4

Nem-hierarchikus osztályozás

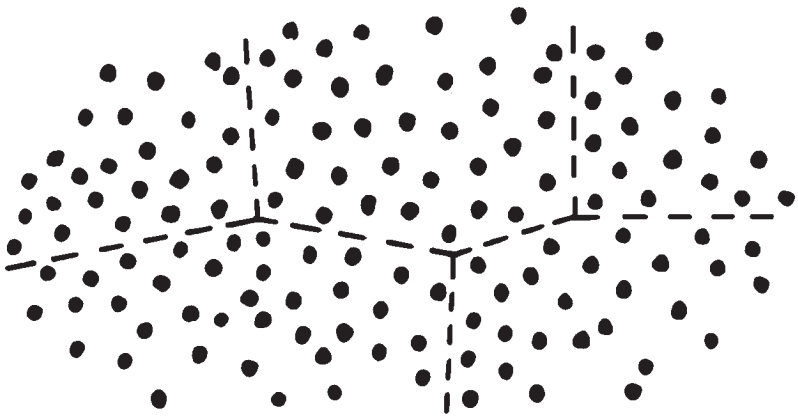
(Egy "ősi tevékenység"... modern formában)

Különféle dolgok csoportokba rendezése, az *osztályozás*, egyik alapvető intellektuális tevékenységünk: nélküle el sem tudnánk igazodni a bennünket körülvevő világban. Csak egyetlen fontos példát említve: a kommunikáció eszköze, a nyelv, elválaszthatatlan az osztályozástól, hiszen a dolgok megnevezése már eleve feltételez valamilyen kategorizálást. A nyelv fejlődése így az osztályozás kifinomulásával egyidejű, attól elválaszthatatlan folyamat[1]. Az osztályozásnak különösen nagy "rendteremtő" szerepe van olyan tudományterületeken, ahol a dolgok sokfélesége, változatossága zavarba ejtően nagymérvű. A szupraindividuális szintű biológiára gondolunk elsősorban, melynek története mindenkor szorosan összefonódott az osztályozással kapcsolatos elvek és módszerek változásával, fejlődésével.

Az osztályozás fogalmának szabatos meghatározása a matematikában az *ekvivalencia-relációkon,* ill. a *halmazokon* alapszik (lásd Izsák et al. 1981:31). Az osztályozás a vizsgált objektumok részhalmazokra (itt: osztályokra) történő felosztása (partíciója) olymódon, hogy a kapott osztályok páronként teljesen elkülönültek (diszjunktak, azaz egyik objektum sem tartozhat egyidejűleg két részhalmazba). Ez a definíció csak az ún. *nem-hierarchikus* vagy *particionáló* módszerek esetében érvényes (jelen fejezet 4.1.1-4 részei). A klasszikus meghatározás kisebb vagy teljes mértékű módosításaival jutunk el a később tárgyalandó *átfedéses*, valamint a *lágy ("fuzzy")* és a hierarchikus osztályozásokhoz.

Érdekes nyelvi sajátosság (s ez nemcsak a magyarban van így) az osztályozás szó kétszeresen kettős jelentése: nemcsak az *eredményt*, hanem az azt létrehozó *folyamatot* is osztályozásnak nevezzük. Ez különösebben nem lehet zavaró, hiszen a kontextusból mindig kiderül, hogy éppen eredményről vagy pedig műveletek sorozatáról, valamilyen algoritmusról van-e szó. Annál több félreértésre adhat okot a másik kettősség, amelyet célszerű jó előre tisztázni. Összhangban a numerikus taxonómia irodalmával (pl. Sneath & Sokal 1973), az osztályozás folyamatán egy olyan műveletsorozatot értünk a továbbiakban, melynek révén ed-

1 Az osztályozás képességét azonban nem lenne szabad kizárólag emberi "előjognak" tekinteni, gondoljunk például az állatok világára: az ehető – nem ehető növények felismerése, vagy a fajtársak, nem fajtársak és ellenségek megkülönböztetése is osztályozásnak tekinthető.



4.1 ábra. Egy viszonylag egyenletesen sűrű erdő fáinak beosztása szektorokra – annak érdekében például, hogy erdei utakon minden erdőrészlet jól megközelíthető legyen – nem tekinthető osztályozásnak. A felosztás ugyanis nem a ponthalmaz szerkezetén alapszik elsősorban.

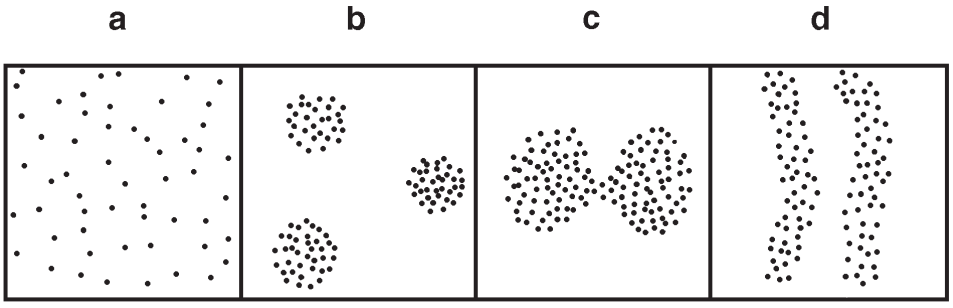
dig *még nem létező osztályokat hozunk létre*. Ezzel szemben a köznyelvben is, de a matematikában is sokszor nevezik osztályozásnak azt, amikor új objektumokat helyezünk el egy már létező osztályozás valamelyik csoportjába. Ez utóbbi műveletet célszerű *azonosítás* (*besorolás, identifikáció*) néven különválasztani az osztályozás egészét megteremtő procedúrától. (Az új osztályozás létrehozása és a besorolás közötti különbség a számítógépes algoritmusokat tekintve a leginkább szembeötlő.[2])

További lényeges szempont, hogy az osztályozás az osztályozott objektumok jellemzőit fejezze ki, az adattérben lévő csoportosulásokat tükrözze. Nem tekintjük tehát osztályozásnak az objektumhalmaz egyszerű "felszeletelését" (*dissection*, Kendall 1966; 4.1 ábra). Ekkor ugyanis nem az objektumok közötti távolság- vagy hasonlóságviszonyok kifejezése a lényeges, hanem *külső* célszerűségi szempontok uralkodnak, amelyeket ráerőltetünk az objektumhalmazra (mint például egy város épületeinek kerületekre történő felosztásában, vagy egy erdő művelési szektorokra bontásában). A 4.1 ábra sűrűn álló, viszonylag egyenletesen elszórt pontjait intuitíve mindenki egyetlen osztályba tartozónak tekintené egyébként is. Az egyenletesség azonban nem az egyetlen ok, hogy a felosztást ne tekintsük osztályozásnak. A randomizáltan elhelyezkedő pontokat se tudjuk értelmes módon osztályokba sorolni, amint azt a 4.2a ábra is szemlélteti.

2 Az osztályozás szakirodalma igen gyakran *"cluster analysis"* vagy *"clustering"* néven utal az adatokban rejlő csoportosulásokat kimutató numerikus módszerekre. Ennek magyarítása a "fürtelemzés" szóval (Füstös et al. 1986) nem volt szerencsés próbálkozás, és nem is honosodott meg a szakzsargonban. A "számítógépes csoportosítás" talán jobban megfelelő kifejezés lenne, annál is inkább, mert a számítógép ma már nélkülözhetetlen ehhez a művelethez. A besorolás tematikája szorosan összefügg a mintázat- (alak-) felismerés szerteágazó tudományterületével, s a fenti kontraszt a *"supervised versus unsupervised pattern recognition"* megkülönböztetés formájában jelentkezik (Therrien 1989).

Felmerül a kérdés: milyen jellegű objektum-objektum kapcsolat esetén beszélhetünk értelmes osztályozásról? Az előző fejezetben megadott távolságok (pl. az euklidészi távolság) felhasználásával egy osztályozást két fő szempont szerint jellemezhetünk: 1) az osztályok belső *kohéziója,* amelyeket az osztályokon *belüli* távolságok segítségével fejezhetünk ki, és 2) az osztályok *szegregációja*, az osztályok *között* mutatkozó távolságok alapján. Ideális esetben az osztályok kohéziója és szegregációja is egyaránt erős (4.2b ábra), ekkor az osztályok jellemzése és elhatárolása egyértelmű s szinte minden módszer azonos eredményre vezet. A gyakorlatban ilyen esetben már "ránézésre" is nyilvánvaló lehet az osztályozás, s a számítógépes csoportosítást nem az osztályok kimutatására, hanem létük igazolására, vagy csupán a klasszifikáció szemléltetésére alkalmazzuk. Speciálisabb esetet jelentenek az erős kohézióval, de a szegregáció hiányával jellemezhető osztályok (4.2c ábra). Ezeket a legtöbb módszer többé-kevésbé érzékeli, de az "átmenetinek" tekinthető, a szegregációt csökkentő objektumok osztályozásában már nagy eltérések mutatkozhatnak. A másik szélsőséget a 4.2d ábra csoportjai képviselik, kifejezett szegregációval és nagyon gyenge belső kohézióval. Az ilyen osztályokat már kevesebb módszer képes felismerni, mint azt a későbbiek során látni fogjuk. A két véglet között természetesen átmenetek végtelen sorozata képzelhető el, s ezek jelentik az igazi problémát az adatelemző kutató számára.

Az eddigiek alapján azt várnánk, hogy a numerikus klasszifikáció során az osztályok kohézióját és szegregációját egyidejűleg fogjuk optimalizálni. Az egyes eljárások azonban nem kezelik egyformán ezt a két alaptulajdonságot: többnyire csak a kohéziót veszik figyelembe közvetlenül (bár látunk majd kivételeket is). Az algoritmusok viszonylag egyszerűek, bemutatásuk és megértésük nem igényel különösebb matematikai ismereteket. Indokolt tehát ezeket elsőként, minden más módszert megelőzve tárgyalni. (Ebből azonban nem következik az, hogy a particionálás jelenti a többváltozós vizsgálódás első lépését. Éppen ellenkezőleg: a nemhierarchikus osztályozásra rendszerint akkor kerül sor, ha más típusú elemzések révén már vannak bizonyos ismereteink az adataink szerkezetéről.)



4.2 ábra. Pontok csoportosulásának speciális esetei kétdimenziós térben. **a:** random elrendeződés, valódi osztályszerkezet nélkül, **b:** "ideális" eset, az osztályok erős kohéziójával és szegregációjával, ***c*:** két osztály erős kohézióval de szegregáció nélkül, **d:** megnyúlt pontfelhők melyek belső kohéziója kicsiny, elválásuk viszont jól érzékelhető.

4.1 Particionáló módszerek

Feladatuk, hogy *m* objektum hagyományos értelemben vett felosztását állítsák elő *k*, páronként diszjunkt osztályra (csoportra)[3]. Egy objektum így csak egy osztályba tartozhat és értelemszerűen minden osztályban van legalább egy objektum (egyébként nem beszélhetnénk *k* osztályról). Az eljárások általában egy *iteratív* stratégián alapulnak: az analízis során egy kezdeti osztályozást javítunk lépésről lépésre mindaddig, amíg további javulást már nem érhetünk el. A kezdeti osztályozás megadása azt jelenti, hogy az osztályok számát, *k*-t, előzetesen ismerjük. Tegyük fel, hogy az osztályozás optimalitását ("jóságát") valamilyen *J* függvénnyel mérjük, melynek értékét a további javítás érdekében csökkentenünk kell az egyes lépésekben. Ezek alapján megadható egy általános particionáló algoritmus (Hartigan 1975, Therrien 1989):

1. Válasszunk ki egy kezdeti osztályozást *k* csoportra és számítsuk ki *J* értékét.

2. Változtassuk meg az osztályozást olymódon, hogy *J* maximálisan csökkenjen *k* változatlan értéke mellett (ne keletkezzen "üres" vagy új osztály).

3. Ha a 2. lépésben nem lehetséges *J* csökkentése, az elemzés megáll és az adott osztályozást fogadjuk el végeredménynek. Ellenkező esetben visszatérünk a 2. lépéshez.

A módszerek az osztályozás jóságát mérő *J* függvényben és az osztályozás 2. lépésbeli megváltoztatásában térnek el egymástól. A fenti particionálási algoritmusra jellemző, hogy a kapott végeredmény esetleg csak egy *lokális optimum*, azaz nem a lehető legjobb osztályozás az adott objektumokra. Lehetséges ugyanis, hogy egy másik kiindulásból *J*-nek egy még alacsonyabb értéke is elérhető. Ezen a problémán rendszerint enyhíthetünk azzal, hogy az elemzést sokszor, különböző kiinduló osztályozásokból is végrehajtjuk s a kapott eredmények közül a legjobbat tartjuk meg. Voltaképpen azonban sohasem lehetünk 100 %-ig biztosak abban, hogy az így kapott végső osztályozás lesz az abszolút optimális (*globális optimum*). Bizonyosat csak akkor állíthatnánk, ha minden lehetséges osztályozásra kiszámítanánk *J* értékét, de ez *m* nagy értékeire megvalósíthatatlan feladat lenne.

Az osztályozás megváltoztatása a 2. lépésben kétféleképpen történhet:

- Az objektumok mindegyikére külön-külön megvizsgáljuk, hogy melyik osztályba áthelyezve csökkentik legnagyobb mértékben a *J* értékét. Azokat az objektumokat, amelyeknél csökkenés mutatkozik, áthelyezzük abba az osztályba, amelyre ez a csökkenés maximális. Az áthelyezés akár az összes objektumot is érintheti s remélhető, hogy az új *J* érték a sok áthelyezés következtében végül is alacsonyabb lesz, mint az előző (vö. Therrien 1989).
- Kiválasztjuk azt az objektumot, amelyre a *J* csökkenése maximális, s csak ezt helyezzük át az új osztályba. Ez a stratégia a *J* mennyiség monoton csökkenéséhez vezet, bár lassabb az előzőnél.

3 Eme hagyományos osztályozásokra *kemény* (azaz *"hard"* vagy *"crisp")* partíciók néven hivatkoznak a legújabb szakirodalomban, utalva arra, hogy a felosztás más típusú, pl. lágy ("fuzzy") is lehet (vö. 4.3 rész).

4.1.1 A k-közép módszer

A particionáló módszerek klasszikus példája a *k-közép eljárás* és különféle változatai (pl. Forgy 1965, Jancey 1966, MacQueen 1967):

1. Kiválasztunk valamilyen – akár önkényes – kezdeti osztályozást k csoportra.

2. Kiszámítjuk a súlypontot (azaz az összes változóra vonatkozó átlagértékeket) minden osztályra.

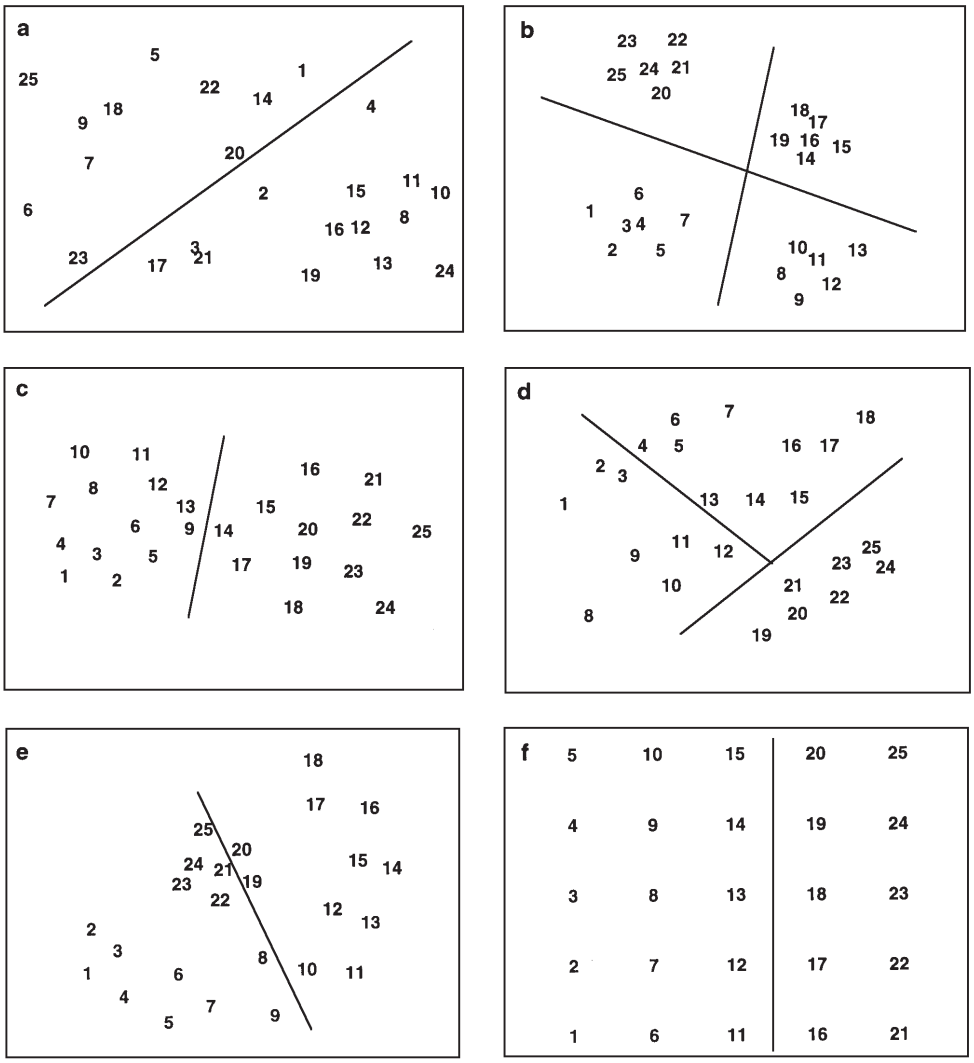
3. Meghatározzuk minden objektum euklidészi távolságát a hozzá tartozó súlyponttól. A jósági kritériumot ezen távolságok négyzetösszegével definiáljuk:

$$J = \sum_{h=1}^{k} \sum_{j \in A_h}^{m_h} \sum_{i=1}^{n} (x_{ij} - z_{ih})^2 \qquad (4.1)$$

ahol z_{ih} az A_h osztály súlypontja ("közepe", innen az elnevezés) az i változóra nézve, m_h az A_h osztály elemszáma (eszerint van a második összegzés), n a változók száma. J tehát az eltérésnégyzet-összeg (amely a 3.106 egyenlet szerint kiszámítható az osztályon belüli objektumok páronkénti távolságaiból is). Ha vannak objektumok, amelyek áthelyezése csökkenti J értékét, akkor azokat átsoroljuk s visszatérünk a 2. lépéshez. Ha nincs egy ilyen objektum sem, az iteráció leáll.

A fenti eljárás "lassú" változata csak egy áthelyezést enged meg minden lépésben. Egy másik változtatási lehetőség, hogy az eltérésnégyzet-összeg kiszámítása elmarad és minden objektumot egyszerűen a hozzá legközelebb eső osztályba sorolunk át. (Ez – ellentétben esetleges várakozásunkkal – nem vezet feltétlenül az előzővel azonos eredményre, mint majd látni fogjuk a 4.1.3 részben). A módszer csak olyan esetekben használható, amikor adataink átlagolhatók és az euklidészi távolság is kiszámítható (pl. a nominális és az ordinális típusú változók kizárandók). Az osztályozás annál jobb, minél nagyobb az osztályok kohéziója (azaz minél kisebb az eltérésnégyzet-összeg). A szegregációt viszont közvetlenül nem mérjük.

A módszer hipergömb alakú, nagy belső kohéziójú ("konvex") pontsereget ismer csak fel, a sok dimenziós térben elnyújtott pontalakzatokat több osztályra is felbonthatja még akkor is, ha azok szegregációja kifejezett. Érdemes tehát figyelembe vennünk a 4.3 ábrát, amely megmutatja, hogy bizonyos tipikus esetekben milyen eredményre jutunk a k-közép módszerrel (erre az ábrára még későbbi fejezetekben is utalni fogunk majd, mert az ábra kétdimenziós ponteloszlásai például a hierarchikus osztályozó módszerekkel történő összevetésre is alkalmasak lesznek). A szerkezet nélküli, random ponthalmazt egyszerűen "megfelezte" az átló mentén (4.3a ábra), a jó szegregációjú és kohéziójú osztályokat gond nélkül elkülönítette (4.3b ábra). A 4.3c ábra nem elváló két osztálya között a határt a 13. és 14. objektumok között húzta meg. (Megjegyzendő, hogy a 14. pont, értékeinek nagyon kis megváltoztatására, már átkerül a másik csoportba, mutatva az ilyen osztályozás viszonylag kis stabilitását.) A k-közép módszer, mint fent említettük, nem képes a hosszú pontfelhők elkülönítésére (4.3d ábra), s akkor is "zavarba jön" ha egy ívelt pontfelhő vesz körül egy másik, viszonylag tömör csoportosulást: mindkettőt kettévágja (4.3e ábra). Az osztályszerkezetet teljességgel nélkülöző, közelítőleg egyenletes pontelrendeződésben, ha k értékét 2-nek választjuk, a kapott csoportok egy "felszeletelésnek" tekinthetők csupán.



4.3 ábra. A *k*-közép módszer eredménye a kétdimenziós adatszerkezet hat alapesetére, *m*=25. Az iterációk 10-10 random kiindulásból történtek, s a legjobb felosztásokat választottuk ki. Az eltérésnégyzet-összegeket nem közöljük, mert az értékek *nem összemérhetők* egymással, annak ellenére, hogy a pontok száma azonos minden esetben. **a:** random ponteloszlás, *k* értékét 2-nek választva, **b**: négy "ideális" osztály, **c:** szegregáció nélküli jó kohéziójú osztályok, **d:** három megnyúlt pontfelhő (=kicsiny kohézió), **e**: kis osztály amelyet egy ívelt, rosszabb kohéziójú osztály ölel körül, **f:** majdnem teljesen szabályos ponteloszlás, amelyet *k*=2 értéke mellett próbálunk particionálni. Az adatokat az A3 táblázat foglalja össze x és y koordináták formájában.

A kezdő osztályozást az alábbiak szerint adhatjuk meg:

- Random osztályozás. Az osztályba tartozást a véletlen dönti el, ezért relatíve több lépésben jutunk el az iteráció végéhez, mint amikor, pl. egy nem önkényes kezdeti osztályozásból indulunk ki.
- Más értékelésből származó végeredmény (pl. hierarchikus osztályozás egy adott szinten, vö. 5. fejezet). Ekkor a kiindulás nagy valószínűséggel előnyösebb az előzőnél, de lehet, hogy csak egy lokális optimumra vezet.
- A felhasználó előre megad *k* számú ún. magpontot, s az összes többi objektumot a magpontoktól való távolság alapján sorolja be a kiinduló osztályokba. Akkor célszerű használata, ha bizonyos tipikus objektumokhoz keresünk jól illeszkedő klasszifikációt. (Természetesen a lokális optimum lehetősége itt is fennáll).
- A magpontokat véletlenszerűen választjuk ki, s ezzel lényegében véve random osztályozást kapunk.
- A kiinduló *k* magpontot az *n*-dimenziós térben egymástól legtávolabb eső *k* objektum jelenti. Az első magpont az összes objektum súlypontjától legtávolabb eső objektum, a második az első ponttól legtávolabbi objektum, a harmadik magpont az, amelynek távolságai az előző kettőtől maximálisak, és így tovább *k*-ig. Ez a kiindulás érzékeny lehet atipikus, osztályba nehezen sorolható objektumok ("outlier"-ek) jelenlétére.
- Egy optimális, *k*–1 osztályt tartalmazó partícióból indulunk ki, s az új osztály kezdőpontjaként a saját osztálya súlypontjától legtávolabb eső objektumot választjuk (Hartigan 1975). Ezt alkalmazzuk a többszörös particionálás néven külön tárgyalt módszernél is (lásd a 4.1.3 részt).

Egyéb kezdési lehetőségeket tárgyal Anderberg (1973: 157-160).

A *k*-közép módszer egy rugalmas módosítása az ISODATA eljárás (Ball & Hall 1965), amelyben *k* rögzítéséhez már nem ragaszkodunk olyan szigorúan (az osztályok száma bizonyos esetekben az analízis során megváltozhat), s a szegregációt is figyelembe vesszük. Ennek ára azonban az, hogy további paraméterek válnak szükségessé, és ez több szubjektív elemet visz az elemzésbe. Az ISODATA eljáráshoz meg kell adnunk a minimális osztályméretet (az ennél kisebb osztályok figyelmen kívül maradnak, *k* értéke tehát csökken). Emellett szükség van a leginkább "kívánatos" osztályszámra is. Ha ezt jelentősen meghaladjuk az iterációk során, akkor az algoritmus megpróbálja a közel eső osztályokat összevonni, ha pedig nagyon alatta maradunk, akkor a leginkább "heterogén" osztályok felbontásával közelítünk a megkívánt értékhez. Az összevonás illetve a kettébontás küszöbértékeit ugyancsak a felhasználó szabja meg (minimális szeparálódás illetve maximális osztályon belüli eltérésnégyzet formájában). Az ISODATA algoritmusa a sok paraméter együttes alkalmazása miatt eléggé bonyolult, s itt nem részletezhetjük (lásd pl. Therrien 1989, pp. 219-222).

4.1.2 Egy általános, index-független particionáló módszer

A *k*-közép módszer, mint láttuk, csak korlátozottan alkalmazható (súlyos feltétel az adatok átlagolhatósága) és ráadásul – az osztályok belső eltérésnégyzet-összegének mérésével – csak a kohéziót veszi figyelembe közvetlenül. Ha a *J* függvényt az alábbiak szerint definiáljuk, mindkét problémán segíthetünk, és egy jóval általánosabban alkalmazható egyszerű particionáló eljárást kapunk. Legyen AVG_b az osztályokon belül kiszámított összes különbözőség

átlaga, AVG_e pedig azon objektumpárok között kifejezhető különbözBUBUségek átlaga, amelyek nem tartoznak egy osztályba. A 3.111 képlet adta meg a belső távolságok átlagát egy osztályra, ezt kiterjesztve k osztályra kapjuk az alábbi formulát:

$$AVG_b = \sum_{i=1}^{k} \sum_{g \in A_i} \sum_{h \in A_i} DIS_{gh} / \sum_{i=1}^{k} m_i(m_i - 1)/2 \tag{4.2}$$

míg az osztályok közötti különbözőségek átlaga egyenlet formájában még "riasztóbb":

$$AVG_e = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{g \in A_i} \sum_{h \in A_i} DIS_{gh} / \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} m_i m_k \tag{4.3}$$

AVG_b tehát a kohézió, AVG_e pedig a szegregáció mérőszáma, a DIS különbözőség pedig a 3. fejezetben bemutatott függvények *bármelyike* lehet, mint pl. a kevert adattípusokra kidolgozott Gower-index. Egy adott partíció "jóságát" mérő J függvényt a kohézió és a szegregáció hányadosaként definiáljuk (ez esetben G-vel jelölve):

$$G = \frac{AVG_b}{AVG_e} \tag{4.4}$$

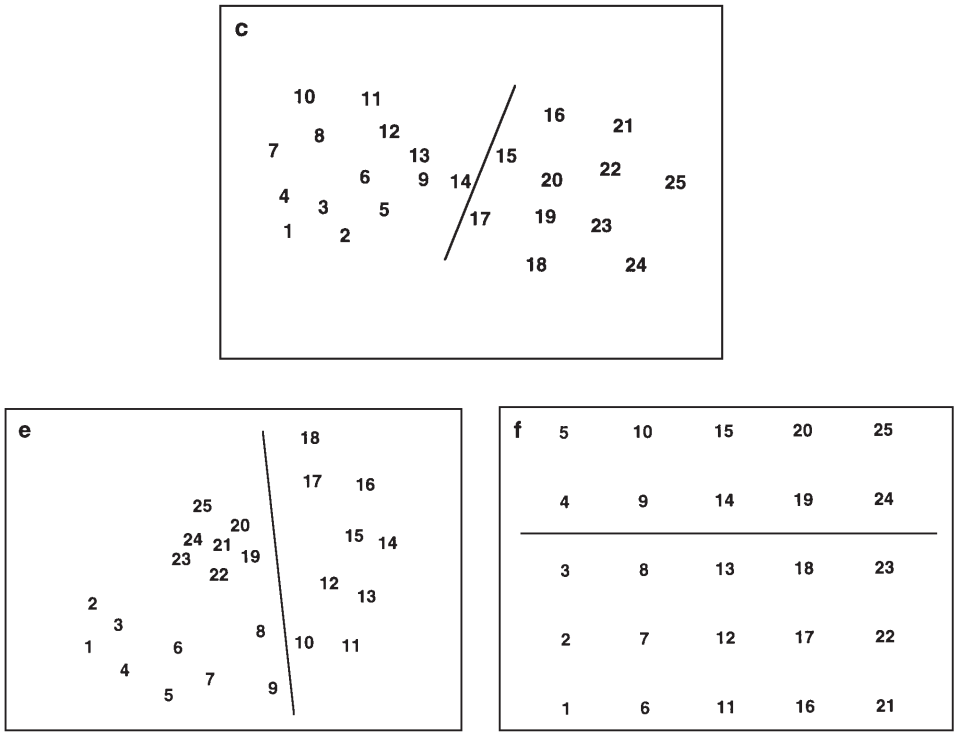
azaz minél nagyobbak a "külső" különbözőségek a "belsőkhöz" képest, annál jobb a felosztás[4]. Egy teljesen véletlenszerű osztályozásnál a G értéke 1 körüli (1-nél nagyobb érték annak a nyilvánvalóan "extra-rossz" esetnek felel meg, amikor a belső különbözőségek átlaga meghaladja a külsőkét). A belső értékek csökkenésével és a külsők növekedésével párhuzamosan G határértékben a 0-hoz tart. Elmondható, hogy G az osztályozás jóságának egy általános, *a különbözőség típusától független* mérőszáma. G előnye, hogy a különféle koefficiensek alapján kapott osztályozások jósága közvetlenül összemérhető egymással, hiszen G teljesen érzéketlen pl. az értéktartományra.

A módszer algoritmusa csak az alkalmazott jósági kritériumban tér el a k-közép módszertől: minden lépésben azt az objektumot helyezzük át, amely G maximális csökkenését idézi elő. A kezdeti osztályozásra azokat a lehetőségeket vehetjük csupán figyelembe, amikor nincs szükség a súlypontok meghatározására.

A 4.4 ábra mutatja a módszer eredményességét a példák esetében, az euklidészi távolság alkalmazása mellett (az euklidészi távolság itt nem lett volna "kötelező", azonban csak így van értelme az összehasonlításnak a k-közép módszerrel). Az **a, b** és **d** esetekben az osztályozás azonos a k-közép módszerrel kapott eredménnyel, így ezeket nem mutatjuk be újra. A **c** esetben egy eltérés jelentkezik: a 14. objektum a baloldali osztályba került, ellentétben a k-közép osztályozással, mutatva az átmenetet jelentő objektumokkal kapcsolatos besorolási problémákat. Az **e** példában valamivel jobb eredményt kaptunk, mint a k-közép eljárással, mert a középső, kompakt csoport legalább egyben maradt. Az **f** esetben természetesen ezúttal sem jöhetett ki más, mint a pontok egy viszonylag önkényes felosztása.

Amit a k-közép módszernél nem tehettünk meg, arra itt lehetőség nyílik: a G értékek közvetlenül összevethetők s így az osztályozások relatív jósága értékelhetővé válik. A legjobb

4 A 4.4 hányados osztályozások *a posteriori* jóságának eldöntésére régen ismert (vö. pl. Hartigan 1975), az osztályozás folyamán azonban, mint jósági kritériumot Podani (1989a) alkalmazta általánosan, a hierarchikus esetben is (lásd az 5.2.4 részt).



4.4 ábra. Az index-független osztályozás eredményei a példaadatokra. Csak a *k*-közép módszerétől eltérő felosztásokat mutatjuk be.

értéket természetesen a **b** esetben kapjuk (G=0,23), s ehhez képest már nagyon magas az éppen "összeérő" két osztály értéke a **c** esetben (G=0,48). A többi esetre még rosszabb az "osztályozhatóság" értéke, főleg a kohézió csökkenése miatt (**d**-re G=0,52, **e**-re pedig G=0,56). Feltűnő, hogy a random (**a**) esetre kapott érték – legalábbis két tizedesjegyig – megegyezik az **e**-vel (G=0,56). A legkevésbé osztályozható nyilván az **f** példa reguláris ponthalmaza, a maga G=0,64-es értékével.

A "belső" és "külső" távolságok figyelembevétele természetesen megtalálható a matematikailag kifinomultabb eljárásokban is, de ezek alkalmazhatósága megint csak az euklidészi esetre redukálódik. Számos szerző javasolta, hogy az eltérésszorzat-összegek mátrixát bontsuk fel két összetevőre, az osztályok közötti ("between-class", **B**) és az osztályokon belüli ("within-class", **W**) részre. Ekkor a teendő egy olyan partíció előállítása, amely maximalizálja a $\mathbf{W}^{-1}\mathbf{B}$ mátrix legnagyobb sajátértékét (Roy kritérium) vagy pedig nyomát (Hotelling kritérium, lásd Anderberg 1973). Amint Gordon (1981) megjegyzi, ezek a kritériumok hajlamosak lehetnek egyenlő méretű osztályok létrehozására. Megemlíthetnénk még egyéb eljárásokat is, de ezek már igen szigorú feltételeket támasztanak az adatokkal szemben (pl. többváltozós normalitás), amelyek ritkán teljesülnek.

4.1.3 Többszörös particionálás

Az osztályok számának előzetes rögzítése elkerülhető a particionáló módszerek (jelen esetben a *k*-közép eljárás) rekurzív alkalmazásával, amely átmenetet jelent a hierarchikus osztályozás felé (5. fejezet). Az objektumhalmazt először két részre bontjuk, majd egy új osztályközéppont

kiválasztásával három osztályra térünk át, és így haladunk tovább addig, amíg az általunk megadott maximális osztályszámot, k_{max}, el nem érjük (a módszert André [1988] nevezte el többszörös particionálásnak). Az algoritmus a következő:

1. Az objektumokat kezdetben egy osztályként kezeljük, s kiszámítjuk a súlypontot. Megkeressük a súlyponttól legtávolabb eső objektumot, s ezt egy új osztály magpontjának tekintjük. Ekkor tehát k=2.

2. Ez a lépés gyakorlatilag egy teljes k-közép elemzés: minden objektumot áthelyezünk abba az osztályba, amelynek súlypontjához a legközelebb esik. Ekkor új súlypontokat kell kiszámítanunk, s további áthelyezésekre lehet szükség. Az áthelyezéseket és a súlypontok átszámítását abbahagyjuk, ha az osztályok már nem változnak, azaz minden objektum abba az osztályba tartozik, amelynek a súlypontjához a legközelebb van.

3. Megnöveljük eggyel k értékét. Ha ez nem nagyobb, mint k_{max}, akkor megkeressük azt az objektumot, amelyik a saját osztályának a súlypontjától a legtávolabb van, és ezt tekintjük az új osztály magpontjának, majd visszamegyünk a 2. lépéshez. Ha k_{max} értékét meghaladná az osztályok száma, akkor az elemzés leáll.

A fenti algoritmust követve végeztük el a példaesetekre az osztályozást. Kiemelendő: most nem az eltérésnégyzet-összeget minimalizáljuk, s ez különbségek forrása a k-közép módszerrel kapott eredményektől. A **c** eset 14. pontját ugyanis a többszörös particionálás (az "index-független" eljáráshoz hasonlóan, vö. 4.4c ábra) a baloldali osztályba tette. Ha azonban alaposabban megvizsgáljuk az adatokat kiderül, hogy a 14. objektum a jobboldali osztályban is éppen olyan jó helyen van: áthelyezése ugyanis a súlypontot úgy változtatja meg, hogy most ahhoz kerül közelebb. A súlyponttól vett távolságok alapján tehát több egyenrangú megoldás is adódhat. Erre az esély jóval kisebb az eltérésnégyzet minimalizálásakor: a konkrét példában eszerint jobb, ha a 14. objektum a "jobboldali" osztályba kerül (lapozzunk vissza a 4.3c ábrához!). A 14. objektum helyzete tehát nagyon bizonytalan, amelyre a k-közép módszer két változata eltérően reagált.

A többszörös particionálás eredménye hierarchikus osztályozás, ha a k+1 értékre kapott új osztály a k érték melletti valamelyik osztály kettébontásából származik, és ez fennáll k minden általunk figyelembe vett értékére. Ez valósult meg a **b** példa osztályozásában, amikor is a kapott osztályok k különböző értékeire a következő sorozatot adták:

k=2 {1 - 19} {20 - 25}

k=3 {1 - 7} {8 - 19} {20 - 25}

k=4 {1 - 7} {8 - 13} {14 - 19} {20 - 25}

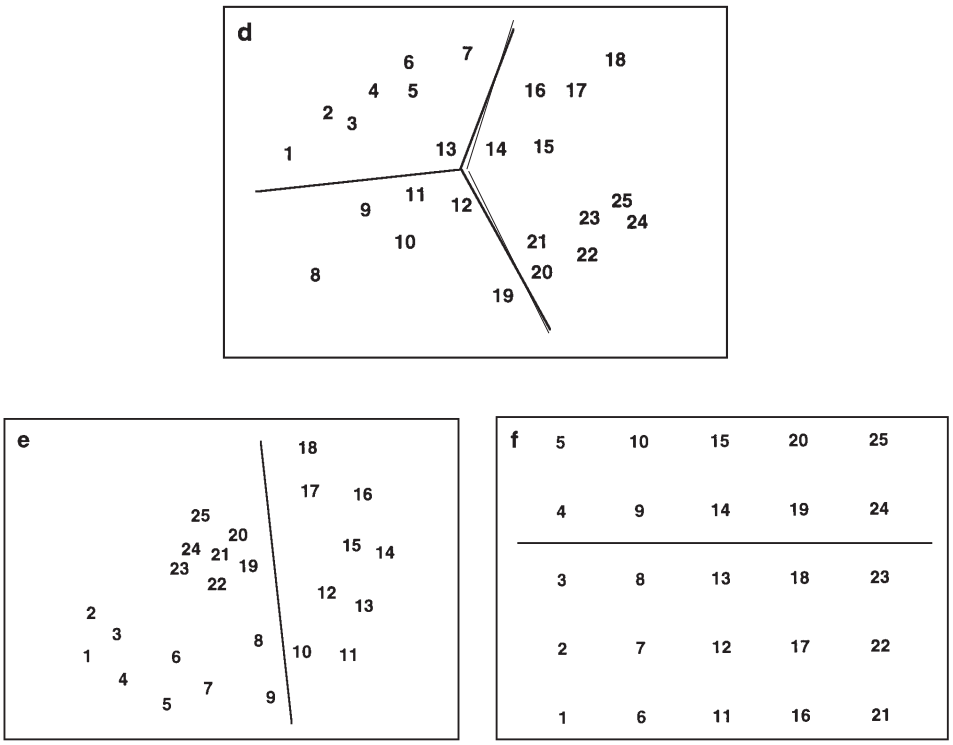
(a k=4 esetben megegyezően a 4.3b ábrával). Ezzel szemben a **d** példára k két különböző értéke mellett már egymásba nem beágyazható osztályokat kaptunk:

k=2 {1 - 11, 13} {12, 14 - 25}

k=3 {1 - 7, 13} {8 - 12, 19} {14 - 18, 20 - 25}

(l. a 4.5**d** ábrát a k=3 esetre, amelynél az analízist befejeztük). Ennek az ellentmondásnak az lehet egy lehetséges értelmezése, hogy az objektumok osztályozhatósága kérdéses k jelen értékei mellett (André 1988), mint ahogy ez valóban így is van a **d** példában: a megnyúlt osztályokat ugyanis e módszerrel nem tudjuk kimutatni.

A többszörös particionálás eredménye teljesen eltér az előzőektől az **e** és az **f** esetekben is.



4.5 ábra. A többszörös particionálás eredménye a példaadatokra. Az egyes lépésekben az áthelyezés a súlypontokhoz való távolság alapján történt, s nem az eltérésnégyzet-összeg minimalizálása volt a cél. Az **a** és **b** esetre az eredmény megegyezik a 4.3 ábrán láthatóval, a **c** esetre pedig a 4.4 ábráéval.

A fenti algoritmus során minden lépésben az osztályok valamelyikét kettéosztottuk (l. a divizív módszereket az 5. fejezetben). Természetesen fordítva is eljárhatunk: az objektumokat elöször k_{max} számú osztályba rendezzük. Miután az optimális osztályozást elértük, azt a két osztályt, amelyek súlypontja a legközelebb esik egymáshoz, összevonjuk. Ezt a k_{max}–1 osztályos felosztást tökéletesítjük az áthelyezésekkel, majd újabb összevonással lépünk tovább (pl. Beale 1969, Wishart 1978). Ezek a módszerek az agglomeratív hierarchikus eljárások felé mutatnak átmenetet.

4.1.4 Nagy objektumhalmazok gyors particionálása

Az előzőekben tárgyalt módszerek számítógépes megvalósításában az objektumhalmaz maximális méretét a rendelkezésünkre álló gyorsmemória szabja meg. Ez például azt jelenti, hogy egy átlagos 640kbyte-os számítógép csak néhány száz objektumot tud elemezni. Előadódhatnak azonban olyan esetek, amikor nemhogy pár száz, hanem több százezer objektumunk van, s ezeket szeretnénk valamilyen módon osztályokba sorolni. Példaként említhetjük a műholdfelvételek alapegységeit, a pixeleket, melyek klasszifikációja a képeken látható mintázatok felismerése és azonosítása szempontjából nélkülözhetetlen. Ekkor, még ha a memóriaprobléma megoldható is különféle mágneses háttértárolók alkalmazásával, a hagyo-

mányos módszerek rendkívül nagy futásidőt igényelnének. Szükség van tehát olyan eljárásokra, amelyek nagy adathalmazok viszonylag gyors osztályozását is lehetővé teszik. A sebesség növelése persze áldozattal jár: igen kicsi esélyünk van arra, hogy a gyors módszerekkel előállított eredmények optimálisak legyenek. Sőt, az eredmény gyakran attól is függ, hogy milyen sorrendben adjuk meg az objektumokat az adatok beolvasásakor. Ugyanakkor viszont a sokszázezres objektumhalmazok néhány száz csoportra egyszerűsödnek, ezután mindegyikből kiválasztható egy-egy objektum mint a csoport képviselője, és az ilymódon redukált adathalmaz már elemezhetővé válik a szabatosabb módszerek segítségével is (és itt most már nemcsak a particionálásra, hanem a későbbi fejezetekben leírt módszerekre is gondolunk, melyeknél a memória és a sebesség még jobban korlátozó tényező lehet).

A gyors particionáló módszerek (az ún. *"quick clustering"* eljárások) egyik alapelve, hogy az adatokat objektumonként olvassuk be mágneslemezről, tehát nem kell tárolni a teljes adattömböt a gyorsmemóriában. Az alaptípus a vezető (*"leader"*) algoritmus (Hartigan 1975), amely mindössze egyetlen egyszer vizsgálja végig az adatmátrixot a következők szerint:

1. Kiválasztunk egy, a problémának leginkább megfelelő távolság vagy különbözőségi függvényt (*DIS*). A 3. fejezetben felsoroltak jelentős része felhasználható erre a célra. Emellett meg kell adnunk a *DIS* egy *T* küszöbértékét is, amely a gyors osztályok méretét (pontosabban "átmérőjét") szabja majd meg az elemzés egyes lépéseiben.

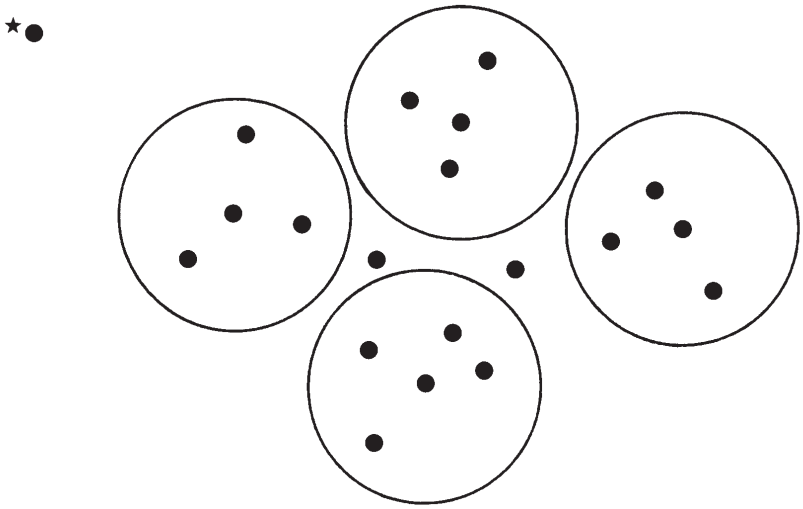
2. Az 1. osztály vezető (kezdő) objektumaként az 1. objektumot választjuk. Jelöljük *j*-vel a többi objektum indexét, azaz $j=2...m$. Az osztályok száma ekkor még $k=1$.

3. Növeljük *j* értékét 1-gyel. Ha $j=m$, az elemzés véget ér.

4. Elkezdjük a már meglevő osztályok vizsgálatát 1-től *k* aktuális értékéig. Amennyiben a *j* objektum távolsága valamely vezető objektumtól kisebb, mint *T*, akkor a *j* objektumot az elsőként adódó ilyen osztályba besoroljuk, s visszatérünk a 3. lépéshez.

5. Ha a *j* objektum minden vezető objektumtól távolabb esett, mint *T*, akkor ezt egy új osztály vezető objektumaként tekintjük, *k* értéke tehát eggyel nő, s visszatérünk a 3. lépéshez.

A módszer kétségtelen előnye a nagy gyorsaság, viszont hátrányos, hogy a végeredmény nagymértékben függ az objektumok sorrendjétől (pl. az 1. objektum mindig vezető). Ez utóbbi hiányosság kiküszöbölhető, ha a vezető objektumokat véletlenszerűen választjuk ki a még nem besorolt objektumok halmazából. Ez viszont a sebesség rovására megy, mert ekkor már többször kell végigfutnunk az adatokon (éppen annyiszor, ahány osztályunk lesz). További hiányosság, hogy az elemzés során először képződő osztályok jóval nagyobbak, mint a későbbiek. Ennek egyik oka az lehet, hogy az először létrejövő (a sok dimenzióban hipergömb alakú) osztályok közötti "üregekben" megrekedhet egy-egy pont, amint azt a 4.6 ábra is szemlélteti két dimenzióra. Megoldásul bevezethető egy második T_2 küszöbérték is (amely valamivel nagyobb *T*-nél), és ennek felhasználásával a kis osztályokba eső objektumok az elemzés egy második fázisában áthelyezhetők a legközelebbi nagy osztályba (COMPCLUS módszer, Gauch 1979, 1980). Ami ezután kis osztály marad, az már jogosabban tekinthető



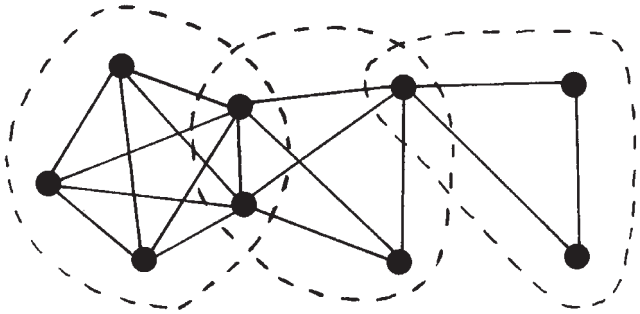
4.6 ábra. A gyors particionálás hátránya, hogy egyes objektumok "beszorulnak" a kezdeti osztályok közötti "üregekbe", s később kialakított kis osztályok magpontjai lesznek. Egy igazi outlier, azaz a többi objektumtól nagyon eltérő objektum a bal felső sarokban található *-gal jelölve.

osztályokba nehezen besorolható, ún. *outlier* (kilógó) egyednek (mint pl. a *- gal jelölt pont a 4.6 ábrán). *T* értékének megválasztásában is ügyesnek kell lennünk. Ha *T*-t túl kicsinynek választjuk, akkor nagyon sok osztályt kaphatunk és az eredmény használhatatlan lesz. Túl nagy *T*-re viszont akár egyetlen egy osztály is elBUBUadódhat. Nyilvánvaló tehát, hogy több elemzést célszerű lefuttatnunk *T* (és a COMPCLUS esetében T_2) különböző értékei mellett, s ezután választható ki a számunkra leginkább megfelelő osztályszám.

A CLUSLA módszer (Louppen & van der Maarel 1979) kombinálja a gyors elemzés fenti módszerét az iteratív áthelyezésekkel, s azokat az objektumokat, amelyek egy másik vezető objektumhoz közelebb vannak, áthelyezik. A vezető algoritmus és a többszörös particionálás között átmenetet jelentő stratégia is alkalmas lehet a gyors osztályozásra (Hartigan 1975). Ebben a többszörös particionálás algoritmusa módosul úgy, hogy az egyes lépésekben nem történik áthelyezés. Az első vezető objektum lehet pl. az összes adat súlypontjához legközelebb eső pont, a második pedig az ettől legtávolabb lévő objektum. Az összes többit egyszerűen ahhoz az objektumhoz soroljuk, amelyikhez a legközelebb esik. A következő lépésben kikeressük azt az objektumot, amelyik a saját vezetőjétől a legtávolabb van, s ez lesz a harmadik osztály vezetője, és így tovább, *k* tetszés szerinti értékéig.

4.2 Átfedéses osztályozások

A 4.3c ill. a 4.4c ábrák egy olyan esetet illusztrálnak, amikor az osztályba tartozás nem nyilvánvaló: a 14. objektum akár az egyik akár a másik osztályba is kerülhet. Mint láttuk, a *k*-közép módszernek a súlypont közelségét figyelembe vevő változata egyformán jónak is találja mindkét megoldást. Felmerülhet a lehetőség, hogy ilyen bizonytalan esetekben "szabaduljunk meg" a hagyományos particionáló módszerek kötöttségétől, az osztályok közötti szükségszerű diszjunkciótól, és mondjuk ki: tartozzon a 14. objektum egyidejűleg mindkét osztályba! Ezzel egy ún. *átfedéses* ("overlapping") klasszifikációt hozunk létre. Az ilyen típusú osztályozásokat



4.7 ábra. A Jardine-Sibson féle B_k osztályozás ábrázolása gráf segítségével. A három teljes részgráf egy-egy osztálynak felel meg, közülük kettő átfed a k=3 szinten, azaz maximum két objektumban.

Jardine & Sibson (1968) javasolta először "B_k clustering" néven, éppen az átmeneti jellegű objektumok miatt nehezen osztályozható halmazok adatszerkezetének valószerűbb jellemzésére. A definíció szerint egy objektumhalmazra osztályozások egész sorozata adható meg k=1, 2, 3 stb. értékeire, amelyben bármely két osztály legfeljebb, de nem feltétlenül, k–1 objektumban fedhet át egymással. A hagyományos partíciók tehát B_1 osztályozások, míg a fenti példa (a két osztályba sorolt 14. objektummal) egy B_2 klasszifikációt reprezentál. (Ez a k nem tévesztendő össze a k-közép módszer osztályszámával; úgy látszik nem volt elég betű az abc-ben, mert a szakirodalom mindmáig ragaszkodik a k-hoz mindkét esetben).

A B_k módszer algoritmusa az eddigieknél kissé komplikáltabb (lásd pl. Ling 1972, Rohlf 1975b) s így csak a főbb alapelveket közöljük. Az objektumokat egy gráf szögpontjaiként kell elképzelnünk, melyben minden szögpont-párt él köt össze, ha a megfelelő két objektum hasonlósága egy T küszöbértéknél nagyobb. Ezután ún. *maximális teljes részgráfokat* kell keresnünk, amelyek a lehető legtöbb pontot tartalmazó olyan részgráfok, ahol minden párosításban van él. Ezen részgráfok közül azok lesznek az átfedő osztályok, amelyek legfeljebb k–1 pontban metszik egymást (k–1 pontban közösek). Egy ilyen esetet mutat be k=3-ra a 4.7 ábra. A keresést természetesen tovább folytathatjuk T csökkenő értékeire, és ekkor átfedéses hierarchikus osztályozáshoz jutunk (vö. a következő fejezettel). Ugyancsak változtatható k értéke is, tehát a kutatónak elég sok mindent át kell tekintenie egyidejűleg, hogy a B_k módszer eredményét megfelelően értékelhesse. Az eredmények ábrázolása is nehézkes, s emiatt sokan nem ajánlják ezt az eljárást. A B_k módszer helyett a következő részben tárgyalt, viszonylag újabb keletű módszert, a fuzzy osztályozást javasolhatjuk.

4.3 "Lágy" (fuzzy) osztályozások

Gyakran találkozhatunk olyan osztályozási problémákkal, amikor bizonyos objektumok nem sorolhatók be egyértelműen egyik osztályba sem. Ezt illusztrálta a 4.3c ábra is, és ezt a problémát próbáltuk áthidalni az átfedéses klasszifikációk segítségével az előző részben. Mint már említettük, sok osztályra és nagyszámú objektumra az átfedéses osztályozások kevéssé alkalmasak, és az eredmények sem ábrázolhatók más eljárások, például ordinációk beiktatása nélkül. Fontos volt tehát az a felismerés, hogy problematikus osztályozások nem írhatók le egyértelműen a korábbi, diszkrét módszerek alkalmazásával. Könnyebben interpretálható, a valós viszonyokat jobban tükröző eredményeket kaphatunk, ha az osztályba tartozás fogalmát kicsit "fellazítjuk". Mindehhez Zadeh (1965) "forradalmian" új elképzelése a lágy (="fuzzy")

halmazokról adta a kiindulást. A klasszikus halmazelmélettel szemben itt megengedjük, hogy egy objektum több részhalmazba is tartozzon úgy, hogy a hovatartozás mértéke különböző is lehet. Fuzzy osztályozások esetén az osztályba tartozás erBUBUsségét súlyokkal fejezzük ki azzal a kikötéssel, hogy egy objektumra nézve a súlyértékek összege 1-et kell adjon. (Ez a feltétel a valószínűségeket juttathatja rögtön eszünkbe, hiszen egy teljes eseményrendszerre a valószínűségek összege is 1. Az analógia azonban nagyon távoli, hiszen a súlyértékek nem az osztályba tartozás valószínűségét jelentik, hanem az objektumok osztályokhoz való affinitását, "vonzódását" fejezik majd ki.) Az osztályozás tehát egy mátrixszal írható le, melynek sorai az objektumok, oszlopai az osztályok, s az egyes értékek a súlyok:

$\mathbf{U} = \{ u_{jc} \}, j=1,...,m,\ c=1,..,k,$ és

$$\sum_{c=1}^{k} u_{jc} = 1 \text{ minden } j\text{-re} \tag{4.5}$$

(az osztályok számát, k-t, előre kell megadnunk, csakúgy, mint a k-közép módszernél). A kérdés "csupán" az, hogy miképpen állítható elő egy ilyen táblázat?

A legegyszerűbb és legáltalánosabban ismert fuzzy osztályozó módszer a c-közép (vagy fuzzy k-közép) eljárás (Bezdek 1981, 1987, Marsili-Libelli 1989). Ennek során az úgynevezett *fuzzy eltérésnégyzet-összeget* kell minimalizálni:

$$FSSQ = \sum_{j=1}^{m}\sum_{c=1}^{k} u_{jc}^{f} d_{jc}^{2} , \tag{4.6}$$

ahol

$$d_{jc}^{2} = \sum_{i=1}^{n} (x_{ij} - v_{ic})^{2} \tag{4.7}$$

a j objektum és a *c osztály súlypontja közötti távolság, és* f (>1) a lágysági paraméter. Minél nagyobb f értéke, annál lágyabb a kapott partíció, azaz annál elmosódottabb lehet az osztályok közötti határvonal. A fuzzy osztályozásnál tehát nemcsak az osztályok számát kell előre megadnunk, hanem f-et is. Ez egyrészt újabb önkényes döntést igényel, másfelől viszont lehetőséget ad arra, hogy a paraméterek változtatásával adatainkat alaposabban elemezhessük.

Az osztályok súlypontjait a következőképpen határozzuk meg:

$$v_{ic} = \frac{\sum_{j=1}^{m} u_{jc}^{f} x_{ij}}{\sum_{j=1}^{m} u_{jc}^{f}} \tag{4.8}$$

Az osztályozás főbb algoritmikus lépései:

1. A kezdő osztályozást az egymástól legtávolabb eső k kezdőpont kiválasztásával adjuk meg. Emellett természetesen más, a 4.1.1 részben ismertetett kiindulás is elképzelhető.

2. A kiindulási súlyértékeket minden *j* objektumra úgy határozzuk meg, hogy azok a súlypontoktól vett távolságaikkal arányosak a (4.5) feltétel teljesülése mellett.

3. Az új súlyértékek meghatározása a következő egyenlet alapján történik:

$$u'_{jc} = \frac{1}{\sum_{h=1}^{k} \left(\frac{d_{jc}}{d_{jh}} \right)^{2/(f-1)}} \tag{4.9}$$

Amennyiben $d_{jc} = 0$, vagyis a *c* osztály súlypontja egybeesik a *j* objektummal, akkor $u_{jc} = 1$ míg az összes többi súlyérték 0 lesz.

4. Kiszámítjuk az új súlypontokat a 4.8 egyenlet segítségével.

5. Az elemzés leáll, ha a mostani, *q*-adik ciklusban kapott új értékek és az előző, *q*–1-edik ciklusban kapott súlyok közötti eltérés nem lépi túl az előre megadott ε küszöböt:

$$\varepsilon = \max_j \max_c |u_{jc}^{(q)} - u_{jc}^{(q-1)}| \tag{4.10}$$

A leállítás kritériuma tehát a két iteráció közötti maximális változáson alapszik. Ha ε túllépi a küszöbértéket, akkor visszatérünk a 3. lépéshez. Egyéb esetben a legutoljára kapott súlyértékek jelentik az osztályozás végeredményét.

A módszert illusztrálandó megvizsgáltuk a 4.3c ábrán látható ponthalmazt a következő kiindulási paraméterekkel: $k=2$, $f=1{,}5$ és $\varepsilon = 0{,}01$. Ezt az ε küszöbértéket már a 4. iterációs lépés után elértük. Az objektumok jelentős része erősen "vonzódik" valamelyik osztályhoz, amint azt a 0,9-nél nagyobb súlyok jelentős száma mutatja (4.1 táblázat). A sok problémát okozó 14. objektum két súlyértéke azonban csaknem azonos (vastagon szedve a táblázatban), jól mutatva a két osztály közötti átmeneti helyzetet.

A fuzzy osztályozások értékelésében rendszerint nem elegendő a súlyértékek egyszerű megvizsgálása. Több lehetőségünk is van például arra, hogy az osztályok "optimális" számát meghatározzuk. Elsőként említendő meg a Bezdek (1974, 1981) -féle *partíciós koefficiens*

$$F_k = \sum_{j=1}^{m} \sum_{c=1}^{k} u_{jc}^2 / m \tag{4.11}$$

amely 1/*k*-tól 1-ig terjed. *k* különböző értékeire a függvény relatív maximumértékeket ér el ott, ahol *k* az optimális osztályszámmal megegyező. A F_c értelmezési tartománya azonban *k*-tól függ, s ezen úgy segíthetünk, ha azt a [0,1] intervallumra kiterjesztjük az alábbiak szerint:

$$F_k = \frac{kF_k - 1}{k - 1} \tag{4.12}$$

A partíció hatákonyságát Dunn szerint az *entrópával* is mérhetjük:

$$H = -\frac{1}{m} \sum_{j=1}^{m} \sum_{c=1}^{k} u_{jc} \log u_{jc} \tag{4.13}$$

4.1 táblázat. A fuzzy osztályozás eredménye a 4.3c ábra pontjaira *k=2 és f*=1.5 mellett.

Objektum	1. osztály	2. osztály
1	.9839	.0161
2	.9819	.0181
3	.9973	.0027
4	.9948	.0052
5	.9901	.0099
6	.9556	.0444
7	.9940	.0060
8	.9979	.0021
9	.9536	.0464
10	.9810	.0190
11	.9723	.0277
12	.9915	.0085
13	.9676	.0324
14	**.5050**	**.4950**
15	.0804	.9196
16	.0547	.9453
17	.1951	.8049
18	.0460	.9540
19	.0023	.9977
20	.0003	.9997
21	.0173	.9827
22	.0012	.9988
23	.0018	.9982
24	.0190	.9810
25	.0104	.9896

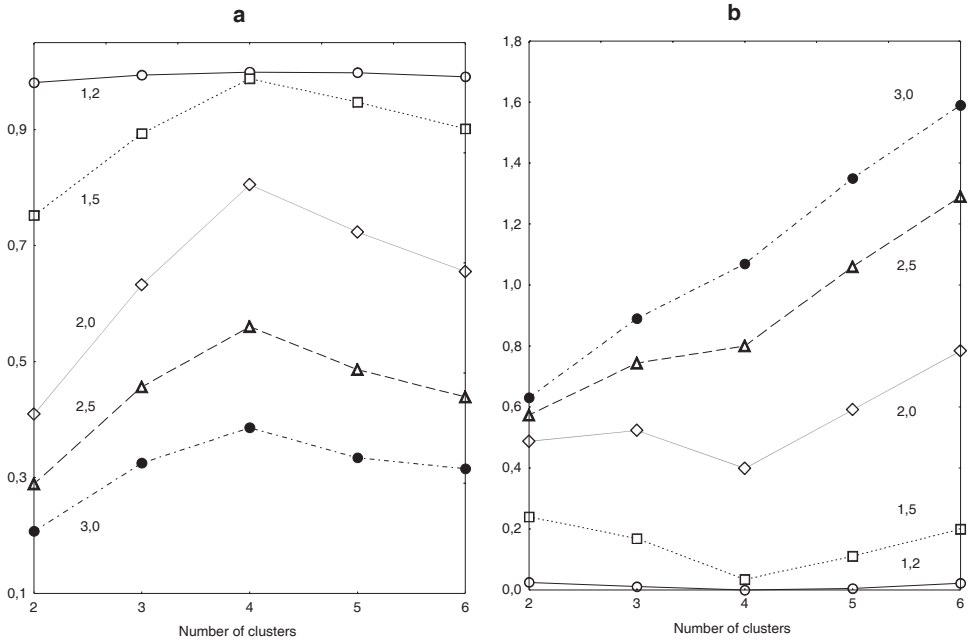
Ennek standard alakja a következő:

$$H' = \frac{H}{1 - k/m} \ . \qquad (4.14)$$

k különféle értékeit végigpróbálva a 4.14 függvény minimuma kikereshető, ezzel elősegítve az optimális osztályszám megállapítását.

A 4.3b ábra nyilvánvalóan 4 osztályt "rejtő" példájára a *k*=2, 3, 4, 5 és 6 értékeket választva, illetve az *f* értékét is fokozatosan növelve (*f*=1,2; 1,5; 2,0; 2,5; és 3,0) meghatároztuk a fuzzy osztályozásokat. Az osztályszám és a partíciós koefficiens illetve a partíciós entrópia közötti összefüggést, *f* különböző értékei mellett, a 4.8 ábra két diagramja ábrázolja. Mint várható is volt, a partíciós koefficiens a maximumot a *k*=4 esetben éri el függetlenül *f* értékétől (bár a maximum kevésbé kifejezett az *f*=1,2 esetben). Ezzel szemben a partíciós entrópia minimum helyét már *f* is befolyásolja: az erősen fuzzy osztályozásoknál (*f*>2) a *k*=2 esetre adja a minimumot, s a várt eredmény csak a kevésbé fuzzy osztályozásokra adódik. E példa alapján tehát a partíciós koefficiens tekinthető az optimális osztályszám előnyösebb indikátorának.

Az ún. *szeparálódási együttható* összefüggésben van a partíciós koefficienssel:



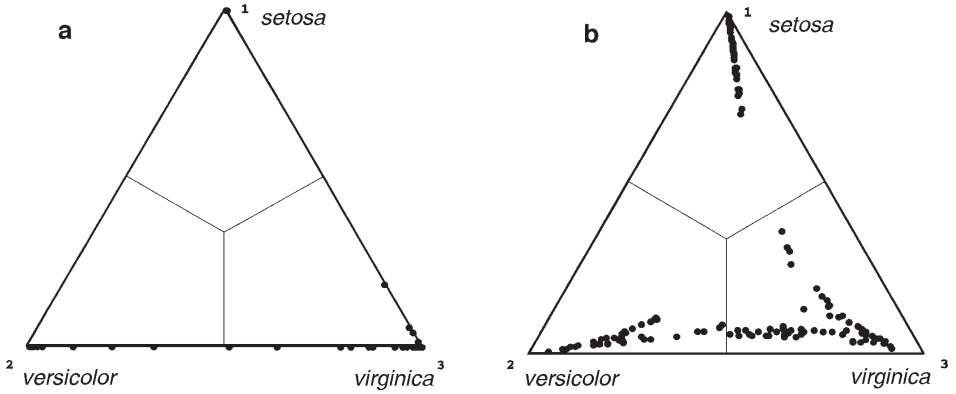
4.8 ábra. A (4.12) partíciós koefficiens (**a**) és a (4.14) partíciós entrópia (**b**) változása az osztályszám függvényében, *f* különböző értékei mellett a 4.3b ábra pontjaira alkalmazott fuzzy osztályozásokban.

$$\Omega = \sum_{j=1}^{m}\sum_{c=1}^{k} u_{jc}^2 \tag{4.15}$$

Ennek értéke *m/k* és *m* közé esik. Minél közelebb van az *m*-hez, annál "keményebb" a felosztás, azaz annál inkább közelítik a súlyok az 1-es értéket. Szélső esetben minden súly akár 1 is lehet, azaz a hagyományos "kemény" partíció voltaképpen a fuzzy osztályozás egy speciális esetének tekinthető. A *b* és *c* osztályok közötti páronkénti elválás az osztályok súlypontjai közötti távolságok felhasználásával fejezhető ki:

$$\delta_{bc} = \frac{\sum_{i=1}^{n}(v_{ib} - v_{ic})^2}{\max(u_{jb}d_{jb}) + \max_{j}(u_{jc}d_{jc})} \tag{4.16}$$

A fuzzy osztályozás táblázatos eredménye grafikus formában is kifejezhető. Ehhez egy olyan koordináta-rendszert kell alkalmaznunk, melynek tengelyei az egyes osztályoknak, a koordináták pedig az objektumok súlyértékeinek felelnek meg. Miután egy objektumra nézve a koordináták összege 1, a koordináta rendszerben a pontok egy hipersíkon helyezkednek el, hasonlóan az összeggel történő standardizáláshoz. (A 2.9c ábrán ui. az "átlóra" rajzolt telt körök fuzzy osztályozásnak is megfelelhetnek a *k*=2 esetre. Ugyancsak az átlón helyezkednek



4.9 ábra. Az A2 táblázatban szereplő három *Iris* faj lágy osztályozása a lágysági koefficiens két különböző értékére, **a:** *f*=1,25; **b**: *f*=2,5.

el a 4.1 táblázat fuzzy osztályozásában szereplő pontok is, a legtöbben az átló valamelyik végénél, míg a 14. pont az átló felénél, ezt azonban – úgy érezzük – felesleges lenne külön ábrán bemutatni.) A papír síkjában persze csak két osztály ábrázolható egyidejűleg a sokból, ennek ellenére a fuzzy osztályozások ilyen – tulajdonképpen *ordinációs* (vö. 7. fejezet) – ábrázolása megkönnyíti az eredmények interpretációját. Itt azonban máris javítani kell magunkat, mert ha az osztályok száma éppen három, akkor a pontok egy egyenlBUBU oldalú háromszögön helyezkednek majd el és ez két dimenzióba áttéve kiválóan ábrázolható. A háromszög csúcsai megfelelnek az egyes osztályoknak, s minél közelebb van egy pont valamely csúcshoz, annál egyértelműbb a hovatartozása. Ha történetesen mindhárom súlyérték 0,33, akkor a pont a háromszög súlypontjába kerül, jól mutatva az objektum maximálisan "bizonytalan" helyzetét. Ha két súlyérték 0,5, a harmadik pedig 0, akkor a pont a háromszög megfelelő szárának felezőjére esik majd.

Ezt a háromszögdiagramos ábrázolást az *Iris* adatok (A2 táblázat) felhasználásával mutatjuk be a 4.9 ábrán, a lágysági együttható két különböző értékére, nyers adatokat elemezve. A fuzzy osztályozást eleve három csoportra hajtjuk végre, hiszen kiindulásképpen is három fajunk volt. Mint az ábra is mutatja, alacsony *f* értékre (*f*=1,25) a három faj elválása eléggé egyértelmű (igen sok pont egybeesik), bár az *Iris versicolor* és *virginica* között egy átmeneti sor is jelentkezik (4.9a ábra). Ha a koefficiens értékét nagyobbnak választjuk (*f*=2,5), a fajok közötti átmenet folyamatosabbá válik, és a *setosa* és a *virginica* között is "megindul" valami. A 4.9b ábra voltaképpen úgy értelmezhető, hogy a *virginica* egyedek egy része inkább a *versicolor*, másik része pedig inkább a *setosa* felé "húz". A háromszögdiagramos ábrázolás voltaképpen minden olyan esetben használható, amikor objektumainkat 3 változóval írjuk le, s ezek értékeinek összege minden objektumra 1 (azaz előzőleg összeggel való standardizálást hajtottunk végre).

4.4 Irodalmi áttekintés

A partíciós módszerek klasszikusnak tekinthető leírásait és alapos jellemzését Anderberg (1973) és Hartigan (1975) műveiben találhatjuk meg. Különösen tág teret szentel e módszereknek Späth (1980) példákkal bőven illusztrált könyve. Everitt (1980) is részletesen tárgyalja a

4.2 táblázat. Nem-hierarchikus osztályozási opciók egyes programcsomagokban.

	BMDP 7	Statistica	SYN-TAX
k-közép módszercsalád	+	+	+
index-független módszer			+
többszörös particionálás			+
gyors particionálás			+
fuzzy osztályozás			+

particionáló munkákat, s külön érdeme, hogy kitér a megoldatlan problémákra is (könyvének újabb kiadása: 1993). Azonban nem minden osztályozásról szóló kézikönyv ilyen részletes, mert a fő hangsúly többnyire a hierarchikus módszereken van (pl. Clifford & Stephenson 1975, Gordon 1981). A biológiai alkalmazásokat áttekintve megállapíthatjuk, hogy a partíciók leginkább az ökológia/cönológia területén jönnek számításba (pl. Orlóci 1978, André 1988, Jancey 1974). Gauch (1982) a nem-hierarchikus osztályozás elsődleges szerepét a nagy objektumhalmazok gyors osztályozásában látja, és ennek megfelelően kezeli is a témát, jó néhány irodalmi hivatkozással segítve a további elmélyedésre vágyókat. Magyar nyelvű kézikönyvként Füstös & Kovács (1989) forgatható haszonnal. A fuzzy osztályozásról a legjobb összefoglalót Bezdek (1981, 1987) munkái adják, s ajánlható még Equihua (1990) és Marsili-Libelli (1989) cikke is. A nem-hierarchikus klasszifikáció és a mintázatfelismerés közötti kapcsolatról sok mindent megtudhatunk Therrien (1989) könyvéből.

4.4.1 Számítógépes programok

Különféle nem-hierarchikus osztályozási módszerek programlistáit számos könyvben fellelhetjük, különösen a 10 évnél régebbi kiadásúakban (pl. Hartigan 1975, Anderberg 1973, Orlóci 1978, Späth 1980, ill. a **COMPCLUS** listája, Gauch 1979). Újabban már nem "divat" a programlisták közlése, hiszen a kutatók a könnyen alkalmazható, "felhasználóbarát" programokat keresik, melyeknél az osztályozást ténylegesen kiszámító programrészlet méreteiben szinte jelentéktelen a "kiszolgáló" rutinokhoz képest. A jelen fejezetben említett osztályozási eljárások "előfordulási helyeit" a 4.2 táblázatban foglaltuk össze.

4.5 Kérdezz - válaszolok

K: *Egyértelműnek tűnik számomra, hogy az általad említett módszerek kivétel nélkül "hipergömb alakú" osztályokat képesek csak kimutatni, a megnyúlt pontfelhőket nem érzékelik. Tudsz-e olyan módszert, ami mondjuk a 4.3d-e ábrák megnyúlt, ill. ívelt pontfelhőit is kimutatná, hiszen ezek is első látásra "létező", jól elkülönülő osztályoknak tűnnek?*

V: A kérdésed teljesen jogos, hiszen a bemutatott példáknál megelégedtünk azzal, hogy láttassuk: az egyes módszerek bizonyos esetekben miként, azaz nem mindig a várt módon "viselkednek". Természetesen van olyan eljárás, amely kimutatja akár a virsli vagy sarló alakú osztályokat is, de erre majd a következő, a hierarchikus módszereket tárgyaló fejezetben kerül sor. Mindenesetre megemlítem, hogy az *egyszerű lánc* módszerről van szó, melynek alapelve egyébként bizonyos komplex particionáló algoritmusokban is szerepel (pl. Orlóci TRGRPS

módszere, 1976b, 1978). A hierarchikus osztályozásokból könnyedén előállíthatunk partíciókat, de erről is majd később.

K: *A másik fő gondom az, hogy valóban csak iterációs, próbálkozásos módon tudunk particionálni? Nincs egy olyan, egyértelmű algoritmus ami mindeképpen előállítja az optimális eredményt?*

V: Igen, az osztályozási problémák jelentős része olyan, hogy nagyon nehéz – vagy lehetetlen – optimalizációs számításmenetet megadni, ami minden esetben egyértelmű megoldást ad és egyben hatékony is. Ez azt jelenti, hogy ha mindenképpen az abszolút optimumot akarjuk, akkor az összes lehetőséget végig kell vizsgálnunk. Kivételes esetek is vannak, pl a *"branch and bound"* algoritmus (Grötschel & Wakabayashi 1990) az eltérésnégyzet-összeg minimalizálására pár tucatnyi objektumra egyértelmű optimumot talál, de ez is igen számításigényes és nagyobb mennyiségű adatra használhatatlan.

K: *Van-e egyáltalán olyan módszer ami mindig egy eredményre vezet? Fontos-e az a szempont, hogy a módszer végeredménye egy s csak egy legyen?*

V: Matematikusok szemszögéből nézve feltétlenül. Más a helyzet persze a biológiában, ahol kérdéseinkre kielégítő választ kaphatunk az ún. *heurisztikus*, azaz módszeresen keresgélő, bár nem feltétlenül az abszolút optimumot adó eljárásokkal is. Gauch (1982) könyve, amely a legkevésbé sem vádolható meg azzal, hogy túlterheli az olvasót a matematikai részletekkel, meg is indokolja ezt. Érvei közül mindenképpen megfontolandó a következő: a biológiai adatgyűjtés és feldolgozás minden lépése annyira telített a szubjektív elemekkel, hogy önbecsapás lenne egy ilyen módszerre való törekvés. Ha választhatunk, persze, a matematikailag is jobban definiált módszert részesítsük mindenképpen előnyben.

K: *Tulajdonképpen hányféleképpen sorolhatunk be m objektumot k osztályba?*

V: A lehetőségek számát az elsőfajú Stirling-formula adja meg, miszerint:

$$S = \frac{1}{k!}\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i}i^m \qquad (4.17)$$

Könnyen meggyőződhetsz arról, hogy 20 objektumot (ami igazán nem sok) 2 osztályba éppen 524287-féleképpen rendezhetünk el! (A képlet egyébként a k=2 esetre a következő egyszerűbb alakot ölti: $S=2^m/2-1$; gondolj elemi kombinatorikai ismereteidre!)

K: *Még egy dolog furdallja nagyon az oldalam: többnyire meg kellett adnunk a keresett osztályok számát is. Ez eléggé önkényesnek látszik, de legalábbis kényelmetlennek, hiszen sokat kell "játszanunk" k-val, amíg végre "értelmesnek" látszó felosztást kapunk.*

V: Engedd meg, hogy erre a kérdésre egy kicsit részletesebben válaszoljak, hiszen az adatokban rejlő osztályok száma a klasszifikáció egy központi kérdése. Nem is fogok itt mindenre kitérni, hiszen a későbbi fejezetekben bőven lesz még utalás erre a problémakörre.

A most ismertetett módszerek valóban olyanok, hogy sok mindent végig kell velük próbálnunk az adatstruktúra teljes feltárásához. Ez azonban valójában nem is olyan nagy feladat, hiszen a mai számítógépek már kellően nagy kapacitásúak és megfelelő sebességűek ehhez a – Te szavaddal élve – "játszadozáshoz". El kell ismernünk azonban, hogy a nem-hierarchikus osztályozás eme módszerei önmagukban kevéssé állják meg a helyüket az adatfeldolgozó

módszerek nagy családjában, s velük párhuzamosan célszerű más típusú módszereket is alkalmazni (a hierarchikus osztályozásra és az ordinációra gondolok). Az ordinációk révén például a sokdimenziós térben elhelyezkedő pontfelhő "láthatóvá válik" (hogy miként, azt majd később), s ennek összevetése a partíciókkal már sokatmondó lehet. Egy hierarchikus osztályozás pedig partíciók sorozataként fogható fel, s igen sok olyan módszer van, amely e sorozatban próbál optimumot keresni (lásd az 5.5.3 részt).

De, hogy ne maradj teljesen csalódott, meg kell mondanom: bizonyos újabb fejlemények már sejtetik, lesz a particionáló módszereken belül is megoldás. Téged mint biológust talán külön is érdekelni fog az úgynevezett "genetikai algoritmusok" (Holland 1975, Goldberg 1989) témaköre. (Jobb volna talán az *"evolúciós algoritmus"* elnevezés, mint majd látni fogod.) Arról van szó, hogy a lehetséges végeredményekből szimulációval előállítunk egy "populációt", megadunk egy "fitness" függvényt, ami a "populáció" egyedeinek az életrevalóságát (osztályozás esetében a jóságát) méri, és valamilyen trükkel lehetőséget nyújtunk arra, hogy a populáció egyedei megváltozhassanak (azaz a *mutáció* is lehetséges). Azon egyedeket, amelyek a fitness növelésének irányába mutálnak megtartjuk és "szaporodni" engedjük, a hátrányosan módosuló egyedeket pedig kiszelektáljuk. Az evolúció mechanizmusait bizonyos ideig szabadon működtetjük, majd megvizsgáljuk, hogy melyek a populáció legfittebb egyedei. Ezek között, ha az evolúció sokáig futott, nagy eséllyel találunk maximális fitnessű egyedeket is, amelyek már semmiféle módosítással nem javíthatók tovább (itt a fő különbség a valódi, biológiai evolúcióval szemben, ahol elvben nem zárul le sohasem a "fejlődés"). Partíciók ilyen evolúciós alakítgatásához szükség van egy új definícióra, ami a *k*-közép módszerrel ellentétben (ahol a középértékek többnyire nem létező objektumokat, csak átlagokat takarnak) a *k* osztályt egy-egy objektummal reprezentálja s a többi objektum az ezektől vett távolságok szerint osztályozódik (*k-medoid* módszer, Lucasius et al. 1993). A populáció minden egyes egyede ekkor egy "kromoszómával" jellemezhető, amely *m* darab 1-es és 0-ás számérték füzére. A "kromoszóma" *i*-edik pozíciójában szereplő 1 azt jelenti, hogy az illető objektum egy medoid, a 0 pedig azt, hogy az objektumot a hozzá legközelebb eső medoidhoz kell sorolnunk. A kromoszóma tehát leír egy osztályozást, melynek jósága sokféleképpen mérhető (Moraczewski et al. 1995 javaslata szerint pl. a nem-metrikus többdimenziós skálázásban alkalmazott stressz függvényt, 7.66, célszerű figyelembe vennünk). A kromoszómán pontmutációkat, sőt két kromoszóma között átkereszteződéseket is végrehajthatunk, majd az új egyedeket megfelelő módon kiszelektáljuk. Ezek a módszerek még csak kísérleti stádiumban vannak, hiszen a pontmutációk és az átkereszteződések gyakorisága, a kiinduló populáció nagysága stb. jelentősen befolyásolja a hatékonyságot (l. Moraczewski et al. 1995 vizsgálatsorozatát). Nem kétséges, hogy idővel az ilyen evolúciós algoritmusok is megjelennek majd a kommerciális programcsomagokban.

K: *Ez egy igen tanulságos kitérő volt számomra, s megmutatja, hogy milyen érdekes kutatási témák rejlenek az osztályozás témakörében. De most hadd térjek vissza az általad bemutatott példákra, mert van velük kapcsolatban még néhány észrevételem. Érdekes, hogy a három összehasonlított módszer a random esetre és a jól elváló, négy aggregátumos esetre adott csak azonos eredményt (a és b esetek). Ez utóbbit még csak értem, hiszen valóban "ideális" csoportosulásokról van szó. Az azonban már nem világos előttem, hogy miért pont a szabályos elrendeződésre adták az egymástól legkülönbözőbb eredményeket (az f ábrákon)?*

V: Hát éppen ez az: a szabályos elrendeződés, amikor is a pontok – némi "zajjal" megspékelve (l. az A3 táblázat utolsó két oszlopát) – egy négyzetrács kereszteződéseiben helyezkednek el, a lehető legkevésbé felel meg az osztályozhatóság követelményeinek. A példával tehát, miután a Te figyelmedet sem kerülte el a dolog, sikerült megmutatnunk, hogy az eredmények közötti jelentős eltérés mindenképpen az osztályozhatatlanság jele.

K: *Nekem úgy tűnik, mintha az index-független particionálás általában jobb eredményt adott volna, mint a másik kettő. Legalábbis...*

V: Hadd szakítsalak máris félbe. Ne hagyd magad félrevezetni! A példákkal nem "bizonyítottunk" semmit, s főleg azt nem, hogy az A módszer minden esetben jobb a B-nél! Az viszont talán kiderült az eddigiekből is, hogy egy-egy eredménnyel nem szabad megelégednünk, s célszerű annyiféle eljárást kipróbálnunk, amennyit csak lehet. A mai számítógépeken ez már igazán nem lehet gond.

K: *Igen ám, de akkor mit csinálok azzal a sok-sok eredménnyel amit ugyanazon adatok különféle feldolgozásaival kapok?*

V: A kérdés – mint már korábban is sokszor – nagyon találó, de hadd várassalak meg a válasszal egészen a 9. fejezetig, amelyet szinte teljes egészében ennek a problémának szentelek.

K: *Akármi is lesz a megoldás, fogadjunk, hogy a térsorok itt is beugranak majd!*

V: Ördögöd van, a fejezet legutolsó példája erre szeretett volna közvetve utalni. Az *f* fuzzy paraméter változtatásával kapott osztályozások sorozata mi más, lenne mint egy térsor? Bár csak két értéket néztünk meg (a 4.9 ábrán), már az is igazolta: az *f* értékek fokozatos változtatásával létrehozható egy *osztályozási sor,* amely sokkal, de sokkal több információt nyújt az osztályozott objektumokról, mint bármelyikük önmagában. De mondom, az értékelés további lehetőségeivel még várnék.

K: *Jó-jó, de akkor még annyit árulj el, hogy mely területeken tekinthető kiemelten fontosnak a nem-hierarchikus osztályozás?*

V: Például a vegetációtérképek készítésében, hiszen maga a térkép – amennyiben különféle vegetációtípusokat más és más színnel jelölünk – is egy klasszifikáció. A rendszertanost is erőteljesen érdekelheti, hogy egy taxonon belül milyen egyenrangú kategóriák különíthetők el (pl. egy faj populációin belül). De, hogy egy számunkra csupán különlegességnek tűnő dolgot is említsek, Kanadában pl. áruházi tolvajok, helyesebben a tolvajlási "stílusok" tipizálására is alkalmazták már a particionálás módszereit (McShane & Noonan 1993).