Döntési fák osztályozó képességének vizsgálata

Felkészülés

A mérés során a döntési fák osztályozó képességét fogjuk vizsgálni. A mérésre fel kell készülni. A szükséges ismeretek megtalálhatók Stuart J. Russel, Peter Norvig MESTERSÉGES INTELLIGENCIA MODERN MEGKÖZELÍTÉSBEN című könyvben. A 2000-es (első magyar nyelvű) kiadásban a „Döntési fák tanulása" című fejezet (VI. rész. Tanulás, 18. Megfigyelések alapján történő tanulás) 629-ik oldalon kezdődik és a 644-ik oldalig tart. Ezt a mérésre át kell nézni. Különösen fontos a döntési fák kialakítása minták alapján (632. o.), a zaj és túlzott illeszkedés (641. o.) és az információelmélet felhasználása (638. o.) című részek. A 2005-ös (második magyar nyelvű) kiadásban szintén a 18. fejezetben van a „Döntési fák megalkotása tanulással" – a 18.3 és következő alfejezetekben (750-dik oldaltól kezdődik a 18. fejezet, a 754-dik oldaltól a 18.3).

A mérésre hozni kell az elkészített házi feladatot (lásd később)!

Néhány ajánlott irodalom annak, akit mélyebben érdekel a téma:

- Stuart J. Russel, Peter Norvig: *Mesterséges intelligencia modern megközelítésben*, Panem, Budapest, 2000
- R. O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification 2nd ed.*, John Wiley and Sons, New York, 2001.
- L. BREIMAN, J. H. FRIEDMAN, R. A. OLSHEN, C. J. STONE, *Classification And Regression Trees*, Chapman & Hall, 1984

Ellenőrző kérdések:

- Milyen döntések találhatók a döntési fa csomópontjaiban?
- Mi az információtartalom?
- Mi az információnyereség egy adott döntés eredményeképp?
- Mi a túlilleszkedés?
- Hogy kerülhetjük el a túlilleszkedést?
- Mi az osztályozási hiba és hogyan számítja ki?

A mérés

A mérés során a *tree* könyvtárban található toolboxot (erről szükség esetén a mérésvezető rövid ismertetést ad a mérés elején) fogjuk használni. Ezt a MATLAB indítása után hozzá kell adni a path-hoz, hogy a MATLAB mindenhonnan elérje.
Parancs: *addpath(genpath('tree'));*

A mérés során minden feladatban az X mátrix a bemenő adatvektorokat jelenti, az y pedig a hozzájuk tartozó választ. Az X mátrix egy sora tehát egy adott bemeneti minta, az y vektor 1 eleme pedig az adott minta osztálya. Az osztályok pozitív egész számok, 1-től felfelé. Az X mátrix elemei valós számok.

A mérés során jegyzőkönyvet kell készíteni. A jegyzőkönyv lényegre törő, világos kell legyen. Minden feladnál tartalmaznia kell a feladat megfogalmazását, a megoldás menetének főbb részleteit, az eredményeket és az *eredmények értékelését*.

Otthon megoldandó feladat a laborra való felkészüléshez

A házi feladatot MATLAB segítségével kell megoldani. A mérésre készülés során próbálja ki a grow_tree, disp_tree, prune_mcc függvényeket!

Töltse be a gyuru2.mat állományt, és jelenítse meg az adatpontokat az osztályuknak megfelelően. Az állomány tartalmaz egy 2 dimenziós (**X**) adatmátrixot, melynek minden sora 1 pontot jelöl a síkban, és egy **y** osztályvektort ahol az adott pontok osztályai (1 ill. 2) találhatók. Írjon függvényt az optimális vágási határ meghatározására az információnyereség kritérium felhasználásával. Mindig csak 1 változót teszteljen egyszerre a kettőből. A meghatározott döntést (pl. X1<hatar1) felhasználva válassza szét a mintahalmazt, majd a kapott részhalmazokra újra határozza meg a döntési határt. Ezt folytatva minimum 3 döntési mélységig határozza meg a döntéseket. A döntéseket felhasználva rajzolja fel (ezt nem kell matlabbal, elég papíron) a kapott döntési fa kezdeményt. Szintén ábrázolja (elég papíron) a mintahalmazt és húzza be a fa döntéseit. A kódot és az ábrákat hozza magával a mérésre.

1. Osztályozás szeparálható mintacsoportokkal

Az első feladatban egy 2 osztályos, 2 dimenziós, szeparálható problémát fogunk megoldani. Töltse be a puzzle2.mat állományt, és jelenítse meg a mintahalmazt.

Hozzon létre tanító és teszt mintahalmazt. Mikre kell figyelni a tanító, illetve teszt halmazok létrehozásánál? Illesszen döntési fát a tanító halmazra. Vizsgálja meg a fa felépítését és ellenőrizze működését. Milyen jellegű a fa?
Mintakód:

```
T=grow_tree(X_learn,Y_learn);
disp_tree(T);
display2D_data_and_tree(X,y,T);
```

Határozza meg a legjobb fát minimális hiba-komplexitás értelemben (T0=prune_mcc(T,X_test,Y_test,1)). (*A döntési fa költségét 2 tényező határozza meg. Az egyik a fa osztályozási hibája, a másik a fa bonyolultsága. Nyilván olyan döntési fát szeretnénk, ami egyszerű és ugyanakkor kicsi osztályozási hibával rendelkezik. Ez ritkán teljesül, legtöbbször meg kell találni az optimális egyensúlyt a komplexitás és a hiba között. Bizonyos esetekben már nem célszerű a komplexitás további növelése – azaz új döntések hozzáadása a fához –, mivel az csak minimális osztályozási javulást okoz, illetve túltanulást eredményez. Az optimális komplexitás meghatározásáról ilyen értelemben a*

mérésen a mérésvezető fog bővebben beszélni.) Vizsgálja meg a fa felépítését és ellenőrizze működését (disp_tree(T), display2D_data_and_tree(X_learn,Y_learn,T)). A tanulási görbe megfelel a vártnak? Miért ilyen jellegű? Milyen komplexitású az ilyen értelemben optimális fa? Mekkora a túlilleszkedés?
Mintakód:

```
T0=prune_mcc(T,X_test,Y_test,1);
disp_tree(T0);
display2D_data_and_tree(X,Y,T0);
```

2. Ellenőrizze az elkészített házi feladatot az előző feladat mintájára

3. Osztályozás nem szeparálható mintacsoportokkal

Keverjen különböző mértékű – nulla várható értékű – zajt (pl. egyenletes eloszlású, 0.025, 0.05, 0.1, 0.2 terjedelmű) a mintákhoz, majd ismét végezze el döntési fa illesztést, illetve az optimális méretű fa meghatározást. Ahogy növeli a zajt, mit tapasztal a maximális, illetve az optimális fa komplexitását illetően? Van túlilleszkedés? Milyen jellegű a tanulási görbe? Megegyezik a szeparálható esetben tapasztalttal? Állítsa az SE vágási paramétert, amennyiben szükségesnek érzi! (A prune_mcc függvény utolsó paramétere. *Gyakorlatilag az adott méretű fa osztályozási hibájának szórása. Amennyiben ezt egységnyi súllyal vesszük figyelembe, akkor a nyeső algoritmus nem azt a fát adja vissza ahol minimális a költség-komplexitás, hanem azt melynek hibája még ennek a minimálisnak 1 szórásnyi körzetében van, de legkisebb a komplexitása.*)

4. Konkrét szenzorhálózat mért eredményei alapján tanított döntési fa

A laborban összeállított egyszerű „üvegház modell”-en végeztünk méréseket, ezek segítségével tanítson egy döntési hálót! A döntés annak felismerésére irányul, hogy az „üvegház” NYITOTT vagy CSUKOTT állapotban van-e. A CSUKOTT állapotban a belső szenzorokat tartalmazó dobozt lefedtük egy plexilappal, amit NYITOTT állapotban eltávolítottunk. Ennek megfelelően a két állapotban más dinamikával hűl, illetve fűtés esetén más dinamikával melegszik fel a belső tér.

A döntési fa tanításához felhasználhatók a következő állományokban található mérések:

FedettNincsBeavatkozasHul.mat
NyitottNincsBeavatkozasHul.mat
NyitottLampavalFutve.mat
FedettLampavalFutve.mat

A nevek alapján nyilvánvaló, hogy 2-2 fedett és 2-2 nyitott állapotban mért értéksorról van szó. Mindkét állapotban egy felfűtés és egy lehűlés során vettünk fel 10.000 időpontban (0,8 másodpercenként) különböző mért értékeket. A mat fájlok betöltésekor a *magy* és *magy1* változókban olvashatók az alapvető információk.

Alakítson ki úgy döntési fát,hogy az adatok egy részét tanításra használja, egy másik részét elkülöníti tesztre! Mennyire jól általánosít döntési fa?