A CHAID ALAPÚ DÖNTÉSI FÁK JELLEMZŐI

HÁMORI GÁBOR

Napjainkban, elsősorban a fejlett statisztikai kultúrájú országokban, egyre szélesedik a döntési, más néven klasszifikációs fák alkalmazási köre. Segítségükkel döntési szabályok hozhatók létre, szegmentálásra, osztályozásra nyílik lehetőség. A fák működésének hátterében meglehetősen bonyolult statisztikai algoritmusok állnak, melyek közül napjainkban négy eljárásnak van letisztult módszertana. Ez a CHAID (Chi-squared Automatic Interaction Detector), az Exhaustive CHAID, a C&RT (Classification and Regression Trees) és a QUEST (Quick, Unbiased, Efficient Statistical Tree).

Jelen tanulmány célja, hogy az első két eljárás, a CHAID és az Exhaustive CHAID algoritmus működését bemutassa.

TÁRGYSZÓ: Automatikus osztályozás. Döntési fák.

A CHAID, olyan többváltozós rekurzív klasszifikáló eljárás, melyet *G. Kass* fejlesztett ki 1980-ban. Az eljárást Kass eredetileg kategóriás kimenetű változókra fejlesztette ki, de később sor került az algoritmus továbbfejlesztésére, mely így már alkalmassá vált mind a függő változó, mind a független változók esetében folytonos ismérvek kezelésére. Az exploratív algoritmus fő célja, hogy a megfigyeléseket a függő változó (Y) szempontjából úgy csoportosítsuk, hogy a csoportokon belüli variancia minél kisebb, míg a csoportok közötti variancia minél nagyobb legyen. Az eljárás során kirajzolódik a magyarázó változók (X_i) hierarchiája is aszerint, hogy a célváltozó varianciáját mekkora mértékben magyarázzák.

Mindezek miatt a CHAID kedvelt szegmentációs technika is és mint ilyen méltó vetélytársa a hagyományos klaszteranalízisnek, mely alapvetően csak mennyiségi változókkal leírható megfigyelések csoportosítására alkalmas. Gyors elterjedésének és népszerűségének fő oka, hogy a kijelölt függő változó és a magyarázó változók közötti kapcsolatrendszert vizuális formában, könnyen értelmezhető fastruktúrában (decision/classification tree) lehet láttatni. Könnyű interpretálhatósága következtében különös népszerűségnek örvend az adatbányászok körében. A modellkészítő statisztikus szempontjából az eljárás nagy előnye, hogy a változók mérési skálájára és azok eloszlására vonatkozóan semmilyen megkötést nem követel meg, folytonos és kategóriás függő és független változókat egyaránt képes kezelni.

AZ ALAPMODELL

Kezdjük az algoritmussal való ismerkedést először a Kass által kifejlesztett eredeti eljárással, melynél a függő változó és a magyarázó változók egyaránt kategóriásak. A függő változó (Y) kijelölése után a CHAID-modellt alkotó rekurzív algoritmus három fő lépésből áll:

– minden egyes magyarázó változó esetében, a függő változóra vonatkozóan, statisztikailag független, pontosabban a statisztikailag legkevésbé összefüggő kategóriák *egyesítése* (merging);

– a megfigyelések, a függő változó tekintetében legkevésbé függetlennek tekinthető magyarázó változó kategóriái szerinti *felosztása* (splitting);

– az algoritmus addig folytatja rekurzív módon a kategóriák egyesítését és az esetek felosztását, míg el nem ér valamely előre definiált *megállítási* kritériumot (stopping).

A független változók kategóriáinak egyesítése

Az első lépésben a CHAID minden magyarázó változó esetében összevonja azokat a kategóriákat, melyek legkevésbé különböznek egymástól az m különféle kategóriával rendelkező célváltozó tekintetében. Ehhez X_i kategóriái közül az összes lehetséges módon kiválaszt kettőt. Amennyiben a vizsgált magyarázó változó K különböző kategóriával rendelkezik, a kiválasztás $K\times(K–1)/2$ féleképpen történhet. Ezt követően $K\times(K–1)/2$ különböző, $(2\times m)$ méretű kontingenciatáblára Pearson-féle khi-négyzet teszt segítségével kiszámolja, hogy milyen p szignifikanciaszinten tekinthetők X_i kiválasztott kategóriapárjai és Y kategóriái függetlennek egymástól.

A következő lépésben kiválasztásra kerül az a kontingenciatábla, mely a *legmagasabb p* értékkel rendelkezik. Ezt az értéket az eljárás összeveti, a modellkészítő által előre lerögzített, $\alpha_{\text{egyesítés}}$ küszöbértékkel (a programcsomagok általában a szokásos 5 százalékos szignifikanciaszintet szokták felkínálni alapértelmezésként). Amennyiben $p > \alpha_{\text{egyesítés}}$ a kontingenciatáblázat X_i kategóriapárja egy új önálló kategóriába kerül egyesítésre. Ebben az esetben X_i eredeti kategóriáinak száma eggyel csökkent, és az algoritmus újból indul az elejétől, azaz az „új” kategóriapárok kiválasztásától (amelyek között nyílván lehetnek olyanok is, melyeket az előző ciklusban is kiválasztottak), az azokhoz rendelt kontingenciatáblákhoz tartozó p értékek kiszámolásáig.

A kategóriák összevonásának ciklusa mindaddig folytatódik, míg a legmagasabb p értékkel rendelkező kontingenciatáblára igaz nem lesz a $p > \alpha_{\text{egyesítés}}$ feltétel. Ekkor a vizsgált magyarázó változó (X_i) esetében a ciklus leáll, és az algoritmus a következő lépésben most már X teljes, lehetséges összevonások utáni, új kategória-struktúrájára számolja ki p értékét. A könnyebb eligazodás végett jelöljük az i magyarázó változó esetében ezt a szignifikanciaszintet $p_{x(i)}$ módon. Ezek után a modellkészítő igénye szerint kerül sor a $p_{x(i)}$ ún. Bonferroni-kiigazításra.[1] (Lásd bővebben a Függelékben.)

[1] A Bonferroni-kiigazítást több hipotézis egyidejű tesztelése során szokták alkalmazni. Amennyiben „n” féle különböző hipotézisünk van, melyeket külön-külön α szignifikanciaszinten tesztelnénk, belátható, hogy együttes fennállásuk esetén a szignifikanciaszintet α/n szinten kell megválasztani ahhoz, hogy az első fajú hiba elkövetésének valószínűsége ne legyen nagyobb, mint α. Esetünkben a különböző és egyidejűleg fennálló hipotéziseket a fastruktúra különböző szintjein vizsgált függetlenségi hipotézisek jelentik.

A CHAID minden magyarázó változó esetében végrehajtja a fent leírtakat, aminek eredményeképpen az összes X_i esetében megtörténnek a lehetséges kategóriaösszevonások, és minden magyarázó változó rendelkezik egy $p_{x(i)}$ (vagy kiigazított $p_{x(i)}$) értékkel.

A felosztás

A következő lépésben az X_i magyarázó változók közül kiválasztásra kerül a *legkisebb* „$p_{x(i)}$" értékkel rendelkező. Ezt az értéket az eljárás összeveti, a modellkészítő által előre meghatározott, $\alpha_{\text{felosztás}}$ küszöbértékkel. Amennyiben $p_{x(i)} < \alpha_{\text{felosztás}}$, megtörténik az esetek felosztása X_i kategóriái szerint. A felosztás eredményeként a megfigyelések adatbázisa annyi részre esik szét, ahány (lehetséges összevonások utáni) kategóriával a felosztás alapjául szolgáló magyarázó változó rendelkezett. A felosztás utáni részadatbázisok fogják a fastruktúra következő szintjét jelenteni.[2] Ha $p_{x(i)} > \alpha_{\text{felosztás}}$, a felosztás nem történik meg, az adott szint tovább már nem bontható.

A megállás

Felosztás után az algoritmus az első pontnál (kategóriák egyesítése) újraindul, azzal a különbséggel, hogy most már az esetek felosztása után létrejövő részadatbázisokon külön-külön folytatódik a magyarázó változók kategóriáinak összevonása, majd az újbóli felosztás. A ciklusok (összevonás–felosztás) mindaddig tartanak, amíg el nem érik valamelyik megállási kritériumot. Ezek a következők lehetnek:

– $p_{x(i)} > \alpha_{\text{felosztás}}$;

– az esetek a magyarázó változók tekintetében nem különböznek egymástól (ugyanazon értékekkel rendelkeznek minden magyarázó változóra vonatkozóan);

– az esetek a célváltozó ugyanazon értékével rendelkeznek;

– a felosztandó részadatbázis esetszáma nem éri el a modellkészítő által előre definiált esetszámot;

– az újbóli felosztás során keletkező új részadatmátrixok valamelyikének esetszáma nem éri el a modellkészítő által előre definiált esetszámot;

– a felosztások száma eléri a modellkészítő által előre definiált számot (A fastruktúra szintjeinek száma = felosztások száma).

A leírtak szemléltetésére vegyünk egy példát, mely a *SPSS Answer Tree* számítógépes programcsomag segítségével készült. A példában egy hitelminősítési problémát találunk, melyben a rendelkezésre álló adatbázis segítségével szeretnénk kategorizálni a kérelmezőket aszerint, hogy mekkora hitelkockázatot jelentenek. Az adatbázis 323 esetet tartalmaz, négy, kategóriás kimenetelű magyarázó változóval. Ezek a következők:[3]

– X_1: korosztály (fiatal, középkorú, idős) – Age Categorical (young, middle, old);

– X_2: van-e AMEX kártyája (igen/nem) – AMEX card (yes/no);

– X_3: fizetését hetente vagy havonta kapja (hetente/havonta) – Paid weekly/monthly (weekly pay/monthly salary);

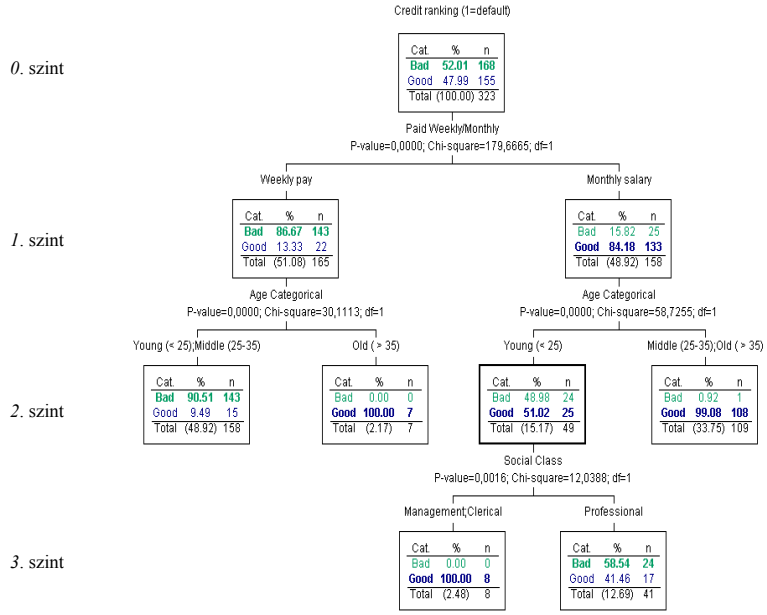
[2] Az első szint, maga a kiindulási adatbázis volt.

[3] Az SPSS Answer Tree eredményei angol nyelven jelennek meg, ezért a felsorolásban magyarul és angolul egyaránt feltüntetik az egyes változók neveit és kategóriáit.

– X_4: foglalkozása (vezető, szabadfoglalkozású, irodai foglalkozású, szakmunkás, segédmunkás) – Social class (management, professional, clerical, skilled, unskilled).

Célváltozó (Y) értelemszerűen legyen a hitelbesorolás, melynek két kategóriája van: *jó* és *rossz*. A CHAID modell megépítésének első lépése, hogy lerögzítjük $\alpha_{\text{egyesítés}}$ és $\alpha_{\text{felosztás}}$ értékeit. Mindkettőnél a program által alapbeállításként ajánlott $\alpha = 0{,}05$ szintet fogadtuk el. A második lépés a megállási kritériumok meghatározása. A program a felosztások maximális számára vonatkozóan négyet javasol, ami a fastruktúrában legfeljebb öt szintet jelent. A felosztásra kerülő részadatbázisok minimális esetszáma 25, a felosztás során keletkező új részadatbázisok minimális esetszáma pedig 1.

Az alapparaméterek beállítása után az SPSS Answer Tree programcsomag a következő ábrán látható CHAID modellt alakította ki.



A *0*. szinten látható a teljes adatbázis eloszlása a célváltozó kategóriái szerint. Egy-egy kis tábla jobb oldali oszlopa tartalmazza az egyes kategóriák elemszámait, a középső oszlop ugyanezek százalékos megoszlását, míg a bal oldali oszlopban láthatjuk feltüntetve az egyes kategóriákat. Ha csak ezt a változót ismernénk, és ennek alapján kellene döntenünk egy hitelkérelemről, a legkisebb hibát akkor követnénk el, ha mindenkit elutasítanánk. Ekkor összességében a minta alapján az esetek 48 százalékánál hibát követnénk el. A CHAID segítségével a téves döntések aránya csökkenthető. Ehhez az algoritmus első részében minden egyes magyarázó változó esetében elvégzi a lehetséges összevonásokat, majd a magyarázó változók közül kiválasztja a legkisebb $p_{x(i)}$ értékkel rendelkezőt. Esetünkben ez a kérelmező fizetésének gyakoriságát leíró változó ($p_{x(i)} = 0{,}0000$, *khi-négyzet = 179,6665*). Ez azt is jelenti, hogy ennek a változónak a kategóriái gyakorolják a

legnagyobb befolyást a hitelkockázatra. A kiinduló adatbázis felosztása ennek a változónak a kategóriái mentén történt meg. Ezzel elérkeztünk a fastruktúra *1*. szintjére. A felosztás eredményeképpen előálló két részadatbázisban újból megfigyelhetők a célváltozó kategóriáinak eloszlásai az egyes részadatbázisokat reprezentáló táblácskákban. Látható, hogy az ismérvkategóriához való tartozás ismerete a hitelezés kockázatával kapcsolatos bizonytalanságunkat lényegesen csökkenti. Ha mindenkit, aki hetente kap fizetést, elutasítanánk és mindenkinek, aki havonta kap jövedelmet hiteleznénk, akkor az esetek (22+25)/(165+158) = 14,55 százalékban döntenénk csak helytelenül az adatbázis által reprezentált világban. A kezdeti 48 százalékos döntési bizonytalanságunkat PRE = (48 – – 14,55) / 48 = 69,7 százalékkal sikerült csökkenteni azáltal, hogy ismerjük a fizetések gyakoriságát.[4] A 14,55 százalékos bizonytalanság tovább csökkenthető, ha az összevonó–felosztó algoritmust tovább folytatjuk és a fastruktúra *2*. szintjére lépünk. Látható, hogy a következő legnagyobb hatású (legkisebb $p_{x(i)}$-szel rendelkező) magyarázó változó a korosztály. A két részadatbázison a korosztály kategóriáinak összevonása másképpen történt meg: a havi fizetéseknél a fiatalok, míg a heti fizetéseknél az idősek képeznek önálló kategóriát, míg a másik két korosztályi kategória összevonásra került. Döntési bizonytalanságunk most már (15+0+24+1)/(158+7+49+109) = 12,38 százalékra csökkent (PRE = 14,9%). A fastruktúrának ezen a szintjén már négy diszjunkt részadatbázisra lett felosztva az eredeti adatmátrix. A fastruktúra heti fizetések ágán a korosztályi kategóriák alapján felosztott két részadatbázist az algoritmus már tovább nem bontotta. Az „idős" kategóriánál ez azért következett be, mert ennek a részadatbázisnak az elemszáma kisebb a modell futtatása előtt beállított huszonötnél, tehát az algoritmus itt elért egy megállási kritériumot. A „középkorú–fiatal" összevont kategóriával jellemzett részadatbázis elemszáma ugyan kellően nagy (158), de a $p_{x(i)}$ értékek egyike sem volt kisebb[5] az előre beállított $\alpha_{\text{felosztás}} = 5$ százalékos értéknél, így a felosztás nem következett be. A fastruktúra havi fizetések ágán, a „fiatal" korosztály ágán tudott az algoritmus a fa *3*. szintjére lépni. A felosztás a foglalkozás szerint történt meg. Látható, hogy a vezető és az irodai dolgozó kategória összevonásra került, és feltűnhet, hogy hiányzik a szakmunkás és a segédmunkás kategória. Ennek az a magyarázata, hogy az induló adatbázis nem tartalmazott olyan esetet, melynél fiatal és havi fizetéssel rendelkező kérelmező szak- vagy segédmunkás lett volna. Az alapparaméterek rögzített szintjén a fastruktúra kiépítése véget ért. A végső struktúra segítségével a hitelkockázattal kapcsolatos döntési bizonytalanságunk 10,2 százalékra csökkent (PRE = 17,6%). Az eljárást összefoglalóan mutatja a következő tábla.

A CHAID eljárás összefoglaló táblája

Szint	Hibás besorolások aránya (százalék)	PRE mutató (százalék)	
		az előző szinthez viszonyítva	a *0*. szinthez viszonyítva
0.	48,00	–	–
1.	14,55	69,7	69,7
2.	12,38	14,9	74,2
3.	10,20	17,6	78,8

[4] A PRE mutató a kapcsolat szorosságának általános mutatószáma, azt méri, hogy egy újabb változó bevonása a magyarázó változók közé hány százalékkal csökkenti a magyarázat bizonytalanságát. Ebben az esetben konkrét jelentése az, hogy a hibás besorolások száma az egyes szinteken hány százalékkal csökken a korábbi szinten mérthez képest.

[5] Itt már csak két magyarázó változó a „foglalkozás" és „Van AMEX kártyája" esetében kerül sor „ $p_{x(i)}$" érték számítására.

A tábla második oszlopa jól mutatja, hogy a növekvő szintek (növekvő számú magyarázó változó) hogyan eredményeznek egyre pontosabb besorolásokat. A harmadik oszlop a viszonylagos hibacsökkenést mutatja. Látható, hogy ez tendenciaszerűen csökkenő, de a csökkenés nem monoton. Végül az utolsó oszlop monoton növekvőn mutatja, hogy az induló állapothoz képest az egyes lépések után mekkora relatív hibacsökkenés érhető el. A végeredményül kapott 78,8 százalék jelentése az, hogy az összes szóba jöhető magyarázó változó együttesen közel 80 százalékkal csökkenti a hitelbesorolásban tapasztalt induló bizonytalanságot. Az ehhez tartozó döntési szabály tehát jó alapot nyújt a banknak a kérelmezők kockázat szerinti besorolására.

A beállítható paraméterek

Általában elmondható, hogy adott adatbázis esetén a fa összetettségét és mélységét (szintjeinek számát) alapvetően az határozza meg, hogy a futtatás előtt milyen értéken rögzítjük az alapparamétereket, melyek összefoglalva a következők:

– $\alpha_{\text{egyesítés}}$,
– $\alpha_{\text{felosztás}}$,
– felosztások maximális száma,
– felosztandó részadatbázis minimális esetszáma,
– a felosztással keletkező részadatbázisok minimális esetszáma.

Minél kisebb $\alpha_{\text{egyesítés}}$, annál több kategória egyesítésére számíthatunk a magyarázó változók esetében. Az $\alpha_{\text{felosztás}}$ kis értéke, viszont a felosztások számát, és ezáltal a fa összetettségét, csökkenti.

A CHAID TOVÁBBFEJLESZTÉSEI

A 80-as években a CHAID-et számos területen alkalmazták. A gyakorlat során merült fel az az igény, hogy a CHAID képes legyen mind a magyarázó változók, mind a célváltozó tekintetében mennyiségi ismérvek kezelésére is. A problémát a magyarázó változók esetében úgy oldották meg, hogy az algoritmus a mennyiségi változókat kategóriás változókká transzformálja oly módon, hogy X decilisei által meghatározott intervallumokat tekinti kategóriáknak. Amennyiben a célváltozó mennyiségi, az algoritmus khi-négyzet tesztek helyett F-teszteteket alkalmaz annak megállapítására, hogy milyen p szignifikanciaszinten tekinthetők a célváltozó X kategóriapárjai által meghatározott részátlagai azonosnak. Az alkalmazás során derült ki az algoritmus azon gyengesége is, miszerint a kategóriák összevonása során nem mindig éri el azt a kategória-struktúrát, melynél a p_x érték a legkisebb, azaz, amelynél a felosztás optimális (amennyiben az adott változó mentén történik az adatbázis felosztása). Ez annak a következménye, hogy az összevonási algoritmus leáll, amennyiben a megmaradt kategóriapárokat az algoritmus $\alpha_{\text{egyesítés}}$ függvényében statisztikailag függetlennek tekinti. A probléma orvoslására javasolta 1991-ben *D. Biggs*, *B. de Ville* és *E. Suen* az eredeti CHAID továbbfejlesztését, amelyet *Exhaustive* CHAID-nek neveztek el. Az *Exhaustive* CHAID csak annyiban kü

lönbözik az eredeti CHAID-től, hogy az összevonási algoritmusnál nincs $\alpha_{\text{egyesítés}}$ összevonási és leállási kritérium. E helyett úgy dolgozik, hogy mindenféleképpen egyesíti az algoritmus első ciklusában azt a kategóriapárt, melynek a legmagasabb a p értéke, majd az így keletkezett új kategória-struktúrára kiszámolja a p_x-t. Az így nyert kategória-struktúrát a hozzá tartozó p_x-szel együtt „megjegyzi". A következő lépésben az eljárás belép a második ciklusba és az előző kategória-struktúrán újból kialakítja a lehetséges kategóriapárokat és egyesíti azt a kategóriapárt, melynek legmagasabb a p értéke. Az új struktúrára megint kiszámolja a p_x értéket és a struktúrával együtt eltárolja a memóriájában. Az eljárás mindaddig folytatódik, míg csak két kategória marad. Ekkor az algoritmus visszamenőleg megkeresi azt a kategória-struktúrát, melyhez a legkisebb p_x érték tartozott. Ha majd az esetek felosztása ennek a változónak a mentén történik, a felosztás alapja az így létrehozott kategória-struktúra lesz. Ettől a ponttól kezdve minden ugyanúgy megy tovább, mint a CHAID esetében (ezután kerül sor a különböző magyarázó változókhoz tartozó p_x értékek összevetésére stb.).

Az Exhaustive CHAID hátránya a CHAID eljárással szemben az, hogy lényegesen számításigényesebb, aminek következtében, különösen nagy adatbázisok esetében a modell felépítésének időtartama jelentősen megnövekedhet. Ráadásul a gyakorlati tapasztalat azt mutatja, hogy sok esetben ugyanannak a problémának a megközelítésekor az CHAID és az Exhaustive CHAID ugyanazon fastruktúra kialakulásához vezet.

FÜGGELÉK

A Bonferroni-kiigazítás. Tegyük fel, hogy két hipotézisünk van, melyeket egyaránt α szignifikanciaszinten kívánunk tesztelni. Mindkét hipotézis esetén α jelenti az első fajú hiba elkövetésének valószínűségét, azaz a két hipotézis vonatkozásában ugyanazt az α szignifikanciaszintet határozzuk meg. Jelöljük A_1-gyel az egyik és A_2-vel a másik hipotézis esetében az első fajú hiba bekövetkezésének eseményét. Ekkor, a valószínűség-elmélet által használt jelölésekkel

$$P(A_1) = P(A_2) = \alpha .$$

Annak a valószínűsége, hogy legalább az egyik hipotézis esetében elkövetjük az első fajú hibát:

$$P(A_1 \cup A_2) = P(A_1) + P(A_2) - P(A_1 \cap A_2) .$$

Jelöljük A_1 komplementerét A_{1c}-vel és A_2 komplementerét A_{2c}-vel. Ezek jelentik azt, hogy az első fajú hibák nem következnek be. Annak együttes valószínűsége, hogy sem az első, sem a második hipotézis esetében nem követjük el az első fajú hibát:

$$P(A_{1c} \cap A_{2c}) = 1 - P(A_1 \cup A_2) = 1 - P(A_1) - P(A_2) + P(A_1 \cap A_2).$$

Mivel tudjuk, hogy $P(A_1 \cap A_2) \geq 0$, felírható a következő, ún. Bonferroni-egyenlőtlenség két hipotézisre:

$$P(A_{1c} \cap A_{2c}) \geq 1 - P(A_1) - P(A_2) = 1 - 2\alpha.$$

Az általános formula n különböző hipotézis esetén:

$$P\left(\bigcap_{i=1}^{n} A_{ic}\right) \geq 1 - n\alpha .$$

Ha például tíz különböző hipotézissel dolgozunk egyszerre, melyek mindegyikét $\alpha = 0{,}05$ szignifikanciaszinten teszteljük, látható, hogy annak a valószínűsége, hogy egyik hipotézis esetében sem követjük el az első fajú hibát, nagyobb vagy egyenlő, mint 50 százalék. Ha az együttes valószínűségre a szokásos $\alpha = 0{,}05$ korlátot szeretnénk definiálni, akkor az egyedi (hipotézisenkénti) szignifikanciaszintet legfeljebb $\alpha/n = 0{,}05/10 = 0{,}005$ értéken szükséges rögzíteni.

IRODALOM

BIGGS, D. – DE VILLE, B. – SUEN, E. (1991): A method of choosing multiway partitions for classification and decision trees. *Journal of Applied Statistics*, 18. sz. 49–62. old.

GNANADESIKAN, R. (1977): *Methods for statistical data analysis of multivariate observations.* John Wiley & Sons. Inc. New York.

KASS, G. (1980) An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29. évf. 2. sz. 119–127. old.

MAGIDSON, J. – SPSS INC. (1993): *SPSS for Windows CHAID release 6.0.* SPSS Inc. Chicago.

SUMMARY

CHAID, or Chi-square Automatic Interaction Detection is a classification tree modelling technique. This exploratory data analysis method is used to study the relationships among a dependent variable and a large series of possible predictor variables and their interactions. The CHAID evaluates complex interactions of the predictors and the dependent variable, and displays the modelling results in an easy-to-interpret tree diagram.