# Bizonytalanság

## Cselekvés bizonytalan tudás esetén

Miért nem jó az első rendű logika:

* Lustaság: túl nagy munkát jelent az ok és okozat eseményhalmazának felsorolása
* Elméleti tudatlanság: nem tudunk eleget a probléma leírásához (pl.: orvosi tudományterület nem teljes)
* Gyakorlati tudatlanság: nem teljesen hozzáférhető a környezet

Az ágens tudása csak bizonyos mértékű meggyőződést nyújthat az adott kijelentéssel kapcsolatban. A meggyőződés értékek kezelésére az elsődleges eszközünk a valószínűség-számítás lesz.

Egy adott kijelentéshez rendelt 0 valószínűség annak az egyértelmű meggyőződésnek felel meg, hogy a mondat állítása hamis, az 1 valószínűség annak a meggyőződésnek felel meg, hogy az mondat igaz. Az állítás persze *valójában* vagy igaz vagy hamis.

meggyőződés mértéke igazságtartalom mértéke

Megj.: Az igazság értéke, mint a meggyőződés mértékének az ellentéte, a fuzzy logika tárgya.

Az észlelések alkotják azt a **tényt** vagy tényállást (evidence), amelyen a valószínűségi kijelentések alapulnak. Minden valószínűségi kijelentésnek hivatkoznia kell azokra a tényekre, amelyek alapján az adott valószínűség az állításhoz lett rendelve. Amint egy ágens új észlelések birtokába jut, ezek figyelembevételével módosítja a valászínűségek becslését.

**a priori valószínűség:** előzetes valószínűség, mielőtt tények birtokába jutunk. Csak akkor használható, ha nincs semmilyen más információ a birtokunkban.

**a posteriori valószínűség:** utólagos valószínűség, miután a tények birtokába jutottunk

Ahhoz, hogy dönteni tudjunk két terv között, fel kell állítani egy **preferencia-sorrendet** a különböző tervek lehetséges kimenetei között. A sorrend felállítására és következtetésre a hasznosságelméletet fogjuk használni.

**Hasznosságelmélet:** minden állapotnak van egy hasznavehetőségi mértéke vagy haszna az ágens számára, és a számára több hasznot hozó állapotokat fogja előnyben részesíteni.

döntéselmélet = valószínűség-elmélet + hasznosságelmélet

**Maximális Várható Haszon (MVH) elve:** egy ágens akkor és csak akkor racionális, ha olyan cselekvést választ, amely az adott cselekvés összes lehetséges kimenetelére átlagolt legmagasabb várható hasznot hozza.

## Valószínűségi alapfogalmak

Miután felállítottuk a racionális ágens általános keretét, szükségünk lesz egy formális nyelvre a bizonytalan tudás leírásához és a következtetéshez. Minden olyan jelölésnek, amely a meggyőződésünk fokának leírására szolgál, képesnek kell lennie két fő dolog kezelésére:

* kijelentések jellege, amelyekhez meggyőződési mértéket akarunk rendelni
* meggyőződés mértékének az ágens tapasztalatától való függése

**állítások:** a hiedelmi mértékeket mindig állításokhoz rendeljük.

**valószínűségi vagy véletlen változó:** a nyelv alapeleme, a kezdetben ismeretlen „állapotú” világ egy „részére” vonatkozik*. Pl.: Lyuk, a bal alsó bölcsességfogam esetleges lyukasságát mutatja*

**értéktartomány (domain):** amelyből a valószínűségi valátozó az értékeit veheti. *Pl.: Lyuk esetén <igaz, hamis>*

**Értéktartomány fajták:**

* (Boole-típusú) logikai véletlen változók *Pl.: <igaz, hamis>*
* Diszkrét véletlen változók *Pl.: <napos, esős, felhős, havazik>*
* Folytonos véletlen válzozók

**elemi esemény:** a világ – amely tekintetében az ágens bizonytalan – állapotának egy *teljes* leírását jelenti.

**elemi esemény tulajdonságai:**

* egymást kölcsönösen kizáró
* Az összes elemi esemény halmaza kimerítő – legalább az egyiknek igaznak kell lennie.
* maga után vonja következményként az összes állítás igazságát vagy hamisságát
* Bármely állítás logikailag egyenértékű azon elemi események diszjunkciójával, amelyekből az állítás következik.

**valószínűség-eloszlás:** a valószínűségi változó minden egyes állapotához rendelt valószínűségi értékekből képzett vektort jelöli*. Pl.: P(Időjárás) = ⟨0,7, 0,2, 0,08, 0,02⟩*

**együttes valószínűség-eloszlás:** valószínűségi változóhalmaz összes lehetséges kombinációjának valószínűségei.

**teljes együttes valószínűség-eloszlás:** minden egyes elemi esemény valószínűségét, és így a kérdéses világgal kapcsolatos összes bizonytalanságot meghatározza.

**valószínűség-sűrűségfüggvény:** a folytonos valószínűségi változókra vonatkozó valószínűségi eloszlás

**feltételes valószínűség:** ha az ágens bizonyos tények birtokába jut a korábban ismeretlen, a tartományra jellemző véletlen változóra vonatkozóan, az a priori valószínűségek helyett a posteriori, feltételes valószínűséget kell használni.

A feltételes valószínűségek megadhatók feltétel nélküliek segítségével:

az utóbbi a szorzat szabály.

## Valószínűségi axiómák

Kolmogorov-axiómák:

1. Minden valószínűség 0 és 1 közé esik.
2. A biztosan igaz (azaz érvényes) állítások valószínűsége 1, a biztosan hamis (azaz kielégíthetetlen) állításoké pedig 0
3. A [diszjunkció](http://mialmanach.mit.bme.hu/fogalomtar/diszjunkcio) valószínűsége P(a ∨ b) = P(a) + P(b) – P(a ∧ b)

Egyetlen változó tetszőleges valószínűségi eloszlása összegzésének 1-et kell adnia.

Változók tetszőleges *halmazának* bármely együttes valószínűségi eloszlását összegezve, eredményül szintén 1-et kapunk.

## Teljes együttes valószínűség-eloszláson alapuló következtetés

**peremeloszlás:** a változók egy részhalmaza vagy egyetlen változó fölötti valószínűségi eloszlás kifejezése adja.

**kiátlagolás, marginalizálás:**

**normalizáló konstans:** biztosítja, hogy a feltételes valószínűségek összege 1 legyen. Jelölése:

## Függetlenség

**függetlenség**: Két valószínűségi változó független, ha a változókra igazak a következők:

 P(a∣b) = P(a) vagy P(b∣a) = P(b) vagy P(a ∧ b) = P(a)P(b)

## A Bayes-tétel és használata

A szorzat szabályból következik, hogy

*P*(*a* ∧ *b*) = *P*(*a* | *b*)*P*(*b*)

*P*(*a* ∧ *b*) = *P*(*b* | *a*)*P*(*a*)

A jobb oldalak egyenlőségéből P(a)-val való osztás után következik, hogy

A normalizált Bayes-tétel általános alakja

**feltételes függetlenség:**

Ha igaz a következő:  *P*(*X*, *Y*∣*Z*) = *P*(*X*∣*Z*)*P*(*Y*∣*Z*)

# Valószínuségi következtetés

## A tudás reprezentálása bizonytalanság esetén

**Bayes-háló:** egy irányított gráf, amelyben minden *csomóponthoz* számszerű valószínűségi információk vannak csatolva.

A teljes megadás a következő:

1. A háló csomópontjait valószínűségi változók egy halmaza alkotja. A változók lehetnek diszkrétek vagy folytonosak.
2. Irányított élek (nyilak) egy halmaza összeköt bizonyos csomópontpárokat. Ha létezik nyíl az X csomóponttól az Y csomópontig, azt mondjuk, hogy az X a szülője az Y-nak.
3. Minden Xi csomóponthoz tartozik egy P(Xi∣Szülők(Xi)) feltételes valószínűség-eloszlás, ami számszerűen megadja a szülők hatását a csomóponti változóra.
4. A gráf nem tartalmaz irányított kört (azaz irányított, körmentes gráf – Directed, Acyclic Graph, DAG)

A háló topológiája – a csomópontok és élek halmaza – megadja a tárgyterületen fennálló feltételes függetlenségi kapcsolatokat. Egy helyesen létrehozott hálóban az X csomópontot az Y csomóponttal összekötő nyíl *intuitív* jelentése rendszerint az, hogy az *X-nek közvetlen befolyása van az Y-ra*. A tárgyterület szakértője számára általában könnyű eldönteni, hogy milyen közvetlen befolyások teljesülnek egy adott területen. Ha a Bayes-háló topológiája kész, már csak az egyes változókhoz tartozó feltételes valószínűség-eloszlásokat kell meghatározni a szülőkkel mint feltételekkel. A topológia és a feltételes eloszlások együttese elegendő, hogy megadja (implicit módon) az összes változó feletti együttes valószínűség-eloszlás függvényt.



Megj.: a hálóban nem szerepel minden tény, mert azoknak egy részét nehézkes lenne számolni, másik része pedig felesleges bonyodalmat okozna. Ezen tények egy bizonyos pontosságú közelítése jelenik meg a változókhoz rendelt valószínűségekben.

**feltételes valószínűségi táblázat (FVT):** a táblázatban minden sor az egyes csomóponti értékek feltételes valószínűségét tartalmazza az adott sorhoz tartozó szülői feltétel esetén. Az egyes sorokban szereplő számok összegének 1-et kell adnia. Bináris változók esetén, gyakran a második számot elhagyjuk.

## A Bayes-hálók szemantikája

Az együttes valószínűség-eloszlás függvény egy bejegyzésének értékét a következő összefüggés adja meg:

**Bayes-hálók építése**: először az „alapvető okokat” adjuk a hálóhoz, majd a változókat, amelyeket befolyásolnak, és ezt addig folytatjuk, amíg el nem érjük a „leveleket”, amelyeknek már nincs közvetlen okozati hatása más változókra.

Megj.: Lehetséges más (de ugyanezen együttes valószínűség-eloszlás függvénynek reprezentálása alkalmas) felépítés is, azonban akkor fennáll annak a veszélye, hogy nem reprezentálja az összes feltételes függetlenségi relációt, és így rengeteg szükségtelen érték meghatározására kényszerül.

**“Numerikus” szemantikából következő feltételes függetlenség**: egy csomópont feltételesen független az őt megelőzőktől, ha a csomópont szülői adottak.

**Topológikus szemantikából következő feltételes függetlenség:**

1. Egy csomópont feltételesen független a nem leszármazottaitól feltéve, hogy a szülei adottak
2. Egy csomópont feltételesen független az összes többi csomóponttól a hálózatban, a szülei, gyermekei és gyermeki szüleinek az ismeretében – azaz a Markov-takarójának ismeretében.



## Feltételes eloszlások hatékony reprezentációja

**determinisztikus csomópont:** értékét a szüleinek az értéke teljesen meghatározza, mindenfajta bizonytalanságtól mentesen. *Pl.: lehet logikai kapcsolat vagy numerikus kapcsolat*

**zajos-VAGY:** megenged bizonytalanságot, hogy egyes szülők okozhatják-e a gyermekek igaz értékét – az okozati kapcsolat a szülő és gyermek között gátolt lehet.

A modell két feltevésre épül:

1. feltételezi, hogy az összes lehetséges ok fel van sorolva
2. felteszi, hogy bármely szülő gátlása független a többi szülő gátlásától

**Bayes-hálók folytonos változóiank kezelése:**

* diszkretizálással
* valószínűség sűrűségfüggvénnyel

**hibrid Bayes-háló:** diszkrét és folytonos változókat is tartalmazó háló

## Egzakt következtetés Bayes-hálókban

**Következtetés felsorolással:**

Feladat: a célváltozók (query variables) egy halmazának a posteriori valószínűség-eloszlásának kiszámítása, egy adott megfigyelt esemény (event) esetén – azaz bizonyítékváltozók (evidence variables) egy halmazához történő érték-hozzárendelés esetén.

**Jelölések**:

X : célváltozó

**E**: bizonyítékváltozók halmaza

e: megfigyelt esemény

**Y**: rejtett változók, azaz nem-bizonyítékváltozók halmaza

**X:** változók teljes halmaza, **X** = {X} ∪ **E** ∪ **Y**

**P**(X∣**e**): egy jellemző posteriori eloszlásra irányuló lekérdezés

Egy lekérdezés megválaszolható a Bayes-háló felhasználásával, kiszámítva a hálóból származó feltételes valószínűségek szorzatainak az összegét.

**Példa:** Vegyük a P(Betörés∣JánosTelefonál = igaz, MáriaTelefonál = igaz) lekérdezést. A rejtett változók ennél a kérdésnél a Földrengés és a Riasztás.

A Bayes-hálók szemantikája pedig az FVT-bejegyzések felhasználásával egy kifejezést ad. Az egyszerűség kedvéért ezt csak a Betörés = igaz esetére adjuk meg:



**változó elimináclós algoritmus:** egyenlet típusú kifejezéseket jobbról sorrendben értékeli ki. A köztes eredményeket eltároljuk, és bármely változó feletti összegzés a kifejezésnek csak azon része felett történik meg, amely függ ettől a változótól.

*Pl.:*



**Következmény:** minden változó, ami nem őse a *célváltozónak* vagy egy *bizonyítékváltozónak*, irreleváns a lekérdezésre.

**egyszeresen összekötött hálók (polifák):** háló bármely két csomópontja között legfeljebb egyetlen irányítatlan út létezik. Az egzakt következtetés idő- és tárkomplexitása a polifákban a háló méretében *lineáris*.

**többszörösen összekötött hálók:** változó eliminálás legrosszabb esetben *exponenciális* idő- és tárkomplexitású lehet, még akkor is, ha a csomópontonkénti szülők száma korlátos.

**Csoportosító algoritmus:** A csoportosítás alapötlete, hogy a háló önálló csomópontjait egyesítjük, klasztercsomópontokat formálva úgy, hogy a kiadódó háló polifa legyen

## Közelítő következtetés Bayes-hálókban

### Közvetlen mintavételezési módszerek:

Az alapelem minden mintavételi algoritmusban a minták generálása egy ismert valószínűség-eloszlásból. A véletlen mintavételezési folyamat legegyszerűbb fajtája Bayes-hálók esetén a hálóból generál olyan eseményeket, amelyekhez nem kapcsolódik bizonyíték. Az ötlet az, hogy mintavételezzünk minden változót egymás után, topológiai sorrendben. A valószínűség-eloszlás, amiből az értéket mintavételezzük, feltételesen függ a változó szüleihez már hozzárendelt értékektől.



Bármilyen mintavételi algoritmusban a válasz kiszámítása a generálás során előálló minták megszámlálása alapján történik. Tételezzük fel, hogy N teljes mintánk van, és jelölje N(x1, …, xn) az x1, …, xn esemény gyakoriságát. Azt várjuk, hogy ez a gyakoriság határértékben konvergáljon a várható értékéhez a mintavételi valószínűség szerint:



**Elutasító mintavételezés Bayes-hálókban:**

Először a háló által megadott a priori eloszlásból generál mintákat, majd elutasítja azokat, amelyek nem illeszkednek a bizonyítékhoz. Végül, a becslés megkapható az X = x előfordulásainak megszámlálásával a megmaradt mintában.

Az elutasító mintavétel legnagyobb hibája, hogy nagyon sok mintát utasít el! Az e bizonyítékkal konzisztens minták aránya exponenciálisan egyre kevesebb, ahogy a bizonyítékváltozók száma nő, így az eljárás egyszerűen használhatatlan komplex problémákban.

**Valószínűségi-súlyozás:**

Elkerüli az elutasító mintavételezés gyengeségét azáltal, hogy csak az e bizonyítékkal konzisztens eseményeket generál.

A valószínűségi-súlyozás rögzíti az E bizonyítékváltozók értékeit, és csak a maradék X és Y változókat mintavételezi. Ez garantálja, hogy minden generált esemény konzisztens a bizonyítékkal. Azonban nem minden esemény egyenlő. Mielőtt megállapítanánk a számlálási eredményeket a célváltozó eloszlásában, minden eseményt súlyozunk azzal a valószínűséggel, amely megadja, hogy az esemény mennyire van összhangban a bizonyítékkal. Ezt a valószínűséget az egyes bizonyítékváltozók feltételes valószínűségeinek a szorzatával mérjük, a szüleik ismeretében. Szemléletesen, azoknak az eseményeknek, ahol a bizonyíték valószínűtlennek tűnik, kisebb súlyt kell adni.

Mivel a valószínűségi súlyozás az összes generált mintát felhasználja, sokkal hatékonyabb lehet, mint az elutasításos mintavétel. Azonban a teljesítménye leromlik, amint a bizonyítékváltozók száma növekszik.



### Következtetés Markov-lánc szimulációval

**Az MCMC (Markov chain Monte Carlo) algoritmus:**

Minden eseményt az azt megelőző esemény véletlen módosításával generál. Ezért a hálót hasznos úgy elképzelni, mint aminek van egy konkrét jelenlegi állapota, ami minden változóra meghatároz egy értéket. A következő állapot generálása egy Xi nem bizonyítékváltozóhoz tartozó érték véletlenszerű mintavételezésével történik, az Xi Markov-takarójába tartozó változók jelenlegi értékeinek feltétele mellett. Az MCMC így véletlen bolyongást végez az állapottérben – a lehetséges teljes érték hozzárendelések terében –, egyszerre egy változót billentve át, de rögzítetten tartva a bizonyítékváltozókat.

**Markov-lánc**: Legyen q(x⟶x′) az a valószínűség, hogy a folyamat x állapotból x′ állapotba lép át. Ez az átmenet-valószínűség definiálja az úgynevezett Markov-láncot az állapottéren.

**stacionárius eloszlás:**

Tegyük fel most, hogy t lépésnyit futtatjuk a Markov-láncot, és legyen πt(x) annak a valószínűsége, hogy a rendszer a t időpillanatban az x állapotban van. Hasonlóan, legyen πt+1(x′) annak a valószínűsége, hogy a rendszer t + 1 időpillanatban x′ állapotban van. Ismerve πt(x)-et a πt+1(x′) kiszámítható úgy, hogy minden olyan állapotra, amiben a rendszer t időpillanatban lehet, összegezzük az állapot valószínűségének és az ebből az állapotból az x′-be való átlépés valószínűségének szorzatát:

 

Akkor mondjuk, hogy egy lánc elérte **a stacionárius eloszlását**, ha πt = πt+1.



## Egyéb módszerek a bizonytalan környezetben történő következtetéshez

Bizonytalansági következtetés szabályalapú eljárásokkal

**Logikai rendszerek tulajdonságai:**

* Lokalitás
* Leválasztás
* Igazságfüggvény

Számos kísérlet történt olyan bizonytalansági következtetések kifejlesztésére, ahol ezek az előnyös tulajdonságok megmaradnak, a rossz hír a szabályalapú rendszerek számára, hogy a lokalitás, a leválasztás és az igazságfüggvény egyszerűen nem alkalmas a bizonytalansági következtetés számára.

Mégis sikeresen használták a módszert számos orvosi diagnosztikai programban.

### Az ismerethiány reprezentálása: a Dempster–Shafer-elmélet

Azzal a céllal hozták létre, hogy megkülönböztethetővé váljon a bizonytalanság (uncertainty) és az ismerethiány (ignorance). Ez az elmélet egy állításnak nem a valószínűségét számítja ki, hanem helyette azt, hogy mennyi annak a valószínűsége, hogy a bizonyíték támogatja az állítást. A bizonyosságnak ezt a mértékét bizonyosságfüggvénynek (belief function) nevezik, és Bel(X)-szel jelölik.

### A meghatározatlanság reprezentálása: fuzzy halmazok és logikák

**fuzzy logika:** egy eljárás olyan logikai kifejezésekkel való következtetésre, amelyek fuzzy halmazbeli tagsági állításokat írnak le

# Időbeli valószínűségi következtetés

## Idő és bizonytalanság

Egy adott problémánál az állapotváltozók és a bizonyítékváltozók halmazának meghatározása után a következő lépés a változók közötti függőségek megadása. Egy nyilvánvaló választás, hogy a változókat a természetes idősorrendjük szerint sorrendezzük, mivel az ok általában megelőzi a hatást, és a változókat lehetőleg az ok-okozati sorrendjük szerint vesszük.

**Probléma**:

1. korlátlan számú feltételes valószínűségi táblát kell megadnunk minden változóra, minden időpillanatban
2. ezek korlátlan számú szülőt tartalmazhatnának

**Megoldás:**

1. Az első problémát megoldja annak feltételezése, hogy a világ állapotának a változásait egy stacionárius folyamat (stationary process) okozza.

**stacionárius folyamat:** egy változási folyamat, amit olyan törvények határoznak meg, amik maguk nem változnak az idővel.

1. A második problémát, a potenciálisan végtelen számú szülő kezelését, az úgynevezett Markov-feltétel elfogadása oldja meg.

**Markov-feltétel**: a jelenlegi állapot a korábbi állapotoknak csak *véges* történetétől függ.

**elsőrendű Markov-folyamat:** a jelenlegi állapot csak az előző állapottól függ, és nem függ egyetlen korábbitól sem:

*P*(*X*t|*X*0:*t*–1) = *P*(*X*t|*Xt*–1)

Az Xt állapotváltozók szüleinek a korlátozásán túl korlátoznunk kell az Et bizonyítékváltozók szüleit is. Tipikusan feltesszük, hogy a bizonyítékváltozók egy t időpillanatban csak az aktuális állapottól függnek:

*P(E*t|*X*0:*t*, *E*0:*t*–1) = *P*(*E*t|*X*t)

**érzékelő modell:** A **P**(**E**t|**X**t) feltételes eloszlás.

Az állapotátmenet-modellhez és az érzékelő modellhez még meg kell adnunk egy P(X0) a priori eloszlást a 0. időpontbeli állapotok felett.

A **teljes együttes eloszlás meghatározása**kor, az összes változó felett, bármely véges t-re azt kapjuk, hogy



Ha a Markov-feltételből adódó közelítés túl pontatlannak bizonyul:

1. lehetőség: a Markov-folyamat rendjének a megnövelése.
2. lehetőség: az állapotváltozók halmazának megnövelése.

*Példa:*



## Következtetés időbeli modellekben

**Alapvető feladatok:**

1. szűrés
2. előrejelzés
3. simítás
4. legvalószínűbb magyarázat

**1. szűrés:** Ez a bizonyossági állapot kiszámításának a feladata – ami a jelenlegi állapot feletti a posteriori eloszlás, az adott időpontig vett összes bizonyíték ismeretében. Azaz szeretnénk kiszámítani a P(Xt|e1:t) mennyiséget, feltéve, hogy a bizonyítékok folyamatos sorozatban érkeznek kezdve a t = 1 időponttól.

**2.** **Előrejelzés:** Ez egy *jövőbeli* állapot feletti a posteriori eloszlás kiszámításának a feladata, az adott időpontig vett összes bizonyíték ismeretében. Azaz, szeretnénk kiszámítani a **P**(**X**t+k|**e**1:t) mennyiséget valamely k > 0 esetén.

**3. Simítás:** Ez egy múltbeli állapot feletti a posteriori eloszlás kiszámításának a feladata, a jelen időpontig vett összes bizonyíték ismeretében. Azaz szeretnénk kiszámítani a **P**(**X**k|**e**1:t) mennyiséget valamely 0 ≤ k < t esetén. A visszatekintés az állapotnak egy jobb becslését adja, mint ami akkor elérhető volt, mivel több bizonyítékot használ fel.

**4. Legvalószínűbb magyarázat:** A megfigyelések egy sorozatának ismeretében lehet, hogy szeretnénk megtalálni azt az állapotsorozatot, ami a leginkább valószínű, hogy az adott megfigyeléseket generálta.

## Dinamikus Bayes-hálók

**dinamikus Bayes-háló (DBH):** egy olyan Bayes-háló, ami időbeli valószínűségi modellt reprezentál.

Egy DBH **létrehozás**ához háromfajta információt kell megadni:

* a **P**(**X**0) a priori eloszlást az állapotváltozók felett
* a **P**(**X**t+1|**X**t) állapotátmenet-modellt
* a **P**(**E**t|**X**t) érzékelő modellt

# Megfigyelések alapján történő tanulás

## Tanulási formák

A tanuló komponens tervezését három dolog befolyásolja alapvetően:

1. A cselekvő elem mely *komponenseit* akarjuk tanítani.
2. Milyen *visszacsatolás* áll rendelkezésre ezen komponensek tanítására.
3. Hogyan *reprezentáljuk* a komponenseket.

Az ágenseknek a **komponensei** a következőket tartalmazzák:

1. Az aktuális állapot feltételeinek közvetlen leképezése cselekvésekre.
2. Olyan lehetőség, amely egy megfigyelési szekvenciából a világ releváns tulajdonságaira képes következtetni.
3. A világ alakulására vonatkozó, valamint az ágens lehetséges cselekvéseinek következményeit leíró információ.
4. A világ lehetséges állapotainak számunkra kívánatos voltát megadó hasznosság- információ.
5. Cselekvés-hasznosság információ, amely az egyes cselekvések kívánatosságát jelzi számunkra.
6. Célok: ezek olyan állapotosztályokat adnak meg, amelyek elérése az ágens hasznosságát maximálja.

**Visszacsatolás** jellege lehet:

* ellenőrzött
* nem ellenőrzött
* megerősítéses

**ellenőrzött tanulás:** egy leképezésnek a bemeneti és kimeneti minták alapján történő megtanulását jelenti. Teljesen megfigyelhető környezet esetén mindig fennáll a lehetőség, hogy az ágens megfigyeli a cselekvésének következményeit, így használhat ellenőrzött tanulási módszereket annak érdekében, hogy megjósolja azokat. Részlegesen megfigyelhető környezetben nehezebb a probléma, mert a közvetlen hatások láthatatlanok maradhatnak.

**nem ellenőrzött tanulás:** a bemeneti minták tanulása történik, de a kimeneti kívánt minták nem biztosítottak.

**megerősítéses tanulás:** helyett hogy egy tanító útmutatását követhetné, egy megerősítéses tanulást végző ágensnek megerősítési információ alapján kell tanulnia. A megerősítéses tanulás tipikusan magában foglalja azt a részproblémát, hogy az ágensnek meg kell tanulnia azt is, hogyan működik a világ.

## Induktív tanulás

**tisztán induktív következtetés:** Adott az f-re vonatkozó minták egy halmaza, ennek alapján határozzunk meg egy h (*hipotézis*) függvényt, amely közelíti f-et.

**konzisztens hipotézis:** valamennyi adatra illeszkedő hipotézis függvény

**Ockham borotvája elv:** részesítsük előnyben a *legegyszerűbb* olyan hipotézist, amely konzisztens az adatokkal.

Nemdeterminisztikus függvények esetén elkerülhetetlen, hogy kompromisszumot kössünk a hipotézis komplexitása és az adatokhoz való illeszkedés pontossága között.

**realizálható probléma:** ha a hipotézistérnek eleme az igazi függvény.

## Döntési fák megalkotása tanulással

**döntési fa:** bemenetként egy attribútumokkal (attributes) leírt objektumot vagy szituációt kap, és egy „döntést” ad vissza eredményként – a bemenetre adott válasz jósolt értékét.

**osztályozás:** egy diszkrét értékkészletű függvény tanuálása

**regresszió:** egy folytonos függvény tanulása

A döntési fa egy tesztsorozat elvégzése során jut el a döntéshez. A fa minden egyes belső csomópontja valamely tulajdonság értékére vonatkozó tesztnek felel meg, a csomópontból kilépő ágakat pedig a teszt lehetséges kimeneteivel címkézzük. Minden egyes levélcsomópont megadja azt az értéket, amelyet vissza kell adnunk, ha ezt a levelet elértük.

Az ítéletlogikai nyelvek területén a döntési fák teljes kifejezőképességgel bírnak, ami azt jelenti, hogy tetszőleges logikai (Boole) függvény felírható döntési faként.

*Pl.:*



**tanító halmaz:**  a példák teljes halmaza

*Pl.:*



A legegyszerűbb megoldás döntési fa létrehozására, a tanító halmazból, ha minden egyes példát egyesével felveszünk. Ekkor azonban csak memorizáljtuk a példákat, a még nem látott problémákra nem feltétlenül lesz jó. Ockham borotvája elvét alkalmazva nekünk a *legkisebb* fára van szükségünk. Ha a legkisebb fa megatározására nem is, de kellően kicsi fa megtalálására van lehetőségünk: a DÖNTÉSI-FA-TANULÁS algoritmus.

Általánosságban megállapítható, hogy miután az első attribútum tesztje csoportokra bontotta a példákat, mindegyik teszteredmény egy újabb döntési fa tanulási problémát eredményez, kevesebb példával és eggyel kevesebb attribútummal. Négy esetet kell áttekintenünk ezekben a rekurzív problémákban:

1. Ha van néhány pozitív és néhány negatív példánk, akkor válasszuk a legjobb attribútumot a szétosztásukra.
2. Ha valamennyi megmaradt példánk pozitív (vagy mind negatív), akkor készen vagyunk: válaszolhatunk Igen-t vagy Nem-et.
3. Ha nem marad példa a teszt egyik kimenetele esetén, akkor ez azt jelenti, hogy nem figyeltünk meg ilyen esetet, és a szülőcsomópontban többségben levő választ adjuk.
4. Ha nem maradt attribútumunk, amelyet tesztelhetnénk, de mind pozitív, mind negatív példáink maradtak, akkor bajban vagyunk. Ez azt jelenti, hogy ezeknek a példáknak pontosan azonos jellemzőik vannak, de különböző osztályokba tartoznak. Ez egyrészt akkor fordulhat elő, ha néhány adat nem megfelelő, azt mondjuk, hogy **zajosak (noise)** az adatok. Másrészt akkor is előállhat ez a helyzet, ha az attribútumok nem adnak elég információt a szituáció teljes leírására, vagy a problématér valójában nemdeterminisztikus. Egyszerű megoldása lehet ennek a problémának a többségi szavazás használata.

Az attribútumok megfelelő sorrendben történő kiválasztása kulcsfontosságú. Erre a feladatra ad megoldást az **attribútum-választás algoritmus.**







Az ATTRIBÚTUM-VÁLASZTÁS függvényben használt heurisztika csupán azt takarja, hogy válasszuk a legnagyobb nyereséget biztosító attribútumot.

A tanuló algoritmus **teljesítményének mérésére** gyakran a példahalmazból elkülönítünk egy teszthalmazt, amin teszteljük a hipotézis pontosságát. A mérés kimenete a tanulási görbe, ami megmutatja, hogy a tanuló halmaz méret függvényében mekkora a mért helyes válaszok aránya.

**kukucskálás:** oka, hogy a hipotézist a teszthalmazon elért eredménye alapján választottuk ki.

**túlilleszkedés:** amikor a hipotézisek nagy halmaza lehetséges, akkor fennál annak a veszélye hogy értelmetlen “szabályosságot” találunk az adatokban. Ez rendkívül általános jelenség, akkor is jelentkezhet, amikor a keresett függvénynek egyáltalán nincs valószínűségi jellege.

**túlilleszkedés megakadályozása:**

1. szignifikanciateszt segítségével
2. keresztvalidáció segítségével

**problémák a döntési fával:**

* hiányzó adatok
* sokértékű attribútumok
* folytonos és egészértékű bemeneti attribútumok
* folytonos értékkészletű kimeneti attribútumok

## Hipotézishalmaz együttes tanulása

**együttes tanulás:** alapötlete, hogy válasszunk ki a hipotézistérből egy teljes hipotéziskollekciót, és kombináljuk az általuk adott predikciókat. Generálhatunk például ugyanazon tanító halmaz alapján száz különböző döntési fát, és egy új példa esetén szavazással alakíthatjuk ki a legjobb predikciós osztályozási eredményt.

**súlyozott tanító halmaz:** minden mintához hozzárendelünk egy wj ≥ 0 súlyt. Minél magasabb ez a súly, annál nagyobb jelentőséget tulajdonítunk az adott mintának a hipotézis tanulása során.

**turbózás:** az összes mintára wj = 1 értékkel (azaz egy normál tanító halmazzal) indul. Ebből a halmazból generálja az első hipotézist, h1-et. Ez a hipotézis egyes tanítómintákat jól osztályoz, másokat rosszul. Azt szeretnénk, ha a következő hipotézis jobb eredményt érne el a rosszul osztályozott mintáko

n, ezért megnöveljük a rosszul osztályozottak súlyait, míg a jól osztályozottakét csökkentjük. Ebből az új, súlyozott mintahalmazból generáljuk a h2 hipotézist. Az eljárást addig folytatjuk, míg M hipotézist nem generáltunk, ahol M a turbó tanulási algoritmus bemenő paramétere. Az algoritmus eredményeként kapott hipotézis együttes nem más, mint az M hipotézis súlyozott többségi szavazással kombinált eredője. A súlyokat aszerint határozzuk meg, hogy az egyes hipotézisek mennyire jól teljesítettek a tanító halmazon.

## Miért működik a tanulás: a tanulás számítási elmélete

**valószínűleg közelítően helyes hipotézis (VKH):** egy kielégítően nagy tanító példahalmazzal konzisztens választ ad

**közelítőleg helyes hipotézis:** error(h) ≤ ε, ahol ε egy kis konstans.

**döntési lista:** egy kötött formájú logikai kifejezés. Egy tesztsorozatból áll, amely tesztek mindegyike literálisok *konjunkciója*. Ha egy teszt valamelyik példa leírására alkalmazva pozitív eredményt ad, akkor a döntési lista határozza meg a példára adandó választ. Ha a teszt eredménye negatív, akkor a lista következő tesztjével folytatódik a feldolgozás. A döntési listák emlékeztetnek a döntési fákra, de a globális struktúrájuk egyszerűbb, ezzel szemben az egyes tesztek bonyolultabbak, mint a döntési fáknál.

*Pl.:*



**döntési lista tanítása:** Ez ismételten talál egy olyan tesztet, amely pontosan a tanító halmaz valamely részhalmazának felel meg. Amikor talál egy ilyen tesztet, akkor azt hozzáadja a döntési listához, egyben eltávolítva a tesztnek megfelelő példákat a tanító halmazból. Ezek után létrehozza a döntési lista hátralevő részét pusztán a maradék példákra alapozva. Ezt addig ismétli, amíg nem marad több példa.

# Statisztikai tanulási módszerek

## Statisztikai tanulás

**Bayes-tanulás:** során egyszerűen kiszámítjuk minden egyes hipotézis valószínűségét az adatokra támaszkodva, majd ennek alapján adunk predikciót. Azaz nem egyetlen „legjobb” hipotézist használunk a predikcióhoz, hanem az összes hipotézist használjuk, valószínűségükkel súlyozva őket.

Reprezentálja D az összes adatot, legyen d a megfigyelt értékek vektora, ekkor az egyes hipotézisek valószínűségét a Bayes-szabállyal adhatjuk meg:

*P*(*hi*|*d*) = α*P*(*d*|*hi*)*P*(*hi*)

Tegyük fel, hogy egy ismeretlen X mennyiségre vonatkozó predikció a célunk. Ebben az esetben:



ahol azt feltételeztük, hogy az összes hipotézis meghatároz X-re valamilyen eloszlást.

A Bayes-megközelítésben a P(hi) **prior hipotézisek**, illetve a hipotézisek mellett fellépő P(d|hi) **adatvalószínűségek** a kulcsmennyiségek.

Bayes-predikció *optimális*, adott a priori hipotéziseloszlás mellett bármely más predikció ritkábban lesz helyes, mint a Bayes-predikció.

A valós tanulási problémáknál a hipotézistér rendszerint nagyon nagy vagy végtelen, így a legtöbb esetben közelítő vagy egyszerűsített megoldásokra kell szorítkoznunk.

**maximum a posteriori (MAP) hipotézis:** legvalószínűbb hipotézis alapján végezzük a predikciót, azaz olyan hi alapján, amely maximálja a P(hi|d)-t.

## Teljes adattal történő tanulás

Standard módszert a **maximum-likelihood** paramétertanulásra:

1. írjunk fel egy – a paraméter(ek)től függő – kifejezést az adatok együttes valószínűségére (írjuk fel a likelihood függvényt)
2. Írjuk fel minden egyes paraméter szerint a log likelihood függvény deriváltját
3. Keressük meg azokat a paraméterértékeket, amelyek mellett a deriváltak nulla értéket vesznek fel.

## Neurális hálók



Az egység kimeneti aktivációja  ahol aj a j-edik egység kimeneti aktivációja és Wj,i a j-től i-ig vezető összeköttetés súlya.

A neurális hálók irányított kapcsolatokkal (link) összekötött csomópontokból vagy egységekből (unit) állnak. A j-edik egységtől az i-edik felé vezető kapcsolat hivatott az aj aktivációt j-től az i-ig terjeszteni. Minden egyes kapcsolat rendelkezik egy hozzá aszszociált Wj,i numerikus súllyal (weight), ami meghatározza a kapcsolat erősségét és előjelét. Minden egyes i egység először a bemeneteinek egy súlyozott összegét számítja ki:



A kimenetét úgy kapja, hogy ezek után egy g **aktivációs** **függvényt** alkalmaz a kapott összegre:

 

Figyeljük meg, hogy használtunk egy **eltolássúlyt** W0,i -t, amelyet egy rögzített értékű a0 = –1 bemenetre kapcsolunk.

Elvárások az aktivációs függvénnyel szemben:

1. legyen „aktív” (+1 körüli kimenet), ha a „helyes” bemeneteket kapja, és „inaktív” (0 körüli kimenet), ha „rossz” bemeneteket kap
2. az aktiváció legyen nemlineáris

Aktivációs függvények:

* küszöbfüggvény
* szigmoid / logisztikus függvény

**Hálóstruktúrák:**

* előrecsatolt hálók
* visszacsatolt hálók

Az előrecsatolt hálókat rendszerint rétegekbe szervezzük oly módon, hogy minden egyes egység csak a közvetlenül megelőző réteg egységeitől kap bemeneti jelet.

### Egyrétegű előrecsatolt neurális háló

**perceptron:** egyrétegű előrecsatolt neurális háló

**lineáris szeparátor:** küszöbperceptron. Azért szeparátor, mert egy hipersíkot határoz meg a bemeneti térben, és akkor ad 1-et, ha bemenet ennek a hipersíknak az egyik oldalán van.

Egy küszöbperceptron csak lineárisan szeparálható függvények reprezentációjára képes.

A nurális hálóban a tanulást formálisan a súlytérben (weight space) végzett optimalizálási keresésként fogalmazzuk meg.

Egyesével sorban végigfuttatja a mintákat a hálón, és minden egyes példa után a hiba csökkentése érdekében kissé módosítja a súlyokat. A mintahalmaz egyszeri végigfuttatását **epoch**nak nevezzük. Az epochokat addig ismételjük, amíg valamilyen leállási feltétel nem teljesül – tipikus, hogy akkor állunk le, amikor a súlyváltozások már nagyon kicsivé válnak.

### Többrétegű előrecsatolt neurális háló

Vizsgáljuk most a rejtett neuronokkal rendelkező hálókat! A rejtett réteg hozzáadásának az az előnye, hogy kiterjeszti a háló által reprezentálható hipotézisek terét.

A hibát **viszszaterjeszthetjük** (back-propagate) a kimeneti rétegről a rejtett rétegekre, így van lehetőség a háló tanítására, vagyis a súlyok változtatására.

**Neurális háltóstruktúrák tanulása:**

* optimális agykárosodás módszere
* csempézés

# Megerősítéses tanulás

## Passzív megerősítéses tanulás

**passzív tanulás:** az ágens π stratégiája rögzített, az s állapotban mindig a π(s) cselekvést hajtja végre. A cél egyszerűen a stratégia jóságának – tehát az Uπ (s) hasznosságfüggvénynek – a megtanulása.

**állapotátmenet modell T(s,a,s’):** annak a valószínűséget adja meg, hogy az a cselekvés hatására az s állapotból az s' állapotba jutunk

**jutalomfüggvény R(s):** minden állapothoz megadja az ott elnyerhető jutalmat

A passzív ágens nem ismeri az állapotátmenet modellt és a jutalomfüggvényt.

**hasznosság:** úgy definiáljuk, mint a π stratégia követése esetén az összegzett (leértékelt) jutalom várható értéke.



ahol γ a **leértékelési tényező**.

**közvetlen hasznosság becslés módszere:**

az állapot hasznossága nem más, mint az adott állapotból kiindulva az összegzett jutalom várható értéke, és minden egyes kísérlet ennek az értéknek egy *mintájával* *szolgál* minden bejárt állapotra. Az algoritmus mindegyik kísérleti lépéssorozat végén kiszámítja az összes állapotra a megfigyelt hátralevő jutalmat, és ez alapján frissíti az állapot hasznosságát, egyszerűen minden egyes állapotra mozgóablak átlagolást végezve, és az eredményt egy táblában tárolva. Ha a kísérletek száma a végtelenhez tart, a minták átlaga az előbbi egyenlettel megadott valódi várható értékhez fog tartani.

Sajnálatos módon egy nagyon fontos információforrás hiányzik, nevezetesen az, hogy az állapotok hasznossága nem független egymástól! Minden egyes állapot hasznossága egyenlő az ebben az állapotban elnyerhető jutalom és az őt követő állapotok várható hasznosságának összegével. Általánosabban megfogalmazva azt mondhatjuk, hogy a közvetlen hasznosságbecslésre úgy tekinthetünk, mint ami jóval nagyobb U hipotézistérben keres, mint amire szükség van. Ennek oka, hogy számos olyan függvényt is számon tart a hipotézisek közt, amelyek sértik a Bellman-egyenletet (ami leírja, hogy az állapotok hasznossága nem független egymástól). Emiatt az algoritmus gyakran csak nagyon lassan konvergál.

**Bellman-egyenlet**:



**adoptív dinamikus programozás (ADP):** lényege, hogy menet közben megtanulja a környezet állapotátmenet-modelljét, és dinamikus programozási módszerrel megoldja a hozzá kapcsolható Markov döntési folyamatot. Egy passzív tanuló ágens számára ez azt jelenti, hogy a megtanult

T(s, π(s), s') állapotátmenet-modellt és a megfigyelt R(s) jutalmakat behelyettesíti a Bellman-egyenletbe, majd kiszámítja az állapotok hasznosságát.

**időbeli különbség (IK) tanulása:** módszerek alapötlete az, hogy először határozzuk meg azokat a feltételeket, amelyek korrekt becslési értékek esetén lokálisan fennállnak, majd állítsunk fel egy olyan értékfrissítésre szolgáló egyenletet, amely ezen ideális „egyensúlyi” egyenlet irányába viszi a becsléseket.

Általánosabban megfogalmazva, ha egy s-ből s'-be történő átmenet lép fel, akkor a következő módon frissítjük az Uπ (s) hasznosságot:

*Uπ*(*s*) ← *Uπ*(*s*) + *α*(*R*(*s*) + *γUπ*(*s'*)− *Uπ*(*s*))

ahol, α az úgynevezett bátorsági faktor.

## Aktív megerősítéses tanulás

A passzív tanuló ágensnek rögzített stratégiája van, ez határozza meg a viselkedését. Az aktív ágensnek viszont el kell döntenie, hogy melyik cselekvést válassza.

**Kényszeregyenlet:**



**mohó ágens:** az ADP-ágens az egyes lépések során tanult optimális stratégia által javasolt cselekvést követi. Ismételt kísérletek azt mutatták, hogy a mohó ágens ebben a környezetben nagyon *ritkán* találja meg az optimális stratégiát, és időnként borzalmas stratégiákhoz konvergál.

**Cselekvés funkciója:**

* kihasználás (a jelenlegi hasznosságbecslésében tükrözött modell alapján történő jutalom maximalizálás)
* felfedezés (a modell javítás)

A rabló probléma az optimális felfedezési stratégiát mélységében tárgyalja.

**VFHM :** végtelen felfedezés határán lehet mohó

**VFHM sémák:**

1. az ágens az idő 1/t részében véletlen cselekvést választ, egyébként mohó stratégiát követ
2. az ágens némi súlyt ad azoknak a cselekvéseknek, amelyeket nem használt még gyakran, miközben igyekszik elkerülni azokat, amelyeknek a hasznosságát kicsinek hiszi

Nézzük az utóbbi (b) megvalósítását:

* Meg tudjuk valósítani úgy, ha úgy módosítjuk a kényszeregyenletet, hogy nagyobb hasznosságot tulajdonítson a relatíve kipróbálatlan állapot-cselekvés pároknak.
* egy optimista priort hoz létre, az ágens kezdetben úgy viselkedik, mintha hatalmas jutalmak lennének szétszórva a helyszínen

Jelölések:

U+(s) : az s állapot hasznosságának optimista becslése

N(a, s) : azt a számot jelöli, ahányszor az s állapotban az a cselekvést választottuk



Itt f (u, n) az úgynevezett felfedezési függvény, ez határozza meg a mohóság és a kiváncsiság közötti kompromisszumot

**Q-tanulás:** cselekvéstér reprezentációt tanul, és nem hasznosságot. Egy Q-függvényt tanuló IK- ágensnek nincs szüksége sem a tanulás, sem a cselekvés kiválasztás modelljére, ezért a Q-tanulás **modellmentes**.