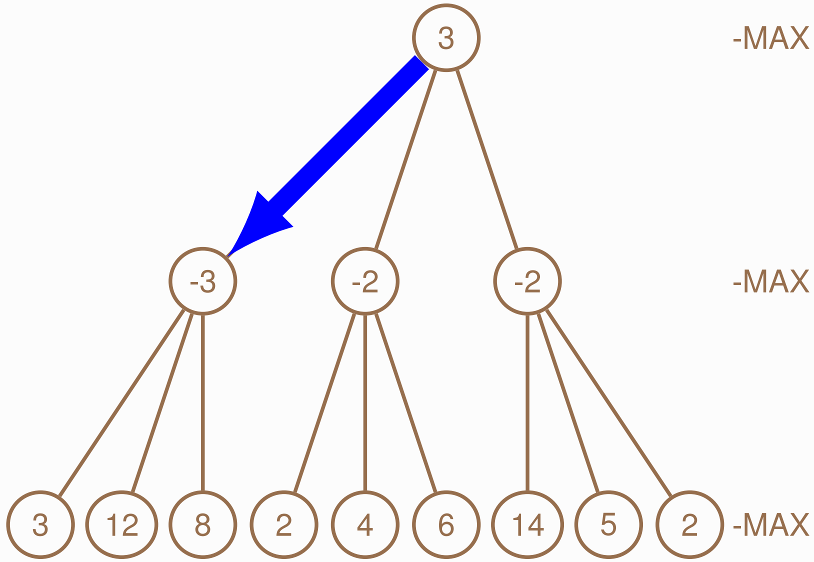
- Játékfa építése adott mélységig;
- Levelek címkézése: kiértékelő függvény
- Értékek visszaterjesztése lásd ▸ Nyerő stratégia;
- A gyökér-címke értékű lépés megtétele;



Negamax algoritmus

- A minimax algoritmusnál a különböző játékosok lépéseinél minimumot vagy maximumot kerestünk;
- A negamax algoritmus egyesíti a kétfajta optimális lépést:
  - minden lépésben maximumot számol,
  - ellenben az előző szint negált értékei szerint.
  - a javasolt lépés a csúcs negált értékű utódjába történő lépés;

Negamax algoritmus működése



Minimax/Negamax tulajdonságok

Tulajdonságok:

- Teljesség – megtalálja az optimális lépést, ha ilyen létezik?
  Ha a játékfa véges. igen
- Optimalitás – a legjobb lépést találja meg?
  Ha az ellenfél is racionális. igen
- Bonyolultság: $\mathcal{O}(b^m)$ kimenő élek – b
  kiértékelés mélysége – m
- Memóriaigény: $\mathcal{O}(bm)$

Sakk-program: $b \approx 35, \quad m \approx 100, \Rightarrow 35^{90}$

$35^{90} = 10^{139}$ levél v.ö: 10^{80} **összes elektron**

Hatékonyság növelése: VÁGÁSOK

Alfa-béta vágás I

- **Minimax/negamax algoritmus költséges** mert nagyon sok csúcsot kell generálni; azonban
- **Tudjuk**, hogy az értékelés a minimax szabály szerint történik;
- Az alfa-béta vágások módszere figyelembe veszi a már kiszámított csúcsok értékét (McCarthy 1956 – sakk)
- és **csak** olyan csúcsokat **nem** értékel ki, ahova racionális játék során eljutunk.

Kiértékelés során bevezetett változók:

α – MAX szint utódjainak maximuma;
csak növekedhet.

β – MIN szint utódjainak minimuma;
csak csökkenhet.

Alfa-béta vágás II

Vágás: művelet, melynek eredményeként nem értékeljük ki egy csúcshoz tartozó többi utódcsúcsot.

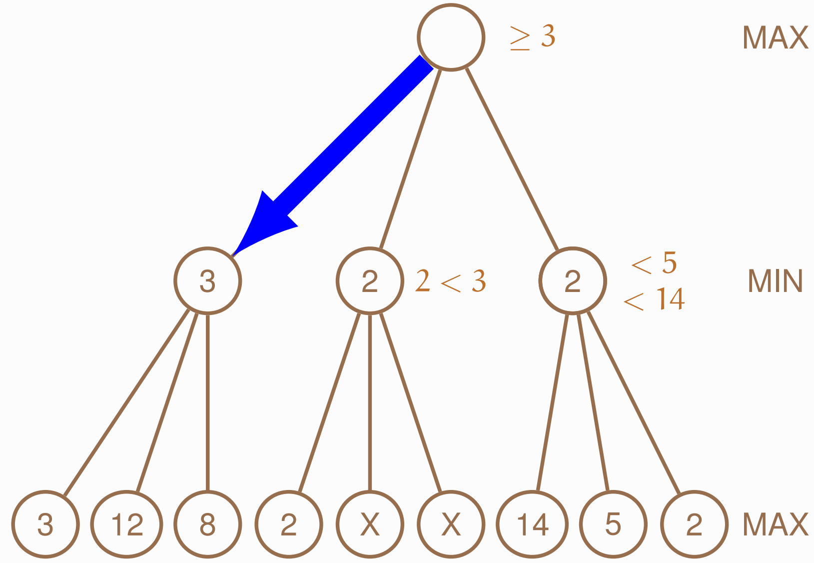
Vágási kritériumok:

- MIN csúcs alatt vágunk, ha az egyik őséhez rendelt α érték nagyobb, mint a csúcs β értéke.
  alfa vágás
- MAX csúcs alatt vágunk, ha az egyik őséhez rendelt β érték kisebb, mint a csúcs α értéke.
  béta vágás

Egy kiértékelést abba lehet hagyni, ha:

$$\alpha \geq \beta$$

Alfa-béta vágás – példa



Alfa-béta algoritmus tulajdonságai

- a vágások nem módosítják a megoldások minőségét;
- hatékony vágások (első eset) megvalósítása a csúcsok rendezésével: idő-komplexitás: $\mathcal{O}(b^{m/2})$
- Fontos a tudás – hol lehet „jó csúcsokat” találni – kódolása.

Létező játékprogramok:

- **Dáma** – Chinook program 40 éves nyerési sorozatot szakított meg ...
- **Sakk** – Deep Blue 1997-ben megverte Kasparov-ot.
- **Reversi** – nem játszanak: gép ... mindig nyer.
- **Go** – nem játszanak: gép ... mindig veszít.

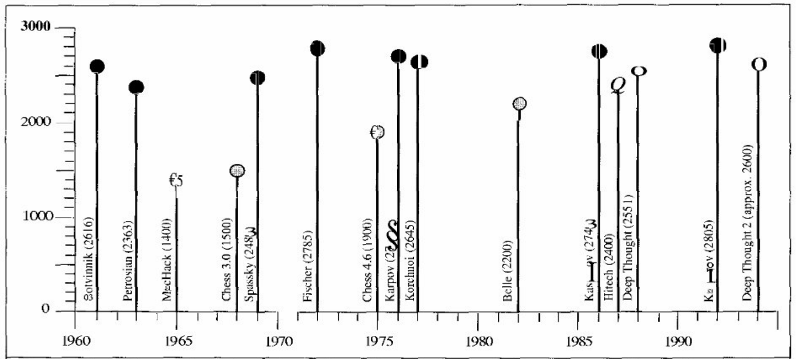


Figure 5.12 Ratings of human and machine chess champions.

Játékelmélet Neumann János

Jellemzők:

- két játékos van: a **sor** illetve az **oszlop** játékos.
- a sor-játékos a számára elérhető m stratégia közül pontosan egyet választ.
- az oszlop-játékos a számára elérhető n stratégia közül pontosan egyet választ.
- A választások egymástól **függetlenek**.
- Ha a sor játékos az i opciót, az oszlop játékos a j opciót választotta, a sor játékos nyeresége a_{ij}.

Stratégia: szabályrendszer, mely előírja, hogy egy játékos mit kell lépjen.

Neumann János tétele - kevert stratégiákra **null-összegű játékokon**.

Stratégia játékokban Példa

Két játékos, mindegyiknek van valahány stratégiája, melyek közül választhat ($A \longrightarrow 3$, illetve $B \longrightarrow 4$).

Nyereségmátrix az A számára; veszteség B-nek.

	B.1	B.2	B.3	B.4
A.1	-1	3	2	9
A.2	5	4	-2	6
A.3	8	-2	6	-2

? Ha A nem tudja, hogy B mit fog lépni, melyik a legjobb lépés?

Minimax tulajdonság

Nyereségmátrix: (a sor játékos számára)

$$\begin{array}{cccc|c} -1 & 3 & 2 & 9 & -1 \\ 5 & 4 & -2 & 6 & -2 \\ 8 & -2 & 6 & -2 & -2 \end{array}$$

Azon stratégiát válassza, mely a legkisebb nyereségek maximumát adja.

Minimax játékos:

$$\max_i \left(\min_j M_{ij} \right) \leq \min_j \left(\max_i M_{ij} \right)$$

azaz: az A legalább $\min\max$-ot nyer.

A sor-játékos az első sort választja -1 nyereséggel

Kevert stratégia

Nyereségmátrix:

$$\begin{array}{cccc|c} -1 & 3 & 2 & 9 & -1 \\ 5 & 4 & -2 & 6 & -2 \\ 8 & -2 & 6 & -2 & -2 \end{array}$$

- **De:** vannak a nyereségmátrixnak **sokkal** nagyobb elemei is, tehát kell legyen jobb választás;
- Kevert stratégiával játszunk: a **sor**játékos válassza:
  - x_1 valószínűséggel az első sort;
  - x_2 valószínűséggel a második sort;
  - x_3 valószínűséggel a harmadik sort ($x_3 = 1 - x_1 - x_2$);

Az oszlop játékos ugyanígy választhat.

Feltételezzük, hogy sokszor játsszuk ugyanazon mátrix szerint.

Kevert stratégia létezési tétele

Neumann-tétel

Minden nulla-összegű játék esetén létezik egy olyan keveréke a stratégiáknak, mely minimax tulajdonsággal rendelkezik.

Bizonyítás: $\mathbf{x} = [x_1, \ldots, x_n] \in \Pi_n \qquad \mathbf{y} \in \Pi_m$
Átlagnyereség: $\mathbf{x}^T M \mathbf{y}$

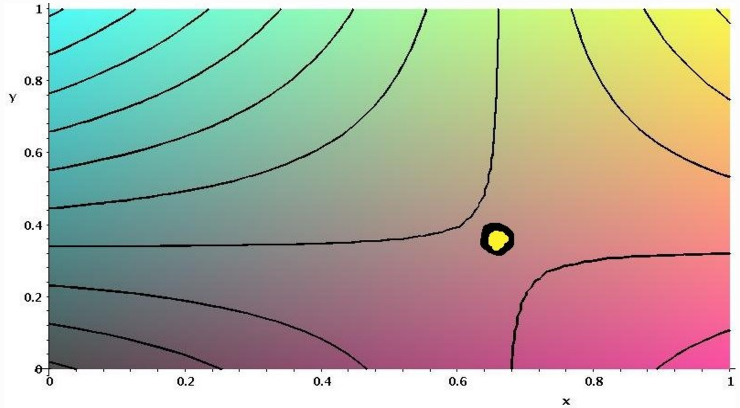
$$\max_{x\in\Pi_n} \min_{m\in\Pi_m} \mathbf{x}^T M \mathbf{y} = \underbrace{V = \overline{V}} = \min_{m\in\Pi_m} \max_{x\in\Pi_n} \mathbf{x}^T M \mathbf{y}$$

teljes biz. Finta - operációs kutatások

Minimax tétel

Példa: $\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 2 & 0 \\ \hline \end{array}$ Arányokkal: $\begin{bmatrix} x \\ 1-x \end{bmatrix}^T M \begin{bmatrix} y \\ 1-y \end{bmatrix}$

XMAPLE KÓD



A nyeresége maximális, ha $2/3$ valószínűséggel az első, $1/3$-dal a második stratégia szerint játszik.

Olló – papír – kő

Szabályok:

- két játékos játszik
- egyszerre mutatnak egy ...
  - papírt
  - követ
  - ollót
- **papír:** becsomagolja a követ;
- **kő:** kicsorbítja az ollót;
- **olló:** elvágja a papírt

0	1	−1
−1	0	1
1	−1	0

Teljes szimmetria $\Rightarrow$ nincs optimális stratégia.

„Optimális" kevert stratégia: $[1/3, 1/3, 1/3]$.

Egy paradoxon – Stratégiák

Székely J.G.: Paradoxonok a véletlen matematikájában.

Ketten játszanak: Q és R.

- Egy vagy két ujjat mutatnak fel;
- Páros ujj-szám esetén Q fizet R-nek, ellenkező esetben fordítva;
- annyit nyernek/veszítenek, ahány ujjat felmutattak.

Ismételt játékok esetén mi a nyerő stratégia?

Nyereség–mátrix:

	Q.1	Q.2
R.1	2	−3
R.2	−3	4

$$\mathbf{r}^T \mathbf{M} \mathbf{q} = 2r_1q_1 - 3r_1q_2 - 3r_2q_1 + 4r_2q_2$$

Mindegyik játékos úgy játszik, hogy az eredmény **ne** függjön a másik játékostól, valamint tudva, hogy $p_2 = 1 - p_1$ és $q_2 = 1 - q_1$, a játék értéke:
$V = (12r_1 - 7)q_1 + (4 - 7r_1)$
R választása: $12r_1 - 7 = 0$.
A megoldás:
$r_1 = 7/12,\ r_2 = 5/12,\ q_1 = 7/12,\ q_2 = 5/12$

Átlagnyereség: $-1/12$

Tizenegyes-rúgások Példa

- Tizenegyes–rúgásn: a játékos **rúg**, a kapus **véd**.
- A kapus a lövés előtt elmozdul.
- A játékos szintén a lövés előtt dönt.

	B	K	J
B	5	8	9
K	8	4	8
J	9	8	5

Milyen stratégiát kövessenek?
A bűntetőt **B**alra, **K**özépre, vagy **J**obbra lehet lőni, a kapus is erre vetődhet.
Kevert stratégiákat keresünk. Játékos $\mathbf{x} = [x_1, x_2, 1 - x_1 - x_2]$ és kapus $\mathbf{y} = [y_1, y_2, 1 - y_1 - y_2]$.

$$\mathbf{x}^T \mathbf{M}\, \mathbf{y} =$$
$$-8y_1x_1 - 4y_1x_2 + 4y_1 - 7y_2x_2 + 3y_2 + 4x_1 - 4x_1y_2 + 3x_2 + 5$$
$$= -x_1 (8y_1 + 4y_2 - 4) - x_2 (4y_1 + 7y_2 - 3) + 4y_1 + 3y_2 + 5$$

Innen: $y_1 = 2/5$; $y_2 = 1/5 \quad \Rightarrow y_3 = 2/5$.
(szimmetria okán a **kapus** ugyanígy kell eljárjon.)

Játékok – összefoglaló

Játékok – fontos szemléltető eszközök:

- **Nincs** tökéletes megoldás, tehát közelítenünk kell;
- **Formalizálás:** „jó ötlet azon gondolkodni, hogy min kell gondolkodni”;
- Játékok: mint a **Formula 1** az autók számára.

Bizonytalanság

Russell & Norvig, 1995, 415. o.

„In which we see what an agent should do when not all is crystal clear.” Futó et.al. 321–372

Ezidáig: következtetések olyan esetekben, ahol

- tudtuk egy esemény – tény – bekövetkezését.
- ismertük az események közötti kapcsolatrendszert: ok–okozat

Problémamegoldás során a tudásunk:

hiányos – nem tudunk/akarunk válaszolni;
nem megbízható – tudjuk, hogy max. 70%...
nem precíz – nincs megfelelő formalizmus a leíráshoz;
ellentmondásos – több forrás ⇒ konfliktus;

Bizonytalanság típusai

Hiányzó adat	Rosszul kitöltött kérdőív
Bizonytalan adat	A betegség csak valószínűsíthető
Bizonytalan fogalmak	Pl. „gyorsan hajtott”
Ellentmondások	Összegzésnél egymásnak ellentmondó adatok

Bizonytalanság kezelése

Numerikus modellek

- Bayes-modell,
- Bayes-hálók,
- Dempster–Schafer: megbízhatóságelmélet,
- Fuzzy modell

Szimbolikus modellek

- nem-monoton rendszerek

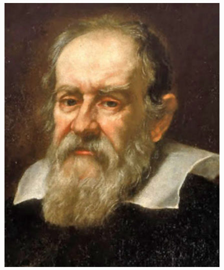
Cardano és Galilei

Gerolamo Cardano (1501–1576): *Liber de ludo aleae* - 1663
„Olasz matematikus, természettudós, csillagász és hazárdjátékos."
Először írja le a *tífuszt.*
Legismertebbek az algebrában elért eredményei: megoldotta a harmad- és negyedfokú egyenleteket – Ars Magna – és a megoldás során felismerte az imaginárius rész szükségességét. Termodinamikai megfontolások alapján tagadta az *örökmozgók* létezését.
Mindig pénzszűkében volt – ?$\Rightarrow$? – sakkozott és kockázott. Az játékokról írt könyve először foglalkozik a **valószínűség** fogalmával.

Cardano wiki

Galileo Galilei (1564–1642)
Olasz természettudós, matematikus, csillagász és filozófus, a tudományos forradalom úttörője.
S. Hawking: „Galileo a modern tudományok úttörője."
Javított a teleszkópok képességein, tökéletesítette a körzőt, bevezette a kinematikát, az asztronómiát, először írta le a Vénusz fázisait.

Kijelentette, hogy a *természet törvényei* matematikai jellegűek, valamint azt, hogy a Föld forog a Nap körül, (majdnem megégett).



Galileo wiki

Bayes modell I

- A legrégebbi modell, legjobban definiált technika (kockajáték);
- Alapja a valószínűségszámítás;
- Első modellt Cardano[4] alkotta; illetve Galilei 1660 körül, majd Borel és Kolmogorov a XX. században;

- **Valószínűségszámítás:** véletlen kísérletek;
- Kísérletek eredményei: **elemi események**;
- **Eseménytér:** összes esemény Ω
  - *teljes esemény* $T = \Omega$,
  - *üres esemény* $T = \emptyset$,
  - *ellentett esemény* $\overline{A} = \Omega \setminus A$,
  - *egymást kizáró események* ha $A \cap B = \emptyset$,

[4] Gerolamo Cardano (1501–1576): *De Ludo Aleae* (1663)

rev. Thomas Bayes

Thomas Bayes (1702–1761): angol matematikus és lelkész, a nevéről elnevezett tétel egy specifikus alakját alkotta meg.

Bayes wiki

Bayes modell II

Valószínűség: $P : 2^\Omega \longrightarrow [0,1]$ függvény, melyre

$P(T) = 1$ — T teljes esemény;

$P(F) = 0$ — F üres esemény;

$P(A \cup B) = P(A) + P(B)$ ha A és B egymást kizáró események.

$P(A)$ – az A esemény bekövetkeztének a valószínűsége ha *semmi információ nem áll rendelkezésünkre* más események bekövetkeztéről.

Tulajdonság: $P(A) + P(\overline{A}) = 1$.

Bayes modell III

Teljes eseményrendszer azon $\{A_1, \ldots, A_N\}$, $N > 0$ halmazok, melyekre

$$A_1 \cup A_2 \cup \cdots \cup A_N = \Omega \quad \text{és} \quad A_i \cap A_j = \emptyset \quad \forall i \neq j$$

Tétel: minden $\{A_1, \ldots, A_N\}$ teljes eseményrendszer esetén

$$P(A_1) + P(A_2) + \cdots + P(A_N) = P(A_1 \cup A_2 \cup \cdots \cup A_N) = P(\Omega) = 1$$

Jelölés: $A \cap B \stackrel{\text{def}}{=} AB$ és $P(A \cap B) \stackrel{\text{def}}{=} P(AB)$.

Feltételes valószínűség:
B esemény hatása az A-ra: $P(A|B)$

Bayes modell IV

Bayes szabály: feltételes valószínűség szabálya

$$\text{azaz} \quad P(A|B) = \frac{P(AB)}{P(B)} \quad \text{ha} \quad P(B) \neq 0$$

Tipikus felírás:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\overline{A})P(\overline{A})} \quad \text{ha} \quad P(B) \neq 0$$

Általános Bayes-szabály

Egy $\{A_1, \ldots, A_N\}$ teljes eseményrendszer esetén

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \quad \text{ha} \quad P(B) \neq 0$$

Bayes modell V

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \quad \text{ha} \quad P(B) \neq 0$$

A B eseményből következtetünk az A_i esemény feltételes valószínűségére.

Ismernünk kell az elsődleges valószínűségeket – **a–priori valószínűségeket:**

- $P(A_j)$;
- $P(B|A_j)$ feltételes valószínűségeket.

Bayes modell — példa

Adott a következő szabály:

Ha a beteg megfázott, **akkor** lázas. (75%)

Kérdés: **Ha** a beteg lázas, **akkor** megfázott. ??%

Jelölés:
M – a beteg megfázott
L – a beteg lázas

Szükségesek:
$P(M) = 0.2$ – megfázott
$P(L|M) = 0.75$ – lázas, feltéve, hogy megfázott
$P(L|\overline{M}) = 0.2$ – lázas, feltéve, hogy nem fázott meg

Ekkor:

$$P(M|L) = \frac{P(L|M)P(M)}{P(L|M)P(M) + P(L|\overline{M})P(\overline{M})} \quad \text{ha} \quad P(L) \neq 0$$

Azaz $P(M|L) = 0.15/0.31 = 0.483$

Bayes modell — összefoglaló

- Alkalmazható, ha minden információval rendelkezünk.
- Gyakorlatban az egymást kizáró eseményrendszer ritka...
- Előnyök:
  - Elméleti alap,
  - Jól definiált szemantika;
- Hátrányok:
  - nagyon sok valószínűséget kell megadni,
  - valószínűséget megadása nehéz,
  - változ(tat)ások követése nehéz.

Bayes hálók — I

- Bayes–modell hátránya a nagy valószínűségi tábla specifikálása,
- Bayes–háló (vélekedésháló, **belief network**) egyszerűsíti ezt a feladatot,
- Eszköze az **oksági kapcsolatok** leírása.

Bayes–háló

Egy adott feladat változóinak oksági struktúráját leíró irányított körmentes gráf.

Csomópontok az állítások, élek a kapcsolatok.

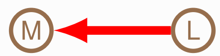
Élekhez rendelünk feltételes valószínűségi táblákat: összegzik a szülő változó hatását.

Bayes hálók — II

$P(M)$
0.2

M	$P(L\|M)$
0	0.20
1	0.75

Fordítva:

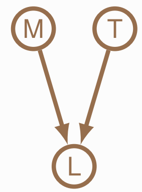


$P(L)$
0.31

L	$P(M\|L)$
0	0.07246
1	0.48387

Kiegészítés:
T - tüdőgyulladásos a beteg

$P(M)$	$P(T)$
0.2	0.03



$M\,T$	$P(L\|M,T)$
0 0	0.10
0 1	0.60
1 0	0.75
1 1	1.00

Bayes hálók III

Bayes–háló – egy adott terület teljes körű leírása.

„Algoritmus”:

1. A területet leíró változók meghatározása;
2. Írjuk fel a többi változótól független csomópontokat – tehát **gyökérváltozók**;
3. Amíg vannak csomópont nélküli változók:
    1. Válasszunk egy olyant, mely **csak** a már leírt változóktól függ.
    2. Képezzük azt a minimális halmazt, melyek mind közvetlenül hatnak az új csomópontra;
    3. Rajzoljuk be az új éleket és töltsük ki a **felt.val.** táblát;

Bayes hálók IV

Háló építésének az alapja

$$P(v_1, v_2, \ldots, v_N) = P(v_1|v_2, \ldots, v_N)P(v_2, \ldots, v_N)$$

Az indirekt függőségek kiesnek.

A felt.val. törvénye érvényes bármely rendezésre:

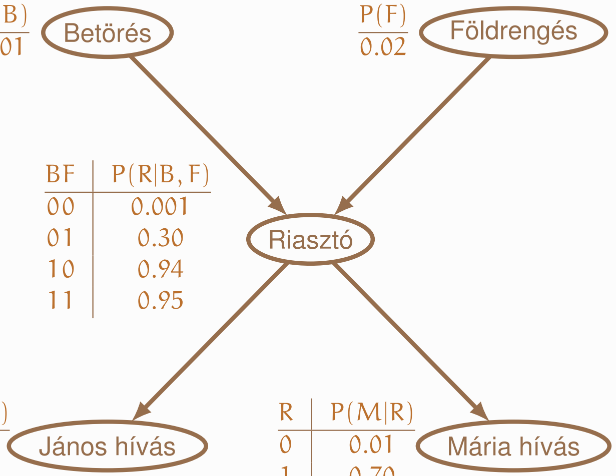
$$P(v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_N}) = P(v_{\pi_1}|v_{\pi_2}, \ldots, v_{\pi_N})P(v_{\pi_2}, \ldots, v_{\pi_N})$$

ahol π egy permutáció.

Jó – könnyen értelmezhető – Bayes–hálónál fontos az építés sorrendje.

Bayes hálók Példa I

Russell & Norvig pp. 437



P(B)
0.01

P(F)
0.02

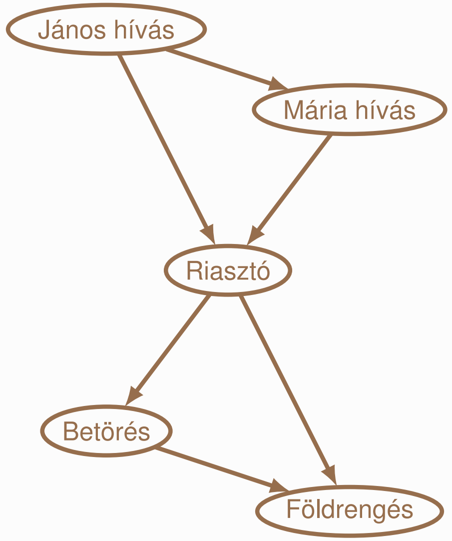
BF	P(R|B, F)
00	0.001
01	0.30
10	0.94
11	0.95

R	P(J|R)
0	0.05
1	0.90

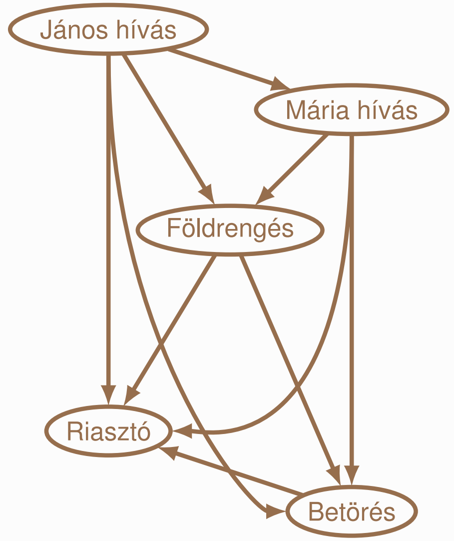
R	P(M|R)
0	0.01
1	0.70

Sorrend a háló építésénél: B, F, R, M, J .

Bayes hálók Példa II



Sorrend: J, M, R, B, F .



Sorrend: J, M, F, B, R .

Bayes hálók — Felhasználások

- Diagnosztizáló: hatásokból az okokra;
- Oksági kapcsolatokat vizsgáló;
- Kölcsönös kapcsolatokat vizsgáló;

Egyszerű műveletek, ha a háló egyszeresen összekötött.

Példa: Számoljuk ki a betörés (B) valószínűségét ha János hívott (J): $P(B|J)$
illetve akkor, ha hallottuk, hogy földmozgás is volt (F): $P(B|J,F)$

Grafikus modelleknél visszatérünk

Esettanulmány – PathFinder I

Russell & Norvig, pp. 457

PATHFINDER:

- szakértői rendszer nyirokmirigy-gyulladások diagnosztizálására;
- Fejlesztője D. Heckermann; Stanford Medical Computer Science ’80-as években;
- > 60 betegségtípus és > 100 szimptóma illetve teszt-eredmények.

http://research.microsoft.com/~heckerman

Esettanulmány – PathFinder II

Verziók:

I – szabályalapú rendszer, bizonytalanság beépítése nélkül;

II – kísérletek különböző modellekkel: Bayes háló a legjobb (10%).

III – Az „alig-valószínű” események.

IV – Valószínűségi háló – *Belief Net* – használata.

- 8 óra – szótár meghatározása;
- 35 óra – háló meghatározása;
- 40 óra – val.-ek becslése (1400 val.g).

PATHFINDER IV – „jobb”, mint az azt alkotó szakértők.

Dempster–Schafer modell — Bevezető I

Motiváció: szeretnénk döntéseket hozni akkor is, ha az események valószínűségeit nem ismerjük.

Például: Pacino-t meggyilkoltatta a maffia. A rendőrség a következő tényeket gyűjtötte össze:

- három bérgyilkos van: Tom, John, illetve Angie;
- a gyilkos kiválasztása a következő volt: ha egy dobás fej, akkor Tom, ellenkező esetben John vagy Angie;
- DNS-vizsgálatok eredménye 80%-ban férfit valószínűsít.

Kérdés: Mekkora az egyes személyek bűnösségének a valószínűsége?

Dempster–Schafer modell   Bevezető II

Nincs megfelelő valószínűségi modell, mely egyesíteni tudná a két állítást.

Bizonytalanság: A tények ismeretében nem tudunk személyt azonosítani.

Ábrázolás: a két ténylhalmazt ábrázoljuk mértékekkel (e_1 és e_2 arányban megbízható információ) :

Első: e_1 – $m(T) = e_1 \cdot 50\%$ és $m(J, A) = e_1 \cdot 50\%$

Második: e_2 – $m(T, J) = e_2 \cdot 80\%$ és $m(A) = e_2 \cdot 20\%$

Ismerethiány: nem tudunk valószínűségi modellt építeni, melyben ábrázolhatjuk mindkét forrást.

Dempster–Schafer modell   Bevezető III

Dempster-Schafer modell

Egy módja a többféle információ–forrás összetételének.

Amennyiben $e_1 + e_2 = 1$

$$m(T) + m(J, A) + m(T, J) + m(A) = 1$$

Ha $e_1 = e_2 = 50\%$, tudjuk a halmazok valószínűségét.

Nem tudjuk a személyek valószínűségeit.

Szeretnénk a fenti információkkal műveleteket végezni.

Dempster–Schafer modell   I

- **Cél:** megkülönböztetni
  - a bizonytalanságot *(uncertainty)*
  - az ismerethiánytól *(ignorance)*.
- Egy atomi B esemény valószínűségét **nem** ismerjük;

A valószínűség-tábla:

$m(T)$	$m(J, A)$	$m(T, J)$	$m(A)$
25%	25%	40%	10%

Például: a fentiek alapján tudjuk, hogy

$p(T) \geq 25\%$, ugyanakkor $p(T) \leq 65\%$.

Dempster–Schafer modell   II

Nyilvántartjuk azt, hogy

- mennyire támogatunk egy állítást $\mathrm{Bel}(F)$; (**belief**)
- mennyire „esélyes” az állítás $\mathrm{Pl}(F)$; (**plausibility**)

Példa: egy érme szabályossága a vizsgálat előtt:

$$\mathrm{Bel}(F) = 0 \quad \text{illetve} \quad \mathrm{Bel}(\overline{F}) = 0$$

de ha 90%-ban megállapítottuk, hogy szabályos, akkor

$$\mathrm{Bel}(F) = 0.5 \cdot 0.9 \quad \text{illetve} \quad \mathrm{Bel}(\overline{F}) = 0.5 \cdot 0.9$$

A „fennmaradó” 10% a bizonytalanságot tükrözi.

Dempster–Schafer modell III

Példa folyt.: intervallum-logika

- vizsgálat előtt: $p(F) \in [0, 1]$;
- vizsgálat után: $p(F) \in [0.45, 0.55]$;

Dempster-Shafer modell: intervallumok kombinálása.

Bayes analógia: alsó- illetve felső korlátai egy esemény valószínűségének.

Segédfüggvények bevezetése:

- $Bel(A_i)$ – alsó korlát; Belief
- $Pl(A_i)$ – felső korlát; Plausibility

Dempster–Schafer modell IV

„Belief”

$$Bel(A_1 \cup A_2) \geq \sum_{i=1}^{2} Bel(A_i) - Bel(A_1 \cap A_2)$$

„Plausibility”

$$Pl(A_1 \cap A_2) \leq \sum_{i=1}^{2} Pl(A_i) - Pl(A_1 \cup A_2)$$

A Bel és Pl függvények jellemzése:
Egy $m : 2^\Omega \to [0, 1]$ függvény segítségével, melyre

$$m(\emptyset) = 0 \,, \qquad \sum_{A \in \Omega} m(A) = 1$$

$$Bel(A) = \sum_{B \subset A} m(B) \,, \qquad Pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$$

Dempster–Schafer modell Tulajdonságok

$$Bel(A) = \sum_{B \subset A} m(B) \,, \qquad Pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$$

- bármely A halmazra $Bel(A) \leq Pl(A)$

$$B \subset A \;\Rightarrow\; B \cap A \neq \emptyset$$

- bármely A halmazra $Bel(A) = 1 - Pl(\overline{A})$

$$B \subset A \;\Leftrightarrow\; \overline{B \cap \overline{A} \neq \emptyset}$$
$$B \subset A \;\Leftrightarrow\; B \cap \overline{A} = \emptyset$$

- a Bel függvény szubadditív:
$Bel(A \cup B) \leq Bel(A) + Bel(B)$

$$C \subset A \wedge C \subset B \;\Rightarrow\; C \subset A \cup B$$

- a Pl függvény szuperadditív:
$Pl(A \cup B) \geq Pl(A) + Pl(B)$

Dempster–Schafer Példa

Jellemezzük az érménket megfigyelés előtt:

- legyen $m(\{F, I\}) = 1$ és minden másra $m(B) = 0$;
- ekkor $Bel(F) = m(\{F\}) = 0$ és $Pl(F) = m(\{F\}) + m(\{F, I\}) = 1$.

Megfigyelések után:

- $m(\{F\}) = m(\{I\}) = 0.45$ és $m(\{F, I\}) = 0.1$;
  – az összeg 1;
- ezért:
  1. $Bel(F) = m(\{F\}) = 0.45$, illetve
  2. $Pl(F) = m(\{F\}) + m(\{F, I\}) = 0.55$;
- azonosítani tudjuk a „határozatlan” részt:
  $m(\{F, I\}) = 0.1$

Dempster–Schafer modell Logikai műveletek

Logikai műveletek definíciója:

- logikai és:

$$\mathrm{Bel}(A_1 \wedge A_2) = \sum_{B \subset (A_1 \cap A_2)} m(B);$$

$$\mathrm{Pl}(A_1 \wedge A_2) = \sum_{B \cap (A_1 \cap A_2) \neq \emptyset} m(B);$$

- logikai vagy:

$$\mathrm{Bel}(A_1 \vee A_2) = \sum_{B \subset (A_1 \cup A_2)} m(B);$$

$$\mathrm{Pl}(A_1 \vee A_2) = \sum_{B \cap (A_1 \cup A_2) \neq \emptyset} m(B);$$

Általánosított entrópia:

$$AU(\mathrm{Bel}) = \max_{p_x} \left(-\sum_{x \in \Omega} p_x \log p_x \right), \quad \mathrm{Bel}(A) \leq \sum_{x \in A} p_x$$

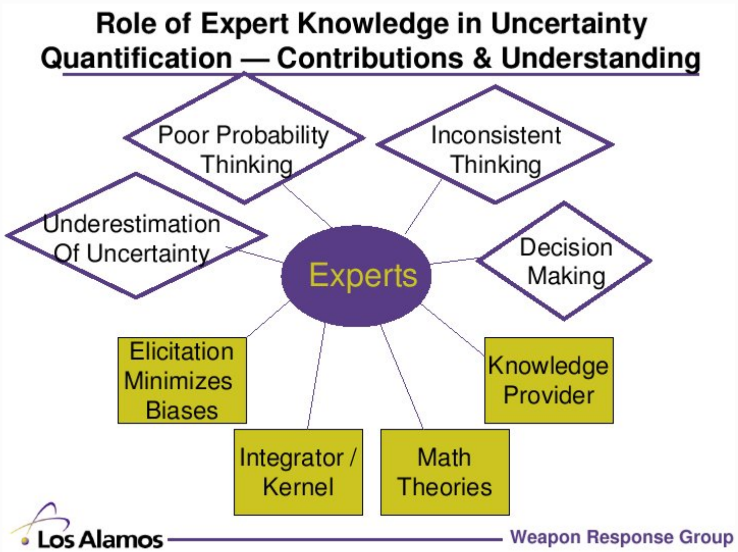
Dempster–Schafer modell Kondicionálás

Feltételes Bel és Pl függvények

- Amennyiben megfigyelünk egy eseményt, a halmazokhoz rendelt súly változik:
  - Ha C igaz, akkor csak a C-t tartalmazó halmazok maradnak, melyekből a C-t töröljük, mint véletlen eseményt:
    1. a súlyokat újraszámoljuk;
    2. normalizáljuk a rendszert.
- Egy más „univerzum” keletkezik.
- A C nem teljesül, akkor csak C-t nem tartalmazó halmazokat használjuk.

Dempster–Schafer modell alkalmazása

Használják: ...



www.c3.lanl.gov/~joslyn/

Dempster–Schafer modell Összefoglaló

- Konzisztens keretrendszer, mellyel lehet számításokat végezni;
- A $\mathrm{Bel}(\cdot)$ és $\mathrm{Pl}(\cdot)$ függvények definíciója „természetes”;
- Könnyű bizonytalanságot rendelni az eseményekhez – ignorancia;
- Megfigyelt események kezelése – kondicionálás –nagyon nehéz, újra kell számolni a függvények értelmezési tartományát.

Döntések bizonytalan helyzetekben – a D.S. rendszerhez hasonlóan akkor, amikor a valószínűség szabályai nem alkalmazhatóak.

Tipikusan:

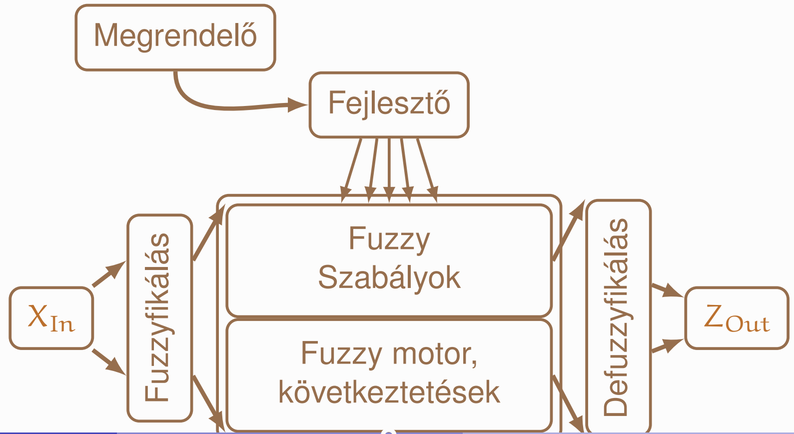
„Tények” gyorsan hajtott, magas ember, kb. 180 cm, olajos ruha, . . .

„Szabályok” ha gyorsan hajt és lassan lélegzik akkor lassítson az elektronika . . .

Szükség van egy **keretrendszerre**, mely a fentieket képes **programozni**:

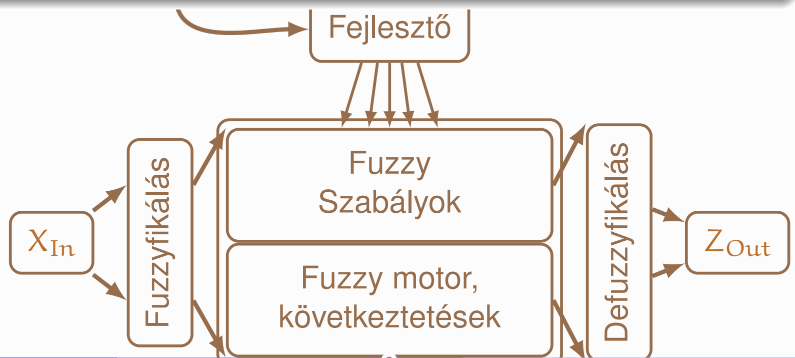
1. Szabályokat tudunk megadni;
2. Információt – adatokat – tudunk bevinni;
3. Következtetéseket tudunk levonni ⇒ döntéshozatal;

Egy fuzzy rendszer építésének logikai sémája:



Egy fuzzy rendszer építésének logikai sémája:

- A megrendelő közli a fejlesztővel a
  - változókat
  - szabályokat
- a fejlesztő a szabályokat „átírja” a fuzzy logika szerint;
- a szabályok köré építi a „következtetőt”, ami a rendszert eredményezi.



Egy fuzzy rendszer működéséhez

Szükségünk van:

- Keretrendszerre, mely a szabályokat értelmezni tudja;
- „Fuzzyfikáló” modulra, mely a megfigyeléseket átalakítja a fuzzy logika nyelvére;
- Következtető modulra;
- „De-fuzzyfikáló” modulra.

Eszköz

- Fuzzy halmazelmélet;
- Fuzzy logika.

Történelmi visszatekintő

Fuzzy: homályos, zavaros, bizonytalan, kócos

’65 – Zadeh: „Fuzzy Sets”;
’70 – Fuzzy elmélet robotikai alkalmazása;
’75 – Japán ...
’80–’95 – Empirikus vizsgálatok és széles körű alkalmazások;
’00–’05 – Technológiai standard,

Lotfi A. Zadeh

Felhasználás:
Ha magas a hőmérséklet, akkor a nyomás is magas;

Fuzzy halmazok I

Fuzzy halmazelmélet

- Eszköz annak a leírására, hogy egy objektum milyen mértékben illeszkedik egy bizonytalan tényhez;
- Alkalmas bizonytalan kijelentések formalizálására.
- **Nem valószínűségi** modell.
- Döntések hozatala „igazság-értékek” alapján.
- **Fuzzy halmazok**: nem „teljes” hozzátartozás specifikálása.

Fuzzy halmazok II

Klasszikus logika
Egy kijelentés igaz: – 1; vagy hamis – 0.

Egy A halmazra:

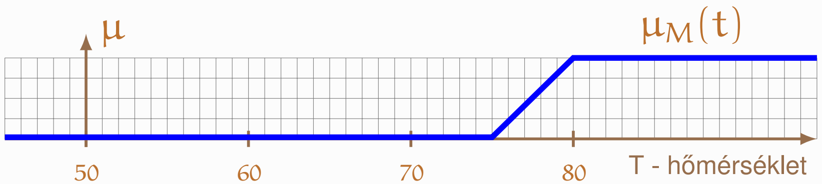
$$I_A(x) = \begin{cases} 1 & \text{ha } x \in A \\ 0 & \text{ha } x \notin A \end{cases}$$

Fuzzy logika
Egy kijelentés igazságértéke $\mu(p) \in [0,1]$

$$\mu_A(x) \in [0,1] \qquad \forall x \in \Omega$$

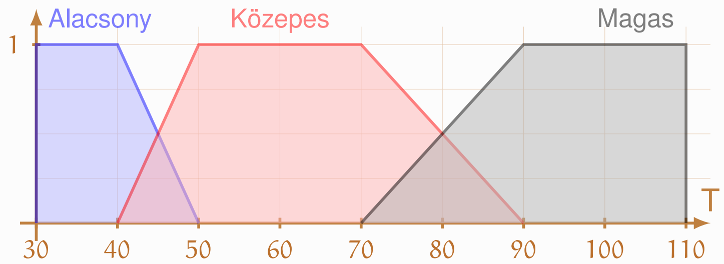
Egy halmaz= μ_A

Például: M – magas hőmérséklet



Fuzzy halmazok III

A definíciós intervallumot általában felosztjuk több fuzzy halmazra:



Egy fuzzy halmaz tehát **függvény**, $\mu_K : \Omega \to [0,1]$:

$$\mu_K(x) = \begin{cases} 1 & \text{ha } x \in [50,70] \\ (x-40)/10 & \text{ha } x \in [40,50] \\ (90-x)/20 & \text{ha } x \in [70,90] \\ 0 & \text{ha } x \leq 40 \text{ vagy } x \geq 90 \end{cases}$$

Fuzzy logika 0

- A logikai műveletek halmaz–műveletekre vezethetők vissza:
  - a logikai **vagy** az egyesítés:
  $$A \vee B \overset{\text{def}}{=} A \cup B$$
  - a logikai **és** a metszet:
  $$A \wedge B \overset{\text{def}}{=} A \cap B$$
  - a logikai **negáció** a komplementer:
  $$\overline{A} \overset{\text{def}}{=} \Omega \setminus A$$
- A fuzzy logikai műveletek fuzzy halmazok egyesítése, metszete, illetve komplementere.
- A három logikai művelet redundáns:
$$\overline{A \vee B} = \overline{A} \wedge \overline{B} \quad \Rightarrow \quad \boxed{A \vee B = \overline{\overline{A} \wedge \overline{B}}}$$
- Csak a konjunkciót és a negációt kell specifikálnunk.

Fuzzy logika I

- A fuzzy rendszerekben a konjunkciót és a negációt specifikáljuk.
  - A konjunkció egy **T-norma**:
  $$T : [0,1] \times [0,1] \rightarrow [0,1]$$
  A $T(1,1) = 1$ kivételével minden sarokpontban az értéke 0 kell, hogy legyen (hogy kapjuk vissza a klasszikus logikát).
  - A negáció egy-argumentumú függvény:
  $C : [0,1] \rightarrow [0,1]$
  A sarokfeltételek: $C(0) = 1$, illetve $C(1) = 0$.
- A diszjunkciót a De-Morgan reláció alapján számoljuk ki (S-konormának nevezzük):
$$S : [0,1] \times [0,1] \rightarrow [0,1]$$
$$\boxed{S(a,b) = C\left(\, T\left(C(a), C(b)\right)\,\right)}$$

Fuzzy logika II

Példa:

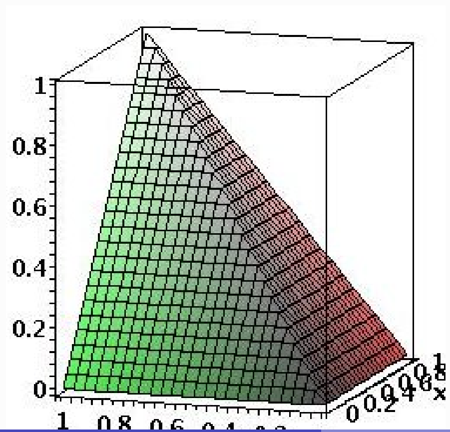
- A konjunkciónak megfelelő T-norma:
$$T(a,b) = \min(a,b)$$
$T(0,0) = T(0,1) = T(1,0) = 0$, illetve $T(1,1) = 1$.
- A negáció legyen:
$$C(x) = 1 - x$$
$C(1) = 0$, illetve $C(0) = 1$.
- A diszjunkció meghatározása:
$$S(a,b) = 1 - T(1-a, 1-b)$$

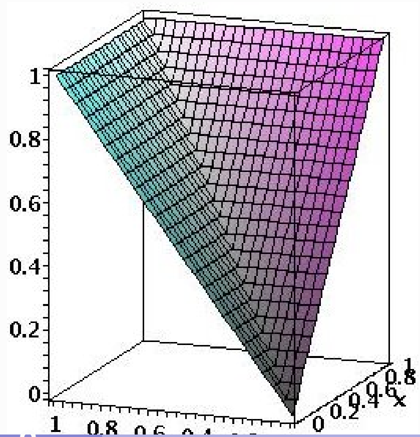
$$\begin{aligned} S(a,b) &= 1 - T(1-a, 1-b) = 1 - \min(1-a, 1-b) \\ &= 1 + \max(-1+a, -1+b) = \max(1-1+a, 1-1+b) \\ &= \max(a,b) \end{aligned}$$

Fuzzy logika III

Logikai szabályok:

$$\begin{aligned} \mu_{A \wedge B}(x) &= \min\left[\mu_A(x), \mu_B(x)\right] \\ \mu_{A \vee B}(x) &= \max\left[\mu_A(x), \mu_B(x)\right] \\ \mu_{\overline{A}}(x) &= 1 - \mu_A(x) \end{aligned}$$





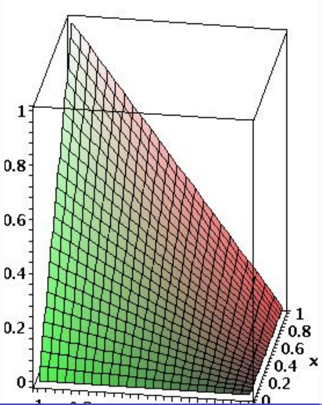
Fuzzy logika IV

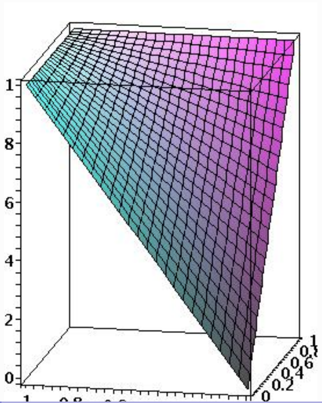
Logikai szabályok: $a, b \in \{0, 1\}$

ÉS = 1 csak ha $a = 1$ és $b = 1$
VAGY = 0 csak ha $a = 0$ és $b = 0$

Alternatív formák:

$$\mu_{A \wedge B}(x) = \mu_A(x) \cdot \mu_B(x)\ , \quad \mu_{\overline{A}}(x) = 1 - \mu_A(x)$$
$$\mu_{A \vee B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$





Fuzzy logika - ! -

Fuzzy logika inkonzisztens

x vagy A-nak vagy $\overline{A}$-nak eleme, azaz $\mu_{A \cap \overline{A}} = 1$.

$$\mu_{A \vee \overline{A}}(x) = \begin{cases} 1 & \text{klasszikusan} \\ \max\left[\mu_A(x), 1 - \mu_A(x)\right] & \text{fuzzy logika 1 def.} \\ 1 - \mu_A(x)\left[1 - \mu_A(x)\right] & \text{fuzzy logika 2 def.} \end{cases}$$

Ugyanígy: szeretnénk, hogy $\mu_{A \wedge \overline{A}} = 0$. Helyette:

$$\mu_{A \wedge \overline{A}}(x) = \begin{cases} 0 & \text{klasszikusan} \\ \min\left[\mu_A(x), 1 - \mu_A(x)\right] & \text{fuzzy logika 1 def.} \\ \mu_A(x)\left[1 - \mu_A(x)\right] & \text{fuzzy logika 2 def.} \end{cases}$$

vizsga feladattípus: határozzuk meg a „modus ponens" szabály fuzzy változatát.

Fuzzy logika Feladat I

A negációt szeretnénk úgy meghatározni, hogy

- legyen folytonos, bijektív;
- teljesüljön a negált negáltja önmaga, azaz

$$C(C(x)) = x$$

- legyen $C(0) = 1$ és $C(1) = 0$

A $C(x)$ bijektív, alkalmazva a $z = C(x) \Leftrightarrow x = C^{-1}(x)$:

$$C(z) = C^{-1}(z)$$

azaz a $C(\cdot)$ függvény a főátló szerint szimmetrikus.

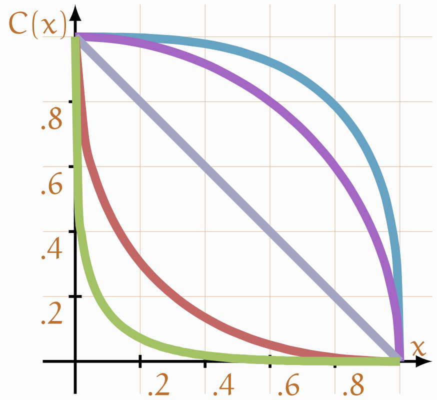
A $C(x) = 1 - x$ **egy** megoldás.

Egy megoldás-család – pozitív q értékekre – a

$$C_q(x) = \sqrt[q]{1 - x^q}$$

Fuzzy logika Feladat II

A $C_q(x) = \sqrt[q]{1 - x^q}$ **grafikus képei** a $q = \{1/3, 1/2, 1, 2, 3\}$ értékekre:



Fuzzy szabályok 0

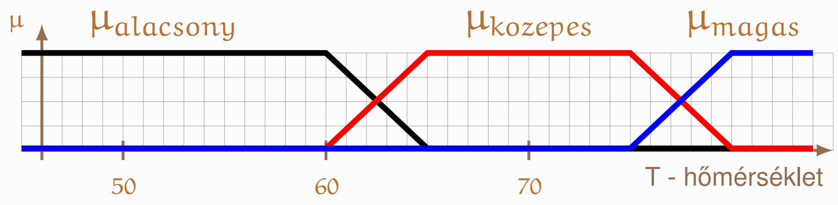
- **Érthető formában** írják le egy rendszer működését:
  - Könnyű megfogalmazni;
  - Könnyen ellenőrizhető;
  - Módosításokhoz nem kell az egész rendszert átírni.
- Használják a fuzzy logika következtetési mechanizmusait;
  - Logikai szabályok alapján;
  - Fuzzy változók értékeit számolják;
  - De-fuzzyfikálás után az eredmény jó kontroll-változó.

Fuzzy szabályok I

Példa:
Ha **magas** a hőmérséklet, akkor a nyomás is **magas**;

Fuzzy szabályok:
HA $\Big[$ X alacsony ÉS Y magas $\Big]$ AKKOR Z **közepes**

Definiáljuk az alacsony, magas és közepes halmazokat:



Fuzzy szabályok II

Fuzzy szabályok:

X	Y		
	Alacsony	**Közepes**	**Magas**
Alacsony	–	Közepes	Alacsony
Közepes	–	Magas	Közepes
Magas	Közepes	Magas	Alacsony

– nem megengedett állapot

Szabályok alkalmazása:
Megfigyelt értékek: $x = 50,\ y = 78.75$

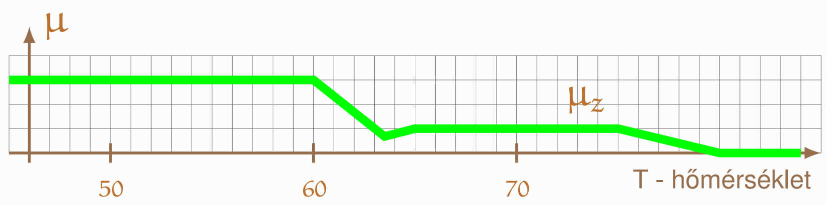
Fuzzyfikálás – fuzzy halmazokra való áttérés: X szerint csak az **Alacsony** csoportot kell vizsgálni.
Y szerint a **Közepes** és **Magas** csoportokat.

Fuzzy szabályok III

$\mu_{al}(x) = 1$ illetve

$$\mu_{mag}(y) = 0.75;$$
$$\mu_{koz}(y) = 0.25;$$

Kombinálás: VAGY$=\max$



Kontroll esetén
Egy érték kiírása: súlyozott közép $Z = 53$.

Fuzzy rendszer – I – Alkalmazás

Mitsubishi légkondicionáló rendszer Ipari légkondicionáló mely rugalmasan reagál a környezeti változásokra.

Megvalósítás:

- 50 fuzzy szabály
- 6 nyelvi változó, közöttük a szoba és a fal hőmérséklete, illetve ezek időbeli változásai;
- 4 nap – prototípus; 20 nap – teszt/integrálás; 80 nap – tesztelés; $\Rightarrow$ mikrokontroller implementálás.
- Előnyök: kevesebb szenzor, változó környezet, 24%-kal kisebb fogyasztás;

Fuzzy rendszer – II – Alkalmazás

Alkalmazások:

- Erőművek kontrollja (Tokio);
- Egyszerűsített robot-kontroll (Fuji, Toshiba, Omron);
- Fényképezőgép automata irányítása (Omron);
- Motor-fogyasztás, irányítás (Nissan,Subaru);
- Gyártósorok szilícium-lapkáinak vágása (Canon);
- Rák diagnózis (Kawasaki Med. School);
- Jogi eljárások szimulációja (Meihi Gakuin Univ, Nagoya Univ.)
- . . . Matsushita, Sony, Canon, Minolta, Sanyo, Hitachi, Ricoh, Fujitech, Toshiba, ...

http://www.esru.strath.ac.uk/Reference/concepts/fuzzy/fuzzy_appl.00.htm

Fuzzy rendszerek Összefoglaló

Fuzzy rendszerek:

- Főként kontroll feladatokra használjuk.
- **Ajánlottak ha:**
  - komplex rendszerek esetén, nincs matematikai modell;
  - nemlineáris modellek esetén;
  - szakértői tudás bevitelére.
- **Nem ajánlottak ha:**
  - hagyományos kontroll megfelelő;
  - létező – egyszerű – matematikai modell;
  - nincs megoldás.

Más alkalmazások:

Sony PalmTop – „fuzzy” döntési fa implementálása karakterfelismerésre (Kanji).

Fellendülőben az MI

Újabb MI-divat? 2007.09.12.

Az MI célja, hogy a gépek **emberre jellemző** tulajdonságokkal rendelkezzenek: nyelvhasználat, **fogalomalkotás, elvonatkoztatás**, nem triviális problémák megoldása, öntökéletesítés. (McCarthy, Minsky, Shannon)

MI a mindennapokban

„A kifejezetten mérnöki munka és alkalmazott tudományok terén elért eredményeket figyelembe véve, a gépi tanulás technikáinak különböző problémákra (webes keresés, pénzügyek, molekuláris rendszerek megértése) történt alkalmazása a legsikeresebb MI–részterület.” – Terry Winograd szerint. „Ezek az eredmények, és más területek, például a robotika fejlődése intelligensebb gépekhez vezetnek.”

http://www.agent.ai Winograd home

Grafikus modellek I

Grafikus modellek
(probabilistic graphical models - PGM) *Kevin Murphy*

A gráfelmélet és a valószínűségszámítás ötvözése.

- gráf, mely egy rendszer függőségeit ábrázolja;
- csúcsokhoz rendelt függvények, melyek a változók közötti kapcsolatot írják le;

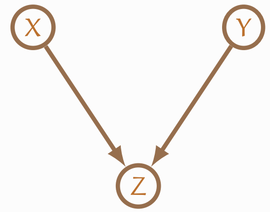
„Minden grafikus modell” szemléletmód [5]

- Bayes–háló – irányított aciklikus gráfú PGM;
- Markov modell – irányítatlan PGM.

[5] „minden feladat szeg”, ha egy kalapácsot tartunk a kezünkben …

Grafikus modellek (GM) II

Irányított GM

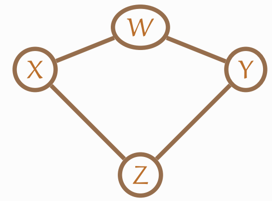


- feltételes függetlenség;
- kauzális relációk;

$$X \perp Y$$

(láttuk)

Irányítatlan GM



- feltételes függetlenség;
- szimmetrikus;

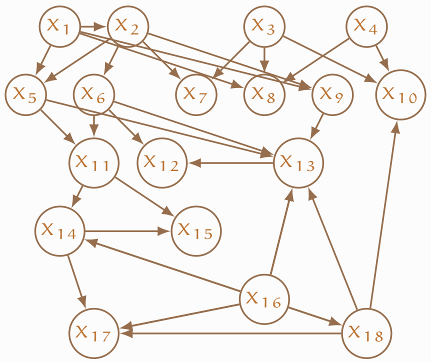
$$X \perp Y \mid \{W, Z\}$$
$$W \perp Z \mid \{X, Y\}$$

GM feladattípusok I

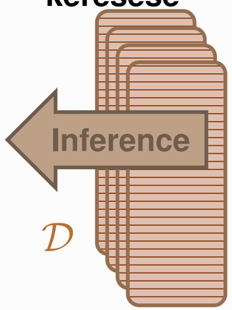
Feladatok – adott GM struktúrára:

- események (felt.) val.-nek kiszámítása;
- állapítsuk meg két csúcs (feltételes) függetlenségét;
- adatok alapján építsünk optimális gráfot (indukció).

Inferencia $P(X_{13}|X_7, X_8, X_9, X_{16})$



Optimális struktúra keresése



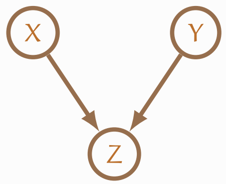
Irányított GM I

Bayes–labda algoritmus: feltételes függetlenség–teszt.

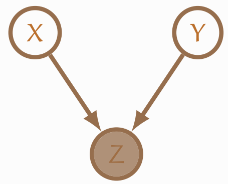
d-szeparáltság

Egy grafikus modellben X az Y-tól **d-szeparált** a Z ismeretében, ha minden X-et és Y-t összekötő irányítatlan úton létezik w úgy, hogy:

- vagy $\rightarrow w \leftarrow$ és a w vagy utódai nincsenek a Z-ben;
- vagy w nem $\rightarrow w \leftarrow$ és $w \in Z$.

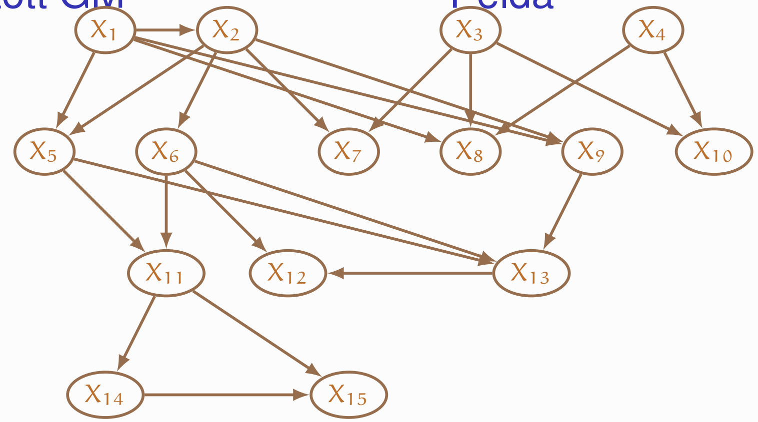


$$X \perp Y$$



$$X \not\perp Y$$

Irányított GM — Példa

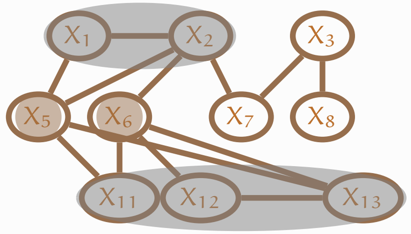


Kérdések

? $X_1 \perp X_{10} \mid \{X_7, X_8, X_9\}$	nem
? $X_1 \perp X_{10} \mid \emptyset$	igen
? $X_1 \perp X_{15} \mid \{X_5, X_6, X_{13}\}$	igen

Irányítatlan GM — I

Különbség: a gráf éleinek nincs iránya; szimmetrikus kapcsolat–modellezés.



?Paraméterezés:

- **felt. val.-gek**: nem jók;
- **? – Nem konzisztens**

$$\prod P(X_i|\mathrm{Szomszed}_i)$$

Függetlenség: gráfon. Pl:

$$\{X_1, X_2\} \perp \{X_{11}, X_{12}, X_{13}\} | \{X_5, X_6\}$$

- **Helyette**

$$\prod \Phi(X_i, \mathrm{Szomszed}_i)$$

klikkek jellemzése.

Irányítatlan GM — II

Klikkek: teljesen összekötött csúcsok.

$$P(\mathbf{X}) = \frac{1}{Z} \prod_{c \in \mathrm{Klikkek}} \phi_c(\mathbf{x}_c) \qquad Z = \sum_{\mathbf{X}} \prod_{c \in \mathrm{Klikkek}} \phi_c(\mathbf{x}_c)$$

ahol

$\phi_c(\mathbf{x}_c) \geq 0$	potenciál-függvény
Z	partíciós függvény

Gibbs–eloszlás

Egy eloszlás, mely ábrázolható egy H gráffal és $\{\phi_c\}_c$ potenciálfüggvényekkel, H fölötti Gibbs–eloszlásnak nevezünk.

Irányítatlan GM — III

A gráf alapján $X \perp Z|Y$

$$P(X, Y, Z) = P(X|Y)P(Y, Z) = \phi_{xy}(X, Y)\phi_{yz}(Y, Z)$$
$$P(X, Y, Z) = P(Z|Y)P(X, Y) = \phi_{yz}(Y, Z)\phi_{xy}(X, Y)$$

ϕ_c **– potenciálfüggvények**

- „kompatibilitást”, „összeférhetőséget” jelölnek;
- nem eloszlásfüggvények.

Irányítatlan GM IV

Exponenciális eloszláscsalád:

$$\begin{aligned} P(\mathbf{X}) &= \frac{1}{Z} \prod_{c \in \text{Klikkek}} \phi_c(\mathbf{x}_c) \\ &= \exp\left[-\log Z(\boldsymbol{\theta}) + \sum_{i \in I} f_i(\mathbf{x}_{c_i}) \right] \end{aligned}$$

ahol

$f_i \stackrel{\text{def}}{=} \log \phi_{c_i}$ $\quad$ I – index-halmaz

Példa: Gaussz-eloszlás (teljes kapcsolat)

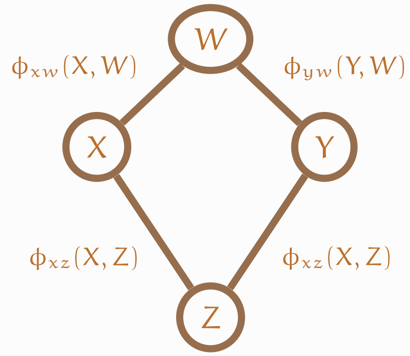
$$P(\mathbf{X}) = \exp\left[-\frac{d \log(2\pi) + \log|\Sigma| + \mathbf{X}^T \Sigma^{-1} \mathbf{X}}{2} \right]$$

Irányítatlan GM Feladat

$X, Y, Z, W \in \{0, 1\}$

Ismerjük:

$[\phi_{xw}, \phi_{yw}, \phi_{xz}, \phi_{yz}]$



Számítsuk ki: $p(Y = 1 | W = 1)$

Milyen esetekben jók/nem jók a GM-ek ?

Osztályozás – USPS I

http://www.cs.ubbcluj.ro/~csatol/mestint/USPS

Osztályozás – USPS II

http://www.cs.ubbcluj.ro/~csatol/mestint/USPS

USPS adatok

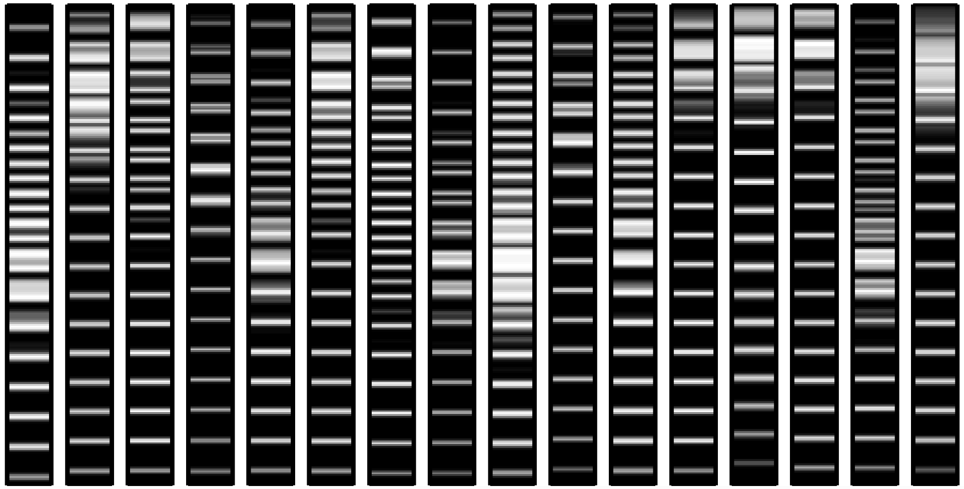
- United States Postal Service;
- kézzel írott számjegyek 16×16-os bit-térképe;
- ≈ 7200 tanulási- és ≈ 2000 teszt-adat;

Feladat:

- Bináris osztályozás: válasszuk el a négyeseket (4) a hetesektől (7).
- Írjunk egy osztályozási algoritmust a következők közül: Döntési fa; Neurális háló; SVM; Bayes osztályozó.
- Értékeljük az algoritmust a következők szerint:
  - osztályozási hiba tanulási adatokon;
  - osztályozási hiba teszt-adatokon – 10% alatt!;
  - tanulási idő;
  - vizsgáljuk meg a rosszul osztályozott „pontokat”;

Osztályozás III

http://www.cs.ubbcluj.ro/~csatol/mestint/USPS

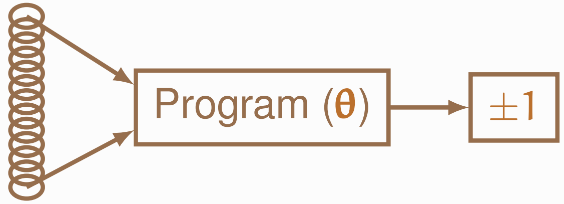


4 7 7 4 4 7 7 4 4 4 4 7 7 7 4 7

-1 $+1$ $+1$ -1 -1 $+1$ $+1$ -1 -1 -1 -1 $+1$ $+1$ $+1$ -1 $+1$

Osztályozás IV

Osztályozó algoritmus:



Tanulás:

- θ_0 kezdőértékek
- tanulási adatok

$$\mathcal{D} = \left\{ (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \right\}$$

- Becslés: $\theta_1 \rightarrow \ldots \rightarrow$
- θ_{post}.

Közben: tanulási hiba mérése.

Tesztelés:

- θ_{post} használata;
- új adatokra:

$$\mathcal{D}_{test} = \ldots$$

- hiba mérése tesztadatokon.

Osztályozás V

Program + dokumentáció: **(dolg...)**

- Algoritmus kiválasztása;
- Tanulás – paraméterek becslése;
  - tanulási adatok beolvasása;
  - adatok transzformációja – pre–processzálás;
  - algoritmus paramétereinek inicializálása;
  - paraméterek tanulása.
- Tesztelés;
  - teszt-adatok beolvasása – **függetlenek a tanulási mintáktól!**;
  - adatok transzformációja – pre–processzálás;
  - algoritmus paramétereinek beolvasása – tanulási eredmény;
  - hiba mérése.
- Dokumentálás: algoritmus, módszer, statisztika

Osztályozás Program

binclass.m

```
clear all;
% data generation
nTr = 200; nTe = 400;
[dX dY] = cl_data(nTr,[],0);
ii_1  = find(dY==1);
d1 = dX(ii_1,:);
d2 = dX(setdiff([1:nTr],ii_1),:);
N_1   = size(d1,1);
N_2   = size(d2,1);

pi1 =N_1/nTr; pi2 = N_2/nTr;

mu1 = sum(d1,1)/N_1; % class means
mu2 = sum(d2,1)/N_2;

d = d1 - repmat(mu1,[N_1 1]);
sig1 = d'*d/N_1; sig1inv = sig1^(-1);

d = d2 - repmat(mu2,[N_2 1]);
sig2 = d'*d/N_2; sig2inv = sig2^(-1);

% generating nTe test data
[dX, dY] = cl_data(nTe,[],0);
ii_1  = find(dY==1);
data1 = dX(ii_1,:);
data2 = dX(setdiff([1:nTe],ii_1),:);
N_1   = size(data1,1);
N_2   = size(data2,1);
```

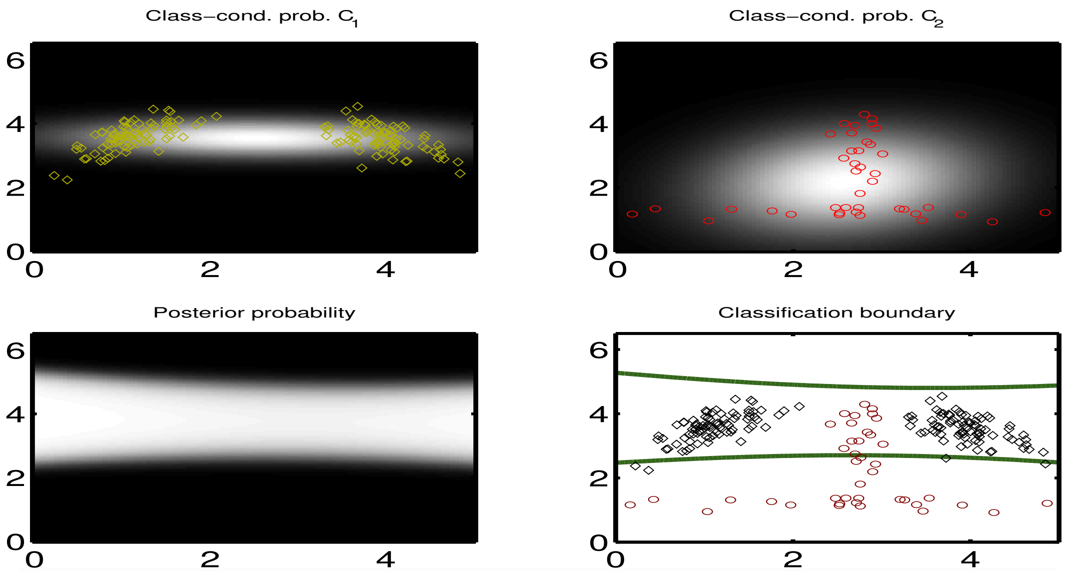
. . . grafikus ábrázolás. . .

```
% performing classification
log_sig1 = log(det(sig1))-log(pi1);
log_sig2 = log(det(sig2))-log(pi2);

% classification for data1
p_1=data1-repmat(mu1,[length(data1) 1]);
p_1=sum(((p_1*sig1inv).*p_1),2);

p_2=data1-repmat(mu2,[length(data1) 1]);
p_2=sum(((p_2*sig2inv).*p_2),2);

class(1,1) = length( ...
   find(-p_1+p_2 > log_sig1-log_sig2) );
class(1,2) = N_1 - class(1,1);

% SIMILARLY for the second dataset
p_1=data2-repmat(mu2,[length(data2) 1]);
p_1=sum(((p_1*sig2inv).*p_1),2);

p_2=data2-repmat(mu1,[length(data2) 1]);
p_2=sum(((p_2*sig1inv).*p_2),2);

class(2,2) = length( ...
   find( -p_1+p_2 > log_sig2-log_sig1));
class(2,1) = N_2 - class(2,2);

% the confusion matrix:
class
```

Osztályozás Eredmény

binclass.m

Eredmény:



Olcsóbb gépi fordítás METIS II

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 gyűjtemény lehet egyetlen (monolingual) vagy két (több) nyelvű (multilingual). A rendszer így tanulja meg, hogy egy nyelv szavai és kifejezései miként kapcsolódnak egy másik nyelv kifejezéseihez és szavaihoz.

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

www.agent.ai ICT project

„Tanuló" rendszerek

pp. 525 – Russell & Norvig, 1995

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

- 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.

Indukció – I – Dedukció

Induktív megoldás

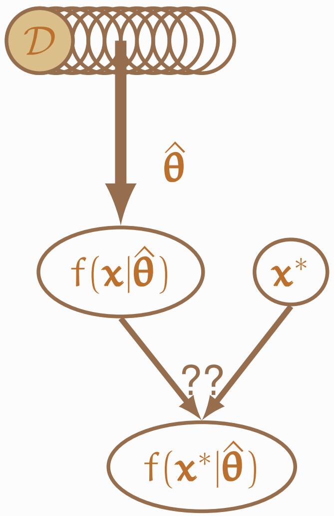
- Modell feltételezése: $f(\mathbf{z}, \boldsymbol{\theta})$,
- Adatok halmaza: $\mathcal{D} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$
- Illesztő – hiba – függvény: $L\,(f(\mathbf{z}))$
- „optimális modell": $\boldsymbol{\theta}^*$
- Predikció: $f(\mathbf{z}, \boldsymbol{\theta}^*)$.

Deduktív megoldás

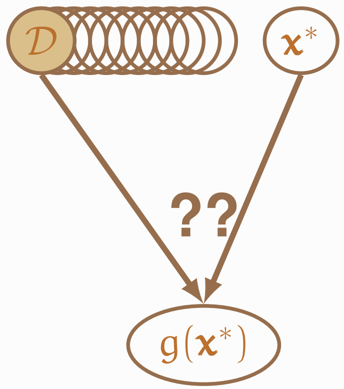
- Nincs modell,
- Adatok: $\mathcal{D} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$
- Illesztési művelet egy $\mathbf{z}$ új mintára,
- használunk minden adatot.

Indukció – II – Dedukció

Induktív megoldás



Deduktív megoldás



„Levezetjük” a g függvény $\mathbf{x}^*$-hoz tartozó értékét.

Indukció – III – Dedukció

„Becslés” egy modell ismeretében.

- Modell-osztály;
- Modellek közötti preferencia $\Rightarrow \hat{\theta}$;
- Predikció a modell alapján;
- Pl: 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.

Induktív módszerek I

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, $\sim$ obtain **general domain knowledge**.

Információ induktív rendszereknél

Egy indukció folyamán az adatokból „nyert” információt a modell paraméterei tárolják.
Egy $\mathcal{F} \stackrel{\text{def}}{=} \{f(\mathbf{x}|\theta), \theta \in \Omega\}$ modell esetén a $\mathcal{D}$ adatokat az optimális $\hat{\theta}$ paraméter helyettesíti.

[6] Hogyan definiáljuk az információ fogalmát?

Induktív módszerek II

„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**.

Paraméterek jóságának a mérése I

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)

Paraméterek jóságának a mérése II

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

Regresszió:
$f : \Omega \to \mathbb{R}$
Osztályozás:
$f : \Omega \to \{\mathrm{Igen}, \mathrm{Nem}\}$

Definíció:

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

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

Döntési fák I

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

Logikai bemenő adatok esetén:

- Bináris fa természetes;

Folytonos bemenő adatok esetén:

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

Döntési fa építése

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 (**C**lassification **a**nd **R**egression Trees) algoritmus.

Netlab I

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 II

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.

Nehéz feladatok

Néha NEM lehet a gradiens-szabályokat alkalmazni mert:

- Túl bonyolult a modellben a paraméter–kimenet kapcsolat. Pl.
- A paraméterek nem numerikusak. Pl. szemantikus hálók vagy Bayes-hálók tanulása.

Azonban:

- képesek vagyunk a θ paraméter jóságának a mérésére,
- Van egy „szomszédság-reláció” a paraméterek között,

Szimulált kifűtés I

Hibafüggvény:
Méri az $\mathcal{F}$ modellcsalád egy elemének – az $f(\mathbf{x}|\boldsymbol{\theta})$ függvénynek – a „jóságát".
Például - négyzetes hiba:

$$L(\mathcal{D}, f(\cdot|\boldsymbol{\theta}), \mathcal{F}) = \sum_n (y_n - f(\mathbf{x}_n|\boldsymbol{\theta}))^2$$

Egy $\mathcal{D}$ adathalmaz hibája – **empirikus** hiba

Szomszédság:
A $\boldsymbol{\theta} \in \Omega$ paraméterek (pl. fák) esetén:

$$\boldsymbol{\theta}' \in \delta_\epsilon(\boldsymbol{\theta}) \quad \text{ha } d(\boldsymbol{\theta}, \boldsymbol{\theta}') < \epsilon$$

A paraméter–tér bejárására szolgál $\rightarrow$ összekötött halmazoknál jó.

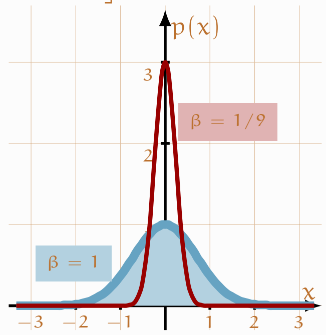
Szimulált kifűtés II

Eloszlás:

$$p(\boldsymbol{\theta}|\beta) \propto \exp\left[-\frac{1}{\beta} L(\mathcal{D}, f(\cdot|\boldsymbol{\theta}), \mathcal{F})\right]$$

$$p(\boldsymbol{\theta}|\beta) = \frac{\exp\left[-\frac{1}{\beta} L(\mathcal{D}, f(\cdot|\boldsymbol{\theta}), \mathcal{F})\right]}{\sum_{\boldsymbol{\theta}\in\Omega} \exp\left[-\frac{1}{\beta} L(\mathcal{D}, f(\cdot|\boldsymbol{\theta}), \mathcal{F})\right]}$$

- Az $f(\mathbf{x}, \hat{\boldsymbol{\theta}})$ modell optimális a legnagyobb $p(\hat{\boldsymbol{\theta}})$–nál;
- β – „hőmérséklet" paraméter;
- β kezdőértéke nagy $\Rightarrow$ nincsenek nagy különbségek;
- $\beta \searrow 0$ - egy állapot választódik ki.



Szimulált kifűtés III

Gyakorlatban nehezen kivitelezhető, mert:

- Nem lehet felsorolni az Ω összes elemét;
- Nem könnyű egy modell hibájának a kiszámítása – gyakran műszeres mérésekről van szó; $\Longrightarrow$

Közelítő módszerek alkalmazása – a „szomszédság" felhasználásával:

$$\boldsymbol{\theta}_0 \rightarrow \boldsymbol{\theta}_1 \rightarrow \cdots \rightarrow \boldsymbol{\theta}_T$$

ahol az iterációk alatt „javul" a $\boldsymbol{\theta}$ illeszkedése az adatokhoz.

Szimulált kifűtés IV

Általában a „hűtést" szabályozzuk: $\{\beta_t\}_1^\infty$.

Algoritmus:

1. Adat-beolvasás: $\mathcal{D}$; $t = 0$;
   Inicializálás: $\boldsymbol{\theta}_0$; Számítjuk: $L(\mathcal{D}, f(\mathbf{x}|\boldsymbol{\theta}_t))$;
2. Választunk a szomszédság alapján: $\boldsymbol{\theta}' \in \delta_\epsilon(\boldsymbol{\theta}_t)$
   Számítjuk: $L(\mathcal{D}, f(\mathbf{x}|\boldsymbol{\theta}')) \quad \rightarrow p(\boldsymbol{\theta}'|\beta_t)$;
3. Ha $p(\boldsymbol{\theta}'|\beta_t) > p(\boldsymbol{\theta}_t|\beta_t)$, akkor $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}'$,
4. ellenkező esetben – $a \in [0, 1]$ véletlen szám – ha $a \le p(\boldsymbol{\theta}'|\beta_t)/p(\boldsymbol{\theta}_t|\beta_t)$ akkor $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}'$
   másképp $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$
5. $t = t + 1$ Goto 2

Szimulált kifűtés — Összefoglaló

Használható esetekben, ahol

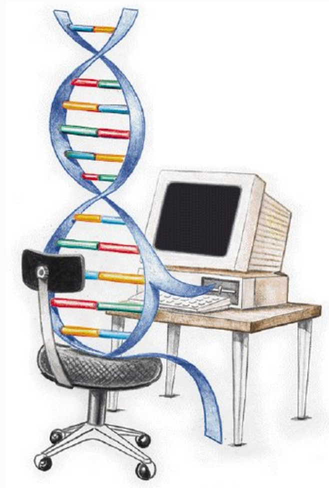
- van hibafüggvény;
- nem folytonos a paraméterhalmaz;
- nincs gradiens információ – vagy nem használható;

Például: VLSI-design.

$\sim$ **használata**

- könnyű – nem igényel sok kódolást vagy tanulmányozást;
- nem hatékony – a futási idő nagyon nagy – egyszerű feladatoknál is.

Genetikus algoritmusok — 0



John R. Koza - Stanford University

http://www.genetic-programming.com/coursemainpage.html

Genetikus algoritmusok — I

A szimulált kifűtés továbbfejlesztése:

1. Nem egyetlen paramétert optimizál;

   populáció $\forall t \{\boldsymbol{\theta}_k^{(t)}\}_{k=1}^K$

2. Definiálja a genetikus operátorokat.

   kereszteződés $cr(\boldsymbol{\theta}_k^{(t)}, \boldsymbol{\theta}_l^{(t)}) = \boldsymbol{\theta}_m^{(t+1)}$

   mutáció $m(\boldsymbol{\theta}_k^{(t)}) = \boldsymbol{\theta}_m^{(t+1)}$

   az operátorok az egyedek szintjén működnek.

Az iterációs séma hasonló

$$\{\boldsymbol{\theta}_k^{(0)}\}_{k=1}^K \rightarrow \{\boldsymbol{\theta}_k^{(1)}\}_{k=1}^K \rightarrow \cdots \rightarrow \{\boldsymbol{\theta}_k^{(T)}\}_{k=1}^K$$

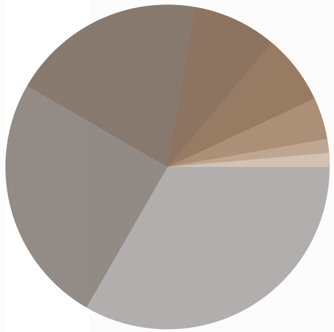
Genetikus algoritmusok — II

Genetikus algoritmusok működése

- „természetes” szelekció;
- genetikus operátorok a paraméter-tér bejárására.

Algoritmus: (minden t időpontban)

$$\{\boldsymbol{\theta}_k^{(t)}\}_{k=1}^K \Leftrightarrow f_k^{(t)} = \exp\left[-\frac{1}{\beta} L(\mathcal{D}, f(\cdot|\boldsymbol{\theta}_k^{(T)}))\right]$$



Következő generáció létrehozása:

1. Az $[f_k^{(t)}]_k$ „fitnessz” faktorral arányosan választunk szülőket: $\boldsymbol{\theta}_l^{(t)}$, $\boldsymbol{\theta}_l^{(t)}$
2. Bemásoljuk azokat a populációba vagy
3. alkalmazzuk a genetikus operátorokat: kereszteződés, mutáció;

G.A. „Fitnessz”

„Fitnessz”-függvény

- méri az egyedek megfelelését – jóságát
- Fordítottan arányos az $L(\cdot)$ hibával

$$\{\boldsymbol{\theta}_k^{(t)}\}_{k=1}^K \Leftrightarrow f_k^{(t)} = \exp\left[-\frac{1}{\beta} L(\mathcal{D}, f(\cdot|\boldsymbol{\theta}_k^{(T)}))\right]$$

- a cél, hogy minél több egyed legyen, melynek nagy a „fitnessz”-értéke;
- β - hőmérséklet-paraméter
  1. a relatív érzékenységet modulálja;
  2. nagy hőmérséklet = különbségek nem jelentősek;
  3. kis hőmérséklet $\beta \searrow 0$ – a leg-„fittebb” egyed sokkal jobb, mint bármely más;

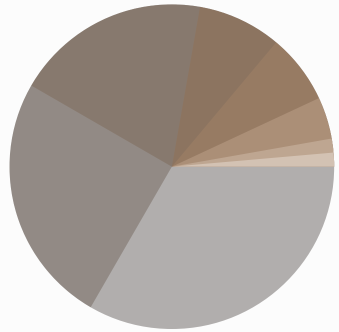
G.A. Szelekció

Szelekció

- garantálja a következő populáció optimalitását;
- csak a kiválasztott – szelektált – egyedek részei a következő generációnak;
- véletlen, de súlyozott választás;

Szelekciós algoritmus:

1. Számoljuk ki: $F(t) = \sum_k f_k^{(t)}$
2. generáljuk $a = \mathrm{rand}(0,1)$
3. $l = 0$, $s = 0$
4. amíg $s < a \cdot F(t)$
   - $l = l + 1$
   - $s = s + f_l^{(t)}$



miért maradnak csak a jobb egyedek?

G.A. Operátorok I

Operátorok — kereszteződés és mutáció

- Kereszteződés
  - a szülők „tulajdonságait” kombinálja;
  - a gyerekek „közel” lesznek a szülőkhöz;

Kereszteződés

„születnek” egyedek, melyek halmozzák a jó tulajdonságokat;

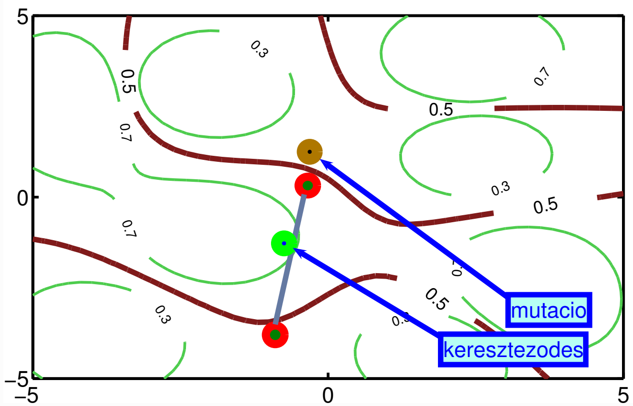
- Mutáció
  - szelekció nyomán csak a jó egyedek maradnak fent;
  - ez a genetikus operátor biztosítja a változatosságot;
  - olyan értékek is kiválasztódnak, melyek nem voltak a korábbi generációkban;

Mutáció

„radikális” változás, mely esetenként jobb eredményhez vezet;

G.A. Operátorok II

Genetikus operátorok: – folyamatos paramétertér

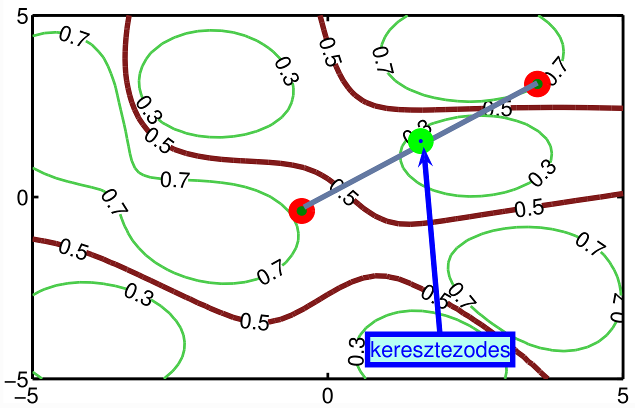


Mutáció: ha $\boldsymbol{\theta} \in \mathbb{R}^m$ $\quad m(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \boldsymbol{\theta} + \delta$

Kereszteződés: $\quad cr(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \stackrel{\text{def}}{=} (\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2)/2$

G.A. Operátorok III

Kereszteződés / mutáció : ! heurisztikák !



Heurisztika: Nem mindig működik;

Jobb eredményeket remélünk, garancia nélkül.

G.A. TSP I

- Genetikus algoritmusok ⇔ az operátorok **definíciója**;
- **Numerikus esetek** kevésbé érdekesek;

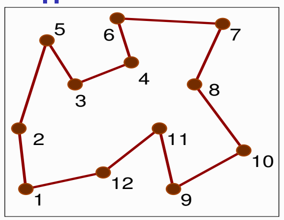
Példa: Utazóügynök (Travelling SalesPerson – TSP)

- Városokat bejárni úgy, hogy a **megtett út** minimális legyen.
- **Egzakt megoldás:** nagyon időigényes; NP-teljes feladat.
- **Genetikus algoritmus:**
  - nem garantálja a legrövidebb út megtalálását;
  - azonban egy közel-optimális utat gyorsan megtalál;

G.A. TSP II

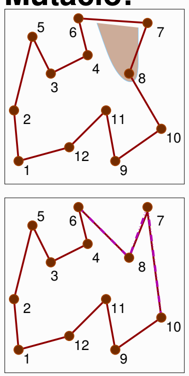
Kódolás: az utak egy permutációja.

Pl. [1, 2, 5, 3, 4, 6, 7, 8, 10, 9, 11, 12]

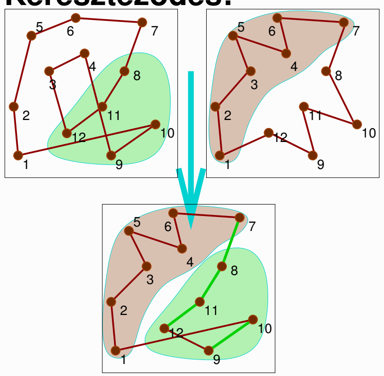


Genetikus operátorok:

Mutáció:

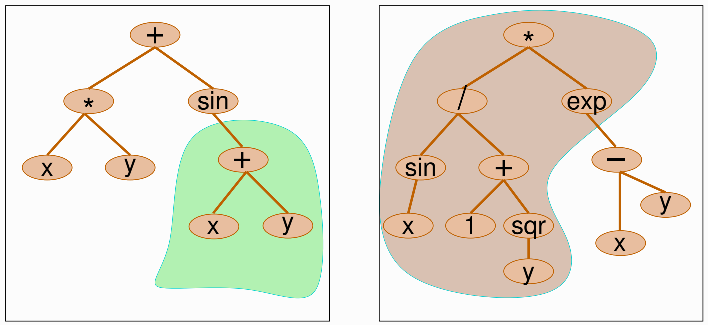


Kereszteződés:



Genetikus programozás

- a populáció egyedei programok;
- programok egy függvényt implementálnak;
- genetikus operátorok a programok terén értelmezettek;
- program = függvény.



Genetikus | Összefoglaló

Genetikus algoritmusok

- populációkkal operálnak;
- szelekciós és reprodukciós operátorok használata;
  - genetikus operátorok gyorsítják a keresést;
  - szükséges a program-/paraméter-tér ismerete.
- a „fitnessz"-érték irányadó ⇔ optimizálás;

„Genetikus" Algoritmus

- általános optimum-keresési módszerek, „szomszédság"-ismerete elégséges;
- ha a fitnessz-függvény differenciálható, numerikus módszerek használata ajánlott;

ET look and disposition | Robotika

„Twendy-One", a *„Wendy"* robot 21.-edik századi reinkarnációja.

A robot képes törékeny tárgyak hordozására valamint emberek segítésére a leülés illetve felállás műveleteiben. Érzékelőin keresztül válaszol az emberi érintésekre.

Képes egy szelet kenyér megfogására, ugyanakkor képes embereket kisegíteni az ágyukból.
„Az első ennyire nagyfokú integrációval rendelkező robot." – mondta Shigeki Sugano, a Waseda Egyetem tanára, a *Twendy-One projekt* vezetője.

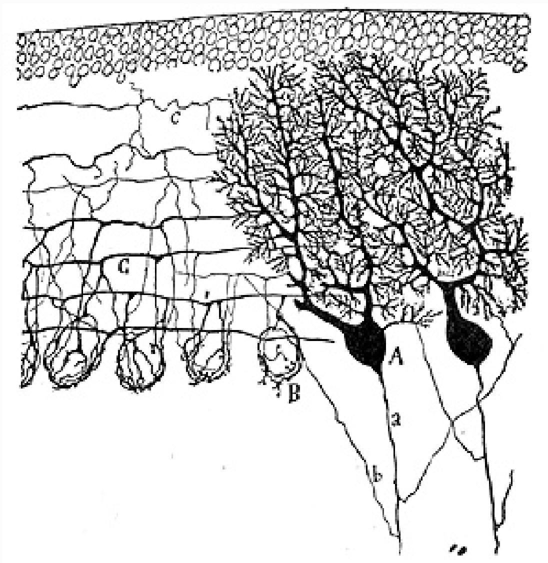
pcWorld hír ABCnews hír

Történeti áttekintő | IdegSejt

Ramon y Cajal

- 1894 – 1900 közötti időszakban;
- idegrendszer vizsgálata;
- építőegység azonosítása: neuron;
- idegsejt – mint az idegrendszer építőeleme

Történeti áttekintő | Neuron



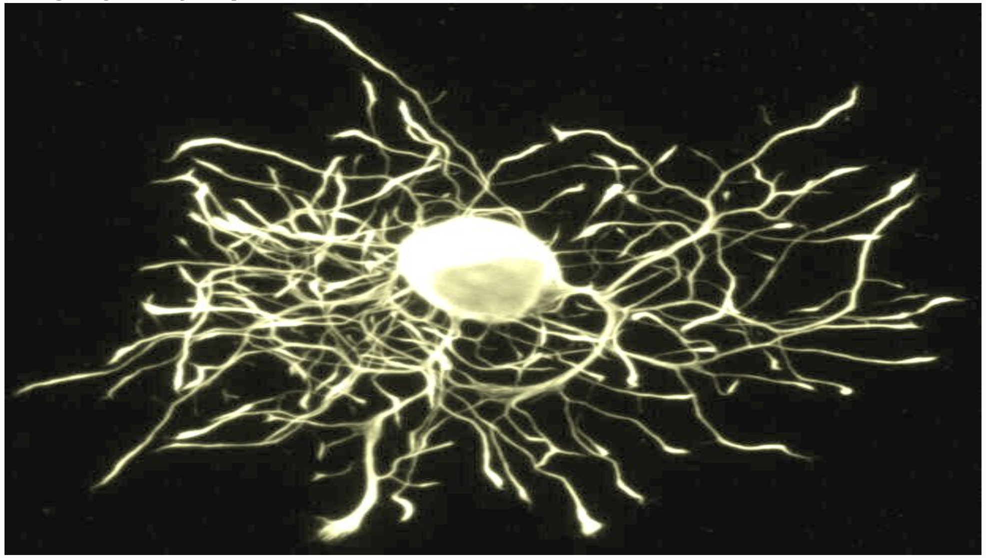
http://nobelprize.org/medicine/articles/cajal/

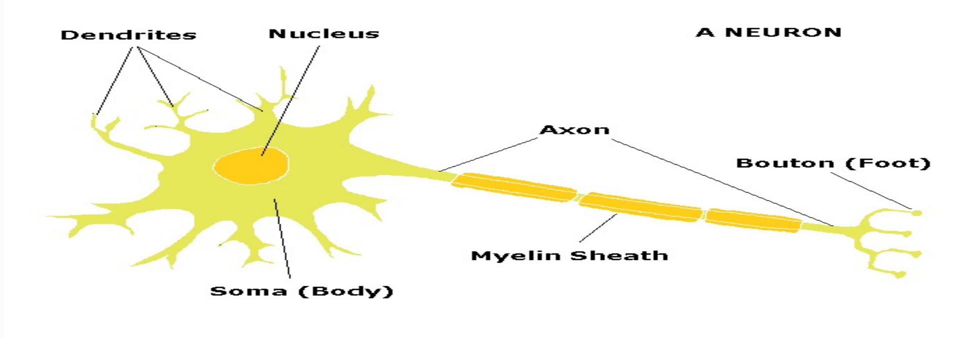
Ramon y Cajal
Nobel díj 1906

The cerebellar cortex (a kitten cerebellum).

The letter A marks the Purkinje cells with dendritic ramifications.

Idegsejt fényképe:





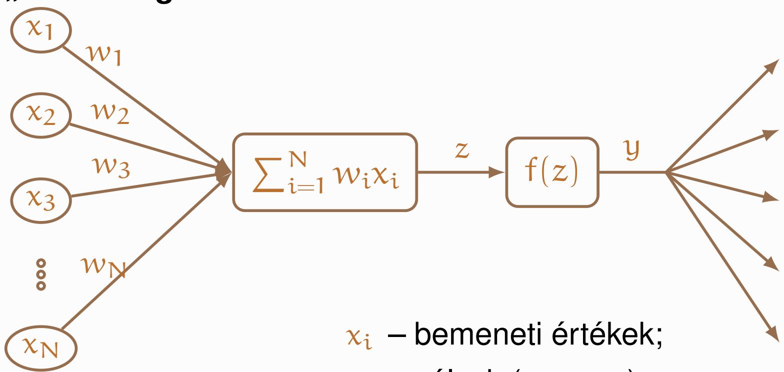
Neuron alkotóelemei

- Szóma (sejtmag) – az információ feldolgozása;
- Dendrit – az információ összegyűjtése;
- Axon – az információ terjesztése;

mi az információ?

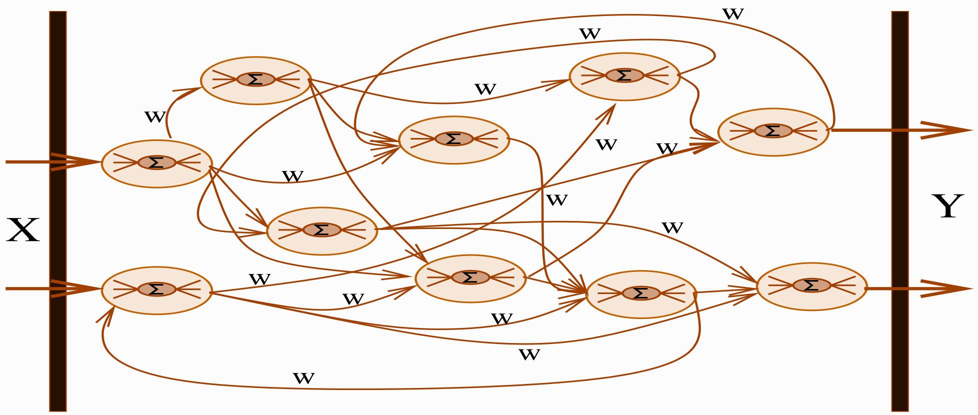
http://www.ship.edu/~cgboeree http://en.wikipedia.org/wiki/Neuron

„Mesterséges" Neuron



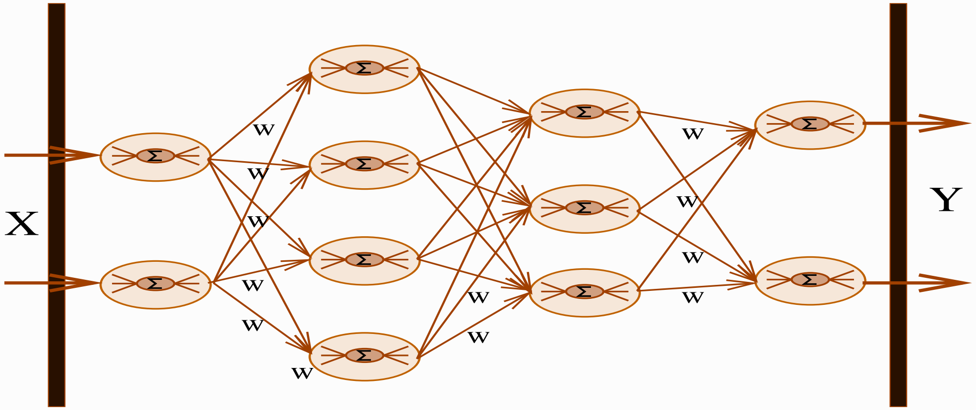
Ahol:

x_i – bemeneti értékek;
w_i – súlyok (dendritek);
$\sum \cdots$ – összegző (szóma);
$f(z)$ – átalakító;
$\longrightarrow$ – kapcsolatok (axonok);



- aszinkron működés;
- nagyon sok kapcsolat;
- ?hogyan? kreáljunk/töröljünk kapcsolatokat;

Történeti áttekintő VI Modell



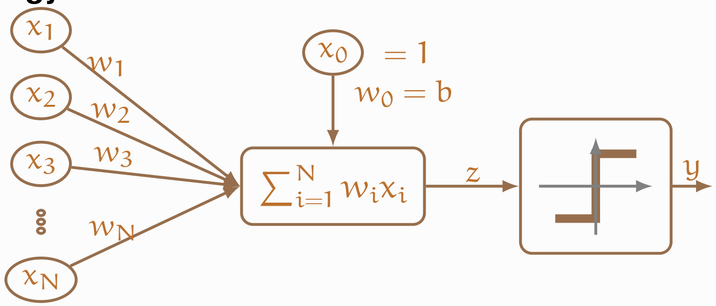
- szinkron – ciklusos – működés;
- rétegek $\Rightarrow$ korlátolt számú kapcsolat;

Történeti áttekintő Összefoglaló

- Idegrendszer $=$ információ-feldolgozó egység.
  Kérdések (nincsenek kellően tisztázva):
  - Mi az *információ*,
  - Hogyan történik a feldolgozás.
- biológiai neuronok aszinkron-működésűek: miért jó a mesterséges neuron szinkronizált jellege.
- mesterséges neuronok egymáshoz kapcsolódása csekély: a biológiai rendszerek nagyságrendekkel több kapcsolatot kezelnek.

Perceptron I

Egyszerűsített természetes neuron-modell:



- Korai neuron-modell: McCullogh-Pitts $\sim$ 1958;
- x_i és y értékek binárisak;
- Aktivációs függvény a lépcsőfüggvény:

$$H(z) = \begin{cases} -1 & \text{ha } z < 0 \\ +1 & \text{ha } z \geq 0 \end{cases}$$

Perceptron II

Perceptron

- ON/OFF neuron állapotok
  A neuronok vagy aktívak vagy nem, mint a bináris logikában
  rezolució-szerű működés
- Tanulási szabály: ha $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$-en hiba történt, akkor

$$w_i(t+1) \Leftarrow w_i(t) + x_i^{(n)} y^{(n)}$$

ahol w_{ij} súly, $\mathbf{x}^{(n)}$ bemeneti minta, $y^{(n)}$ a minta osztálya.

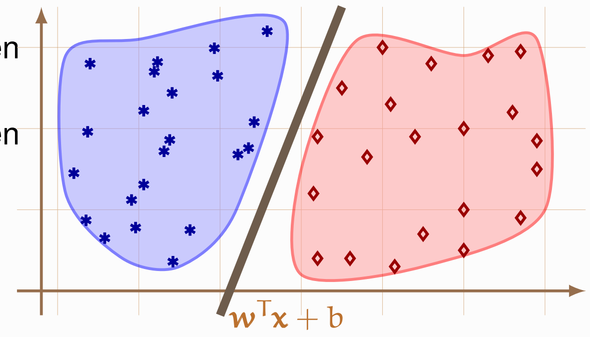
- Konvergencia-tétel: Ha a
  - tanulási minták $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$
  - elválaszthatóak, def (osztályozási feladat)

  akkor a perceptron konvergál.

Elválaszthatóság I

kék osztály kódja legyen $+1$;

piros osztály kódja legyen -1;

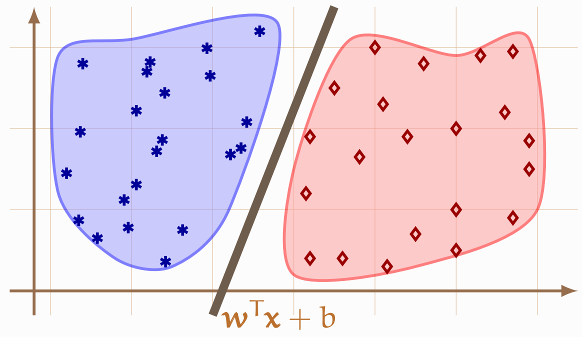


Szeparáló hipersík

a $\{\mathbf{w}, b\}$ vektorral illetve valós szám által meghatározott felület, ha

$$\begin{array}{lll} \mathbf{w}^T\mathbf{x}_i + b > 0 & \forall i \ ; \ y_i > 0. \\ \mathbf{w}^T\mathbf{x}_i + b < 0 & \forall i \ ; \ y_i < 0. \end{array}$$

Elválaszthatóság II



Szeparáló hipersík:

$$y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) > 0 \qquad \forall i = 1, N.$$

ahol $\mathbf{x}_i$ - i-edik mintavektor és y_i az ahhoz tartozó osztály kódja.

vissza

Perceptron Konvergenciatétel

Konvergencia*tétel*

A tanulási szabály alkalmazásával a perceptron **véges** idő alatt konvergál.

Biz:

Módszer: találjunk – felső és alsó – korlátot a lépések számának.

1. Újradefiniáljuk a bemeneti adatokat:

$$\begin{array}{lll} \hat{\mathbf{x}}_i & \stackrel{\text{def}}{=} & [\mathbf{x}_i, 1]^T \in \mathbb{R}^{d+1} \qquad \Rightarrow \qquad \mathbf{w} \stackrel{\text{def}}{=} [\mathbf{w}, b]^T \in \mathbb{R}^{d+1} \\ \mathbf{x}_i & \stackrel{\text{def}}{=} & y_i\hat{\mathbf{x}}_i \end{array}$$

Konvergenciatétel II

2. **Új**radefiniált feladat

   Találjunk **egy (1)** $\mathbf{w} \in \mathbb{R}^{d+1}$ értéket úgy, hogy

$$\forall i \ ; \ \mathbf{w}^T\mathbf{x}_i > 0$$

   minden $\mathbf{x}_i$-re (már tartalmazzák a címkéket is).

3. Tanítási szabály

   Ha hiba történt, akkor módosítjuk a perceptron súlyait:

$$w_i(t+1) \Leftarrow w_i(t) + x_i^{(n)}$$

   (rekurzívan visszafejtve: $\mathbf{w}(t+1) = \mathbf{x}^1 + \cdots + \mathbf{x}^t$)

4. Hiba $\Leftrightarrow$ $\mathbf{w}^T(t)\mathbf{x}^{(n)} < 0$.

(1) Amennyiben van egy érték, akkor nagyon sok elfogadható érték létezik.

Konvergenciatétel III

5 Feltételezzük, hogy létezik szeparáló hipersík:

$$\exists \mathbf{w}_0 \in \mathbb{R}^{d+1} \forall i \quad ; \quad \mathbf{w}_0^T \mathbf{x}^{(n)} \geq \alpha > 0$$

6 Vizsgáljuk a $\mathbf{w}_0^T \mathbf{w}(t+1)$ skaláris szorzatot:

$$\mathbf{w}_0^T \mathbf{w}(t+1) = \mathbf{w}_0^T \mathbf{x}^1 + \cdots + \mathbf{w}_0^T \mathbf{x}^t \geq t\alpha$$

7 Cauchy-Schwartz egyenlőtlenség $\|a\|^2 \|b\|^2 \geq \left(a^T b\right)^2$ alkalmazzuk:

$$\|\mathbf{w}_0\|^2 \|\mathbf{w}(t+1)\|^2 \geq \left(\mathbf{w}_0^T \mathbf{w}(t)\right)^2 \geq (t\alpha)^2$$

vagyis

$$\|\mathbf{w}(t+1)\|^2 \geq \frac{(t\alpha)^2}{\|\mathbf{w}_0\|^2}$$

Konvergenciatétel IV

Másrészt - minden tanuló lépésnél:

8 $\mathbf{w}(t+1) = \mathbf{w}(t) + \mathbf{x}^{(t)}$, tehát:

$$\begin{aligned}
\|\mathbf{w}(t+1)\|^2 &= \|\mathbf{w}(t) + \mathbf{x}^{(t)}\|^2 \\
&= \|\mathbf{w}(t)\|^2 + 2\mathbf{w}(t)\mathbf{x}^{(t)} + \|\mathbf{x}^{(t)}\|^2 \\
&\leq \|\mathbf{w}(t)\|^2 + \|\mathbf{x}^{(t)}\|^2 = t\beta^2
\end{aligned}$$

Ahol $\beta^2 = \max\left\{\|\mathbf{x}^{(n)}\|^2 \;;\; n = 1..N\right\}$

9 Van tehát két egyenlőtlenség, melyből kiszámítható a t maximális értéke:

$$t_{max} \leq \frac{\|\mathbf{w}_0\|^2 \beta^2}{\alpha^2}$$

Tehát a perceptron konvergál □

Matlab Demó I

Kódoljuk a perceptron algoritmust:

- generáljuk az adatokat. Specifikáljuk:
  - az adatok dimenzióját, számát;
  - az elválasztó hipersíkot;
  - generáljuk a bemeneti mintákat illetve az azokhoz tartozó kimeneti osztálykódokat.
- Transzformálunk a konvergenciatétel szerint.
- Inicializáljuk a perceptront zeró értékekkel.
- Futtatjuk az algoritmust:
  - mérjük a hibajavító lépéseket;
  - mérjük a ciklusok számát.

Matlab Demó II

Matlab kód:

```
% ADATOK parameterei
dim = 9;
N   = 200;
% az elvalaszto hipersik
w_0 = [1;-3; 4;-1; 2;-6;-5; 5; 3];
b   =  2;
% adatgeneralas
X = sign(rand(N,dim) - 0.5);
% cimkek generalasa
Y = sign(X*w_0 + b);
% hatar-mintak szurese
idx = find(~Y);
idx = setdiff([1:N],idx);
X = X(idx,:); Y=Y(idx);
N = length(Y);
```

Matlab Demó III

```
%% Perceptron tanulas %%%%%%
% 1 mintak atalakitasa
X = [ X ones(N,1)];
% 2
XX= diag(Y) * X;
%% Tanulas -- lokalis valtozok beallitasa
w = zeros(dim+1,1);
Term = 0;
steps = 0; sweeps = 0;
```


Matlab Demó IV

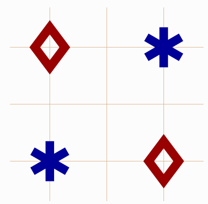
```
while ~Term;
  Term = 1;

  for ii=1:N;
    if XX(ii,:) * w <= 0
      w = w + XX(ii,:)';
      steps = steps + 1;
      fprintf('%d:',steps);  % megj.
      fprintf('%4d',w);fprintf('\n');
      Term = 0;
    end;
  end;
  sweeps = sweeps +1;
end;
```


Matlab Demó V

Kiíratjuk a statisztikákat:

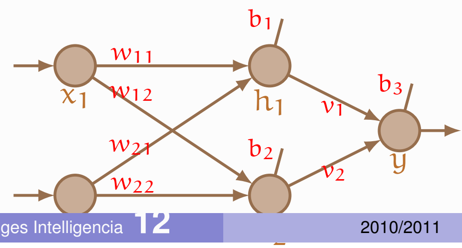
```
fprintf('\nTeljes bejarasok szama: %d\n',sweeps);
fprintf('Eredmeny: ');
fprintf('%4d ',w); fprintf('\n');
fprintf('Kezdo hipersik:');
fprintf('%4d',[w_0;b]); fprintf('\n');
```


Összefoglaló



- ’60-as évek tanulási paradigmája;
- Logikai függvényeket keres – bemenet és kimenet binárisak;
- Cél a gondolkodás modellezése
- Nem tudja szeparálni az XOR művelet kimeneteit (Minsky és Papert ’62)

Feladat:
találjunk konfigurációt, mely az XOR-t helyesen osztályozza.



Perceptron — Kitekintő

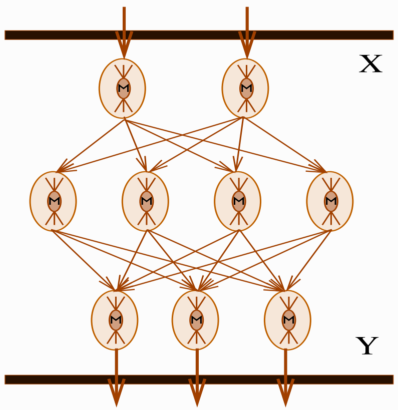
- A perceptron egy *black-box* – fekete doboz – módszer;
- Valós életből vett feladatok megoldására lehet használni:
  1. Gyűjtünk adatokat,
  2. Építünk modellt,
  3. Tanítjuk a modellt,
  4. Paraméterek = modell kiválasztása.
- A későbbi neurális háló modellek jó példái a fenti automatizált feladatmegoldási paradigmának.
- A neurális hálózat modelleket nagyon sokáig – ma is – használ(j)ták modellezésre.

A robotok programozásánál **nem volt lehetséges** speciális előprogramozás nélkül összeszedni a mosatlant vagy megetetni a cicákat.
A robotok hagyományosan 3D-s modellt készítenek környezetükről. Erre általában két kamerát használtak, a távolságokat idő– és számítási kapacitás-igényes munkával, az adott tárgy pontjainak tömege alapján határozták meg.
A STAIR alternatív megoldást kínál: pontok sokasága helyett az algoritmus az adott tárgy megfogható részének a középpontját azonosítja. Élek hosszúságát számolja ki, majd az eredményt az adatbázisban levő statisztikailag hasonló tárgyakkal veti össze.
A szoftver ezek alapján hozza létre a háromdimenziós modelleket; lényegesen gyorsabban és kevesebb munkával, mint a régi megközelítésben.

Stanford STAIR agens.ai

Többrétegű Hálók — I

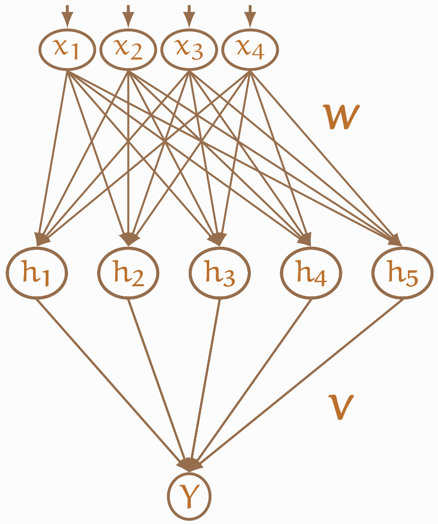
- Az információ – számok – a bemeneti X rétegből a kimeneti Y réteg fele halad.
- Egy mintára kiszámítjuk a kimeneti értéket, minden neuron értékét kiszámítva.
- Feltételezzük, hogy minden – adat és súly – folytonos.



MLP NEM Bayes-háló

A neurális hálók kapcsolatai kommunikációt, a Bayes-hálók kapcsolatai függőséget jelentenek.

Többrétegű Hálók — II



MI az információ?

„Formálisan”:

- A neuronháló csúcsai az információt feldolgozzák.

$$y_{ki} \stackrel{\text{def}}{=} f_{akt}(\mathbf{x}_{be})$$

- Az élek: a csúcsok közötti kapcsolatok.

$$x_{be}^{(i)\prime} \stackrel{\text{def}}{=} \sum_j w_{ij} x_{ki}^{(j)}$$

- A fenti lépések ismétlődnek.

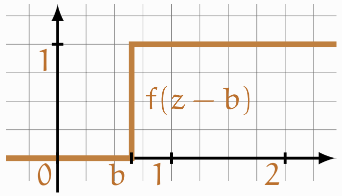
definíció szerint $\mathbf{x}_{ki} \in \mathbb{R}^k$

Többrétegű Hálók III

A bemenő aktivitás:
$y_j = \sum_i w_{ij} x_i + b_j$.

A perceptron tanulási algoritmushoz hasonlóan:

- előbb: $\mathbf{x}_i' = [x_1, \ldots, x_d, 1]^T$
- $\Rightarrow \quad \mathbf{w}_{\cdot j}' = [w_1, \ldots, w_d, b]^T$



$f(z) = \mathrm{sgn}(z)$

Univerzális approximáció

A következő rendszer:

$$f_M(z|\mathbf{w}, \mathbf{b}) = \sum_{m=1}^{M} w_m f(z - b_m)$$

univerzális approximátor, azaz $\forall \delta > 0\ \forall g \in L^2(\mathbb{R}^d)$
$\exists M, \mathbf{w}, \mathbf{b}$ ú.h. $\|g - f_M(z|\mathbf{w}, \mathbf{b})\|_{L_2} \le \delta$

Többrétegű Hálók IV

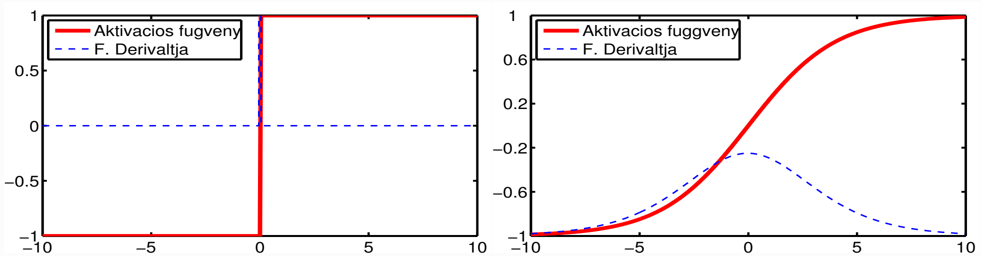
Univerzális approximátor: bizonyítás a szint-halmazokon alapszik (lásd Lebesgue integrálás).

Vizualizálás:
Mintavételezés a $\mathbf{w}$ és $\mathbf{b}$ értékekből

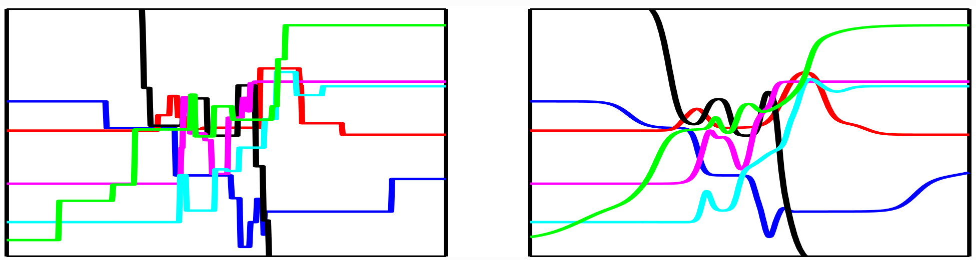
```
%Generatorfuggvenyek
f = inline('((x>0)-0.5).*2');
g = inline('2./(1+exp(-2*x))-1');
% komponensek szama
mComp = 10;
% fv.-nyek szama
nF    = 6;
% X tengely
xx=-10:0.05:10;
xd=repmat(xx,[mComp,1]);
% plot init
clf; hf = subplot(1,2,1);
set(gca, ...
 'Position',[0.05 0.1 0.42 0.85],...
 'YTick',[],'XTick',[]);
hold on; box on; ylim([-25,25]);
set(gca,...
  'Position',[0.55 0.1 0.42 0.85],...
  'YTick',[],'XTick',[]);
hold on; box on; ylim([-25,25]);
```

```
hg = subplot(1,2,2);
% megjelenitesi stilus
style = {'b','r','k','m','c','g'};
% F.v.-nyek generalasa
for ii=1:nF;
  % egyutthatok
  ss=(2*rand(mComp,1) -1)*6;
  ww=(2*rand(mComp,1) -1)*6;
  bb=(2*rand(mComp,1) -1)*10;

  bd=diag(bb)*ones(size(xd));
  y_f = sum(diag(ww) * f(diag(ss)*xd + bd));
  y_g = sum(diag(ww) * g(diag(ss)*xd + bd));
  % megjelenites
  plot(hf,xx,y_f,style{ii},'LineWidth',3);
  plot(hg,xx,y_g,style{ii},'LineWidth',3);
end;
% nyomtatas
print -depsc2 rand_fn.eps
```


Aktivációs függvények



Véletlen függvények a függvényosztályokból:



Backpropagation I

- **Nem bináris függvényt közelít.**
  A bemeneti/kimeneti értékek folytonosak.

$$f_{NN} : \mathbb{R}^d \rightarrow \mathbb{R}$$

- Folytonos esetben használható a négyzetes hibafüggvény:

$$E(\mathbf{w}_{NN}) = \frac{1}{2} \sum_{n=1}^{N} \left(y_n - f_{NN}(\mathbf{x}_n)\right)^2$$

  ahol $\mathbf{w}_{NN}$ a neuronháló paraméterei.

- Differenciálható akt. függvényeknél a gradiens szabály alkalmazható $\Rightarrow$ **backpropagation szabály**.

Backpropagation II

- Error back-propagation $\Leftrightarrow$ négyzetes hiba visszaterjesztése
- Egy adott tanulási halmazhoz és egy $\boldsymbol{w}_{NN}$ neurális hálóhoz rendelt négyzetes hiba

$$E(\boldsymbol{w}_{NN}) = \frac{1}{2}\sum_{n=1}^{N}\left(y_n - f_{NN}(\boldsymbol{x}_n)\right)^2$$

akkor zéró, ha a neurális háló minden adatot jól közelít.

- Egy adott $\boldsymbol{w}_{NN}^{(0)}$ hálónál jobban közelítő hálót kapunk ha egy kis lépést végzünk a negatív gradiens irányába.

Backpropagation III

- **„Képletesen”**

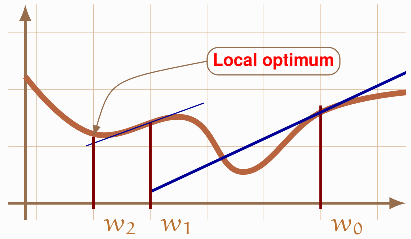
$$\begin{aligned}\boldsymbol{w}_{NN}^{(t+1)} &\Leftarrow \boldsymbol{w}_{NN}^{(t)} - \alpha \frac{\partial E(\boldsymbol{w}_{NN}^{(t)})}{\partial \boldsymbol{w}_{NN}^{(t)}} \\ &\Leftarrow \boldsymbol{w}_{NN}^{(t)} - \alpha\, \partial_{\boldsymbol{w}} E(\boldsymbol{w}_{NN}^{(t)})\end{aligned}$$

ahol

α – tanulási együttható;

$\partial_{\boldsymbol{w}} E(\boldsymbol{w}_{NN}^{(t)})$ – a hiba gradiense (vektorok!).

Kérdés: milyen α értékek megfelelőek?
Kis lépések $\Rightarrow$ hosszú konvergencia.
Nagy lépések $\Rightarrow$ nem konvergál.



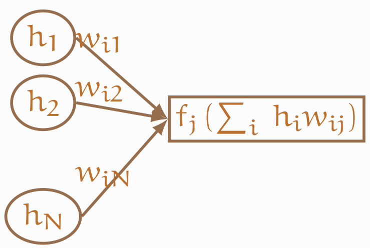
Backpropagation IV

Differenciálás szabálya:

$$\frac{\partial f(g_1(w_i),\ldots,g_k(w_i))}{\partial w_i} = \sum_{g_j} \frac{\partial f(g_1,\ldots,g_k)}{\partial g_j}\frac{\partial g_j}{\partial w_i}$$

Neuronháló függvénye:

$$y = \cdots\left(\cdots, f_j\left(\sum_i h_i w_{ij}\right), \cdots\right)$$



$$\partial_{w_{ij}} E(\cdot) = \sum_j \partial_{f_j} E(\cdot)\, \partial_{w_{ij}} f_j\left(\sum_i w_{ij}h_i\right)$$

Backpropagation V

(folyt)

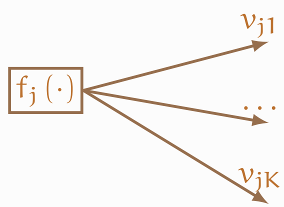
$$\begin{aligned}\partial_{w_{ij}} E(\cdot) &= \partial_{f_j} E(\cdots)\, \partial_{w_{ij}} f_j\left(\sum\nolimits_{n=1}^{N} w_{ij}h_i\right) \\ &= h_i\ \ f_j'\left(\sum\nolimits_{n=1}^{N} w_{ij}h_i\right) \partial_{f_j} E(\cdot)\end{aligned}$$

A bemeneti érték és a kimeneti hiba szorzódnak.

És $\partial_{f_j} E(\cdot) = \partial_{f_j} E\left(\ldots, g_k\left(\sum_j v_{jk} f_j\right), \ldots\right)$

Lánc-deriválás szerint:

$$\partial_{f_j} E(\cdot) = \sum_k v_{jk}\ \ \partial_{g_k} E(\cdot) g_k'(\cdot)$$



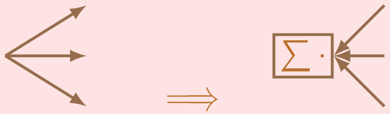
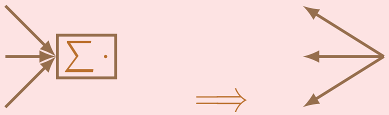
Az egyéni hibák súlyozottan összeadódnak.

Backpropagation VI

(folyt)

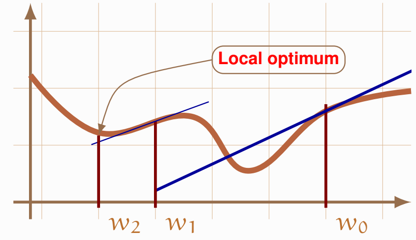
Tehát: amíg a működésnél a jel előre terjed, a tanulás folyamán a hiba **visszafele**; a kimeneti réteg felől a háló bemenete felé „terjed".

Hiba-visszaterjesztés (BP)

- 
  azaz az osztópontok összegzővé alakulnak;
- 
  azaz az összegző csomópontok meg hiba-elosztóvá.

Backpropagation Összefoglaló

- A BP algoritmus a gradiens szabály alkalmazása a neuronhálókra;
- Az alap a négyzetes hiba;
- Egyszerűen implementálható;
- Nagyon sok alkalmazás;



Hátrányok:

- Mivel gradiens, lassú $\Rightarrow$ nagyon sokáig tarthat a tanulás.
- Más módszerekkel állítani be a súlyokat: *Newton*, *Konjugált gradiens*.
- Nincs a becsült mennyiségre konfidencia $\Rightarrow$ valószínűségi módszerek szükségesek.

Rosenbrock

Backpropagation Példa I

Feladat: keressük a $h(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$ függvény minimumát a gradiens szabály használatával.

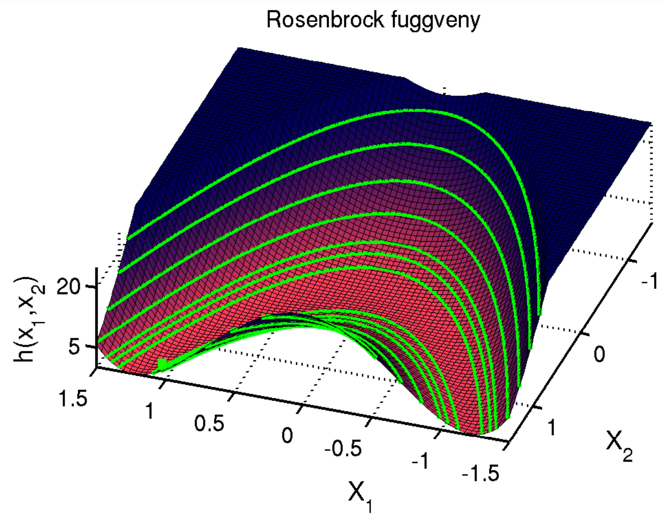
$$\partial_1 h(\cdot) = 400(x_2 - x_1^2)x_1 + 2(x_1 - 1)$$

$$\partial_2 h(\cdot) = 200(x_2 - x_1^2)$$

A második egyenlet $\leftrightarrow \quad x_2 = x_1$

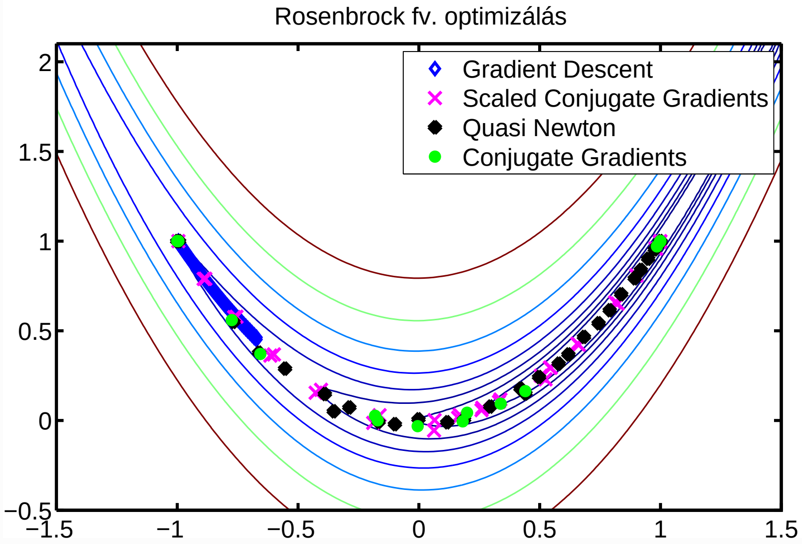
Behelyettesítve $x_1 = 1$.

Induljunk a $(-1, 1)$ pontból.



Backpropagation Példa II

BP szabály = gradiens – általában nagyon lassú.



M.L.P. Összefoglaló

- Neurális hálók **fekete-doboz** módszerek:
  1. minden feladat megoldására potenciálisan alkalmazhatóak;
  2. jó eredményekhez (majdnem mindig) kell egy kis igazítás a módszeren;
  3. siker függ a tapasztalattól;[7]
- **BackPropagation = gradiens tanulás**
  1. általában nagyon lassú;
  2. fejlett módszerek: konjugált gradiens módszerek, quasi-Newton, *etc.* gyorsítanak a konvergencián.
  3. lokális optimumok elkerülésére többszálú optimizálás, pl. mintavételezéssel kijelölni a kezdőértékeket.

[7] Érdemes tehát tovább tanulni.

D.O. Hebb I

D.O. Hebb
The organization of behavior

A neurális moduláció alapelvei:

- Két neuron között a kapcsolat erőssége arányos a neuronok pre-, illetve poszt-szinaptikus tevékenységével;
- Azon neuron–csoportok, melyek általában egyszerre tüzelnek, egy egységet alkotnak; (modularitás)
- A *gondolkodás* folyamata ezen sejt-csoportok szekvenciális aktivációja;

D.O. Hebb II

Organization of Behavior:
When an axon of cell A is near enough to excite B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

(p. 62)

Neuroscience $\Leftrightarrow$ „Hebb synapse"

„Hebb szabály"

$$\Delta w_{ij} = \alpha a_i a_j$$

Ahol

a_i, a_j – neuronok aktivitásai;
w_{ij} – az összekötő szinapszis erőssége;
α – tanulási együttható.

Önszervező Hálók I

Haykin: Neural Networks
A comprehensive foundation, IEEE press, 1994

Cél:

- Struktúrák felfedezése az adatokban;
- „Felügyelet" nélküli működés $\Rightarrow$ önszervezés.

Tanulási szabályok:

- lokális szabályok, melyek – sok neuronra alkalmazva – egy globális függvényt határoznak meg.
- Alapja (Turing 1952):
  „Global order can arise from local interactions."

Önszervező Hálók II

Önszervező rendszerek alapelvei (von der Malsburg):

- *A szinapszisok önmegerősítőek*

  A pre-, illetve posztszinaptikus neuronok egyidejű aktivációja erősíti a szinapszist (lásd Hebb).

- *A szinapszisok versengenek az erőforrásokért*

  A legerősebb szinapszis biztos, hogy túléli. Ez a többi rovására történik (winner-takes-all).

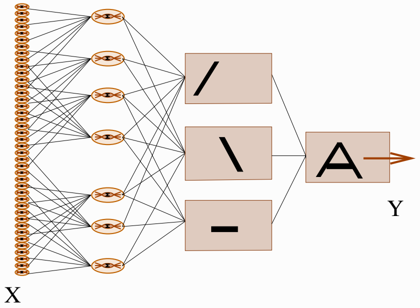
- *A szinapszisok változásai koordináltak*

  Egy szinapszis nem kódol robusztusan, ezért egy régió neuronjai – majdnem – azonos bemenet–kimenet kapcsolatot kódolnak.

Önszervező Hálók III

Önszervező tanulás ≡ **Struktúra-felismerés**

- A vizuális rendszer önszervező:
  - fejlődésének elején csak a felismerésre való képesség van jelen;
  - a működés során egyes kép-csoportok – klaszterek – együttesen lesznek ábrázolva;
- A származtatott fogalmak ábrázolása a hierarchia egy felsőbb szintjén történik.



Példa:

- neurális rendszer pixelcsoportokat keres.
- némely pixelcsoport gyakori; ezen csoportok rözgülnek a rendszerben.
- a felső szinten a rendszer címkézést végez.

Hebb-féle tanulás I

Hebb tanulás - auto-asszociatív rendszerek

- A rendszer bemenete egy minta - nincs kimeneti jel.
- A „kimeneten” vagy:
  - az eredeti mintát; vagy
  - egy csoportosítást, „sűrített” ábrázolást

  szeretnénk visszakapni.
- csoportosításkor a kimeneti neuronok **versengenek**:

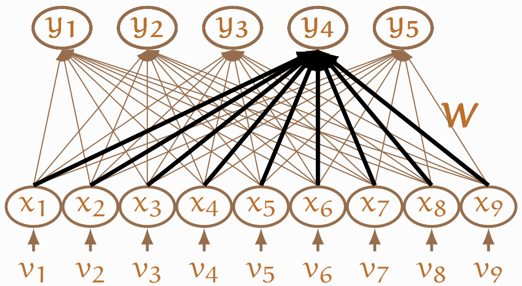
$$\hat{y}_j = \sum_i x_i w_{ij}$$

$$y_l = \begin{cases} 1 & l = k \\ 0 & l \neq k \end{cases} \qquad \text{ahol } k = \operatorname*{argmax}_j \hat{y}_j$$

Hebb-féle tanulás II

$$y_l = \begin{cases} 1 & l = k \\ 0 & l \neq k \end{cases} \qquad \text{ahol } k = \operatorname*{argmax}_j \hat{y}_j$$

- Végleges kimeneti érték az argmax művelet után.
- **Versengés** a bemeneti mintákért.



- Feltételezve egy kezdő W – véletlen – súlyvektort, a v minta bemutatása után
- csak a $w_{\cdot 4}$ súlyok változnak a kompetitív tanulás során.

Hebb-féle tanulás III

$$\Delta w_{ij} = \alpha x_i y_j$$

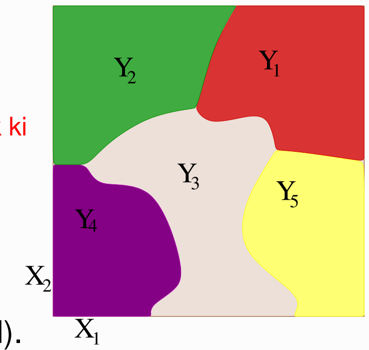
- minden y_k neuron attraktorként működik.

receptív mezők alakulnak ki

- $\alpha \to 0$ tanulási mód, pl.

$$\alpha_t = \frac{K}{t}$$

- az **asszociatív háló** konvergál (lásd).



ami nincs: kapcsolatok a kimeneti neuronok között:

Kohonen-hálók

Kohonen hálók I

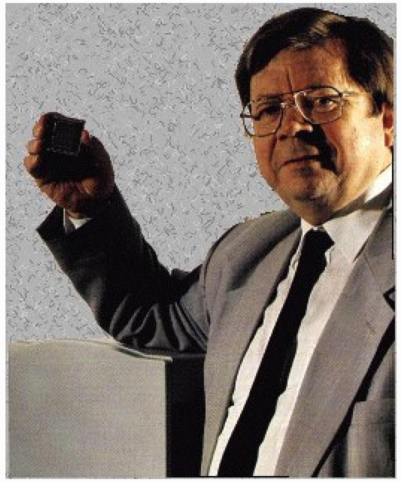
Teuvo Kohonen

Since the 1960's, ~ introduced several new concepts to neural computing: theories of distributed associative memory and optimal associative mappings, the self-organizing feature maps (SOMs), the learning vector quantization (LVQ), . . .

the Adaptive-Subspace SOM (ASSOM) in which invariant-feature filters emerge . . .

A new SOM architecture WEBSOM has been developed in his laboratory for exploratory textual data mining.

http://websom.hut.fi/websom/



Wikipedia link

Kohonen hálók II

Jellemzők:

- kétrétegű háló, egy bemeneti és egy kimeneti réteggel;

(mint előbb)

- kompetitív háló: csak egy neuron lesz aktív;

(mint előbb)

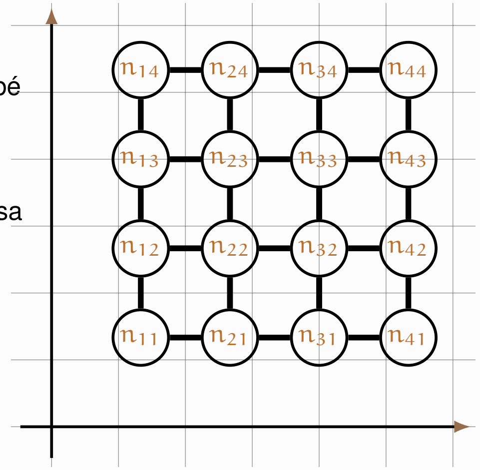
- a kimeneti rétegen **topológia** van értelmezve

(ÚJ)

Kohonen hálók Topológia

Topológia – szomszédságot jelent

- segít a tanulásban:
  a szomszédok is tanulnak – kevésbé ...
- értelmezhető eredmény:
  a szomszédok aktivitása is aktivitása alapján pontosabb rekonstrukció
- a kimeneti n_{ij} neuronok megcímkézhetőek:
  Kohonen: ... fonéma annotáció



Kohonen hálók — Tanulás I

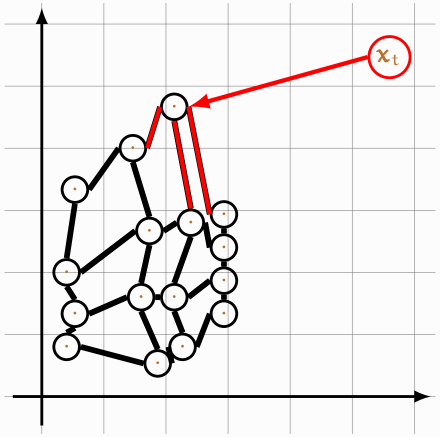
Az algoritmus iteratív:

1. minták egyenként; a t-edik időpontban legyen $\mathbf{x}_t$
2. kiszámtítjuk a kimeneti neuronok aktivitását:
$$y_j = \sum_i w_{ij} x_i(t) = \mathbf{W}\mathbf{x}(t)$$
3. megkeressük a nyertes kimeneti neuront:
$$k \stackrel{\text{def}}{=} \underset{j}{\operatorname{argmax}}\ y_j$$
4. Az összes szomszédos neuron súlyát módosítjuk:
$$w_{ij}(t+1) = w_{ij}(t) + \alpha\ \eta(j,k) x_i(t)$$
Normáljuk a súlyokat $w_{\cdot j} \Leftarrow w_{\cdot j}/\|w_{\cdot j}\|$
5. GOTO 1

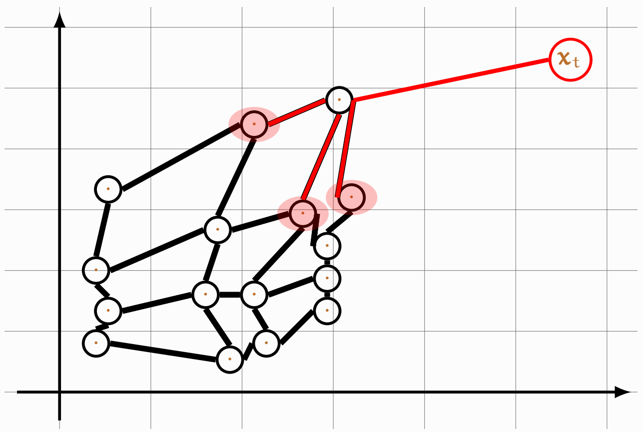
Kohonen hálók — Tanulás II

Szavakban:

- a nyertes neuron „aktivizálódik”;
- A nyertes neuront és szomszédait „közelebb” hozzuk a bemenethez;
- Mindegyik neuron specializálódik a mintákra: arra lesz a legnagyobb a kimenete;



Kohonen hálók — Tanulás III



Eredmény: a neuronok elhelyezkedését a bemeneti minták sűrűségfüggvénye határozza meg – több neuron kerül a sűrűbb régiókba.

Önszervező rendszerek — Összefoglaló

- Az adatokhoz nincs kimeneti érték rendelve;
- A rendszer általában az adatok egy csoportosítását valósítja meg;
- Ez implicit módon a bemeneti adatok terének a felosztását jelenti;
- Zajszűrés/Sűrítés: a teljes bemeneti adatok helyett a hozzá tartozó osztály kódját küldjük.

Példa – Térképészgép

Walter Bischof, Jun Zhou (U. Alberta) és Terry Caelli (Australian N.U) a kartográfiát (részben) automatizáló programmal álltak elő.

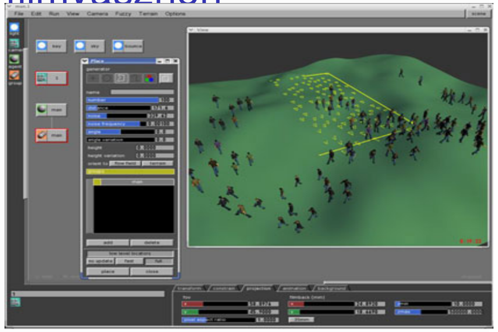
A program embertől tanulja, miként fedezzen fel és azonosítson légi felvételeken korábban **nem szereplő vagy megváltozott** elemeket: utakat, vasutakat, épületeket.

Ágens portál *Walter Bischof projektek*

Működése:
Egyre többet tud meg – „okosabb” lesz. Kellő mennyiségű gyakorlás után az operátor rábízza a munkát. *Bayes-féle következtetést használ*: korábbi mérések alapján végez új becsléseket. A legjobbakból állapítja meg az adott országút soron következő pontjait, és ezt mindaddig folytatja, amíg az új elemek pozíciója *megfelel az elvárásoknak*.

Példa – Ágensek filmvásznon

A *Massive céggel* a Gyűrűk Ura trilógiában találkozhattunk 2001-ben.



Az összes *csatajelenet az ő programjukkal készült*. Nem véletlen, hogy a tavalyi (2005) King Kong-ban *statiszták helyett* a Massive ágenseit használták.

Ágens portál *Walter Bischof projektek*

Hangya-kolónia szimuláció egy kedvelt M.I. téma. A „buta” hangya-ágensek *feromon nyomokat* hagynak maguk után; ezek alapján mozognak. Nincs globális információ a környezetről: csak a feromon-nyomot követik. A kolónia *intelligens* viselkedést mutat: megtalálja a legrövidebb utat a fészek és az élelemforrás között.

Gulyás László, ELTE

Gépi tanulás

Történelmi háttér / Motiváció:

- nagyon nagy mennyiségű adat, melyet szeretnénk automatikusan feldolgozni;
- A matematikai modellek általánosak – nem egy adott feladatra vannak kiélezve,
- Szükség van esetenkénti tanulmányozására egy-egy feladattípusnak ⇒ „tudomány–ág”, mely a modelleket a feladatokhoz „idomítja”.

Gépi tanulás – Meghatározás

Gépi tanulás

Módszerek – statisztikai, valószínűségi – gyűjteménye valós feladatok megoldására.

Például:

- Zajszűrés: (nem–)lineáris regresszió és (nem–)normális zajt feltételezve;
- Osztályozás: bináris, több–osztályos, illetve
- Részlegesen címkézett adatokra;
- Klaszterezés – adatok csoportosítása,
- Függvényinverzió,
- Sűrűségbecslés,
- „Kakukktojások” felfedezése – *novelty detection*.

Szükséges az adatok **modellezése**.

Gépi tanulás — Mitchell

T. Mitchell: **Machine Learning (ML)**, textbook

…is concerned with the question of how to construct programs that *automatically* improve with *experience*.

In recent years *(1997)* successful ML algorithms have been developed in

- data-mining – fraud detection,
- filtering for user's preferences,
- vehicles that learn to drive on highways;
- advances in the theory and algorithms that form the foundations of tis field.

Thomas Mitchell

Tanuló rendszer

olyan számítógépes program, melyek képességei a működés során javulnak.

Adatmodellezés

$f(\mathbf{x})$ $\xrightarrow[(\mathbf{x}_i, y_i)]{\text{Megfigyelés}}$ $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, $\vdots$, $(\mathbf{x}_1, y_1)$

Feltételezzük, hogy:

- „Valahol": létezik a $t = f(x)$ függvény, mely generál adatokat;
- A megfigyelt értékek zajosak: nem $y = f(\mathbf{x})$-et kapunk, hanem például:

$$\begin{aligned} y_n &= t_n + \epsilon && \text{additív zaj} \\ y_n &= h(t_n, \epsilon) && h \text{ módosító függvény} \end{aligned}$$

- **Feladat: találjuk meg az $y = f(x)$ függvényt.**

Rejtett változós modellek I

$\hat{f}(\mathbf{x})$ $\xleftarrow[f \in \mathcal{F}]{\text{Inferencia}}$ $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, $\vdots$, $(\mathbf{x}_1, y_1)$

Szükséges:

- **Adatok halmaza** – megfigyelés során gyűjtjük.
- **Függvény osztály feltételezése**. Lehet:
  - K-ad fokú polinomok,
  - Fourier polinomok,
  - Wavelet-ek;
- A megfigyelési folyamat ismerete – a zaj kódolása;
- Algoritmus a legmegfelelőbb függvény kiválasztására.

Rejtett változós modellek II

Adottak:

- **Adathalmaz** $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$.
- Függvényosztály (feltételezzük, hogy a megoldász megfelelő):

$$(1) \quad \mathcal{F} = \{\mathbf{w}^T\mathbf{x} + b | \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}\}$$

$$(2) \quad \mathcal{F} = \Big\{a_0 + \sum_{k=1}^{K} a_k \sin(2\pi k x) + \sum_{k=1}^{K} b_k \cos(2\pi k x) \,|\, \mathbf{a}, \mathbf{b} \in \mathbb{R}^K,\ a_0 \in \mathbb{R}\Big\}$$

- A megfigyelési folyamat egy modellje:

$$y_n = f(\mathbf{x}_n) + \epsilon \quad \text{with } \epsilon \sim N(0, \sigma^2).$$

Rejtett változós modellek III

Általánosan:

1. **Adatok:** $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$.
2. **Függvényosztály:**
$$\mathcal{F} = \left\{ f(\mathbf{x}, \boldsymbol{\theta}) | \boldsymbol{\theta} \in \mathbb{R}^p \right\}$$
3. **Megfigyelés** – definiáljuk a hibafüggvényt:
$$L\left(y_n, f(\mathbf{x}_n, \boldsymbol{\theta})\right)$$
Gauss-zaj esetén:
$$L(y_n, f(\mathbf{x}_n, \boldsymbol{\theta})) = (y_n - f(\mathbf{x}_n, \boldsymbol{\theta}))^2.$$

Paraméterbecslés Bevezető

Paraméterbecslés

Találjuk meg a $\boldsymbol{\theta}$ **paraméter-vektor** optimális értékét.
Optimalitás:
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Omega} L(\mathcal{D}, \boldsymbol{\theta})$$
ahol

- Ω a paraméterek értelmezési tartománya.
- $L(\mathcal{D}, \boldsymbol{\theta})$ az adatokhoz rendelt **hibafüggvény**.

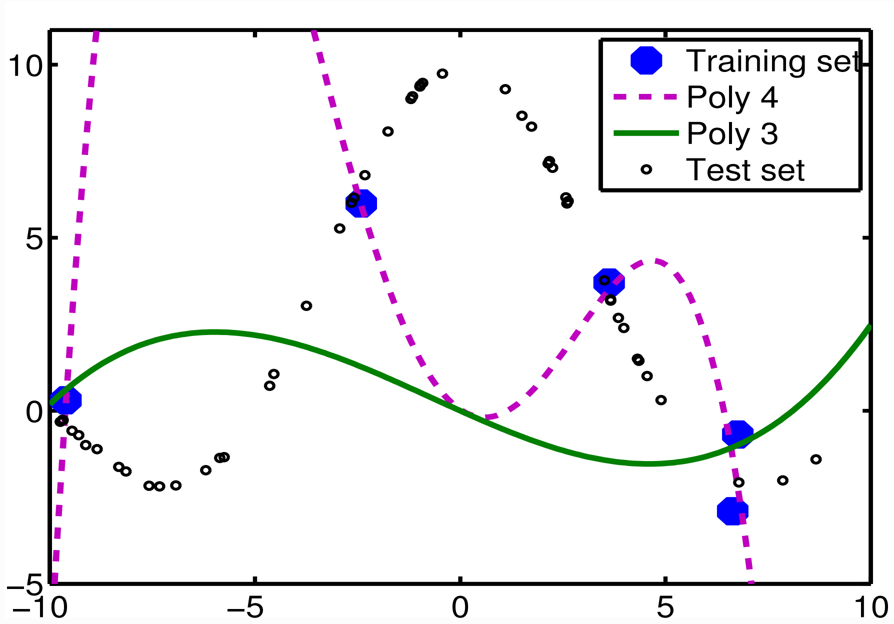
Példa a hibafüggvényre
Független megfigyelések esetén:
$$L(\mathcal{D}, \boldsymbol{\theta}) = \sum_{n=1}^{N} L(y_n, f(\mathbf{x}_n, \boldsymbol{\theta}))$$
miért független?

Paraméterbecslés M.L.

$L(\mathcal{D}, \boldsymbol{\theta})$ – (log)likelihood függvény.

M.L. = maximum likelihood értékű $\boldsymbol{\theta}$:
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Omega} L(\mathcal{D}, \boldsymbol{\theta})$$
Példa – legkisebb négyzetes hibával történő becslés:
$$L(\mathcal{D}, \boldsymbol{\theta}) = \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n, \boldsymbol{\theta}))^2$$
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Omega} \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n, \boldsymbol{\theta}))^2$$
Hátrány: Túl pontos illeszkedés – **over-fitting**.

M.L. becslés Over-fitting



Maximum a–posteriori becslés — MAP

MAP becsléshez szükségünk van valószínűségekre:

- A $\mathcal{D}$ adathalmaz valószínűsége (log–lik).
  $\boldsymbol{\theta}$ paraméterhez – az adatok által – rendelt fgv:
  $$P(y_n|\mathbf{x}_n,\boldsymbol{\theta},\mathcal{F}) \propto \exp\left[-L(y_n, f(\mathbf{x}_n,\boldsymbol{\theta}))\right]$$
  $\propto$ – normálni is kell.
- A-priori – előzetes – valószínűség
  Milyen $\boldsymbol{\theta}$ értékek a legvalószínűbbek
  $$p_0(\boldsymbol{\theta}) \propto \exp\left[-\frac{\|\boldsymbol{\theta}\|^2}{2}\right]$$
  az **adatok ismerete előtt specifikáljuk.**

MAP — II

A becslés a fenti valószínűségek – lik. & prior – kombinálása.

Bayes-szabály alkalmazása a $\boldsymbol{\theta}$ és $\mathcal{D}$változókra:

$$p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D}|\mathcal{F})}$$
$$p(\mathcal{D}|\mathcal{F}) = \int_\Omega d\boldsymbol{\theta}\ P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})$$

ahol $p(\mathcal{D}|\mathcal{F})$ – az adatok teljes valószínűsége az adott $\mathcal{F}$ modellre.

$P(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})$ **minden θ értékhez egy valószínűség**

$\Rightarrow$ egyszerűsítés szükséges.

MAP — III

M.A.P. becslés – a legvalószínűbb $\boldsymbol{\theta}$ megtalálása:
$$\boldsymbol{\theta}^*_{MAP} = \arg\max_{\boldsymbol{\theta}\in\Omega} p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})$$

Példa:

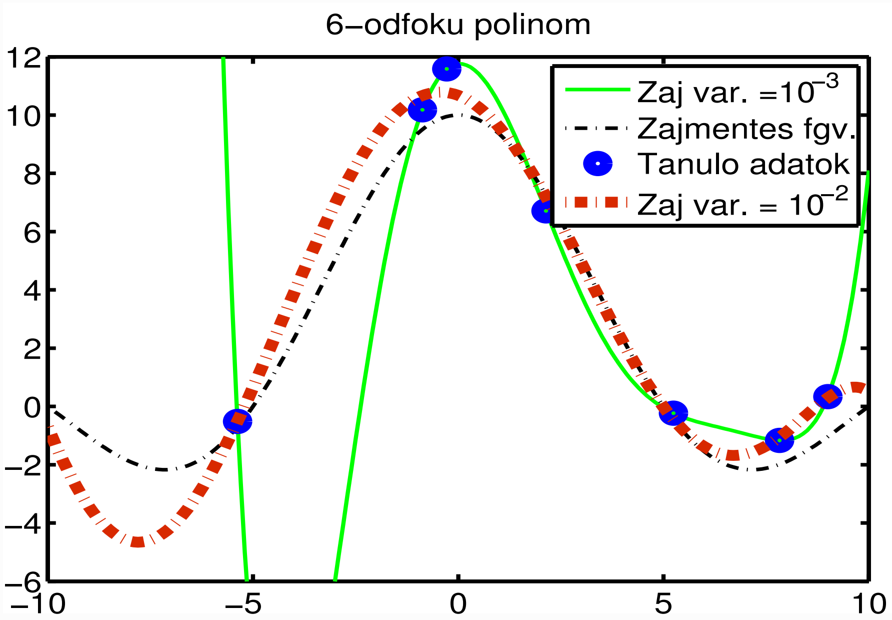
Az általános $L(y_n, f(\mathbf{x}_n,\boldsymbol{\theta}))$ hibafüggvényt és Gauss-féle a-priori valószínűségre.

A $\boldsymbol{\theta}^*_{MAP}$ kiszámítása (exp-ben levő tagok):
$$\boldsymbol{\theta}^*_{MAP} = \arg\max_{\boldsymbol{\theta}\in\Omega} K - \sum_n L(y_n, f(\mathbf{x}_n,\boldsymbol{\theta})) - \frac{\|\boldsymbol{\theta}\|^2}{2\sigma_o^2}$$

Ha $\sigma_o^2 \to \infty$ a maximum likelihood módszer.

. egy előjelcsere és $\max \to \min$ helyettesítés után.

MAP — Példa



Paraméter–becslések Bayes

Bayes – feltételes valószínűség – szabálya:

$$p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D}|\mathcal{F})}$$

$$p(\mathcal{D}|\mathcal{F}) = \int_\Omega d\boldsymbol{\theta}\ P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})$$

és megpróbáljuk a teljes eloszlást tárolni.

Miért?

Mivel nincs mindig képlet a poszt. eloszlásra, a

$$p_{post}(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})$$

sűrűségfüggvényt közelítjük.

Statisztikák ...szünet ...

Apróbetű: A statisztikák szerepe:

Az adatok helyettesítése

- Feltételezzük, hogy létezik egy generátor, mely az adatokat létrehozta;
- Ismerjük a generátor alakját: $(\mathbf{x}_i, y_i) = \Gamma(\mathbf{z}_i|\theta_1,\ldots,\theta_k)$.
- $\mathbf{z}_i$ – paraméterek, változók.
- $\boldsymbol{\theta} = [\theta_1,\ldots,\theta_k]^T$ – a modell paraméterei.

Az elemzéshez

- a **teljes** adathalmazt helyettesítjük a
- megfelelő **paraméterekkel**.
- A **predikcióhoz** nem használjuk az adatokat,
- „csak” a kinyert paramétereket és a **modellt**.

Bayes–becslés II

Bayes–becslés folyamán

Foglalkozunk

- a poszterior-eloszlás közelítésével:

$$\hat{p}(\boldsymbol{\theta}) \approx p_{post}(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})$$

- egy új adat valószínűségével – predikció:

$$\hat{p}(y^*|\mathbf{x}^*) \approx \int_\Omega d\boldsymbol{\theta}\ \hat{p}(\boldsymbol{\theta})P(y*|\mathbf{x}^*,\boldsymbol{\theta})$$

- Teljes valószínűségnek a közelítésével:

$$p(\mathcal{F}|\mathcal{D}) = p(\mathcal{D}|\mathcal{F})p_0(\mathcal{F})$$

A $p(\mathcal{F}|\mathcal{D})$ használható – elvileg – két különböző modell összehasonlítására.

Bayes–becslés közelítés

$$\hat{p}(\boldsymbol{\theta}) \approx p_{post}(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})$$

A közelítés alapja egy divergencia: méri, hogy a közelítő eloszlás minőségét:

$$\hat{p}(\boldsymbol{\theta}) = \underset{p(\boldsymbol{\theta})\in\Omega}{\operatorname{argmin}}\, d\left(p_{post}(\boldsymbol{\theta}|\mathcal{D},\mathcal{F}), p(\boldsymbol{\theta})\right)$$

A Bayes-becslések során fontos:

- **milyen családban** keressük az optimális eloszlást,
- **milyen divergenciát használunk** a „távolságok” mérésére. Gyakori a Kullback-Leibler divergencia.

Kullback–Leibler ...szünet ...

Gyakori a Kullback-Leibler divergencia:

$$d_{KL}\left(p_1(\boldsymbol{\theta})|p_2(\boldsymbol{\theta})\right) = \int_{\Omega_{\boldsymbol{\theta}}} d\boldsymbol{\theta}\, p_1(\boldsymbol{\theta}) \log \frac{p_1(\boldsymbol{\theta})}{p_2(\boldsymbol{\theta})}$$

Tulajdonságok:

- $d_{KL}(p_1|p_2) \geq 0$

  $$0 = \log\left(\int d\boldsymbol{\theta} p_2(\boldsymbol{\theta})\right) = \log\left(\int d\boldsymbol{\theta} p_1(\boldsymbol{\theta}) \frac{p_2(\boldsymbol{\theta})}{p_1(\boldsymbol{\theta})}\right)$$

  alkalmazzuk a Jensen egyenlőtlenségét:

  $$0 > \int d\boldsymbol{\theta} p_1(\boldsymbol{\theta}) \log \frac{p_2(\boldsymbol{\theta})}{p_1(\boldsymbol{\theta})}$$
  $$0 > -d_{KL}(p_1|p_2)$$

- $d_{KL}(p_1|p_2) = 0 \quad \Leftrightarrow \quad p_1(\boldsymbol{\theta}) = p_2(\boldsymbol{\theta})$

  kivéve egy nulla-mértékű halmazt...

- $d_{KL}(p_1|p_2) \neq d_{KL}(p_2|p_1)$
- Nem teljesül a háromszög-egyenlőtlenség.

  $$d_{KL}(p_1|p_2) \not\geq d_{KL}(p_1|p_3) + d_{KL}(p_3|p_2)$$

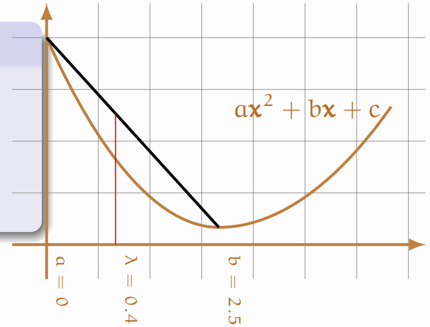
következő

Jensen egyenlőtlenség ...szünet II. ..

Jensen egyenlőtlenség

Legyen $f(\mathbf{x})$ egy konvex függvény az $[a, b]$ intervallumon. Ekkor $\forall \lambda \in [0, 1]$

$$f(\lambda a + (1-\lambda) b) \leq \lambda f(a) + (1-\lambda) f(b)$$



A **logaritmus:** függvény konkáv, tehát:

$$\log(\lambda a + (1-\lambda) b) \geq \lambda \log(a) + (1-\lambda)\log(b) \qquad \text{ahol } a, b > 0.$$

Sok együtthatóra:

$$\log\left(\sum_i \pi_i a_i\right) \geq \sum_i \pi_i \log(a_i) \qquad \text{ahol } \sum_i \pi_i = 1 \text{ és } \pi_i \geq 0.$$

$$\log\left(\int d\boldsymbol{\theta} p(\boldsymbol{\theta})\; f(\boldsymbol{\theta})\right) \geq \int d\boldsymbol{\theta} p(\boldsymbol{\theta})\; \log(f(\boldsymbol{\theta})) \qquad \text{ahol } f(\boldsymbol{\theta}) > 0.$$

vissza

Bayes–becslés közelítés

Példa közelítésre:

$$\Omega = \left\{ \exp\left[-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})^T \Sigma_{\boldsymbol{\theta}}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right] \mid \boldsymbol{\mu}_{\boldsymbol{\theta}} \in \mathcal{R}^d,\ \Sigma \in \mathcal{R}^d \right\}$$

ahol Σ egy pozitív-definit mátrix.

Paraméter–optimizálás lépései:

- A $p_{post}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{F})$ a-posteriori eloszlás és az Ω eloszláscsalád meghatározása;
- $d_{KL}\left(p_{post}(\boldsymbol{\theta})|p(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)\right)$ kiszámítása – $(\boldsymbol{\mu}, \Sigma)$ vált.;
- $(\hat{\boldsymbol{\mu}}, \hat{\Sigma})$ meghatározása:

  $$(\hat{\boldsymbol{\mu}}, \hat{\Sigma}) \Rightarrow \underset{p \in \Omega}{\operatorname{argmin}}\, d_{KL}\left(p_{post}(\boldsymbol{\theta})|p(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)\right)$$

Bayes–becslés predikció

Feltételezzük, hogy

- meghatároztuk a legjobban közelítő $\hat{p}(\boldsymbol{\theta})$ eloszlást.
- ismerjük a bemenet–kimenet összefüggéseket: a

  $P(y|\mathbf{x}, \boldsymbol{\theta})$ felt. valószínűséget.

Predikció:

Adott bemenetre mi lesz a kimeneti értékek eloszlása?

Ha $P(y|\mathbf{x}, \boldsymbol{\theta})$ **eloszlás az** y **szerint**, akkor a prediktív disztribúció:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int_{\Omega_{\boldsymbol{\theta}}} d\boldsymbol{\theta} \hat{p}(\boldsymbol{\theta}|\mathcal{D})\; P(y^*|\mathbf{x}^*, \boldsymbol{\theta})$$

Bayes–predikció — predikció II

Prediktív disztribúció:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int_{\Omega_{\boldsymbol\theta}} d\boldsymbol\theta \hat{p}(\boldsymbol\theta|\mathcal{D})\; P(y^*|\mathbf{x}^*, \boldsymbol\theta)$$

A Gyakorlatban:

- fontos a modell választása: ha $\mathcal{M}$ a modell-család, akkor minden valószínűség függ az $\mathcal{M}$-től:

$$\hat{p}(\boldsymbol\theta|\mathcal{D}) \quad \stackrel{\text{def}}{=} \quad \hat{p}(\boldsymbol\theta|\mathcal{D}, \mathcal{M})$$

- A $p_{post}(\boldsymbol\theta)$-hoz hasonlóan a prediktív eloszlás sem írható fel analitikusan: közelítések szükségesek.
- A prediktív eloszlást általában a $\hat{p}(\boldsymbol\theta)$-hoz hasonló módszerekkel keressük.

Bayes–modellek — Példa

Függvénycsalád: legyen az

$$\mathcal{F} = \left\{ f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{k=0}^{6} \theta_k \alpha_k \mathbf{x}^k \;\middle|\; \theta_k \sim N(0,1),\; \alpha_k = \sqrt{\binom{6}{k}} \right\}$$

ahol $\boldsymbol\theta = [\theta_1, \ldots, \theta_6]^T$ a függvény paraméterei.

Ekkor átlagban:

$$\langle f(\mathbf{x})\rangle_{\theta_0,\ldots,\theta_6} = \left\langle \sum_{k=0}^{6} \theta_k \alpha_k \mathbf{x}^k \right\rangle_{\theta_0,\ldots,\theta_6} = \sum_{k=0}^{6} \mathbf{0}\alpha_k \mathbf{x}^k = \mathbf{0}$$

illetve ...

Bayes–modellek — Példa II

$$\begin{aligned}\langle f(\mathbf{x}_1) f(\mathbf{x}_2)\rangle_{a_0,\ldots,a_6} &= \left\langle \left(\sum_{k=0}^{6} \theta_k \alpha_k \mathbf{x}_1^k\right)\left(\sum_{k=0}^{6} \theta_k \alpha_k \mathbf{x}_2^k\right)\right\rangle \ldots \\ &= \sum_{k=0}^{6} 1 \alpha_k^2 \mathbf{x}_1^k \mathbf{x}_2^k = \sum_{k=0}^{6} \binom{6}{k} (\mathbf{x}_1\mathbf{x}_2)^k \\ &= (1 + \mathbf{x}_1\mathbf{x}_2)^6\end{aligned}$$

$\mathcal{F}$ tehát egy függvényosztály, melynek átlaga 0 és szórása fent látható.

Feladat:
Közelítsünk a Bayes–modell segítségével.

Bayes–példa — Zaj

Szükséges a zaj ismerete; Feltételezzük, hogy az Gauss eloszlású, tehát:

$$P(y|f(\mathbf{x}, \boldsymbol\theta), \sigma_o) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(y - f(\mathbf{x}))^2}{2\sigma_o^2} \right]$$

ahol σ_n a zaj (noise) szórása.

Az adatok: $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

Feltételes valószínűségük:

$$P(\mathcal{D}|\boldsymbol\theta, \sigma_o) = \prod_{n=1}^{N} P(y_n|f(\mathbf{x}_n, \boldsymbol\theta), \sigma_n)$$

Bayes–példa — a–post I

A–posteriori eloszlás:

$$p_{post}(\boldsymbol{\theta}|\mathcal{D},\sigma_o) = \frac{P(\mathcal{D}|\boldsymbol{\theta},\sigma_o)\; p_0(\boldsymbol{\theta})}{\int d\boldsymbol{\theta}\; P(\mathcal{D}|\boldsymbol{\theta},\sigma_o)p_0(\boldsymbol{\theta})}$$

ahol $p_0(\boldsymbol{\theta})$ a változók feltételezett – a–priori – eloszlása.

Megjegyzések:

- Normál a–priori eloszlás esetén:

$$\log p_0(\boldsymbol{\theta}) \propto -\frac{\|\boldsymbol{\theta}\|^2}{2\sigma_p^2}$$

  azaz regularizációs megkötés a paramétereken.
- A modell a paraméterekben lineáris $\Rightarrow$ gaussz-eloszlás lesz az eredmény is.
- a jobb oldalon eloszlás van, a nevező normalizáló $\Rightarrow$ annak értékét nem kell kiszámítani.

Bayes–példa — a–post II

$$-2\log p_{post}(\boldsymbol{\theta}|\mathcal{D},\sigma_o) = \log p(\mathcal{D}|\mathcal{M}) + \sum_{n=1}^{N}\frac{(y_n - f(\mathbf{x}_n|\boldsymbol{\theta}))^2}{\sigma_n^2} + \frac{\|\boldsymbol{\theta}\|^2}{\sigma_p^2}$$

ahol $\log p(\mathcal{D}|\mathcal{M})$ a normalizáló konstans.
Jelölések:

$$\hat{\mathbf{x}} \stackrel{\text{def}}{=} [\mathbf{x}^0,\mathbf{x}^1,\dots,\mathbf{x}^6]^T.$$

$$f(\mathbf{x}_n|\boldsymbol{\theta}) \stackrel{\text{def}}{=} \boldsymbol{\theta}^T\mathbf{x}_n \qquad \|\boldsymbol{\theta}\|^2 = \boldsymbol{\theta}^T\boldsymbol{\theta}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} x_1 0 & \dots & x_1^6 \\ \vdots & \ddots & \vdots \\ x_N^0 & \dots & x_N^6 \end{bmatrix}$$

A fenti jelölésekkel az összeg szorzattá alakul. Csoportosítva:

$$\boldsymbol{\theta}^T\left(\frac{\mathbf{X}^T\mathbf{X}}{\sigma_n^2} + \frac{\mathbf{I}_7}{\sigma_p^2}\right)\boldsymbol{\theta} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + K_1$$

Bayes–példa — a–post III

$$\boldsymbol{\theta}^T\left(\frac{\mathbf{X}^T\mathbf{X}}{\sigma_n^2} + \frac{\mathbf{I}_7}{\sigma_p^2}\right)\boldsymbol{\theta} \quad -\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} \quad +K_1$$
$$\boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta} \quad -\mathbf{b}^T\mathbf{A}\boldsymbol{\theta} \quad +K_1$$

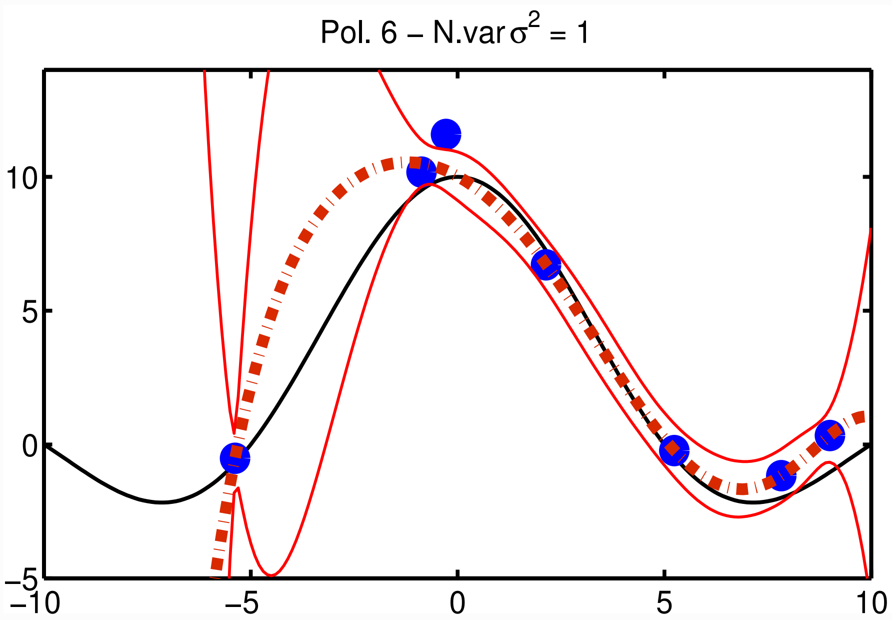
ahol $\mathbf{A} \stackrel{\text{def}}{=} \frac{\mathbf{X}^T\mathbf{X}}{\sigma_n^2} + \frac{\mathbf{I}_7}{\sigma_p^2}$; $\mathbf{b} \stackrel{\text{def}}{=} \mathbf{A}^{-1}\mathbf{X}^T\mathbf{y}$

A $\log p_p ost(\boldsymbol{\theta}|\mathcal{D},\sigma_o)$ tehát tartalmaz egy teljes négyzetet és egy konstanst.

Mivel a teljes négyzet Gauss-eloszlást jelent, a konstans értékét ismerjük. Az eredmény:

$$\log p_p ost(\boldsymbol{\theta}|\mathcal{D},\sigma_o) = \mathcal{N}\left(\boldsymbol{\theta}|\mathbf{b},\mathbf{A}^{-1}\right)$$

Bayes–példa — Rajz



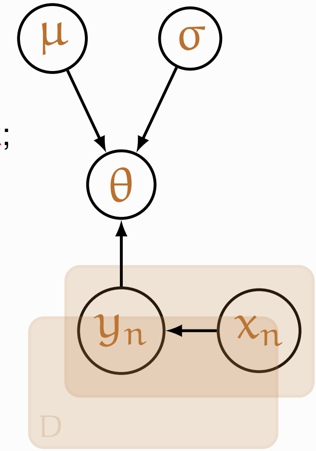
Becslések — Összefoglaló

- Max–Lik. becslésnél: nincs a–priori eloszlás, a becslés esetenként rossz.
- M.A.P. becslésnél: eloszlásokról beszélünk, azonban a becslés eredménye nem valószínűségi, hanem egy érték.
- Bayes–becslésnél: a becslés eredménye egy valószínűségi eloszlás.

Grafikus Modellek — Példa

A paraméterek függőségi gráfja a regressziós példánál.

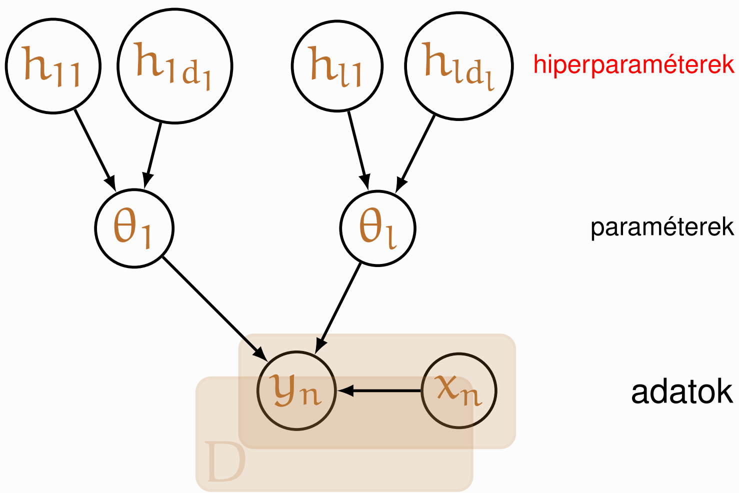
Lényeges, hogy a nem a θ paramétert becsüljük; hanem a $p(\theta)$ eloszlás paramétereit,
$\Longrightarrow$ jelen esetben ezek μ és σ.



Grafikus modelleket akkor használunk, ha egy modell egyes paramétereit megfigyeljük – pl. (x_n, y_n) – másokra meg következtetni kell.

A következtetés alapja a megfigyelések és a **modell**.

Grafikus Modellek .



Vizsgatematika I

- M.I – racionális vs. imitáló illetve cselekvő/gondolkodó.
- Tudásreprezentáció – állapottér, feladat definíció.
- Megoldás-keresések az állapottérben: hegymászó, backtracking.
- Gráfkeresés – kapcsolata a M.I. rendszerekkel.
- Dekompozíciós módszer.
- Predikátumkalkulus, rezolúció.
- Gráfok, utak, irányított gráfok, /Sigma/delta/ tulajdonságok.
- ÉS/VAGY gráfok, hiperutak és-vagy gráfokban.
- ÉS/VAGY gráfok átalakítása irányított gráffá.
- Játékok gráfjai, puzzle, 4 királynő.
- Gráfkereső alapalgoritmus.
- Gráfkereső algoritmus tulajdonságai.
- Keresési algoritmusok: mélységi, szélességi, egyenletes, előretekintő.
- A-* algoritmus család, tulajdonságok.
- Szemantikus hálók, tulajdonságok, operációk.
- Keretrendszerek, démonok.

Vizsgatematika II

- Játékok: keresés, kétszemélyes játékok, stratégia.
- NIM játék: leírás, nyerő stratégia (b).
- Nyerő stratégia teljes információs játékoknál (b).
- Minimax algoritmus, minimax tétel (b).
- Minimax, negamax algoritmusok.
- Alfa-béta vágás.
- Bizonyt.ság: Bayes-modell.
- Bayes-hálók, grafikus modellek.
- Dempster-Shafer modell, Bel, Pl és $\mathfrak{m}$ függvények.
- Fuzzy logika, $\sim$ rendszerek.
- Tanulás: induktív és/vagy deduktív rendszerek.
- Tanuló rendszerek típusai.
- Exploration vs. exploitation – visszacsatolásos tanulás.
- Genetikus algoritmusok, hamiltoni rendszerek,
- Neurális hálózatok, topológiák.
- Neurális hálózatok tanítási szabályai, optimizálás.