TDK-dolgozat

Puskás Levente

Bizonytalan és folytonos adatok kezelése döntési fákkal

Szerző:
Puskás Levente
Msc. Programming Informatics 2.semester
Konzulens:
Dombi József
Professor

Szegedi Tudományegyetet
Természettudományi és Informatikai Kar
Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszék

Kivonat

A döntési fák az egyszerűségük és belső interpretálhatóságuk miatt az adatbányászat és gépi tanulás egyik legfontosabb eszközei közé tartoznak. Az ID3 volt az első algoritmus, amelyet döntési fák létrehozására fejlesztettek ki, azonban számos korlátja miatt az idők során több, továbbfejlesztett változata jelent meg, amelyek célja az eredeti algoritmus hatékonyságának és alkalmazhatóságának növelése. A jelen dolgozat a Learning Decision Trees in Continuous Space című cikk folytatása, amelyben egyszerűsítettük az ott bevezetett bizonytalanság mértéket (vagueness measure). Az adatelemzési problémákban gyakran előfordul, hogy a bemeneti vagy kimeneti változókról csupán valószínűség-alapú becslések állnak rendelkezésre. Ez jelentős kihívás elé állítja a modellezést, mivel egy ilyen adathalmaz nehezen értelmezhető, és nem egyértelmű, hogy mi az elvárt kimenet. A dolgozatban olyan döntési fa algoritmust fejlesztettünk, amely kezeli ezt a bizonytalanságot. A korábbi cikk bemutatta a döntési tér szeparálását köröket és egyeneseket felhasználva és lehetőséget nyújtott a folyamatos döntési terek strukturált feldolgozására. Tanulmányunkban ezt az eljárást továbbfejlesztettük, újabb függvényeket és eljárásokat vezettünk be ilyen alakzatokkal való szeparálásra. Továbbá olyan módszert is fejlesztettünk, ami a szeparált döntési teret Voronoid-diagrammá alakította. Ez a megközelítés nagyobb rugalmasságot és pontosságot nyújt a döntési határok megadásában, különösen olyan adatok esetén, amelyekben a különböző osztályok komplex, nem lineáris határvonalakkal is szeparálhatók.

1. Bevezetés

A kutatás folytatása a [J D01] cikknek. A jelen cikkben ismertetjük a döntési fa konstrukcióját valós értékű adatbázison tetszőleges implicit függvények felhasználásával és bizonytalan értékekkel. A döntési fa első koncepciója[Bre+84][Qui86] a tanulóalgoritmusokhoz tartozik: adatbázisban rekordokhoz hozzá rendelt az osztályhoztartozás. A példákhoz hozzárendelt tulajdonság lehet diszkrét vagy folytonos érték. A klasszikus döntési fa esetében a tulajdonságok diszkrét értékek. Ezek meglétét vagy hiányát 0 - 1 el reprezentált. Ebben az esetben a döntési fa megadja, hogyan készítsünk egy olyan eljárást amely csomópontjai a tulajdonságok a legalsó szinten lévő levelek pedig megadják hogy milyen osztályba tartozik az adott rekord.
Ilyen döntési fa mindig készíthető ha az adatok konzisztensek. A tanuló algoritmus feladata minimális mélységű fa konstruálása, amit úgy ér el hogy a döntési fa csomópontjain a megfelelő tulajdonságot választjuk ki.A probléma matematikai szempontból nehéz (NP teljes[HR76]) azaz nem adható rá polinomiális megoldás. A gyakorlati megvalósításban a fa építése során heurisztikákat alkalmaznak amit az entrópia függvény segítségével konstruált. A legkorábban kifejlesztett ilyen algoritmus az ID3, de azóta sok változata jött létre az eredeti algoritmus hiányosságainak kiküszöbölésére. Az egyik legjelentősebb hátrány, ha egy jellemző sok diszkrét értékre bontható Pl.: Vitorázás esetén a szél erősségét öt kategóriára bontjuk. A heurisztikus algoritmus előnyben részesíti azokat az osztályokat amelyeknek több diszkrét értéke van. A másik probléma ami ineffektívvé teheti az ID3 - at ha a tanulandó osztályok száma magas. Jelen dolgozatban egy olyan döntési fa konstrukciót vizsgáltunk aminek célfüggvénye egy eldöntendő kérdés, azaz egy igen/nem van a kimeneten, pl.: menjünk e vitorlázni vagy adott betegségtől szenved e az illető. A dolgozatban azt is vizsgáltuk hogyha a tulajdonságok nem diszkrétek, hanem folytonos változó értékek.

A heurisztikus entrópia függvényt egy egyszerűbben számítható bizonytalanság függvénnyel helyettesítjük. Egy lényeges újítás hogy a szeparáció ami a klasszikus döntési fában egy számegyenes pontjainak megkeresése. Az eljárást úgy fejlesztjük hogy egy n dimenziós szeparálást keresünk

pl.: 2 dimenzió esetében egy kör középpontját és sugarát keressük meg, ahol a teret szeparáló kör belsejében a pozitív kívül a negatív példák vannak. Vagy a 2 dimenziós térben egy polinomot mkeresünk ahol a feladat a polinom egyenletének megtalálása. Megadtunk olyan algoritmusokat amely kombinálja a kör alapú és Voronoi alapú szeparálásokat ezzel egy jobb módszert adva a folytonos értékek kezelésére. Az eljárás követi az ID3 eredeti koncepcióját. A döntési fa klasszikus számításait egy új táblázat bevezetésével egyszerűsítjük és a koncepció alapján a tulajdonság meglétének valószínűségét is kezelni tudjuk. Azaz ha a szél ereje 3 kategóriára bontott akkor egy 3 hosszú vektor a klasszikus megközelítésben lehet [0,1,0] ami jelenthet közepes szélerősséget, de e helyett ezek lehetnek valószínűségek pl [0.1, 0.3, 0.6], egy $n, n > 2$ tulajdonságú változó előnyét úgy eliminájuk hogy a tulajdonságokat tovább bontjuk 2 csoportra, így az előző példán: [0.9,0.1], [0.3,0.7], [0.6,0.4]. A koncepció a válaszokra is kiterjszthető azaz azok bizonytalanságát is tudjuk majd kezelni, az új koncepció megnöveli a döntési fák alkalmazhatóságának körét.

2. Klasszikus döntési fa

Ahhoz hogy ismertessük az új koncepciónkat először megmutatjuk a klasszikus koncepciót az új jelöléseinkel. Az adataink következő képpen állnak rendelkezésünkre ha a feladat egy eldöntendő kérdésre a minimális döntési fa meghatározása:

1. táblázat. Eredeti táblázat

	C_1	$\cdots$	C_k	$\ldots$	C_n	R
a_1	$C_1^{(1)}$	$\cdots$	$C_k^{(1)}$	$\ldots$	$C_n^{(1)}$	r_1
a_2	$C_1^{(2)}$	$\cdots$	$C_k^{(2)}$	$\ldots$	$C_n^{(2)}$	r_2
a_j	$C_1^{(j)}$	$\cdots$	$C_k^{(j)}$	$\ldots$	$C_n^{(j)}$	r_j
$\vdots$	$\vdots$			$\vdots$		$\vdots$
a_m	$C_1^{(m)}$	$\cdots$	$C_k^{(m)}$	$\ldots$	$C_n^{(m)}$	r_m

A következő változók bevezetése szükséges a számítások elvégzéséhez:

S összes példák száma

S^+ pozitív példák száma

S^- negatív példák száma

$S_{k,i}$ k jellemző i lehetséges diszkrét értéke

$S_{k,i}^+$A pozitív példák száma $S_{k,i}$ - ben

$S_{k,i}^-$ A negatív példák száma $S_{k,i}$ - ben

A klasszikus ID3 algoritmusban a fát iteratívan építjük és minden iterációban a csúcsot a maximális információ nyereségű jellemző alapján választjuk. Az információ nyereség[Vet00] definiálásához szükségünk van az entrópiára ami a következő:

$$\mathcal{E}(x_i) = -k \sum_{i=1}^{n} x_i ln(x_i)$$

Használva a jelöléseiket a következő egyenletet kapjuk:

$$J(S_{k,i}) = -\frac{1}{ln(2)} \left(\frac{S^+_{k,i}}{S_{k,i}} ln \frac{S^+_{k,i}}{S_{k,i}} + \frac{S^-_{k,i}}{S_{k,i}} ln \frac{S^-_{k,i}}{S_{k,i}} \right).$$

Az egyes jellemzők várható értéke:

$$E_S(C_k) = \frac{S_{k,1}}{S} J(S_{k,1}) + \frac{S_{k,2}}{S} J(S_{k,2}) + \ldots \frac{S_{k,n_k}}{S} J(S_{k,n_k}) \tag{1}$$

Az információ nyereség:

$$IG(A, C_k) = J(S) - E_S(C_k)$$

A következő példán megmutatunk egy klasszikus esetet.

2.1. Példa

Adatbázis:

	C_1	C_2	C_3	R
1	B	3	b	+
2	A	3	a	-
3	A	2	b	+
4	B	1	b	-
5	A	1	b	-
6	A	3	b	+
7	A	1	a	-
8	B	3	a	-

1. jellemző	(A, B)	(C_1)	$S_{1,1} = A$	$S_{1,2} = B$	
2. Jellemző	$(1, 2, 3)$	(C_2)	$S_{2,1} = 1$	$S_{2,2} = 2$	$S_{2,3} = 3$
3. jellemző	(a, b)	(C_3)	$S_{3,1} = a$	$S_{3,2} = b$	

Ha az 1. jellemző entrópiáját akarjuk számolni akkor tudjuk hogy

$$\begin{array}{lll} S_{1,1} = 5 & S_{1,2} = 3 & S = 8 \\ S^+_{1,1} = 2 & S^-_{1,1} = 3 & S_{1,1} = 5 \\ S^+_{1,2} = 1 & S^-_{1,2} = 2 & S_{1,2} = 3 \end{array}$$

Így a számítások a következőek:

$$J(S_{11}) = -\frac{1}{ln(2)}\left(\frac{2}{5}ln\frac{2}{5} + \frac{3}{5}ln\frac{3}{5}\right) = 0.971$$
$$J(S_{12}) = -\frac{1}{ln(2)}\left(\frac{1}{3}ln\frac{1}{3} + \frac{2}{3}ln\frac{2}{3}\right) = 0.918$$
$$E_S(C_1) = \frac{5}{8}J(S_{11}) + \frac{3}{8}J(S_{12}) = 0.951$$

Ha ezt a számítást a C_2 és C_3 jellemzőkön is végigvinnénk akkor kapánk hogy

$$E_S(C_2) = 0.5$$
$$E_S(C_3) = 0.607$$

Ez alapján a C_2 jellemzőnek minimális az entrópiája így ezt választanánk.

3. Az entrópia függvény és bizonytalanság

Az entrópia függvény helyett használhatunk más függvényeket is, mi a bizonytalanság függvényt fogjuk használni. A bizonytalanságnak a következő az egyenlete:

$$J(S) = 4\frac{|S^+|}{|S|}\left(1 - \frac{|S^+|}{|S|}\right) = 4\frac{|S^+||S^-|}{|S|^2} \tag{2}$$

Ekkor a $J(S_k)$ értékek:

$$J(S_{k1}) = 4\frac{|S^+_{k1}||S^-_{k1}|}{|S_{k1}|^2}$$
$$J(S_{k2}) = 4\frac{|S^+_{k2}||S^-_{k2}|}{|S_{k2}|^2}$$
$$\vdots$$
$$J(S_{kn_k}) = 4\frac{|S^+_{kn_k}||S^-_{kn_k}|}{|S_{kn_k}|^2}$$

A C_k bizonytalansága ekkor a 1. egyenlet alapján egyszerűsíthető:

$$E_D(C_k) = 4\sum_{i=1}^{n_k}\frac{S_{ki}}{S}\frac{S^+_{ki}S^-_{ki}}{S^2_{ki}} = \frac{4}{S}\sum_{i=1}^{n_k}\frac{S^+_{ki}S^-_{ki}}{S^+_{ki} + S^-_{ki}}, \tag{3}$$

Ha az előző példát véigszámolnánk bizonytalanság mérték alapján is akkor a következő eredményeket kapnánk:

$$E_D(C_1) = \frac{4}{8}\left(\frac{2\cdot 3}{5} + \frac{1\cdot 2}{3}\right) = 0.933$$
$$E_D(C_2) = 0.5E_D(C_3) = 0.6$$

Így az általunk adott függvény a példán konzisztens volt a Shannon entrópiával.

4. ID3 a bizonytalanság mérték alapján

Ahhoz hogy összehasonlítsuk a bizonytalanság mértéket az entrópiával visszatérünk az előző példához.

4.1. A táblázat transzformálása

A továbbiakban új jelölésrendszert vezetünk be és átalakított táblázatot használunk.

2. táblázat. Új táblázat

	$\overbrace{C_{1,1} \quad C_{1,2} \quad \dots \quad C_{1,n}}^{C_1}$	$\cdots$	$\overbrace{C_{k,1} \quad C_{k,2} \quad \dots \quad C_{k,n}}^{C_k}$	$\overbrace{R^+ \quad R^-}^{R}$
a_1	$C_{1,1}^{(1)} \quad C_{1,2}^{(1)} \quad \dots \quad C_{1,n}^{(1)}$	$\cdots$	$C_{k,1}^{(1)} \quad C_{k,2}^{(1)} \quad \dots \quad C_{k,n}^{(1)}$	$r_1^+ \quad r_1^-$
a_2	$C_{1,1}^{(2)} \quad C_{1,2}^{(2)} \quad \dots \quad C_{1,n}^{(2)}$	$\cdots$	$C_{k,1}^{(2)} \quad C_{k,2}^{(2)} \quad \dots \quad C_{k,n}^{(2)}$	$r_2^+ \quad r_2^-$
a_j	$C_{1,1}^{(j)} \quad C_{1,2}^{(j)} \quad \dots \quad C_{1,n}^{(j)}$	$\cdots$	$C_{k,1}^{(j)} \quad C_{k,2}^{(j)} \quad \dots \quad C_{k,n}^{(j)}$	$r_j^+ \quad r_j^-$
$\vdots$	$\vdots$		$\vdots$	$\vdots \quad \vdots$
a_m	$C_{1,1}^{(m)} \quad C_{1,2}^{(m)} \quad \dots \quad C_{1,n}^{(m)}$	$\cdots$	$C_{k,1}^{(m)} \quad C_{k,2}^{(m)} \quad \dots \quad C_{k,n}^{(m)}$	$r_m^+ \quad r_m^-$
$\sum$	$S_{1,1} \quad S_{1,2} \quad \dots \quad S_{1,n}$	$\dots$	$S_{k,1} \quad S_{k,2} \quad \dots \quad S_{k,n}$	$S^+ \quad S^-$

a_j az j. rekord
C_i az i. jellemző
$C_{i,n}$ a C_i jellemző n. lehetséges diszkrét értéke
$C_{i,n}^j$ az a_j -hez tartozó $C_{i,n}$ diszkrét változó n. lehetséges értéke, boolean
r_j^+ az a_j - hez tartozó osztály, 1 ha pozitív példa
r_j^- az a_j - hez tartozó osztály, 1 ha negatív példa
S^- negatív példák száma
S^+ pozitív példák száma
S_{i_n} az adott jellemző diszkrét lehetséges értékei közül az i. előfordulásának a száma

Bevezetjük a következő jelöléseket:

$$\begin{array}{lll} S_{k,1} = S^+_{k,1} + S^-_{k,1} & x^+_{k,1} = \frac{S^+_{k,1}}{S^+} & x^-_{k,1} = \frac{S^-_{k,1}}{S^-} \\ S_{k,2} = S^+_{k,2} + S^-_{k,2} & x^+_{k,2} = \frac{S^+_{k,2}}{S^+} & x^-_{k,2} = \frac{S^-_{k,2}}{S^-} \\ \vdots & \vdots & \\ S_{k,n_k} = S^+_{k,n_k} + S^-_{k,n_k} & x^+_{k,n_k} = \frac{S^+_{k,n_k}}{S^+} & x^-_{k,n_k} = \frac{S^-_{k,n_k}}{S^-} \end{array} \tag{4}$$

$$S = S^+ + S^- \qquad w^+ = \frac{S^+}{S} \qquad w^- = \frac{S^-}{S},$$

Az előzőekből egyértelműen következik hogy:

$$w^+ + w^- = 1 \qquad w\epsilon[0,1]$$

$$\sum_{i=1}^{n_k} x^+_{k,i} = 1 \qquad \sum_{i=1}^{n_k} x^-_{ki} = 1 \qquad x^+_{k,i}\epsilon[0,1],\ x^-_{k,i}\epsilon[0,1]$$

4.2. Bizonytalanság érték kezelése

Azért hogy az új táblázatunkhoz egyszerűsítsük a számolást átalakítjuk a bizonytalanság mértéket. A 3 egyenletből kiindulva és a $x^+_{ki}, x^-_{ki},\ w^+$ és w^- definíciókat felhasználva kapjuk hogy:

$$E_D(C_k) = \frac{4}{S} \sum_{i=1}^{n_k} \frac{S^+ x^+_{ki}\ S^- x^-_{ki}}{S^+ x^+_{ki} + S^- x^-_{ki}}.$$

$$\begin{aligned} E_D(C_k) &= \frac{4S^+S^-}{S} \sum_{i=1}^{n_k} \frac{x^+_{ki} x^-_{ki}}{S^+ x^+_{ki} + S^- x^-_{ki}} \\ &= \frac{4S^+S^-}{S^2} \sum_{i=1}^{n_k} \frac{x^+_{ki} x^-_{ki}}{\frac{S^+}{S^+ + S^-} x^+_{ki} + \frac{S^-}{S^+ + S^-} x^-_{ki}} \\ &= 4w^+w^- \sum_{i=1}^{n_k} \frac{x^+_{ki} x^-_{ki}}{w^+ x^+_{ki} + w^- x^-_{ki}} \end{aligned} \tag{5}$$

A súlyozott Dombi konjuktív operátor:

$$c_D(u, v; x, y) = \frac{1}{1 + u\frac{1-x}{x} + v\frac{1-y}{y}},$$

Erre az alakra hozhatjuk az egyenletet.

$$4w^+w^- \sum_{i=1}^{n_k} \frac{x^+_{ki} x^-_{ki}}{w^+ x^+_{ki} + w^- x^-_{ki}} = 4w^+w^- \sum_{i=1}^{n_k} \frac{1}{1 + w^+ \frac{1-x^-_{ki}}{x^-_{ki}} + w^- \frac{1-x^+_{ki}}{x^+_{ki}}}. \tag{6}$$

Ekkor kifejezhetjük az egyenletünket az alábbi módon:

$$E_D(C_k) = 4w^+w^- \sum_{i=1}^{n_k} c_D(w^+, w^-; x^-_{ki}, x^+_{ki}). \tag{7}$$

4.3. A bizonytalanság mérték további egyszerűsítése

A (6) egyenletettel nehéz számolni, azért, átalakítjuk a döntési kritériumok kimeneti számításának menetét. Az alábbi átalakítások után a módszer csak olyan jellemzőket kezel, ami két osztállyal rendelkezik. Ez nem korlátoz, mert tetszőleges m diszkrét változóval rendelkező jellemzőt szétbonthatunk m jellemzőre, mivel

$$x_{k_1}^+ + x_{k_2}^+ = 1 \quad \text{és} \quad x_{k_1}^- + x_{k_2}^- = 1$$

Amiből következik hogy,

$$x_{k_1}^+ = 1 - x_{k_2}^+ \quad \text{és} \quad x_{k_1}^- + = 1 - x_{k_2}^- \tag{8}$$

Tehát minden jellemző implicit módon megadja a másik lehetséges osztályát. Ezért az eredetileg is két osztállyal rendelkező jellemzők esetén elég csak az egyik osztályt megadni .
Az alábbi egyenletet felhasználva:

$$K = \frac{x_{k_1}^+ x_{k_1}^-}{w^+ x_{k_1}^+ + w^- x_{k_1}^-} + \frac{x_{k_2}^+ x_{k_2}^-}{w^+ x_{k_2}^+ + w^- x_{k_2}^-}$$

És a (8) -at felhasználva:

$$K = \frac{x_{k_1}^+ x_{k_1}^-}{w^+ x_{k_1}^+ + w^- x_{k_1}^-} + \frac{(1 - x_{k_1}^+)(1 - x_{k_1}^-)}{w^+ (1 - x_{k_1}^+) + w^- (1 - x_{k_1}^-)}$$

Az egyenlet első részét átalakítva:

$$\sum_i^S r_i^+ = S^+ = R^+ \qquad \sum_i^S (1 - r_i^+) = S - S^+ = R^-$$

Felhasználjuk hogy:

$$w^+ = \frac{S^+}{S} \qquad w^- = \frac{S - S^+}{S}$$

Az egyenlet nevezőben álló részét. Az $x_{k,i}$ -t a (4) alapján behelyettesítve kapjuk hogy:

$$w^+ x_{k_1}^+ = \frac{S^+}{S} \frac{1}{S} \sum a_i r_i \qquad w^- x_{k_1}^- = \frac{S - S^+}{S} \frac{1}{S - S^+} \sum a_i (1 - r_i)$$

Így az első egyenlet nevezője:

$$w^+ x_{k_1}^+ + w^- x_{k_1}^- = \frac{1}{S} \sum a_i$$

Ekkor a számláló:

$$\frac{1}{S^+} \frac{1}{S - S^+} (\sum a_i r_i)(\sum a_i (1 - r_i))$$

$$K_1 = \frac{S}{S^+ S^-} \frac{(\sum a_i r_i)(\sum a_i (1 - r_i))}{\sum a_i}$$

Alakítsuk át az egyenlet másik felét is:

$$K_2 = \frac{(1 - x_{k_1}^+)(1 - x_{k_1}^-)}{w^+ (1 - x_{k_1}^+) + w^- (1 - x_{k_1}^-)}$$

Az előzőekhez hasonló átalakításokkal kapjuk hogy

$$w^+(1-x^+_{k_1}) = \frac{S^+}{S}\left(1-\frac{1}{S^+}\left(\sum a_i r_i\right)\right)$$
$$w^-(1-x^-_{k_1}) = \frac{S-S^+}{S}\left(1-\frac{1}{S-S^+}\left(\sum a_i(1-r_i)\right)\right)$$

$$= \frac{1}{S}\left(S^+ - \sum a_i r_i\right) + \frac{1}{S}\left(S^- - \sum a_i(1-r_i)\right) = \frac{1}{S}\left(S^+ + S^- - \sum a_i\right) = \frac{1}{S}\left(S - \sum a_i\right)$$

Ekkor az egyenlet második része:

$$K_2 = N\frac{(1-\frac{1}{S^+}\sum a_i r_i)(1-\frac{1}{S^-}\sum a_i(1-r_i))}{S-\sum a_i} = \frac{S}{S^+S^-}\frac{(S^+-\sum a_i r_i)(S^- - \sum a_i(1-r_i))}{S-\sum a_i}$$

És a kétegyenletből a teljes egyenlet:

$$K = \frac{S}{S^+S^-}\frac{(\sum a_i r_i)(\sum a_i(1-r_i))}{\sum a_i} + \frac{(S^+-\sum a_i r_i)(S^- - \sum a_i(1-r_i))}{S-\sum a_i}$$

Vezessük be az alábbi jelöléseket

$$Z = \sum a_i \qquad A = \sum a_i r_i \qquad B = \sum a_i(1-r_i) = Z - A$$

Ezeket behelyettesítve kapjuk hogy:

$$K = \frac{S}{S^+S^-}\left(\frac{AB}{Z} + \frac{(S^+-A)(S^- - B)}{S-Z}\right) \tag{9}$$

Az egyenletnek ezt a feormáját érdemes gyakorlatban használni.

4.4. Példa

Induljunk ki a fent adott példa adatbázisából, és alakítsuk át.

3. táblázat. Átalakított adatbázis

	C_1	C_2	C_3	C_4	C_5	R
1	0	0	0	1	0	1
2	1	0	0	1	1	0
3	1	0	1	0	0	1
4	0	1	0	0	0	0
5	1	1	0	0	0	0
6	1	0	0	1	0	1
7	1	1	0	0	1	0
8	0	0	0	1	1	0
$\sum$	5	3	1	4	3	3

Számoljuk ki a konstansokat.

$$N = 8 \qquad w^+ = \frac{R}{N} = \frac{3}{8} \qquad w^- = \frac{N-R}{N} = \frac{5}{8}$$

Számítsuk ki a K értéket minden jellemzőre.

$$C_1 \quad Z=5 \quad A=2 \quad B=3 \quad K_1=\frac{8}{15}\left(\frac{6}{5}+\frac{(3-2)(5-3)}{8-5}\right)=0.9955$$

$$C_2 \quad Z=3 \quad A=0 \quad B=3 \quad K_2=\frac{8}{15}\left(\frac{0}{5}+\frac{(3-0)(5-3)}{8-3}\right)=0.64$$

$$C_3 \quad Z=1 \quad A=1 \quad B=0 \quad K_3=\frac{8}{15}\left(\frac{0}{1}+\frac{(3-1)(5-0)}{8-1}\right)=0.76$$

$$C_4 \quad Z=4 \quad A=2 \quad B=2 \quad K_4=\frac{8}{15}\left(\frac{4}{4}+\frac{(3-2)(5-2)}{8-4}\right)=0.9333$$

$$C_5 \quad Z=3 \quad A=0 \quad B=3 \quad K_5=\frac{8}{15}\left(\frac{0}{3}+\frac{(3-0)(5-3)}{8-3}\right)=0.64$$

4.5. példa

Így kapjuk:

$$E_D(C_2)=4\left(\frac{3}{8}\frac{5}{8}\right)\frac{1}{1+\frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}}+\frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}}+\frac{1}{1+\frac{3}{8}\frac{1-\frac{0}{5}}{\frac{0}{5}}+\frac{5}{8}\frac{1-\frac{1}{3}}{\frac{1}{3}}}+\frac{1}{1+\frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}}+\frac{5}{8}\frac{1-\frac{2}{3}}{\frac{2}{3}}}=0.5$$

$$E_D(C_3)=4\left(\frac{3}{8}\frac{5}{8}\right)\frac{1}{1+\frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}}+\frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}}+\frac{1}{1+\frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}}+\frac{5}{8}\frac{1-\frac{3}{3}}{\frac{3}{3}}}=0.6$$

5. Az algoritmus kiterjesztése valószínűséggel rendelkező értékekre

5.1. Ha a bemenetek valószínűségek

A valós világban előfordulhatnak valószínűségi bemenetek, a fentiek alapján az l. bemenetre az i. jellemző lehetséges értékei:

$$C_{i,k}^{(l)}=\{C_{i,1}^{(l)},C_{i,2}^{(l)}\dots C_{i,k}^{(l)}\}$$

és nyilvánvaló hogy mivel csak egy értéke lehet egy jellemzőnek egy adott példára ezért:

$$\sum_k C_{i,k}^{(1)}=1$$

Ebből kiindulva a $C_{i,k}^{(1)}$ értékei lehetnek valószínűségek is bináris vektor helyett.
Például legyen a mért hőmérséklet pl 28 fok, és ezt akarjuk 3 érték valamelyikébe:hideg, enyhe, meleg sorolni, ekkor ahelyett hogy azt mondanánk hogy meleg van megadhatjuk valószínűségekként: 0.1, 0.4, 0.5 Az előző jelölésekkel megmutatjuk hogy hogyan kezeljük a valszínűségeket.
Példa 3.:

Adatok:

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
1	0.4	0.6	0.1	0.1	0.8	0.0	1.0	1	0
2	0.6	0.4	0.3	0.3	0.4	1.0	0.0	0	1
3	0.7	0.3	0.0	1.0	0.0	0.0	1.0	1	0
4	0.3	0.7	0.9	0.1	0.0	0.0	1.0	0	1
5	0.8	0.2	0.8	0.2	0.0	0.0	1.0	0	1
6	0.8	0.2	0.2	0.2	0.6	0.0	1.0	1	0
7	0.7	0.3	0.4	0.3	0.3	1.0	0.0	0	1
8	0.1	0.9	0.0	0.0	1.0	1.0	0.0	0	1

Ekkor kapjuk (C_1) -re

$$x_{11}^+ = \frac{1.9}{3} \qquad x_{11}^- = \frac{2.5}{5}$$
$$x_{12}^+ = \frac{1.1}{3} \qquad x_{12}^- = \frac{2.5}{5}$$

Ebből végigszámolva kapjuk:

$$E(C_1) = 4 * \frac{3}{8}\frac{5}{8}\frac{1}{1+\frac{3}{8}\frac{1-\frac{2.5}{5}}{\frac{2.5}{5}}+\frac{5}{8}\frac{1-\frac{1.9}{3}}{\frac{1.9}{3}}} + \frac{1}{1+\frac{3}{8}\frac{1-\frac{2.5}{5}}{\frac{2.5}{5}}+\frac{5}{8}\frac{1-\frac{1.1}{3}}{\frac{1.1}{3}}} = 0.9217$$

A (C_2) jellemzőre

$$x_{21}^+ = \frac{0.3}{3} \qquad x_{21}^- = \frac{2.4}{5}$$
$$x_{22}^+ = \frac{1.3}{3} \qquad x_{22}^- = \frac{0.9}{5}$$
$$x_{23}^+ = \frac{1.4}{3} \qquad x_{23}^- = \frac{1.7}{5}$$

$$E(C_2) = 4*\frac{3}{8}\frac{5}{8}\frac{1}{1+\frac{3}{8}\frac{1-\frac{2.4}{5}}{\frac{2.4}{5}}+\frac{5}{8}\frac{1-\frac{0.3}{3}}{\frac{0.3}{3}}}+\frac{1}{1+\frac{3}{8}\frac{1-\frac{0.9}{5}}{\frac{0.9}{5}}+\frac{5}{8}\frac{1-\frac{1.3}{3}}{\frac{1.3}{3}}}+\frac{1}{1+\frac{3}{8}\frac{1-\frac{1.7}{5}}{\frac{1.7}{5}}+\frac{5}{8}\frac{1-\frac{1.4}{3}}{\frac{1.4}{3}}} = 0.7831$$

És a (C_3) jellemzőre

$$x_{31}^+ = \frac{0}{3} \qquad x_{31}^- = \frac{3}{5}$$
$$x_{32}^+ = \frac{3}{3} \qquad x_{32}^- = \frac{2}{5}$$

$$E(C_3) = 4 * \frac{3}{8}\frac{5}{8}\frac{1}{1+\frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}}+\frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}} + \frac{1}{1+\frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}}+\frac{5}{8}\frac{1-\frac{3}{3}}{\frac{3}{3}}} = \frac{16}{25} = 0.6$$

A C_3 értéke lesz minimális így ez lesz a kiválasztott jellemző.

5.2. A kimenetek valószínűségekkel rendelkeznek

Ha a bemenetek és kimenetek is valószínűségek, akkor azt várjuk el hogy a bizonytalanság nőjön. Az ilyen jellegű be és kimeneteket úgy lehet értelmezni, hogy ha előre tudta egy adott száz fős populáció hogy 30% eséllyel esni fog és 70% eséllyel nem akkor például 10 ember kiment horgászni

90 pedig nem. Ugyanakkor az ilyen jellegű kimenetet lehet zajnak is tekinteni, mert ugyanarra a bemenetre nem ugyanaz a kimenet két jellemzőiben megegyező ember esetén, ha a tízes és a kilencvenes csoportból választjuk őket. Az ilyen zajt nehéz modellezni ismeretlen bemenetekre. Ebben az esetben a kimenet egy vagy nulla lesz, de a modell építésénél figyelembe vesszük. Úgy kezeljük ezt a valószínűséget hogy az eddigiekben használt w^+ és w^- számításakor:

$$w^+ = \frac{\sum_i^N r_i^+}{S}$$
$$w^- = \frac{\sum_i^N r_i^-}{S}$$

Eddig az $x_{k,i}$ k számítása során csak a pozitív vagy negatív példákat használtuk fel a számoláshoz most viszont mivel csak valószínűségek állnak rendelkezésünkre, az összes elemmel kell szoroznunk.

$$x_{k,i}^+ = \frac{r_i^+ C_{k,n}^{(i)}}{w^+}$$
$$x_{k,i}^- = \frac{r_i^- C_{k,n}^{(i)}}{w^-}$$

5.3. ID3 valószínűségekre

Az allgoritmus elvárt kimenete a legvalószínűbb esemény, adott valószínűségekkel megadott jellemzők esetén. Ahhoz hogy ezt elérjük, a döntési fát az ID3 hoz hasonlóan építjük fel:
Kiválasztunk egy minimálisan bizonytalan jellemzőt, majd ennek a jellemzőnek lehetséges diszkrét kimenetein megyünk tovább. A lehetséges diszkrét kimenetek alapján szeparáljuk az eredeti adatbázisunkat. A szeparációt egy az adott jellemzőhöz kiválasztott küszöbérték alapján végezzük el. Ezt a t küszöbértéket úgy kapjuk hogy:

$$t = \frac{1}{n} \tag{10}$$

Ahol n a lehetséges diszkrét értékek száma adott jellemzőre.
Tehát ha adott példán $A = 0.3$ és $B = 0.7$ akkor a példa B útvonalra kerül. Az így szeparált adatbázison ismét bizonytalanságot számolunk és ismételjük a lépéseket.
Az algoritmus akkor áll meg ha már nincs több jellemző, egy eleme van a szeparált halmaznak vagy ha a kimenetek mindegyike nagyobb vagy egyenlő mint egy adott t_2 küszöb érték. Attól függően hogy milyen döntési fát akarunk kapni a t_2 értéket különböző módon választhatjuk meg. Ha a $t_2 = 0.5$ akkor egy egy olyan döntési fához jutunk ahol az alapján soroljuk az elemeket + vagy − osztályokba hogy melyik a valószínűbb. Egy másik koncepció ha a küszöbértéket a következő alapján választjuk:

$$t_2 = w^+ \tag{11}$$

Ez úgy értelmezett hogy akkor pozitív egy példa ha a többi pozitív példa átlagánál nagyobb. Ezzel a koncepcióval egy olyan döntési fához jutunk amelyben azok a példák lesznek pozitívak ahol ennek az esélye a legnagyobb a többihez képest, azaz előnyben részesíti a kiugróan magas értékeket. Ez olyan esetekben hasznos ahol mindenképpen kell választani pozitív példát, ebben az esetben ha az összes r_i^+ kicsi szám akkor ezek várható értékénél nagyobb elemek a legvalószínűbbek, így ezek lesznek a pozitív példák.

5.4. Példa

Az alábbi példán demonstráljuk az algoritmus működését.

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
1	0.4	0.6	0.1	0.1	0.8	0.0	1.0	0.5	0.5
2	0.6	0.4	0.3	0.3	0.4	1.0	0.0	0.3	0.7
3	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.6	0.4
4	0.3	0.7	0.9	0.1	0.0	0.0	1.0	0.1	0.9
5	0.8	0.2	0.8	0.2	0.0	0.0	1.0	0.4	0.6
6	0.8	0.2	0.2	0.2	0.6	0.0	1.0	0.7	0.3
7	0.7	0.3	0.4	0.3	0.3	1.0	0.0	0.2	0.8
8	0.1	0.9	0.0	0.0	1.0	1.0	0.0	0.4	0.6

A táblázatból számolás után kapjuk hogy:

$$R^+ = \sum r_i^+ = 3.2 \qquad R^- = \sum r_i^- = 4.8$$

$$x_{1,1}^+ = \sum \frac{C_{1,1}^{(i)} r_i^+}{R^+} = 0.523 \qquad x_{1,2}^+ = \sum \frac{C_{1,2}^{(i)} r_i^+}{R^+} = 0.477$$

$$x_{1,1}^- = \sum \frac{C_{1,1}^{(i)} r_i^-}{R^-} = 0.591 \qquad x_{1,2}^- = \sum \frac{C_{1,2}^{(i)} r_i^-}{R^-} = 0.409$$

$$x_{2,1}^+ = \sum \frac{C_{2,1}^{(i)} r_i^+}{R^+} = 0.402 \qquad x_{2,2}^+ = \sum \frac{C_{2,2}^{(i)} r_i^+}{R^+} = 0.243 \qquad x_{2,3}^+ = \sum \frac{C_{2,3}^{(i)} r_i^+}{R^+} = 0.354$$

$$x_{2,1}^- = \sum \frac{C_{2,1}^{(i)} r_i^-}{R^-} = 0.240 \qquad x_{2,2}^- = \sum \frac{C_{2,2}^{(i)} r_i^-}{R^-} = 0.322 \qquad x_{2,3}^- = \sum \frac{C_{2,3}^{(i)} r_i^-}{R^-} = 0.437$$

$$x_{3,1}^+ = \sum \frac{C_{3,1}^{(i)} r_i^+}{R^+} = 0.437 \qquad x_{3,2}^+ = \sum \frac{C_{3,2}^{(i)} r_i^+}{R^+} = 0.562$$

$$x_{3,1}^- = \sum \frac{C_{3,1}^{(i)} r_i^-}{R^-} = 0.281 \qquad x_{3,2}^- = \sum \frac{C_{3,2}^{(i)} r_i^-}{R^-} = 0.718$$

Ekkor a bizonytalanság képletébe behelyettesítve kapjuk a következő értékeket.

$$E_D(C_1) = 0.95570 \qquad E_D(C_2) = 0.9315 \qquad E_D(C_3) = 0.9350$$

Ez alapján tehát kiválasztjuk a C_2 -t az első csúcspontnak, majd meghatározzuk a t és t_2 küszöbértékeket. A t_2 értéket válasszuk 0.5 -nek. A t értéke ebben az esetben $t = 0.33$ mert 10. Ekkor az eredeti adatbázisunk három részre bomlik:
$C_2 = 1$ rész:

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
4	0.3	0.7	0.9	0.1	0.0	0.0	1.0	0.1	0.9
5	0.8	0.2	0.8	0.2	0.0	0.0	1.0	0.4	0.6
7	0.7	0.3	0.4	0.3	0.3	1.0	0.0	0.2	0.8

$C_2 = 2$ rész:

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
3	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.6	0.4

$C_2 = 3$ rész:

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
1	0.4	0.6	0.1	0.1	0.8	0.0	1.0	0.5	0.5
2	0.6	0.4	0.3	0.3	0.4	1.0	0.0	0.3	0.7
6	0.8	0.2	0.2	0.2	0.6	0.0	1.0	0.7	0.3
8	0.1	0.9	0.0	0.0	1.0	1.0	0.0	0.4	0.6

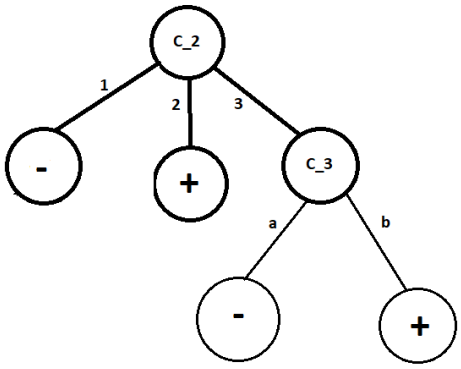
A választott t_2 értékünk alapján tiszták a $C_2 = 1$ és $C_2 = 2$ osztályok így ezekből leveleket képezünk. A C_3 osztály tovább bomlik. A kapott bizonytalanságok $C_1 = 0.9973$ és $C_3 = 0.9349$, választjuk a C_3. A két új adatbázis:
$C_3 = a$ rész:

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
2	0.6	0.4	0.3	0.3	0.4	1.0	0.0	0.3	0.7
8	0.1	0.9	0.0	0.0	1.0	1.0	0.0	0.4	0.6

$C_3 = b$ rész:

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
1	0.4	0.6	0.1	0.1	0.8	0.0	1.0	0.5	0.5
6	0.8	0.2	0.2	0.2	0.6	0.0	1.0	0.7	0.3

Az generált döntési fa a következőképpen áll elő:



1. ábra. Az adatokon képzett döntési fa

6. Folytonos értékek kezelése

Folytonos terekben a döntési fákat a gyakorlatban általában valamilyen küszöbérték alapján építik fel[Qui87][CL]. Ez azonban a valós döntési felületek alakjától függően rendkívül nagy méretű döntési fákhoz vezethet, a döntési felület komplexitásától függően. Erre a problémára alternatív megoldásokat adtunk, amelyek célja ennek a modell hatékonyabbá tétele. Az alábbiakban tárgyalt esetekben a döntési fa csúcsai az általunk meghatározott halmazok lesznek, és ha egy halmaz teljesen homogén, vagy ha a benne lévő bizonytalanság kisebb az általunk megadott küszöbértéknél, akkor döntésthozunk.

7. Szeparálás Körökkel

A döntési felület szeparálható körökkel, erre két különböző függvényt adunk és hasonlítunk össze.

7.1. távolság alapú függvény

Induljunk ki az alábbi függvényből:

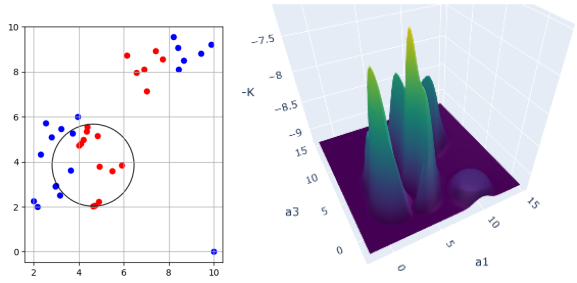
$$g(x, y, x_0, y_0, r, \lambda) = \frac{1}{1 + e^{-\lambda * \frac{(x-x_0)^2+(y-y_0)^2}{r^2}}} \tag{12}$$

Ahol az x_0 és y_0 az optimalizálandó paraméter r a rádiusz és λ a függvény hiperparamétere. Ha $(x, y, x_0, y_0, r, \lambda)$ -t behelyettesítjük a (9) egyenletben az a_i helyére akkor kapjuk hogy

$$K(G, R, N) = \frac{(\sum r_i g_i)((\sum g_i) - (\sum r_i g_i))}{\sum g_i} + \frac{(\sum r_i^+ - \sum r_i g_i)((\sum r_i^- - ((\sum g_i) - (\sum r_i g_i))}{N - \sum g_i}$$

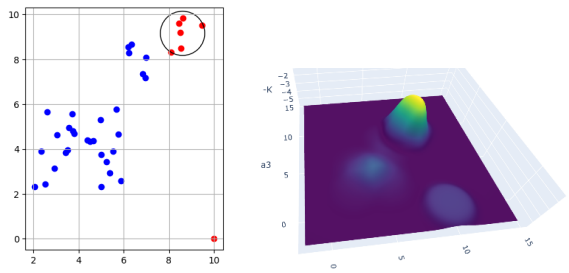
Ez lesz az optimalizálandó függvény. Feltesszük hogy r előre adott paraméter, ekkor optimalizálásnál keressük adott rádiusz mellett a leghomogénebb ponthalmazokat.

Ha egy példán szemléltetjük akkor a következő döntési feleületet kapjuk adot pontokra:



2. ábra. A döntési felület adott pontokra r=2 és λ=1 kezdőértékekre

A függvénnyel viszont a probléma hogy a döntési felületent mindig a legmagasabban a bizonytalanságot legjobban csökkentő klaszterek lesznek. Ez azt fogja eredményezni hogy ha a kiindulásnál egyenlő volt a két halmaz elemeinek száma és választottunk mondjuk egy pirosakat tartalmazó kört akkor ha a szekvenciális lefedés algoritmusával megyünk tovább akkor a következő választásunk is piros lesz. így az algoritmus addig megy amég ki nem választja az összes piros pontot. Erre példa az alábbi ábra:



3. ábra. A döntési felület biasa r=2 és λ=1 kezdőértékekre

Mint láthatjuk a függvény a két piros ponthalmazt priorizálja. Ezzel a módszerrel egy mély fához jutunk és ha zajosak az adataink akkor a szükségesnél több iteráció kellhet hozzá.

7.2. Lefedettség alapú függvény

A körök középpontjainak optimalizálásához az egyik legegyszerűbb megközelítés ha megszámoljuk az adott középpont által fix r mellett a lefedett pontok számát. Mivel tudjuk hogy az osztályozásunk bináris ezért módosíthatjuk az osztálycímkéinket hogy könnyebb legyen velük számolni.

$$r_i^{'} = \begin{cases} 1 & \text{ha } r_i = 1 \\ -1 & \text{ha } r_i = -1 \end{cases}$$

A pontokon amik már az átalakított címkével rendelkeznek végigiterálunk és a függvény értékét a következőképpen határozzuk meg.

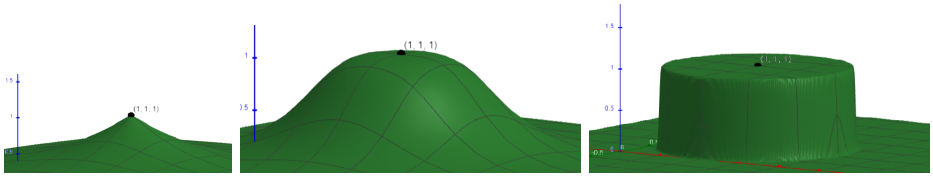
$$H(x, y, x_0, y_0, r) = \begin{cases} \sum_{i=1}^{n} r_i & \text{ha } \sqrt{(x - x_0)^2 + (y - y_0)^2} < r \\ 0 & \text{különben} \end{cases} \tag{13}$$

Tehát összegezzük az adott középpontú kör által lefedett pontokat. Ha ezeket önmagukban felhasználnánk akkor egy olyan döntési felülethez jutnánk ahol sok sík és lokális optimum van, egy

ilyen felületet nehéz optimalizálni. Azért hogy ezen a felületen javítsunk egy olyan függvényre van szükségünk ami a kapott középponttól az r függvényében folyamatosan csökken. Egy ilyen függvény az alábbi[Józ06]:

$$D(x_0, y_0, r, \lambda) = \frac{1}{1 + \left(\frac{(x-x_0)^2+(y-y_0)^2}{r^2}\right)^{\lambda}} \tag{14}$$

A függvény képe ekkor különböző λ értékekkel a következő: A felületünket tehát az alábbi függvény

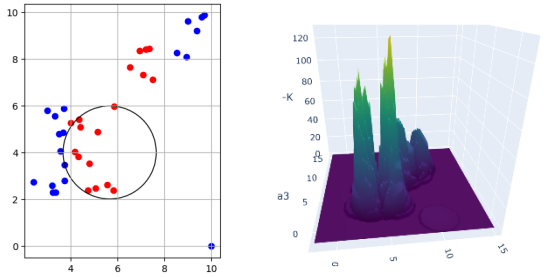


4. ábra. A (14)függvény képe (1, 1) középponttal r=1 sugárral $\lambda = 0.6, 2, 100$ értékekre

adja:

$$K(x, y, x_0, y_0, r, \lambda) = |H(x, y, x_y, y_0, r)| * D(x_y, y_0, r, \lambda) \tag{15}$$

Az eljárás minden pontra kiszámolja ennek a kifejezésnek az értékét majd a döntési felület egy pontjának mehatározásához vesszük ezek összegét fix a_1 és a_3 mellett. Ha egy az előzőekhöz hasonló példát generálunk akkor a döntési felületünk a következő:



5. ábra. A 2. függvénnyel képzett döntési felület $\lambda = 2$ értékkel

7.3. felület optimalizálása

Mivel tudjuk hogy a kör optimális középpontja közel van a pontjainkhoz ezért az optimum megtalálásához 9 különböző helyről indítottunk csökkenő gradiens módszert, a maximum és minimum x és y értékek alapján egy négyszög sarkairól, és az oldalfelezőkről, valamint a középpontból. A jelenlegi függvényeinkel mé viszont előfordulhatnak problémák mert ha a függvényeink által adott felületet megvizsgáljuk a (5) és (3) ábrákon akkor azt figyelhetjük meg hogy a gradiens 0 a pontok közvetlen környezetén kívül. Ezt gyakorlatban azzal oldottuk meg hogy az első függvényből levontuk az alábbi függvényt a másodikhoz pedig hozzáadtuk.

$$d(x, y, x_0, y_0, r) \frac{1}{1 + e^{\sqrt{\frac{(x-x_0)^2+(y-y_0)^2}{r^2}}}}$$

Azért hogy csak a gradienshez legyen releváns és ne módosítsa a már létező felületünket drasztikusan csak egy kis súllyal adtuk hozzá az előző függvényeinkhez. A függvény abból következik hogy A felületen nehéz globális optimumot találni de nekünk elég egy jó lokális optimum is. A talált

lokális optimumot a következőképpen javítottuk:

1. A kör sugarát addig csökkentettük amég a többségben lévő pontok legtávolabbikát el nem éri Ezzel egyfajta szűrést is alkalmazva rajta amellett hogy az eddigi tetszőleges r értékét módosítottuk.

2. A kör középpontját módosítjuk a többségben lévő pontok középpontjára majd az első lépéshez hasonlóan csökkentjük a sugarat.

3. Előfordulhat hogy a 2. lépés után rosszabb körünk van, ekkor visszaállítjuk a kört az első lépés utáni állapotára.

7.4. Faépítés körökkel

A fát a körök kunstruálásával egy időben építjük, minden új kör egy új csúcs lesz a fában. Az algoritmus a következőképpen működik:

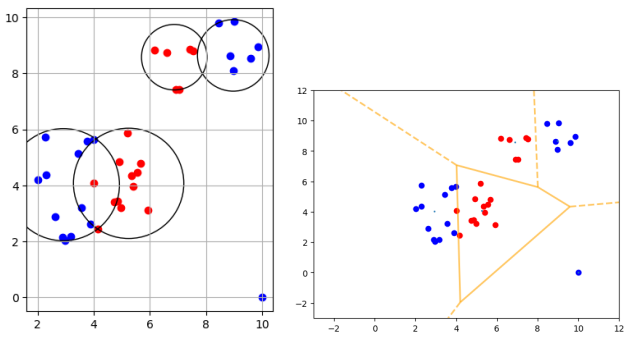
1. Egy kör legenerálásakor megvizsgáljuk a bizonytalanságát és ha egy általunk meghatározott küszöb érték alatt van akkor döntést hozunk rá. folytatjuk 3. lépéssel
Ha a küszöbérték felett van 2. lépés.

2.Az adott kör fabeli csúcspontjából kiindulva új gyerekeket hozunk létre, létrehozási sorrendben futtatjuk rájuk az 1. lépést.

3. új kör hozzáfűzése a fához a generálásban őt megelőző nodejához majd 1. lépés

8. Szeparálás polinomokkal

A kör alapú módszerrel az a probléma hogy a körökön kívüli régióna nincsen értéke. Ahhoz hogy értéket adjunk ennek az üres régiónak a Voronoi diagrammot foglyuk használni. A módszer a körökhöz legközelebb eső pontokhoz a kör címkéit rendeli. A probléma a módszerrel viszont az

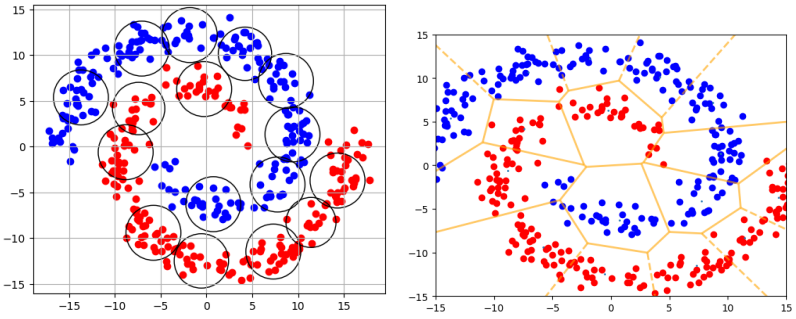


6. ábra. Adott körök és a rajta képzett Voronoi diagram.

hogy nem kezel különböző méretű köröket mert csak a körök középpontjait veszi figyelembe a diagram generálásakor. Erre a prbolémára kétféle megoldást adhatunk, Vagy egységes méretű körökkel számolunk csak vagy módosíthatjuk a voronoi eljárást.

8.1. Szeparálás régiókkal

A célunk egy olyan algoritmus kidolgozása volt amely képes a kör alapú eljárásnál kevesebb lépésből megadni egy jó döntési határt. Vegyük az alábbi példát és vizsgáljuk meg.



7. ábra. A körök a spiral adatbázison, és a belőlük képzett voronoi diagram

Ha megvizsgáljuk ezt a diagramot akkor láthatjuk hogy néhány esetet leszámítva a voronoi diagram jól szeparál, viszont sok iterációt igényel. Ahoz hogy csökkentsük ezt az iterációszámot olyan köröket kell találnunk amik segítik a voronoi által adott döntési határ jobbátételét. Ezeket a következő képpen kerestük:

1. Menjen a körképzés amég nem talál legalább egy olyan kört amiben az eddigiektől eltérő a többségben lévő szín majd 2.

2. Képezzük a jelenlegi körök Voronoi diagramját majd 3

3. számoljuk ki a bizonytalanságot az egyes régiókon belül, ahol a küszöbérték feletti ott 4., ha alatta van nincs tehendőnk

4. A régióban keressünk köröket majd 2.

A 3. lépés esetén ha módosítjuk a voronoi eljárást hogy a körök méreteit is figyelembe vegye akkor a döntési határok különböző sugarú körök esetén hiperbolák lesznek de ezekkel költségesebb a régiókon belüli pontok meghatározása. Az ilyen Voronai diagram(súlyozott Voronai diagram [AB86]) kirajzolásához a pontok távolságát kell meghatározni az egyes körröktől és a ponthoz legközelebbi kör régiójába fog tartozni.

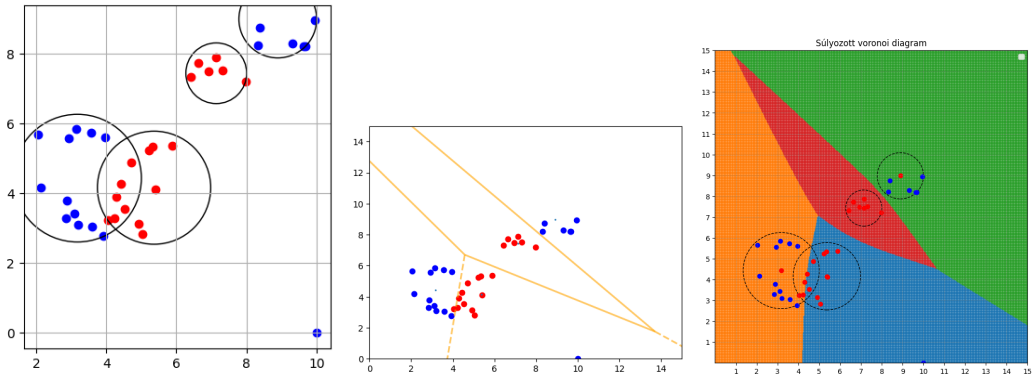
Erre az eljárásra azért van szükség mert a döntési felület konstruálása során nem vesszük figyelembe a szekvenciális lefedés kimenetét csak mohó módon az éppen legjobbnak tűnő középpontot választjuk.

$$argmin(\sqrt{(P_x - x_0)^2 + (P_y - y_0)^2} - r_0) \tag{16}$$

Ennek a minimumát keressük a $P_{x,y}$ pontra úgy hogy végigiterálunk az összes kör középponton. Ha a (9) függvényéből kivesszük az r_0 sugarat akkor a súlyozatlan Voronoi diagrammot kapjuk.

8.2. Fa építése voronai és kör alapú módszer alapján

A vizsgált (9) alapján láthatjuk hogy a diagram rosszab döntési felületet adott mint a körök, ennek az az oka hogy a diagram építése során nem vesszük figyelembe a körök létrehozásának sorrendjét, ami azért fontos mert ha a középső kékeket tartalmazó kört hoztuk volna létre először akkor más



8. ábra. Különböző méretű körök és a súlyozottés súlyozatlan voronoi diagramjuk

középponttal találtuk volna meg, mert az eredeti verzióban a szekvenciális lefedés hatására kivontuk a piros pontokat a keresésből mikor konstruáltuk a kékeket. Másrészről a voronoi diagram egy sokkal általánosabb döntési felületet eredményezett mint a körös lefedés. A célunk tehát valamiféle sorrendiséget bevezetni és ebből egy döntési fát konstruálni. Erre a megoldásunk az hogy a már megkonstruált köröket együttesen kezeljük voronoi által adott döntési határokkal úgy hogy összekombináljuk az első két algoritmust és kiegészítjük a faépítését:

1. Egy kör legenerálásakor megvizsgáljuk a bizonytalanságát és ha egy általunk meghatározott küszöb érték alatt van akkor döntést hozunk rá 4.
Ha a küszöbérték felett van 3. lépés.

2. vizsgáljuk a körhöz(csak ha nem más körön belül van) tartozó voronoi régiót ha a bizonytalansága küszöb érték alatt van akkor döntést hozunk rá 4. ha nincs akkor 3.

3. Adott régióban/körben keresünk új köröket kisebb kezdeti rádiusszal majd 2.

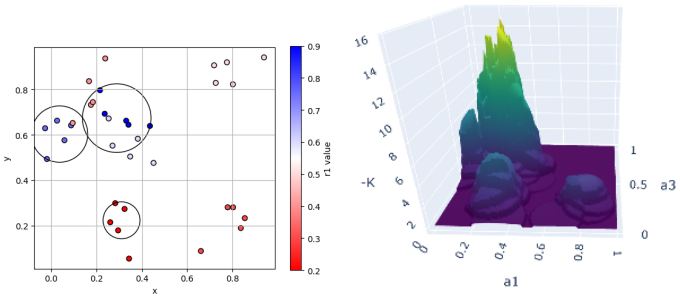
4. új kör/régió hozzáfűzése a fához, ha a egy régiót hozzáfűzünk a fához akkor a körökhöz hasonlóan ennek a pontjai is levonódnak a következőleg vizsgáltakból.

Ennek az algoritmusnak az az előnye azzal hogy külön legenerálnánk a fát majd a rrégiókat hogy a már korábban tárgyalt módon keres köröket a Voronai régiókon belül ezért összesen kevesebb kört keresünk.

9. Valószínűségekkel rendelkező értékek kezelése folytonos térben

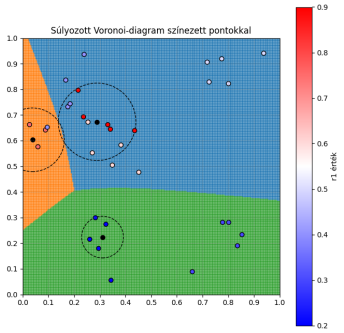
Már korábban bevzettük a valószínűségekkel rendelkező inputokra a döntési fát, de vizsgáljuk meg hogy ha a valószínűségi teret tekintjük döntési térnek milyen eredményhez jutunk. Az ötlet az hogy a folytonos térben jobban szeparálhatóak a döntések mert kevésbé hajlamos a túltanulásra és kevesebb körrel szeparálhatóak az adatok mert ki tudjuk használni a több dimenziós tér által nyújtott előnyöket. A módszer olyan klasztereket keres az előzőekben bemutatott eljárásokkal amikre az egyik címke maximális. A döntési teret úgy definiáljuk hogy minden jellemző egy dimenziót jelent, ez megadható minden esetben mert az eddig ismertetett módon megmutattuk hogy tetszőleges

lehetséges értékű jellemző szétbontható kétértékű jellemzőkre. A pontok osztályát pedig a $[0, 1]$ -ről$[-0.5, 0.5]$ -re képezzük és ezek összegét helyettesítjük be H helyére a (15). egyenletbe.



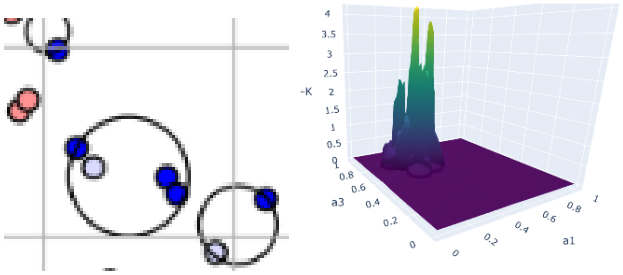
9. ábra. Az eljárás kimenete valószínűségekre 3 iteráció után

Erre a halmazra a következő Voronoi diagrammot kapjuk. Mint láthatjuk a Voronoi diagrammal



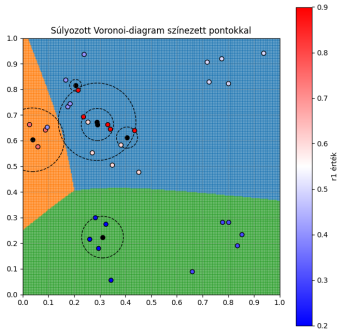
10. ábra. A voronoi diagram a 3. iteráció után

3 kör után le tudtuk szűkíteni a teret a kék régióra mert itt nincsen még jó szeparálásunk, így a következőekben ezt vizsgáljuk. Az előző szekcióban bevezetett algoritmus alapján a következő lépésben a kör bizonytalanságának csökkentése.



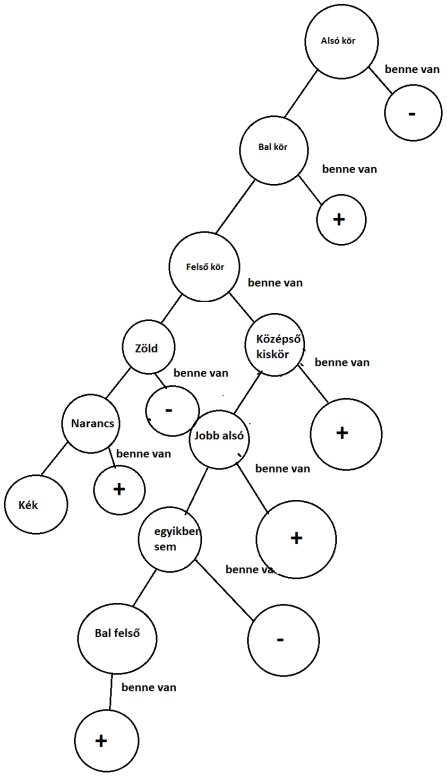
11. ábra. A kör felbontása

A felbontott körbeli 3 kört most a döntési fához adhatjuk a nagyobb kör gyerekeként. A kék



12. ábra. A voronoi diagram a 3. lépés után

régión belüli fehér pontokat nem kell külön kategorizálnunk mert ezek olyan pontok amiknek a bizonytalansága magas, azaz 0.5 körül van a kimenet valószínűsége. Az algoritmus továbbfejleszthető ha az így alkotott klaszterközéppontok alapján a szülőt is régiókra bontanánk, de csak az adott körön belül. A példánkon a pozitív elemek a pirosak voltak, ez alapján a döntési fánk a következőképpen áll elő.



13. ábra. Az előállt döntési fa

10. Eljárás magasab dimenziós térben

A gyakorlatban általában több mint két dimenziós terekkel dolgozunk, ezért megvizsgáljuk hogy az eljárásainkk hogyan viselkednek egy ilyen térben. Ha a vizsgáljuk a függvényeinket (12,14) akkor

észrevehetjük hogy mivel mind a két függvény az euklédeszi távolságot használja ezért ezekhez hozzá kell venni az új dimenziókat. A felület optimalizálási és faépítési algritmusok is működnek magasabb dimenzióban. A Voronoi diagramm által megadott módszerrel lehetnek problémáink attól függően hogy milyen algoritmussal számítjuk, mert például a népszerű Pásztázó egyenes algoritmus([For87]) csak 2D -ben működik. A legegyszerűbb megközelítésben használhatjuk a (9) képletet. Az általunk adott fa építő algoritmus szintén működik magasabb dimenzióban.

11. Konklúzió

A dolgozatban sikeresen átalakítottuk a bizonytalanság mértéket, és ezt felhasználva tudtuk kezelni a valószínűség alapú be és kimeneteket. A dolgozat elődjéhez hasonlóan nagy fókuszt fektet a folytonos terek szeparálására, erre adtunk alternatívát, és kitárgyaltuk a különböző függvények előnyét és hátrányát szeparáló kritériumként. Javítottuk az optimum megtalálásának esélyeit és konstruáltunk olyan faépítési algoritmust ami kevesebb lépésben képes megatlálni a döntési határokat. Megmutattuk hogy az eljárás alkalmazható valószínűségekre is. Az algoritmus pontosságát lehetne javítani ha az összes lehetséges sugárra is vizsgáljuk a középpontok által alkotott optimalizálandó felületet, de ez nagyon erőforrás igényes. Erre alternatíva ha a talált középponttól iteratívan változtatjuk a sugár, majd középpontokat. Ebben az esetben az algoritmus akkor áll meg ha az $i+1$. lépésben kapott sugár és középpont megegyezik az i. lépésben kapottal.

Hivatkozások

[HR76] L. Hyafil és R. Rivest. “Constructing optimal binary decision trees is np-complete”. *Information Processing Letters* 5 (1976).

[Bre+84] L. Breiman és tsai. *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[AB86] Peter F. Ash és Ethan D. Bolker. “Generalized Dirichlet tessellations”. *Geometriae Dedicata* 20 (1986), 209–243. old.

[Qui86] J. R. Quinlan. “Induction of decision trees”. *Machine Learning* (1986).

[For87] Steven Fortune. “A sweepline algorithm for Voronoi diagrams”. *Algorithmica* (1987).

[Qui87] J. R. Quinlan. “Simplifying decision trees”. *International Journal of Man-Machine Studies* (1987).

[Vet00] R. Vetschera. “Entropy and the value of information”. *Central European Journal of Operations Research* 8 (2000).

[J D01] Á. Zsiros J. Dombi. “Learning Decision Trees in Continuous Space”. *Acta Cybernetica* 15 (2001), 213–224. old.

[Józ06] József D. Dombi József Dombi. “Dynamic System Using Conjunctive Operator”. *Acta Polytechnica Hungarica* 1 (2006).

[CL] K. J. Cios és N. Liu. “A machine learning method for generation of a neural network achitecture: a continuous ID3 algorithm”. 3, no. 2, pp. 280-291, March 1992 ().