LATVIJAS UNIVERSITĀTE
FIZIKAS, MATEMĀTIKAS UN OPTOMETRIJAS FAKULTĀTE
MATEMĀTIKAS NODAĻA

TRĪS VEIDU K-VIDĒJO KLASTERIZĀCIJAS ALGORITMU TEORĒTISKAIS PAMATOJUMS

BAKALAURA DARBS

Autors: **Ksenija Satinova**
Stud. apl. ks18008
Darba vadītājs: Olga Grigorenko

RĪGA 2022

Anotācija

Bakalaura darba mērķis ir izpētīt K-vidējo klasterizāciju metodes. Darbs ir balstīts uz trīs metodēm, tas ir: klasiskā K-vidējo klasterizācija, nestriktā C-vidējo klasterizācija, un C-vidējo klasterizācija nestriktas ekvivalences gadījumā, kas ir C-vidējo speciālgadījums. Apskatītājām metodēm tiek aprakstītas tas darbošanās algoritmi, kā arī tiek sniegts ieskats nestrikto attiecību teorijā. K-vidējo un C-vidējo algoritmi tiek programmēti *Python* valodā un tiek uztaisītas simulācijas algoritmu iterāciju skaita salīdzināšanai.

Atslēgvārdi: klasterizācija, K-vidējo klasterizācija, nestrikta klastrizācija, nestriktas ekvivalences.

Abstract

The purpose of this bachelor thesis is to investigate K-means clustering methods. The work is based on three methods: classical K-means clustering, fuzzy -means clustering, and C-means clustering under fuzzy equivalences, which is a special case of the fuzzy C-means clustering. The algorithms of the considered methods are described theoretically and the work provides also an introduction to the theory of fuzzy relations. The K-means and C-means algorithms are programmed in *Python* and simulations are made to compare the number of iterations made by the algorithms.

Keywords: clustering, K-means clustering, fuzzy clustering, fuzzy relations.

Saturs

Ievads

Klasterizācija – līdzīgu objektu apvienošana grupās – ir viens no fundamentālajiem uzdevumiem datu analīzes jomā. Jomu saraksts, kur tā tiek lietota, ir plašs: mārketings, medicīna, cilvēka orgānu formu, novietojumu un izmēru atpazīšana, attēlu segmentācija, prognozēšana, tekstu analīze, krāpšanas apkarošana, zemestrīces analīze un daudzi citi. Mūsdienās klasterizācija bieži vien ir pirmais solis datu analīzē.

Ja ir dota datu kopa $A = \{a_1, ..., a_m\}$, kas satur m objektus un katru objektu nosaka n pazīmes $a_i = (a_i^1, ..., a_i^n)$, tad klasterizācijas uzdevums ir grupēt kopas A objektus k apakškopās, kas apzīmētas ar $\Pi = \{\pi_1, ..., \pi_k\}$. Tādējādi k ir klasteru skaits.

Ir svarīgi arī saprast, vai kopas A sadalīšana Π klasteros ir *laba* vai *nē*. K-vidējo klasterizācijas gadījumā, ja mēs definējam kādu attālumam-līdzīgu funkciju $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, mēs varam izmērīt datu kopas A labas sadalīšanas pakāpi (Π klasteros), šādi

$$F = \sum_{j=1}^{k} \sum_{a \in \pi_j} d(c_j, a),$$

kur c_j ir klasteru centroīdi (vai klasteru pārstāvji). Klasterizācijas uzdevums ir funkcijas F minimizēšana.

Nestriktajā klasterizācijā tiek ievesta pakāpe $u_{ij} \in [0,1]$, kas norāda uz objekta a_i piederības pakāpi klasterim π_j . Citiem vārdiem sakot, u_j ir piederības funkcija vai nestrikta kopa. Tādējādi $u_j(a_i)$ atklāj elementa a_j piederības pakāpi klasterim π_j, šo pakāpi vienkāršības pēc apzīmē ar u_{ij}.

Turklāt, ja mēs definējam kādu attālumam-līdzīgu funkciju $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, mēs varam izmērīt datu kopas A labas sadalīšanas pakāpi (Π klasteros), šādi

$$Q = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij}^q d(c_j, a_i),$$

kur c_j ir kopu centroīdi un q ir *fuzzifier*. Arī šajā gadījumā klasterizācijas uzdevums ir funkcijas Q minimizēšana.

Nestriktajā klasterizācijā nestriktās ekvivalences gadījumā piederības pakāpes u_{ij} ir aizvietotas ar ekvivalences attiecībām un tādējādi ietekmējot objektu izvietojumu klasteros. Ideja ir balstīta uz to, ka klasteru analīzes mērķis bieži tiek aprakstīts, kā līdzīgu objektu izvietošana vienā klasterī un atšķirīgu objektu izvietošana dažādos klasteros.

Šajā darbā būs apskatītas trīs veidu K-vidējo klasterizācijas algoritmi: parastais K-vidējo algoritms, nestriktais K-vidējo algoritms (tālāk tiek apzīmēts ar C-vidējo), un C-vidējo algoritms

nestriktas ekvivalences gadījumā, kas ir C-vidējo speciālgadījums. Tiek dots ieskats nestriktas attiecības teorijā, kas ir vajadzīgs, lai izveidotu C-vidējo klasterizāciju nestriktas ekvivalences gadījumā. Darbā arī ir apskatītas četras metodes piemērotākai klastera skaita izvēlei, tas ir Calinski-Harabasz indekss, Davies-Bouldin indekss, Silueta platuma kritērijs un Dunna indekss. K-vidējo un C-vidējo klasterizācijas metodes būs apskatītas arī praktiski, programmējot algoritmus *Python* valodā un izveidojot simulācijas algoritmu iterāciju skaita salīdzināšanai.

1. Kopas pārstāvis

Sadaļā "kopas pārstāvis" ir izklāstītas nepieciešamas definīcijas un teorēmas, lai patvaļīgā kopā izvēlētos pārstāvi kuram ir mazāka attālumu summa līdz kopas objektiem. Runājot par klasterizāciju mēs klastera pārstāvi sauksim par *centroīdu*.

1.1. Definīcija. *[2] Funkciju $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ sauc par **attālumam-līdzīgu funkciju**, ja tā apmierina sekojošus nosacījumus:*

1. $d(x,y) = 0 \Leftrightarrow x = y$;
2. $x \mapsto d(x,y)$ *ir nepārtraukta funkcija kopā* $\mathbb{R}^n$;
3. $\lim_{\|x\|\to\infty} d(x,y) = +\infty$ *katram fiksētam* $y \in \mathbb{R}^n$.

Svarīgais piemērs ir mazāko kvadrātu (***MK***) (vai Eiklīda metrika kvadrātā) attālumam-līdzīga funkcija

$$d_{MK}(x,y) = \|x-y\|^2 = \sum_{k=1}^{n} |x_k - y_k|^2. \tag{1}$$

Pierādīsim, ka $d_{MK}(x,y)$ ir attālumam-līdzīga funkcija:

1. ja $x = y \Rightarrow d_{MK}(x, y = x) = \|x-x\|^2 = \|0\|^2 = 0$,

   ja $d_{MK}(x,y) = \|x-y\|^2 = 0 \Rightarrow x = y$;
2. $x \mapsto \|x-y\|^2$ ir nepārtraukta funkcija kopā $\mathbb{R}^n$;
3. katram fiksētam $y \in \mathbb{R}^n$,
   $\lim_{\|x\|\to\infty} d_{MK}(x,y) = \lim_{\|x\|\to\infty} \|x-y\|^2 = \lim_{\|x\|\to\infty} \sum_{k=1}^{n} (x-y)^2 =$
   $= \sum_{k=1}^{n} \lim_{\|x\|\to\infty} (x-y)^2 = \sum_{k=1}^{n} \lim_{\sqrt{\sum_{k=1}^{n} x^2}\to\infty} (x-y)^2 = \sum_{k=1}^{n} \lim_{\sqrt{n}x\to\infty} (x-y)^2 = +\infty.$

Tā kā klasterizācijas uzdevums reducējas uz minimizēšanas uzdevumu, ir svārīga sekojoša lemma, kura pamato, ka uzdevumam eksistē risinājums.

1.2. Lemma. *[1] Pieņemsim, ka $A = a_i : i = 1,...,m \subset \mathbb{R}^n$ ir datu punktu kopa ar svariem $w_1,...,w_m > 0$, pieņemsim, ka $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ ir attālumam-līdzīga funkcija, un funkcija $F : \mathbb{R}^n \times \mathbb{R}^n$ ir $F(x) = \sum_{i=1}^{m} w_i d(x, a_i)$. Tad eksistē tāds punkts $c^* \in \mathbb{R}^n$, ka*

$$F(c^*) = \min_{x\in\mathbb{R}^n} F(x).$$

Pierādījums

Tā kā $F(x) \geq 0, x \in \mathbb{R}^n$, *eksistē* $F^* := \inf_{x\in\mathbb{R}^n} F(x)$. *Lai* (c_k) *būtu tāda virkne kopā* $\mathbb{R}^n$, *ka* $\lim_{k\to+\infty} F(c_k) = F^*$. *Parādīsim, ka virkne* (c_k) *ir ierobežota. Pieņemsim pretējo, ka eksistē tāda apakšvirkne* (c_{k_l}), *ka* $\|c_{k_l}\| \to +\infty$. *Tad, saskaņā ar definīcijas* (1.1) *īpašībām 1. un 2., izriet, ka* $\lim_{\|c_{k_l}\|\to\infty} F(c_{k_l}) = +\infty$, *pretēji tam, ka* $\lim_{l\to\infty} F(c_{k_l}) = F^*$. *Visbeidzot, virknei* (c_k), *kas ir ierobežota, ir konverģējoša apakšvirkne* (c_{k_j}), *kurai* c^* *ir robeža. Tad* $F(c^*) = F(\lim_{j\to\infty} c_{k_j}) = \lim_{j\to\infty} F(c_{k_j}) = \lim_{j\to\infty} F(c_k) = F^*$, *kas parāda, ka* $F(c^*) = \min_{x\in\mathbb{R}^n} F(x)$.

1.3. Definīcija. *[1] Pieņemsim, ka* $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ *ir attālumam-līdzīga funkcija un* $w_1, ..., w_m > 0$ *ir svari. Kopas A,* ***labākais pārstāvis*** *ir jebkurš punkts*

$$c^* \in \arg\min_{x\in\mathbb{R}^n} \sum_{i=1}^{m} w_i d(x, a_i).$$

1.4. Definīcija. *Pieņemsim, ka* $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ *ir attālumam-līdzīga funkcija. Kopas A* ***labākais pārstāvis****,(bez svariem) ir jebkurš punkts*

$$c^* \in \arg\min_{x\in\mathbb{R}^n} \sum_{i=1}^{m} d(x, a_i).$$

1.5. Teorēma. *Pieņemsim, ka funkcija* $F_{MK}(x) = \sum_{i=1}^{m} w_i (x-a_i)^2$ *sasniedz savu globālo minimumu vienīgajā punktā* $c^*_{MK} = \arg\min_{x\in\mathbb{R}} \sum_{i=1}^{m} w_i d_{MK}(x, a_i) = \frac{1}{W}\sum_{i=1}^{m} w_i a_i$, *kur* $W = \sum_{i=1}^{m} w_i$.

Pierādījums

Lai atrastu funkcijas ekstrēmus, doto funkciju jāatvasina, pēc tam atvasināto funkciju jāpielīdzina nullei, lai atrastu punktu, kurā funkcijai būs ekstrēms. Lai saprastu, vai tas punkts ir maksimums vai minimums, to jāatvasina otro reizi un jānovērtē, vai funkcija ir lielāka vai mazāka par 0.

$F'_{MK}(x) = \left(\sum_{i=1}^{m} w_i (x-a_i)^2\right)' = \sum_{i=1}^{m} 2w_i (x-a_i)$

$F'_{MK}(x) = 0 \Rightarrow F'_{MK}(x) = \sum_{i=1}^{m} 2w_i (x-a_i) \Rightarrow x = \frac{\sum_{i=1}^{m} w_i a_i}{\sum_{i=1}^{m} w_i} = \frac{1}{W}\sum_{i=1}^{m} w_i a_i = c^*_{MK}$

$F''_{MK}(x) = \left(F'_{MK}(x)\right)' = \left(\sum_{i=1}^{m} 2w_i (x-a_i)\right)' = \sum_{i=1}^{m} 2w_i > 0$, *jo svari ir lielāki par 0, kas nozīme, ka* $x = c_{MK}$ *ir globālais minimums.*

Lai atrastu punktu MK attālumam-līdzīgas funkcijas gadījumā, kurš pēc iespējas labāk apraksta noteiktu punktu kopu $A = \{a_i = (a_i^1, ..., a_i^n) \in \mathbb{R}^n : i = 1, ..., m\}$, ja ir doti svari, ir sekojoša teorēma:

1.6. Teorēma. *[1] Pieņemsim, ka ir dota MK attālumam-līdzīgas funkcija, punktu kopa* $A = \{a_i = (a_i^1, ..., a_i^n) \in \mathbb{R}^n : i = 1, ..., m\}$ *un ir doti svari* $w_1, ..., w_m > 0$. *Tad labākais kopas A pārstāvis ir tās centroīds*

$$c^*_{MK} = \arg\min_{c\in\mathbb{R}^n} \sum_{i=1}^{m} w_i d_{MK}(c, a_i) = \arg\min_{c\in\mathbb{R}^n} \sum_{i=1}^{m} w_i \|c - a_i\|^2 = \frac{1}{W}\sum_{i=1}^{m} w_i a_i,$$

kur $W = \sum_{i=1}^{m} w_i$, *un atbilstošā samazināšanas funkcija ir*

$$F_{MK}(c) = \sum_{i=1}^{m} w_i \|c - a_i\|^2. \tag{2}$$

Pierādījums

Vajag pierādīt, ka funkcija $F_{MK}(c)$ *sasniedz savu globālo minimumu punktā* c^*_{MK}.

$F_{MK}(c) = \sum_{i=1}^{m} w_i \|c - a_i\|^2 = \sum_{i=1}^{m} w_i \sum_{k=1}^{n} |c^k - a_i^k|^2 = \sum_{k=1}^{n} \sum_{i=1}^{m} w_i |c^k - a_i^k|^2 \overset{teorēma(1.5)}{\geq}$
$\sum_{k=1}^{n} \sum_{i=1}^{m} w_i |\frac{1}{W} \sum_{j=1}^{m} w_j a_j^k - a_i^k|^2 = \sum_{i=1}^{m} \sum_{k=1}^{n} w_i |\frac{1}{W} \sum_{j=1}^{m} w_j a_j^k - a_i^k|^2 = \sum_{i=1}^{m} w_i \sum_{k=1}^{n} |c_{MK}^{*k} - a_i^k|^2 = \sum_{i=1}^{m} w_i \|c_{MK}^{*k} - a_i^k\|^2 = F_{MK}(c^*_{MK})$

Vispirms tika pielietota mazāko kvadrātu attālumam-līdzīga funkcija (1)*, pēc tām galīgas summas tika mainītas vietām. Izmantojot teorēmu* (1.5)*, kur ir pierādīts, ka viendimensionāla funkcija* $F_{MK}(x)$ *sasniedz savu minimumu punktā* c^*_{MK}*, tika izveidota tieši šāda nevienādība, kas parāda uz to, ka tas ir minimālais punkts. Pēc tām atpakaļ tika mainītas vietām galīgas summas un iegūta funkcija* $F_{MK}(c^*_{MK})$*, tāpēc secinājums ir, ka centroids* c^*_{MK} *ir labākais pārstāvis.*

Tika pierādīts fakts par labāko kopas A pārstāvi gadījumā, ja ir svari. Apskatīsim arī gadījumu bez svariem.

1.7. Teorēma. *Funkcija* $F_{MK}(x) = \sum_{i=1}^{m} (x - a_i)^2$ *sasniedz savu globālo minimumu vienīgajā punktā* $c^*_{MK} = \arg\min_{x \in \mathbb{R}} \sum_{i=1}^{m} d_{MK}(x, a_i) = \frac{1}{m} \sum_{i=1}^{m} a_i$.

Pierādījums

Pieradījums ir līdzīgs teorēmai (1.5)*.*

$F'_{MK}(x) = (\sum_{i=1}^{m} (x - a_i)^2)' = \sum_{i=1}^{m} 2(x - a_i)$

$F'_{MK}(x) = 0 \Rightarrow F'_{MK}(x) = \sum_{i=1}^{m} 2(x - a_i) \Rightarrow x = \frac{1}{m} \sum_{i=1}^{m} a_i = c^*_{MK}$

$F''_{MK}(x) = (F'_{MK}(x))' = (\sum_{i=1}^{m} 2(x - a_i))' = 2m > 0$*, jo* $m > 0$*, kas nozīme, ka* $x = c_{MK}$ *ir globālais minimums.*

1.8. Teorēma. *Lai atrastu punktu MK attālumam-līdzīgas funkcijas gadījumā, kas pēc iespējas labāk attēlo noteiktu punktu kopu* $A = \{a_i = (a_i^1, ..., a_i^n) \in \mathbb{R}^n : i = 1, ..., m\}$ *bez svariem, labākais kopas A pārstāvis ir tās centroīds*

$$c^*_{MK} = \arg\min_{c \in \mathbb{R}^n} \sum_{i=1}^{m} d_{MK}(c, a_i) = \arg\min_{c \in \mathbb{R}^n} \sum_{i=1}^{m} \|c - a_i\|^2 = \frac{1}{m} \sum_{i=1}^{m} a_i,$$

un atbilstošā samazināšanas funkcija ir

$$F_{MK}(c) = \sum_{i=1}^{m} \|c - a_i\|^2.$$

Pierādījums

Vajag pierādīt, ka funkcija $F_{MK}(c)$ sasniedz savu globālo minimumu punktā c^*_{MK}.

Pieradījums ir līdzīgs teorēmai (1.6)

$F_{MK}(c) = \sum_{i=1}^{m} \|c - a_i\|^2 = \sum_{i=1}^{m}\sum_{k=1}^{n} |c^k - a_i^k|^2 = \sum_{k=1}^{n}\sum_{i=1}^{m} |c^k - a_i^k|^2 \overset{teorēma(1.7)}{\geq}$
$\sum_{k=1}^{n}\sum_{i=1}^{m} |\frac{1}{m}\sum_{j=1}^{m} a_j^k - a_i^k|^2 = \sum_{i=1}^{m}\sum_{k=1}^{n} |\frac{1}{m}\sum_{j=1}^{m} a_j^k - a_i^k|^2 = \sum_{i=1}^{m}\sum_{k=1}^{n} |c^{*k}_{MK} - a_i^k|^2 =$
$\sum_{i=1}^{m} \|c^{*k}_{MK} - a_i^k\|^2 = F_{MK}(c^*_{MK})$

2. K-vidējo klasterizācija

Klasterizācija sadala datu kopu, kas sastāv no m novērojumiem, k klasteros $\Pi = \{\pi_1, ..., \pi_k\}$ ar iepriekš nezināmiem parametriem. Šajā gadījumā tiek veikta centroīdu $c = \{c_1, ..., c_k\}$ meklēšana, kas atrodas pēc iespējas tālāk viens no otra, ar minimālu izkliedi katra klastera ietvaros. [1, 16, 17]

K-vidējo metode veic klasterizāciju šādi:
Dota ierobežota apakškopa $A = \{a_1, ..., a_m\} \subset \mathbb{R}^n$ un k atšķirīgi punkti $z_1, ..., z_k \in \mathbb{R}^n$,

1. lai noteiktu klasterus $\pi_j, j = 1, ..., k$, piemēro minimālā attāluma principu, lai iegūtu sadalījumu $\Pi = \{\pi_1, ..., \pi_k\}$.

$$\pi_j := \pi_j(z_j) = \{a \in A : d(z_j, a) \leq d(z_s, a)\ ,\ s = 1, ..., k\};$$

   Paskaidrojums: Katru punktu no apakškopas A piešķirsim tādam klasterim π_j, kura attālums līdz centram ir minimāls salīdzinājumā ar attālumu līdz cita klastera centram.

2. dotajam sadalījumam $\Pi = \{\pi_1, ..., \pi_k\}$ no apakškopas A noteikt klasteru centrus $c_j = \arg\min_{x \in \mathbb{R}^n} \sum_{a \in \pi_j} d(x, a), j = 1, ..., k$ un aprēķināt mērķa funkcijas vērtību $F(\Pi)$,

$$F(\Pi) = \sum_{j=1}^{k} \sum_{a \in \pi_j} d(c_j, a). \tag{3}$$

   $z_j = c_j\ ,\ j = 1, ..., k$, nonākam pie 1.soļa un atkārtojam algoritmu.

Ja $\varepsilon > 0$, iteratīvs process apstājas, kad

$$\frac{F_{j-1} - F_j}{F_j} < \varepsilon,$$

kur F_j ir mērķa funkcijas vērtība, kas iegūta j iterācijā.

2.1. Teorēma. *Lai $A \subset \mathbb{R}^n$ ir kopa, $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ attālumam-līdzīga funkcija, un F mērķa funkcija, kas uzdota ar* (3)*. Izmantojot k-vidējo algoritmu, mērķa funkcijas F vērtība nepalielināsies nevienā solī.*

Pierādījums *Lai $\Pi^{(t)} = \{\pi_1^{(t)}, ..., \pi_k^{(t)}\}$ ir sadalījums ar centriem $c^{(t)} = \{c_1^{(t)}, ..., c_k^{(t)}\}$ un $F(\Pi^{(t)})$ ir atbilstošā mērķa funkcijas vērtība.*

Izmantojot 1.punktu (minimālā attāluma principu) no K-vidējo metodes, kopai A ar centriem $c^{(t)}$*, iegūstam jauno sadalījumu* $\Pi^{(t+1)} = \{\pi_1^{(t+1)}, ..., \pi_k^{(t+1)}\}$ *apmierinot*

$$F(\Pi^{(t)}) = \sum_{j=1}^{k} \sum_{a \in \pi_j^{(t)}} d\left(c_j^{(t)}, a\right) \geq \sum_{j=1}^{k} \sum_{a \in \pi_j^{(t+1)}} d\left(c_j^{(t)}, a\right).$$

Tālāk, piemērojot 2.soli katram klasterim $\pi_j^{(t+1)}$ *(lai noteiktu jaunus centrus* $c_j^{(t+1)}$*), iegūvām*

$$\sum_{j=1}^{k} \sum_{a \in \pi_j^{(t+1)}} d\left(c_j^{(t)}, a\right) \geq \sum_{j=1}^{k} \sum_{a \in \pi_j^{(t+1)}} d\left(c_j^{(t+1)}, a\right) =: F\left(\Pi^{(t+1)}\right).$$

Punktu attālums līdz jaunajam centram ir mazāk nekā līdz iepriekšējam, tāpēc ir arī mazāka mērķa funkcijas vērtība. Varam secināt, ka $F(\Pi^{(t)}) \geq F(\Pi^{(t+1)})$.

3. Nestriktas klasterizācijas metodes

Gadījumos, kad paredzams, ka daži kopas A elementi varētu piederēt divām vai vairākiem klasteriem, jāpiemēro nestrikta klasterizācija. Literatūrā nestrikta klasteru veidošana tiek uzskatīta par sava veida mīkstu klasterizāciju. [3, 5, 15, 6, 7, 18, 19]

3.1. Nestriktas attiecības

3.1.1. T-norma

3.1. Definīcija. *[11] Par **t-normu** sauc attēlojumu $T : [0,1] \times [0,1] \to [0,1]$, kas apmierina sekojošus nosacījumus $(x, y, z \in [0,1])$:*

1. $T(x,y) = T(y,x)$
   var arī pierakstīt kā
   $x * y = y * x$ *(simetrijas nosacījums);*

2. $T(T(x,y),z) = T(x,T(y,z))$
   vai $(x * y) * z = x * (y * z)$ *(asociativitāte);*

3. $x1 \leq x2 \Rightarrow T(x1,y) \leq T(x2,y)$
   vai $x1 \leq x2 \Rightarrow x1 * y \leq x2 * y$ *(monotonitāte);*

4. $T(x,1) = x$
   vai $x * 1 = x$.

3.2. Piemērs. *[11]*

1. $T_M(x,y) = x \wedge y$ *(minimuma t-norma)*

2. $T_P(x,y) = xy$ *(reizinājuma t-norma)*

3. $T_L(x,y) = \max(x + y - 1, 0)$ *(Lukaševiča t-norma)*

4. $T_W(x,y) = \begin{cases} \min(x,y), \text{ ja } x \vee y = 1 \\ 0, \text{ ja } x \vee y < 1 \end{cases}$

$$T_W < T_L < T_P < T_M$$

3.3. Definīcija. *[11] Apskatīsim nestrikto attiecību $P : X \times X \to [0,1]$, kur T apzīmē t-normu, tad P sauc par **T-tranzitīvu** $\Leftrightarrow$*

$$\forall x, y, z \in X : T(P(x,y), P(y,z)) \leq P(x,z).$$

3.4. Definīcija. *[11] t-normu T sauc par* ***Arhimeda t-normu*** $\Leftrightarrow$ *ja* $\forall (x,y) \in (0,1)^2$ *eksistē* $n \in \mathbf{N}$*, kurš apmierina nevienādību*

$$y > x_T^{(n)} = \begin{cases} x, \text{ ja } n = 1 \\ T(x,...,x), \text{ ja } n = 2,... \text{ (x ir n reizes)} \end{cases}.$$

Minimuma t-norma nav Arhimeda t-norma, bet reizinājuma un Lukaševiča t-normas pieder Arhimeda t-normu klasei.

Mēs turpinām ar vienu no spēcīgākiem rīkiem t-normu izveidošanai, kas ietver tikai vienas vietas reālo funkciju (aditīvo ģeneratoru) un saskaitīšanu. Turklāt mēs izmantojam to pašu rīku atlikumu un nestriktas ekvivalences veidošanai.

3.5. Definīcija. *[12] Par* ***aditīvo ģeneratoru*** *sauc nepārtrauktu, stingri dilstošu funkciju*

$$f : [0,1] \to [0,\infty], \; kur f(1) = 0.$$

3.6. Definīcija. *[12]* $f : [a,b] \to [c,d]$ *ir monotona funkcija, kur* $[a,b]$ *un* $[c,d]$ *ir pagarinātās reālās līnijas slēgtie apakšintervali* $[-\infty,\infty]$*. Pseido-inversais* $f^{(-1)} : [c,d] \to [a,b]$ *no* f *ir definēts kā*

$$f^{(-1)}(y) = \begin{cases} sup\{x \in [a,b] \mid f(x) < y\}, & ja\ f(a) < f(b), \\ sup\{x \in [a,b] \mid f(x) > y\}, & ja\ f(a) > f(b), \\ a, & ja\ f(a) = f(b). \end{cases}$$

3.7. Definīcija. *[12] t-normas T aditīvs ģenerators* $f : [0,1] \to [0,\infty]$ *ir stingri dilstoša funkcija, kas arī ir nepārtraukta no labas puses un apmierina* $f(1) = 0$*, tāds, ka visiem* $(x,y) \in [0,1]^2$ *ir*

$$f(x) + f(y) \in Ran(f) \cup [f(0),\infty],$$

$$T(x,y) = f^{(-1)}(f(x) + f(y)).$$

3.8. Teorēma. *[12] Funkcija* $T : [0,1]^2 \to [0,1]$ *ir nepārtraukta Arhimēda t-norma tad un tikai tad, ja eksistē nepārtraukts aditīvs ģenerators* f*, kur visiem* $x,y \in [0,1]$ *ir spēkā*

$$T(x,y) = f^{(-1)}(\min(f(x) + f(y), f(0))).$$

Ģenerators t ir definēts līdz pozitīvai multiplikatīvai konstantei.

3.9. Piemērs. *1. Reizinājuma t-norma*

$T_P(x,y) = xy$

Aditīvs generators: $f(x) = -\ln x$

Inversais: $f^{(-1)}(y) = e^{-x}$

2. *Lukaševiča t-norma*

$T_L(x,y) = max(x+y-1,0)$

Aditīvs generators: $f(x) = 1-x$

Inversais: $f^{(-1)}(y) = 1-x$

3.1.2. T-ekvivalence

3.10. Definīcija. *[13] Par **nestrikto ekvivalenci vai T-ekvivalenci** mēs sauksim attiecību, kas apmierina:*

1. *refleksivitāti:* $P(x,x) = 1$*;*
2. *simetriju:* $P(x,y) = P(y,x)$*;*
3. *T-tranzitivitāti:* $T(P(x,y),P(y,z)) \leq P(x,z)$.

3.11. Definīcija. *[13] Nestriktas attiecības sauc par **atdalītām** (separated)* $\Leftrightarrow$

$$\forall x,y \in X : P(x,y) = 1 \Leftrightarrow x = y,$$

ja attiecība uzdota ar matricu, tad šāda īpašība nozīme, ka vieniniekiem jābūt tikai uz galvenas diagonāles un nekur citur.

3.12. Lemma. *[13]*

1. *Katra strikta ekvivalences attiecība ir nestrikta ekvivalences attiecība jebkurai t-normai. Strikta attiecība ir atdalīta.*

   ***Pierādījums:** Lai pierādītu, vajag pārbaudīt vai strikta ekvivalences attiecība ir nestrikta attiecība ar definīciju* (3.10)*.*

   (a) *Refleksivitāte:*

   Zināms, ka $\forall x,y \in X : P(x,y) = 1 \Leftrightarrow x = y$.

   Ja $x = y$ *tad* $P(x,y = x) = P(x,x) = 1$

(b) *Simetriskums:*

Zināms, ka $\forall x,y \in X : P(x,y) = 1 \Leftrightarrow x = y$.

Tā kā strikta attiecība var pieņemt tikai vertības $\{0,1\}$,

ja $x \neq y \Rightarrow P(x,y) \neq 1$, *tad, ja* $P(x,y) \neq 1 \Rightarrow P(x,y) = 0$

Jebkurā gadījuma, kad $x \neq y$, $P = 0 \Rightarrow P(x,y) = P(y,x) = 0$

(c) *Tranzitivitāte*

P var pieņemt tikai dīvas vērtības 0 vai 1

- $P(x,z) = 1 \Leftrightarrow x = z$

  $T(P(x,y),P(y,z)) = T(P(x,y),P(y,z = x)) = T(P(x,y),P(y,x)) = T(P(x,y),P(x,y)) = P(x,y) = \begin{cases} 1, ja\ x = y \\ 0, ja\ x \neq y \end{cases}$

  Šajā gadījumā tranzitivitāte izpildās, jo $\{1,0\} \leq 1$.

- $P(x,z) = 0 \Leftrightarrow x \neq z$

  Ja $y = x \Leftrightarrow T(P(x,x),P(x,z)) = T(1,0) = 0$

  Ja $y = z \Leftrightarrow T(P(x,z),P(z,z)) = T(0,1) = 0$

  Ja $x \neq z \neq y \leftrightarrow T(P(x,y),P(y,z)) = T(0,0) = 0$

  Šajā gadījumā tranzitivitāte ari izpildās, jo $0 \leq 0$.

Izpildās refleksivitāte, simetriskums un tranzitivitāte, tāpēc varam konstatēt, ka katra strikta ekvivalences attiecība ir nestriktas ekvivalences attiecība jebkurai t-normai.

2. *Ja* $T_1 \leq T_2$, *tad jebkura* T_2*-ekvivalence ir arī* T_1*-ekvivalence*

T_1*-ekvivalence*	T_2*-ekvivalence*
(a) $P(x_1,x_1) = 1$	(a) $P(x_2,x_2) = 1$
(b) $P(x_1,y_1) = P(y_1,x_1)$	(b) $P(x_2,y_2) = P(y_2,x_2)$
(c) $T_1(P(x_1,y_1),P(y_1,z_1)) \leq P(x_1,z_1)$	(c) $T_2(P(x_2,y_2),P(y_2,z_2)) \leq P(x_2,z_2)$

Ja $T_1 \leq T_2 \Rightarrow$

$T_1(P(x_1,y_1),P(y_1,z_1)) \leq P(x_1,z_1) \leq T_2(P(x_2,y_2),P(y_2,z_2)) \leq P(x_2,z_2) \Rightarrow$

$T_1(P(x_1,y_1),P(y_1,z_1)) \leq P(x_2,z_2)$

Šis rezultāts nosaka nestriktas ekvivalences attiecības konstruēšanas principus no pseidometrikam.

3.13. Teorēma. *T ir nepārtraukta Arhimēda t-norma ar aditīvo ģeneratoru f. Jebkurai pseidometrikai d, attēlojums*

$$P_d(x,y) = f^{(-1)}(\min(d(x,y), f(0)))$$

ir T-ekvivalence. [14]

Pierādījums

1. Refleksivitāte: Ja $x = y$ un $d(x,x) = 0$, tad

$$P_d(x,x) = f^{(-1)}(\min(d(x,x), f(0))) = f^{(-1)}(\min(0, f(0))) = f^{(-1)}(0) = 1.$$

2. Simetriskums:

Tā kā $d(x,y) = d(y,x) \Rightarrow P_d(x,y) = f^{(-1)}(\min(d(x,y), f(0))) = f^{(-1)}(\min(d(y,x), f(0))) = P_d(y,x)$.

3. T-tranzitivitāte:

Vajag pierādīt, ka

$$T(P(x,y), P(y,z)) \leq P(x,z),$$

kas izmantojot t-normas definīciju ir ekvivalenti

$$f^{(-1)}(min(f(P(x,y)) + f(P(y,z)), f(0)) \leq P(x,z).$$

Ir divi gadījumi, kad:

1. *$min(f(P(x,y)) + f(P(y,z)), f(0)) = f(0)$, tad*

$$f^{(-1)}(min(f(P(x,y)) + f(P(y,z)), f(0)) = f^{(-1)}(f(0)) = 0 \leq P(x,z);$$

2. *$min(f(P(x,y)) + f(P(y,z)), f(0)) = f(P(x,y)) + f(P(y,z))$, tad vajag pierādīt, ka*

$$f^{(-1)}\left(f\left(f^{(-1)}(\min(d(x,y), f(0)))\right) + f\left(f^{(-1)}(\min(d(y,z), f(0)))\right)\right) \leq$$

$$\leq f^{(-1)}(\min(d(x,z), f(0))).$$

Ja minimums nav vienāds ar $f(0)$,

$$f^{(-1)}(d(x,y) + d(y,z)) \leq f^{(-1)}(d(x,z)).$$

Mēs zinām, ka ir spēka trijstūru attiecība

$$d(x,y) + d(y,z) \geq d(x,z),$$

$$f^{(-1)}(P(x,z)) \leq f^{(-1)}(P(x,y)) + f^{(-1)}(P(y,z)).$$

Tā kā $f^{(-1)}$ ir dilstoša funkcija, no tā izriet, ka

$$f\left(f^{(-1)}(P(x,z))\right) \geq f\left(f^{(-1)}(P(x,y)) + f^{(-1)}(P(y,z))\right),$$

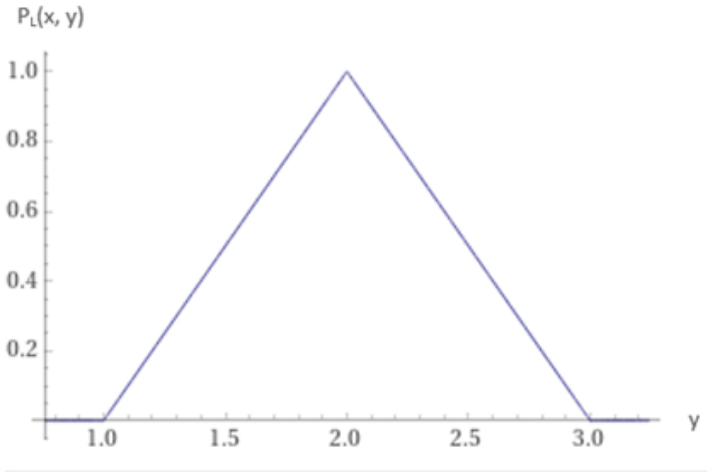
un $f\left(f^{(-1)}(P(x,z))\right) = P(x,z)$.

3.14. Piemērs. *Apskatīsim reālo skaitļu kopu $X = \mathbb{R}$ un metriku $d(x,y) = |x-y|$ tajā. Ņemot vērā, ka $f_L(x) = 1-x$ ir T_L (Lukaševiča t-norma) aditīvs ģenerators un ka $f_P(x) = -ln(x)$ ir T_P (reizinājuma t-norma) aditīvs ģenerators, mēs iegūstam divas nestriktas ekvivalences attiecības:*

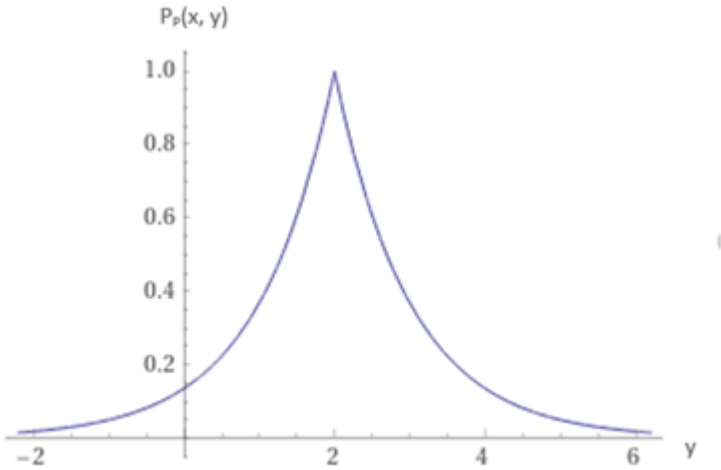
$$P_L(x,y) = \max(1-|x-y|,0);$$
$$P_P(x,y) = e^{-|x-y|}.$$

Apskatīsim grafiski.

Fiksēsim x vērtību (x=2), tad grafiki $P_L(x,y)$ un $P_P(x,y)$ izskatīsies šādi:



3.1. att. **Lukaševiča t-normai**



3.2. att. **Reizinājuma t-normai**

3.2. Piederības funkcijas

Šajā sadaļā mēs apskatīsim piederības funkcijas u_{ij}, kuras mēs izmantosim vēlāk nestriktās klasterizācijas algoritmā.

Mēs noteiksim piederības funkcijas u_{ij} tā, lai funkcija

$$Q(c,U) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij}^{q} d(c_j, a_i), \tag{4}$$

kas atbilst nosacījumiem

$$\sum_{j=1}^{k} u_{ij}(c) = 1, \quad i = 1, ..., m, \tag{5}$$

sasniegtu savu minimumu. Tādējādi mēs definējam Lagranža funkciju [3]

$$\mathscr{J}(c,U,\lambda) := \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij}^{q} d(c_j, a_i) - \sum_{i=1}^{m} \lambda_i \left(\sum_{j=1}^{k} u_{ij} - 1 \right).$$

Pieņemot, ka $a_r \neq c_s$, daļējais atvasinājums

$$\frac{\partial \mathscr{J}(c,U,\lambda)}{\partial u_{rs}} = q u_{rs}^{q-1} d(c_s, a_r) - \lambda_r$$

ir vienāds ar 0, ja

$$u_{rs}(c) = \left(\frac{\lambda_r}{q d(c_s, a_r)} \right)^{\frac{1}{q-1}}. \tag{6}$$

Ievietojot formulu (7) formulā (5), iegūstam

$$\sum_{j=1}^{k} \left(\frac{\lambda_r}{q d(c_j, a_r)} \right)^{\frac{1}{q-1}} = 1 \Rightarrow \lambda_r = \frac{q}{\left(\sum_{j=1}^{k} \left(\frac{1}{d(c_j, a_r)} \right)^{\frac{1}{q-1}} \right)^{q-1}}.$$

Aizstājot λ_r ar formulu (6), mēs iegūstam

$$u_{rs}(c) = \frac{1}{\sum_{j=1}^{k} \left(\frac{d(c_s, a_r)}{d(c_j, a_r)} \right)^{\frac{1}{q-1}}}, \quad a_r \neq c_s. \tag{7}$$

Var gadīties, ka kāds datu punkts $a_i \in A$ sakrīt ar kādu centru c_j. Ja tā notiek, funkcija (7) nav definēta. Tāpēc

$$u_{ij}(c) = \begin{cases} \frac{1}{\sum_{s=1}^{k} \left(\frac{d(c_j, a_i)}{d(c_s, a_i)} \right)^{1/(q-1)}} \text{ , ja } I_i = \varnothing \\ \frac{1}{|I_i|} \text{ , ja } I_i \neq \varnothing \text{ un } j \in I_i \\ 0 \text{ , ja } I_i \neq \varnothing \text{ un } j \notin I_i \end{cases}, \tag{8}$$

kur $I_i = \{s : c_s = a_i\} \subseteq \{1, ..., k\}$; [4]

3.3. Nestrikta C-vidējo metode

Pieņemsim, ka vēlamies datu kopas $A \subset \mathbb{R}^n$ elementus sagrupēt klasteros $\pi_1, ..., \pi_k$, ar iespēju, ka daži elementi $a_i \in A$ nonāk vairākos klasteros līdz zināmai pakāpei.

Nestrikta **C-vidējo metode** veic klasterizāciju šādi [5]:

Dota ierobežota kopa $A \subset \mathbb{R}^n$ un k atšķirīgi punkti $z_1, ..., z_k \in \mathbb{R}^n$,

1. noteikt piederības matricu $U \in [0,1]^{m \times k}$ pēc formulas (8);

2. dota matrica $U \in [0,1]^{m \times k}$, noteikt atbilstošus klasteru centrus $c_1, ..., c_k \in \mathbb{R}^n$,

$$c_j = \left(\sum_{i=1}^{m} u_{ij}^q \right)^{-1} \sum_{i=1}^{m} u_{ij}^q a_i \ , \ \ j = 1, ..., k.$$

Aprēķināt mērķa funkcijas vērtību $Q(c,U)$ pēc formulas (4),

ar nosacījumiem (5) un $\sum_{i=1}^{m} u_{ij}(c) > 0, \quad j = 1, ..., k.$

Ja $\varepsilon > 0$, iteratīvs process apstājas, kad

$$\frac{Q_{j-1} - Q_j}{Q_j} < \varepsilon \ , \ \ Q_j = Q(c^{(j)}, U^{(j)}),$$

vai, kad

$$\|U^{(j)} - U^{(j-1)}\| < \varepsilon.$$

3.4. C-vidējo klasterizācija nestriktas ekvivalences gadījumā

Mēs ievērosim nestrikto C-vidējo algoritmu un izmantosim nestriktās ekvivalences piederības funkciju vietā. Mēs ieviešam n nestriktas ekvivalences attiecību punktveida agregāciju, kas definētas katram raksturlielumam atsevišķi. Lai atklātu pazīmju svarīgumu, ir iespējams izmantot svarus p_i, $i = 1, ..., n$.

3.15. Teorēma. *[15] Lai A ir m datu punktu kopa ($a_i \in \mathbb{R}^n$) ar svariem $w_1, w_2, ..., w_m > 0$. Vislabākais kopas A pārstāvis c^* ar svariem $w_1, w_2, ..., w_m > 0$, ko apzīmē ar $c^* = arg \min\limits_{c \in \mathbb{R}^n} \sum\limits_{i=1}^{m} w_i ||c - a_i||^2$ ir tā svērtais centroids:*

$$c^* = \left(\frac{1}{W} \sum_{i=1}^{m} w_i a_i^1, \frac{1}{W} \sum_{i=1}^{m} w_i a_i^2, ..., \frac{1}{W} \sum_{i=1}^{m} w_i a_i^n \right),$$

kur $W = \sum\limits_{i=1}^{m} w_i$.

Metode veic klasterizāciju šādi: [15]

Ir iepriekš izvēlēts klasteru skaits k un katram datu punktam nejauši piešķirti koeficienti par atrašanos klasteros (tādējādi iegūstot piederības matricu $U \in [0,1]^{m\times k}$).

1. dotai datu kopai A un piederības matricai $U \in [0,1]^{m\times k}$ mēs aprēķinām labākos pārstāvjus jeb centroidus, izmantojot teorēmu (3.15);

2. tālāk mēs pārrēķinām piederības matricu $U \in [0,1]^{m\times k}$ šādi:

   (a) Lai a_i ir objekts un c_j ir klastera centrs, tad izveidosim nestrikto ekvivalences attiecību katrai pazīmei $l = 1,...,m$ $P^l(a_i^l, c_k^l)$, ņemot vērā metriku $d^l(a_i, c_j) = |x_i^l - c_j^l|$. Dažādām t-normām ir iegūstas atbilstošas ekvivalences attiecības:
   $P_L^l(a_i^l, c_j^l) = \max\left(1 - |a_i^l - c_j^l|, 0\right)$;
   $P_P^l(a_i, c_j) = e^{-|a_i^l - c_j^l|}$;
   $P_H^l(a_i, c_j) = \frac{1}{1+|a_i^l - c_j^l|}$.
   Šeit mēs izmantojam vienu un to pašu d^l metriku katram atribūtam l, bet tie varētu atšķirties. Izmantojot nestrikto ekvivalences attiecību P_L, datu kopai jābūt standartizētai;

   (b) Tālāk ideja ir apvienot informāciju par visām nestriktam ekvivalences attiecībām P^l un iegūt globālo nestrikto ekvivalences attiecību P, kas ietver informāciju par visām nestriktām ekvivalences attiecībām P^l un tādējādi informāciju par visām līdzībām kopās $a_1^l,...,a_m^l$. Ieviesīsim attēlojumu $O : [0,1]^k \to [0,1]$, kas agreģē nestriktas ekvivalences attiecības:

   $$P(x,y) = O\left(P^1(x,y),...,P^n(x,y)\right).$$

   Agregācijas funkcijas O definīcija:

   3.16. Definīcija. *[6] Agregācijas funkcija ir attēlojums $O : [0,1]^n \to [0,1]$, kas atbilst šādām īpašībām:*
   1. $O(x^1,...,x^n) \le O(y^1,...,y^n)$, ja $x^i \le y^i$ visiem $i \in 1,...,n$ (monotonitāte);
   2. $O(0,...,0) = 0$ un $O(1,...,1) = 1$ (robežnosacījumi);

   Īpašības:

   i. ja $P^l(x^l, y^l) = 1$ visiem l (t.i., x^l ir līdzīgs y^l visiem l), tad globālai pakāpei arī jābūt 1, tas nozīmē, ka x ir līdzīgs y. Citiem vārdiem sakot: $O(1,...,1) = 1$;

ii. ja "x^l ir līdzīgs y^l" nav pilnībā izpildīts visiem l, tad arī globālajai pakāpei jābūt 0: $O(0,...,0) = 0$;

iii. ja viena pakāpe $P^l(x,y)$ palielinās, bet pārējās paliek nemainīgas, vispārējā pakāpe nedrīkst samazināties, t.i., O nedrīkst būt augoša katrā komponentē;

Ir arī dabiski pieprasīt, lai globālā nestrikta attiecība atbilstu tām pašām īpašībām, kādas piemīt atsevišķām nestriktam attiecībām.

Refleksivitātes saglabāšana ir diezgan skaidra, pēc agregācijas funkcijas robežnosacījuma. Simetrijas saglabāšana ir arī acīmredzama. Daudz interesantāks un sarežģītāks ir jautājums par tranzitivitātes saglabāšanu. Šeit mēs izmantojam rezultātus par tranzitivitātes saglabāšanu, kas pētīta [7], kur parādīts, ka tranzitivitātes saglabāšana ir ekvivalenta t-normas $*$ dominancei ar agregācijas operatoru A.

3.17. Definīcija. *[7] Aplūkojiet n-argumentu agregācijas funkciju* $O : [0,1]^n \to [0,1]$ *un t-normu* $*$. *Mēs sakām, ka O dominē* $*$, *ja visiem* $x^i \in [0,1]$ *ar* $i \in \{1,...,m\}$ *un* $y^i \in [0,1]$ *ar* $i \in \{1,...,n\}$, *tad ir šāda īpašība:*

$$O(x^1,...,x^n) * O(y^1,...,y^n) \leq O(x^1 * y^1,...,x^n * y^n).$$

3.18. Teorēma. *[7] Lai* $*$ *ir t-norma. Agregācijas funkcija O saglabā nestrikto attiecību tranzitivitāti tad un tikai tad, ja O pieder pie klases agregācijas funkcijām, kas dominē* $*$.

3.19. Piemērs. *[7] Jebkuram* $n > 2$ *un jebkuram* $p = (p_1,...,p_n)$ *ar* $\sum_{i=1}^{n} p_i \geq 1$ *un* $p_i \in [0,\infty]$, *agregācijas funkcijai*

$$O_p(x^1,...,x^n) = \max\left(\sum_{i=1}^{n} x^i \cdot p_i + 1 - \sum_{i=1}^{m} p_i, 0\right)$$

dominē Lukaševiča t-norma $*_L$.

3.20. Piemērs. *[7] Jebkuram* $n > 2$ *un jebkuram* $p = (p_1,...,p_n)$ *ar* $\sum_{i=1}^{n} p_i \geq 1$ *un* $p_i \in [0,\infty]$, *agregācijas funkcijai*

$$O_p(x^1,...,x^n) = \prod_{i=1}^{n} x^{ip_i}$$

dominē reizinājuma t-norma $*_P$.

3.21. Teorēma. *Lai $*$ ir t-norma. Ja P_i visiem $i \in \{1,...,n\}$ ir nestrikta ekvivalences attiecības (attiecībā uz t-normu $*$), tad*

$$P(x,y) = O\left(P^1(x,y),...,P^m(x,y)\right)$$

ir arī $*$-ekvivalences attiecība, ja O pieder agregācijas klases funkcijām, kas dominē $*$.

Tāpēc mēs izveidojām jaunu nestrikto ekvivalences attiecību, agreģējot nestriktas ekvivalences attiecības P^l :

$$P(a_i,c_k) = O\left(P^1(a_i^1,c_k^1),P^2(a_i^2,c_k^2),P^3(a_i^3,c_k^3),...,P^m(a_i^m,c_k^m)\right),$$

kur mēs izmantojam agregācijas operatoru O, kas saglabā nestriktas ekvivalences attiecības.

Tādējādi dažādām t-normām mēs iegūstam atbilstošas globālās ekvivalences attiecības jeb līdzības:

$$P_L(a_i,c_k) = \max\left(\sum_{l=1}^{n} P^l(a_i^l,c_k^l)\cdot p_l + 1 - \sum_{l=1}^{n} p_l, 0\right) =$$

$$= \max\left(\sum_{l=1}^{n}(1-|a_i^l-c_k^l|)\cdot p_l + 1 - \sum_{l=1}^{n} p_l, 0\right).$$

Ja $p_l = \frac{1}{n}$, tad (šajā gadījumā mēs atribūtiem nepievienojam papildu prioritātes, jo visi svari ir vienādi):

$$P_L(a_i,c_k) = 1/n\sum_{l=1}^{n}\left(1-|a_i^l-c_k^l|\right).$$

Ja p_l nav vienādi, bet $\sum_{l=1}^{n} p_l = 1$, tad (šajā gadījumā mēs papildus pievienojam atribūtu prioritātes):

$$P_L(a_i,c_k) = \sum_{l=1}^{n} p_l\left(1-|a_i^l-c_k^l|\right).$$

Reizinājuma t-normai iegūstam šādu ekvivalences attiecību:

$$P_P(a_i,c_k) = \prod_{l=1}^{n} e^{-p_l|a_i^l-c_j^l|}.$$

Ja $p_l = \frac{1}{n}$, tad (šajā gadījumā mēs atribūtiem nepievienojam papildu prioritātes, jo visi svari ir vienādi):

$$P_P(a_i,c_k) = \prod_{l=1}^{n} e^{-\frac{1}{n}|a_i^l-c_j^l|}.$$

Tāpēc iestatām $u_{ij} = P(a_i, c_k)$, aprēķinām mērķa funkcijas vērtību Q un nonākam pie 1. soļa.

Līdzīgi kā ar standartu K-vidējo algoritmu vai ar nestrikto C-vidējo algoritmu, arī šis algoritms rada monotonu dilstošu secību no mērķa funkcijas vērtībām Q. Tāpēc process apstājas, kad

$$\frac{Q_{j-1} - Q_j}{Q_j} < \varepsilon,$$

ja $\varepsilon > 0$ un, kur Q_j ir mērķa funkcijas vērtība, kas iegūta j iterācijā.

4. Piemērotākā klasteru skaita izvēle

4.1. Calinski–Harabasz indekss

Izmantojot MK attāluma līdzīgu funkciju, lai atrastu optimālo k klasteru skaitu $\Pi^* = \{\pi_1^*, ..., \pi_k^*\}$, mērķa funkcija F izskatās šādi:

$$F_{MK}(\Pi) = \sum_{j=1}^{k} \sum_{a \in \pi_j} \|c_j - a\|^2.$$

Vērtība $F_{MK}(\Pi^*)$ parāda kopējo klasteru $\pi_1^*, ..., \pi_k^*$ elementu dispersiju no tas centriem $c_1^*, ..., c_k^*$. Jo mazāka ir F_{MK} vērtība, jo mazāka dispersija, kas nozīmē, ka kopas ir iekšēji kompaktākas.

Tāpēc pieņemsim, ka CH indekss ir apgriezti proporcionāls mērķa funkcijas $F_{MK}(\Pi^*)$ vērtībai.

No otras puses, papildus funkcijas F_{MK} minimizēšanai var arī maksimizēt atbilstošo duālo funkciju

$$G(\Pi) = \sum_{j=1}^{k} |\pi| \|c_j - c\|^2,$$

kur $c = \arg\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \|x - a_i\|2 = \frac{1}{m} \sum_{i=1}^{m} a_i$ ir visas kopas A centroids.

Vērtība $G(\Pi^*)$ parāda kopējo centroidu $c_1^*, ..., c_k^*$ svērto atdalīšanu no klasteriem $\pi_1^*, ..., \pi_k^*$. Jo lielāka ir funkcijas G vērtība, jo lielāka ir MK-attālumu summa starp centroidiem c_j^* un visas kopas centroidu c, kur attālumi tiek svērti pēc klastera elementu skaita. Tas nozīmē, ka centroidi c_j^* ir savstarpēji maksimāli atdalīti.

Tāpēc mēs pieņemsim, ka optimālā sadalījuma CH indekss ir proporcionāls duālās mērķa funkcijas $G(\Pi^*)$ vērtībai.

Izmantojot MK attālumam līdzīgu funkciju, Calinski–Harabasz indekss tiek definēts kā

$$CH(k) = \frac{m-k}{k-1} \frac{G(\Pi^*)}{F_{MK}(\Pi^*)}, \quad 1 < k < m.$$

Iekšēji kompaktāki un labāk atdalīti klasteri rada lielāku CH skaitli. [8] [9]

4.1. Teorēma. *Lai $A \subset \mathbb{R}^n$ ir galīga kopa, un lai Π_1 un Π_2 ir divas dažādas k skaitu klasteri no kopas A. Tad*

$$CH(\Pi_1) \geq CH(\Pi_2) \Leftrightarrow F_{MK}(\Pi_1) \leq F_{MK}(\Pi_2).$$

Pierādījums *Lai $m = |A|$ un $c = mean(A)$. Apzimējam $k := \sum_{i-1}^{m} \|c - a_i\|^2$. Ņemot verā, ka $\sum_{i=1}^{m} \|c - a_i\|^2 = F_{MK}(\Pi) + G(\Pi)$, tad*

$$G(\Pi_1) = k - F_{MK}(\Pi_1) \;\; un \;\; G(\Pi_2) = k - F_{MK}(\Pi_2).$$

Tāpēc,

$$CH(\Pi_1) \geq CH(\Pi_2) \Leftrightarrow \frac{m-k}{k-1}\frac{G(\Pi_1)}{F_{MK}(\Pi_1)} \geq \frac{m-k}{k-1}\frac{G(\Pi_2)}{F_{MK}(\Pi_2)}$$
$$\Leftrightarrow \frac{k-F_{MK}(\Pi_1)}{F_{MK}(\Pi_1)} \geq \frac{k-F_{MK}(\Pi_2)}{F_{MK}(\Pi_2)}$$
$$\Leftrightarrow \frac{k}{F_{MK}(\Pi_1)} - 1 \geq \frac{k}{F_{MK}(\Pi_2)} - 1$$
$$\Leftrightarrow F_{MK}(\Pi_1) \leq F_{MK}(\Pi_2).$$

4.2. Davies–Bouldin indekss

Davies–Bouldin (DB) indekss ir definēts tā, lai iekšēji kompaktākam sadalījumam, kura klasteri ir labāk savstarpēji atdalīti, būtu mazāka DB vērtība.

Lai $c \in \mathbb{R}^2$ ir punkts plaknē, ap kuru ir m nejauši punkti a_i, kas ģenerēti no divfaktoru normālā sadalījuma $\mathcal{N}(c, \sigma^2 I)$ ar kovariācijas matricu $\sigma^2 I$, kur I ir identitātes 2×2 matrica. Šī punktu kopa veido sfērisku datu kopu, ko apzīmē ar A.

Ir zināms, ka aptuveni 68% no visiem kopas A punktiem atrodas apļa $K(c, \sigma)$ ar centru c un rādiusu σ iekšpusē. Šo apli sauksim par datu kopas A galveno apli.

Lai Π^* ir optimāls kopas A sadalījums ar klasteriem $\pi_1^*, ..., \pi_k^*$ un to centroidiem $c_1^*, ..., c_k^*$. Fiksēsim vienu no klasteriem, piemēram π_j^*, un aplūkosim tā sakarību ar pārējiem klasteriem. Ievērojiet, ka lielums

$$D_j := \max_{s \neq j} \frac{\sigma_j + \sigma_s}{\|c_j^* - c_s^*\|}, \text{ kur } \sigma_j^2 = \frac{1}{|\pi_j^*|} \sum_{a \in \pi_j^*} \|c_j^* - a\|^2, \tag{9}$$

identificē klastera π_j^* lielāko pārklāšanos ar jebkuru citu klasteri. Lielums

$$\frac{1}{k}(D_1 + ... + D_K) \tag{10}$$

ir skaitļu (9) vidējais lielums, un tas ir vēl viens sakritības rādītājs iekšējās kompaktumu un klasteru ārējo atdalīšanu sadalījumā. Ir skaidrs, ka mazāks skaitlis (10) nozīmē, ka klasteri ir iekšēji kompaktāki un labāk atdalīti. Tāpēc kopas A optimālā sadalījuma *DB* indekss ar klasteriem $\pi_1^*, ..., \pi_k^*$ un to centroidiem $c_1^*, ..., c_k^*$ definē šādi

$$DB(k) := \frac{1}{k}\sum_{j=1}^{k} \max_{s \neq j} \frac{\sigma_j + \sigma_s}{\|c_j^* - c_s^*\|}, \text{ kur } \sigma_j^2 = \frac{1}{|\pi_j^*|} \sum_{a \in \pi_j^*} \|c_j^* - a\|^2, \quad 1 < k < m.$$

Iekšēji kompaktāki un labāk atdalīti klasteri rada mazāko *DB* indeksu. [10]

4.3. Silueta platuma kritērijs

Izmantojot MK attāluma līdzīgu funkciju, lai atrastu optimālo k klasteru skaitu $\Pi^* = \{\pi_1^*, ..., \pi_k^*\}$, silueta platuma kritēriju (SWC) definē šādi:

$$SWC(k) = \frac{1}{m}\sum_{i=1}^{m}\frac{\beta_{ir} - \alpha_{ir}}{\max\{\alpha_{ir}, \beta_{ir}\}}, \quad 1 < k < m,$$

kur katram $a_i \in \pi_r^*$,

$$\alpha_{ir} = \frac{1}{|\pi_r^*|}\sum_{b\in\pi_r^*} d(a_i, b), \quad \beta_{ir} = \min_{q\neq r}\frac{1}{|\pi_q^*|}\sum_{b\in\pi_q^*} d(a_i, b). \tag{11}$$

Iekšēji kompaktāki un labāk atdalīti klasteri rada lielāku *SWC* skaitli.

Tā kā skaitliskā procedūra *SWC* indeksa aprēķināšanai ir diezgan sarežģīta, parasti izmanto vienkāršoto silueta platuma kritēriju (SSWC), kas vidējās vērtības (11) vietā izmanto attālumu starp elementu $a_i \in \pi_r^*$ un centriem $c_1^*, ..., c_k^*$:

$$SSWC(k) = \frac{1}{m}\sum_{i=1}^{m}\frac{\beta_{ir} - \alpha_{ir}}{\max\{\alpha_{ir}, \beta_{ir}\}}, \quad 1 < k < m,$$

kur $\alpha_{ir} = d(a_i, c_r^*)$, $\beta_{ir} = \min_{q\neq r} d(a_i, c_q^*)$.

Iekšēji kompaktāki un labāk atdalīti klasteri rada lielāku *SSWC* skaitli. [8]

4.4. Dunna indekss

Dunna indekss ir definēts tā, lai iekšēji kompaktāks sadalījums ar labāk savstarpēji atdalītiem klasteriem ir mazāka Dunna vērtība.

Lai $\Pi^* = \{\pi_1^*, ..., \pi_k^*\}$ ir optimālais k-sadalījums, kas iegūts, izmantojot attālumam līdzīgu funkciju d. Atdalīšanas mēru starp klasteru pāriem nodalījumā nosaka, aprēķinot attālumus starp katru klasteru pāri $\{\pi_i^*, \pi_j^*\}$

$$D(\pi_i^*, \pi_j^*) = \min_{a\in\pi_i^*, b\in\pi_j^*} d(a, b),$$

un klasteru iekšējā kompaktuma mērvienībai $\pi_i^* \in \Pi^*$ ņemam to diametrus $diam\pi_i^* = \max_{a,b\in\pi_i^*} d(a, b)$.

Tāpēc optimālā sadalījuma Π^* Dunna indekss ir proporcionāls skaitlim $\min_{1\leq i<j\leq k} D(\pi_i^*, \pi_j^*)$ un apgriezti proporcionāls skaitlim $\max_{1\leq s\leq k} diam\pi_s^*$ un ir definēts kā šo divu vērtību koeficients:

$$Dunn(k) = \frac{\min_{1\leq i<j\leq k} D(\pi_i^*, \pi_j^*)}{\max_{1\leq s\leq k} diam\pi_s^*}, \quad 1 < k < m.$$

Iekšēji kompaktāki un labāk atdalīti klasteri rada mazāko Dunna indeksu. [8]

5. Praktiskais salīdzinājums

Lai pārbaudītu divu, K-vidējo un nestrikto C-vidējo, klasterizacijas algoritmu efektivitāti, bija veiktas simulācijas.

5.1. Datu simulācija

Standarta normālo kombināciju modelis paredz, ka dati $(X_1,...,X_n)$ tiek modelēti kā IID dati no sadalījuma ar blīvumu

$$f(x;\theta) = \sum_{j=1}^{G} \pi_j \phi(x_i, \mu_j, \sigma_j^2), \tag{12}$$

kur $\phi(.,\mu,\sigma^2)$ ir normāla sadalījuma blīvums ar vidējo μ un dispersiju σ^2, π_j ir j-tās kombinācijas komponenta īpatsvars un $\sum_{j=1}^{G} \pi_j = 1$. Parametru vektors θ satur visas proporcijas, vidējus lielumus un variācijas.

Parametra θ maksimālās ticamības novērtējums (MLE) no datu kopas $x_n = \{x_1, x_2, ..., x_n\}$ var iegūst ar šādu formulu

$$\hat{\theta}_n = \arg\max_{\theta\in\Theta} \sum_{i=1}^{n} \log f(x_i, \theta),$$

kur $\Theta = \{\theta | \sigma_j^2 \geq s, j = 1,2,...,G, \sum_{j=1}^{G} \pi_j = 1\}$, vēloties $s > 0$. Lai izvairītos no logaritmiskās ticamības funkcijas deģenerācijas, ir nepieciešams no apakšas ierobežot dispersiju.

Pamatojoties uz θ_n novērojumiem, x_i var klasificēt kā komponentu

$$k = \arg\max_{p=1,...,G} \hat{\tau}_{ip},$$

kur lielums $\hat{\tau}_{ip}$ ir novērtēta varbūtība, ka x_i ir ģenerēta ar p-to kombinācijas komponentu:

$$\hat{\tau}_{ip} = \frac{\hat{\pi}_p \phi(x_i, \hat{\mu}_p, \hat{\sigma}_p^2)}{f(x_i, \hat{\theta})}.$$

Uz modeļiem balstītas klasterizācijas filozofija ir tāda, ka (12) ne vienmēr tiek pieņemts, ka "patiess", bet drīzāk, normālais sadalījums tiek uzskatīts par klastera formas prototipu, ņemot vērā, ka daudzus sadalījumus var precīzi aproksimēt ar normālo.

5.2. Rezultāti

Praktiski bija veikti 5 testi ar 1000 simulācijam katrā (*Python* kodu var apskatīt *1. pielikumā*), un algoritms izskatās šādi:

1. katrā simulācijā bija ģenerēta kopa no 100 punktiem ar funkciju **sklearn.datasets.make_blobs**;
2. bija palaists K-vidējo algoritms ar iebūvētas funkcijas **sklearn.cluster.KMeans** palīdzību un tiek iegūta iterācija, kurā $\frac{F_{j-1}-F_j}{F_j} < e^{-4}$;
3. bija palaists nestrikto C-vidējo algoritms ar pašrakstīto funkcijas palīdzību un tiek iegūta iterācija, kurā $\frac{Q_{j-1}-Q_j}{Q_j} < e^{-4}$;
4. iterācijas numuri bija saglabāti masīvā un saskaitīta vidēja vērtība un standartnovirze katrai metodei.

Datu ģenerēšanas posmā, ja standartnovirze ir fiksēta un vienāda ar 0.4, varam redzēt tādu sakarību:

1. mainot parametru **center_box =(x,y)** funkcijā **make_blobs** (parametrs nosaka klastera centra robežas):
   - ja $|x - y| \geq 5$, tad vidējais iterāciju skaits un standartnovirze mazāk K-vidējo algoritmam.

	C mean	K mean	C std	K std
0	4.654000	2.558000	2.750688	0.699025
1	4.758000	2.576500	2.745621	0.759044
2	4.708333	2.571667	2.747592	0.743548
3	4.751750	2.585500	2.753202	0.731225
4	4.765200	2.582200	2.752829	0.721695

5.1. att. **Iterāciju vidēja vērtība un standartnovirze priekš divām metodēm, ja center_box = (5,10)**

Attēlā redzamā tabulā varam ieraudzīt, ka iterāciju vidējais skaits katrā testā K-vidējo algoritmam ir mazāks (2.5 iterāciju robežās) nekā C-vidējo algoritmam (4.7 iterāciju robežās). Standartnovirze arī ir mazāk, nepārsniedz 1;

- ja $|x-y| \leq 4$, tad vidējais iterāciju skaits un novirze mazāk nestriktam C-vidējo algoritmam.

	C mean	K mean	C std	K std
0	2.000000	7.298000	0.000000	3.031699
1	2.002500	7.230000	0.111775	3.050262
2	2.001667	7.265333	0.091272	3.088516
3	2.004000	7.268000	0.146574	3.073056
4	2.003200	7.274600	0.131110	3.071546

5.2. att. **Iterāciju vidēja vērtība un standartnovirze priekš divām metodēm, ja center_box = (1,4)**

Attēlā redzamā tabulā varam ieraudzīt, ka iterāciju vidējais skaits katrā testā C-vidējo algoritmam ir mazāks (2 iterāciju robežās) nekā K-vidējo algoritmam (7.2 iterāciju robežās). Standartnovirze arī ir mazāk, nepārsniedz 1;

- ja $|x-y| \in (4,5)$, tad ir nenoteiktība, katrā testā vidējais iterāciju skaits ir vai vienāds vai mazāks vienam no algoritmiem un standartnovirzes ir diezgan lielas.

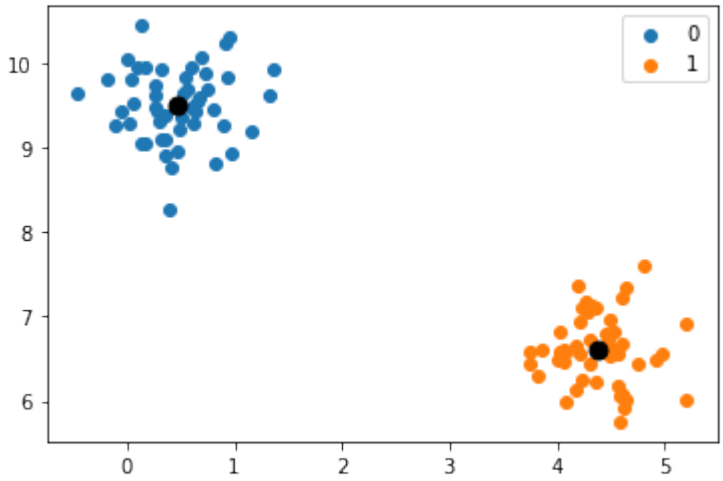
	C mean	K mean	C std	K std
0	3.95400	3.854000	2.797836	2.020565
1	3.88800	3.912000	2.751264	2.110274
2	3.87500	3.859667	2.744092	2.054257
3	3.83375	3.845500	2.728390	2.057214
4	3.82320	3.843800	2.727846	2.051293

5.3. att. **Iterāciju vidēja vērtība un standartnovirze priekš divām metodēm, ja center_box = (0,4.5)**

Attēlā redzamā tabulā varam apskatīt, ka iterāciju vidējais skaits katrā testā ir apmēram vienāds katram algoritmam (svārstās 3.8 iterācijas robežās), bet precīzi noteikt kuram ir mazāks skaits nevaram. Bet standartnovirze ir mazāk K-vidējo algoritmam;

2. vizuāli:

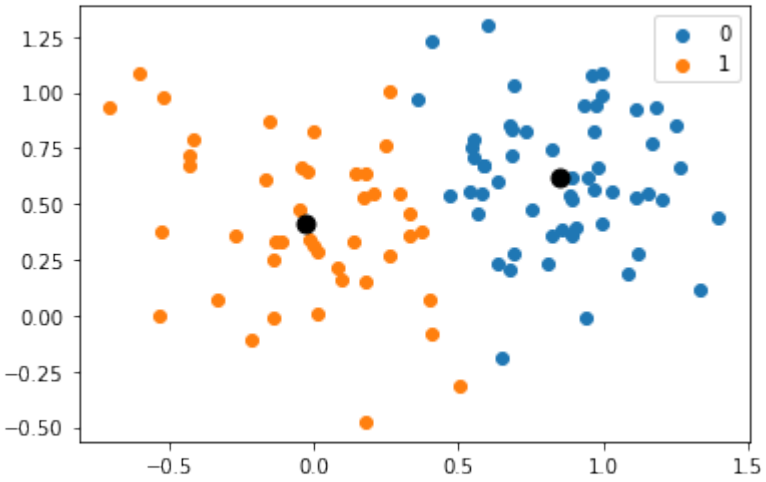
- ja klasteru centri ir tālu viens no otra un acīmredzami var noteikt klasteru kopas, tad mazāko iterāciju skaitu padara K-vidējo algoritms.



5.4. att. **Simulēto datu sadalījums pa klasteriem**

Attēlā redzamā gadījumā ir acīmredzami klasteri ar diezgan tālu esošiem centriem. K-vidējo algoritms izdarīja 2 iterācijas, bet C-vidējo algoritms 7 iterācijas;

- ja klasteru centri ir tuvu viens otram un klasteri nav acīmredzami, tad mazāko iterāciju skaitu padara nestriktais C-vidējo algoritms.



5.5. att. **Simulēto datu sadalījums pa klasteriem**

Attēlā redzamā gadījumā nevaram uzreiz noteikt klasterus un centri ir diezgan tuvu viens otram. Nestriktais C-vidējo algoritms izdarīja 2 iterācijas, bet K-vidējo algoritms 5 iterācijas.

Varam izdarīt secinājumu, ka, kad $|x-y| \geq 5$, tad pārsvara ģenerēti dati uzreiz sadalās acīmredzamajos kopās. Gadījumā, kad $|x-y| \leq 4$, tad otrādi, klasteri nav uzreiz redzami. Kad $|x-y| \in (4,5)$, tad tas ietver sevī abus gadījumus.

Secinājumi

Darba gaitā tiek izpētītas trīs klasterizācijas metodes: K-vidējo klasterizācija, nestrikta C-vidējo klasterizācija un C-vidējo klasterizācija nestriktas ekvivalences gadījumā.

Idejiski K-vidējo un nestriktais algoritms ir vienādi. Vispirms tiek izvēlēti sākotnēji klasteru centri, pēc tam, izmantojot attālumam-līdzīgo funkciju, katrs datu punkts tiek piešķirts kādām klasterim un tiek aprēķināti jauni centroidi un mērķa funkcija, pēc tām algoritms atkārtojas līdz noteiktam nosacījumam. Atšķirība starp metodēm ir klasteru piešķiršanas posmā, kur tiek izmantotas atšķirīgas formulas, un, protams, centroidu un mērķa funkcijas aprēķinos.

C-vidējo klasterizācija un nestriktas ekvivalences gadījums ir pilnība identiskie, izņemot piederības matricas aprēķinu, kur nestriktas ekvivalences gadījumā tā tiek iegūta, agreģējot nestriktas ekvivalences attiecību.

Arī bija apskatītas četras metodes piemērotākai klastera skaita izvēlei, tas ir Calinski-Harabasz indekss, Davies-Bouldin indekss, Silueta platuma kritērijs un Dunna indekss. Praktiski, izmantojot reālos, nevis simulētus datus, šis posms ir ļoti svarīgs, lai saprastu cik klasterus ņemt un, lai pēc klasterizācijas saņemtu kvalitatīvu rezultātu.

Pēc divu metožu, K-vidējo un C-vidējo, praktiska salīdzinājuma var secināt, ka rezultatīvums ir atkarīgs no standartnovirzes un no izvēlētas funkcijas parametra **center_box** vērtības, kas norāda uz simulēto datu kopu attālumu starp centriem. Izvēloties standartnovirzi 0.4, var secināt, ka, ja attālums ir liels un acīmredzami var atdalīt klasterus, tad labāka metode pēc ātruma ir K-vidējo metode. Ja attālums nav liels un noteikt klasterus uzreiz nevar, tad labāk darbojas nestrikta C-vidējo metode. Var apskatīt citus gadījumus ievēšot atkarību no standartnovirzes, un ir hipotēze, ka neatkarība no tas, ja datus uzreiz var acīmredzami sadalīt uz klasteriem, tad mazāko iterāciju skaitu padarīs K-vidējo algoritms.

Var izvirzīt otro hipotēzi, ja atrast minimālo attālumu starp punktiem no dažādiem klasteriem, tad, jo mazāks ir attālums, jo ātrāk strādās C-vidējo algoritms, būs mazākais iterāciju skaits.

Ja domāt par šo metožu praktisko pielietojumu dzīvē, tad bez datu vizuālas interpretēšanas neiztikt. Ja, analizējot, uzreiz ir redzams, ka visi dati ir kopā un tos nevar uzreiz sadalīt klasteros, tad viennozīmīgi labāka metodes izvēle ir C-vidējo klasterizācijas algoritms. Ja dati tālu viens no otra, tad labāka metode būs K-vidējo klasterizācijas algoritms, bet dzīvē ļoti reti var sastapt tādu gadījumu.

Literatūra

[1] Rudolf Scitovski, Kristian Sabo, Francisco Martínez-Álvarez, Šime Ungar, Cluster Analysis and Applications, 2021

[2] M.Teboulle, A unified continuous optimization framework for center-based clustering methods. J. Mach. Learn. Res. 8, 65–102 (2007)

[3] H. Frigui, R. Krishnapuram, Clustering by competitive agglomeration. Pattern Recogn. 30, 1109–1119 (1997)

[4] F. Höppner, F. Klawonn, A contribution to convergence theory of fuzzy c-means and derivatives. IEEE Trans. Fuzzy Syst. 11, 682–694 (2003)

[5] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th edn. (Academic Press, Burlington, 2009)

[6] M. Grabisch, J.-L. Marichal, R. Mesiar and E. Pap, Aggregation Functions (Encyclopedia of Mathematics and its Applications), *Cambridge University Press, UK*, 2009.

[7] S.Saminger, R.Mesiar and U.Bodenhofer, Domination of aggregation operators and preservation of transitivity, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **10(Suppl.)** (2002), 11-35.

[8] L. Vendramin, R.J.G.B. Campello, E.R. Hruschka, On the comparison of relative clustering validity criteria, in Proceedings of the SIAM, *International Conference on Data Mining*, 2009, Sparks, Nevada, USA, pp. 733–744

[9] T. Calinski, J. Harabasz, A dendrite method for cluster analysis. Commun. Stat. 3, 1–27 (1974)

[10] I. Vidovic, D. Bajer, R. Scitovski, A new fusion algorithm for fuzzy clustering. Croat. Oper. ´ Res. Rev. 5, 149–159 (2014)

[11] Aleksandrs Šostaks, L-kopas un L-vērtīgas struktūras, 2001

[12] Erich Peter Klement, Radko Mesiar, Endre Pap, Triangular norms

[13] Ulrich Bodenhofer, A Similarity-Based Generalization of Fuzzy Orderings, Universitätsverlag Rudolf Trauner, 1999

[14] Bernard De Baets, Metrics and T-Equalities, *cJournal of Mathematical Analysis and Applications*, **267**, 531–547 (2002)

[15] O.Grigorenko, V. Mihailovs, Aggregated fuzzy equivalence relations in clustering process, *accepted for publication in the Communications in Computer and Information Science (CCIS) series, by Springer*

[16] Шитиков В. К., Мастицкий С. Э., Классификация, регрессия, алгоритмы Data Mining с использованием R. - 2017

[17] Edy Umargono, Jadmiko Endro Suseno and Vincensius Gunawan S.K., K-Means Clustering Optimization using the Elbow Method and Early Centroid Determination Based-on Mean and Median, 2020

[18] Mohamed Fadhel Saad and Adel M. Alimi, Modified Fuzzy Possibilistic C-means, Proceedings of the *International MultiConference of Engineers and Computer Scientists*, March 18 - 20, 2009, Hong Kong

[19] Virender Kumar Malhotra, Harleen Kaur, M.Afshar Alam, An Analysis of Fuzzy Clustering Methods, *International Journal of Computer Applications (0975 – 8887)*, May 2014

Pielikumi

1. pielikums

1. **Python funkcijas, lai īstenotu C-vidējo algoritmu**

   - *Piederības matrica*

```
def initializeMembershipWeights():
"""
membership_mat = []
for i in range(n):
  wts = []
  sum=0;
  for j in range(k):
    weight = np.random.random_integers(1,10)
    wts.append(weight)
    sum = sum + weight
  weights = [w/sum for w in wts]
  membership_mat.append(weights)
  print(membership_mat)
  """
weight = np.random.dirichlet(np.ones(k),n)
weight_arr = np.array(weight)
return weight_arr
```

   - *Klasteru centri*

```
def computeCentroids(weight_arr):
C = []
for i in range(k):
  weight_sum = np.power(weight_arr[:,i],m).sum()
  Cj = []
  for x in range(d):
    numerator = ( df.iloc[:,x].values *
    np.power(weight_arr[:,i],m)).sum()
```

```
    c_val = numerator/weight_sum;
    Cj.append(c_val)
  C.append(Cj)
return C
```

- *Atjaunota piederības m-ca*

```
def updateWeights(weight_arr,C):
denom = np.zeros(n)
for i in range(k):
  dist = (df.iloc[:,:].values - C[i])**2
  dist = np.sum(dist, axis=1)
  dist = np.sqrt(dist)
  denom  = denom + np.power(1/dist,1/(m-1))

for i in range(k):
  dist = (df.iloc[:,:].values - C[i])**2
  dist = np.sum(dist, axis=1)
  dist = np.sqrt(dist)
  weight_arr[:,i] = np.divide(np.power(1/dist,1/(m-1)),denom)
return weight_arr
```

- *Mērķa f-ja*

```
def countP(weight_arr,C):
l=[]
for i in range(k):
  dist = (df.iloc[:,:].values - C[i])**2
  dist = np.sum(dist, axis=1)
  l.append(dist)
  dis=np.array(l)

ott=[]
for i in range(n):
  ot = (np.power(weight_arr[:][i],m)*dis[:,i]).sum()
```

```
    ott.append(ot)
    P=np.sum(ott)
  return P
```

- *Algoritms*

```
def FuzzyMeansAlgorithm():
weight_arr = initializeMembershipWeights()
P=[]
R=[]
RR = 1
while RR > math.exp(-4):
  C = computeCentroids(weight_arr)
  updateWeights(weight_arr,C)
  PP = countP(weight_arr,C)
  P.append(PP)
  if len(P)>=2:
    RR = (P[-2]-PP)/PP
    R.append(RR)
return (weight_arr,C, P)
```

2. **Python funkcijas, lai īstenotu K-vidējo algoritmu**

- *Algoritms*

```
def train_kmeans(X):
  kmeans = KMeans(n_clusters=k,tol=1e-4,init='random',
  verbose=2, n_init=1)
  kmeans.fit(X)
return kmeans
```

- *Palīgfunkcija, lai dabūtu mērķa f-jas F vērtību*

```
def redirect_wrapper(f, inp):
  old_stdout = sys.stdout
```

```
            new_stdout = io.StringIO()
            sys.stdout = new_stdout

            returned = f(inp)                #<- Call function
            printed = new_stdout.getvalue()  #<- store printed output

            sys.stdout = old_stdout
            return returned, printed
```

3. **Simulācijas**

```
CC=[]
KK=[]
cc=[]
kk=[]
for z in range(5):
  IterP=[]
  IterF = []
  for i in range(1000):
    features, clusters = make_blobs(n_samples = 100,
                                    n_features = 2,
                                    centers = 3,
                                    cluster_std = 0.4,
                                    shuffle = True,
                                    center_box=(0,random.uniform(1,4)))
    df=pd.DataFrame(features)
    n=len(df)
    final_weights,Centers, P = FuzzyMeansAlgorithm() #C-means
    itP = len(P) #iteraciju skaits
    IterP.append(itP)
    returned, printed = redirect_wrapper(train_kmeans, features) #K-means
    inertia = [float(i[i.find('inertia')+len('inertia')+1:]) for i in
    printed.split('\n')[1:-2]]
    itF=len(inertia) #iteraciju skaits
```

```
    IterF.append(itF)
  K=np.mean(IterF)
  C=np.mean(IterP)
  p=np.std(IterF)
  q=np.std(IterP)
  CC.append(C)
  KK.append(K)
  cc.append(q)
  kk.append(p)

c1=pd.DataFrame(cc)
k1=pd.DataFrame(kk)
C1=pd.DataFrame(CC)
K1=pd.DataFrame(KK)

l=pd.concat([C1,K1,c1,k1], axis=1)
l.columns=['C mean','K mean','C std','K std']
l
```

Bakalaura darbs ”Trīs veidu K-vidējo klasterizācijas algoritmu teorētiskais pamatojums” izstrādāts LU Fizikas, matemātikas un optometrijas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā noslēguma darba elektroniskā versija atbilst LUIS augšupielādētā darba elektroniskai kopijai.

Autors: Ksenija Satinova

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītājs: docente Olga Grigorenko

Recenzents: asoc. prof. Ingrīda Uļjane

Darbs iesniegts Matemātikas nodaļā 2022.gada 3. jūnijā

Dekāna pilnvarotā persona: metodiķe Inita Šneidere

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

_____. prot. Nr. _____.