KÉT FUZZY OSZTÁLYOZÓ MÓDSZER[1]

BORGULYA ISTVÁN
Janus Pannonius Tudományegyetem Közgazdaságtudományi Kar

Több kritériummal jellemzett alternatívák esetén az osztályozás, vagy klaszterek képzése gyakori feladat, több módszer létezik megoldásukra. Dolgozatomban két új heurisztikus módszert kívánok bemutatni, melyek több kritériummal jellemzett fuzzy, vagy éles (crisp) alternatívák osztályozását, valamint adott számú osztályra, klaszterre bontását teszik lehetővé.

Az osztályozó algoritmus minden alternatívához (elemhez) egy fuzzy rendező módszerrel, mint függvénnyel egy értéket rendel. Minden osztálynál értelmezi a centrumot, mint az osztály elemek függvényértékeinek átlaga, valamint minden osztályban egy prototípust, amely az osztály olyan eleme, melynek függvényértéke a legközelebb esik a centrum értékéhez. Az algoritmus a kritériumok súlyszámának változtatásával megkeresi az egyes osztályok prototípusait és centrumait. A klaszterképző algoritmus alapgondolata, lépései majdnem megegyeznek az osztályozó algoritmuséval, de értelemszerűen a helyes osztályba sorolást nem tudja ellenőrizni. Mindkét rendszerrel a "tanulás" után további esetek is osztályozhatók

Más megközelítéssel készült módszerekkel, pl. két neurofuzzy osztályozó módszerrel összehasonlítva, a tesztelés hasonló eredményeket adott (97%-os pontosság). Az osztályozó algoritmus relatíve magas belső szórású osztályok elkülönítésére is alkalmas, osztályonként több csoport megkülönböztetésével.

1. Bevezetés

Több kritériummal jellemzett alternatívák (esetek) osztályozása, vagy klaszterek képzése az alapfeladatok közé tartozik. Több módszer létezik megoldásukra. Utóbbi időben különösen a fuzzy osztályozó, klaszterező algoritmusok terén olvashatunk újabb módszereket. A gyakori módszerektől eltérően, egy olyan módszert kívánok bemutatni, amely a neurális hálózatok tanulásához hasonló elemeket tartalmaz.

A neurális hálózatok tanulása vetette fel az ötletet, hogy az alternatívák rendezését fuzzy rendszerrel is meg lehet tanulni, és a több kritériummal

[1]Készült az OTKA T18562 kutatás keretében. Beérkezett: 1998. január 24.

jellemzett alternatívák sorrendje, sorszámaik ismeretében, a kritériumok súlyszámának változtatásával, az ismert adatokból megtanulható [Borgulya 97b]. Ha az ismert osztályok azonosítóit az alternatívákhoz mint "sorszámot" rendeljük, az alternatívák osztályok alapján történő rendezése egyúttal osztályozást is fog jelenteni. Vagy ha ismeretlenek az osztályok, és k darab klaszterbe akarjuk sorolni az alternatívákat, a klaszterek centrumait "sorszámnak" tekintve, az alternatívák az egyes centrumok alapján szintén sorba rendezhetők.

Több kisebb példán ellenőrizve az elképzelést, az eredmények részben igazolták a feltevést. Egyszerű, könnyen szeparálható osztályok esetén olyan sorrendet kaptam, amely az azonos osztályba tartozó alternatívákat egy csoportba gyűjti, egymás után helyezve el őket. Komplikáltabb osztályozás esetén már a csoportok közt átfedések alakultak ki, fuzzy "csoportok" keletkeztek, ill. egyre több hibás osztályozás lépett fel. A precízebb osztályozás érdekében ezért az osztályba sorolásnál belső hasonlóságot is vizsgáltam, és a "sorszám" és a hasonlóság együtt határozta meg az alternatívák sorrendjét. Ez utóbbi esetben már lényegesen csökkent a hibák száma, az ellenőrző tesztek szerint 97-100%-os pontosságot sikerült elérni.

Mivel a területen számos algoritmus ismert, először tekintsük át nagy vonalakban a fuzzy klaszternalízis és a fuzzy osztályozó algoritmusok csoportjait, majd részletesebben a két heurisztikus algoritmust és tesztelésének eredményeit.

2. Fuzzy klaszteranalízis és fuzzy osztályozás

A klaszteranalízis célja az adatok olyan csoportosítása, melyben a hasonló adatok egy osztályba (klaszterbe), a kevésbé hasonlók különböző osztályokba kerülnek. Az elmúlt 20 évben számos klaszteranalízis eljárást (KAE) fogalmaztak meg. Az eljárások, melyek közt fuzzy módszerek is találhatók, különbözőképpen csoportosíthatók. Egy lehetséges csoportosítás az alkalmazott technikák alapján [Höppner et al, 97], pl.:

- *Nem teljes KAE-ok.* Különböző geometriai, ábrázolási, projekciós technikák jellemzik. Az osztályok 2-3 dimenziós vetületeit állítják elő.
- *Determinisztikus KAE-ok.* Minden adat csak egy osztályba tartozik és az osztályok az adathalmazt teljesen felosztják.
- *Átfedő KAE-ok.* Minden adatot legalább egy osztályhoz hozzárendelnek, azaz egy adat több osztályhoz is hozzárendelhető.
- *Valószínűségi KAE-ok.* Egy valószínűségi eloszlás minden adatnál megadja, milyen valószínűséggel tartozik az adat valamely osztályhoz. Ha

a valószínűséget hozzátartozás foknak értelmezzük, akkor ezek az eljárások fuzzy KAE-nak is tekinthetők (a valószínűségek összege egy).

- *Lehetségességi KAE-ok.* Minden adatnál megadják, hogy milyen lehetségességi, vagy hozzátartozási fokkal tartozik valamely osztályhoz.
- *Hierarchikus KAE-ok.* Az eljárások több lépésben egyre finomabb bontását állítják elő az induló osztálynak, ill. fordítva: az induló kis osztályokat egyre nagyobb osztályokká olvasztják össze.
- *Objektív függvényt alkalmazó KAE-ok.* Az objektív függvény olyan cél, vagy értékelő függvény, melynek az optimumát keressük. Az objektív függvény minden osztály felosztáshoz egy hibaértéket, vagy jóság mértéket rendel, ami alapján a legkedvezőbb osztályozást választhatjuk ki.

A fuzzy KAE-ok többsége az objektív függvényt alkalmazó rendszerekhez tartozik, és az objektív függvény alkalmazása mellett általában valószínűségeloszlás, vagy lehetségességi fokok alapján történik a klaszterek kialakítása.

A fuzzy KAE-ok többsége akár valószínűségi, akár lehetségességi klaszter felosztással definiálható. Az első, és legismertebb algoritmus a *Fuzzy-c Means-algoritmus* (FCM). A módszer éles adatokon megfogalmazott első változatát Duda és Hart (1973) fogalmazta meg, a jelenleg használatos változatát 1992-ben publikálták [Bezdek et al, 92]. A FCM eljárás kúpalakú ponthalmazokat képes elkülöníteni a p-dimenziós térben. Minden klasztert egy középponttal jellemez, melynek koordináta pontjai a "prototípusok". A klasztereket a prototípusok képviselik, és az osztályba sorolás az adatok és a prototípusok távolsága alapján történik. Az eljárás feladata egy iterációs eljárás során az optimális klaszter középpontok meghatározása a hozzá tartozó hozzátartozási fokokkal együtt. Az iteráció mindaddig folytatódik, amíg a középpontok stabillá nem válnak.

A FCM-ből az alkalmazott távolságdefiníció megváltoztatásával a kúpformájú klaszterek helyett ellipszis, ill. ellipszoid alakú klaszterek felismerése is elérhető. E módosítást a Gustafson-Kessel algoritmus fogalmazza meg [Gustafson et al, 79]. A FCM algoritmus, ill. a Gustafson-Kessel algoritmus további, olyan változatairól is beszámolhatunk, melyek már egy fuzzy kovariancia mátrixot használnak fel a klaszterek alakjának, nagyságának beállítására.

A FCM algoritmus egyes változatai, különösen a képfeldolgozásban használhatók fel sikeresen. A *lineáris klaszter eljárások* (pl. Fuzzy-c-Variaties-algoritmus, vagy adaptive-Fuzzy Clastering-algoritmus) a képeken az egyeneseket, mint klasztereket ismerik fel. A *shell-klaszter eljárások* (pl. fuzzy-C-Shell-algoritmus, vagy fuzzy-C-Ellipsoidal-Shell-algoritmus stb.), még tovább lépnek. Körök, ellipszisek, parabolák, hiperbolák kontúrjait, mint klasztereket képesek felismerni. A FCM továbbfejlesztett változatai speciális törtvonalak kontúrjait, mint klasztereket ismernek fel (pl. Fuzzy-C-Rectangular-Shell-algoritmus, vagy Fuzzy-C-2-Rectangular-Shell-algoritmus). Így téglalapok, szabályos sokszögek, vagy téglalapok metszeteként előállítható törtvonalakat, mint klaszterek kontúrjait ismerik fel.

E fuzzy KAE-ok mind iteratív eljárások, melyek az iteráció minden lépésében finomítják a hozzátartozási fokok és a klaszter középpontok becslését. A módszerek egy másik csoportja egy lépésben, iteráció nélkül számolja ki az előbbi értékeket. Ilyen módszer pl. Yager és Filev Montain Clustering algoritmusa, amely ismeretlen számú klaszter meghatározására készült [Yager et al, 94].

Az *osztályozás* nagyon hasonló probléma a klaszteranalízishez. Egyes feladatoknál a klaszter algoritmussal kapott osztályok egybeesnek a várt osztályokkal, azaz a kétféle feladat eredménye ugyanaz. Összetettebb feladatoknál azonban a klaszter algoritmusok nem adnak kielégítő eredményt az osztályozási feladatra. Kombinálni kell a klaszteranalízis módszereit más módszerekkel, vagy teljesen más nézőpontból kell megközelíteni a problémát.

Az osztályozást megvalósító fuzzy rendszerek (FR) az esetek többségében szabályalapú rendszerek, melyek, minden osztályhoz legalább egy szabállyal rendelik az elemeket. A probléma formálisan a következőképpen írható le:

Legyenek $X = (x_1, x_2, \ldots x_n)$ n dimenziós elemek, amelyek a $C = C_1$, C_2, ..., C_k osztályokba tartoznak (Bármely két osztály metszete üres). Az elemek koordinátáit írjuk le az A_1, A_2, ..., A_n fuzzy halmazok segítségével, és az egyes osztályokat kijelölő R_1, R_2, ..., R_k szabályok formája legyen

R_i: IF (A_1 is x_1) AND (A_2 is x_2) AND ... AND (A_n is x_n) THEN class is i,

ahol a class változó a konklúzióként kapott C_i osztály sorszámát adja. Minden szabály előállít valamilyen osztálysorszámot, és az osztályozáshoz a legjobban illeszkedő osztályt kell kiválasztani. Azt a szabályt kell tehát keresni, amelynek premisszáit a legjobban kielégíti az adott elem, azaz amelyik osztályhoz a legnagyobb hozzátartozási fokkal besorolható. Ha az A_1, A_2, ..., A_n fuzzy halmazok tartalmazási függvényeit μ_1, μ_2, ..., μ_n-nel jelöljük, akkor az i-edik szabálynál a hozzátartozási fok

$$\mu_{R_i}(x_1, x_2, \ldots, x_n) = \min\{\mu_1(x_1), \mu_2(x_2), \ldots, \mu_n(x_n)\} .$$

A keresett osztályt a legnagyobb $\mu_{R_i}(x_1, x_2, \ldots, x_n)$ értékű osztály adja.

A szabályok maghatározása többféleképpen történhet. Ha 2 dimenzióban, lineárisan szeparálható osztályokról van szó, akkor az osztályokat elválasztó egyenest egy lépcsősfüggvénnyel közelíthetjük, és minden "lépcsőhöz" egy-egy szabályt írhatunk fel. Összetettebb, több osztályt tartalmazó n-dimenziós osztályozási feladatoknál azonban ez az út nem járható.

Az újabb osztályozási módszerek neurofuzzy, vagy KAE-ok segítségével osztályoznak. A neurofuzzy rendszerek az osztályozásra kerülő adatokból tanulják meg a szükséges fuzzy halmazokat és szabályokat. A klaszteranalízisen alapuló módszerek a felismert klaszterek alapján generálnak halmazokat és szabályokat. Neurofuzzy osztályozó módszert alkalmaz pl. a NEFCLASS rendszer, amely egy 3 rétegű fuzzy perceptron segítségével állítja elő a fuzzy osztályozó rendszer halmazait, szabályait [Nauck et al, 95].

A felsorolt fuzzy osztályozó rendszerek fuzzy halmazokat és szabályokat generálnak az adatok alapján. Az ismertetésre kerülő súlyszám tanulással történő osztályozásnál mások a kiinduló feltételek. Itt ismertek az elemek leírásához szükséges kritériumok éles adatai, vagy fuzzy halmazai, és centrumokat, prototípusokat, súlyszámokat kell keresni.

A heurisztikus fuzzy osztályozó algoritmus leginkább a FCM algoritmushoz hasonló. Az osztályozáshoz osztály centrumokat, prototípusokat határoz meg. Az osztályok elemeit elsősorban hasonlóság alapján állapítja meg, mely egy lehetségességi klaszter felosztásnak felel meg. Értékelő függvényként olyan hibafüggvényt alkalmaz, amely egy fuzzy rendező módszer eredményei alapján ellenőrzi a sorrendet és a centrumok stabilitását. Eltérően a FCM-től, a centrumok, prototípusok keresésére a kritérium súlyok változtatását is felhasználja. E ponton a neurofuzzy rendszerekhez hasonló az algoritmus.

3. Fuzzy osztályozás súlyszám tanulással

Az osztályozó algoritmus alapgondolata

Rendeljünk minden alternatívához (elemhez) egy fuzzy rendező módszerrel, mint függvénnyel egy értéket, mely alapján az alternatívák sorrendje ellenőrizhető. Az osztályok elemeit hasonlóság alapján soroljuk egy-több csoportba, és rendeljünk minden csoporthoz egy centrumot és egy prototípust. A centrum az osztály (ill. csoportbeli) elemek függvényértékeinek átlaga, a prototípus pedig az osztály olyan eleme, melynek függvényértéke a legközelebb esik a centrum értékéhez. Az algoritmus a kritériumok súlyszámának változtatásával megkeresi, megtanulja az egyes osztályokat, csoportokat jellemző prototípusokat, centrumokat. Hogy az algoritmus a relatíve magas belső szórású osztályokat is fel tudja ismerni, osztályonként a hibás osztályozástól függően automatikusan több csoport definiálását is lehetővé teszi.

Osztályozás esetén minden elemnek ismerjük az osztályát. A fuzzy rendszernek (FR) ezt a helyes osztályt kell a kritériumok értékei alapján felismernie és az elemhez rendelnie. Mivel az osztályok a centrumokkal és a kapcsolódó prototípusokkal egyértelműen jellemezhetők, olyan algoritmust kell megfogalmazni, amely egy elemhez azt az osztályt rendeli, melynek prototípusához a leghasonlóbb és ez egyúttal azonos a helyes osztállyal is.

Az osztályozást fokozatosan tanulja meg rendszer. Egyenként, véletlenszerűen veszi elő az elemeket, és minden elemet hasonlósági mérték alapján valamelyik osztályba (csoportba) sorol. Hogy a centrumok egy stabil érték felé konvergáljanak, a kritériumok súlyszámait úgy módosítja, hogy az elem új függvényértéke a megfelelő centrum felé közeledjék. A súlyszámokat minden olyan esetben módosítja, ha az osztályba sorolás téves, vagy az utolsó két elemhez tartozó függvényértékek sorrendje eltérő osztály centrumaik sorrendjétől (a súlyszám módosítás a neurális hálózatok delta szabályához hasonló képlettel történik). Az algoritmus mindaddig folytatódik, amíg adott pontossággal nem ismeri fel az osztályokat, és a centrumok stabillá nem válnak (változásuk értéke adott küszöbszám alá nem csökken).

A hibás sorrend megállapításához figyelembe kell venni, hogy

- az egy osztályba tartozó elemek sorrendje közömbös, bármilyen sorrendjük elfogadható;
- az osztályok azonosító sorszáma, és az osztályok rendezésnél kapott sorrendje eltérő lehet;
- az egyes osztályok, ill. az osztályt alkotó csoportok sorrendjét centrumaik véletlenszerűen kialakult sorrendjével azonosíthatjuk.

Nézzük e heurisztikus algoritmust közelebbről. Jelöljük az osztályozásra kerülő elemeket a_1, a_2, ..., a_n-nel, és ismert osztályaikat pedig jelöljük az 1, 2, ..., k sorszámokkal. Az elemeket leíró kritériumokat jelöljük K_1, K_2, ..., K_m-mel, és az l-edik kritériumot, mint nyelvi változót értelmezzük. A választott fuzzy rendező módszer (FRM) a K_1, K_2, ..., K_m nyelvi változók alapján minden a_i elemhez egy y_i valós számot rendel ($i = 1, 2, \ldots, n$).

Az algoritmus megfogalmazásához három lényeges pontban kell dönteni: a hasonlóság kezelésében, a súlymódosítás szabályában és az alkalmazható FRM választásában.

Hasonlóság

A hasonlóságot azonos formában kezelhetjük éles és fuzzy elemek esetén. Éles kritériumokkal adott a_i, a_j elemeknél a hasonlóság mértéke legyen

$$H(a_i, a_j) = 1/(1 + d(a_i, a_j)) \ , \tag{1}$$

ahol $d(a_i, a_j)$ az éles elemek euklideszi távolsága. Fuzzy kritériumok esetén a hasonlóság mértékére ismét az (1) összefüggés alkalmazható, de a távolságképletben Munda (1995) szemantikus távolságát használjuk fel. E szerint két tetszőleges a_i, a_j elem távolsága ($p = 2$ esetén az „euklideszi" távolság):

$$d(a_i, a_j) = |a_i - a_j| = \left[\sum_{l=1}^{N} w_l * \left(\iint |x - y| f_l(x) g_l(y)\, dy\, dx\right)^p\right]^{\frac{1}{p}}$$

ahol folytonos, konvex tartalmazási függvényeket feltételezve, esetünkben $f_l(x) = c_{1l} * \mu_{K_l}(x)$ és $g_l(x) = c_{2l} * \mu_{K_l}(x)$ (c_{1l}, c_{2l} valós számok), és f_l, g_l-re teljesülni kell:

$$\int f_l(x)\, dx = \int g_l(x)\, dx = 1\,.$$

Az algoritmus egy elem osztályba tartozását hasonlóság alapján fogja megállapítani. Amelyik prototípushoz a leghasonlóbb, annak az osztályához fog az eset tartozni. Tekintettel arra, hogy a relatíve magas belső szórású osztályoknál egy prototípussal nem lehet minden elemet jól jellemezni, osztályonként több prototípust is alkalmazhatunk. Ekkor hibás osztályozás esetén újabb prototípust vezethetünk be.

Súlymódosítás

Az osztály centrumok alapján való rendezést az algoritmus véletlenszerűen kiválasztott elemek alapján ellenőrzi. Ha az utoljára vizsgált két elem sorrendje nem megfelelő, "kis lépésekkel" a helyes sorrend irányába változtatja az elemekhez rendelt y számokat. Ehhez minden olyan kritérium súlyát változtatja, és a helyes irányban növeli, ill. a másik irányban csökkenti, ahol a két elemnél az azonos kritériumok értékei eltérőek.

A hasonlóság alapján történő osztály hozzárendeléseknél sorrendhiba kétféle okból keletkezhet:

1. két tetszőleges, különböző osztályhoz tartozó elemnél az osztályok centrumainak sorrendje ellentétes az elemekhez az FRM által rendelt függvényértékek sorrendjével, vagy
2. egy elemnél az algoritmus által meghatározott osztály nem azonos a helyes osztállyal.

Ha az i és j-edik elemnél az osztályozás helyes, csak a FRM által hozzárendelt függvényértékek sorrendje hibás, az éles, vagy fuzzy kritériumoknak megfelelően módosítani kell a súlyokat. A súlyszám változtatás mértékét:

- az utolsó két elemnél az azonos kritériumok értékeinek különbsége,
- az utolsó két elemhez tartozó osztályok centrumainak különbsége,

- az utolsó elem hasonlósága a helyes prototípusához: $H(a_i, p_z)$,
- a "tanulási ráta", amely megadja a súlyszám változtatás egységét

befolyásolja. Ha a vizsgált párnál az a_i és a_j elemek sorrendje nem megfelelő, akár több kritérium súlyszámát is szükséges módosítani. Ha az algoritmus által meghatározott osztály eltér a helyes osztálytól, korrigáljuk úgy a hibát, hogy az elemet hasonlóbbá tesszük a helyes osztály prototípusához. A súlymódosító szabály ebben az esetben is ugyanaz, csak előző elemnek a helyes osztály prototípusát kell választani.

A súlymódosítás szabálya különbözik éles, ill. fuzzy kritériumok esetén. Éles kritériumnál a súlymódosítás képlete:

$$\Delta w_l = -\eta * \mathrm{sign}(y_i - y_j) * (k_{li} - k_{lj}) * (oc - c_z) H(a_i, p_z) \ , \tag{2}$$

ahol η a tanulási ráta, k_{li}, k_{lj} az előző és az utolsó kritériumérték az l-edik kritériumnál, p_z a helyes osztály prototípusa, és oc, c_z az elemek helyes centrumait jelöli.

Fuzzy kritériumok esetén nem tudunk a k_{li}, k_{lj} valós értékek alapján következtetni a szükséges súlymódosítás irányára, nagyságára. A kritériumok, mint fuzzy halmazok közti kisebb, nagyobb relációt viszont meghatározhatjuk minden kritérium esetén Munda (1995) javaslata alapján az

$$\mathrm{int}_l = \iint (x - y) f_l(x) g_l(y) \, dy \, dx$$

integrál előjele alapján. Ha pozitív, akkor a nagyobb, különben a kisebb reláció igaz a kritériumok aktuális értékei közt. A súlymódosításnál ezt felhasználva a (2)-es képlet új alakja:

$$\Delta w_l = -\eta * \mathrm{sign}(y_i - y_j) * \mathrm{int}_l * (oc - c_z) H(a_i, p_z) \ , \tag{3}$$

Megjegyzés: Az algoritmus programozásánál egyszerűsítve számoltam: a szemantikus távolságot közelítésként folytonos helyett diszkrét esetre alkalmaztam, és int_l helyett a

$$\mu_{K_l}(x) * \mu_{K_l}(y) * (x - y)$$

képlettel számoltam, ahol x, y a kritérium input értékei.

A rendező módszer

FRM-nek az algoritmus tesztelése közben a max-min módszert [Yager 78], valamint az "osztályozó módszert" [Borgulya 95, 97a] választottam. Éles kritériumok esetén a két módszer hasonló eredményt adott. Tekintettel arra, hogy az osztályozó módszer fuzzy kritériumoknál általánosabban használható, az osztályozó módszert alkalmaztam végül az algoritmusnál. Nézzük a továbbiakban e rendező módszert vázlatosan.

Olyan több kritériummal jellemzett a_1, a_2, ..., a_n alternatívákat kell rendezni, ahol a K_1, K_2, ..., K_m kritériumok súlyozottak és a kritérium értékek fuzzy halmazok (g_1, g_2, ..., g_m a kritériumokhoz tartozó súlyszámok, és a súlyszámok maximális értéke 1).

E probléma megoldását kétféleképp közelítjük meg:

- speciális esetként feltesszük, hogy a kritériumok értékei az oktatásban alkalmazott osztályzatokkal kerülnek megadásra, vagy ilyen osztályzatokká transzformálhatók;

- általános esetben magukat a kritérium értékeket tekintjük osztályzatnak anélkül, hogy az oktatásban alkalmazott osztályzatok tulajdonságaival rendelkeznének.

Mindkét esetben egyetlen számot, "extra osztályzatot" rendel a módszer minden alternatívához, melyek ez alapján rendezhetők.

Általános eset

Legyen $X = \{a_1, a_2, \ldots, a_n\}$ az alternatíváknak egy véges halmaza, legyen $K = \{K_1, K_2, \ldots, K_m\}$ a fuzzy kritériumoknak egy véges halmaza, legyenek továbbá g_1, g_2, ..., g_m a kritériumokhoz tartozó súlyszámok, a súlyszámok maximális értékét egynek választva.

Tekintsünk X felett minden K_j fuzzy kritériumot nyelvi változónak ($1 \leq j \leq m$) és legyen

$$K_j = \{S_{j1}, S_{j2}, \ldots, S_{jp_j}\} ,$$

ahol S_{j1}, S_{j2}, ..., S_{jp_j} a nyelvi változó értékei. Az a_i alternatívát ($1 \leq i \leq n$) a K_j kritérium S_{j1}, S_{j2}, ..., S_{jp_j} fuzzy halmazaival értékelhetjük.

Definiáljuk továbbá a E eredményhalmazt a kritérium halmazok uniójaként:

$$E = \bigcup_{j=1}^{m} K_j .$$

Definiáljuk a követkő FAM szabályokat:

$$
\begin{array}{lll}
(g_1^2) & \texttt{IF } k_1 = S_{11} & \texttt{THEN } E = S_{11} \\
(g_1^2) & \texttt{IF } k_1 = S_{12} & \texttt{THEN } E = S_{12} \\
 & \ldots & \\
(g_1^2) & \texttt{IF } k_1 = S_{1p_1} & \texttt{THEN } E = S_{1p_1} \\
(g_2^2) & \texttt{IF } k_2 = S_{21} & \texttt{THEN } E = S_{21} \\
(g_2^2) & \texttt{IF } k_2 = S_{22} & \texttt{THEN } E = S_{22} \\
 & \ldots & \\
(g_2^2) & \texttt{IF } k_2 = S_{2p_2} & \texttt{THEN } E = S_{2p_2} \\
 & \ldots & \\
(g_m^2) & \texttt{IF } k_m = S_{m1} & \texttt{THEN } E = S_{m1} \\
(g_m^2) & \texttt{IF } k_m = S_{m2} & \texttt{THEN } E = S_{m2} \\
 & \ldots & \\
(g_m^2) & \texttt{IF } k_m = S_{mp_m} & \texttt{THEN } E = S_{mp_m}
\end{array}
$$

ahol (g_i^2) $i = 1, 2, \ldots, m$ a szabályok súlyszámai, és k_1, k_2, …, k_m egy alternatíva adatai.

Az előállított fuzzy rendszer a szabályok leképezéseit aggregálva (T-konorma), a súlypont defuzzifikáló eljárással minden a_i alternatívához egy y_i értéket rendel, mely alapján rendezhetők az alternatívák [Borgulya 97a].

Speciális eset

Az általános osztályozó módszer speciális eseteként a rendezni kívánt alternatíváknál a kritériumok értékei legyenek egységesen, nem szükségképpen azonos fokozatú, fuzzy halmazként értelmezett osztályzatok [Borgulya 95]. Az osztályzatok ($S_1, S_2, \ldots, S_p$) legyenek szimmetrikus háromszögekkel, vagy Gauss-függvényekkel megadott fuzzy halmazok a [jegy-1, jegy+1] intervallum felett, ahol jegy=1, 2, …, p lehet. (Az E halmaz most csak a lehetséges p darab S_1, S_2, …, S_p érték uniójából áll). A speciális esetben az alternatívákhoz rendelt y_i értékek osztályzatként értelmezhetők, és rendezhetők segítségükkel az alternatívák.

A heurisztikus algoritmusban fuzzy kritériumok esetén az osztályozó módszer általános, éles kritériumok esetén a speciális változatát alkalmazzuk. Éles kritériumok esetén a kritérium értékeket transzformáljuk egységesen, pl. a $[0, 5]$ intervallumra, és utána alkalmazható a FRM.

A heurisztikus fuzzy osztályozó algoritmus lépései

Legyen az osztályok száma k, és engedjük meg $k + t$ számú prototípus alkalmazását (t lehet nulla is). Legyen ro egy adott hasonlósági érték, amely egy osztály több csoportra bontását szabályozza, legyen ϵ a centrumok stabilitásának ellenőrzésénél a küszöbszám, és legyen γ a még elfogadható hibás osztályozások száma. Jelöljük az elemek helyes osztályait ho_1, ho_2, ..., ho_n-nel.

1. Definiáljuk az osztályok (ill. csoportok) induló üres halmazait $C = \{C_i\}$, $i = 1, 2, \ldots, k+t$. Legyenek c_1, c_2, ..., c_{k+t}, valamint p_1, p_2, ..., p_{k+t} az osztályok centrumai, valamint prototípusai. A C_i halmazhoz tartozó centrum, prototípus, osztály sorszám legyen c_i, p_i, o_i. Legyen $c_i = 0$, $p_i = 0$, és $o_i = 0$ $(i = 1, 2, \ldots, k+t)$.

2. Válasszunk véletlenszerűen egy w_1, w_2, ..., w_m súlyszám sorozatot. Válasszunk véletlenszerűen két indexet: i, j-t. Legyen az első a_i elem nulla. Az induló pár (a_i, a_j). Legyen $oc = 0$ és $y_i = 0$.

3. Legyen FRM: $a_j \rightarrow y_j$

4. Hasonlóság vizsgálat. Ha $H(a_j, p_z) = \max_q H(a_j, p_q)$ (alkalmazva az (1) képletet éles, vagy fuzzy esetben), ahol $q, z \in [1, k+t]$, akkor az a_j-hez rendelt osztály o_z. Legyen FRM: $p_z \rightarrow y$.

5. Prototípus módosítás.

   a) Ha $(ho_j = o_z) \wedge (c_z - y_j < c_z - y) \wedge (H(a_j, p_z) \geq ro)$ akkor $p_z = a_j$, és ha $a_j = p_x$ $(x = 1, 2, \ldots, z-1, z+1, \ldots, k+t)$, akkor $c_x = 0$, $o_x = 0$.

   b) Ha $(ho_j \neq o_z) \wedge (c_l = 0)$ $l = ho_j$ akkor $p_l = a_j$, $c_l = y_j$, $o_l = ho_j$ és $z = l$. Különben,

   ha $(ho_j \neq o_z) \wedge (\exists c_l)(c_l = 0)$ $l = k+1, \ldots, t$, akkor $p_l = a_j$, $c_l = y_j$, $o_l = ho_j$ és $z = l$.

6. Súlymódosítás.

   a) Ha $(ho_j = o_z) \wedge (oc < c_z \wedge y_i > y_j) \vee (oc > c_z \wedge y_i < y_j)$, akkor éles kritériumok esetén a (2), fuzzy kritériumok esetén a (3) súlymódosító képletet alkalmazzuk.

   b) Ha $(ho_j \neq o_z) \wedge (\exists x)(o_x = ho_j)$ akkor $p_x \rightarrow a_i$, $y_i = c_x$, $oc = 2c_x$, $z = x$, és éles kritériumok esetén a (2), fuzzy kritériumok esetén a (3) súlymódosító képletet alkalmazzuk.

7. Súlyok normalizálása.

$$\|W\| = \sum_{i=1}^{m} w_i \qquad \text{és} \qquad w_i = \frac{w_i}{\|W\|}, \quad i = 1, 2, \ldots, m.$$

8. $i = j$, $oc = c_z$ és válasszunk véletlenszerűen újabb a_j elemet. Az új pár ismét (a_i, a_j).

9. Ellenőrizzük az eredményt az iteráció minden k-dik lépésében. Számoljuk újra a centrumok értékeit: $uc_1, uc_2, \ldots, uc_{k+t}$,

$$uc_l = \frac{1}{db_l} \sum_{s=1}^{db_l} y_s ,$$

ahol db_l az l-edik osztály (csoport) elemeinek száma, és FRM: $a_s \rightarrow y_s$ az osztály elemeihez rendelt értékek. Ha a

$$\left(\sum_{l=1}^{k+t} |c_l - uc_l| < \epsilon \right) \wedge (hiba < \gamma)$$

hibafüggvény értéke igaz (ahol "hiba" a hibás osztály-meghatározások száma), vége az eljárásnak. Különben $c_l = uc_l$, $l = 1, 2, \ldots, k + t$ és folytatás a 3. pontnál.

Ha véletlenszerűen vesszük elő az elemeket, megfelelő ro, η, ϵ és γ választása esetén az algoritmus konvergens lesz, és minden osztály centruma külön-külön konvergál egy stabil értékhez. E ponton hasonló a rendszer működése a neurális háló tanulásához. Pl. a tanulási ráta értékétől függően gyorsabb, lassabb lehet a konvergencia, vagy esetleg nem lesz konvergens a tanulási folyamat. A különböző paraméter értékek megválasztása függ az adott példától. Általában a $ro = 0.95$, $\eta = 0.0005$, $\eta = 0.01$ és $\gamma = n/10$ értékek mellett konvergens a tanulási folyamat.

A rendszer alkalmazása

Ha a tanulási folyamat befejeződik, megjeleníthetők az eredmények, ill. újabb esetek osztályozhatók a kész rendszerrel. Az algoritmus a centrumok, prototípusok, súlyszámok ismertében minden alternatívánál megállapítja melyik osztály prototípusához a leghasonlóbb, és a prototípusához tartozó osztályt rendeli az alternatívához. Ez a hozzárendelés tartalmazhat hibákat. Enyhítve a hasonlóság mértékét, és pl. minden alternatívánál a legnagyobb hasonlósági érték $- 0.05$ értékkel számolva, a nagyobb hasonlósági intervallum alapján egyes alternatíváknál két, vagy több osztályt is megadhat a rendszer mint lehetséges osztályt. Természetesen a hasonlóság tetszőlegesen tovább enyhíthető. (Egy osztály alternatíváihoz rendelt hasonlósági értékek olyan függvényt képeznek, amely egy osztály tartalmazási függvényéhez hasonlóan értelmezhető).

4. Fuzzy klaszter algoritmus

A fuzzy osztályozó algoritmus természetes kiterjesztéseként adódik a lehetőség, hogy osztály kialakításra, klaszterképzésre is alkalmassá tegyük az algoritmust. Klaszter kialakítás esetén a fuzzy rendszernek k darab különböző osztályba, klaszterbe kell sorolnia az alternatívákat úgy, hogy a leghasonlóbbak egy osztályba kerüljenek.

Ha a fuzzy osztályozó algoritmushoz hasonlóan itt is egy alternatívához azt az osztályt rendeljük, melynek prototípusához a leghasonlóbb, az előző feladatra visszavezethető a probléma. Ellentétben az osztályozással, az osztály hozzárendelés helyességét nem tudjuk ellenőrizni. Ellenőrizhető viszont most is bármely két elemhez rendelt függvényértékének sorrendje a hozzájuk rendelt osztályok centrumai alapján, valamint az egyes alternatívák és a prototípusok hasonlósága. Ha a pillanatnyi sorrend hibás, alkalmazható az előző súlymódosító szabály éles, vagy fuzzy változata.

Ha véletlenszerűen vesszük elő az elemeket, ez a heurisztikus algoritmus is konvergens lesz és folyamatosan módosítja az egyes osztályok elemeit, prototípusait, centrumait mindaddig, amíg a centrumok stabillá nem válnak.

Az algoritmus lépései

1. Definiáljuk az osztályok (ill. csoportok) induló üres halmazait $C = \{C_i\}$, $i = 1, 2, \ldots, k$. Legyenek $c_1, c_2, \ldots, c_k$, valamint $p_1, p_2, \ldots, p_k$ az osztályok centrumai, valamint prototípusai. A C_i halmazhoz tartozó centrum és prototípus legyen c_i, p_i. Legyen $c_i = 0$, $p_i = 0$ $(i = 1, 2, \ldots, k)$.

2. Válasszunk véletlenszerűen egy w_1, w_2, …, w_m súlyszám sorozatot. Válasszunk véletlenszerűen két indexet: i, j-t. Legyen az első a_i elem nulla. Az induló pár (a_i, a_j).

3. Legyen FRM: $a_j \rightarrow y_j$

4. Hasonlóság vizsgálat. Ha $H(a_j, p_z) = \max_q H(a_j, p_q)$ (alkalmazva az (1) képletet éles, vagy fuzzy esetben), ahol $q, z \in [1, k]$, akkor legyen $a_j \in C_z$. Legyen FRM: $p_z \rightarrow y$.

5. Prototípus módosítás.

   Ha $(c_z \neq 0) \wedge (c_z - y_j < c_z - y)$ akkor $p_z = a_j$, és ha $a_j = p_x$ $(x = 1, 2, \ldots, z-1, z+1, \ldots, k)$, akkor $c_x = 0$. Különben, ha $(\exists z)(c_z = 0)$, akkor $p_z = a_j$, és $c_z = y_j$.

6. Súlymódosítás.

   a) Ha $(oc < c_z \wedge y_i > y_j) \vee (oc > c_z \wedge y_i < y_j)$, akkor éles kritériumoknál a (2), fuzzy esetben a (3) súlymódosító képletet alkalmazzuk.

7. Súlyok normalizálása.

$$\|W\| = \sum_{i=1}^{m} w_i \qquad \text{és} \qquad w_i = \frac{w_i}{\|W\|}\,, \quad i = 1, 2, \ldots, m\,.$$

8. $i = j$, $oc = c_z$ és válasszunk véletlenszerűen újabb a_j elemet. Az új pár ismét (a_i, a_j).

9. Ellenőrizzük az eredményt az iteráció minden k-dik lépésében. Számoljuk újra a centrumok értékeit: uc_1, uc_2, …, uc_k,

$$uc_l = \frac{1}{db_l} \sum_{s=1}^{db_l} y_s\,,$$

   ahol db_l az l-edik osztály elemeinek száma, és FRM: $a_s \rightarrow y_s$ az osztály elemeihez rendelt értékek. Ha a

$$\sum_{l=1}^{k} |c_l - uc_l| < \epsilon$$

   hibafüggvény értéke igaz, vége az eljárásnak. Különben $c_l = uc_l$, $l = 1, 2, \ldots, k$, és folytatás a 3. pontnál.

Megjegyzés: Felvethető az algoritmus olyan továbbfejlesztése, amely osztályonként több prototípust alkalmaz. Ennek előnyei csak további teszteléssel mutathatók ki.

A rendszer alkalmazása

Ha a tanulási folyamat befejeződik, a kész rendszerben az eredmények az osztályozó algoritmushoz hasonlóan jeleníthetők meg, ill. újabb alternatívák osztályozhatók. Az algoritmus minden alternatívánál megállapítja melyik osztály prototípusához a leghasonlóbb az alternatíva és a prototípushoz tartozó osztályt rendeli hozzá. Enyhítve a hasonlóság mértékét, nagyobb hasonlósági intervallum alapján már az egyes alternatívákhoz két, vagy több osztályt is megadhat a rendszer, mint lehetséges osztályt. (Egy osztály alternatíváihoz rendelt hasonlósági értékek most is olyan függvényt képeznek, amely egy osztály tartalmazási függvényéhez hasonlóan értelmezhető).

5. Mintapéldák, tesztelés

Fuzzy osztályozó algoritmus alkalmazása

Első példa. Összehasonlításképpen a többek által benchmark-ként alkalmazott IRIS adatsort választottam teszt adatsornak, és eredményemet a NEFCLASS, valamint FuNeGen I. neuro-fuzzy osztályozó programokkal [Nauck 1995, Halgamuge 1994] is összevetettem (A programok szabadon elérhetők az interneten).

A 150 elemű IRIS adatsor, melyet Fischer (1934) publikált először, az írisz virág 3 típusát, osztályát írja le 4-4 adat alapján. E feladat összetettebb osztályozási problémát jelent, mivel két osztálya lineárisan nem szeparálható. Az adatsor egy részlete:

5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
...			

ahol az első $k1$ kritérium a $[4, 8]$, $k2$ a $[2, 5]$, $k3$ az $[1, 7]$ és $k4$ a $[0, 3]$ intervallumban veszi fel értékeit. Mivel a példához kapcsolódó fuzzy halmazokra nincs semmilyen megkötés, külön-külön a $[0, 5]$ intervallumra transzformáltam az adatokat és minden kritériumnál a nyelvi változók értékeit, mint "osztályzatokat" az $[i-1, i+1]$ $(i = 1, 2, \ldots, 5)$ intarvallumok felett szimmetrikus háromszög alakú tartalmazási függvényekkel adtam meg. Fuzzy rendező módszernek az "osztályozó"-módszert választottam.

Az adatsor két egyenlő részből: IRIS1 és IRIS2 áll (IRIS1: 75 elem, melyben 25 elem tartozik minden osztályhoz, és IRIS2: 75 elem, melyben szintén 25 elem tartozik minden osztályhoz). Training-re az adatsor első felét választottam és az adatsor második felével teszteltem a kész fuzzy rendszert. A súlymódosításon alapuló osztályozó módszer (jelöljük FCW-vel) szinte azonos pontosságú eredményeket adott a másik két módszerrel (1. ábra):

	IRIS1	IRIS2
FuNeGen I.	99.0%	99.0%
NEFCLASS	96.0%	97.3%
FCW	97.3%	97.3%

1. ábra IRIS adatsor összehasonlító eredményei 3 prototípus esetén (a százalékok a helyes osztályozásokat jelentik).

Ha enyhítjük a hasonlóság mértékét és minden alternatívánál a kapott maximális hasonlósági értéknél (MHi) 0.05-el kisebb értékeket is elfogadunk, a hibás osztályozások száma háromról kettőre csökken és a korábbi tévesen osztályozott alternatívához két osztályt fog rendelni. Ha a hasonlóságot tovább enyhítjük, csak a [MHi-0.3, MHi intervallumba eső értékeknél tűnnek el a téves osztályozások, viszont ekkor az alternatívák 49%-ánál (37-nél) 2, vagy 3 osztályt rendel már minden alternatívához. A hasonlóság mértékétől függően tehát a hibás osztályozások, ill. egy alternatívához rendelt osztályok száma változó (2. ábra):

	Hasonlóság mértéke $> MHi - \delta$				
	$\delta = 0$	$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.3$
téves osztályozás	3	2	2	2	0
több osztály hozzárendelés	0	3	9	23	37

2. ábra Az eredmények és a hasonlóság kapcsolata $(i = 1, 2, \ldots, 75.)$

Az eljárás konvergenciáját a 3. ábra szemlélteti, amely különböző paraméter értékek esetén mutatja a konvergencia gyorsaságát (iterációk száma) és az osztályozások helyességét (százalékok). A $+++$ jelölés az adott paraméterek mellett nem konvergens tanulást mutatja (ebben az esetben egyébként a pontosság kb. 75%-os).

t. ráta	ro	Prototípus szám	ϵ	Kritérium	IRIS1	IRIS2	Iteráció
0.0005	-	3	0.001	Fuzzy	97.3%	97.3%	6751
0.05	0.85	5	0.001	Fuzzy	96.0%	96.0%	2326
0.05	0.9	5	0.001	Fuzzy	80.0%	80.0%	885
0.005	0.85	5	0.001	Éles	98.5%	98.5%	5116
0.0005	0.9	5	0.1	Éles	98.5%	98.5%	976
0.0005	0.95	11	0.1	Fuzzy	98.5%	100.0%	724
0.0005	0.9	5	0.01	Fuzzy			$+++$

3. ábra Paraméterek és a konvergencia összefüggése

Második példaként állattartási peresetek osztályozását nézzük. L. Philipps (1994) számol be arról, hogy állattartással kapcsolatos bírói döntéseket vizsgálva, az állattartó érdekei és felelőssége alapján hat jellemző tulajdonsággal, kritériummal írhatók le az esetek. A hat tulajdonság ($k1$,$k2$,...,$k6$) értékeit fuzzy értéknek tekintve, fuzzy logika műveletekkel állította elő a bírói döntést. 18 eset adatait felhasználva csupán 3 esetben tért el az eredmény a bírói döntéstől.

Ezen adatok (4. ábra) osztályozhatók is, bár lényegesen összetettebb osztályozási problémát jelentenek, mint az IRIS adatok osztályozása. Fuzzy osztályozó módszerekkel a bírói igen/nem döntésnek megfelelően pontosan besorolhatók két osztályba az esetek.

Tekintsük most is az adatokat osztályzatoknak, és alkalmazzuk az előbbi példa fuzzy halmazait. A kritérium értékekre a következő megfeleltetéseket fogadjuk el: null =0; nagyon csekély=0.5; csekély=1; közepes = 2.5; magas = 3.75; nagyon magas =4.5; teljes = 5. Osztályozva az eseteket mind a FCW, mind a FuNeGen I. és a NEFCLASS osztályozó programokkal, az eredmények jobbak lettek, mint a fuzzy logika műveletekkel kapott értékek. A bírói döntést nulla, vagy egy hibával reprodukálták (5. ábra).

	k1	k2	k3	k4	k5	k6	bírói d.
1.	n.csekély	csekély	csekély	null	teljes	közepes	nem
2.	teljes	null	csekély	magas	null	magas	igen
3.	csekély	közepes	közepes	közepes	teljes	közepes.	igen
4.	teljes	csekély	teljes	teljes	null	magas	igen
5.	null	n.magas	közepes	magas	n.magas	n.magas	igen
6.	teljes	n.csekély	közepes	csekély	n.csekély	n.csekély	igen
7.	null	null	közepes	csekély	n.csekély	közepes	nem
8.	közepes	magas	csekély	magas	teljes	közepes	igen
9.	csekély	közepes	teljes	null	teljes	null	nem
10.	teljes	teljes	magas	magas	null	teljes	igen
11.	csekély	közepes	null	n.csekély	teljes	n.csekély	nem
12.	teljes	teljes	teljes	teljes	teljes	teljes	igen
13.	null	közepes	csekély	teljes	teljes	null	igen
14.	teljes	közepes	magas	null	null	teljes	igen
15.	null	közepes	n.csekély	magas	csekély	teljes	igen
16.	null	csekély	közepes	n.csekély	közepes	csekély	nem
17.	csekély	magas	n.csekély	közepes	teljes	közepes	igen
18.	teljes	teljes	teljes	teljes	közepes	teljes	igen

4. ábra. Jogesetek adatai

	Helyes osztályozások	
	száma	százaléka
FuNeGen I. (10 generált szabállyal)	17	94.4%
NEFCLASS (18 generált szabállyal)	18	100%
FCW	18	100%

5. ábra Jogesetek osztályozása

(A hasonlóság mértékét e feladatnál nem kell enyhíteni, minden osztály besorolás helyes). A 100%-os pontosságot 0.0005 tanulási ráta, 2 prototípus, fuzzy kritériumok alkalmazásával, $\gamma = 1$ esetén 1218 iteráció után kaptuk.

Harmadik példaként nézzünk egy olyan feladatot, amely csak plusz prototípusok alkalmazásával oldható meg elfogadható pontossággal. Válasszuk a relatíve magas belső szórású osztályokat tartalmazó következő feladatot: két éles kritériummal adott elemeket kell két osztályba sorolni, ahol az elemek: $C_1 = \{(0,6),(12,0),(1,8),(1,10),(14,4)\}$; $C_2 = \{(3,12),(15,6),(16,8),(5,16),(17,10)\}$

Az osztályozó algoritmus a prototípusok számától függően különböző gyorsasággal, pontossággal oldotta meg a feladatot. A 100%-os megoldást már 4 prototípus alkalmazása mellett nyújtotta (6. ábra)

t. ráta	ro	Prototípus szám	Pontosság	Iteráció
0.005	-	2	80%	2400
0.0005	0.6	4	100%	55
0.0005	0.9	5	90%	310
0.05	0.8	5	80%	1160
0.0005	0.8	6	100%	55
0.0005	0.95	9	90%	2060

6. ábra Különböző paraméterértékeknél kapott eredmények ($\epsilon = 0.01$, $\gamma = 2$).

Fuzzy klaszter algoritmus alkalmazása

Osztály kialakítására nézzük először ismét az *IRIS adatsort*. Ha az előző megközelítést választjuk: azaz a $[0,5]$ intervallumra transzformálását a kritérium értékeknek, akkor alkalmazhatjuk az osztályképző algoritmust. Mivel az IRIS adatsor osztályozását ismerjük, utólag ellenőrizhetjük az osztályok megfelelő kialakítását.

Ismét az „osztályozó" fuzzy rendező módszert választva az algoritmus mind a training-halmaznál (IRIS1), mind a teszthalmaznál (IRIS2) azonos, 95%-os pontossággal az ismert osztályozást adta vissza. A hasonlóság mértékének enyhítésével, $[MHi - 0.05, Mhi]$ intervallumbeli hasonlósági értékeknél már eltűntek a hibák, és 7 alternatívát 2-2 osztályba sorolt (összehasonlításra más fuzzy klaszterező szoftverek nem álltak rendelkezésre).

Osztály kialakításra *második példaként* nézzünk egy szokványos, valós számokkal adott feladatot (éles változat). Legyen adott 25 ország néhány statisztikai adata (7. ábra), és soroljuk 5 osztályba az országokat.

	k1	k2	k3	k4	k5	k6	k7
Afganisztán	40.4	18.7	21.6	1881.6	41.0	42.0	7.6
Bangla-desh	42.2	15.5	26.7	119.0	56.9	55.9	5.53
Kambodzsa	41.1	16.6	24.8	130.0	47.0	49.9	4.71
Kína	21.2	6.7	14.5	32.0	68.0	70.9	2.45
Ciprus	188.6	8.2	10.4	12.0	73.9	78.3	2.33
Egy. Arab E.	22.8	3.8	19.0	26.0	68.0	72.9	4.82
Hong-Kong	11.7	4.9	6.8	6.1	68.6	80.0	1.23
India	30.5	10.2	20.3	91.0	52.5	52.1	4.2
Indonézia	28.6	9.4	19.2	75.0	58.5	62.0	3.48
Irán	42.5	11.5	31.0	108.1	55.75	55.04	5.2
Irak	42.6	7.8	34.8	69.0	63.0	64.8	6.35
Izrael	22.3	6.3	16.0	9.7	73.87	77.44	3.03
Japán	9.9	6.7	3.3	4.5	75.9	81.77	1.54
Kuvait	26.8	2.2	24.6	15.6	71.2	75.4	4.03
Mongólia	36.1	8.8	27.3	68.0	60.0	62.5	5.0
Nepál	39.6	14.8	24.8	128	50.88	48.1	5.94
Pakisztán	30.3	8.1	22.2	107.7	59.04	59.2	6.48
Fülöp-sz.	33.2	7.7	25.5	45.0	62.5	66.1	4.33
Szaud-A.	42.1	7.6	34.5	71.0	61.7	65.2	7.17
Szingapúr	17.8	5.2	12.6	7.5	68.7	74.0	1.97
Sri-L.	21.2	6.2	125.1	19.4	67.78	71.6	2.96
Szíria	44.6	7.0	37.6	48.0	64.42	68.05	6.76
Thaiföld	22.3	7.7	15.3	28.0	63.82	68.05	2.6
Törökо.	29.2	8.4	20.8	76.0	62.5	65.8	3.69
Vietnam	31.8	9.5	22.3	64.0	63.66	67.89	

7. ábra. Országok adatai (ahol k1: élveszületési ráta, k2: halálozási ráta, k3: természtes növekedési ráta, k4: csecsemőhalandósági ráta, k5: férfiak várható élettartama, k6: nők várható élettartama, k7: termékenységi ráta)

Bár most az adatokat nem tekintjük fuzzy értékeknek, az algoritmusnál felhasznált fuzzy rendező módszerhez továbbra is szükséges a fuzzy értelmezés. Akárcsak az előző esetekben, alkalmazzuk most is a fuzzy "osztályzatokat" és transzformáljuk a kritériumok értékét a $[0, 5]$ intervallumra. Az osztály kialakító algoritmus éles távolságokkal fog számolni, de a függvényértékeket továbbra is a FRM határozza meg.

Az eredmény ellenőrzéshez két különböző módszerrel is készítettem osztályokat. A statisztika egyik hierarchikus klaszteranalízis programjával, valamint egy neurális háló módszerrel, Kohonen önszervező térképével (SOM). Mind a három eredmény hasonló lett, de a fuzzy klaszterek tértek el a legjobban a többitől. A fuzzy klaszter algoritmus eredménye egyébként legjobban a statisztikai módszer eredményéhez hasonlítható (8. ábra).

Ez az eredmény már nem olyan jó, mint az osztályozás eredményei. A fuzzy algoritmussal csak 75-80%-ban kapunk azonos eredményt az éles módszerek eredményével. Ha átfedő osztályokat alkalmazunk, és a hasonlóság mértékét enyhítjük, a [MHi-0.15, Mhi] intervallumbeli értékekre már 12 országot két, esetleg három osztályba sorol és a kritikus, korábban eltérést mutató helyeken "megfelelő" osztályt is választhatunk. (Az algoritmus most is hasonló konvergenciát mutatott a fuzzy osztályozó algoritmuséval: 0.0005 tanulási ráta és $\eta = 0.001$ mellett 1024 iteráció után stabilizálódtak a centrumok.)

klaszteranalízis	neurális háló	fuzzy klaszter
AAABC	AAABC	AAACC
BCDDA	BCADA	CCDDA
DCCBD	DCCCD	BCCCD
AAEDC	AAEDC	ADEBC
BEBDD	CEBDE	CECDD

8. ábra. Osztályozás eredményei

5. Összefoglalás

A bemutatott két heurisztikus algoritmus közös jellemzője, hogy a rendezni kívánt alternatívákból a neurális hálók tanulásához hasonlón határozza meg az osztályozás, vagy klaszterezés alapját képező információkat. Az osztályozás helyességét olyan hibafüggvénnyel ellenőrzi, amely egy fuzzy rendező módszer eredményeit is felhasználja. A helyes osztályozás kialakítása érdekében a kritériumok súlyait, melyek általában a fuzzy rendező módszer szabályinak súlyai is egyben, a neurális hálózatok delta szabályához hasonló képlettel változtat egy iteratív folyamatban.

A tanulási folyamat végén minden osztályhoz egy centrumot, az osztályra jellemző prototípust, ill. a fuzzy rendező módszer szabályainak súlyait határozzák meg az algoritmusok. A fuzzy osztályozó algoritmus egy osztályon belül több csoportot és csoportonként külön centrumot, prototípust is értelmezhet; ezzel a nagyobb belső szórású osztályok helyes felismerésére is képes. A tanulás konvergenciája a neurális hálózatokhoz hasonló tulajdonságú.

Egy kész rendszer az éles, vagy fuzzy kritériumok ismeretében, valamint a kapott prototípusok, súlyok felhasználásával a továbbiakban önállóan képes osztályozni az elemeket. A tesztek eredményei alapján a fuzzy osztályozást, klaszterképzést 97-100%-os pontossággal valósítják meg, lehetővé téve átfedő, fuzzy osztályok értelmezését is. Ezen eredmények pl. a neurofuzzy osztályozó módszerekkel hasonló pontosságúak.

Irodalom

1. Bezdek, J. C., Hathaway, R. H.: Numerical Convergence and Interpretation of the Fuzzy C-shells Clustering Algorithm. IEEE Trans. Neural Networks 3 1992. 787–793.
2. Borgulya, I.,: Egy osztályzaton alapuló fuzzy rendező módszer. Szigma XXVI. 1995. pp. 117–132.
3. Borgulya, I.,: A Ranking Method for Multiple Criteria Decision Making. International Journal of Systems Science, 1997a. 28. pp. 905–912.
4. Borgulya, I.,: Generalisation Weightnumbers of Criteria by Fuzzy Systems. In: Proceeding of the 19th Int. Conf. Information Technology Interfaces 1997b. Pula. pp. 99–104.
5. Duda, R., Hart, P.: Pattern Classificaction and Scene Analysis. Wiley, New York, 1973.
6. Fischer. R. A., The Use of Multiple Measurements in Taxonomic Problems. Annual Eugenics, 7. 1936. (Part II). pp. 179–188.
7. Gustafson, E. E., Kessel, W. C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. IEEE CDC, San Diego, kalifornien 1979. 761–766.
8. Halgamuge, S. K., Glesner, M., Neural Networks in Designed Fuzzy Systems for Real World Applications. Fuzzy Sets and Systems, 65. 1994. pp.1–12.
9. Höppner, F., Klawonn, F., Kruse, R.: Fuzzy-Clusteranalyse Vieweg, Braunschweig/Wiesbaden, 1997.
10. Munda, G., Multicriteria Evaluation in a Fuzzy Environment. Heidelberg, Physica-Verlag. 1995
11. Nauck. D., Kruse.R., NEFCLASS - A Neuro-Fuzzy Approach for the Classification of Data. In: Proc. of the 1995 ACM Symposium on Applied Computing. ACM Press, pp. 461–465.
12. Philipps L.: Kompensatorische Verknüpfungen in der Rechtsanwendung – ein Fall für Fuzzy Logic. In: Martinek M., Schmidt J., Walde E.: Festschrift für Günther Jahr zum siebzigsten Geburtstag. J. C. B. Mohr, Tübingen 1994. pp. 169–180.
13. Yager. R. R., Fuzzy Decision Making Including Unequal Objectives. Fuzzy Sets and Systems. 1, 1978. pp. 87–95.
14. Yager R. R., Filev D.: Generation of Fuzzy Rules by Mountain Clustering. Journal of Intelligent & Fuzzy systems, Vol. 2, No. 3, pp. 209–219.

TWO FUZZY CLASSIFICATION METHODS

Two heuristic fuzzy algorithms are shown in this paper: a fuzzy classification and a fuzzy cluster algorithm. The algorithms classify multiple-criteria fuzzy or crisp alternatives and the task of the classification is traced back to ranking of alternatives, as well as to learning weight numbers of criteria similarly to neural networks. The algorithm is suitable to separate classes with relatively high intrinsic scatter.