Univerzitet u Nišu
Elektronski fakultet

Seminarski rad

Rad sa nedostajućim podacima

Predmet: Prikupljanje i predobrada podataka za mašinsko učenje

Niš, 2023.

Mentor: Aleksandar Stanimirović

Student: Andrija Malbaša

Sadržaj

1. Uvod

Prilikom rada sa realnim podacima, u procesu prikupljanja podataka informacije se često gube. Ovaj gubitak informacija je izazvan prisustvom nedostajućih vrednosti (NV u nastavku ovog rada). NV se mogu pojaviti iz mnogih razloga, među kojima su greške prilikom ručnog unosa podataka, greška u opremi za prikupljanje podataka (česta pojava u industrijskim procesima), greške u merenju itd. Prisustvo ovih imperfekcija obično zahteva preprocesiranje tokom koga se podaci pripremaju i prečišćavaju kako bi bili adekvatni za ekstrakciju znanja tj. zaključaka iz njih. Objavljeno je nekoliko načina obrađivanja NV u zadacima prikupljanja i obrade podataka među kojima je najjednostavniji odbacivanje primera (instanci) koji ih sadrže. Međutim, ovaj metod je upotrebljiv samo onda kada podaci sadrže mali broj primera koji sadrže NV i kada analiza svih primera neće dovesti do sklonosti ka progresnom zaključivanju tj. pristrasnosti (*bias*). Kada to nije slučaj, NV mogu znatno otežati analizu podataka. Neadekvatno rukovanje NV u analizi veoma često izaziva pristrasnost, što dovodi do pogrešnih zaključaka.
U većini slučajeva, NV se mogu obraditi na tri načina:

1) Prvi pristup je odbacivanje primera koji sadrže NV. To podrazumeva odbacivanje opservacija koje sadrže NV kao i atributa koji sadrže previše NV.
2) Drugi pristup je korišćenje procedura maksimalne verovatnoće, gde se procenjuju parametri modela za celokupnu porciju podataka, a kasnije se koriste za imputaciju pomoću uzorkovanja.
3) Treći pristup je korišćenje klase procedura za imputaciju NV koja ima za cilj popunjavanje NV procenjenim vrednostima. U većini slučajeva, atributi skupa podataka nisu međusobno nezavisni i kroz identifikaciju veza među atributima se NV mogu utvrditi.

2. Pretpostavke i mehanizmi nedostajanja vrednosti

Najpre je potrebno utvrditi mehanizme koji dovode do NV. Pretpostavke koje pravimo o mehanizmima nedostajanja i obrascima pojavljivanja NV mogu uticati na metod koji će se primeniti na rukovanje NV. Kada razmatramo mehanizam nedostajanja, treba uzeti u obzir raspodelu verovatnoće koja leži ispod instanci pravougaonih skupova podataka, gde redovi označavaju različite slučajeve ili instance, a kolone atribute ili promenljive.

Neka X označava $n \times m$ pravougaonu matricu podataka, gde je x_i i-ti red matrice X. Ukoliko pretpostavimo nezavisnu i identičnu raspodelu instanci (veoma česta pretpostavka), funkcija verovatnoće podataka se može predstaviti na sledeći način:

$$P(X|\theta) = \prod_{i=1}^{n} f(x_i|\theta)$$

, gde je f funkcija verovatnoće jedne instance, a θ parametri modela koji daje tu konkretnu instancu. Glavni problem je što su vrednosti parametara θ za date podatke često nepoznati, zbog čega se često razmatraju raspodele koje se često mogu naći u prirodi i čija su svojstva poznata. Tri raspodele koje se najčešće razmatraju su:

- Multivarijatna normalna raspodela u slučaju parametara koji sadrže kontinualne vrednosti;
- Multinominalni model za unakrsno klasifikovanje kategoričkih podataka kada parametri sadrže nominalne vrednosti;
- Mešani modeli za podatke sa parametrima koji sadrže i nominalne i kontinualne vrednosti

2.1 Missing at random (MAR)

Ukoliko sa X_{obs} i X_{mis} označimo redom posmatrani (opservacije) i nedostajući deo matrice X, tako da je $X = (X_{obs,} X_{mis})$, možemo na intuitivan način da predstavimo pojam MAR (*missing at random*) tj. nasumično nedostajanje. Informativno govoreći, kada verovatnoća da opservacija nedostaje može da zavisi od X_{obs} ali ne i od X_{mis} , može se reći da podaci nedostaju nasumično. U slučaju MAR mehanizma nedostajanja, kada su zadate vrednost ili vrednosti skupa atributa koji pripadaju X_{obs}, raspodela ostalih atributa je ista među posmatranim slučajevima kao što je i među nedostajućim. Pretpostavimo da raspolažemo matricom P reda $n \times p$ čije su vrednosti 1 ili 0 kada su elementi posmatrani i nedostajući, respektivno. Raspodela od B bi trebalo da se odnosi na X i na neke nepoznate parametre ζ, tako da raspolažemo modelom verovatnoće za B koji se označava sa $P(B|X, \zeta)$. Pretpostaviti da je mehanizam nedostajanja MAR bi značilo da ova raspodela ne zavisi od X_{mis}:

$$P(X|X_{obs}, X_{mis}, \zeta) = P(B|X_{obs}, \zeta)$$

2.2 Missing completley at random (MCAR)

Poseban slučaj MAR, je MCAR (*Missing completley at random*). Kod ovog mehanizma nedostajanja, raspodela verovatnoće prisustva NV kod nekog atributa za neki primer ne zavisi ni od posmatranih ni od neposmatranih podataka:

$$P(X|X_{obs}, X_{mis}, \zeta) = P(B|\zeta)$$

Ovo implicira da su razlozi za prisustvo NV potpuno nezavisni od podataka. Konsekvecijalno, mogu se ignorisati mnoge kompleksnosti koje proizilaze zbog NV, izuzev očiglednog gubitka informacija. MCAR je restriktivniji od MAR. MAR samo zahteva da se sve NV ponašaju kao nasumični uzorci svih vrednosti u nekim određenim potklasama koje su definisane posmatranim podacima. Na taj način MAR dozvoljava da verovatnoća nedostajanja informacije zavisi od same nje, ali isključivo indirektno kroz posmatrane vrednosti. Razlika između MAR i MCAR se može razumeti i intuitivno preko primera: vaga za merenje težine može proizvesti više nedostajućih vrednosti kada se postavi na mekanu površinu nego na tvrdu. Takvi podaci nisu MCAR. Međutim, ukoliko nam je poznat tip površine na koju je postavljena vaga i ako možemo pretpostaviti MCAR u okviru tipa površine, onda su podaci MAR. MAR je opštiji i realističniji od MCAR. **Moderni metodi za rukovanje NV uglavnom počinju pretpostavkom o MAR.**

2.3 Not missing at random (NMAR)

Treći, i najteži slučaj, nastaje kada MAR ne važi zbog zavisnosti NV i od posmatranih vrednosti i od nedostajućih vrednosti. Tada bi važilo:

$$P(X|X_{obs}, X_{mis}, \zeta) = P(B|X_{obs}, X_{mis}, \zeta)$$

Ovaj model se uobičajeno naziva NMAR (*not missing at random)* ili MNAR (*missing not at random*)*.* Kada su podaci NMAR, nedostajanje podataka je povezano sa događajima i faktorima koji nisu izmereni tj. razmatrani. U tom slučaju, najbolji pristup bi bio ponovo pokupati prikupiti te podatke.

2.4 Utvrđivanje mehanizma nedostajanja

Razlika je najpre objašnjena preko jednostavnog primera u kome se od ispitanika koji su predstavljeni preko svog IQ-a traži da pruže ocenu za svoj Job Performance.

	Job Performance Ratings			
IQ	Complete	MCAR	MAR	MNAR
78	9	Missing	Missing	9
84	13	13	Missing	13
84	10	Missing	Missing	10
85	8	8	Missing	Missing
87	7	7	Missing	Missing
91	7	7	Missing	Missing
92	9	9	9	9
94	9	9	9	9
94	11	11	11	11
96	7	Missing	7	Missing
99	7	7	7	Missing
105	10	10	10	Missing
105	11	11	11	11
106	15	15	15	15
108	10	10	10	10
112	10	Missing	10	10
113	12	12	12	12
115	14	14	14	14
118	16	16	16	16
134	12	Missing	12	12

Slika 1. Ilustracija mehanizama nedostajanja

Na slici 1 se može videti razlika između MCAR, MAR i NMAR. U MCAR koloni ne postoji veza između IQ i nedostajanja podataka. Podjednako je verovatno da nedostaje Job Performance ocena i za visoke i za niske IQ. U MAR koloni može se zaključiti da nedostaju Job Performance ocene samo za nekoliko najnižih IQ. U NMAR koloni, IQ se ne može koristiti da se predstavi obrazac nedostajanja zato što se na osnovu IQ ne može zaključiti zašto Job Performance ocene nedostaju.

Definisanje modela nedostajanja je izazovno, budući da jedini način da se pribavi nepristrasna procena je da se modelira nedostajanje. Ovo je veoma kompleksan zadatak jer je potreno kreirati model koji bi obračunao nedostajuće podatke koji je kasnije potrebno uvrstiti u kompleksniji model koji se koristi za procenu NV. Najćešći pristup je najpre utvrditi da li je mehanizam nedostajanja MAR ili MCAR.

3. Jednostavni pristupi u rukovanju sa nedostajućim podacima

Najjednostavniji metodi za rukovanje NV ne uzimaju u obzir mehanizme nedostajanja.

3.1 Do not impute (DNI)

Ovo je najjednostavniji pristup u rukovanju NV. Ovim pristupom sve NV ostaju nezamenjene, tako da algoritmi za Data Mining moraju da koriste svoje pordrazumevane strategije za rukovanje NV, ako uopšte poseduju takve strategije. Često je cilj ovog pristupa utvrditi da li metode za imputaciju omogućuju metodama za klasifikaciju bolje performanse nego nego kada bi se koristili originalni podaci. Kao alternativa za metode učenja koje ne mogu da rade sa NV (ako NV sadrže neku specijalnu vrednost npr.) je metoda konvertovanja NV u novu numeričku vrednost, ali takav pojednostavljeni metod je dovodio do ozbiljnih problema u izvođenju zaključaka.

3.2 Ignore missing (IM)

Veoma čest pristup u specijalizovanoj literaturi. Ovim pristupom se odbacuju sve instance koje imaju barem jednu NV. Ovo je prihvatljivo rešenje ukoliko je sigurno da podaci u skupu podataka koji se analizira nisu MNAR i ukoliko je broj instanci dovoljno velik da odbacivanje nekoliko njih neće izvazvati gubitak sposobnosti generalizacije modela koji se grade nad tim podacima.

Ukoliko su podaci MNAR (nedostajanje podatka zavisi od nekog fenomena iz realnog okruženja u stvarnom životu), brisanje instanci koje sadrže NV će izazvati visok nivo pristrasnosti u modelu. Utvrđivanje ovog problema zahteva domensko znanje i uglavnom jedini način da se on proveri je kroz ručnu inspekciju.

Iako se ovim metodom značajno umanjuje veličina uzorka koji je dostupan za analizu, njegova primena ima nekoliko značajnih prednosti. Konkretno, ukoliko su podaci MCAR, daje nepristrasne procene parametara. Nažalost, čak i ako su podaci MCAR prisutan je gubitak informacija, pogotovo ako treba odbaciti značajan broj instanci. Kada podaci nisu MCAR, ovaj metod daje pristrasne rezultate (npr. ukoliko individue sa niskim prihodom često ne objave koliki je taj prihod, resultujuća srednja vrednost prihoda svih individua ukazuje da većina individua ima mnogo viši prihod nego što je to u realnosti).

3.3 Zero imputation (ZM)

Ova metoda podrazumeva zamenu nedostajućih vrednosti nulama. Ovo je jednostavan i čest metod u radu sa NV, pogotovo u slučajevima kada su nule značajne ili prikladne zamene za NV. Mnoga istraživanja su eksperimentalno potvrdila da imputacija nulama rezultuje u značajnom padu performansi modela mašinskog učenja. Imputacija nulama

može uvesti pristrasnost u slučajevima kada nedostajuće vrednosti nisu zaista nule i iskriviti raspodelu feature-a. To vodi ka tome da model uči pogrešne obrasce i proizvodi netačne predikcije. Takođe, ovaj metod može promeniti vezu među atributima u skupu podataka što vodi ka pogrešnim korelacijama među atributima. Iako sam po sebi ovaj metod nije među najefektivnijima za rukovanje NV, često je prisutan kao inicijalni korak kod metoda imputacije kod kojih je neophodno da podaci ne sadrže NV (što je slučaj kod algoritama za klasterizaciju npr.).

3.4 Most common (MC)

MC predstavlja metodu zamene NV globalno najčešćom vrednošću za nominalne atribute i globalno srednjom vrednošću za numeričke atribute, što je prihavtljivo rešenje ukoliko previše instanci sadrži NV.

3.5 Concept Most Common (CMC)

CMC je varijanta MC. Za nominalne atribute se NV zamenjuju najčešćom vrednošću za taj atribut dok za realne atribute zamenjuju srednjom vrednošću za taj atribut, pri čemu se razmatraju samo instance u okviru iste klase kao i referentna instanca (instanca čije se NV zamenjuju). U ovu kategoriju se mogu svrstati neki stariji Data Mining pristupi. Jedan od tih pristupa je i Hot Deck imputacija koja je stara preko 50 godina, i u svoje vreme često korišćena.

4. Hot deck imputacija

Iako Hot deck ne pretpostavlja ništa o mehanizmima nedostajanja, dodatno je obrađen u ovom radu zato što je često korišćen od strane praktičara istraživanja i anketa za obrađivanje „praznih odgovora“ tj. NV.

Hot Deck imputacija podrazumeva zamenu nedostajućih vrednosti jedne ili više promenljivih za „primaoca“ posmatranim vrednostima „donatora“ koji je sličan primaocu u odnosu na karakteristike koje su posmatrane za oba slučaja. U nekim verzijama, donator se nasumično bira iz skupa potencijalnih donatora (donor pool). Ove metode se nazivaju Random Hot Deck metode. U drugim verzijama, identifikuje se jedan donator i vrednosti za imputaciju se uzimaju iz njega. Donator se obično bira uz pomoć „Nearest Neighbour“ algoritma, na osnovu neke metrike udaljenosti među tačkama podataka. Ove metode se nazivaju Deterministic Hot Deck metode. Prisutni su i metodi koji imputiraju vrednosti sumarne statistike skupa donatora kao što su srednja vrednost ili medijana umesto individualnih vrednosti, ali te metode se ne smatraju Hot Deck metodama.

Postoji nekoliko razloga zbog kojih su Hot Deck metode popularne među praktičarima istraživanja. Kao i sa svim metodama imputacije, rezultat je pravougaoni skup podataka

koji mogu koristiti drugi analtičari podataka koji na dalje ne moraju obraćati pažnju na NV i mogu koristiti jednostavnije metode u svojim analizama. Njime se izbegava problem nedoslednosti među korisnicima do kojeg može doći kada analitičari koriste sopstvene strategije za NV. Hot Deck se ne oslanja na model fitting da bi se imputirala varijabla zbog čega je potencijalno manje osetljiva na propuste (pogrešnu specifikaciju) modela od metoda imputacije zasnovanog na parametrijskom modelu (kao što je imputacija regresijom). Međutim, Hot Deck i dalje pravi par implicitnih pretpostavki koje se odnose na izbor odgovarajuće metrike za uparivanje donatora sa primaocem. Još jedna dobra karakteristika ovog metoda je da se mogu imputirati samo verodostojne vrednosti, s obzirom da one dolaze od posmatranih informacija iz skupa donatora. To znači da je moguć dobitak u efikasnosti u odnosu na analizu kompletnog slučaja, budući da se informacije u nepotupunim slučajevima zadržavaju. Takođe može dovesti do smanjenja non-response pristrasnosti koja se javlja kada učesnici istraživanja ne žele ili ne mogu da odgovore na pitanja.

4.1 Kreiranje skupova donatora

Donor pools, takođe poznati i kao imputacione klase ili ćelije za prilagođavanje, formiraju se na osnovu pomoćnih promenljivih koje su posmatrane i za donatore i za primaoce. Skup donatora kreira u dva koraka:

1) Adjustment cells metod
2) Uparivanje na osnovu metrike

4.1.1 Adjustment cells:

Zbog jasnoće, fokusiramo se na upotrebu kovarijantnih informacija x(promenljive koje su potencijalno povezane i sa nezavisnim i sa zavisnim promenljivama u statističkim analizama čiji se efekti kontrolišu prilikom analize veze između zavisne i nezavisne promenljive) za imputaciju jedne promenljive y. Ovo se najjednostavnije postiže pomoću metode „adjustment cells“ kojom se na osnovu x klasifikuju instance koje daju i koje ne daju odgovore u imputacione klase tj. ćelije za prilagođavanje (adjustment cells).
Pre kreiranja ćelija, svi neprekidni kovarijanti se kategorišu. Imputacija se zatim sprovodi nasumično za svakog „non-respondent“-a u svakoj ćeliji. Unakrsna klasifikacija prema broju kovarijanata može dovesti do kreiranja previše imputacionih klasa, što se može rešiti odbacivanjem promenljivih ili smanjivanjem kategorija realnih promenljivih dok se odgovarajući donor ne nađe. Još jedan potencijalni problem je retka posednutost imputacione klase koja može dovesti do preteranog korišćenja jednog donatora. Ovo se može rešiti uvođenjem ograničenja d za broj puta koliko se može koristiti jedan donator za imputiranje u primaoca. Optimalan izbor za d zavisi od veličine uzorka i od razmene između

dobitka u preciznosti zbog ograničavanja i povećane pristrasnosti od smanjenog kvaliteta uparivanja donatora i primaoca.
Dva ključna svojstva varijable koja se koristi za kreiranje imputacionih klasa su:

1. Da li je u vezi sa nedostajućom promenljivom y;
2. Da li je u vezi sa binarnom promenljivom koja ukazuje na prisustvo y

		Association with outcome	
		Low	**High**
Association with non-response	Low	Bias: -	Bias: -
		Var: -	Var: ↓
	High	Bias: -	Bias: ↓
		Var: ↑	Var: ↓

Slika 2. Uticaj jačine povezanosti između promenljivih imputacione klase i non-response i ishoda non-response na pristrasnost i varijansu srednje vrednosti

4.1.2 Definisanje metrike za uparivanje donatora i primaoca:

Neka je $x_i = [x_{i1}, x_{i2}, \dots, x_{iq}]$ vektor vrednosti q kovarijanata subjekta i koji se koriste za kreiranje imputacionih klasa i neka je $C(x_i)$ ćelija kojoj subjekat pripada i koja je rezultat unakrsne klasifikacije. Uparivanje primaoca i sa donatorima j u istoj ćeliji je isto kao i uparivanje na osnovu metrike:

$$d(i,j) = \begin{cases} 0, & j \in C(x_i) \\ 1, & j \notin C(x_i) \end{cases}$$

Druge mere bliskosti potencijalnih donatora sa primaocem se mogu definisati, tako da nema potrebe za kategorizacijom neprekidnih promenljivih:

1) Maximum deviation:
$$d(i,j) = \max |x_{ik} - x_{jk}|$$
gde je x_k skalirano na adekvatan način tako da razlika bude uporediva
2) Mahalanobisova distanca:
$$d(i,j) = (x_i - x_j)^T \widehat{Var}(x_i)^{-1}(x_i - x_j)$$
gde je $\widehat{Var}(x_i)$ procena matrice kovarijanse od x_i
3) Predictive mean:

$$d(i,j) = \left(\hat{y}(x_i) - \hat{y}(x_j)\right)^2$$

gde je $\hat{y}(x_i) = x_i{}^T\hat{\beta}$ predviđena vrednost za y za neispitanika i koja je rezultat regresije od y nad x koristeći samo podatke ispitanika.

Uključivanje nominalnih promenljivih pomoću ovih metrika nije jednostavno za sve mere rastojanja. Najjednostavnije je sa predictive mean metrikom, koja zahteva samo konverziju u skup promenljivih za uključivanje u regresioni model. Ukoliko su sve varijable za kreiranje imputacione klase kategoričke i sve interakcije između njih uključene u regresioni model, predictive mean uparivanje se svodi na adjustment cells metod. Za druge metode su potrebni kompleksniji pristupi i drugačije mere rastojanja za nominalne promenljive. Korišćenjem generalizovanih linearnih modela poput logističke regresije za modelovanje, predictive mean omogućuje se da se ova metrika koristi i za diskretne i za kontinualne ishode.

Nakon što se metrika odabere, postoji nekoliko načina da se definiše skup donatora za svakog primaoca. Jedan način je definisati skup donatora za neispitanika j kao skup ispitanika i sa $d(i,j) < \delta$, za prethodno određenu maksimalnu distancu δ. Donator se potom bira nasumičnim izvlačenjem ispitanika iz skupa donatora. Alternativno, ukoliko se bira ispitanik koji je najbliži neispitaniku, metod se naziva deterministički ili nearest-neighbour hot deck.

5. Metodi imputacije maksimalne verovatnoće

Prethodno opisane metode ne razmatraju mehanizme nedostajanja zbog čega se savetuje protiv njihovog korišćenja u većini situacija. U idealnom i retkom slučaju kada su parametri distribucije θ poznati, uzorak iz takve raspodele uslovljen vrednostima drugih atributa ili ne u zavisnosti od toga da li važe MAR, MCAR ili NMAR bi bila pogodna imputirana vrednost za onu koja nedostaje. Problem je što su ti parametri retko poznati i vrlo ih je teško proceniti. U jednostavnom slučaju, kao što je bacanje novčića, $P(heads) = \theta$ i $P(heads) = 1 - \theta$. Vrednost θ može varirati u zavisnosti od toga da li je novčić nekako namešten ili ne, pa je zbog toga nepoznata. Onda je jedino rešenje bacati novčić n puta, dobijajući time h puta glavu i $n - h$ puta pismo. Procena za θ bi onda bila $\hat{\theta} = h/n$. U opštem slučaju, verovatnoća od θ dobija se pomoću binomne distribucije $P(\theta) = \binom{h}{n}\theta^n(1-\theta)^{n-h}$. Može se dokazati da je $\hat{\theta}$ procena maksimalne verovatnoće (tačka u parametarskom prostoru koja maksimizira funkciju verovatnoće) za θ. Problem maksimiziranja funkcije verovatnoće se lako može rešiti i za jednodimenzionalnu Gausovu raspodelu nalaženjem μ i σ parametara. Međutim, u realnim situacijama nije tako lako rešiti ovaj problem. Raspodela se može ponašati nepredvidivo ili imati previše parametara, što bi postupak maksimiziranja funkcije verovatnoće učinilo računski previše kompleksnim. Jedan način da se reši ovaj problem je

upotrebom skrivenih promenljivih kako bi se pojednostavnila funkcija verovatnoće i time se obračunale NV. Posmatrane promenljive se mogu odrediti na osnovu podataka, dok skrivene promenljive utiču na podatke ali nisu izmerene. Upotrebom skrivenih promenljivih se ne može skroz rešiti problem, tako da je potrebno uvesti dodatne korake. Jedan od najčešćih je iterativni pristup kojim se dobijaju procene nekih parametara, koristi regresija za imptuaciju vrednosti i ceo postupak ponavlja. Međutim, imputirane vrednosti zavise od procenjenih parametara θ pa ne pružaju neke nove informacije.za proces i mogu se ignorisati. Postoji nekoliko tehnika za dobijanje procena maksimalne verovatnoće.

5.1 Expectation-Maximization (EM)

Expectation-Maximization algoritam je pristup za procenu parametara raspodele verovatnoće nekompletnih podataka. Ta verovatnoća je funkcija nekih parametara. Procena maksimalne verovatnoće je metod koji određuje vrednosti parametara modela tako da oni maksimalno povećaju verovatnoću da je proces opisan modelom proizveo podatke koji su zaista i posmatrani. Termin *nekompletni podaci* implicira postojanje dva prostora uzoraka: X i Y, kao i više-prema-jedan mapiranje iz X ka Y. Posmatrani podaci y su realizacija iz Y, dok podaci x iz X nisu posmatrani direktno, već preko mapiranja $x \rightarrow y(x)$. Za x znamo jedino da leži u $X(y)$ koji predstavlja podskup od X koji je određen izrazom $y = y(x)$, gde su y posmatrani podaci. x referiramo kao „kompletni podaci“. Odnos između specifikacije kompletnih podataka $f(\ldots|\ldots)$ i nekompletnih podataka $g(\ldots|\ldots)$ (gde su g i f funkcije gustine verovatnoće) je:

$$g(y|\theta) = \int_{X(y)} f(x|\theta)dx$$

EM algoritam treba da pronađe vrednost θ koja maksimizuje $g(y|\theta)$ kada su dati posmatrani podaci y.

Algoritam se odvija u dva koraka koji se ponavljaju do konvergencije. Zavisne slučajne promenljive su modelovane preko posmatrane promenljive a i skrivene promenljive b.

1) E korak:

Računanje očekivane vrednosti za $logP_\theta(y,x)$ ukoliko su dati posmatrani podaci y i trenutni parametri distribucije θ. E korak je defnisan na sledeći način:

$$Q(\theta,\theta') = \sum_x P_{\theta'}(b|a)logP_\theta(b,a)$$

, gde su θ' novi parametri distribucije.

2) M korak**:**

Razmatraćemo uslovnu verovatnoću u kojoj su A i B slučajne promenljive koje izvlačimo iz raspodela $P(a)$ i $P(b)$, respektivno. Ona se može odrediti preko sledećeg izraza:

$$P(b|a) = \frac{P(b,a)}{P(a)}$$

, a potom i preko izraza:

$$E[b] = \sum_b P(b)b$$

Za funkciju koja zavisi od B, $h(B)$, očekivanje se računa trivijalno:

$$E[h(B)] = \sum_b P(b)h(b).$$

Za A $(A = a)$, očekivanje je:

$$E[h(B)|a] = \sum_b P(b|a)h(b).$$

Želimo da nađemo θ koje maksimizira logaritamsku verovatnoću posmatranih a i neposmatranih b promenljivih. Uslovno očekivanje od $logP_\theta(b,a)$ za dato a i θ' je:

$$E[logP(b,a|\theta)|a,\theta'] = \sum_x P(b|a,\theta')logP(b,a|\theta)$$
$$= \sum_x P_{\theta'}(b|a)logP_\theta(b,a)$$

, i treba naći nove parametre distribucije θ koji maksimiziraju gornji izraz.
Ukoliko je $\sum_b P_{\theta'}(b|a)logP_\theta(b,a) > \sum_b P_{\theta'}(b|a)logP_{\theta'}(b,a)$, onda je $P_\theta(a) > P_{\theta'}(a)$, onda se očekivanje logaritma verovatnoće povećava, što samim tim znači da se modelovanje posmatranih varijabli a poboljšava.
U realnim problemima radi se sa nizovima atributa $x_1, \dots, x_n$. Ukoliko pretpostavimo nezavisne i identično raspodeljene slučajne varijable, očekivanje se može izračunati sumiranjem svih tačaka:

$$Q(\theta,\theta') = \sum_{i=1}^{n}\sum_x P_{\theta'}(b|x_i)logP_\theta(b,x_i)$$

EM algoritam za imptuaciju:
Da bi se EM algoritam koristio za imputaciju, treba pretpostaviti da podaci pripadaju nekoj raspodeli verovatnoće, što je i glavna mana ovog pristupa. EM najbolje radi sa raspodelama koje je lako maksimizirati, kao što su mešoviti Gausovi modeli (*Gaussian mixture models*). Kada EM radi sa multivarijatnom Gausovom raspodelom, korišćenje multivariajtnih Gausovih podataka se može parametrizovati preko srednje vrednosti i matrice kovarijanse. U svakoj iteraciji EM algoritma za imputaciju, procene za srednje vrednosti μ i kovarijanse Σ su predstavljene preko matrice i korigovane u tri faze, do konvergencije. Ovi parametri se koriste za primenu regresije nad NV korišćenjem celih podataka.

1) <u>Faza 1 (E-korak)</u>:
   Računaju se srednje vrednosti za svaki atribut posmatranih podataka (instance sa NV se ne razmatraju prilikom računanja srednjih vrednosti). Nakon toga, računa se matrica kovarijanse, takođe samo na osnovu posmatranih podataka. Procedure

maksimalne verovatnoće se koriste za procenu jednačina regresije kojima se modeluju veze atributa sa svim ostalim atributima.

2) Faza 2 (E-korak):
   NV se imputiraju svojim vrednostima uslovnog očekivanja na osnovu raspoloživih celih i na osnovu procenjenih koeficijenata regresije.

$$x_{mis} = \mu_{mis} + (x_{obs} - \mu_{obs})B + e$$

   , gde je rezidual $e \in \mathbb{R}^{1 \times n_{mis}}$ nasumični vektor sa srednjom vrednošću 0 i nepoznatom matricom kovarijanse, a B su procenjeni koeficijenti regresije. Instanca x od n atributa je podeljena na deo sa posmatranim vrednostima x_{obs} i na deo sa NV x_{obs} (matrica srednje vrednosti i kovarijanse je takođe podeljena na taj način).

3) Faza 3 (M-korak):
   Ponovna procena srednje vrednosti i matrice kovarijanse. Za srednju vrednost se uzima srednja vrednost uzorka kompletiranog skupa podataka, a kovarijansa je uzorak matrice kovarijanse i matrica kovarijanse imputacionih grešaka:

$$\hat{B} = \hat{\Sigma}_{obs,obs}^{-1}\hat{\Sigma}_{obs,mis}, \text{ i}$$
$$\hat{C} = \hat{\Sigma}_{mis,mis} - \hat{\Sigma}_{mis,obs}\hat{\Sigma}_{obs,obs}^{-1}\hat{\Sigma}_{obs,mis}$$

   Nakon što se ažuriraju B i C, srednja vrednost i matrica kovarijanse se ažuriraju na sledeći način:

$$\hat{\mu}^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$\hat{\Sigma}^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{S}_i^{(t)} - (\hat{\mu}^{(t+1)})\hat{\mu}^{(t+1)}\right]$$

   , gde za svaku instancu $x = X_i$, uslovno očekivanje $\hat{S}_i^{(t)}$ unakrsnih proizvoda se sastoji od tri dela. Dva dela koja uključuju dostupne vrednosti u instanci:

$$E\left(x_{obs}^T x_{obs} | x_{obs}; \hat{\mu}^{(t)}, \hat{\Sigma}^{(t)}\right) = x_{obs}^T x_{obs}$$

   i

$$E\left(x_{mis}^T x_{mis} | x_{obs}; \hat{\mu}^{(t)}, \hat{\Sigma}^{(t)}\right) = x_{mis}^T x_{mis} + \hat{C}$$

   su suma unakrsnog proizvoda imputiranih vrednosti i rezidualne matrice kovarijanse $\hat{C} = Cov(x_{miss}, x_{mis} | x_{obs}; \hat{\mu}^{(t)}, \hat{\Sigma}^{(t)})$ (uslovne matrice kovarijanse imputacione greške)

Prva procena srednjih vrednosti i matrice kovarijanse mora da se izvrši nad kompletno posmatranim skupom podataka. Jedno rešenje je popuniti skup podataka inicijalnim procenama srednjih vrednosti i matrice kovarijanse i proces se završava kada se procene srednjih vrednosti i matrica kovarijanse više ne menjaju (iznad nekog praga). EM imputacija

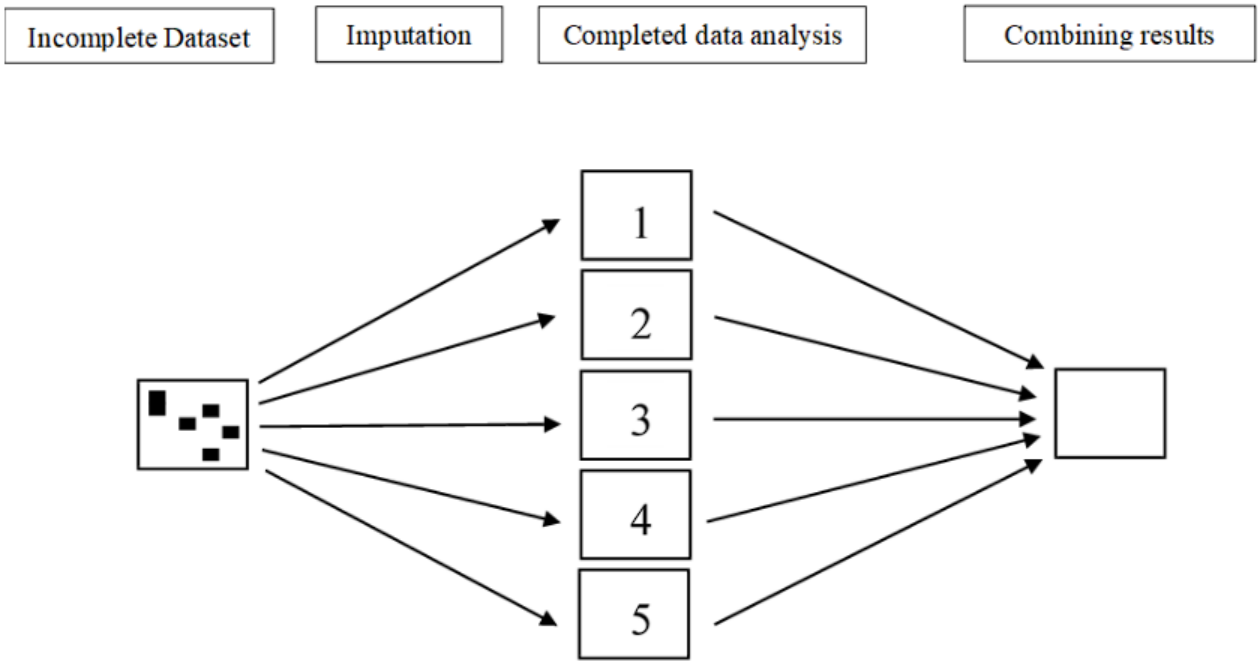
je najpogodnija za numeričke skupove podataka. EM algoritam je validan i dan danas, ali je uglavnom deo nekog sistema u kome promaže pri evoluciji neke distribucije kao što je GTM neuronska mreža.

5.2 Višestruka imputacija (Multiple Imptuation - MI)

Jedan problem kod metoda maksimalne verovatnoće je taj što se zanemaruju greške koje mogu nastati tokom procesa estimacije. MI je dizajniran kako bi smanjio te greške o trošku računske kompleksnosti.

MI je Monte Karlo pristup u kome se generiše više vrednosti za imputiranje na osnovu posmatranih podataka. Za razliku od EM, MI izvršava nekoliko imputacija koje kao rezultat daju nekoliko kompletnih skupova podataka. Imputacije se vrše pomoću metoda Monte Karlo Markovljevih lanaca, budući da se nekoliko vrednosti za imputaciju dobija uz pomoć ubacivanja slučajne komponente u proces (obično iz standardne normalne raspodele). MI takođe pretpostavlja da su procene parametara zapravo procene uzoraka. Zbog toga se parametri ne procenjuju direktno na osnovu dostupnog skupa podataka, već se tokom procesa izvlače iz njihovih posteriornih Bjesovih raspodela na osnovu podataka koji su trenutno dostupni. Ove pretpostavke ukazuju da se MI može primeniti samo ako važe MAR i MCAR mehanizmi nedostajanja. Korišćenje više mogućih potencijalno verodostojnih vrednosti za imputaciju obezbeđuje kvantifikaciju neizvesnosti u proceni NV. Zbog toga je ovaj algoritam, iako računski zahtevniji, precizniji od EM algoritma i od algoritama za imputaciju jedne vrednosti generalno.

MI poseduje Bajesovu prirodu koja forsira korisnika da specificira prvobitnu raspodelu za parametre θ modela iz kog se očekivana vrednost e izvlači. U praksi, rezultati zavise više od izbora raspodele podataka nego od raspodele θ. Kod EM se kreira po jedna vrednost za imputaciju za svaku NV (pa se zato izvlači samo jedno e). Za razliku od EM, MI će kreirati nekoliko verzija skupa podataka, kod kojih su posmatrani podaci isti a imputirane vrednosti za nedostajuće su različiti. Ovaj proces je poznat kao **augmentacija podataka.**



Slika 5. Grafička reprezentacija MI procedure

1. Skup podataka sa NV (tj. nekompletni skup podataka) se duplira nekoliko puta. Dovoljno je napraviti 3 do 5 kopija (tj. izvršiti 3 do 5 imputacija za svaku NV). Efikasnost konačne estimacije koja je izvršena na osnovu m imputacija je približno $(1+\frac{\gamma}{m})^{-1}$, gde je γ deo skupa podataka koji je nedostajući;
2. NV se imputiraju vrednostima iz svake kopije skupa podataka. U svaku kopiju se imputiraju drugačije vrednosti;
3. Imputirani skupovi podataka se analiziraju uz pomoć statističkih testova i rezultati se sumarizuju.

Za početak je potrebno zadati početne procene vrednosti za srednju vrednost i matrice kovarijanse. One se obično pribavljaju iz EM algoritma nakon što se vrednosti stabilizuju pri kraju izvršenja. Nakon toga, proces augmentacije podataka počinje naizmeničnim popunjavanjem NV nakon čega se pribavljaju zaključci o nepoznatim parametrima na stohastički način. Vrednosti za imputaciju se kreiraju na osnovu dostupnih vrednosti parametara NV a potom se izvlače nove vrednosti parametara na osnovu Bajesovske posteriorne raspodele korišćenjem i posmatranih i nedostajućih podataka. Nadovezivanje ovog procesa simuliranja NV i parametara je šta kreira Markovljev lanac koji će u nekom trenutku konvergirati. Raspodela parametara θ će se stabilizovati ka posteriornoj raspodeli prosečnoj za sve NV, a raspodela NV će se stabilizovati ka prediktivnoj raspodeli: odgovarajućoj distribuciji koja je potrena za izvlaćenje vrednosti za NV.
Teško je proceniti da li je proces augmentacije podataka konvergirao, pošto nasumične vrednosti parametara se neprestano menjaju tokom procesa. Predložena je interpretacija konvergencije u smislu nedostatka serijske zavisnosti: proces je konvergirao za k ciklusa

ukoliko je vrednost bilo kog parametra u iteraciji $t \in 1, 2, \ldots$ statistički nezavisan od svoje vrednosti u iteraciji $t + k$. Vrednost k indikuje kada prestati sa formiranjem Markovljevih lanaca. Tipičan proces uključuje da se za svaku imputaciju od 1 do m izvrši augmentacija podataka tokom k ciklusa. Nakon kreacije m skupova podataka, oni mogu biti analizirani bilo kojom standardnom metodom (npr. logistička regresija ili linearna regresija) koja se primenjuje na svaki od tih skupova podataka kako bi se dobila varijabilnost m rezultata kojom se opisuje neizvesnost NV.

Rubinovo pravilo za dobijanje skupa procenjenih koeficijenata i standardnih greški je sledeći: Neka $\hat{R}$ označava procenu od interesa i U je procenjena varijansa (R je ili procenjeni koeficijent regresije ili kernel parameter SVM-a, zavisno od modela koji se koristi). Nakon što se dobave sve imputacije, dobijaju se procene $\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m$ i njihove respektivne varijanse $U_1, U_2, \ldots, U_m$. Krajnja procena (MI procena) je izražena preko:

$$\bar{R} = \frac{1}{m}\sum_{i=1}^{m} \hat{R}_i.$$

Varijansa procene ima dve komponente: varijabilnost unutar svakog skupa podataka i između svih skupova podataka:

$$\bar{U} = \frac{1}{m}\sum_{i=1}^{m} U_i,$$

$$B = \frac{1}{m-1}\sum_{i=1}^{m} (\hat{R}_i - \bar{R})^2.$$

Ukupna varijansa T je korigovana suma ove dve komponente sa faktorom koji uračunava grešku simulacije u $\hat{R}$:

$$T = \hat{U} + (1 + \frac{1}{m})B$$

Za intervale pouzdanosti, Rubin daje aproksmaciju:

$$\bar{R} \pm t_v\sqrt{T}$$

, gde je stepen slobode v procenjen kao:

$$v = (m-1)\left\{1 + \frac{W}{\left(1 + \frac{1}{m}\right)B}\right\}^2,$$

a t_v odgovarajući fraktil centralne t raspodele.

5.3 Bajesova analiza principijalnih komponenti (BPCA)

Neka je dat skup podataka N posmatranih D-dimenzionalnih vektora y. Konvencionalna PCA uključuje računanje matrice kovarijanse skupa podataka, nalazi sopstvene vektore i sopstvene vrednosti matrice kovarijanse i zadržava K vektora koji odgovaraju K najvećim sopstvenim vrednostima i oni predstavljaju principijalne kompontente preko kojih se može predstaviti skup podataka. Bitno ograničenje konvencionalne PCA je to što ne definiše

raspodelu verovatnoće jer onda korišćenje određenih statističkih tehnika zaključivanja nije moguće. Međutim, PCA se može reformulisati kao rešenje maksimalne verovatnoće specifičnog modela latentnih promenljivih. Neka je data latentna promenljiva x dimenzije q, čije je raspodela $p(x) = N_k(0, I_K)$. Onda je posmatrana promenljiva y definisana kao:

$$y = Wx + \mu + \epsilon$$

, gde su W matrica čije su kolone principijalne komponente, μ je D-dimenzionalni vektor srednjih vrednosti, a ϵ je vektor distribuiran Gausovom raspodelom kovarijanse $\sigma^2 I_D$. Funkcija gustine verovatnoće posmatrane promenljive y je onda:

$$p(t|x) = N(Wx + \mu, \sigma^2 I_D).$$

Marginalna raspodela posmatrane promenljive je data konvoluciom dve multivarijatne Gausove raspodele, što daje ponovo višestruku Gausovu raspodelu:

$$p(y) = \int p(y|x)\, p(x) dx = N(\mu, C)$$

, gde je C matrica kovarijanse čija je vrednost $WW^T + \sigma^2 I_D$. Model predstavlja multivarijatnu Gausovu raspodelu vođenu parametrima μ, W i σ^2. Logaritamska verovatnoća parametara nad posmatranim skupom podataka je:

$$L(W, \mu, \sigma^2) = -\frac{N}{2}\{d\ ln(2\pi) + \ln|C| + Tr[C^{-1}S]\}$$

, gde je S početna matrica kovarijanse uzorka skupa podataka. Rešenje maksimalne verovatnoće po μ je $\boldsymbol{\mu_{ML}} = \overline{\boldsymbol{y}}$. Stacionarne tačke logaritamske verovatnoće u odnosu na W zadovoljavaju $\boldsymbol{W_{ML}} = \boldsymbol{U_K}(\boldsymbol{\Lambda_K} - \boldsymbol{\sigma^2 I_K})^{\mathbf{1/2}}$, gde su U_K i Λ_K sopstveni vektori i njima odgovarajuće sopstvene vrednosti kojih ima K i koje su najveće među sopstvenim vrednostima matrice S. Maksimalna verovatnoća je postignuta kada je K najvećih sopstvenih vrednosti izabrano. Onda je $\boldsymbol{\sigma^2} = \frac{\mathbf{1}}{\boldsymbol{D-K}} \sum_{\boldsymbol{i=k+1}}^{\boldsymbol{D}} \boldsymbol{\lambda_i}$ što predstavlja prosečnu varijansu izgubljenu po odbačenoj dimenziji (konvencionalni PCA: $\sigma^2 \to 0$). Probabilistički PCA je uspešno primenjen na probleme u kompresiji podataka, proceni gustine i vizuelizaciji podataka. Međutim, kao i kod konvencionalnog PCA, model ne pruža mehanizam za određivanje vrednosti za K.

Metod estimacije NV baziran na BPCA se sastoji od 3 elementarna procesa:

1) PC regresija;
2) Bajesova procena;
3) Repetativni algoritam po uzoru na EM

1. PC regresija:

Tokom PC regresije, y_{miss} deo vektora y se procenjuje na osnovu posmatranog dela y_{obs} koristeći rezultate PCA. Neka su w_{obs}^l i w_{miss}^l delovi principijalne ose w_l koji odgovaraju posmatranim i nedostajućim delovima, respektivno, u y. Slično, neka je $W = (W_{obs}, W_{miss})$ gde W_{obs} ili W_{miss} označava matricu čiji su kolona-vektori $w_{obs}^1, \dots, w_{obs}^K$ ili $w_{miss}^1, \dots, w_{miss}^K$, respektivno ($K$ je broj principijalnih osa). Latentne promenljiva $x = (x_1, \dots, x_k)$ za vektor y se dobija minimizacijom rezidualne greške

$err = \|y_{obs} - W_{obs}x\|^2$. Ovo je dobro poznat problem regresije, i rešenje najmanjih kvadrata daje:

$$x = (W_{obs}^T W_{obs})^{-1} W_{obs}^T y_{obs}.$$

Korišćenjem x, nedostajući deo se procenjuje na sledeći način:

$$y_{miss} = W_{miss}x.$$

U PC regresiji, W treba da je poznato od ranije.

2. **Bajesova procena:**
   Bajesova procena pribavlja posteriornu raspodelu za θ i X, prema Bajesovoj teoremi:

$$p(\theta, X|Y) \propto p(Y, X|\theta)p(\theta)$$

   , gde je $p(\theta)$ priorna raspodela, koja označava prvobitnu preferencu za parametar θ. Priorna raspodela je deo modela koji mora biti definisan pre procene. Pretpostavljene su konjugovane priorne raspodele za μ i τ (sklarani inverz varijanse od greške ϵ) i hijerarhijska priorna raspodela za W kojom upravlja hiperparametar $\alpha \in \mathbb{R}^K$. Kada je Euklidska norma principijalne ose, $\|w_j\|$ mala u odnosu na varijansu šuma $1/\tau$, hiperparametar α postaje veliki i principijalna osa w_j se smanjuje skoro do 0. Na taj način, redundantne principijalne ose se automatski suzbijaju.
3. **Repetativni algoritam po uzoru na EM:**
   Koristi se Varijacioni Bajesov algoritam (VB) da bi se izvršila Bajesova procena i za parametre modela θ i za nedostajuće vrednosti, Y_{miss}. Iako VB agloritam liči na EM algoritam koji vraća procene maksimalne verovatnoće za θ i Y_{miss}, on zapravo pribavlja posteriorne raspodele za θ i Y_{miss} na repetativan način. VB algoritam je implementiran na sledeći način:
   a. Posteriorna raspodela NV, $q(Y_{miss})$ se inicijalizuje imputacijom svih NV prosečnim vrednostima širom instance;
   b. Posteriorna raspodela parametra θ, $q(\theta)$ se procenjuje uz pomoć posmatranih podataka Y_{obs} i trenutne posteriorne raspodele NV, $q(Y_{miss})$;
   c. Posteriorna raspodela NV se procenjuje uz pomoć trenutne $q(\theta)$;
   d. Hiperparametar α se ažurira uz pomoć trenutnih $q(\theta)$ i $q(Y_{miss})$
   e. Koraci b i d se ponavljaju do konvergencije.

   Za VB algoritam je dokazano da uvek konvergira ka lokalnom optimumu, i to uvek ka jednom rešenjul, dok globalni optimum nije zagarantovan. Potom se NV u matričnom izrazu imputiraju na sledeći način:

$$\hat{Y}_{miss} = \int y_{miss} q(Y_{miss}) dY_{miss}.$$

6. Metodi imputacije bazirani na mašinskom učenju

Imputacione metode iz sekcije 5 oslanjaju se na primene statistike i modeluju veze među vrednostima traženjem skrivenih raspodela verovatnoće. U oblasti veštačke inteligencije, modelovanje nepoznatih veza između atributa i zaključaka izvedenih iz implicitnih

informacija sadržanih u uzorkovanom skupu podataka se obavlja uz pomoć metoda mašinskog učenja. Primećeno je da se mnogi procesi kojima se vrši predviđanje kontinualne ili nominalne vrednosti na osnovu prethodnog procesa učenja u regresiji ili klasifikaciji mogu iskoristiti za predviđanje NV.

6.1 Imputacija uz pomoć K-najbližih suseda (KNNI)

KNNI algoritam računa K-najbližih suseda svaki put kada naiđe na NV u trenutnoj instanci i vrednost iz njih se imputira. Za nominalne vrednosti se uzima najčešća vrednost među susedima, a za neprekidne se uzima prosećna vrednost suseda. Za nalaženje suseda je neophodna neka mera bliskosti odnosno udaljenosti među instancama. Najčešće korišćena mera je Euklidska udaljenost.

Kako bi se procenila NV y_{ih} u i-tom vektoru y_i uz pomoć KNNI, najpre se bira K primera čije su vrednosti atributa slične kao kod vektora y_i. Potom se NV procenjuje kao aritmetička sredia vrednosti odgovarajućeg atributa iz svakog od K vektora. NV y_{ih} se računa na sledeći način:

$$y_{\hat{i}h} = \frac{\sum_{j \in I_{Kih}} y_{jh}}{|I_{Kih}|}$$

, gde je I_{Kih} skup indeksa K-najbližih suseda za i-ti primer. Ako y_{jh} nedostaje, j-ti atribut se izbacuje iz I_{Kih} (ovaj korak često zahteva primenu neke heuristike, pri čemu je ona koja je korišćena u ovom odeljku ujedno i najčešće korišćena). K vrednost je potreno odrediti empirijski.

6.2 Imputacija uz pomoć težinskih K-najbližih suseda (WKNNI)

WKNNI bira instance sa vrednostima sličnim instanci koja sadrži NV na osnovu neke mere bliskosti, pri čemu uzima u obzir činjenicu da se udaljenosti instance od njenih suseda razlikuju. Mera bliskosti između dva primera y_i i y_j je definisana Euklidskom udaljenošću koja se računa između odgovarajućih posmatranih parova atributa. Definisana mera je:

$$\frac{1}{s_i} = \sum_{h_i \in O_i \cap O_j} (y_{ih} - y_{jh})^2$$

, gde je O_i ={$h|h$-ta komponenta od y_i posmatrana}. NV y_{ih} se računa kao aritmetička sredina vrednosti pomnožene odgovarajućom merom bliskosti:

$$y_{\hat{i}h} = \frac{\sum_{j \in I_{Kih}} s_i(y_j) y_{jh}}{\sum_{j \in I_{Kih}} s_i(y_j)}$$

, gde je I_{Kih} skup indeksa K-najbližih suseda za i-ti primer. Ako y_{jh} nedostaje, j-ti atribut se izbacuje iz I_{Kih} (ovaj korak često zahteva primenu neke heuristike, pri čemu je ona koja je korišćena u ovom odeljku ujedno i najčešće korišćena). K vrednost je potreno odrediti empirijski.

6.3 Imputacija K-means klasterovanjem (KMI)

U K-means klasterovanju, udaljenost objekata unutar klastera meri se sumiranjem udaljenosti objekata od centroida klastera kome su oni dodeljeni. Centroid klastera predstavlja srednju vrednost objekata tog klastera. Za skup objekata, cilj klasterovanja je podeliti ga na grupe na osnovu sličnosti objekata i da se minimizuje udaljenost objekata unutar klastera. Nakon što se klasteri formiraju, za svaki nekompletan objekt se popunjavaju NV na osnovu informacija o klasteru kome taj objekat pripada. Objekti koji pripdaju nekom klasteru su jedni drugima ujedno i najbliži susedi, što omogućuje ovom algoritmu da primeni algoritam najbližih suseda za imputaciju NV na sličan način kao i KNNI.

Neka je dat skup od N objekata $X = x_1, x_2, \dots, x_N$, gde svaki objekat ima S atributa. Sa x_{ij} označavamo vrednost atributa j u objektu x_i. Objekat x_i je kompletan objekat ako je $\{x_{ij} \neq \phi | \forall j, 1 \leq j \leq S\}$, a nekompletan ako je $\{x_{ij} = \phi | \exists j, 1 \leq j \leq S\}$ (kažemo da objekat x_i ima NV kod atributa j). Za bilo koji nekompletan objekat x_i , $R = \{j | x_{ij} \neq \phi, 1 \leq j \leq S\}$ označava skup atributa čije su vrednosti dostupne i oni se nazivaju referentni atributi. Cilj je odrediti nereferentne atribute za nekompletne objekte. Najpre se skup X deli na K klastera. Neka je $V = v_1, v_2, \dots, v_k$ skup od K klastera, gde v_k predstavlja centroid klastera k (v_k je takođe vektor u S-dimenzionalnom prostoru). $d(v_k, x_i)$ predstavlja udaljenost između centroida v_k i objekta x_i.

KMI se deli na tri procesa:

1) Nasumično se bira K kompletnih objekata koji predstavljaju inicijalne cetnroide;
2) Iterativno se modifikuju particije kako bi se redukovala suma udaljenosti svakog objekta od centroida klastera kome je objekat dodeljen. Proces se završava nakon što je suma udaljenosti manja od unapred zadatog praga tolerancije ε ili nakon što su centroidi prestali da se menjaju u poslednjoj iteraciji. U ovom procesu se vrši minimizacija funkcije:

$$J = \sum_{k=1}^{K} \sum_{x_i \in F_k} d(v_k, x_i)$$

   , gde je $d(v_k, x_i)$ mera udaljenosti između centroida v_k klastera F_k i objekta x_i koji pripada tom klasteru.
3) Objekti koji pripadaju istom klasteru su najbliži susedi jedni drugom i primenjuje se algoritam najbližih suseda za imputaciju nereferentnih atributa.

6.4 Imputacija Fuzzy K-means klasterovanjem (KMI)

U realnim situacijama, granice među klasterima mogu da se preklapaju, zbog čega je nejasno da li neki objekat pripada u celosti nekom klasteru. Da bi takve situacije mogle da se razmatraju, uveden je pojam funkcije članstva.

U fuzzy klasterovanju, svaki objekat poseduje funkciju članstva $\mu_k: X \rightarrow [0,1]$ koja opisuje stepen pripadnosti određenom klasteru. Što je veća vrednost funkcije, to objekat više

pripada klasteru prema kome se pripadnost računa. Funkcija koju treba minimizovati postaje:

$$J = \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_k^q(x_i) d(v_k, x_i)$$

, gde je $\mu_k(x_i)$ funkcija članstva, a $q > 1$ je konstanta poznata kao indeks nejasnoće koja kontroliše količinu nejasnoće tj. fuzziness-a. Budući da minimizacija ove funkcije može dovesti do trivijalnih rešenja, uvedena su dva ograničenja tokom procesa minimizacije:

$$\sum_{i=1}^{N} \mu_k(x_i) > 0, \qquad \forall k \in \{1,2, \dots, K\}$$
$$\sum_{i=1}^{N} \mu_k(x_i) = 1, \qquad \forall i \in \{1,2, \dots, N\}.$$

Prvo ograničenje garantuje da nema praznih klastera, dok drugo nameće uslov da je suma funkcija članstva sa svakim klasterom jednaka jedinici. Diferencijacijom funkcije koju treba minimizovati dobija se:

$$\mu_k(x_i) = \frac{1}{\sum_{h=1}^{K} \left(\frac{d(v_k, x_i)}{d(v_h, x_i)} \right)^{2/(q-1)}}, \forall i \in \{1, \dots, N\}, k \in \{1, \dots, K\}$$

$$m_k = \frac{\sum_{i=1}^{N} \mu_k^q(x_i) \times x_i}{\sum_{i=1}^{N} \mu_k^q(x_i)}, k = 1,2 \dots, K$$

Izrazi dobijeni diferencijacijom funkcije J se koriste se u iterativnom procesu kojim se ažuriraju članstva i centroidi. Ažuriranje se nastavlja sve dok promene u vrednostima funkcija članstva ne postanu zanemarljive ili dok se ne dostigne određen broj iteracija.
Kao i kod KMI, algoritam se deli na tri procesa:

1) Proces inicijalizacije: bira se K ravnomerno raspoređenih centroida kako bi se izbegao problem zaglavljivanja u lokalnom minimumu.
2) Proces ažuriranja funkcija članstva i centroida dok promene ne postanu zanemarljive (može se odrediti pragom tolerancije ε kao i kod KMI) ili se ne dostigne određeni broj iteracija. U ovom procesu se objekat ne može dodeliti klaster kao što je to slučaj kod KMI zato što važi pretpostavka da svaki objekat pripada svakom klasteru sa različitim stepenom pripadnosti tj. funkcijom članstva.
3) Proces imputiranja nereferentnih atributa za svaki nekompletan objekat na osnovu informacija o funkcijama članstva i vrednostima centroida na sledeći način:

$$x_{i.j} = \sum_{k=1}^{K} \mu_k^q(x_i) \times v_{k,j}, za\ bilo\ koji\ nereferentni\ atribut\ j \notin R$$

Fuzzy klasterovanje se primenjuje kada klasteri nisu dovoljno dobro međusobno odvojeni (što je slučaj kod skupa podataka u kome su prisutni nedostajući podaci) i manje je podložan zaglavljivanju u lokalnom minimumu od klasičnog K-means klasterovanja.

6.5 Imputacija metodom potpornih vektora (SVMI)

Metode potpornih vektora (Support Vector Machines) su alati za nelinearnu regresiju i klasifikaciju. Oni nude karakteristike efikasnog treniranja koje poseduju parametarske tehnike, ali i mogućnost učenja nelinearnih zavisnosti kao i neparametarski metodi. Zbog ovih osobina, SVM spadaju u polu-parametarske metode.

Formalno, SVM za regresiju (SVR) modeluje uslovnu očekivanu vrednost imputacione šromenljive:

$$SVR: E(Y|X_1, X_2, \dots, X_n)$$

Za binarnu klasifikaciju ($Y \in \{+1, -1\}$), SVM daje najverovatniju od dve izlazne klase:

$$SVC: argmax(P(Y|X_1, X_2, \dots, X_n))$$

Može se iskoristiti SVR za predviđanje uslovnih atributa (tj. ulazi u SVR, $X_1, X_2, \dots, X_n$) čije vrednosti nedostaju. Najpre se biraju primeri čiji atributi nemaju NV. Potom atributi odluke (tj. izlazi iz SVR, Y) postaju uslovni atributi, a uslovni atributi čije vrednosti nedostaju postaju atributi odluke. Na taj način SVR može odrediti atribute koji sadrže NV.

6.6 Imputacija dekompozicijom singularnih vrednosti (SVDI)

SVDI je metod baziran na dekompoziciji singularnih vrednosti (SVD) koja predstavlja tehniku faktorizacije matrice koja vrši dekompoziciju matrice na 3 odvojene matrice. Izraz za SVD matrice A je:

$$A_{n\times m} = U_{n\times r}\Sigma_{r\times r}{V_{r\times m}}^T$$

, gde su U i V ortogonalne matrice, a Σ je dijagonalna matrica koja sadrži singularne vrednosti matrice A. Vrednost singularne vrednosti predstavlja količinu informacije ili varijabilnost koju sadrži njen odgovarajući singularni vektor. Ključna ideja kod SVDI je rekonstrukcija originalnog skupa podataka korišćenjem singularnih vektora čije su singularne vrednosti najveće.

Najpre je potrebno popuniti NV prosečnim vrednostima za atribute u kojima su NV, kako bi dekompozicija mogla da se izvrši. Nakon što se SVD izvrši, potrebno je postaviti singularne vrednosti čije su vrednosti manje od nekog zadatog praga na 0. Time se redukuje rang matrice i smanjuje dimenzionalnost podataka. Nakon toga se matrica rekonstruiše matričnim množenjem modifikovanih matrica i dobija matrica A' koja je manjeg ranga od originalne matrice A za broj singularnih vrednosti čije su vrednosti postavljene na 0. NV se zamenjuju odgovarajućim vrednostima iz matrice A'. Moguće je ponavljati navedene korake sve dok razlika između vrednosti dobijenih matrica ne postanu minimalne.

7. Praktični deo

Praktični deo sadrži dva dela:

1) Demonstracija Random Hot Deck metode imputacije nad skupom podataka koji sadrži kategoričke podatke;
2) Demonstracija metoda imputacije nad skupom podataka koji sadrži kontinualne podatke

Metode imputacije su demonstrirane na skupu podataka koji sadrži informacije o 1030 uzoraka betonskih kompozicija i njihovoj kompresivnoj snagi koju treba predvideti nekim od modela regresije. Budući da je teško naći skup podataka koji već nije prečišćen, NV su uvedene na sintetički način: od originalnog skupa podataka kreirana su tri, čije nedostajuće vrednosti prate MAR i MCAR mehanizme nedostajanja, sa procentom NV od 20%. Za svaki od ta tri slučaja, ivršena je istraživačka analiza podataka, provera raspodele, postojanja duplikata i outlier-a i provera korelacije među atributima. Utvrđeno je da nema duplikata, da feature-i većinom prate normalnu raspodelu ili raspodelu nalik njoj i da nema korelacije među feature-ima. Nakon početnog istraživanja i prečišćavanja, izvršena je demonstracija svake od opisanih metoda za imputaciju NV na sledeći način: nakon izvršene imputacije, podaci su bili predati modelima za regresiju čiji je zadatak bio predviđanje kompresivne snage uzoraka betona. Kod metoda imputacije koji su sadržali dodatne parametre (kao što je slučaj kod algoritama za klasterovanje npr.), vršena je dodatna analiza tih parametara u cilju maksimizacije performansi i rezultata odgovarajućih metoda imputacije. Primenjena je 10-Fold unakrsna validacija nad rezultatima svakog od regresionih modela i rezultati su pamćeni u cilju dalje analize. Medju modelima regresije, ubedljivo je najbolji bio Random Forest Regressor. Kod linearnih modela, najbolje su se pokazali Expectation-Maximization, KNN, WKNN i K-Means imputacija. Bilo je za očekivati da če se EM dobro pokazati budući da feature-i prate normalnu raspodelu ili raspodelu nalik njoj. Ubedljivo najgori metod je Zero imputation, kod svakog od linearnih i nelinearnih modela. Kod nelinearnih modela, najbolji je bio Weighted KNN imputation ($0.9155\ r2\ score$), dok su na drugom mestu bili Expectation-Maximization, KNN imputation, K-means imputation i SVDi (oko $0.912\ r2\ score$).
Kod MCAR slučaja, procenat NV je isti kao i kod MAR s tim što su se one nalazile u svakom feature-u sem u zavisnoj varijabli. Ovim pristupom je cilj bio proveriti dejstvo metoda imputacije kada svi feature-i sadrže NV. SVR imputacija nije radila u ovom slučaju pošto u u svakom koraku ovog algoritma je potrebno uzeti deo feature-a da bude ulaz u SVR, a pri tome nema incijalnog popunjavanja NV, što znači da je neophodno da postoji deo skupa podataka u kome su svi feature-i posmatrani, što se nije desilo u ovom slučaju. Kod linearnih metoda, najbolje se pokazala K-Means imputacija, dok je kod nelinearnih to bila SVD imputacija. Najgore su se pokazali imputacija nulama kod linearnih, i MI kod nelinearnih metoda.

8. Zaključak

U ovom radu, istražene su različite tehnike za rukovanje podacima koji nedostaju kao i mehanizmi nedostajanja podataka. Razumevanjem mehanizama i uzroka nedostajanja, moguće je primeniti odgovarajuće tehnike kojima se minimizuje pristrasnost i održava integritet analize. Takođe se može zaključiti da izbor metoda imputacije treba da zavisi od prirode i količine nedostajanja, kao i od pretpostavki koje imaju data-mining procesi koji se vrše nad podacima. Rad je pružio pregled teorijskih osnova i praktičnih razmatranja koji se primenjuju u radu sa nedostajućim podacima. Na osnovu njih, analitičari mogu da usvoje informisane strategije za ublaživanje uticaja nedostajanja vrednosti, jačanje validnosti njihovih analiza i napredovanje polja analize podataka na robustniji i pouzdaniji način.

9. Literatura:

[1] (Intelligent Systems Reference Library) Salvador García, Julián Luengo, Francisco Herrera - Data preprocessing in data mining-Springer (2015)

[2] Mechanisms of Missingness | How to Deal with Missing Data (osu.edu)

[3] A Review of Hot Deck Imputation for Survey Non-response - PMC (nih.gov)

[4] Maximum Likelihood from Incomplete Data via the EM Algorithm A. P. Dempster; N. M. Laird; D. B. Rubin Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39, No. 1. (1977), pp. 1-38.

[5] https://www.machinelearningplus.com/machine-learning/mice-imputation/

[6] Bayesian PCA Christopher M. Bishop Microsoft Research St. George House, 1 Guildhall Street Cambridge CB2 3NH, u.K. cmbishop@microsoft.com

[7] https://github.com/Duuuuuu/Probabilistic-and-Bayesian-PCA

[8] Fuzzy c-means clustering — skfuzzy v0.2 docs (pythonhosted.org)

[9] Imputing missing values with SVD - Python and R Tips (cmdlinetips.com)