UDC 004.89:004.738.5 INFO M: str. 23-29

PROJEKTOVANJE PROCESA KLASTEROVANJA POMOĆU PATERNA
DESIGNING CLUSTERING PROCESS WITH REUSABLE COMPONENTS

Kathrin Kirchner, Boris Delibašić, Milan Vukićević

REZIME: Tipičan proces otkrivanja zakonitosti u podacima (dejta majning, u daljem tekstu OZP), prema CRISP-DM metodologiji se sastoji od nekoliko faza, počevši od razumevanja poslovnog procesa i podataka, preko predprocesiranja, modelovanja i evaluacije. Za svaku od ovih faza, predstavljeno je nekoliko generičkih zadataka koje treba sprovesti. Kod rešavanja praktičnih problema, jako je teško odlučiti koji specijalizovani zadatak najviše odgovara odgovarajućoj generičkoj fazi. Razlog za ovakav problem leži najmanje u tri razloga. Kao prvo, postoji jako puno specijalizovanih zadataka u literaturi i njihovih implementacija u softverima za OZP. Drugo, dosta ovih zadataka je enkapsulirano u algoritmima i ne mogu se izvršavati nezavisno od algoritma. Kao treće, specijalizovani zadaci (ponovo upotrebljive komponente, u daljem tekstu PUK) nisu dobro organizovani. Na primer, nije lako odabrati odgovarajuću PUK za generički zadatak (pod-problem) konkretnog poslovnog problema. U ovom radu, predstavljamo predlog metodologije modelovanja, baziranog na principu „belih kutija“, koji podržava proces OZP. Takođe, dat je prikaz metodologije za probleme grupisanja podataka (u daljem tekstu klasterovanje) kao i predlog konkretnih paterna, zasnovanim na korišćenju PUK, za pod-probleme koji se često pojavljuju kod klasterovanja podataka, pred-procesiranja i post-procesiranja.
KLJUČNE REČI: Klasterovanje, otkrivanje zakonitostu u podacima, paterni, CRISP-DM
ABSTRACT: A typical data mining process, as it is described e.g. in the CRISP-DM approach, consists of several phases starting from business and data understanding and proceeds with preprocessing, modeling and evaluation. For each of these phases, several generic tasks are described that have to be carried out. In practice, however, there are difficulties to decide which specialized task solves a generic task best. There are at least three reasons for this. First, a galore of specialized tasks is proposed in the literature and available in data mining software. Second, a lot of these tasks are encapsulated in algorithms, and can't be used independently of the algorithm. Third, specialized tasks (reusable components - RCs) are not well-organized, i.e. it is not easy to select the appropriate RC for a generic task (sub-problem). In this paper, we propose a white box modeling methodology that supports the design of the data mining process. Our paper concentrates on clustering algorithms only. Thus, we propose RCs for commonly appearing sub-problems in clustering, as well as pre- and post-processing RCs.
KEY WORDS: Clustering, data mining, paterns, CRISP-DM

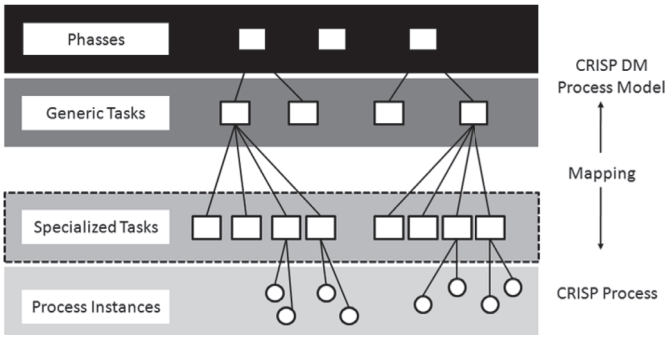
1. UVOD

Procesni modeli za OZP pomažu kompanijama da bolje razumeju process otkrivanja znanja i obezbeđuju proceduru za planiranje i sprovođenje OZP projekata (Kurgan & Musilek, 2006). Korišćenje procesnih modela na ovaj način rezultira uštedama u vremenu i novcu, kao i u boljem razumevanju lakšem prihvatanju takvih projekata i njihovih rezultata.

CRISP-DM (Wirth & Hipp, 2000) je jedan od standnih pristupa koji pomaže u transformisanju poslovnih problema u zadatke OZP uključujući pripremu podataka i evaluaciju rezultata. To je iterativno-inkrementalni proces koji se sastoji iz šest faza: razumevanje poslovnog procesa, razumevanje podataka, priprema podataka, modelovanje (izbor algoritma, podešavanje parametara, primena algoritma i tumačenje rezultata), evaluacija i primena. Svaka od ovih faza je hijerarhijski podržana skupom zadataka na tri nivoa apstrakcije (Slika 1.)

Na drugom nivou, svaka faza se sastoji od nekoliko generičkih zadataka, koji opisuju opšte probleme koje treba rešiti. Npr, u fazi razumevanja podataka, treba prikupiti podatke, opisati ih i oceniti kvalitet podataka. Opis zadataka je dovoljno uopšten za svaki problem otkrivanja znanja.

Treći nivo je fokusiran na rešavanje generičkih zadataka u specifičnim situacijama OZP, npr. u slučaju izbora metrike nominalnih promenljivih. Kasnije u rešavanju konkretnih problema OZP projekta, neophodno je odlučiti koje zadatke je potrebno izvršiti. Zbog toga, četvrti nivo opisuje zadatke koji su bili izvršeni u konkretnom OZP projektu.



Slika 1. – *Četiri nivoa CRISP-DM metodologije (Chapman et al., 2000)*

Faze i generički zadaci su dobro opisani u CRISP-DM procesnom modelu (Chapman et al., 2000), ali je malo urađeno po pitanju opisa specijalizovanih zadataka na nivou 3. U ovom radu se fokusiramo na definisanje specijalizovanih zadataka, kako bi olakšali odluku pri izboru zadatka u konkretnoj situaciji OZP-a.

Da bi opisali specijalizovane zadatke moguće je primeniti ideju o ponovo upotrebljivim komponentama (PUK), prikazanim u Tabeli 1. Prema Sametinger (1997), ponovo upotrebljive

softverske komponente su nezavisni, jasno odredivi artifakti koji opisuju i/ili obavljaju specifične funkcije, imaju jasne interfejse, adekvatnu dokumentaciju, kao i status ponovne upotrebe.

Tabela 1. – *Analogija između CRISP-DM nivoa i PUK dizajna*

	CRISP-DM	PUK dizajn
Nivo 2	Generički zadaci	Pod – problem
Nivo 3	Specijalizovani zadaci	Ponovo upotrebljive komponente
Nivo 4	Instance procesa	Odabrane PUK

Smatramo da dizajn baziran na PUK može poboljšati CRISP-DM process. Izabrali smo problem klasterovanja kao jedno od najprimenjivanijih polja u OZP.

Cilj klasterovanja je da odredi grupe podataka čiji su objekti slični unutar grupa i različiti između grupa. Berkhin (2006) klasifikuje algoritme klasterovanja u sledeće grupe:

– *Partitivni algoritmi*: Korisnik zadaje broj grupa na koje će skup podataka (*eng. dataset*) biti podeljen. Tipični algoritmi su: "k-means" (MacQueen, 1967) i "k-medoids" (Kaufman & Rousseeuw, 1990).
– *Hijerarhijski algoritmi*: Ovi algoritmi dele ili spajaju skup podataka hijerarhijski. Aglomerativni hijerarhijski algoritmi počinju tako što na početku svaki objekat predstavlja zaseban klaster, a zatim se hijerarhijski spajaju dok se svi objekti ne spoje u jedan klaster. Divizioni algoritmi počinju tako što su na početku svi objekti u jednom klasteru, a, a zatim se hijerarhijski dele sve dok svaki objekat ne postane poseban klaster. Primeri za ovu vrstu algoritama su Agnes i Diana (Kaufman & Rousseeuw, 1990).
– *Algoritmi zasnovani na gustini*: Klasteri su definisani kao oblasti gustine, gde je gustina definisana kao broj objekata na unutar okoline radijusa ε. Tipičan predstavnik algoritama zasnovanih na gustini je DBScan (Ester et al. 1996).
– *Mrežni algoritmi*: Prostor podataka je podeljen na ćelije. Klaster se sastoji od povezanih ćelija sa visokom gustinom objekata. Popularni predstavnici ove grupe algoritama su: STING (Wang et al. 1997) i WaveCluster (Sheikholeslami et al. 1998).
– Algoritmi bazirani na modelima: Ovi algoritmi su bazirani na matematičkim modelima i statističkim metodama. Primer je COBWEB (Fisher 1987).

Ovaj rad je organizovan na sledeći način. Prvo, opisujemo tipične probleme kod procesa klasterovanja i prikazujemo njihova rešenja, koja su korišćena u različitim algoritmima i softverima za OZP. Pošto se ovi problemi često pojavljuju, dajemo definiciju ponovo upotrebljivih komponenti (PUK) koje se mogu koristiti za poboljšanje postojećih algoritama ili dizajn potpuno novih algoritma. Zbog toga, prikazujemo PUK koje se mogu koristiti u CRISP-DM procesu na nivou specijalizovanih zadataka. Takođe predstavljamo okruženje za vođenje OZP projekata. Po našem mišljenju PUK obezbeđuju veću transparentnost procesa klasterovanja, u odnosu na dosadašnji način rada i analitičarima mogu dati bolje smernice nego generički zadaci dati u CRISP-DM procesu.

2. TIPIČNI PROBLEMI KOD KLASTEROVANJA I ČESTO KORIŠĆENA REŠENJA

Za vreme trajanja OZP projekta, po CRISP-DM standard, potrebno je izvršiti šest faza i odgovarajućih generičkih zadataka. U ovom radu analiziramo CRISP-DM process za tipičnu primenu klasterovanja. Zbog toga ćemo opisati tipične probleme i njihova rešenja za primenu klasterovanja u različitim fazama CRISP-DM procesa.

2.1 *Razumevanje poslovnog procesa*

Cilj ove faze projekta je da razume ciljeve projekta i zahteve iz perspektive poslovanja, kao i da definiše OZP problem koji bi trebao biti rešen unutar OZP procesa. Ova faza treba da obezbedi analitičaru dovoljno informacija koje će mu omogućiti dalju specifikaciju procesa OZP. U ovoj fazi generički i specijalizovani zadaci mogu biti povezani sa pod-problemima i PUK. Sa druge strane, u ovoj fazi ne postoji sistematičan pristup za rešavanje pod-problema. Slično istraživanje o primeni ponovo upotrebljivih komponenti u kompanijama je sprovedeno od strane (Coplien and Harrison, 2005).

2.2 *Razumevanje podataka*

U ovoj fazi, analiziraju se podaci da bi se odredile osobine podataka na osnovu koje će pomoći pri adekvatnoj pripremi podataka, izboru algoritma, kao i modela evaluacije. Ova faza je veoma bitna, zato što nerazumevanje osobina podataka, može da prouzrokuje dobijanje pogrešnih rezultata u fazi modelovanja. U CRISP-DM procesu predloženo je nekoliko generičkih zadataka:

1. Prikupljanje početnih podataka,
2. Opis podataka,
3. Istraživanje podataka i
4. Ocena kvaliteta podataka.

Ovi zadaci treba da obezbede ulaz za fazu pripreme podataka, u kojoj podaci treba da budu prilagođeni algoritmima.

2.2.1. *Prikupljanje početnih podataka*

Ovaj generički zadatak (pod-problem) se bavi učitavanjem i/ili spajanjem podataka koji će se koristiti pri analizi. Kod OZP alata, obično postoji više načina za učitavanje podataka (npr. Učitavanje iz tekstualnih datoteka, baza podataka, SPSS, SAS, Excel itd.). Ovi alati takođe predlažu specijalizovane zadatke (PUK) za spajanje podataka iz različitih izvora u jednu tabelu.

2.2.2. *Opis podataka*

Kod ovog zadatka podaci se grubo opisuju (npr. broj redova i atributa, tipovi atributa itd.). Kod OZP alata, specijalizovani zadaci omogućuju uvid u osnovne osobine učitanih podataka.

2.2.3 *Istraživanje podataka*

U ovom zadatku se izvršavaju jednostavne statističke analize kao što su raspodela atributa ili veze između atributa. Alati za OZP obezbeđuju brojne komponente za istraživanje podataka. Kod ovog zadatka je korisno analizirati podatke uz pomoć grafova (npr. *Multiplot, Distribution, Histogram*) koji su najčešće sastavni deo alata za OZP.

2.2.4 *Ocena kvaliteta podatak*

Nedostajuće vrednosti i "outlajeri" mogu uticati na kvalitet rezultata modela. Zbog toga je jako važno proceniti kvalitet podataka i predložiti akcije koje ga mogu poboljšati ukoliko je potrebno. Ovde se može koristiti nekoliko grafova i statistika. PUK opisane u prethodnim zadacima obezbeđuju informacije potrebne za izveštaj o kvalitetu podataka. Izveštaj o kvalitetu podataka, treba biti formalizovan, tako da korisniku bude jasno šta izveštaj treba da uključuje i kako bi trebao da koristi informacije iz izveštaja.

2.3 *Priprema podataka*

Posle razumevanja poslovnih zahteva (razumevanje poslovnog procesa) i ograničenja podataka (razumevanje podataka), podatke treba pripremiti, kako bi se mogli primenjivati algoritmi i kako bi rezultati bili smisleni i razumljivi.

CRISP-DM process predlaže nekoliko generičkih zadataka u ovoj fazi:

1. Izbor podataka,
2. Čišćenje podataka,
3. Kreiranje izvedenih podataka,
4. Integracija podataka i
5. Formatiranje podataka.

2.3.1. *Izbor podataka*

U ovom zadatku, analitičar treba da izabere podatke koji će biti korišćeni u izvršenju algoritma. Analitičar treba da odluči da li će raditi sa uzorkovanim podacima ili sa kompletnim skupom podataka. Tipični OZP alati nude nekoliko PUK za rešavanje ovog problema npr. *Select* (bira podskup podataka na osnovu definisanih uslova), *Sample* (bira određenu količinu podataka na osnovu neke metode uzorkovanja), *Balance* (ispravlja imbalans u podacima[1]). Sa druge strane, analitičar treba da odabere attribute koji će uticati na kreiranje modela. Neki od klasičnih modela za izbor najznačajnijih atributa su: analiza glavnih komponenti (PCA) i faktorska analiza (FA) (Costello & Osborne 2005). Postoje i drugi načini za izbor najznačajnijih atributa kao npr. Fodor (2002).

2.3.2 *Čišćenje podataka*

Obezbeđivanje kvaliteta podataka je jedan od ključnih faktora koji utiču na kvalitet OZP rešenja. Često algoritmi ne mogu da rade, ukoliko postoje nedostajuće vrednosti atributa. Postoji više načina za rešavanje ovog problema. Slogovi koji sadrže nedostajuće vrednosti mogu se ukloniti pre početka rada algoritma (npr. *Filter*). Pored toga moguće je izvršiti "imputaciju" nedostajućih vrednosti (ubacivanje neke vrednosti umesto nedostajuće). Postoje različite metode koje mogu poslužiti za rešavanje ovog zadatka (npr. srednja vrednost, srednja vrednost bliskih tačaka, medijana bliskih tačaka, linearna interpolacija itd.). OZP alati obezbeđuju različite komponente za rešavanje problema nedostajućih vrednosti (npr. *Filler*), a zadatak analitičara je da odredi koja komponenta najviše odgovara konkretnim podacima.

2.3.3 *Konstrukcija podataka*

Ponekad je potrebno izvoditi nova polja (atribute) ili transformisati numeričke u kategoričke podatke (npr. *Binning*). Ove transformacije imaju veliki uticaj na kvalitet i interpretabilnost rezultujućeg modela. Zbog toga je veoma bitno doneti odluku o tome, da li će se koristiti originalni ili izvedeni podaci, kao i koju metodu transformacije treba upotrebiti.

2.3.4. *Integracija podataka*

U ovom zadatku CRISP-DM predlaže specijalizovane zadatke za integraciju podataka. Ovaj korak je isti kao i u fazi razumevanja podataka, ali ovoga puta se primenjuje na drugim podacima.

2.3.5. *Formatiranje podataka*

Na kraju, podaci moraju da budu formatirani, tako da budu primenljivi u algoritmu. Redosled slogova u skupu podataka, takođe može da ima uticaj na izlaz algoritma. Za ovaj zadatak takođe postoje PUK (npr. *Sort*). Ukoliko OZP softver otkrije određeni atribut u drugačijem formatu, u odnosu na ono što je analitičar mislio, neophodno je prilagoditi format podataka algoritmu.

2.4 *Modelovanje*

U fazi modelovanja, treba izabrati odgovarajući OZP algoritam ili dizajnirati novi algoritam, kao što je predloženo u (Delibasic et al. 2009). Algoritam (izabran iz postojećih algoritama ili dizajniran) kasnije može biti korišćen za rešavanje poslovnog problema. CRISP-DM predlaže sledeće generičke zadatke:

1. Izbor tehniku modelovanja,
2. Generisanje okruženja za testiranje,
3. Izgradnja modela,
4. Izvršenje modela.

2.4.1 *Izbor tehnike modelovanja*

Svaki algoritam ima odredjene karakteristike, kao što su: zadatak algoritma (klasterovanje, klasifikacija itd.), podaci sa

[1] Imbalans se pojavljuje kada uzorak uključuje više slogova sa određenim vrednostima atributa, nego sa ostalima (npr. muški pol). Zbog ove pojave algoritmi često kreiraju pristrasne modele. Zbog toga se preporučuje da se ponovo uzorkuje postojeći uzorak, kako bi se imbalans smanjio na minimum.

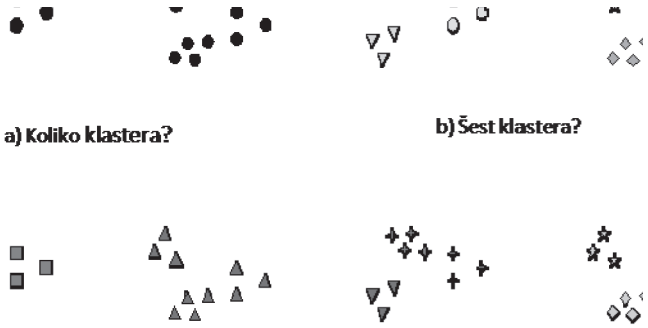
kojima algoritam može da radi (nominalni, kategorički, mešani), količine podataka sa kojima može da radi (Bentley, 1975).

U (Delibasic et al., 2009), za fazu modelovanja predložen je pristup belih kutija (*eng. white-box*) koji koristi dizajn uz pomoć ponovo upotrebljivih komponenti. Kod ovog pristupa za dizajn algoritama ponuđen je skup pod-problema koji treba da budu rešeni izborom ponovo upotrebljivih komponenti. Na taj način je moguće rekonstruisati postojeće algoritme ili kreirati hibridne algoritme koji su prilagođeni konkretnim podacima. Ovi pod-problemi su:

1. Definicija broja klastera,
2. Inicijalizacija centroida,
3. Definicija mere odstojanja,
4. Izračunavanje centroida i
5. Deljenje klastera.

2.4.1.1 Definicija broja klastera

Definicija broja klastera (izbor parametra k), najčešće predstavlja odluku baziranu na prethodno stečenom znanju o problemu i podacima, pretpostavkama i praktičnom iskustvu (Slika 2). Ovaj zadatak je jako težak ukoliko su podaci višedimenzioni, čak iako su dobro razdvojeni (Milligan & Cooper, 1985).



Slika 2. – *Određivanje pravog broja klastera*

K-means (MacQueen, 1967) pretpostavlja da je broj klastera poznat pre početka izvršenja algoritma.

Hijerarhijski algoritmi klasterovanja takođe ne pokušavaju da odrede pravi broj klastera, već ovu odluku prepuštaju korisnicima.Algoritmi zasnovani na gustini sami određuju broj klastera, ali korisnik treba da odredi druge parametre algoritma, koji utiču na konačan broj klastera.

Kao jedno od mogućih rešenja za ovaj problem je *silhouette* graf. On prikazuje siluete svih klastera koji se nalaze jedan do drugog, tako da se kvalitet klstera može uporediti (Rousseeuw, 1987). Ukoliko je širina siluete (*silhouette width*) s_i blizu 1, onda se objekat i dodeljuje pravom klasteru (na slici 3, to su objekti D i B). Ukoliko je s_i=0 tada se ne zna da li objekat i pripada ovom klasteru ili nekom drugom (na slici 3 objekti GR). Ukoliko je s_i negativno, objekat je svrstan u pogrešan klaster (objekat P na slici 3). *Silhouette* graf je jako koristan za određivanje pravog broja klastera. Korisnici mogu puštati algoritam po nekoliko puta, svaki put sa različitim brojem klastera i onda porediti siluet grafove. Prosečna širina siluete se može koristiti za izbor "optimalnog" broja klastera, tako što se bira onaj, sa najvećom vrednošću širine siluete. Da bi se *silhouette* graf mogao koristiti kao mera kvaliteta klastera, algoritam treba da izračuna meru pripadnosti objekta određenom klasteru.

Slika 3. – ***Silhouette*** **graf**

U slučaju hijerarhijskog klasterovanja, za vizualizaciju deljenja podataka se može koristiti *dendrogram*, tako da korisnik može da odredi koliko klastera želi da kreira. Još jedan koristan dijagram je *banner* graf, koji prikazuje hijerarhiju klastera. *Banner* iscrtava odstojanja na kojima se objekti i klasteri spajaju (Rousseeuw 1986).

(Pelleg & Moore, 2000) predlažu hijerarhijsko-divizionu strategiju za particionisanje podataka, da bi našli pravi broj klastera. Oni pokušavaju da odrede pravi broj klastera, tako što kreću od minimalnog pretpostavljenog broja, a zatim iterativno dele klastere sve dok se ne dostigne *BIC* (Bayesian Information Criteria), mera kvaliteta, koja služi kao kriterijum zaustavljanja.

Bischoff et al. (1999) koriste minimalnu opisnu dužinu (minimum description length - *MDL*) da bi odredili koliko model odgovara podacima. Algoritam počinje sa visokom vrednošću k i postepeno ga smanjuje, ukoliko se smanji broj centara, onda se smanjuje i opisna dužina.

Hamerly and Elkan (2003) opisuju algoritam G-means. Algoritam počinje sa malim brojem k (centara klastera) i povećava broj centara ukoliko podaci dodeljeni centru klastera nemaju normalnu raspodelu, koja se testira *Anderson-Darling* statistikom.

2.4.1.2 Inicijalizacija centroida

Postoji puno načina da se inicijalizuje k predstavnika klastera (u daljem tekstu centroidi). K means koristi slučajnu (*random*) inicijalizaciju centroida. Iz skupa podataka slučajno se k objekata koji predstavljaju početne centroide klastera. Ovo je verovatno najgori način inicijalizacije, pošto izbor outlajera (*eng. outliers*) za centroide, može jako loše uticati na rezultujuće klastere.

(Arthur & Vassilvitski 2007) predlažu *k-means* ++ algoritam, koji inicijalizuje k centroida na sledeći način. Prvi centro-

id se bira slučajno. Drugi je izabran kao tačka koja je najdalja u odnosu na prvi centroid. Svi ostali se biraju kao najdalji od već izabranih.

(Kaufman & Rousseeuw, 1990) u algoritmu DIANA predlažu način deljenja skupa podataka na dva dela (klastera). Ovaj pristup se koristi u hijerarhijsko-divizionim klasterovanju, ali se takođe može koristiti za određivanje dva centroida (k=2).

(Hammerly & Elkan, 2003) predlažu dva načina za inicijalizaciju centroida za slučaj kada je k=2. Centroidi bi trebalo da budu inicijalizovani kao dve tačke pozicionirane oko prosečne tačke klastera *c* (izračunate kao prosečna vrednost svih tačaka skupa podataka). Dve tačke su $c \pm m$, gde se *m* može definisati na dva načina: prvo, može biti izabrano kao relativno mala vrednost u poredjenju sa prosečnom udaljenošću tačaka od tačke c. Drugo, može biti izabrana kao $m = s*(2\lambda/\pi)^{1/2}$ gde *s* predstavlja prvu glavnu komponentu podataka, koja ima sopstvenu vrednost λ.

(Ding & He, 2004) predlažu "blizu-optimalan" način za inicijalizaciju k=2 centroida uz pomoć analize glavnih komponenti (*PCA*). Oni takođe proračunavaju prvu glavnu komponentu i množe centrirani skup (svaka kolona podeljena sa njenom prosečnom vrednošću) podataka i tako kreiraju vektor kolonu za svaki objekat. Ukoliko je vrednost unutar ovog vektora 0 tada se objekat dodeljuje prvom klasteru, u suprotnom, drugom klasteru.

(Pelleg & Moore, 2000) predlažu još jedan način za inicijalizaciju k=2 centroida u njihovom *X-means* algoritmu.U ovom pristupu, takođe se određuje *m* da bi se odredile dve tačke kao $c \pm m$, pri čemu je *m* takođe definisano kao prosečna udaljenost objekata od centroida *c*. Jedina razlika je u tome što je vector $c \pm m$ postavljen u slučajno izabranom smeru.

2.4.1.3 Merenje odstojanja od centroida i dodela objekta klasteru

Ima puno načina za merenje udaljenosti. Najčešće korišćena metrika je *Euklidska*. Međutim, euklidska metrika nije primenljiva na sve tipove podataka i ne mora uvek biti adekvatna mera za konkretne podatke. Analitičar podataka može izabrati i druge mere odstojanja da bi odredio udaljenosti objekata od centroida.

Neke od popularnih mera odstojanja za numeričke podatke su:

1. *Cosine* odstojanje,
2. *I-divergence*,
3. *City block* odstojanje, itd.

Kada su sva odstojanja izračunata, svi objekti se dodeljuju najbližim centroidima. Sve tačke pripisane centroidu kao i sam centroid, definišu klaster.

2.4.1.4 Preračunavanje centroida

Kada su određeni klasteri u prvoj iteraciji, moguće je premeštati objekte u druge klastere kako bi se dobili što bolji rezultati. Ovo se može uraditi određivanjem novih centroida klastera. Pitanje je da li novi centar treba da bude medijana ili aritmetička sredina objekata u klasteru. Drugo, treba odrediti da li novi centroid treba da bude objekat unutar klastera ili novi, preračunati objekat.

2.4.1.5 Deljenje klastera

Da bi se dobili što bolji klasteri moguće je primeniti hijerarhijsko-divizionu strategiju da bi se originalni klasteri podelili na više klastera. Tada, analitičar može da testira da li su klasteri deca kvalitetniji od klastera roditelja. Postoji puno metoda kojima se može testirati kvalitet klastera.

(Pelleg & Moore, 2000) u X-means algoritmu koriste Bajesov Informacioni Kriterijum (BIC) da bi evaluirali klastere. Ukoliko je suma BIC za decu klastere veća od BIC clastera roditelja, klaster treba podeliti, u suprotnom, roditelj se ne deli.

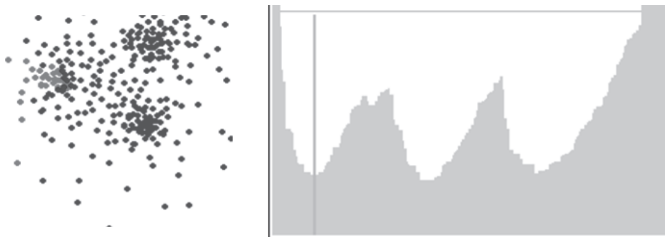
(Hammerly & Elkan, 2003) koriste Anderson-Darling (A-D) statistiku da bi proverili da li su klasteri deca, bolje normalno rasporedjeni nego klasteri roditelji. Ukoliko deca imaju bolji rezultat na A-D testu, klaster treba podeliti, u suprotnom nema podele.

(Rousseeuw, 1987) predlaže *silhouette* grafove za evaluaciju kvaliteta klastera. Za svaki objekat u klasteru, moguće je definisati meru koja definiše koliko objekat odgovara klasteru u poređenju sa klasterom koji predstavlja najbližeg suseda. Ukoliko objekat postigne vrednost veću od 0.6 obično je dobro klasterovan, u suprotnom ga treba smestiti u drugi klaster. Ova mera se može koristiti i za deljenje klastera. Ukoliko deca klasteri postižu bolji kvalitet *silhouette* mere od roditelja, onda se vrši deljenje klastera.

2.4.2 *Generisanje okruženja za testiranje*

Pre nego što se model primeni na konkretne podatke, analitičar treba da odluči na koji način će testirati kvalitet i validnost modela. U slučaju klasterovanja mogu biti korišćene npr. *silhouette* graf (Rousseeuw, 1987) ili Anderson Darling statistika (Hammerly & Elkan, 2003).

Kompaktnost klastera se može meriti različitim merama, npr. gustina. Kod ocene uz pomoć gustine, objekti u klasteru treba da imaju veću gustinu nego oni van klastera. Korisnik treba da definiše parametre za takav algoritam. Da bi lakše odredio parametre korisnik može da analizira OPTICS graf (Ankerst et.al. 1999, slika. 4).



Slika 4. – ***Graf gustine korišćen u OPTICS za konkretan skup podataka (http:\\www.dbs.informatik.uni-muenchen.de Forschung\KDDClustering\OPTICSDemo)***

Oblik mogućih klastera takođe utiče na rezultate različitih algoritma. (Josiger & Kirchner 2003) su pokazali da se neki algoritmi ponašaju veoma loše sa objektima koji su raspoređeni u ne-sferične oblike. Takođe mnogi algoritmi ne rade dobro sa podacima koji imaju šum.

2.4.3 *Izgradnja modela*

Posle definicije algoritma koji će se koristiti, algoritam se može puštati sa različitim parametrima da bi se izgradio model. Ukoliko je algoritam izgrađen u generičkom zadatku "izbor modela" uz pomoć PUK, problem izbora broja klastera i mere odstojanja je već rešen (pogledati 2.4.1).

2.4.4 *Procena modela*

U ovom zadatku se vrši sumarizovanje rezultata, prikaz kvaliteta modela i rangiranje na osnovu tehničkih karakteristika. U ovom generičkom zadatku sledeće dve komponente su korisne:

1.1.1.1 Tipično odstojanje, tipična vrednost klastera

Ovi zadaci su jako važni ukoliko su klasteri opisani. Kod opisa klastera, važno je uporediti objekte jednog klastera sa svim ostalim objektima da bi se odredile karakteristike objekata konkretnog klastera.

1.1.1.2 Stabla odlučivanje za opisivanje klastera

Dobar način za opisivanje klastera je kombinacija klastering algoritama sa stablima odlučivanja. Kada je klaster algoritam izvršen, moguće je koristiti stablo odlučivanja za klasifikaciju objekata, pri čemu se kao izlazna promenljiva koristi klasni atribut. Na osnovu pravila generisanih stablom odlučivanja moguće je objasniti zašto određeni objekat pripada određenom klasteru.

2.5 EVALUACIJA

Glavni cilj ovog koraka je da:

1. Evaluira rezultate u smislu poslovnih zahteva,
2. Pregleda ceo proces da bi odlučili koje korake treba ponoviti ili šta je propušteno,
3. Odredi sledeći korak: da li će projekat ući u fazu primene ili su potrebne dodatne iteracije.

2.6 PRIMENA

Tokom faze primene, pravila i paterni pronađeni u podacima se koriste za rešavanje poslovnih problema definisanih na početku procesa OZP. Sastoji se iz sledećih zadataka :

1. Primena plana,
2. Praćenje plana i održavanje,
3. Kreiranje finalnog izveštaja,
4. Ocena projekta.

3. SKLADIŠTE PONOVO UPOTREBLJIVIH KOMPONENTI

U poglavlju 2, odredili smo specijalizovane zadatke (PUK) za rešavanje generičkih zadataka definisanih CRISP-DM procesom. Analizom CRISP-DM procesa, predložili smo PUK za faze razumevanja podataka, pripreme podataka i modelovanja. Ostale faze CRISP-DM procesa nisu u fokusu ovog rada. Komponente koje smo definisali su sumarizovane u Tabeli 2. PUK koje su definisane u literature ili u OZP alatima, bi trebalo da postoje skladištima, koja se mogu upotrebljavati u procesu OZP.

Naša hipoteza je da je moguće formalizovati process OZP, povezivanjem PUK različitih faza CRISP-DM procesa (predprocesiranje, modelovanje i post-procesiranje). Uz pomoć PUK moguće je formalno opisati process otkrivanja znanja. Takođe predlažemo da analitičari koriste PUK kako bi bolje razumeli OZP process.

Zadatak analitičara je da pronađe pravu kombinaciju PUK. Kombinovanje PUK u procesu pronalaženja znanja iz podataka predstavlja optimizacioni problem. Verujemo da će u budućnosti, korisniku biti na raspolaganju moćni optimizacioni softveri koji će mu davati podršku u procesu otkrivanja znanja. Skladišta PUK treba da igraju glavnu ulogu u ovom zadatku. One treba da budu formalizovane, tako da se lako mogu integrisati u OZP alate. Takođe trebaju biti dobro opisane, npr. karakteristike PUK treba da budu raspoložive i povezane sa karakteristikama skupa podataka i zadataka OZP. Dizajn OZP je prvi korak ka pomenutim ciljevima za budućnost.

Tabela 2. – *Predložene ponovo upotrebljive komponente (PUK) za podršku fazama CRISP-DM procesa*

CRISP-DM faze	Pod-problemi	Ponovo upotrebljive komponente
Razumevanje podataka	Prikupljanje podataka	*Load, Merge*
	Opis podataka	Jednostavne statističke metode
	Istraživanje podataka	*Plot, Multiplot, Histogram, Distribution, Detailed statistics*
	Ocena kvaliteta podataka	Izveštaj o kvalitetu pogodaka
Priprema podataka	Izbor podataka	*Select, Sample, Balance, PCA/FA*
	Čišćenje podataka	*Filter, Filler*
	Konstrukcija podataka	*Derive, Binning*
	Integracija podataka	*Merge*
	Formatiranje podataka	*Sort*

Modelovanje	Izbor tehnike modelovanja	Definicija broja klastera PUK, Inicijalizacija centroida PUK, Mera odstojanja PUK, Preračunavanje centroida PUK, Deljenje klastera PUK
	Generisanje okruženja za testiranje	*BIC, Anderson-Darling statistic, Silhouette Plot, OPTICS graph*
	Izgradnja modela	Mera odstojanja PUK, Definicija broja klastera PUK
	Ocena modela	Tipična odstojanja i vrednosti, Model stable odlučivanja

LITERATURA

[1] Ankerst, M. & Breunig, M. K. H. S. J. (1999). *OPTICS: Ordering points to identify clustering structure*. New York: ACM Press.

[2] Arthur, D. & Vassilivitskii, S. (2007). k-means++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (*pp. 1027-1035).

[3] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM , 18*(9), 509-517.

[4] Berkhin, P. (2006). A Survey of Clustering Data Mining Techniques. In J. Kogan, C. Nicholas & M. Teboulle (Ed.), *Grouping multidimensional data - recent advances in clustering (*pp. 25-71). Berlin, Heidelberg: Springer.

[5] Bischof, H., Leonardis, A. & Selb, A. (1999). MDL principle for robust vector quantisation. *Pattern Analysis and Applications* (2), 59-72.

[6] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R. (2000). *CRISP-DM 1.0. Step-by-Step Data Mining Guide.*

[7] Coplien, J.O., Harrison, N.B. (2005). *Organizational patterns of agile software development*. Upper Saddle River, NJ: Pearson Prentice Hall.

[8] Costello, A. B. & Osborne, J. W. (2005). Best Practices in Exploratory Factor Analysis: Four Recommendations for Getting the Most From Your Analysis. *Practical Assesment, Research & Evaluation, 10* (7).

[9] Delibašic, B., Kirchner, K., Ruhland, J., Jovanovic, M. & Vukicevic, M. (2009). Reusable components for partitioning clustering algorithms. *Artificial Intelligence Review, 32* (1-4), 59-75.

[10] Ding, C. & He, X. (2004). K-means clustering via principal component analysis. *Proceedings of the twenty-first international conference on Machine learning (*pp. 29-36). Banff, Alberta, Canada.

[11] Ester, M., Kriegel, H., Sander, J. & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd ACM SIGKDD (*pp. 226-231). Portland, Oregon.

[12] Fisher, D. (1987). Knowledge aquisition via incremental conceptual clustering. *Machine Learning, 2* (2), 139-172.

[13] Fodor, I. K. (2002). A Survey of Dimension Reduction Techniques. *Technical Report*, Lawrence Livermore National Lab, USA.

[14] Hamerly, G. & Elkan, C. (2003). Learning the k in k-means. In S. Thrun, L. Saul & B. Schölkopf (Ed.), *Advances in Neural Information Processing Systems (*MIT Press).

[15] Han, J. & Kamber, M. (2006). *Data mining*. Amsterdam: Elsevier/Morgan Kaufmann.

[16] Josiger, M. & Kirchner, K. (2003). Moderne Clusteralgorithmen - eine vergleichende Analyse auf zweidimensionalen Daten. In A. Hotho & G. Stumme (Ed.), *Proceedings Fachgruppentreffen Maschinelles Lernen (FGML 2003) (*pp. 80-84).

[17] Kaufman, L. & Rousseeuw, P. J. (1990). *Finding groups in data*. New York: John Wiley & Sons.

[18] Kurgan, L. A. & Musilek, P. (2006). A survey of Knowledge Discovery and Data Mining process models. *The Knowledge Engineering Review, 21* (1), 1-24.

[19] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability (*Vol.1, pp. 281-297).

[20] Milligan, G. W. & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data. *Psychometrika, 50* (2), 159-179.

[21] Pelleg, D. & Moore, A. (2000). *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*

[22] Rousseeuw, P. J. (1986). A visual display for hierarchical classification. In E. Diday, Y. Escoufier, L. Lebart, J. Pages, Y. Schektman & R. Tomassone (Hrsg.), *Data Analysis and Informatics 4 (*S. 743-748). Amsterdam: North Holland.

[23] Rousseeuw, P. J. (1987). Silhouettes – A graphical aid to the interpretation and validation of cluster-analysis. *Journal of Computational and Applied Mathematics , 20*, 53-65.

[24] Sametinger, J. (1997). *Software Engineering with Reusable Components*. Springer.

[25] Sheikholeslami, G., Chatterjee, S. & Zhang, A. (1998). WaveCluster: A multiresolution clustering approach for very large spatial databases. *Proc. 24th VLDB Conf. (*pp. 428-439). New York, USA.

[26] Wang, W., Yang, J. & Muntz, R. M. (1997). STING: A Statistical Information Grid Approach to Spatial Data Mining. In M. Jarke (Ed.), *Proceedings of the Twenty-third International Conference on Very Large Data Bases - Athens, Greece, 26 - 29 August, 1997 (*pp. 186-195). San Francisco, Calif: Morgan Kaufmann.

[27] Wirth, R. & Hipp, J. (2000). CRISP-DM: Towards a standard process modell for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining (*pp. 29-39).

[28] Xu, R. & Wunsch, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks, 16* (3), 654-678.

Kathrin Kirchner, Fakultet za ekonomiju i poslovnu administraciju Friedrich Schiller, Univerzitet u Jeni, Nemačka
Oblasti interesovanja: poslovna informatika, spo, gis, dejta majning

Boris Delibašić, Fakultet organizacionih nauka, Univerzitet u Beogradu, Srbija
Oblasti interesovanja: Poslovna inteligencija, skladištenje podataka, dejta majning, menadžment znanja, poslovno odlučivanje.

Milan Vukićević, Fakultet organizacionih nauka, Univerzitet u Beogradu, Srbija
Oblasti interesovanja: Poslovna inteligencija, skladištenje podataka, dejta majning, menadžment znanja, poslovno odlučivanje.