Razporejanje podatkov meritev s pomočjo mehke metode k-središč

Dragan Kusić[1], Boštjan Žagar[1], Rajko Svečko[2]

[1]CAP – Center za aplikativne polimere, Pot Draga Jakopiča 22, 1231 Ljubljana - Črnuče
[2] Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, Smetanova 17, 2000 Maribor
E-pošta: info@polymer.si , rajko.svecko@um.si

Clustering of measurement data by using Fuzzy C-Means method

In this paper, the clustering results obtained with the use of basic fuzzy c-means algorithm on acoustic emission (AE) signal amplitudes are presented. AE signal amplitudes were captured during total of six production cycles of standard polypropylene test specimens under two processing conditions. The final results revealed that the objective function minimization during iterations is very dependent on the number of selected clusters. In case of higher number of clusters the fuzzy c-means algorithm clearly needs more iteration runs to cluster the input dataset.

1 Uvod

Analiza razporejanja v gruče je naloga oz. opravilo, v kateri so podatki združeni v niz podobnih objektov. Objektom v isti skupini pravimo gruče (angl. *clusters*) in so bolj podobni drug drugemu kot tistim v drugih skupinah (ali gručam). Podobni objekti na primer lahko predstavljajo zbirko vzorcev, ki se običajno oblikujejo kot vektor meritev v večdimenzionalnem prostoru. Če predstavimo podatke z manj gručami, potem običajno izgubimo določene fine podrobnosti, lahko pa dosežemo poenostavitev. Poznamo dve vrsti analize razporejanja v gruče in sicer:
- trdo razporejanje, kjer lahko vsak objekt pripada gruči, vendar ne nujno,
- mehko razporejanje (angl. *fuzzy clustering*), pri kateri vsak objekt pripada vsaki gruči z določeno stopnjo.

Analiza razporejanja v gruče ima pomembno vlogo pri številnih aplikacijah podatkovnega rudarjenja, kot npr. medicinska diagnostika, pridobivanje informacij, raziskovanje industrijskih podatkov, klasifikacija vzorcev ipd.

Tipična analiza razporejanja v gruče vključuje samo nekaj preprostih korakov, kot so: predstavitev vzorca, definicija merilne razdalje (funkcija, kot je npr. Evklidska razdalja), ki je primerna za dane podatke, združevanje (razporejanje), abstrakcija prikaza podatkov (če je potrebno) in oceno dobljenega rezultata.

Eden najbolj pogosto uporabljenih algoritmov za izvedbo analize razporejanja podatkov v gruče v številnih industrijskih aplikacijah je metoda voditeljev, ki je predstavljena v naslednjem poglavju.

2 Metoda voditeljev

Metoda voditeljev spada v osnovi med nehierarhične metode združevanja velike množice podatkov v skupine in temelji na lokalni optimizaciji. Prednost metode voditeljev je v tem, da je zmožna razvrščati v skupine večje število podatkov oz. enot, medtem ko je njena slabost predvsem dejstvo, da je potrebno vnaprej določiti, v koliko skupin želimo enote razvrstiti.

Metoda voditeljev uporablja Evklidsko razdaljo, kjer imamo ob začetku množico vnaprej podanih predstavnikov posameznih skupin, ki predstavljajo t.i. voditelje. Metoda voditeljev nato priredi enote najbližjim voditeljem in poišče centroide, ki predstavljajo težišča tako dobljenih sredin in ponazarjajo v bistvu aritmetične sredine. Ti centroidi postanejo nato novi voditelji in tem novim voditeljem so zopet prirejene najbližje enote. Postopek se konča, ko se nova množica voditeljev ne razlikuje od množice voditeljev, dobljene korak pred njo.

Začetno množico voditeljev lahko določamo na različne načine. Najpreprosteje je, če so določeni slučajno ali pa dobljeni npr. iz meritev. Najbolj učinkovit pristop je, če predhodno temeljito preučimo npr. merilne podatke in nato postavimo domneve iz katerih lahko kasneje izluščimo koristne informacije.

Pri metodi voditeljev skušamo dobiti čim boljšo razvrstitev, kar storimo tako, da postopek ponovimo večkrat z različnimi začetnimi množicami voditeljev. Ustreznost razvrstitev merimo pogosto z Wardovo kriterijsko funkcijo (kvadrat Evklidske razdalje), ki ima to lastnost, da monotono pada.

2.1 Metoda k-središč

Običajna metoda k-središč je izvedbena različica metode voditeljev za številske podatke in se še vedno precej uporablja v industrijskih aplikacijah.

Pri metodi k-središč si uporabnik naključno izbere *k* točk, ki so začetni voditelji. Vsako podatkovno točko dodamo najbližjemu voditelju dokler niso vse dodane. S tem so narejene nove skupine za katere se izračunajo težišča novo nastalih skupin, ki postanejo novi voditelji. Ta postopek se ponavlja tako dolgo, dokler se voditelji ne premikajo več. Nato se izračuna vrednost iste kriterijske funkcije, pri čemer je dobljena razvrstitev odvisna od začetne pozicije voditeljev. Iz tega razloga se ta postopek velikokrat ponavlja z različnimi začetnimi nastavitvami voditeljev, kjer se za najboljšo

razvrstitev vzame tisto, ki ima najmanjšo vrednost kriterijske funkcije.

2.2 Mehka metoda k-središč (FCM)

Pri mehkem razporejanju v gruče lahko podatkovne točke pripadajo več kot eni gruči. Pripadnostni nivoji so povezani z vsako podatkovno točko in se kasneje uporabljajo za dodelitev teh podatkovnih točk eni ali več skupinam.

Med mnogimi algoritmi mehkega '*fuzzy*' razporejanja z uporabo metode k-središč je najpogosteje uporabljen FCM algoritem (angl. *Fuzzy C-Means*), ki ga je uvedel Bezdek in nadgradili še ostali avtorji [1-16]. Večina FCM algoritmov bazira na optimizaciji ciljne funkcije, ki je formulirana kot

$$J(Z;U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m\left\|z_k-v_i\right\|_A^2, \qquad (1)$$

kjer je

$$U=\left[\mu_{ik}\right]\in M_{fc}, \qquad (2)$$

mehka particijska matrika podatkov $Z=\{z_1, z_2, \ldots z_N\}$ in element pripadnostne funkcije M_{fc},

$$V=\left[v_1,v_2,\ldots,v_c\right], \quad v_i\in\mathbb{R}^n \qquad (3)$$

je skupek vektorjev centrov gruč, katere iščemo,

$$D_{ikA}^2=\left\|z_k-v_i\right\|_A^2=\left(z_k-v_i\right)^T A\left(z_k-v_i\right), \qquad (4)$$

kjer je D_{ikA} kvadrirana distanca norme A iz točke z_k do centra gruče i, in

$$m\in\left[1,\infty\right), \qquad (5)$$

je parameter, ki določa mehčanje rezultirajoče gruče [1]. Minimizacija ciljne funkcije predstavlja nelinearni optimizacijski problem, katerega lahko rešimo preko iteracij minimizacije in tudi npr. s pomočjo genetskih algoritmov. Lahko se dokaže, da če je $D^2_{ikA} > 0$, za vse i,k in $m > 1$, potem $(U,V)\in M_{fc}$ lahko minimizira ciljno funkcijo samo, če velja [1]

$$\mu_{ik}=\frac{1}{\sum_{j=1}^{c}(D_{ikA}/D_{jkA})^{2/(m-1)}},\ 1\le i\le c,\ 1\le k\le N, \qquad (6)$$

in

$$v_i=\frac{\sum_{k=1}^{N}(\mu_{ik})^m z_k}{\sum_{k=1}^{N}(\mu_{ik})^m}, \quad 1\le i\le c. \qquad (7)$$

FCM algoritem izvaja ključne korake iteracije preko enačb (6) in (7). Zadnja enačba daje v_i kot uteženo povprečje podatkov, ki pripadajo gruči, medtem ko uteži predstavljajo pripadnostne stopnje.

Kratek povzetek FCM algoritma lahko strnemo v par preprostih korakov: Za dani podatkovni niz Z, moramo najprej definirati število gruč $1 < c < N$, toleranco prekinitve $\varepsilon > 0$, normo matrike A in utežnostni eksponent $m > 1$. Mehka particijska matrika je naključno inicializirana v takšni obliki, da velja $U\in M_{fc}$. Iterativni koraki so:

For l = 1,2…max. repeat

- *korak 1: izračunaj povprečje gruče*

$$v_i^{(l)}=\frac{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m}, \quad 1\le i\le c.$$

- *korak 2: izračun distance*

$$D_{ikA}^2=\left(z_k-v_i^{(l)}\right)^T A\left(z_k-v_i^{(l)}\right),\ 1\le i\le c,\ 1\le k\le N.$$

- *korak 3: posodobi particijsko matriko za k=1:N, in če $D_{ikA}>0$, in i=1:c potem*

$$\mu_{ik}^{(l)}=\frac{1}{\sum_{j=1}^{c}(D_{ikA}/D_{jkA})^{2/(m-1)}},\ \textit{drugače}\ \ \mu_{ik}^{(l)}=0\ \ \textit{če}$$

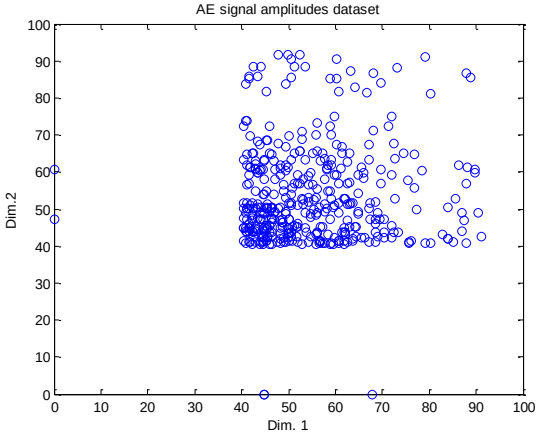
$D_{ikA}>0$, *in* $\mu_{ik}^{(l)}\in[0,1]$ *z* $\sum_{i=1}^{c}\mu_{ik}^{(l)}=1$.

until $\left\|U^{(l)}-U^{(l-1)}\right\|<\varepsilon$.

Kot je razvidno se proces razporejanja zaustavi, ko je doseženo maksimalno število iteracij ali pa če je prej minimizacija ciljne funkcije med zaporednima iteracijama manjša od specificirane tolerance zaustavitve ε.

2.3 Priprava podatkov

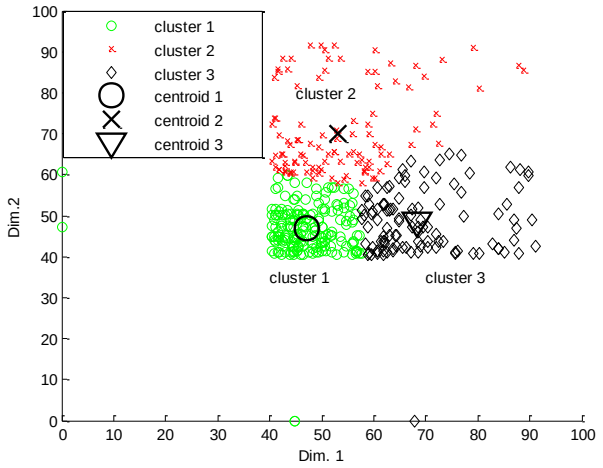
Za testiranje FCM algoritma smo uporabili amplitude izbruhov akustičnih signalov (izražene v dB), ki smo jih izmerili med procesom brizganja standardnih testnih vzorcev iz polipropilena. Tekom brizganja smo varirali hitrost brizganja od 40 mm/s do 50 mm/s, čas naknadnega tlaka od 3 s do 5 s in čas hlajenja od 6 s do 10 s. Na tako pridobljenih podatkih, ki so prikazani v dvodimenzionalnem prostoru (osi označene z Dim.1 in Dim. 2) na sliki 1, smo v nadaljevanju izvedli analizo razporejanja amplitud akustičnih signalov v posamezne gruče.



Slika 1. Priprava podatkov za razporejanje v gruče

3 Rezultati

Na izbranem testnem naboru podatkov smo testirali FCM algoritem z določitvijo dveh različnih števil skupin oz. gruč. V prvem delu smo se opredelili za 3 skupine in izračunali končno pozicijo centroidov in površino pripadnostne funkcije. Nastavili smo utežnostni eksponent na 2, toleranco prekinitve na 0,00001 ter maksimalno število iteracij na 100. V drugem delu smo se opredelili za 6 skupin in testirali algoritem pod enakimi vnaprej določenimi vrednostmi. Pred začetkom se naključno inicializirali samo mehko particijsko matriko. Utežnostni eksponent je pomemben parameter, saj ima velik vpliv na mehčanje rezultirajoče particije. Na sliki 2 so predstavljeni končni rezultati, ki so bili pridobljeni za tri skupine. Končni položaj vsake centroide gruče oz. skupine je prikazan v tabeli 1.



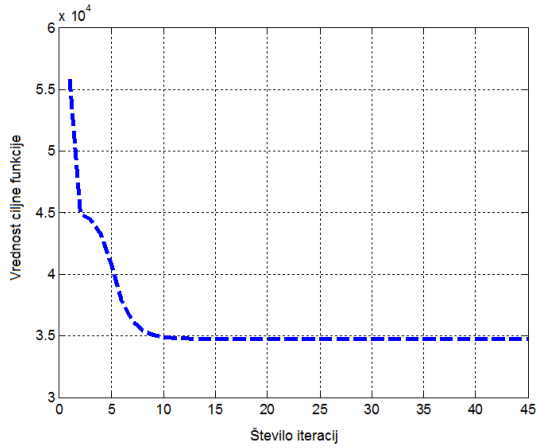
Slika 2. Končni rezultat razporejanja podatkov v 3 gruče

Tabela 1. Končna pozicija centroid za 3 gruče

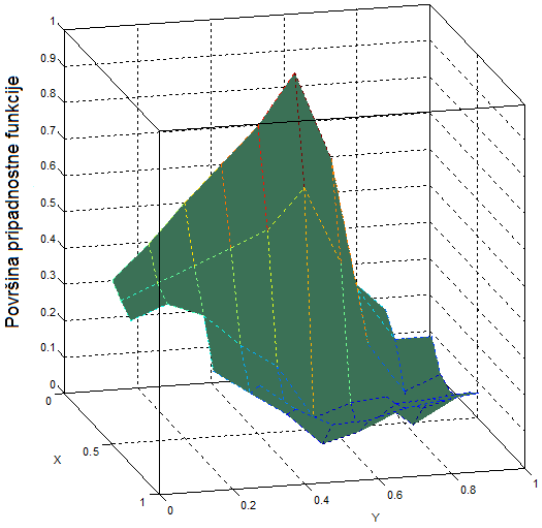
Gruča (cluster)	Dim.1	Dim.2
1	47.077	47.027
2	53.095	70.161
3	68.252	48.731

Iz slike 3 lahko vidimo, da je bila ciljna funkcija uspešno minimizirana med iteracijami in da je FCM algoritem razporedil podatke v 3 gruče po skupno 16 iteracijah. Slika 4 prikazuje dobljeno površino pripadnostne funkcije, ki smo jo dobili za gručo 1 iz katere je lepo razvidno, da ciljna funkcija razrešuje nelinearni optimizacijski problem.

V drugem delu smo izbrali 6 gruč oz. skupin, kjer je končni položaj vsakega centroida gruče prikazan v tabeli 2. Na sliki 5 lahko opazimo, da je bila ciljna funkcija med ponovitvami uspešno minimizirana in da je FCM algoritem uspešno grupiral gruče po 45 iteracijah. Na sliki 6 so prikazani končni rezultati za 6 gruč s centroidi. S primerjavo rezultatov na sliki 2 in sliki 6 smo potrdili, da definirano število gruč oz. skupin močno vpliva na učinkovitost FCM algoritma med iteracijami. Kot je razvidno iz slike 3 in slike 5 je število potrebnih iteracij za doseganje lokalnega minimuma ciljne funkcije v primeru 6 gruč cca. 3-krat večje.



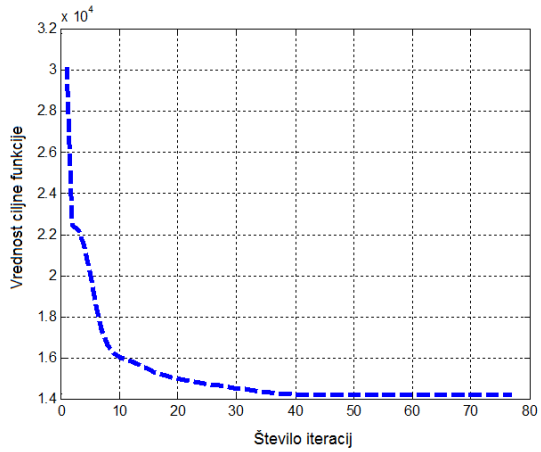
Slika 3. Minimizacija ciljne funkcije med posameznimi iteracijami (primer za 3 gruče)



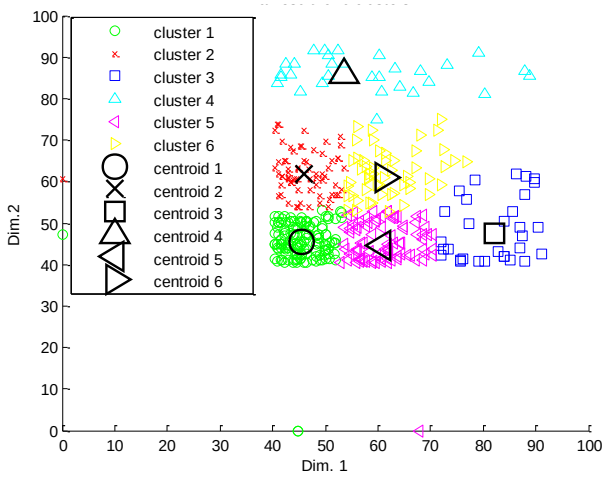
Slika 4. Površina pripadnostne funkcije za centroid gruče 1

Tabela 2. Končna pozicija centroid za 6 gruč

Gruča (cluster)	Dim.1	Dim.2
1	45.441	45.519
2	45.888	61.848
3	82.167	47.468
4	53.499	85.877
5	60.699	44.636
6	61.166	61.027



Slika 5. Minimizacija ciljne funkcije med posameznimi iteracijami (primer za 6 gruč)



Slika 6. Končni rezultat razporejanja podatkov v 6 gruč

4 Zaključek

V tem članku so predstavljeni rezultati analize razporejanja podatkov v gruče, ki so bili izvedeni na podatkovnem nizu iz meritev amplitud akustičnih signalov ter ovrednoteni s pomočjo FCM algoritma.

Na osnovi razporejenih amplitud akustičnih signalov lahko bistveno lažje karakteriziramo proces brizganja v različnih proizvodnih fazah, kar bi bilo v nasprotnem primeru bistveno težje opravilo.

Naši rezultati potrjujejo, da vnaprej izbrano število gruč močno vpliva na učinkovitost FCM algoritma v primeru, ko so ostali parametri fiksni in niso aktivno spreminjani, kot npr. utežnostni eksponent, toleranca zaustavitve in norma matrike A. Ugotovili smo, da so dobljeni rezultati in njihova razporeditev v gruče odvisni od različnih tehnoloških parametrov. V našem primeru sta na razporejanje v gruče najbolj vplivala dva procesna parametra in sicer hitrost brizganja ter čas delovanja naknadnega tlaka, kjer so bile izmerjene maksimalne amplitude.

Na splošno delovanje takšnih algoritmov za razporejanje v gruče vpliva tudi gostota posameznih gruč oz. skupin, njihovih prostorskih relacij in njihovih razdalj. Pomanjkljivost FCM algoritma je dejstvo, da njegova funkcionalnost konvergira proti lokalnem minimumu. Torej v primerih, kadar uporabljamo drugačno inicializacijo, dobimo pogosto precej različne rezultate. Dodatne izboljšave tega algoritma so možne predvsem v smeri spreminjanja ciljne (objektivne) funkcije, merjenja razdalje in spreminjanja zgoraj omenjenih parametrov tega algoritma.

Zahvala

Delo je bilo financirano s strani podjetja CAP d.o.o., Ljubljana - Črnuče in Evropske komisije (EUREKA program, projekt E!-9574 IMPROMON).

Literatura

[1] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences, Vol. 10, 191-203, 1984.

[2] J.C. Bezdek, S.K. Chuah, D. Leep, Generalized k-nearest neighbor rules, Fuzzy Sets and Systems, Vol. 18, 237-256, 1986.

[3] N.R. Pal, J.C. Bezdek, R.J. Hathaway, Sequential Competitive Learning and the Fuzzy c-Means Clustering Algorithms, Neural Networks, Vol. 9, 787-796, 1996.

[4] A. Flores-Sintas, J. Cadenas, F. Martin, Membership functions in the fuzzy C-means algorithm, Fuzzy Sets and Systems, Vol. 101, 49-58, 1999.

[5] J. Liu, M. Xu, Kernelized fuzzy attribute C-means clustering algorithm, Fuzzy Sets and Systems, Vol. 159, 2428-2445, 2008.

[6] Z. Ji, Q. Sun, Y. Xia, Q. Chen, D. Xia, D. Feng, Generalized rough fuzzy c-means algorithm for brain MR image segmentation, Computer Methods and Programs in Biomedicine, Vol. 108, 644-655, 2012.

[7] S. R. Kannan, S. Ramathilagam, R. Devi, E. Hines, Strong fuzzy c-means in medical image data analysis, Journal of Systems and Software, Vol. 85, 2425-2438, 2012.

[8] K. S. Chuang, H. L. Tzeng, S. Chen, J. Wu, T. J. Chen, Fuzzy c-means clustering with spatial information for image segmentation, Computerized Medical Imaging and Graphics, Vol. 30, 9-15, 2006.

[9] H. Wang, B. Fei, A modified fuzzy C-means classification method using a multiscale diffusion filtering scheme, Medical Image Analysis, Vol. 13, 193-202, 2009.

[10] P. Huang, D. Zhang, Locality sensitive C-means clustering algorithms, Neurocomputing, Vol. 73, 2935-2943, 2010.

[11] J. F. Yang, S. S. Hao, P. S. Chung, Color image segmentation using fuzzy C-means and eigenspace projections, Signal Processing, Vol. 82, 461-472, 2002.

[12] G. E. Tsekouras, On the use of the weighted fuzzy c-means in fuzzy modeling, Advances in Engineering Software, Vol. 36, 287-300, 2005.

[13] D. Q. Zhang, S. C. Chen, A novel kernelized fuzzy C-means algorithm with application in medical image segmentation, Artificial Intelligence in Medicine, Vol. 32, 37-50, 2004.

[14] A. B. Goktepe, S. Altun, A. Sezer, Soil clustering by fuzzy c-means algorithm, Advances in Engineering Software, Vol. 36, 691-698, 2005.

[15] D. P. Mukherjee, P. Pal, J. Das, Sodar image segmentation by fuzzy c-means, Signal Processing, Vol. 54, 295-301, 1996.

[16] Y. Chtioui, D. Bertrand, D. Barba, Y. Dattee, Application of fuzzy C-Means clustering for seed discrimination by artificial vision, Chemometrics and Intelligent Laboratory Systems, Vol. 38, 75-87, 1997.