Algoritmusok fuzzy-összefüggőségre

Németh Gábor

2009. május 10.

Színtér

- $\mu_\alpha(c,d)$: egy reflexív, szimmetrikus reláció $\mathbb{Z}^2$-n
- színtér: $\mathcal{C} : (C, f)$, ahol $C \subset \mathbb{Z}^2$ és $f : C \to [L, H]$
  - C: ponthalmaz
  - f: képfüggvény

Hasonlóság, az összefüggőség erőssége

- Hasonlóság: $\mu_\kappa(c,d) = h(\mu_\alpha(c,d), f(c), f(d), c, d)$
- Fuzzy κ-háló: Legyen P_{cd} a $c, d \in \mathcal{C}$ közötti utak halmaza. Az $\mathcal{N}_\kappa$ fuzzy κ–háló a P_{cd} olyan részhalmaza, amelyben a halmazhoz tartozást (az összefüggőség erősségét) bármely $p_{cd} \in P_{cd}$ útra az úton lévő legkisebb affinitással definiáljuk.

$$\mu_{\mathcal{N}_\kappa}(p_{cd}) = \min_{j=1,\ldots,m-1} \mu_\kappa(c_j, c_{j+1})$$

- A fuzzy κ-összefüggőség $\mathcal{C}(K)$:

$$\mu_K(c,d) = \max_{p_{cd} \in P_{cd}} \mu_{\mathcal{N}_\kappa}(p_{cd})$$

Fuzzy κ_θ komponens

- $\theta \in [0,1]$ egy küszöbérték
- K_θ egy ekvivalencia reláció a C halmazon a következő képpen:

$$\mu_{K_\theta}(c,d) = \begin{cases} 1 & \text{ha } \mu_\kappa(c,d) \geq \theta \\ 0 & \text{különben} \end{cases}$$

- $O_\theta(o)$ egy K_θ osztály, amely tartalmazza az $o \in C$ elemet
- $\Omega_\theta(o) = \{c \in C | \mu_K(o,c) \geq \theta\}$
- a Fuzzy-összefüggőség részleges számítása kifejezhető a kövező azonossággal

$$O_\theta(o) = \Omega_\theta(o)$$

Fuzzy-összefüggő objektum

Az o-t tartalamazó $\mathcal{C}$ színtér fuzzy κ_θ objektumát $\mathcal{O}_\theta(o)$-val jelöljük.

$$\mu_{\mathcal{O}_\theta}(c) = \begin{cases} \eta(c) & \text{ha } c \in O_\theta(o) \\ 0 & \text{különben} \end{cases}$$

$$\mu_{\mathcal{O}_\theta}(c) = \begin{cases} \eta(c) & \text{ha } c \in \Omega_\theta(o) \\ 0 & \text{különben} \end{cases}$$

ahol η megfelel egy objektumhoz tartozásnak minden képelem esetén úgy, hogy $f(c)$ és $\mu_K(o, c)$.

Visszavezetés gráf-keresési problémára

- A képelemek megfelelnek a gráf csúcsainak.
- A térbeli elemek oldalai a gráf élei.
- A fuzzy képelem-affinitás az élek költségei a gráfban.
- A fuzzy-összefüggőség a páronkénti legrövidebb út a gráfban.
- A fuzzy-összefüggő objektumok az összefüggő gráfkomponensek.

Dinamikus programozás (*label-correcting*)

Input: $\mathcal{C}$, $o \in C$, κ
Output: Egy K-összefüggő $\mathcal{C}_o = (C_o, f_o)$ színtere a $\mathcal{C}$
Segédstruktúra: egy Q sor a pontok számára

állítsuk $\mathcal{C}_o$ minden elemét 0-ra, kivéve o-t, amit 1-re állítunk
rakjunk be Q-ba minden olyan $c \in C_o$ elemet, amelyre $\mu_\kappa(o, c) > 0$
while $Q \neq \emptyset$ **do**
  vegyünk ki egy c elemet a Q-ból
  $f_{val} \leftarrow \max_{d \in C_o}[min(f_o(d), \mu_\kappa(c, d))]$
  if $f_{val} > f_o(c)$ **then**
    $f_o(c) \leftarrow f_{val}$
    rakjunk be Q-ba minden olyan e pontot, amelyre $\mu_\kappa(c, e) > 0$,
    $f_{val} > f_o(e)$, $\mu_\kappa(c, e) > f_o(e)$
  end if
end while

Algoritmusok csoportosítása

- Interakció szerint
  - Félautomatikus: minimális vagy nagyobb interakció
    - Seed pontok
    - befoglaló téglatest
  - Automatikus
- Cél szerinti csoportosítás
  - Egy fuzzy-összefüggő objektum meghatározása
    - Csak egy bizonyos objektumot szeretnénk meghatározni
  - Több objektumot szeretnénk meghatározni, ezeket külön címkével látjuk el.
    - Minden képelemet soroljunk be egy objektumba.

Fuzzy affinitás komponensei

- **Fuzzy képelem összefüggőség**: $\mu_\alpha(c, d)$ jelöli a képelemek térbeli összefüggőségének fokát.
- **Homogenitás-alapú komponens**: $\mu_\psi(c, d)$ jelöli a képelemek lokális együttes-függőségét az intenzitásuk hasonlóságának alapján.
- **Objektum-tulajdonság-alapú komponens**: $\mu_\phi(c, d)$ jelöli a képelemek lokális együttes-függőségét valamilyen objektum-tulajdonság alapján.

- Nem-skála alapú
- Skála alapú
- Vektoros skála alapú

Nem-skála alapú fuzzy képelem affinitás

A c és d pontok között az affinitás erősségét befolyásolhatja, hogy

- milyen közel vannak egymáshoz térben
- mennyire hasonló az intenzitásuk
- mi az aktuális helyzetük a színtérben

$$\mu_\kappa(c,d) = w_1 g_1(f(c), f(d)) + w_2 g_2(f(c), f(d))$$

ahol

- $w_1, w_2 \geq 0$
- g_1 a $\frac{f(c)+f(d)}{2}$ Gauss-függvénye, g_2 a $\frac{|f(c)-f(d)|}{2}$ Gauss-függvénye
- g_2 várható értékke 0,
- g_1 szórása és várható értéke az objektumétól függ

Fuzzy képelem affinitás

$$\mu_\kappa(c,d) = \mu_\alpha(c,d) g(\mu_\psi(c,d), \mu_\phi(c,d))$$

g tulajdonságai:

- az értékkészlete [0,1] közé esik
- mindkét argumentumában nem-növekvő

Példa:

$$\mu_\kappa = \frac{1}{2}\mu_\alpha(\mu_\psi + \mu_\phi)$$
$$\mu_\kappa = \mu_\alpha(\sqrt{\mu_\psi \cdot \mu_\phi})$$

Homogenitás alapú komponens

$$\mu_\psi(c,d) = W_\psi(|f(c) - f(d)|)$$

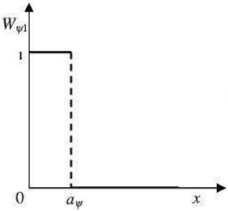
W_ψ tulajdonságai:

- az értékkészlet [0,1] közé esik és $W_\psi(0) = 1$
- monoton nemnövekvő
- minden homogenitásra meg kell, hogy feleljen

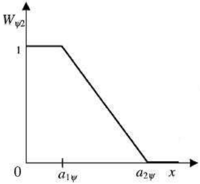
Példa:

jobb oldali része egy megfelelő trapezoid vagy Gauss-függvény

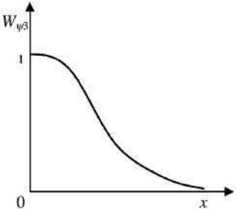
Homogenitás alapú komponens



$$W_{\psi 1}(x)=\begin{cases}1, & 0 \le x \le a_\psi,\\ 0, & x > a_\psi.\end{cases}$$



$$W_{\psi 2}(x)=\begin{cases}1, & 0 \le x \le a_{1\psi},\\ \dfrac{a_{2\psi}-x}{a_{2\psi}-a_{1\psi}}, & a_{1\psi} \le x \le a_{2\psi},\\ 0, & x > a_{2\psi}.\end{cases}$$



$$W_{\psi 3}(x)=e^{-\frac{x^2}{2k_\psi^2}}, \quad k_\psi > 0.$$

Objektum-tulajdonság alapú komponens

$$\mu_\phi(c,d) = \begin{cases} 1 & \text{ha } c = d \\ \frac{\mathcal{W}_o(c,d)}{\mathcal{W}_b(c,d)+\mathcal{W}_o(c,d)} & \text{különben} \end{cases}$$

$$\mathcal{W}_o(c,d) = min[W_o(f(c)), W_o(f(d))]$$

$$\mathcal{W}_b(c,d) = max[W_b(f(c)), W_b(f(d))]$$

A $\mu_\phi(c,d)$ mérték azt mutatja meg, hogy mennyire tartozik az o pont osztályába s mennyire más osztályba.

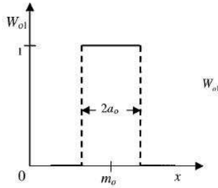
W_o és W_b tulajdonságai:

- az értékkészlet [0,1] közé esik
- monoton nemnövekvő

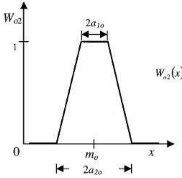
Példa:

eltolt trapezoid vagy Gauss-függvény

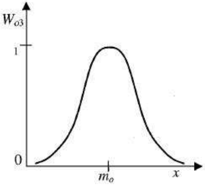
Objektum-tulajdonság alapú komponens



$$W_{o1}(x)=\begin{cases}0, & x<m_o-a_o,\\ 1, & m_o-a_o\le x\le m_o+a_o,\\ 0, & m_o+a_o<x.\end{cases}$$



$$W_{o2}(x)=\begin{cases}0, & x<m_o-a_{2o},\\ \dfrac{x-(m_o-a_{2o})}{a_{2o}-a_{1o}}, & m_o-a_{2o}\le x\le m_o-a_{1o},\\ 1, & m_o-a_{1o}\le x\le m_o+a_{1o},\\ \dfrac{(m_o+a_{2o})-x}{a_{2o}-a_{1o}}, & m_o+a_{1o}\le x\le m_o+a_{2o},\\ 0, & m_o+a_{2o}<x.\end{cases}$$



$$W_{o3}(x)=e^{-\frac{(x-m_o)^2}{2k_o^2}},\quad k_o>0.$$

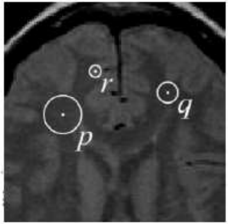
Skála alapú affinitás

Meg kell felelnie az alábbi követelményeknek:

- térbeli összefüggőség
- homogenitás (lokális és globális)
- objektum tulajdonság (intenzitás alapú)
- objektum skála

Objektum skála

Az **objektum skála** egy $\mathcal{C}$ szńtérben valamely $c \in C$ középpontú, $r(c)$ sugarú hipergömb, amely ugyanahhoz az objektum-régióhoz tartozó területet tartalmazza.



A skála méret egyszerűen és hatékonyan, szegmentálas nélkül meghatározható. Egy c középponthoz tartozó hipergömböt addig növelünk, ameddig meg tudjuk őrozni a homogenitási kritériumot.

Az objektum skála kiszámítása

Input:$\mathcal{C}$, $c \in C$, $r \in [0,1]$
Output:$r(c)$ **begin**

$k \leftarrow 1$
while $FO_k(c) \geq \tau$ **do**
$k \leftarrow k + 1$
end while
$r(c) \leftarrow k$

end

ahol $FO_k(c)$-t a c középpontú, k sugarú gömbök adják:

$$FO_k(c) = \frac{\sum_{d \in B_k(c)} W_{\psi_s}(|f(c) - f(d)|)}{|B_k(c) - B_k(c)|}$$

Szomszédság kiválasztása a skála alapú számításhoz

Egyetlen c képelem alapján

$$B_r(c) = \{e \in C \mid ||e - c|| \leq r(c)\}$$

Egy c és egy d képelem-pár alapján

$$B_{cd}(c) = \{e \in C \mid ||e - c|| \leq \min[r(c), r(d)]\}$$

$$B_{cd}(d) = \{e \in C \mid ||e - d|| \leq \min[r(c), r(d)]\}$$

Intenzitáskülönbség a szomszédságokban

A c és d pontok körül bizonyos inhomogenitás lehetséges ($\mu_\alpha(c, d) > 0$), aminek két oka lehet

- objektumon belüli (*intra-object*) szórás: véletlenszerű, 0 középértékkel
- objektumok közötti (*inter-object*): valamilyen irány is felfedezhető, nagyobb, mint az objektumon belüli (pontpárokat hasonlítunk össze)

Homogenitás alapú komponens

Az irány szerinti inhomogenitás c és d körül kiszámolható:

$$D(c,d) = \sum_{\substack{e \in B_{cd}(c) \\ e' \in B_{cd}(d) \\ e-c=e'-d}} (1 - W_h(f(e) - f(e')))\omega_{cd}(||e-d||)\frac{f(e) - f(e')}{|f(e) - f(e')|}$$

Homogenitás alapú komponens:

$$\mu_{\psi_s}(c,d) = 1 - \frac{|D(c,d)|}{\sum_{e \in B_{cd}(c)} \omega_{cd}(||e-c||)}$$

Objektum-tulajdonság alapú komponens

Egy szürt változatát vesszük az $f(c)$ környezetének

$$f_a(c) = \frac{\sum_{e \in B_r(c)} f(e)\omega_c(||e-c||)}{\sum_{e \in B_r(c)} \omega_c(||e-c||)}$$

Objektum-tulajdonság alapú komponens:

$$\mu_{\phi_s}(c,d) = \begin{cases} 1 & \text{ha } c = d \\ \frac{\mathcal{W}_{os}(c,d)}{\mathcal{W}_{bs}(c,d)+\mathcal{W}_{os}(c,d)} & \text{különben} \end{cases}$$

$$\mathcal{W}_{os}(c,d) = min[W_{os}(f_a(c)), W_{os}(f_a(d))]$$

$$\mathcal{W}_{bs}(c,d) = max[W_{bs}(f_a(c)), W_{bs}(f_a(d))]$$

Hatékonysági problémák

- túl sokáig tart kiszámolni $\longrightarrow$ gyorsabb számítógép

Hatékonysági problémák

- túl sokáig tart kiszámolni $\rightarrow$ gyorsabb számítógép
- a számitási idő az affinitás számításával telik el $\rightarrow$ csak a használt pontokra számítsuk ki (35-55%)

Hatékonysági problémák

- túl sokáig tart kiszámolni $\longrightarrow$ gyorsabb számítógép
- a számitási idő az affinitás számításával telik el $\longrightarrow$ csak a használt pontokra számítsuk ki (35-55%)
- sok pontot feleslegesen vizsgálunk $\longrightarrow$ egy előre meghatározott küszöb alkalmazása (70-90%)

Hatékonysági problémák

- túl sokáig tart kiszámolni → gyorsabb számítógép
- a számitási idő az affinitás számításával telik el → csak a használt pontokra számítsuk ki (35-55%)
- sok pontot feleslegesen vizsgálunk → egy előre meghatározott küszöb alkalmazása (70-90%)
- sok pontot újra megvizsgálunk → hatékony adatstruktúra
  - label-correcting ↔ label-setting algoritmus
  - LIFO, FIFO
  - tördelő táblák
  - dinamikus tömbök
  - bináris, d-ary, Fibonacci-halmok

Dinamikus programozás (*label-correcting*)

Input: $\mathcal{C}$, $o \in C$, κ
Output: Egy K-összefüggő $\mathcal{C}_o = (C_o, f_o)$ színtere a $\mathcal{C}$
Segédstruktúra: egy Q sor a pontok számára

állítsuk $\mathcal{C}_o$ minden elemét 0-ra, kivéve o-t, amit 1-re állítunk
rakjunk be Q-ba minden olyan $c \in C_o$ elemet, amelyre $\mu_\kappa(o, c) > 0$
while $Q \neq \emptyset$ **do**
  vegyünk ki egy c elemet a Q-ból
  $f_{val} \leftarrow \max_{d \in C_o}[min(f_o(d), \mu_\kappa(c, d))]$
  if $f_{val} > f_o(c)$ **then**
    $f_o(c) \leftarrow f_{val}$
    rakjunk be Q-ba minden olyan e pontot, amelyre a $\mu_\kappa(c, e) > 0$, $f_{val} > f_o(e)$, $\mu_\kappa(c, e) > f_o(e)$ feltételek egyikét használjuk
  end if
end while

Dijkstra-szerű algoritmus (*label-setting*)

Input: $\mathcal{C}$, $o \in C$, κ
Output: Egy K-összefüggő $\mathcal{C}_o = (C_o, f_o)$ színtere a $\mathcal{C}$
Segédstruktúra: egy Q prioritási sor a pontok számára

  állítsuk $\mathcal{C}_o$ minden elemét 0-ra, kivéve o-t, amit 1-re állítunk
  rakjunk be o-t Q-ba
  while $Q \neq \emptyset$ **do**
    vegyünk ki egy c elemet a Q-ból, amire $f_o(c)$ maximális
    for all minden e pontra, amelyre $\mu_\kappa(c, e) > 0$ **do**
      $f_{val} \leftarrow min(f_o(d), \mu_\kappa(c, d))$
      if $f_{val} > f_o(c)$ **then**
        $f_o(c) \leftarrow f_{val}$
        módosítsuk e-t Q-ban, vagy rakjuk bele, ha még nem tettük
      end if
    end for
  end while

Dinamikus programozás küszöbértékkel ($\kappa\theta_x FOE$)

Input: $\mathcal{C}$, $o \in C$, κ, θ_x
Output: Egy K-összefüggő $\mathcal{C}_o = (C_o, f_o)$ színtere a $\mathcal{C}$
Segédstruktúra: egy Q sor a pontok számára

állítsuk $\mathcal{C}_o$ minden elemét 0-ra, kivéve o-t, amit 1-re állítunk
rakjunk be Q-ba minden olyan $c \in C_o$ elemet, amelyre $\mu_\kappa(o, c) > 0$
while $Q \neq \emptyset$ **do**
  vegyünk ki egy c elemet a Q-ból
  if $f_o(c) < x$ **then**
    $f_{val} \leftarrow \max_{d \in C_o}[min(f_o(d), \mu_\kappa(c, d))]$
    if $f_{val} > f_o(c)$ **and** $f_{val} \geq x$ **then**
      $f_o(c) \leftarrow f_{val}$
      rakjunk be Q-ba minden olyan e pontot, amelyre $\mu_\kappa(c, e) > 0$
    end if
  end if
end while
Azok a $c \in C_o$ elemek, amelyekre $f_o(c) \neq 0$, azok értéke legyen $f(c)$, a többi pedig 0.

Melyiket használjuk?

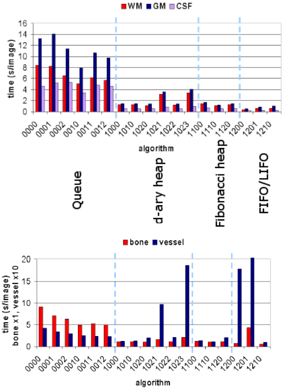
- küszöbértékkel gyorsabb lehet a számitás
- küszöbérték nélkül általánosabban használható

Címkézés ($\kappa\theta_x FOL$)

Input: $\mathcal{C}$, κ, θ_x
Output: Egy K-összefüggő $\mathcal{C}_o = (C_o, f_o)$ színtere a $\mathcal{C}$
Segédstruktúra: egy C' lista, amive átmásoljuk C elemeit

másoljuk át C-t C'-be
repeat
vegyünk ki egy c elemet a Q'-ből
határozzunk meg egy o-t tartalmazó $\kappa\theta_x$ -objektumot ($\mathcal{O}_{\theta_x}(o)$), amely megfelel a $\kappa\theta_c FOE$ algoritmusnak.
legyen az **output** $\mathcal{O}_{\theta_x}(o)$
vegyük ki C'-ből azokat a pontokat, amelyek most $\mathcal{O}_{\theta_x(o)}$-hoz tartoznak.
until $Q \neq \emptyset$

Összehasonlítás



Nem biztos, hogy egy adatsruktúra minden esetben a legjobb futásidőt produkálja.

Szegmentálás

1. Tanítás, paraméterek beállítása
   - szakértelmet igényel
   - sokáig tarthat
   - csak egy kisebb halmazra kell végrehajtani, hogy meghatározzuk a paramétereket
2. Szegmentálás végrehajtása (pontok kijelölése)
   - gyorsabb a számítás, mivel már ki vannak számolva a paraméterek és esetleg az affinitások is
   - le kell futtatni a teljes képhalamzra