Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/07BE40Ilijazovic

# КЛАСТЕРИЗАЦИЈА ВИСОКОДИМЕНЗИОНАЛНИХ ПОДАТАКА ЕЛЕКТРОЕНЕРГЕТСКИХ ПОТРОШАЧА У ПРОГРАМСКОМ ЈЕЗИКУ R

# HIGH-DIMENSIONAL CLUSTERING OF ELECTRIC CONSUMER DATA IN R PROGRAMMING LANGUAGE

Елведин Илијазовић, *Факултет техничких наука, Нови Сад*

**Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**

**Кратак садржај –** *Тема рада је поређење перформанси софтверског решења за кластеризацију профила потрошње у програмском језику R у односу на решење имплементирано у .NET развојном окружењу. У раду је описан програмски језик R и његове библиотеке, које су коришћене у реализацији решења. Описани су недостаци због којих се јавља потреба за кластеризацијом профила као и метода редуковања димензионалности података уз минимални губитак информација, а све зарад тачности саме кластеризације.*

**Кључне Речи:** *Кластеризација, машинско учење, програмски језик R*

**Abstract** – *In this work it will be shown how data clustering of Consumer Load Profiles in R programming language performs, opposed to same implementation in .NET framework. Programming language R alongside with its libraries that have been used for this implementation has been described. The reason of why data clustering is used has also been described with methods of machine learning.*

**Keywords:** *Clustering, Machine Learning, R Programming Language*

## 1. УВОД

Напретком технологије имамо могућност складиштења података о потрошњи електричне енергије на нивоу појединачних потрошача и у релативно честим временским интервалима. На основу ових података о понашању потрошача могуће је естимирати производњу електричне енергије и смањити губитке Прикупљени подаци су у сировој форми и, као такви, нису подобни за коришћење.

Један од разлога је сама количина података те их је потребно груписати по сличности како би били корисни у аналитичким процесима система управљања дистрибуцијом.

У овом раду је описана кластеризација података о електричној потрошњи у програмском језику R. Приказани су алгоритми кластеризације од којих је детаљно описан ***K-means***.

---

**НАПОМЕНА:**
**Овај рад проистекао је из мастер рада чији ментор је био др Александар Купусинац, ванр.проф.**

Уводни део описа решаваног проблема описује потребу за прикупљањем података као и потребу за њиховом кластеризацијом. Описани су профили потрошње, као и потреба за прикупљањем не само мерења о активној снази потрошача, већ и мерења о његовој реактивној снази.

Такође, описана је анализа главних компоненти у циљу редуковања димензионалности простора са подацима као и алгоритам најближих суседа у циљу класификације потрошача са невалидним сетом података.

Поглавље са описом постојећег решења садржи детаљан опис формирања сета података који се кластерују као и процес самог кластеровања.

У раду су описане коришћене технологије, пре свега програмски језик ***R*** који је служио за имплементацију решења овог рада као и ***.NET*** развојно окружење у којем је имплементирано постојеће решење. Такође је описана ***Microsoft SQL Server*** релациона база података која служи за складиштење података.

Приликом описа решења приказани су исечци R кода са методама које су позиване, њихов опис као и опис параметара које им се прослеђују. Упоређено је постојеће решење имплементирано у .NET развојном окружењу са решењем овог рада. Измерене су перформансе између ове две апликације које су вршиле кластеризацију података са идентичном поставком улазних параметара.

## 2. ОПИС РЕШАВАНОГ ПРОБЛЕМА

У овом поглављу биће описани проблеми са којом се суочава електродистрибуциона мрежа. Такође ће бити описани профили потрошње као и потреба за њиховом кластеризацијом. Дискутовано је постојеће решење као и алгоритми кластеризације, њихова валидација, редукција димензионалности и алгоритам класификације.

### 2.1 СНАБДЕВАЊЕ И ПОТРАЖЊА ЕНЕРГИЈЕ

У електроенергетском систему баланс између генерисања електричне енергије и њене потрошње (у потрошњу се урачунава и енергија губитка на мрежи) представља важан аспект. Веома је важно да снабдевена генерисана енергија одговара потрошњи, јер у супротном долази до нестабилности електричне мреже, што може довести до прекида снабдевања енергијом. Како енергетски систем спада у систем са

критичном мисијом, ово може имати значајне последице јер утиче на животе великог броја људи.

Електроенергетске компаније овај проблем решавају проценом потражње електричне енергије и на основу ове естимације планирају производњу. Потрошња се естимира на основу претходне потрошње, а за то су нам потребна мерења потрошача на основу којих се креирају профили потрошње.

## 2.2 ПРОФИЛИ ПОТРОШЊЕ

У електроинжењерству, профил потрошње представља граф промене електричне потражње односно потрошње по специфичном времену. Произвођачи електричне енергије користе ове информације како би испланирали колико електричне енергије ће бити потребно обезбедити за неко време у будућности.

Прикупљањем података о потрошњи сваког потрошача јавља се нови проблем – обим података. Уколико се користи петнаестоминутни интервал, за измерене вредности активне и реактивне снаге за период од једне године, може настати и до неколико десетина, па чак и стотина гигабајта података. Ова огромна количина података није погодна за рачунарску обраду у реалном времену, па се пре тога ови подаци агрегирају у профиле потрошње и над њима се врши кластеризација како би се редуковао скуп података са што мањим губитком информација.

## 2.5 КЛАСТЕРИЗАЦИЈА

Кластеризација представља груписање објеката на такав начин да су објекти у групи (кластеру) више слични (у неком смислу) међусобно, у односу на објекте група којима не припадају [1]. Ово је главни задатак *data mining-a* у статистичкој анализи података. За решење нашег проблема одабран је модел центроида. Потрошачки профили који припадају истој групи добијају вредност средње вредности свих чланова групе и тиме се редукује количина података потребна за прорачуне.

### 2.5.1 *K-MEANS* АЛГОРИТАМ

Метода средњих вредности (***k-means***) је метода векторске квантизације која је популарна за кластеризацију приликом анализе података. Ова метода има за циљ да распореди $\boldsymbol{N}$ тачака $\boldsymbol{I}$-димензионалног простора у $\boldsymbol{k}$ група, односно кластера, у којој свака тачка припада кластеру са најсличнијим карактеристикама. Тачке означавамо са $\{\boldsymbol{x}^{(n)}\}$, где се $\boldsymbol{n}$ креће од 1 до броја тачака $\boldsymbol{N}$. Претпоставка је да је простор у коме се налази $\boldsymbol{x}$ реалан и да постоји метрика којом се дефинише дистанца између тачака, као на пример једначина за израчунавање дистанце у еуклидском простору:

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2} \qquad (1)$$

Најчешћа коришћена верзија алгоритма користи итеративну технику за проналажење кластера. Овај алгоритам се може поделити у три корака: корак иницијализације, корак доделе и корак ажурирања.

У кораку иницијализације се постављају иницијалне средње вредности $\{\mathbf{m}^{(k)}\}$ за кластере. Иницијалне средње вредности кластера се могу изабрати насумично, али постоје и посебне методе за иницијализацију.

У кораку доделе, свака тачка $\boldsymbol{n}$ се додељује кластеру чијем је центру најближа по Еуклидском растојању. Означимо нашу претпоставку да тачка $\boldsymbol{x}^{(n)}$ припада кластеру $\boldsymbol{k}^{(n)}$ као $\widehat{\boldsymbol{k}}^{(n)}$ тако да:

$$\hat{k}^{(n)} = \underset{k}{\operatorname{argmin}}\left\{d(\mathrm{m}^{(k)}, x^{(n)})\right\} \qquad (2)$$

Алтернативно, можемо ову доделу тачака кластерима представити и кроз параметре „задужења“ $\boldsymbol{r}_{\boldsymbol{k}}^{(\boldsymbol{n})}$. У кораку доделе параметеру $\boldsymbol{r}_{\boldsymbol{k}}^{(\boldsymbol{n})}$ се додељује вредност 1 ако је средња вредност кластера $\boldsymbol{k}$ најближа тачки $\boldsymbol{x}^{(\boldsymbol{n})}$, у супротном $\boldsymbol{r}_{\boldsymbol{k}}^{(\boldsymbol{n})}$ добија вредност 0. Ово је представљено једначином:

$$r_k^{(n)} = \begin{cases} 1, & \hat{k}^{(n)} = k \\ 0, & \hat{k}^{(n)} \neq k \end{cases} \qquad (3)$$

Након корака доделе следи корак ажурирања, у коме се рачунају нове позиције центара кластера које представљају средње вредности тачака које су му додељене тј. за које је задужен.

$$\mathrm{m}^k = \frac{\sum_n r_k^{(n)} x^{(n)}}{R^{(k)}} \qquad (4)$$

где је $R^{(k)}$ укупно задужење средње вредности $k$, и једначина му је:

$$R^{(k)} = \sum_n r_k^{(n)} \qquad (5)$$

Кораци доделе и ажурирања се понављају све до тренутка када задужења за све вредности кластера не мењају, тј. када алгоритам конвергира.

Алгоритам методе средњих вредности не гарантује да ће увек конвергирати глобалном оптимуму. Резултат кластеровања увелико зависи од почетне иницијализације средњих вредности кластера. Како је овај алгоритам у већини случајева брз, честа је пракса да се покрене више пута са различитим сетом иницијалних вредности.

### 2.5.2 *K-MEANS++* АЛГОРИТАМ

У анализи података, ***k-means++*** представља алгоритам за одабир иницијалних вредности који се користе као улаз у методу средњих вредности. Овај алгоритам је предложен 2007. године од стране Дејвида Артура и Сергеја Василвитског, у виду алгоритма апроксимације за НП тешке проблеме кластеровања по принципу средњих вредности као начин за избегавање повремено лошег кластеровања од стране стандардног алгоритма средњих вредности.

Овај алгоритам креће са интуицијом да је боље раширити позиције иницијалних центара кластера по простору са подацима. Први центар је насумично одабран из објеката који се групишу, после чега се сваки наредни центар бира од остатака објеката са вероватноћом пропорционалном његовој квадратној дистанци од најближег постојећег центра кластера.

### 2.5.3 ИНДЕКСИ ВАЛИДНОСТИ

Индекс валидности кластеровања служи за мерење квалитета резултата кластеровања у поређењу са резултатима добијеним другим алгоритмом кластеризације, или истим алгоритмом са другачијим улазним параметрима. Ови индекси су обично погодни за мерење кластера које се међусобно не преклапају [2].

### 2.5.4 АНАЛИЗА ГЛАВНИХ КОМПОНЕНТИ *(PCA)*

Кластеризација производи лошије резултате што је број димензија већи. Из тог разлога је потребно редуковати димензионалност на такав начин да се и даље очува већина информација о подацима. Ово се постиже помоћу анализе главних компоненти.

Анализа главних компоненти (***Principal component analysis*** (***PCA***)) представља процедуру у статистици која користи ортогоналну трансформацију у циљу конверзије сета објеката(обсервација), са могућим узајамним везама између варијабли (ентитети који имају неку нумерчку репрезентацију), у сет вредности линеарно неповезаних варијабли које се зову главне компоненте (***principal conponents***).

***PCA*** може бити замишљен као процес постављања ***n*** димензионог елипосоида подацима, где свака оса елипсоида представља главну компоненту. Уколико су неке осе елипсоида мале, тада је варијанса по тој оси такође мала и уклањањем те осе и главних компоненти које им припадају долази до малог губитка информација.

### 2.5.5 АЛГОРИТАМ К НАЈБЛИЖИХ СУСЕДА

У препознавању узорака (***pattern recognition***), алгоритам к најближих суседа (***k-nearest neighbours*** (***k-NN***)) се користи као непараметризована метода за класификацију и регресију [3]. У оба случаја, улаз се састоји од *k* најближих тест примера у функцији простора.

## 2.4 ПОСТОЈЕЋЕ РЕШЕЊЕ

Постојеће решење је имплементирано у ***.NET*** развојном окружењу и изведено је у два корака, корак претпроцесирања, који служи за припрему података и корак процесирања, који врши кластеризацију над подацима.

### 2.4.1 КОРАК ПРЕТПРОЦЕСИРАЊА

Овај корак неће бити имплементиран у *R* језику, али је описан како би се приказала структура података који се кластерују.

У овом кораку се креирају потрошачки профили за сваког потрошача на основу достављених података и конфигурације. Конфигурација се састоји из годишњих сезона и типичних дана(нпр. радни дан, викенд). Подаци се агрегирају тако што се израчуна просек потрошње који је измерен код потрошача за дати временски тренутак за сваку комбинацију конфигурисаних сезона и типичних дана. Свако мерење представља једну димензију простора по којој се подаци кластерују. Број димензија зависи од итнервала мерења и конфигурације, и добија се множењем броја мерења у једном дану, бројем сезона, бројем типичних дана и врстом мерења (мере се и активна и реактивна снага за дати трентуак). Тако се за нпр. 48 дневних мерења, 12 сезона и 3 типична дана добије 3.456 димензија.

Такође се у овом кораку врши нормализација на начин да се све вредности активне снаге потрошача поделе са његовом просечном годишњом активном снагом. Излаз овог корака су нормализоване агрегиране криве намапиране на одговарајућу комбинацију конфигурационих параметара.

### 2.4.2 КОРАК ПРОЦЕСИРАЊА

У овом кораку се врши кластеризација над подацима генерисаним у претпроцесирању. Пре самог кластеровања потребно је, за сваког потрошача, све комбинације кривих спојити у једну. Врши се и селекција потрошача на валидне и невалидне. Валидни потрошачи су они који садрже криве за сваку комбинацију, а остали се прогласе за невалидне и они ће бити класификовани у цетроиде креиране на основу валидних потрошача.

Постојеће решење за алгоритам кластеровања користи *k-means++* који као улазни параметар и, поред самих података, захтева и број кластера. Како нам је податак о улазним параметрима непознат, задаје се опсег група од минималног до максималног броја кластера. Алгоритам се извршава за сваки број група из опсега и узима се најбоље решење. Одабир најбољег решења врши се уз помоћ индекса валидности кластера.

Излаз овог корака су центроиде кластера (потрошачке групе) као и њихово мапирање на потрошаче који им припадају. Ово значи да типичну криву неког потрошача добијамо множењем његовог потрошачког профила са средњим годишњим снагама.

## 3. ПРОГРАМСКИ ЈЕЗИК *R*

***R*** је програмски језик и бесплатно софтверско окружење за статистичке прорачуне и графику које подржава ***R*** фондација. ***R*** језик је широко коришћен међу статистичарима за развој статистичког софтвера и анализу података. Овај програмски језик омогућава компајлирање и покретање на великом броју ***UNIX*** платформи, као и на ***Windows*** и ***MacOS*** оперативним системима. Анкете, анализе података и студије научне литературе показују значајно повећање популарности овог програмског језика. Од октобра 2019. године, ***R*** је на 15. месту по ***TIOBE*** индексу мерења популарности програмских језика [4].

## 4. *R* ИМПЛЕМЕНТАЦИЈА РЕШЕЊА

Први корак у апликацији је добављање нормализованих дневних профила потрошње из базе података које су креиране у кораку претпроцесирања. Калстеризација се обавља над валидним потрошачима. На основу ових података генерише се матрица података са редовима потрошача и колонама спојених комбинација нормализованих мерења. Ова матрица, на реалном примеру са 12 сезона, 3 карактеристична дана и петнаестоминутним интер-валом (96 дневних мерења), садржи 6.912 колона (димензија).

Следећи корак је редукција димензионалности помоћу ***PCA*** алгоритма. Ово се постиже помоћу ***prcomp***

функције која врши анализу главних компоненти улазне матрице података.

```
PCA.result = prcomp(data, center = TRUE, scale = TRUE)
```

Одабир главних компоненти које садрже 95% информација је имплементиран на следећи начин:

```
#Računanje varijanse:
variance = PCA.result$sdev ^ 2
#Računanje kumulativnih proporcija:
variance.proportion = cumsum(variance / sum(variance))
#Odabir dimenzija koje sadrže 95% informacija:
reduced.dim = GetRowCount
(variance.proportion[variance.proportion <= 0.95]) + 1
#Dimenzije koje su nam od značaja:
PCA.result$x[, 1:reduced.dim]
```

Након редукције димензија следи корак кластеровања. Ово је постигнуто помоћу ***KMeans_rcpp*** функције из пакета ***ClusterR***:

```
ClusterR::KMeans_rcpp(data = PCA.result,
                      clusters = k,
                      num_init = n,
                      initializer = "kmeans++",
                      seed = sample(1:10000, 1))
```

Резултат функције ***KMeans_rcpp*** је мапирање објекта са групом којој припада.

Након завршетка кластеризације, врши се израчунавање индекса валидности. За индекс валидности се користи Дејвис-Боулдин индекс валидности из ***clusterSim*** пакета. Позив функције је следећи:

```
clusterSim::index.DB(data,
                     cl=clusters,
                     d=NULL,
                     centrotypes="centroids",
                     p=2,
                     q=2)$DB
```

Калстеризација се извршава за опсег група ***kmin*** - ***kmax*** са ***n*** понављања и као резултат се узима она вредност кластеризације која има најбољи индекс валидности.

Наредни корак је класификација потрошача са невалидним мерењима, тј. потрошача који немају све податке за комбинацију сезона и карактеристичних дана. Ово се постиже помоћу функције ***predict.***

Добијен резултат апликације је матрица са центроидама група и мапирање потрошача на групу којој припада.

## 5. ПОРЕЂЕЊЕ СА ПОСТОЈЕЋИМ РЕШЕЊЕМ

Два решења у овом раду су поређена над истим скупом података и са истим улазним конфигурационим параметрима и извршена су над истом машином. Тестирање се извршавало над скупом од 11.494 потрошача, од којих су 8.055 валидни, а 3.494 невалидни. Број димензија за задату комбинацију конфигурације износи 6.912. Опсег група је 50-100 и број понављања 3. Резултат итвршавања апликација је приказан у табели 1.

Табела 1. *Поређење времена извршавања*

| | .NET имплементација | R имплементација |
|---|---|---|
| Време трајања PCA алгоритма | 03:16:12 | 00:04:43 |
| Број димензија након редукције | 64 | 205 |
| Време трајања кластеризације | 07:48:33 | 2:15:05 |
| Број кластера | 53 | 54 |
| Укупно средње време извршавања | 11:04:45 | 02:19:48 |

Из табеле 4.5 се може закључити да је време проналажења кластера ***R*** имплементације за 79% краће у односу на време проналажења ***.NET*** имплементације.

## 6. ЗАКЉУЧАК

У раду је приказано решење за кластеризацију које је имплементирано у програмском језику ***R.*** Приказано решење је имало боље перформансе у односу на постојеће решење имплементирано у ***.NET*** развојном окружењу. Иако ***PCA*** алгоритам у ***R*** имплементацији редукује димензионалност на 205 димензија, у односу на постојеће решење које димензионалност смањује на 64 димензије, ***R*** имплементација и даље долази до резултата у краћем временском интервалу.

Треба напоменути и лакоћу имплементације. Намена ***R*** програмског језика је управо у статистичким прорачунима над подацима и као такав садржи већ имплементиране библиотеке које су се користиле у нашој имплементацији. Постојеће (***.NET***) решење је морало имплементирати све алгоритме наведене у овом раду. Ово значи да је имплементација у ***R*** језику неупоредиво једноставнија и, самим тим што садржи мање кода, једноставнија за одржавање и евентуалне измене и унапређења.

## 7. ЛИТЕРАТУРА


[1] S. Guha, R. Rastogi and K. Shim: *CURE: an efficient clustering algorithm for large databases*, Proc. of ACM SIGMOD International Conference on Management of Data, pp. 73 – 84, 1998.

[2] Csaba Legány, Sándor Juhász and Attila Babos; "*Cluster Validity Measurement Techniques*"; Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, February 15-17, 2006 (pp388-393)

[3] Altman, N. S. (1992). "*An introduction to kernel and nearest-neighbor nonparametric regression*". The American Statistician. 46 (3): 175–185. doi:10.1080/00031305.1992.10475879. hdl:1813/31637

[4] TIOBE Index - The Software Quality Company, TIOBE, https://www.tiobe.com/tiobe-index/ (приступљено октобра 2019.)


**Кратка биографија:**


**Елведин Илијазовић** рођен је у Новом Саду 1992. год. Факултет техничких наука уписао је 2011. год. Бечелор рад из области Електротехнике и рачунарства – рачунарске науке и информатика, одбранио је 2016. год.