Statistističke funkcije dubine, J. W. Tukey i analiza velikih podataka

Milan Merkle i Milica Bogićević

Elektrotehnički fakultet Univerziteta u Beogradu
emerkle@etf.rs antomripmuk@yahoo.com

MI SANU - Seminar za računarstvo i primenjenu matematiku

Beograd, 18. decembar 2018.

Svetski kongres matematičara-ICM 2018

Rio de Janeiro, avgust 2018 - www.icm2018.org

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018

Fields-ove medalje:

- *Akshay Venkatesh (Stanford), Number theory*
- *Peter Scholze (Bonn), Algebraic geometry*
- *Alessio Figalli (ETH), Optimal control*
- *Caucher Birkar (Cambridge), Algebraic geometry 2x !*

Gauss-ova nagrada (za primene matematike):

- *David Donoho (Stanford), for his fundamental contributions to the mathematical, statistical and computational analysis of important problems in signal processing.-Compressed sensing*

Nevenlinna medal:

- *Constantinos Daskalakis (MIT), Nash equilibrium- computer science, game theory*

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\Longrightarrow$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\Longrightarrow$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\implies$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\implies$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\implies$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\implies$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\implies$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics

ICM 2018 -u nekoliko rečenica

- Potpuno digitalizovano
- Nista papirno (0)
- $\Longrightarrow$ bez postera (:
- U paviljonima Rio-Centro izložbenog i kongresnog centra
- U organizaciji instituta IMPA
- ICM 2018 YouTube-Plenarna predavanja i predavanja po pozivu-video
- https://impa.br/icm2018/ - proceedings po oblastima
- Tatiana Roque: IMPA???s coming of age in a context of international reconfiguration of mathematics


impa
Instituto de
Matemática
Pura e Aplicada
Unité Mixte Internati

Marcelo Viana, direktor

Impa-biblioteka

ICM2018- u vezi sa naukom o podacima

- Michael Jordan (Berkeley): Dynamic, symplectic, and stochastic perspectives on gradient-based optimization
- Sanjeev Arora (MIT): The mathematics of machine learning and deep learning
- Gil Kalai (Hebrew University Jerusalem): Noise Stability, Noise Sensitivity and the Quantum Computer Puzzle
- Jonathan E. Taylor (Stanford) : A selective survey of selected inference

ICM2018- u vezi sa naukom o podacima

- Michael Jordan (Berkeley): Dynamic, symplectic, and stochastic perspectives on gradient-based optimization
- Sanjeev Arora (MIT): The mathematics of machine learning and deep learning
- Gil Kalai (Hebrew University Jerusalem): Noise Stability, Noise Sensitivity and the Quantum Computer Puzzle
- Jonathan E. Taylor (Stanford) : A selective survey of selected inference

ICM2018- u vezi sa naukom o podacima

- Michael Jordan (Berkeley): Dynamic, symplectic, and stochastic perspectives on gradient-based optimization
- Sanjeev Arora (MIT): The mathematics of machine learning and deep learning
- Gil Kalai (Hebrew University Jerusalem): Noise Stability, Noise Sensitivity and the Quantum Computer Puzzle
- Jonathan E. Taylor (Stanford) : A selective survey of selected inference

ICM2018- u vezi sa naukom o podacima

- Michael Jordan (Berkeley): Dynamic, symplectic, and stochastic perspectives on gradient-based optimization
- Sanjeev Arora (MIT): The mathematics of machine learning and deep learning
- Gil Kalai (Hebrew University Jerusalem): Noise Stability, Noise Sensitivity and the Quantum Computer Puzzle
- Jonathan E. Taylor (Stanford) : A selective survey of selected inference

Nauka o podacima na ICM2018

- Plenarna predavanja i predavanja po pozivu
- Za razliku od sličnih zaokreta u prošlosti (Finansijska matematika, genetika), u kojima se postojeća teorija primenjuje u novim oblastima, ovde je slučaj da primena ide ispred teorije.
- U mnogim procedurama koje se koriste u nauci o podacima nedostaju objašnjenja- zadatak matematičara.
- Tema koju ćemo predstaviti ima svoju istoriju, teoriju i motivaciju, a uklapa se u primene sa velikim brojem visoko-dimenzionalnih podatka.

Nauka o podacima na ICM2018

- Plenarna predavanja i predavanja po pozivu
- Za razliku od sličnih zaokreta u prošlosti (Finansijska matematika, genetika), u kojima se postojeća teorija primenjuje u novim oblastima, ovde je slučaj da primena ide ispred teorije.
- U mnogim procedurama koje se koriste u nauci o podacima nedostaju objašnjenja- zadatak matematičara.
- Tema koju ćemo predstaviti ima svoju istoriju, teoriju i motivaciju, a uklapa se u primene sa velikim brojem visoko-dimenzionalnih podatka.

Nauka o podacima na ICM2018

- Plenarna predavanja i predavanja po pozivu
- Za razliku od sličnih zaokreta u prošlosti (Finansijska matematika, genetika), u kojima se postojeća teorija primenjuje u novim oblastima, ovde je slučaj da primena ide ispred teorije.
- U mnogim procedurama koje se koriste u nauci o podacima nedostaju objašnjenja- zadatak matematičara.
- Tema koju ćemo predstaviti ima svoju istoriju, teoriju i motivaciju, a uklapa se u primene sa velikim brojem visoko-dimenzionalnih podatka.

Nauka o podacima na ICM2018

- Plenarna predavanja i predavanja po pozivu
- Za razliku od sličnih zaokreta u prošlosti (Finansijska matematika, genetika), u kojima se postojeća teorija primenjuje u novim oblastima, ovde je slučaj da primena ide ispred teorije.
- U mnogim procedurama koje se koriste u nauci o podacima nedostaju objašnjenja- zadatak matematičara.
- Tema koju ćemo predstaviti ima svoju istoriju, teoriju i motivaciju, a uklapa se u primene sa velikim brojem visoko-dimenzionalnih podatka.

Funkcija dubine na $\mathbb{R}$

Za skup $S = \{x_1, , \ldots, x_n\}$ *(podaci) dubina tačke* $x \in \mathbb{R}$ *u odnosu na* S *je*

$$D(x) = \min\left(\#\{x_i \in S \,|\, x_i \le x\}, \#\{x_i \in S \,|\, x_i \ge x\}\right)$$

Tačke van intervala $(x_{(1)}, x_{(n)})$ *imaju* $D(x) = 0$.

Preko verovatnoće: Ako je X *slučajna promenljiva na skupu* S:

$$D(x) = \{\min\{P(X \le x), P(X \ge x)\}$$

Kad $x \to \pm\infty$, $D(x) \to 0$.

Uzoračka raspodela (ovaj termin se koristi u statistici) dodeljuje verovatnoću $1/n$ *svakoj tački u skupu* S, *računajući i ponavljanje iste tačke.*

$$D(x) = \frac{1}{n}\min\left(\#\{x_i \in S \,|\, x_i \le x\}, \#\{x_i \in S \,|\, x_i \ge x\}\right)$$

Funkcija dubine na $\mathbb{R}$

Za skup $S = \{x_1, , \ldots, x_n\}$ (podaci) dubina tačke $x \in \mathbb{R}$ u odnosu na S je

$$D(x) = \min\left(\#\{x_i \in S \,|\, x_i \le x\}, \#\{x_i \in S \,|\, x_i \ge x\}\right)$$

Tačke van intervala $(x_{(1)}, x_{(n)})$ imaju $D(x) = 0$.

Preko verovatnoće: Ako je X slučajna promenljiva na skupu S:

$$D(x) = \{\min\{P(X \le x), P(X \ge x)$$

Kad $x \to \pm\infty$, $D(x) \to 0$.

Uzoračka raspodela (ovaj termin se koristi u statistici) dodeljuje verovatnoću $1/n$ svakoj tački u skupu S, računajući i ponavljanje iste tačke.

$$D(x) = \frac{1}{n}\min\left(\#\{x_i \in S \,|\, x_i \le x\}, \#\{x_i \in S \,|\, x_i \ge x\}\right)$$

Funkcija dubine na $\mathbb{R}$

Za skup $S = \{x_1, , \ldots, x_n\}$ *(podaci) dubina tačke* $x \in \mathbb{R}$ *u odnosu na* S *je*

$$D(x) = \min\left(\#\{x_i \in S \,|\, x_i \le x\}, \#\{x_i \in S \,|\, x_i \ge x\}\right)$$

Tačke van intervala $(x_{(1)}, x_{(n)})$ *imaju* $D(x) = 0$.

Preko verovatnoće: Ako je X *slučajna promenljiva na skupu* S:

$$D(x) = \{\min\{P(X \le x), P(X \ge x)$$

Kad $x \to \pm\infty$, $D(x) \to 0$.

Uzoračka raspodela (ovaj termin se koristi u statistici) dodeljuje verovatnoću $1/n$ *svakoj tački u skupu* S, *računajući i ponavljanje iste tačke.*

$$D(x) = \frac{1}{n}\min\left(\#\{x_i \in S \,|\, x_i \le x\}, \#\{x_i \in S \,|\, x_i \ge x\}\right)$$

Dubina na $\mathbb{R}$-nastavak

U opštem slučaju, imamo verovatnosnu meru μ_X na $\mathbb{R}$ i dubinu

$$D(x; \mu_X) = \min\{\mu((-\infty, x]), \mu([x, +\infty))\}$$

Osobine funkcije dubine na $\mathbb{R}$:

- Afina invarijantnost: $D(ax + b, \mu_{aX+b}) = D(x, \mu_X)$, $a \neq 0$
- Funkcija D dostiže maksimum $\geq \frac{1}{2}$ u medijani (tačka ili kompaktni interval)
- $D(x; \mu_X) \to 0$ kad $x \to \pm\infty$

Za svako $\alpha \in [0, \frac{1}{2}]$ definišemo oblast dubine $S_\alpha = \{x \in \mathbb{R} | D(x) \geq \alpha)\}$.

- $S_\alpha = [K_\alpha, K_{1-\alpha}]$
- U dimenziji $d \geq 2$?

Dubina na $\mathbb{R}$-nastavak

U opštem slučaju, imamo verovatnosnu meru μ_X na $\mathbb{R}$ i dubinu

$$D(x;\mu_X) = \min\{\mu((-\infty, x]), \mu([x, +\infty))\}$$

Osobine funkcije dubine na $\mathbb{R}$:

- Afina invarijantnost: $D(ax+b, \mu_{aX+b}) = D(x, \mu_X)$, $a \neq 0$
- Funkcija D dostiže maksimum $\geq \frac{1}{2}$ u medijani (tačka ili kompaktni interval)
- $D(x;\mu_X) \to 0$ kad $x \to \pm\infty$

Za svako $\alpha \in [0, \frac{1}{2}]$ definišemo oblast dubine $S_\alpha = \{x \in \mathbb{R} | D(x) \geq \alpha)\}$.

- $S_\alpha = [K_\alpha, K_{1-\alpha}]$
- U dimenziji $d \geq 2$?

Dubina na $\mathbb{R}$-nastavak

U opštem slučaju, imamo verovatnosnu meru μ_X na $\mathbb{R}$ i dubinu

$$D(x; \mu_X) = \min\{\mu((-\infty, x]), \mu([x, +\infty))\}$$

Osobine funkcije dubine na $\mathbb{R}$:

- Afina invarijantnost: $D(ax + b, \mu_{aX+b}) = D(x, \mu_X)$, $a \neq 0$
- Funkcija D dostiže maksimum $\geq \frac{1}{2}$ u medijani (tačka ili kompaktni interval)
- $D(x; \mu_X) \to 0$ kad $x \to \pm\infty$

Za svako $\alpha \in [0, \frac{1}{2}]$ definišemo oblast dubine $S_\alpha = \{x \in \mathbb{R} | D(x) \geq \alpha)\}$.

- $S_\alpha = [K_\alpha, K_{1-\alpha}]$
- U dimenziji $d \geq 2$?

Dubina na $\mathbb{R}$-nastavak

U opštem slučaju, imamo verovatnosnu meru μ_X na $\mathbb{R}$ i dubinu

$$D(x; \mu_X) = \min\{\mu((-\infty, x]), \mu([x, +\infty))\}$$

Osobine funkcije dubine na $\mathbb{R}$:

- Afina invarijantnost: $D(ax + b, \mu_{aX+b}) = D(x, \mu_X)$, $a \neq 0$
- Funkcija D dostiže maksimum $\geq \frac{1}{2}$ u medijani (tačka ili kompaktni interval)
- $D(x; \mu_X) \to 0$ kad $x \to \pm\infty$

Za svako $\alpha \in [0, \frac{1}{2}]$ definišemo oblast dubine $S_\alpha = \{x \in \mathbb{R} | D(x) \geq \alpha)\}$.

- $S_\alpha = [K_\alpha, K_{1-\alpha}]$
- U dimenziji $d \geq 2$?

Dubina na $\mathbb{R}$-nastavak

U opštem slučaju, imamo verovatnosnu meru μ_X na $\mathbb{R}$ i dubinu

$$D(x; \mu_X) = \min\{\mu((-\infty, x]), \mu([x, +\infty))\}$$

Osobine funkcije dubine na $\mathbb{R}$:

- Afina invarijantnost: $D(ax + b, \mu_{aX+b}) = D(x, \mu_X)$, $a \neq 0$
- Funkcija D dostiže maksimum $\geq \frac{1}{2}$ u medijani (tačka ili kompaktni interval)
- $D(x; \mu_X) \to 0$ kad $x \to \pm\infty$

Za svako $\alpha \in [0, \frac{1}{2}]$ definišemo oblast dubine $S_\alpha = \{x \in \mathbb{R} | D(x) \geq \alpha)\}$.

- $S_\alpha = [K_\alpha, K_{1-\alpha}]$
- U dimenziji $d \geq 2$?

Motivacija: Parametri lokacije i njihove ocene

The standard and most common location parameter is mathematical expectation, $\mathrm{E}(X)$*, which is also called a mean value. Here* X *is in general a vector of dimension* d*. In combination with variance, it is completely adequate within Normal (Gaussian) framework and makes mathematics easier.*
Let $X_1, \dots, X_n$ *be a sample of size* n *from* X*, and let* $T^{(n)}(X_1, \dots, X_n)$ *be an estimator for* $\mathrm{E}\,X$*. It sounds logical that any estimator of location should be*

(1) **Permutation invariant:** $T^{(n)}(X_{\pi(1)}, \dots X_{\pi(n)}) = T^{(n)}(X_1, \dots, X_n)$

(2) **Translation equivariant:**
$T^{(n)}(X_1 + b, \dots, X_n + b) = T^{(n)}(X_1, \dots, X_n) + b$

and it is desirable that T *is*

(3) **Affine equivariant:** $T(A \cdot X + b) = A \cdot T(X) + b$ *for any non-singular* $d \times d$ *matrix* A *and an arbitrary* $d \times 1$ *vector* b*.*

Motivacija: Parametri lokacije i njihove ocene

The standard and most common location parameter is mathematical expectation, $\mathrm{E}(X)$*, which is also called a mean value. Here* X *is in general a vector of dimension* d*. In combination with variance, it is completely adequate within Normal (Gaussian) framework and makes mathematics easier.*

Let $X_1, \ldots, X_n$ *be a sample of size* n *from* X*, and let* $T^{(n)}(X_1, \ldots, X_n)$ *be an estimator for* $\mathrm{E}\,X$*. It sounds logical that any estimator of location should be*

(1) **Permutation invariant:** $T^{(n)}(X_{\pi(1)}, \ldots X_{\pi(n)}) = T^{(n)}(X_1, \ldots, X_n)$

(2) **Translation equivariant:** $T^{(n)}(X_1 + b, \ldots, X_n + b) = T^{(n)}(X_1, \ldots, X_n) + b$

and it is desirable that T *is*

(3) **Affine equivariant:** $T(A \cdot X + b) = A \cdot T(X) + b$ *for any non-singular* $d \times d$ *matrix* A *and an arbitrary* $d \times 1$ *vector* b*.*

Motivacija: Parametri lokacije i njihove ocene

The standard and most common location parameter is mathematical expectation, $\mathrm{E}(X)$, which is also called a mean value. Here X is in general a vector of dimension d. In combination with variance, it is completely adequate within Normal (Gaussian) framework and makes mathematics easier.
Let $X_1, \ldots, X_n$ be a sample of size n from X, and let $T^{(n)}(X_1, \ldots, X_n)$ be an estimator for $\mathrm{E}\,X$. It sounds logical that any estimator of location should be

(1) **Permutation invariant:** $T^{(n)}(X_{\pi(1)}, \ldots X_{\pi(n)}) = T^{(n)}(X_1, \ldots, X_n)$

(2) **Translation equivariant:**
$T^{(n)}(X_1 + b, \ldots, X_n + b) = T^{(n)}(X_1, \ldots, X_n) + b$

and it is desirable that T is

(3) **Affine equivariant:** $T(A \cdot X + b) = A \cdot T(X) + b$ *for any non-singular $d \times d$ matrix A and an arbitrary $d \times 1$ vector b.*

Motivacija: Parametri lokacije i njihove ocene

The standard and most common location parameter is mathematical expectation, $\mathrm{E}(X)$*, which is also called a mean value. Here* X *is in general a vector of dimension* d*. In combination with variance, it is completely adequate within Normal (Gaussian) framework and makes mathematics easier.*
Let $X_1, \ldots, X_n$ *be a sample of size* n *from* X*, and let* $T^{(n)}(X_1, \ldots, X_n)$ *be an estimator for* $\mathrm{E}\,X$*. It sounds logical that any estimator of location should be*

(1) **Permutation invariant:** $T^{(n)}(X_{\pi(1)}, \ldots X_{\pi(n)}) = T^{(n)}(X_1, \ldots, X_n)$

(2) **Translation equivariant:**
$T^{(n)}(X_1 + b, \ldots, X_n + b) = T^{(n)}(X_1, \ldots, X_n) + b$

and it is desirable that T *is*

(3) **Affine equivariant:** $T(A \cdot X + b) = A \cdot T(X) + b$ *for any non-singular* $d \times d$ *matrix* A *and an arbitrary* $d \times 1$ *vector* b*.*

Motivacija: Robusnost i tačka preloma

Suppose that we have a sample which is contaminated with some "bad data"(outliers-off the model). The estimator should not change much with a small amount of contamination. The resistance of the estimator to small changes in the input data is called **robustness**. *The measure of robustness is so called* **breakdown point-tačka preloma**. *Suppose that we have a dataset X of size n and one bad dataset Y of size m. If by appropriate choice of Y the difference $T(X \cup Y) - T(X)$ can be made as large as desired, we say that T breaks down at X under contamination of size m. Let*

$$m^* = \min\{m \mid sup_{\#Y=m}|T(X \cup Y) - T(X)| = \infty\}$$

Breakdown point is calculated as follows (Donoho 1982)

$$\varepsilon^* = \frac{m^*}{n + m^*}$$

Examples in dimension $d = 1$:

- *Arithmetic mean (as estimator for mathematical expectation)* $\frac{1}{n+1}$ *(isto i u $d \geq 2$)*
- *Sample median (as estimator for median of a distribution)* $\sim \frac{1}{2}$
- *α-trimmed mean (as estimator for mathematical expectation)* $\sim \frac{\alpha}{1+\alpha}$

How to extend a notion of the median to dimensions greater than 1? To do this we have to extend a notion of deepness to higher dimensions.

Examples in dimension $d = 1$:

- *Arithmetic mean (as estimator for mathematical expectation)* $\frac{1}{n+1}$ *(isto i u $d \geq 2$)*
- *Sample median (as estimator for median of a distribution)* $\sim \frac{1}{2}$
- *α-trimmed mean (as estimator for mathematical expectation)* $\sim \frac{\alpha}{1+\alpha}$

How to extend a notion of the median to dimensions greater than 1? To do this we have to extend a notion of deepness to higher dimensions.

Examples in dimension $d = 1$:

- *Arithmetic mean (as estimator for mathematical expectation)* $\frac{1}{n+1}$ *(isto i u $d \geq 2$)*
- *Sample median (as estimator for median of a distribution)* $\sim \frac{1}{2}$
- *α-trimmed mean (as estimator for mathematical expectation)* $\sim \frac{\alpha}{1+\alpha}$

How to extend a notion of the median to dimensions greater than 1? To do this we have to extend a notion of deepness to higher dimensions.

Examples in dimension $d = 1$:

- *Arithmetic mean (as estimator for mathematical expectation)* $\frac{1}{n+1}$ *(isto i u $d \geq 2$)*
- *Sample median (as estimator for median of a distribution)* $\sim \frac{1}{2}$
- α*-trimmed mean (as estimator for mathematical expectation)* $\sim \frac{\alpha}{1+\alpha}$

How to extend a notion of the median to dimensions greater than 1? To do this we have to extend a notion of deepness to higher dimensions.

Examples in dimension $d = 1$:

- *Arithmetic mean (as estimator for mathematical expectation)* $\frac{1}{n+1}$ *(isto i u $d \geq 2$)*
- *Sample median (as estimator for median of a distribution)* $\sim \frac{1}{2}$
- α*-trimmed mean (as estimator for mathematical expectation)* $\sim \frac{\alpha}{1+\alpha}$

How to extend a notion of the median to dimensions greater than 1? To do this we have to extend a notion of deepness to higher dimensions.

Funkcije dubine na $\mathbb{R}^d, d \geq 1$

In dimensions $d \geq 2$, there is no unique natural approach to defining the depth. It is desirable that the depth $D(\mathbf{x})$ retains some properties from $d = 1$, like

- *Depth does not depend of coordinate system (affine invariance)*
- *Attains maximum ($\geq \frac{1}{2}$?) at some point*
- *$D(\mathbf{x}) \to 0$ as $\|\mathbf{x}\| \to \infty$.*

There are many different generalizations to dimensions ≥ 2. Only a few of them satisfy all 3 conditions. In this talk we focus to the first known depth function- Tukey's depth.

Funkcije dubine na $\mathbb{R}^d, d \geq 1$

In dimensions $d \geq 2$, there is no unique natural approach to defining the depth. It is desirable that the depth $D(\mathbf{x})$ retains some properties from $d = 1$, like

- *Depth does not depend of coordinate system (affine invariance)*
- *Attains maximum $(\geq \frac{1}{2}\ ?)$ at some point*
- *$D(\mathbf{x}) \to 0$ as $\|\mathbf{x}\| \to \infty$.*

There are many different generalizations to dimensions ≥ 2. Only a few of them satisfy all 3 conditions. In this talk we focus to the first known depth function- Tukey's depth.

Funkcije dubine na $\mathbb{R}^d, d \geq 1$

In dimensions $d \geq 2$, there is no unique natural approach to defining the depth. It is desirable that the depth $D(\mathbf{x})$ retains some properties from $d = 1$, like

- *Depth does not depend of coordinate system (affine invariance)*
- *Attains maximum $(\geq \frac{1}{2}\ ?)$ at some point*
- *$D(\mathbf{x}) \to 0$ as $\|\mathbf{x}\| \to \infty$.*

There are many different generalizations to dimensions ≥ 2. Only a few of them satisfy all 3 conditions. In this talk we focus to the first known depth function- Tukey's depth.

Funkcije dubine na $\mathbb{R}^d, d \geq 1$

In dimensions $d \geq 2$, there is no unique natural approach to defining the depth. It is desirable that the depth $D(\mathbf{x})$ retains some properties from $d = 1$, like

- *Depth does not depend of coordinate system (affine invariance)*
- *Attains maximum $(\geq \frac{1}{2}\ ?)$ at some point*
- $D(\mathbf{x}) \to 0$ *as* $\|\mathbf{x}\| \to \infty$.

There are many different generalizations to dimensions ≥ 2. Only a few of them satisfy all 3 conditions. In this talk we focus to the first known depth function- Tukey's depth.

Funkcije dubine na $\mathbb{R}^d, d \geq 1$

In dimensions $d \geq 2$, there is no unique natural approach to defining the depth. It is desirable that the depth $D(\mathbf{x})$ retains some properties from $d = 1$, like

- *Depth does not depend of coordinate system (affine invariance)*
- *Attains maximum $(\geq \frac{1}{2}\ ?)$ at some point*
- $D(\mathbf{x}) \to 0$ *as* $\|\mathbf{x}\| \to \infty$.

There are many different generalizations to dimensions ≥ 2. Only a few of them satisfy all 3 conditions. In this talk we focus to the first known depth function- Tukey's depth.

Tukey-eva (poluprostorna) dubina

Tukey's depth, or halfspace depth is defined as follows.

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 1$. For a given probability measure μ on $\mathbb{R}^d$, define

$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \mid \mathbf{x} \in H \in \mathcal{H}\}.$$

- The set of deepest points is usually called Tukey median. For $d \geq 2$ the maximal depth is between $1/(d+1)$ and $1/2$ (the upper bound is valid for finite sample in general position).
- The depth region or α-level set S_α is defined as

$$S_\alpha = \{\mathbf{x} \in \mathbb{R}^d \mid D(\mathbf{x}, \mu) \geq \alpha\}.$$

Examples in $\mathbb{R}^2$:

- *Triangle in $\mathbb{R}^2$;*
- *Uniform distribution in a ring*

Tukey-eva (poluprostorna) dubina

Tukey's depth, or halfspace depth is defined as follows.

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 1$. For a given probability measure μ on $\mathbb{R}^d$, define

$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \,|\, \mathbf{x} \in H \in \mathcal{H}\}.$$

- The set of deepest points is usually called Tukey median. For $d \geq 2$ the maximal depth is between $1/(d+1)$ and $1/2$ (the upper bound is valid for finite sample in general position).
- The depth region or α-level set S_α is defined as

$$S_\alpha = \{\mathbf{x} \in \mathbb{R}^d \mid D(\mathbf{x}, \mu) \geq \alpha\}.$$

Examples in $\mathbb{R}^2$:

- *Triangle in $\mathbb{R}^2$;*
- *Uniform distribution in a ring*

Tukey-eva (poluprostorna) dubina

Tukey's depth, or halfspace depth is defined as follows.

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 1$. For a given probability measure μ on $\mathbb{R}^d$, define

$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \,|\, \mathbf{x} \in H \in \mathcal{H}\}.$$

- The set of deepest points is usually called Tukey median. For $d \geq 2$ the maximal depth is between $1/(d+1)$ and $1/2$ (the upper bound is valid for finite sample in general position).
- The depth region or α-level set S_α is defined as

$$S_\alpha = \{\mathbf{x} \in \mathbb{R}^d \mid D(\mathbf{x}, \mu) \geq \alpha\}.$$

Examples in $\mathbb{R}^2$:

- *Triangle in $\mathbb{R}^2$;*
- *Uniform distribution in a ring*

Tukey-eva (poluprostorna) dubina

Tukey's depth, or halfspace depth is defined as follows.

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 1$. For a given probability measure μ on $\mathbb{R}^d$, define

$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \,|\, \mathbf{x} \in H \in \mathcal{H}\}.$$

- The set of deepest points is usually called Tukey median. For $d \geq 2$ the maximal depth is between $1/(d+1)$ and $1/2$ (the upper bound is valid for finite sample in general position).
- The depth region or α-level set S_α is defined as

$$S_\alpha = \{\mathbf{x} \in \mathbb{R}^d \mid D(\mathbf{x}, \mu) \geq \alpha\}.$$

Examples in $\mathbb{R}^2$:

- *Triangle in $\mathbb{R}^2$;*
- *Uniform distribution in a ring*

Tukey-eva (poluprostorna) dubina

Tukey's depth, or halfspace depth is defined as follows.

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 1$. For a given probability measure μ on $\mathbb{R}^d$, define

$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \,|\, \mathbf{x} \in H \in \mathcal{H}\}.$$

- The set of deepest points is usually called Tukey median. For $d \geq 2$ the maximal depth is between $1/(d+1)$ and $1/2$ (the upper bound is valid for finite sample in general position).
- The depth region or α-level set S_α is defined as

$$S_\alpha = \{\mathbf{x} \in \mathbb{R}^d \mid D(\mathbf{x}, \mu) \geq \alpha\}.$$

Examples in $\mathbb{R}^2$:

- *Triangle in $\mathbb{R}^2$;*
- *Uniform distribution in a ring*

Tukey-eva (poluprostorna) dubina

Tukey's depth, or halfspace depth is defined as follows.

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 1$. For a given probability measure μ on $\mathbb{R}^d$, define

$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \,|\, \mathbf{x} \in H \in \mathcal{H}\}.$$

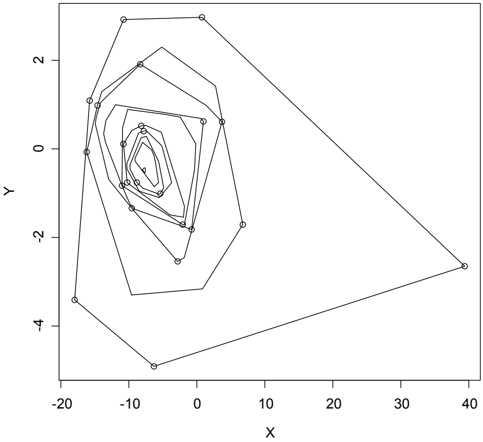
- The set of deepest points is usually called Tukey median. For $d \geq 2$ the maximal depth is between $1/(d+1)$ and $1/2$ (the upper bound is valid for finite sample in general position).
- The depth region or α-level set S_α is defined as

$$S_\alpha = \{\mathbf{x} \in \mathbb{R}^d \mid D(\mathbf{x}, \mu) \geq \alpha\}.$$

Examples in $\mathbb{R}^2$:

- *Triangle in $\mathbb{R}^2$;*
- *Uniform distribution in a ring*

Primer: tipičan skup podataka i oblasti dubine, $d = 2$



Iako govorimo o visoko-dimenzionalnim podatacima, svi primeri su u $d = 2$, zbog teškoća u vizualizaciji kod $d > 2$. (Tukey!)

Tukey's half space depth- 2

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 2$. For a given probability measure μ on $\mathbb{R}^d$, the Tukey depth of a point $\mathbf{x}$ is defined as

$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \,|\, \mathbf{x} \in H \in \mathcal{H}\}.$$

- In the case of sample distribution, it is shown in Donoho (1982) that the depth can be expressed via one - dimensional projections of sample points (here X is the sample set of size n)

$$D(x, X) = \frac{1}{n} \inf_{\|u\|=1} \#\{X_i : \langle u, X_i \rangle \leq \langle u, x \rangle\}$$

This was being used as a starting point in all known algorithms other than ours.

Tukey's half space depth- 2

- Let $\mathcal{H}$ be a collection of open half spaces in $\mathbb{R}^d$, $d \geq 2$. For a given probability measure μ on $\mathbb{R}^d$, the Tukey depth of a point $\mathbf{x}$ is defined as
$$D(\mathbf{x}, \mu) = \inf\{\mu(H) \mid \mathbf{x} \in H \in \mathcal{H}\}.$$
- In the case of sample distribution, it is shown in Donoho (1982) that the depth can be expressed via one - dimensional projections of sample points (here X is the sample set of size n)
$$D(x, X) = \frac{1}{n} \inf_{\|u\|=1} \#\{X_i : \langle u, X_i \rangle \leq \langle u, x \rangle\}$$

This was being used as a starting point in all known algorithms other than ours.

Tukey's depth - history and state of art

- This is the oldest and most known depth function. The idea traces back to Tukey (1975) and Tukey (1977), and it was first formalized in Donoho's Ph.D thesis (1982), in a technical report of Gasko and Donoho (1987) and in the AS paper Donoho and Gasko (1992).
- The lack of efficient programs for calculations in really high dimensions kept this area mostly at the level of theory. Rousseeuw and Ruts (1998) pioneered with an exact algorithm HALFMED for Tukey median in two dimensions. Struyf and Rousseeuw (2000) DEEPLOC algorithm calculates approximate Tukey median in higher dimensions, with certain limitation that will be discussed later. Random algorithm by Chan (2004) is exact, but not implemented due to high complexity. There is a recent work of Mozharovskyi (2014, PhD thesis) and Dyckerhoff and Mozharovskyi (2016) on an exact algorithm, whose performances in really high dimensions have not been investigated so far.
- Aside of Tukey's depth there are many other depth notions, see the survey in Small (1990) and classification in Zuo and Serfling (2000).

Tukey's depth - history and state of art

- This is the oldest and most known depth function. The idea traces back to Tukey (1975) and Tukey (1977), and it was first formalized in Donoho's Ph.D thesis (1982), in a technical report of Gasko and Donoho (1987) and in the AS paper Donoho and Gasko (1992).
- The lack of efficient programs for calculations in really high dimensions kept this area mostly at the level of theory. Rousseeuw and Ruts (1998) pioneered with an exact algorithm HALFMED for Tukey median in two dimensions. Struyf and Rousseeuw (2000) DEEPLOC algorithm calculates approximate Tukey median in higher dimensions, with certain limitation that will be discussed later. Random algorithm by Chan (2004) is exact, but not implemented due to high complexity. There is a recent work of Mozharovskyi (2014, PhD thesis) and Dyckerhoff and Mozharovskyi (2016) on an exact algorithm, whose performances in really high dimensions have not been investigated so far.
- Aside of Tukey's depth there are many other depth notions, see the survey in Small (1990) and classification in Zuo and Serfling (2000).

Tukey's depth - history and state of art

- This is the oldest and most known depth function. The idea traces back to Tukey (1975) and Tukey (1977), and it was first formalized in Donoho's Ph.D thesis (1982), in a technical report of Gasko and Donoho (1987) and in the AS paper Donoho and Gasko (1992).
- The lack of efficient programs for calculations in really high dimensions kept this area mostly at the level of theory. Rousseeuw and Ruts (1998) pioneered with an exact algorithm HALFMED for Tukey median in two dimensions. Struyf and Rousseeuw (2000) DEEPLOC algorithm calculates approximate Tukey median in higher dimensions, with certain limitation that will be discussed later. Random algorithm by Chan (2004) is exact, but not implemented due to high complexity. There is a recent work of Mozharovskyi (2014, PhD thesis) and Dyckerhoff and Mozharovskyi (2016) on an exact algorithm, whose performances in really high dimensions have not been investigated so far.
- Aside of Tukey's depth there are many other depth notions, see the survey in Small (1990) and classification in Zuo and Serfling (2000).

Naučni interes i primene

- Istraživanja u oblasti funkcija dubina imaju primene u statistici - robusne ocene parametara lokacije, klasifikacija, kategorizacija, detetekcija podataka koji ne odgovaraju modelu (outliers), testiranje hipoteza itd.
- U računarskoj geometriji se proučavaju funkcije dubine u kontekstu geometrije i algoritama, sa drugačijom terminologijom.
- Funkcionalna analiza podataka (Horváth and Kokoszka, 2011), u razvoju, Serfling and Wijesuriya (2016).
- Tema može biti interesantna matematičarima raznih profila, inženjerima i fizičarima.

Naučni interes i primene

- Istraživanja u oblasti funkcija dubina imaju primene u statistici - robusne ocene parametara lokacije, klasifikacija, kategorizacija, detetekcija podataka koji ne odgovaraju modelu (outliers), testiranje hipoteza itd.
- U računarskoj geometriji se proučavaju funkcije dubine u kontekstu geometrije i algoritama, sa drugačijom terminologijom.
- Funkcionalna analiza podataka (Horváth and Kokoszka, 2011), u razvoju, Serfling and Wijesuriya (2016).
- Tema može biti interesantna matematičarima raznih profila, inženjerima i fizičarima.

Naučni interes i primene

- Istraživanja u oblasti funkcija dubina imaju primene u statistici - robusne ocene parametara lokacije, klasifikacija, kategorizacija, detetekcija podataka koji ne odgovaraju modelu (outliers), testiranje hipoteza itd.
- U računarskoj geometriji se proučavaju funkcije dubine u kontekstu geometrije i algoritama, sa drugačijom terminologijom.
- Funkcionalna analiza podataka (Horváth and Kokoszka, 2011), u razvoju, Serfling and Wijesuriya (2016).
- Tema može biti interesantna matematičarima raznih profila, inženjerima i fizičarima.

Naučni interes i primene

- Istraživanja u oblasti funkcija dubina imaju primene u statistici - robusne ocene parametara lokacije, klasifikacija, kategorizacija, detetekcija podataka koji ne odgovaraju modelu (outliers), testiranje hipoteza itd.
- U računarskoj geometriji se proučavaju funkcije dubine u kontekstu geometrije i algoritama, sa drugačijom terminologijom.
- Funkcionalna analiza podataka (Horváth and Kokoszka, 2011), u razvoju, Serfling and Wijesuriya (2016).
- Tema može biti interesantna matematičarima raznih profila, inženjerima i fizičarima.

Oblasti dubine u $\mathbb{R}^d$

- The depth region or α-level set S_α is defined as

$$S_\alpha = \{x \in \mathbb{R}^d \mid D(x,\mu) \geq \alpha\}.$$

- It is well known that level sets can be represented as intersection of half spaces of probability greater than $1-\alpha$ (Donoho (1982), more general setup in Zuo and Serfling (2000) and Merkle (2010)):

$$S_\alpha = \bigcap_{H \in \mathcal{H}:\ \mu(\bar{H}) > 1-\alpha} \bar{H},$$

- For a data set with n points, we use the counting measure and so $\alpha = \frac{k}{n}, k \in \{0, 1, \ldots, n\}$.

Oblasti dubine u $\mathbb{R}^d$

- The depth region or α-level set S_α is defined as

$$S_\alpha = \{x \in \mathbb{R}^d \mid D(x, \mu) \geq \alpha\}.$$

- It is well known that level sets can be represented as intersection of half spaces of probability greater than $1 - \alpha$ (Donoho (1982), more general setup in Zuo and Serfling (2000) and Merkle (2010)):

$$S_\alpha = \bigcap_{H \in \mathcal{H}:\ \mu(\bar{H}) > 1-\alpha} \bar{H},$$

- For a data set with n points, we use the counting measure and so $\alpha = \frac{k}{n}, k \in \{0, 1, \ldots, n\}$.

Oblasti dubine u $\mathbb{R}^d$

- The depth region or α-level set S_α is defined as

$$S_\alpha = \{x \in \mathbb{R}^d \mid D(x, \mu) \geq \alpha\}.$$

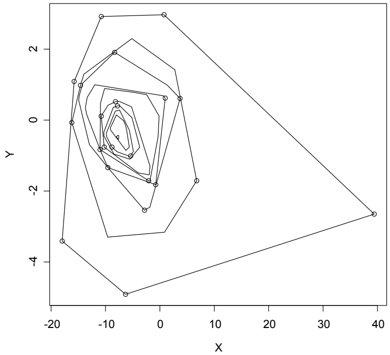
- It is well known that level sets can be represented as intersection of half spaces of probability greater than $1 - \alpha$ (Donoho (1982), more general setup in Zuo and Serfling (2000) and Merkle (2010)):

$$S_\alpha = \bigcap_{H \in \mathcal{H}:\ \mu(\bar{H}) > 1-\alpha} \bar{H},$$

- For a data set with n points, we use the counting measure and so $\alpha = \frac{k}{n}, k \in \{0, 1, \ldots, n\}$.

Example: NY Crime data

The true depth regions for a finite data set are convex and compact sets - polyhedra.



The benchmark data set NY CRIME with depth regions plotted by ISODEPTH.

Izračunavanje dubine

Here we consider a sample of n points. The depth of any point can take only $n+1$ possible values, and the number of different level sets can not be greater than $n+1$. Sets $S_{k/n}$ are nested and decreasing with k.

- The Tukey median can be obtained as the the smallest non-empty level set.
- The depth of a point x equals to k/n if and only if $x \in S_{\frac{k}{n}}$ and $x \notin S_{\frac{k+1}{n}}$
- So, having the level sets, we can calculate both median and the depth of any point. Exact calculation of level sets (depth contours) has complexity $\sim n^d$. We propose a procedure that uses an approximation to S_α in the form of a discrete set of points.

Izračunavanje dubine

Here we consider a sample of n points. The depth of any point can take only $n+1$ possible values, and the number of different level sets can not be greater than $n+1$. Sets $S_{k/n}$ are nested and decreasing with k.

- The Tukey median can be obtained as the the smallest non-empty level set.
- The depth of a point x equals to k/n if and only if $x \in S_{\frac{k}{n}}$ and $x \notin S_{\frac{k+1}{n}}$
- So, having the level sets, we can calculate both median and the depth of any point. Exact calculation of level sets (depth contours) has complexity $\sim n^d$. We propose a procedure that uses an approximation to S_α in the form of a discrete set of points.

Izračunavanje dubine

Here we consider a sample of n points. The depth of any point can take only $n+1$ possible values, and the number of different level sets can not be greater than $n+1$. Sets $S_{k/n}$ are nested and decreasing with k.

- The Tukey median can be obtained as the the smallest non-empty level set.
- The depth of a point x equals to k/n if and only if $x \in S_{\frac{k}{n}}$ and $x \notin S_{\frac{k+1}{n}}$
- So, having the level sets, we can calculate both median and the depth of any point. Exact calculation of level sets (depth contours) has complexity $\sim n^d$. We propose a procedure that uses an approximation to S_α in the form of a discrete set of points.

Oblasti dubine preko preseka kugli u $\mathbb{R}^d$

- Sets S_α can be found as well by intersection of balls (Merkle 2010):

$$S_\alpha = \{x : D(x) \ge \alpha\} = \bigcap_{B:\mu(B)>1-\alpha} B.$$

This is obvious by the representation of a ball as the intersection of tangent half spaces.

- For a sample of size n, with $\alpha = \frac{k}{n}$, the balls have to contain $n-k+1$ sample points.
- First approximation-$\hat{S}_\alpha$: Chose N points to be centers of balls, and define balls $B_1, \dots B_N$ with required number of sample points. Then define $\hat{S}_\alpha$ as the intersection of these N balls.

Oblasti dubine preko preseka kugli u $\mathbb{R}^d$

- Sets S_α can be found as well by intersection of balls (Merkle 2010):

$$S_\alpha = \{x : D(x) \geq \alpha\} = \bigcap_{B:\mu(B)>1-\alpha} B.$$

This is obvious by the representation of a ball as the intersection of tangent half spaces.

- For a sample of size n, with $\alpha = \frac{k}{n}$, the balls have to contain $n - k + 1$ sample points.
- First approximation-$\hat{S}_\alpha$: Chose N points to be centers of balls, and define balls $B_1, \dots B_N$ with required number of sample points. Then define $\hat{S}_\alpha$ as the intersection of these N balls.

Oblasti dubine preko preseka kugli u $\mathbb{R}^d$

- Sets S_α can be found as well by intersection of balls (Merkle 2010):

$$S_\alpha = \{x : D(x) \geq \alpha\} = \bigcap_{B:\mu(B)>1-\alpha} B.$$

  This is obvious by the representation of a ball as the intersection of tangent half spaces.
- For a sample of size n, with $\alpha = \frac{k}{n}$, the balls have to contain $n - k + 1$ sample points.
- First approximation-$\hat{S}_\alpha$: Chose N points to be centers of balls, and define balls $B_1, \dots B_N$ with required number of sample points. Then define $\hat{S}_\alpha$ as the intersection of these N balls.

Second (discrete) approximation

- Second approximation-$\hat{\hat{S}}$: Calculations of ball intersection is an NP-problem. On the other hand, we note that it is easy to determine whether or not any given points belongs to ball intersection $\hat{S}_\alpha$. So we choose M random points ("artificial points") in a convex domain that contains all sample points. The discrete set $\hat{\hat{S}}$ of artificial points x such that $x \in \hat{S}_\alpha$ is the final approximation that we use instead of true S_α.
- This novel idea leads to algorithms for median and for depth of a given point with complexity which is linear in d.
- In the sequel we will shortly present our algorithms and its behavior in some benchmark datasets, comparison with other approximate algorithms and analysis of errors.

Second (discrete) approximation

- Second approximation-$\hat{\hat{S}}$: Calculations of ball intersection is an NP-problem. On the other hand, we note that it is easy to determine whether or not any given points belongs to ball intersection $\hat{S}_\alpha$. So we choose M random points ("artificial points") in a convex domain that contains all sample points. The discrete set $\hat{\hat{S}}$ of artificial points x such that $x \in \hat{S}_\alpha$ is the final approximation that we use instead of true S_α.
- This novel idea leads to algorithms for median and for depth of a given point with complexity which is linear in d.
- In the sequel we will shortly present our algorithms and its behavior in some benchmark datasets, comparison with other approximate algorithms and analysis of errors.

Second (discrete) approximation

- Second approximation-$\hat{\hat{S}}$: Calculations of ball intersection is an NP-problem. On the other hand, we note that it is easy to determine whether or not any given points belongs to ball intersection $\hat{S}_\alpha$. So we choose M random points ("artificial points") in a convex domain that contains all sample points. The discrete set $\hat{\hat{S}}$ of artificial points x such that $x \in \hat{S}_\alpha$ is the final approximation that we use instead of true S_α.
- This novel idea leads to algorithms for median and for depth of a given point with complexity which is linear in d.
- In the sequel we will shortly present our algorithms and its behavior in some benchmark datasets, comparison with other approximate algorithms and analysis of errors.

Konvergencija $\hat{\hat{S}}$ ka S?

Za fiksirano α, pod kojim odnosom između broja kugli (N) i broja dodatnih tačaka (M) važi da je

$$\lim_{M,N \to +\infty} d(\hat{\hat{S}}_{M,N}, S) = 0,$$

gde je d Hausdorff-ova metrika?

Jensenova nejednakost za Tukey-eve medijane

- **Teorema (Merkle, 2010).** *Neka je* $\mathbf{X}$ *slučajni vektor u* $\mathbb{R}^d$*, i oblast dubine* $S_\alpha(\mathbf{X})$ *neprazan skup. Neka je* f *[konveksna] realna funkcija na* $\mathbb{R}^d$ *i* $Q_{1-\alpha}$ *najveći kvantil reda* $1-\alpha$ *za* $f(\mathbf{X})$*. Tada za svaku tačku* $\mathbf{x} \in S_\alpha$ *važi da je*

$$(*) \quad f(\mathbf{x}) \le Q_{1-\alpha}(f(\mathbf{X}))$$

- *(zašto se ovo zove Jensenova nejednakost?)*

Jednakost u $(*)$ *nastaje za* $f(\mathbf{x}) = 1 - D(\mathbf{x})$ *Nejednakost u* $(*)$ *znači da je* S_α *podskup skupa* $T_\alpha = \{\mathbf{x} \,|\, f(\mathbf{x}) \le c\}$, $c = Q_{1-\alpha}(f(X))$ *-jednostavno za izracunavanje.*

Problem: *Za dato* $\mathbf{X}$*, ispitati mogućnost da se preko adekvatne funkcije* f *dobije aproksimativna karakterizacija nivoa* S_α*.*

Jensenova nejednakost za Tukey-eve medijane

- **Teorema (Merkle, 2010).** *Neka je* $\mathbf{X}$ *slučajni vektor u* $\mathbb{R}^d$, *i oblast dubine* $S_\alpha(\mathbf{X})$ *neprazan skup. Neka je* f *[konveksna] realna funkcija na* $\mathbb{R}^d$ *i* $Q_{1-\alpha}$ *najveći kvantil reda* $1-\alpha$ *za* $f(\mathbf{X})$. *Tada za svaku tačku* $\mathbf{x} \in S_\alpha$ *važi da je*

$$(*) \quad f(\mathbf{x}) \leq Q_{1-\alpha}(f(\mathbf{X}))$$

- *(zašto se ovo zove Jensenova nejednakost?)*

Jednakost u $(*)$ *nastaje za* $f(\mathbf{x}) = 1 - D(\mathbf{x})$ *Nejednakost u* $(*)$ *znači da je* S_α *podskup skupa* $T_\alpha = \{\mathbf{x} \,|\, f(\mathbf{x}) \leq c\}$, $c = Q_{1-\alpha}(f(X))$ *-jednostavno za izracunavanje.*

Problem: *Za dato* $\mathbf{X}$, *ispitati mogućnost da se preko adekvatne funkcije* f *dobije aproksimativna karakterizacija nivoa* S_α.

Jensenova nejednakost za Tukey-eve medijane

- **Teorema (Merkle, 2010).** *Neka je* $\mathbf{X}$ *slučajni vektor u* $\mathbb{R}^d$*, i oblast dubine* $S_\alpha(\mathbf{X})$ *neprazan skup. Neka je* f *[konveksna] realna funkcija na* $\mathbb{R}^d$ *i* $Q_{1-\alpha}$ *najveći kvantil reda* $1-\alpha$ *za* $f(\mathbf{X})$*. Tada za svaku tačku* $\mathbf{x} \in S_\alpha$ *važi da je*

$$(*) \quad f(\mathbf{x}) \leq Q_{1-\alpha}(f(\mathbf{X}))$$

- *(zašto se ovo zove Jensenova nejednakost?)*

Jednakost u $(*)$ *nastaje za* $f(\mathbf{x}) = 1 - D(\mathbf{x})$ *Nejednakost u* $(*)$ *znači da je* S_α *podskup skupa* $T_\alpha = \{\mathbf{x} \,|\, f(\mathbf{x}) \leq c\}$, $c = Q_{1-\alpha}(f(X))$ *-jednostavno za izracunavanje.*

Problem: *Za dato* $\mathbf{X}$*, ispitati mogućnost da se preko adekvatne funkcije* f *dobije aproksimativna karakterizacija nivoa* S_α*.*

Jensenova nejednakost za Tukey-eve medijane

- **Teorema (Merkle, 2010).** *Neka je* $\mathbf{X}$ *slučajni vektor u* $\mathbb{R}^d$*, i oblast dubine* $S_\alpha(\mathbf{X})$ *neprazan skup. Neka je* f *[konveksna] realna funkcija na* $\mathbb{R}^d$ *i* $Q_{1-\alpha}$ *najveći kvantil reda* $1-\alpha$ *za* $f(\mathbf{X})$*. Tada za svaku tačku* $\mathbf{x} \in S_\alpha$ *važi da je*

$$(*) \quad f(\mathbf{x}) \leq Q_{1-\alpha}(f(\mathbf{X}))$$

- *(zašto se ovo zove Jensenova nejednakost?)*

Jednakost u $(*)$ *nastaje za* $f(\mathbf{x}) = 1 - D(\mathbf{x})$ *Nejednakost u* $(*)$ *znači da je* S_α *podskup skupa* $T_\alpha = \{\mathbf{x} \,|\, f(\mathbf{x}) \leq c\}$, $c = Q_{1-\alpha}(f(X))$ *-jednostavno za izracunavanje.*
Problem: *Za dato* $\mathbf{X}$*, ispitati mogućnost da se preko adekvatne funkcije* f *dobije aproksimativna karakterizacija nivoa* S_α*.*

Funkcije dubine preko parcijalnog uređenja

Neka je $\preceq$ relacija pacijalnog uređenja na $\bar{\mathbb{R}}^n$. Definišemo

$$[\mathbf{a}, \mathbf{b}] = \{\mathbf{x} \in \bar{\mathbb{R}}^d \,|\, \mathbf{a} \preceq x \preceq b\}$$

Primer: Konveksnim konusom K sa vrhom u nuli, definiše se parcijalno uređenje

$$\mathbf{x} \preceq \mathbf{y} \iff \mathbf{y} - \mathbf{x} \in K$$

$$[\mathbf{a}, \mathbf{b}] = (\mathbf{a} + K) \cap (\mathbf{b} - K),$$

Preko ovih intevala, dobijamo medijanu koja ima dubinu $\geq 1/2$, ali nije afino invarijantna.

- *Da li se u stvarnim problemima može preko transformacija koordinatnog sistema (data driven coordinate system) dobiti dubina koja bi bila prihvatljiva aproksimacija afinoj invarijantnosti?*
- *Da li je ovo jedino parcijalno uređenje koje dovodi do medijane sa dubinom $\geq 1/2$?*

Funkcije dubine preko parcijalnog uređenja

Neka je $\preceq$ relacija pacijalnog uređenja na $\bar{\mathbb{R}}^n$. Definišemo

$$[\mathbf{a}, \mathbf{b}] = \{\mathbf{x} \in \bar{\mathbb{R}}^d \,|\, \mathbf{a} \preceq x \preceq b\}$$

Primer: Konveksnim konusom K sa vrhom u nuli, definiše se parcijalno uređenje

$$\mathbf{x} \preceq \mathbf{y} \iff \mathbf{y} - \mathbf{x} \in K$$

$$[\mathbf{a}, \mathbf{b}] = (\mathbf{a} + K) \cap (\mathbf{b} - K),$$

Preko ovih intevala, dobijamo medijanu koja ima dubinu $\geq 1/2$, ali nije afino invarijantna.

- Da li se u stvarnim problemima može preko transformacija koordinatnog sistema (data driven coordinate system) dobiti dubina koja bi bila prihvatljiva aproksimacija afinoj invarijantnosti?
- Da li je ovo jedino parcijalno uređenje koje dovodi do medijane sa dubinom $\geq 1/2$?

Funkcije dubine preko parcijalnog uređenja

Neka je $\preceq$ relacija pacijalnog uređenja na $\bar{\mathbb{R}}^n$. Definišemo

$$[\mathbf{a}, \mathbf{b}] = \{\mathbf{x} \in \bar{\mathbb{R}}^d \,|\, \mathbf{a} \preceq x \preceq b\}$$

Primer: Konveksnim konusom K sa vrhom u nuli, definiše se parcijalno uređenje

$$\mathbf{x} \preceq \mathbf{y} \iff \mathbf{y} - \mathbf{x} \in K$$

$$[\mathbf{a}, \mathbf{b}] = (\mathbf{a} + K) \cap (\mathbf{b} - K),$$

Preko ovih intevala, dobijamo medijanu koja ima dubinu $\geq 1/2$, ali nije afino invarijantna.

- *Da li se u stvarnim problemima može preko transformacija koordinatnog sistema (data driven coordinate system) dobiti dubina koja bi bila prihvatljiva aproksimacija afinoj invarijantnosti?*
- *Da li je ovo jedino parcijalno uređenje koje dovodi do medijane sa dubinom $\geq 1/2$?*

Funkcije dubine preko parcijalnog uređenja

Neka je $\preceq$ *relacija pacijalnog uređenja na* $\bar{\mathbb{R}}^n$. *Definišemo*

$$[\mathbf{a},\mathbf{b}] = \{\mathbf{x} \in \bar{\mathbb{R}}^d \,|\, \mathbf{a} \preceq x \preceq b\}$$

Primer: Konveksnim konusom K *sa vrhom u nuli, definiše se parcijalno uređenje*

$$\mathbf{x} \preceq \mathbf{y} \iff \mathbf{y} - \mathbf{x} \in K$$

$$[\mathbf{a},\mathbf{b}] = (\mathbf{a} + K) \cap (\mathbf{b} - K),$$

Preko ovih intevala, dobijamo medijanu koja ima dubinu $\geq 1/2$*, ali nije afino invarijantna.*

- *Da li se u stvarnim problemima može preko transformacija koordinatnog sistema (data driven coordinate system) dobiti dubina koja bi bila prihvatljiva aproksimacija afinoj invarijantnosti?*
- *Da li je ovo jedino parcijalno uređenje koje dovodi do medijane sa dubinom* $\geq 1/2$*?*

Funkcije dubine preko parcijalnog uređenja

Neka je $\preceq$ *relacija pacijalnog uređenja na* $\bar{\mathbb{R}}^n$. *Definišemo*

$$[\mathbf{a},\mathbf{b}] = \{\mathbf{x} \in \bar{\mathbb{R}}^d \,|\, \mathbf{a} \preceq x \preceq b\}$$

Primer: Konveksnim konusom K *sa vrhom u nuli, definiše se parcijalno uređenje*

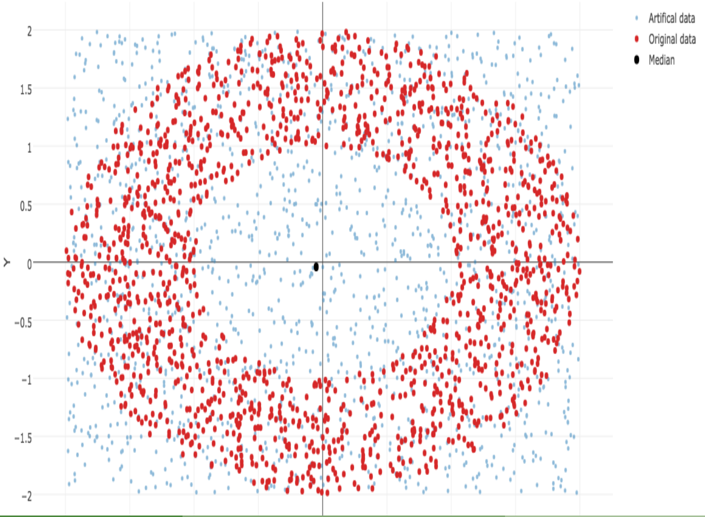
$$\mathbf{x} \preceq \mathbf{y} \iff \mathbf{y} - \mathbf{x} \in K$$

$$[\mathbf{a},\mathbf{b}] = (\mathbf{a} + K) \cap (\mathbf{b} - K),$$

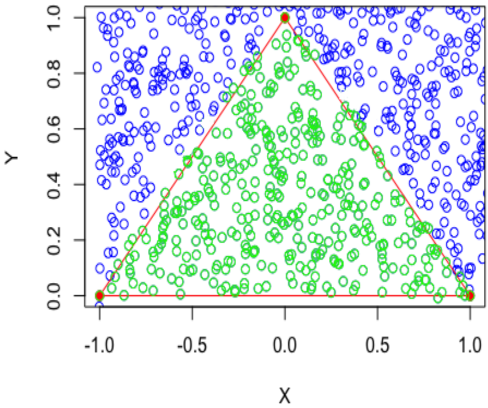
Preko ovih intevala, dobijamo medijanu koja ima dubinu $\geq 1/2$, *ali nije afino invarijantna.*

- *Da li se u stvarnim problemima može preko transformacija koordinatnog sistema (data driven coordinate system) dobiti dubina koja bi bila prihvatljiva aproksimacija afinoj invarijantnosti?*
- *Da li je ovo jedino parcijalno uređenje koje dovodi do medijane sa dubinom* $\geq 1/2$?

Example 1: Uniform distribution in a ring

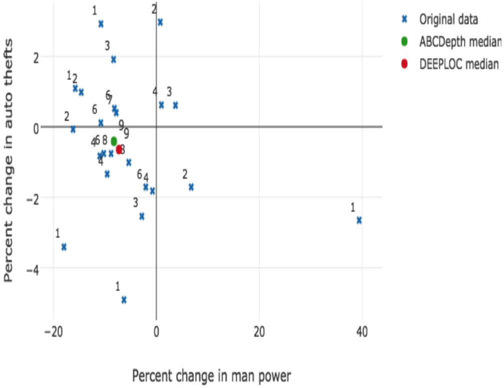


Example 2: Triangle



Triangle (red) and artificial data (green) with depth 1/3

Example 3: Real data set - NY crime data set



NY crime data - points depths and medina: ABCDepth and DEEPLOC

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*

  Input: $X_n = (\mathbf{x_1}, \mathbf{x_1}, ..., \mathbf{x_n}) \in \mathbb{R}^{d \times n}$
  Calculate Euclidian inter-distances.
  Ouput: strictly triangular matrix in a list of lists structure, where i-th list $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*

  Input: list of lists structure with distances
  Sort distances per each point
  Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.
- *Phase 3, balls intersection, iteration phase:*

  Input: hash map structure
  Intersect balls that contains $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  Output: List of level sets, $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, *where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where i-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, *where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where i-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, *where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where i-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha) + 1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, *where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where i-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, *where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where* i*-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$*, where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where* i*-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, *where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where* i*-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$*, where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where* i*-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha) + 1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$*, where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where i-th list $(i = 1, ..., n-1)$ contains distances $d_{i+1,j}, j = 1, \ldots, i$.*
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains $\lfloor n(1-\alpha)+1 \rfloor$ points.*
  - *Output: List of level sets, $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, where S_{α_m} represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where i-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, *where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d\times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where* i*-th list* $(i = 1, ..., n-1)$ *contains distances* $d_{i+1,j}, j = 1, \ldots, i$.
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains* $\lfloor n(1-\alpha)+1 \rfloor$ *points.*
  - *Output: List of level sets,* $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$*, where* S_{α_m} *represents a Tukey median.*

ABCDepth algoritm: median calculation

- *ABCDepth algorithm for finding a Tukey median is implemented in three phases.*
- *Phase 1, calculate distances:*
  - *Input:* $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
  - *Calculate Euclidian inter-distances.*
  - *Ouput: strictly triangular matrix in a list of lists structure, where i-th list $(i = 1, ..., n-1)$ contains distances $d_{i+1,j}, j = 1, \ldots, i$.*
- *Phase 2, construct balls:*
  - *Input: list of lists structure with distances*
  - *Sort distances per each point*
  - *Output: hash map structure, key is a center of a ball, and value is a list with sorted nearest points.*
- *Phase 3, balls intersection, iteration phase:*
  - *Input: hash map structure*
  - *Intersect balls that contains $\lfloor n(1-\alpha)+1 \rfloor$ points.*
  - *Output: List of level sets, $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, where S_{α_m} represents a Tukey median.*

ABCDepth algoritm: pseudocode

Data: Original data, $X_n = (\boldsymbol{x_1}, \boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{d \times n}$
Result: List of level sets, $S = \{S_{\alpha_1}, S_{\alpha_2}, ..., S_{\alpha_m}\}$, where S_{α_m} represents a Tukey median

for $i \leftarrow 2$ **to** n **do**
    for $j \leftarrow 1$ **to** $i - 1$ **do**
        Calculate Euclidian distance between point $\boldsymbol{x_i}$ and point $\boldsymbol{x_j}$;
        Add distance to the list of lists ;
    end
end

for $i \leftarrow 1$ **to** n **do**
    Sort distances for point $\boldsymbol{x_i}$;
    Populate structure with balls ;
end

$size = n$, $\alpha_1 = \frac{1}{d+1}$, $i = 1$;
while $size > 1$ **do**
    $S_{\alpha_i} = \{\bigcap_j^n B_j, \left|B_j\right| = \lfloor n(1 - \alpha_i) + 1 \rfloor$, w.r.t. to original points only $\}$;
    $size = |S_{\alpha_i}|$;
    $\alpha_{i+1} = \alpha_i + \frac{1}{n}$;
    Add S_{α_i} to S ;
end

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*

  The first for loop (line 1) takes all n points, so its complexity is $O(n)$.
  The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time
  Calculation of Euclidian distance takes $O(d)$ time.
  Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$
- *Complexity of Phase 2:*

  The first for loop (line 8) runs in $O(n)$ time.
  For sorting the distances per each point using quicksort takes $O(n \log n)$.
  Overall complexity: $O(n^2 \log n)$.
- *Complexity of Phase 3:*

  While loop repeats k times, where $1 \le k \le m \sim \frac{n}{2}$
  Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.
  Overall complexity: $O(kn^2)$.
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \le k \le m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \le k \le m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of* $O((d+k)n^2 + n^2 \log n)$ *time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is* $O(n)$.
  - *The second for loop (line 2) runs in* $O(\frac{n-1}{2})$ *time*
  - *Calculation of Euclidian distance takes* $O(d)$ *time.*
  - *Overall complexity:* $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in* $O(n)$ *time.*
  - *For sorting the distances per each point using quicksort takes* $O(n \log n)$.
  - *Overall complexity:* $O(n^2 \log n)$.
- *Complexity of Phase 3:*
  - *While loop repeats k times, where* $1 \leq k \leq m \sim \frac{n}{2}$
  - *Intersection of n balls that contains* $\lfloor n(1-\alpha_k)+1 \rfloor$ *points runs in* $O(n^2)$.
  - *Overall complexity:* $O(kn^2)$.
- *ABCDepth algorithm complexity:* $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

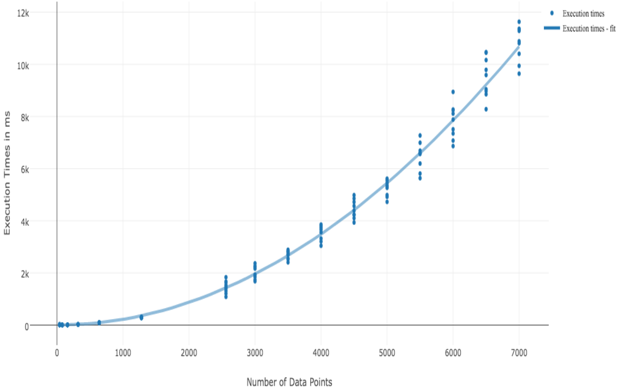
- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

ABCDepth algoritm: complexity

- *ABCDepth algorithm for finding approximate Tukey median has order of $O((d+k)n^2 + n^2 \log n)$ time complexity, where k is the number of iterations in the iteration phase.*
- *Complexity of Phase 1:*
  - *The first for loop (line 1) takes all n points, so its complexity is $O(n)$.*
  - *The second for loop (line 2) runs in $O(\frac{n-1}{2})$ time*
  - *Calculation of Euclidian distance takes $O(d)$ time.*
  - *Overall complexity: $O(\frac{nd(n-1)}{2}) \sim O(dn^2)$*
- *Complexity of Phase 2:*
  - *The first for loop (line 8) runs in $O(n)$ time.*
  - *For sorting the distances per each point using quicksort takes $O(n \log n)$.*
  - *Overall complexity: $O(n^2 \log n)$.*
- *Complexity of Phase 3:*
  - *While loop repeats k times, where $1 \leq k \leq m \sim \frac{n}{2}$*
  - *Intersection of n balls that contains $\lfloor n(1-\alpha_k)+1 \rfloor$ points runs in $O(n^2)$.*
  - *Overall complexity: $O(kn^2)$.*
- *ABCDepth algorithm complexity: $O(dn^2) + O(n^2 \log n) + (kn^2) = O((d+k)n^2 + n^2 \log n)$*

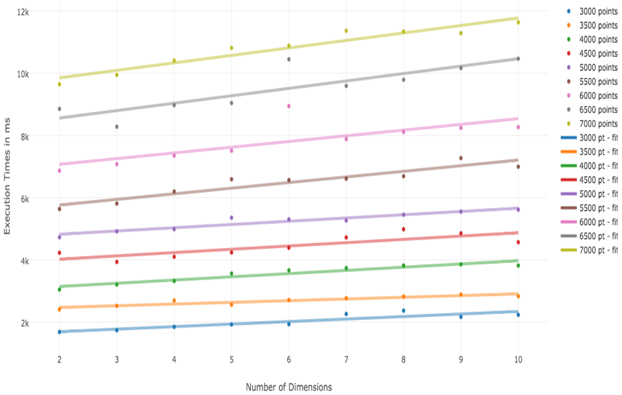
Complexity analysis

- When number of points increases the execution time has growth of order $n^2 \log n$:



Complexity analysis

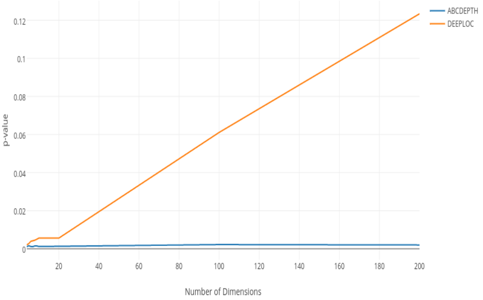
- The execution time grows linearly with dimensionality:



Comparisons

Error size of approximate median points in terms of p-values $P(\chi^2(d) \leq \|\hat{m}\|^2)$ - Sample of size n from standard normal distribution in dimension d

n	Algorithm	d									
		4	5	6	7	8	9	10	20	100	200
1000	DEEPLOC	0.0021	0.0027	0.0039	0.0041	0.0045	0.0051	0.0056	0.0056	0.0612	0.1234
	ABCDEPTH	0.0011	0.0016	0.0012	0.0011	0.0015	0.0015	0.0012	0.0012	0.0022	0.002



Comparisons

Compare DEEPLOC and ABCDepth execution times in seconds.

d	Algorithm	n												
		320	640	1280	2560	3000	3500	4000	4500	5000	5500	6000	6500	7000
50	Deeploc	4.43	7.15	12.65	23.87	30.93	31.79	37.66	45.35	50.72	63.13	63.75	84.13	69.61
	ABCDepth	0.15	0.63	2.86	4.95	7.27	8.65	12.51	14.18	17.51	22.18	25.86	29.24	37.34
100	Deeploc	19.42	22.85	33.81	77.45	69.04	105.56	97.39	120.05	140.04	131.85	127.36	212.42	183.27
	ABCDepth	0.22	0.92	2.03	7.83	9.78	13.14	17.89	23.52	30.6	39.18	49.03	68.46	82.02
500	Deeploc	-	1616.53	*	*	*	*	*	*	*	*	*	*	*
	ABCDepth	0.693	3.181	8.4	27.9	41.61	53.73	71.95	89.36	109.22	140.18	151.45	180.5	213.01
1000	Deeploc	-	-	*	*	*	*	*	*	*	*	*	*	*
	ABCDepth	1.165	3.99	14.389	54.18	74.38	98.73	129.85	164.96	203.37	246.54	286.17	344.94	39.16
2000	Deeploc	-	-	-	*	*	*	*	*	*	*	*	*	*
	ABCDepth	2.21	7.86	27.25	107.46	132.77	180.02	243.1	297.6	386.75	475.87	554.23	666.4	764.74

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data

- According to Wikipedia, the term has been coined in 1990's, to denote data sets with sizes beyond the ability of commonly used software tools to process data within a tolerable time. The adjective "Big"can be related to quantity of data points or/and their dimensionality.
- How much Big Data was big before Internet?
- Collected data (human genomes, health care data...) and user generated data.
- Fun facts:
  - The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
  - We perform 40,000 search queries every second (on Google alone).
  - Facebook users send on average 31.25 million messages and view 2.77 million videos every minute.
  - Facebook stores, accesses, and analyzes 30+ Petabytes of user generated data.
- User generated data is used to predict and to modify human behavior as a means to produce revenue and market control. *[Shoshana Zuboff: Big other - surveillance capitalism and the prospects of an information civilization, Journal of Information Technology (2015) 30, 75-89].*
- Healthcare could save as much as $300 billion a year ??? that???s equal to reducing costs by $1000 a year for every man, woman, and child.
- Fun or not? What should we do with Big Data?

Big Data - Big Muzzy

Who is this guy?

- 
- 
- Is he a programmer, computer scientist, statistician or data scientist?

Who is this guy?

- 
- 
- Is he a programmer, computer scientist, statistician or data scientist?

Who is this guy?

- 
- □ 
- Is he a programmer, computer scientist, statistician or data scientist?

Data Science

- Data scientist before data science was popular - John W. Tukey
- John W. Tukey: Am I a statistician or a data analyst? In "Future of Data Analysis", AMS (1962)
- John W. Tukey, "Exploratory Data Analysis"(EDA) (1977)
- In the year 1972 creates a program PRIM-9 for visualization of high dimensional data.

- *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt...All in all, I have come to feel that my central interest is in data analysis.*

Data Science

- Data scientist before data science was popular - John W. Tukey
- John W. Tukey: Am I a statistician or a data analyst? In "Future of Data Analysis", AMS (1962)
- John W. Tukey, "Exploratory Data Analysis"(EDA) (1977)
- In the year 1972 creates a program PRIM-9 for visualization of high dimensional data.

- *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt...All in all, I have come to feel that my central interest is in data analysis.*

Data Science

- Data scientist before data science was popular - John W. Tukey
- John W. Tukey: Am I a statistician or a data analyst? In "Future of Data Analysis", AMS (1962)
- John W. Tukey, "Exploratory Data Analysis"(EDA) (1977)
- In the year 1972 creates a program PRIM-9 for visualization of high dimensional data.

- *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt...All in all, I have come to feel that my central interest is in data analysis.*

Data Science

- Data scientist before data science was popular - John W. Tukey
- John W. Tukey: Am I a statistician or a data analyst? In "Future of Data Analysis", AMS (1962)
- John W. Tukey, "Exploratory Data Analysis"(EDA) (1977)
- In the year 1972 creates a program PRIM-9 for visualization of high dimensional data.

- *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt...All in all, I have come to feel that my central interest is in data analysis.*

Data Science

- Data scientist before data science was popular - John W. Tukey
- John W. Tukey: Am I a statistician or a data analyst? In "Future of Data Analysis", AMS (1962)
- John W. Tukey, "Exploratory Data Analysis"(EDA) (1977)
- In the year 1972 creates a program PRIM-9 for visualization of high dimensional data.

- *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt...All in all, I have come to feel that my central interest is in data analysis.*

Data Science

- Data scientist before data science was popular - John W. Tukey
- John W. Tukey: Am I a statistician or a data analyst? In "Future of Data Analysis", AMS (1962)
- John W. Tukey, "Exploratory Data Analysis"(EDA) (1977)
- In the year 1972 creates a program PRIM-9 for visualization of high dimensional data.

- *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt...All in all, I have come to feel that my central interest is in data analysis.*

John Wilder Tukey (1915-2000)

- Born 1915, B.Sc. in Chemistry at Brown University 1936, M.Sc Chemistry 1937
- Ph.D Mathematics (topology) 1939, Princeton. His dissertation was included in Halmos' list of most significant mathematical contributions in USA. Tukey's partial order is still discussed in contemporary literature (2017).

John Wilder Tukey (1915-2000)

- Born 1915, B.Sc. in Chemistry at Brown University 1936, M.Sc Chemistry 1937
- Ph.D Mathematics (topology) 1939, Princeton. His dissertation was included in Halmos' list of most significant mathematical contributions in USA. Tukey's partial order is still discussed in contemporary literature (2017).

Data Science

- Data science is derived from data analysis and computer science + necessary domain knowledge.
- "Data scientiest (n.) : Person who is better at statistics than any software engineer and better at software engineering than any statistician."
- "What is a Data Scientist? An analyst who lives in California."
- Jack of all trades, master on none...
- Still better than master of one.

Data Science

- Data science is derived from data analysis and computer science + necessary domain knowledge.
- "Data scientiest (n.) : Person who is better at statistics than any software engineer and better at software engineering than any statistician."
- "What is a Data Scientist? An analyst who lives in California."
- Jack of all trades, master on none...
- Still better than master of one.

Data Science

- Data science is derived from data analysis and computer science + necessary domain knowledge.
- "Data scientiest (n.) : Person who is better at statistics than any software engineer and better at software engineering than any statistician."
- "What is a Data Scientist? An analyst who lives in California."
- Jack of all trades, master on none...
- Still better than master of one.

Data Science

- Data science is derived from data analysis and computer science + necessary domain knowledge.
- "Data scientiest (n.) : Person who is better at statistics than any software engineer and better at software engineering than any statistician."
- "What is a Data Scientist? An analyst who lives in California."
- Jack of all trades, master on none...
- Still better than master of one.

Data Science

- Data science is derived from data analysis and computer science + necessary domain knowledge.
- "Data scientiest (n.) : Person who is better at statistics than any software engineer and better at software engineering than any statistician."
- "What is a Data Scientist? An analyst who lives in California."
- Jack of all trades, master on none...
- Still better than master of one.

Application: ABCD Clustering algorithm - Analyze the data

- The initial and the most important task in every data science work is to understand the data and to understand the problem that should be solved.
- We analyze acoustic data set that contains probability density functions of frequency dependent angular distributions for external noise incident energies.
- Noise incident energies are taken from $l = 12$ locations, L_i, $i = 1 \ldots l$ and each location is described with $n = 10$ continuous functions at a certain frequency band.
- Each of those continuous functions are discretized into $m = 91$ noise incidence angles, so each L_i can be represented as a matrix $m \times n$

$$L_i = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix}$$

Application: ABCD Clustering algorithm - Analyze the data

- The initial and the most important task in every data science work is to understand the data and to understand the problem that should be solved.
- We analyze acoustic data set that contains probability density functions of frequency dependent angular distributions for external noise incident energies.
- Noise incident energies are taken from $l = 12$ locations, L_i, $i = 1 \ldots l$ and each location is described with $n = 10$ continuous functions at a certain frequency band.
- Each of those continuous functions are discretized into $m = 91$ noise incidence angles, so each L_i can be represented as a matrix $m \times n$

$$L_i = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix}$$

Application: ABCD Clustering algorithm - Analyze the data

- The initial and the most important task in every data science work is to understand the data and to understand the problem that should be solved.
- We analyze acoustic data set that contains probability density functions of frequency dependent angular distributions for external noise incident energies.
- Noise incident energies are taken from $l = 12$ locations, L_i, $i = 1...l$ and each location is described with $n = 10$ continuous functions at a certain frequency band.
- Each of those continuous functions are discretized into $m = 91$ noise incidence angles, so each L_i can be represented as a matrix $m \times n$

$$L_i = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix}$$

Application: ABCD Clustering algorithm - Analyze the data

- The initial and the most important task in every data science work is to understand the data and to understand the problem that should be solved.
- We analyze acoustic data set that contains probability density functions of frequency dependent angular distributions for external noise incident energies.
- Noise incident energies are taken from $l = 12$ locations, L_i, $i = 1...l$ and each location is described with $n = 10$ continuous functions at a certain frequency band.
- Each of those continuous functions are discretized into $m = 91$ noise incidence angles, so each L_i can be represented as a matrix $m \times n$

$$L_i = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix}$$

ABCD Clustering algorithm - Analyze the data

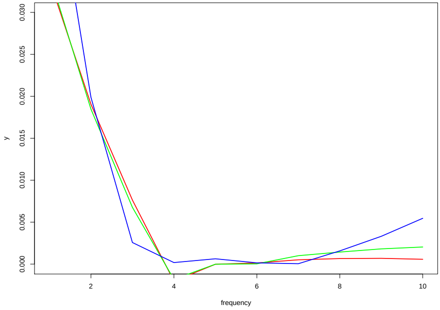
- Or:

$$L_i = \{\mathbf{X}_j\}, j = 1, .., m, \mathbf{X}_j = \{x_{k,j}\},\ \ k = 1, .., n$$

- Alternatively, we consider location, L_i as a set of m functions:

$$L_i = \{f_1(\mathrm{f}_k, \Theta_1), f_2(\mathrm{f}_k, \Theta_2), ..., f_m(\mathrm{f}_k, \Theta_m)\}, k = 1, .., n$$

- Three functions for $j = 18$ are shown, i.e. for three randomly picked locations we show their $f_{18}(\mathrm{f}_k, \Theta_{18})$ function. The red and green functions have similar y values, unlike the blue function. We conclude that two locations are similar if they have as many similar functions as possible.



Functions of three locations for $j = 18$

ABCD Clustering algorithm - Analyze the data

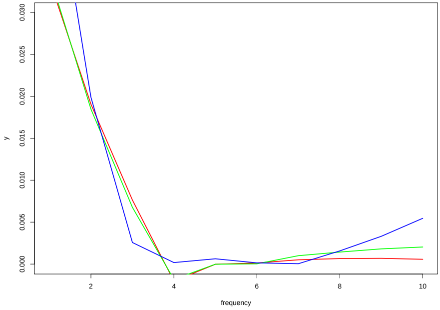
- Or:

$$L_i = \{\mathbf{X}_j\}, j = 1, .., m, \mathbf{X}_j = \{x_{k,j}\}, \ k = 1, .., n$$

- Alternatively, we consider location, L_i as a set of m functions:

$$L_i = \{f_1(\mathrm{f}_k, \Theta_1), f_2(\mathrm{f}_k, \Theta_2), ..., f_m(\mathrm{f}_k, \Theta_m)\}, k = 1, .., n$$

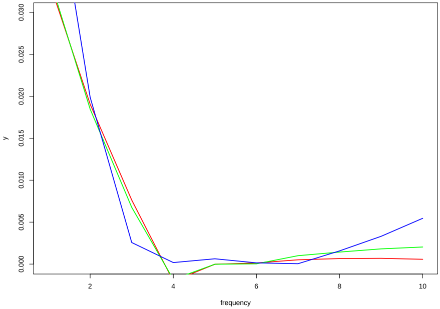
- Three functions for $j = 18$ are shown, i.e. for three randomly picked locations we show their $f_{18}(\mathrm{f}_k, \Theta_{18})$ function. The red and green functions have similar y values, unlike the blue function. We conclude that two locations are similar if they have as many similar functions as possible.



Functions of three locations for $j = 18$

ABCD Clustering algorithm - Analyze the data

- Or:
$$L_i = \{\mathbf{X}_j\}, j = 1, .., m, \mathbf{X}_j = \{x_{k,j}\},\ \ k = 1, .., n$$
- Alternatively, we consider location, L_i as a set of m functions:
$$L_i = \{f_1(\mathrm{f}_k, \Theta_1), f_2(\mathrm{f}_k, \Theta_2), ..., f_m(\mathrm{f}_k, \Theta_m)\}, k = 1, .., n$$
- Three functions for $j = 18$ are shown, i.e. for three randomly picked locations we show their $f_{18}(\mathrm{f}_k, \Theta_{18})$ function. The red and green functions have similar y values, unlike the blue function. We conclude that two locations are similar if they have as many similar functions as possible.



Functions of three locations for $j = 18$

ABCD Clustering algorithm - Analyze the data

- Define the problem and the aim:
- Each location represents a different type of a street - some streets are wide, some of them are narrow, streets are bordered with high-rise or low-rise buildings, parking lots or trains can be close to the streets, streets are more or less busy etc. Every of those parameters has an influence on noise incidence energies.
- The aim is to cluster those locations, i.e. to find the way the make clusters that relies on locations' similarities. Based on locations similarities, a proper facade noise isolation can be found for each location type (cluster).

ABCD Clustering algorithm - Analyze the data

- Define the problem and the aim:
- Each location represents a different type of a street - some streets are wide, some of them are narrow, streets are bordered with high-rise or low-rise buildings, parking lots or trains can be close to the streets, streets are more or less busy etc. Every of those parameters has an influence on noise incidence energies.
- The aim is to cluster those locations, i.e. to find the way the make clusters that relies on locations' similarities. Based on locations similarities, a proper facade noise isolation can be found for each location type (cluster).

ABCD Clustering algorithm - Analyze the data

- Define the problem and the aim:
- Each location represents a different type of a street - some streets are wide, some of them are narrow, streets are bordered with high-rise or low-rise buildings, parking lots or trains can be close to the streets, streets are more or less busy etc. Every of those parameters has an influence on noise incidence energies.
- The aim is to cluster those locations, i.e. to find the way the make clusters that relies on locations' similarities. Based on locations similarities, a proper facade noise isolation can be found for each location type (cluster).

ABCD Clustering algorithm - Dimensionality reduction and median calculation

- ABCD Clustering algorithm uses median value to get the distances between data points and median point, i.e. each multidimensional point is represented by its distance from median value and the multidimensional data set is reduced to one dimension.
- The approach of dimensionality reduction is based on ABCDepth algorithm described in the previous section.
- We calculate median for $\mathbf{X}_j$ vectors from each L_i matrix:

$$med_j(\{L_1(X_j), L_2(X_j), .., L_i(X_j)\}), \; i = 1, ..12 \; and \; j = 1, ..., m.$$

- For each $\mathbf{X}_j$ from L_i matrix, the algorithm calculates the distance between X_j and median med_j:

$$dist(L_i(X_j), med_j) = d_{i_j}.$$

- That way, we have m one-dimensional data points, instead of m points of dimension n, and each X_j is represented with its distance:

$$L_i = \{d_{i_1}, d_{i_2}, ..., d_{i_j}\}, \; where j = 1, ..., m.$$

ABCD Clustering algorithm - Dimensionality reduction and median calculation

- ABCD Clustering algorithm uses median value to get the distances between data points and median point, i.e. each multidimensional point is represented by its distance from median value and the multidimensional data set is reduced to one dimension.
- The approach of dimensionality reduction is based on ABCDepth algorithm described in the previous section.
- We calculate median for $\mathbf{X}_j$ vectors from each L_i matrix:

$$med_j(\{L_1(X_j), L_2(X_j), .., L_i(X_j)\}),\ i = 1, ..12\ and\ j = 1, ..., m.$$

- For each $\mathbf{X}_j$ from L_i matrix, the algorithm calculates the distance between X_j and median med_j:

$$dist(L_i(X_j), med_j) = d_{i_j}.$$

- That way, we have m one-dimensional data points, instead of m points of dimension n, and each X_j is represented with its distance:

$$L_i = \{d_{i_1}, d_{i_2}, ..., d_{i_j}\},\ where j = 1, ..., m.$$

ABCD Clustering algorithm - Dimensionality reduction and median calculation

- ABCD Clustering algorithm uses median value to get the distances between data points and median point, i.e. each multidimensional point is represented by its distance from median value and the multidimensional data set is reduced to one dimension.
- The approach of dimensionality reduction is based on ABCDepth algorithm described in the previous section.
- We calculate median for $\mathbf{X}_j$ vectors from each L_i matrix:

$$med_j(\{L_1(X_j), L_2(X_j), .., L_i(X_j)\}), \ i = 1, ..12 \ and \ j = 1, ..., m.$$

- For each $\mathbf{X}_j$ from L_i matrix, the algorithm calculates the distance between X_j and median med_j:

$$dist(L_i(X_j), med_j) = d_{i_j}.$$

- That way, we have m one-dimensional data points, instead of m points of dimension n, and each X_j is represented with its distance:

$$L_i = \{d_{i_1}, d_{i_2}, ..., d_{i_j}\}, \ wherej = 1, ..., m.$$

ABCD Clustering algorithm - Dimensionality reduction and median calculation

- ABCD Clustering algorithm uses median value to get the distances between data points and median point, i.e. each multidimensional point is represented by its distance from median value and the multidimensional data set is reduced to one dimension.
- The approach of dimensionality reduction is based on ABCDepth algorithm described in the previous section.
- We calculate median for $\mathbf{X}_j$ vectors from each L_i matrix:

$$med_j(\{L_1(X_j), L_2(X_j), .., L_i(X_j)\}),\ i = 1, ..12\ and\ j = 1, ..., m.$$

- For each $\mathbf{X}_j$ from L_i matrix, the algorithm calculates the distance between X_j and median med_j:

$$dist(L_i(X_j), med_j) = d_{i_j}.$$

- That way, we have m one-dimensional data points, instead of m points of dimension n, and each X_j is represented with its distance:

$$L_i = \{d_{i_1}, d_{i_2}, ..., d_{i_j}\},\ where j = 1, ..., m.$$

ABCD Clustering algorithm - Dimensionality reduction and median calculation

- ABCD Clustering algorithm uses median value to get the distances between data points and median point, i.e. each multidimensional point is represented by its distance from median value and the multidimensional data set is reduced to one dimension.
- The approach of dimensionality reduction is based on ABCDepth algorithm described in the previous section.
- We calculate median for $\mathbf{X}_j$ vectors from each L_i matrix:

$$med_j(\{L_1(X_j), L_2(X_j), .., L_i(X_j)\}), \ i = 1, ..12 \ and \ j = 1, ..., m.$$

- For each $\mathbf{X}_j$ from L_i matrix, the algorithm calculates the distance between X_j and median med_j:

$$dist(L_i(X_j), med_j) = d_{i_j}.$$

- That way, we have m one-dimensional data points, instead of m points of dimension n, and each X_j is represented with its distance:

$$L_i = \{d_{i_1}, d_{i_2}, ..., d_{i_j}\}, \ wherej = 1, .., m.$$

ABCD Clustering algorithm - Clustering of distances

- ABCD Clustering algorithm uses *MultiKMeansPlusPlusClusterer* function to cluster distances calculated in the previous step.
- Iteratively, algorithm groups distances d_{i_j} from each matrix L_i. The number of iterations is equal to m.
- At the end of each iteration, there are maximal l (number of locations) clusters, in the case if each location belongs to the different cluster.
- In other words, algorithm groups $\{d_{1_1}, d_{2_1}, ..., d_{12_1}\}$ distances in the first iteration, $\{d_{1_2}, d_{2_2}, ..., d_{12_2}\}$ distances in the second iteration, etc.
- In ABCD Clustering algorithms, clusters obtained from described process are called: clusters of type C.

ABCD Clustering algorithm - Clustering of distances

- ABCD Clustering algorithm uses *MultiKMeansPlusPlusClusterer* function to cluster distances calculated in the previous step.
- Iteratively, algorithm groups distances d_{i_j} from each matrix L_i. The number of iterations is equal to m.
- At the end of each iteration, there are maximal l (number of locations) clusters, in the case if each location belongs to the different cluster.
- In other words, algorithm groups $\{d_{1_1}, d_{2_1}, ..., d_{12_1}\}$ distances in the first iteration, $\{d_{1_2}, d_{2_2}, ..., d_{12_2}\}$ distances in the second iteration, etc.
- In ABCD Clustering algorithms, clusters obtained from described process are called: clusters of type C.

ABCD Clustering algorithm - Clustering of distances

- ABCD Clustering algorithm uses *MultiKMeansPlusPlusClusterer* function to cluster distances calculated in the previous step.
- Iteratively, algorithm groups distances d_{i_j} from each matrix L_i. The number of iterations is equal to m.
- At the end of each iteration, there are maximal l (number of locations) clusters, in the case if each location belongs to the different cluster.
- In other words, algorithm groups $\{d_{1_1}, d_{2_1}, ..., d_{12_1}\}$ distances in the first iteration, $\{d_{1_2}, d_{2_2}, ..., d_{12_2}\}$ distances in the second iteration, etc.
- In ABCD Clustering algorithms, clusters obtained from described process are called: clusters of type C.

ABCD Clustering algorithm - Clustering of distances

- ABCD Clustering algorithm uses *MultiKMeansPlusPlusClusterer* function to cluster distances calculated in the previous step.
- Iteratively, algorithm groups distances d_{i_j} from each matrix L_i. The number of iterations is equal to m.
- At the end of each iteration, there are maximal l (number of locations) clusters, in the case if each location belongs to the different cluster.
- In other words, algorithm groups $\{d_{1_1}, d_{2_1}, ..., d_{12_1}\}$ distances in the first iteration, $\{d_{1_2}, d_{2_2}, ..., d_{12_2}\}$ distances in the second iteration, etc.
- In ABCD Clustering algorithms, clusters obtained from described process are called: clusters of type C.

ABCD Clustering algorithm - Clustering of distances

- ABCD Clustering algorithm uses *MultiKMeansPlusPlusClusterer* function to cluster distances calculated in the previous step.
- Iteratively, algorithm groups distances d_{i_j} from each matrix L_i. The number of iterations is equal to m.
- At the end of each iteration, there are maximal l (number of locations) clusters, in the case if each location belongs to the different cluster.
- In other words, algorithm groups $\{d_{1_1}, d_{2_1}, ..., d_{12_1}\}$ distances in the first iteration, $\{d_{1_2}, d_{2_2}, ..., d_{12_2}\}$ distances in the second iteration, etc.
- In ABCD Clustering algorithms, clusters obtained from described process are called: clusters of type C.

ABCD Clustering algorithm - Clustering of distances

- After m iterations, the algorithm counts how many times location L_i appeared in the same cluster of type C with some other location.
- ABCD Clustering does re-clustering of locations (those new clusters are called clusters of type C') in the following way: location, L_x, $x \in \{1...12\}$ is placed in a new cluster of type C' with some other location L_y, $y \in \{1...12\} \setminus x$ iff L_x appeared most times with location L_y in clusters of type C.
- In that case, we say that L_x and L_y are in relation:

$$L_x \rho L_y$$

.

- Clusters of type C' are in a form of connected components of a weighted graph whose reachability is an equivalence relation. According to its transitive property:

$$L_x \rho L_y) \wedge (L_y \rho L_z) \implies L_x \rho L_z.$$

ABCD Clustering algorithm - Clustering of distances

- After m iterations, the algorithm counts how many times location L_i appeared in the same cluster of type C with some other location.
- ABCD Clustering does re-clustering of locations (those new clusters are called clusters of type C') in the following way: location, L_x, $x \in \{1...12\}$ is placed in a new cluster of type C' with some other location L_y, $y \in \{1...12\} \setminus x$ iff L_x appeared most times with location L_y in clusters of type C.
- In that case, we say that L_x and L_y are in relation:

$$L_x \rho L_y$$

.

- Clusters of type C' are in a form of connected components of a weighted graph whose reachability is an equivalence relation. According to its transitive property:

$$L_x \rho L_y) \wedge (L_y \rho L_z) \implies L_x \rho L_z.$$

ABCD Clustering algorithm - Clustering of distances

- After m iterations, the algorithm counts how many times location L_i appeared in the same cluster of type C with some other location.
- ABCD Clustering does re-clustering of locations (those new clusters are called clusters of type C') in the following way: location, L_x, $x \in \{1...12\}$ is placed in a new cluster of type C' with some other location L_y, $y \in \{1...12\} \setminus x$ iff L_x appeared most times with location L_y in clusters of type C.
- In that case, we say that L_x and L_y are in relation:

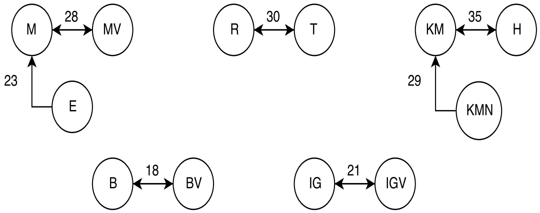
$$L_x \rho L_y$$

.

- Clusters of type C' are in a form of connected components of a weighted graph whose reachability is an equivalence relation. According to its transitive property:

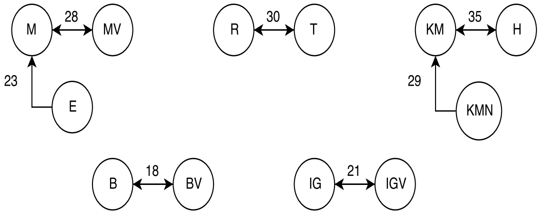
$$L_x \rho L_y) \wedge (L_y \rho L_z) \implies L_x \rho L_z.$$

ABCD Clustering algorithm - Results



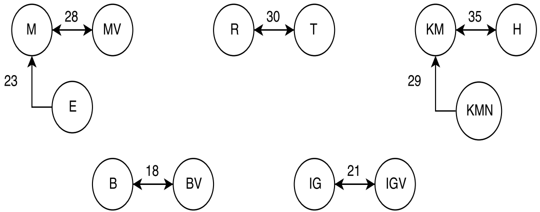
- Each connected component represent one cluster of type C'.
- Nodes represent locations.
- Extraverted edges shows how many times locations L_x and L_y appeared with each other in clusters of type C. Locations M and MV appeared the most (28) times with each other in clusters of type C
- Directed edges from location L_x to location L_y show how many times location L_x appeared in the same cluster of type C with location L_y. Lcation E appeared the most (23) times with location M.
- Due to the transitive property explained above, locations M, MV and E make one cluster of type C'.
- An application of ABCD Clustering algorithm with detailed data set description can be found in Milos̑ Bjelić's PhD thesis, *Analiza ugaone raspodele incidentne energije spoljas̑nje buke primenom mikrofonskog niza*, section 5.4.

ABCD Clustering algorithm - Results



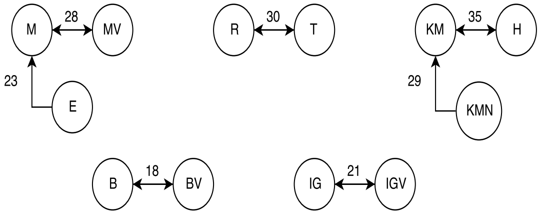
- Each connected component represent one cluster of type C'.
- Nodes represent locations.
- Extraverted edges shows how many times locations L_x and L_y appeared with each other in clusters of type C. Locations M and MV appeared the most (28) times with each other in clusters of type C
- Directed edges from location L_x to location L_y show how many times location L_x appeared in the same cluster of type C with location L_y. Lcation E appeared the most (23) times with location M.
- Due to the transitive property explained above, locations M, MV and E make one cluster of type C'.
- An application of ABCD Clustering algorithm with detailed data set description can be found in Milos̑ Bjelić's PhD thesis, *Analiza ugaone raspodele incidentne energije spoljas̑nje buke primenom mikrofonskog niza*, section 5.4.

ABCD Clustering algorithm - Results



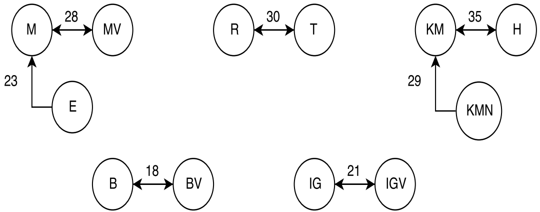
- Each connected component represent one cluster of type C'.
- Nodes represent locations.
- Extraverted edges shows how many times locations L_x and L_y appeared with each other in clusters of type C. Locations M and MV appeared the most (28) times with each other in clusters of type C
- Directed edges from location L_x to location L_y show how many times location L_x appeared in the same cluster of type C with location L_y. Lcation E appeared the most (23) times with location M.
- Due to the transitive property explained above, locations M, MV and E make one cluster of type C'.
- An application of ABCD Clustering algorithm with detailed data set description can be found in Milos̑ Bjelić's PhD thesis, *Analiza ugaone raspodele incidentne energije spoljas̑nje buke primenom mikrofonskog niza*, section 5.4.

ABCD Clustering algorithm - Results



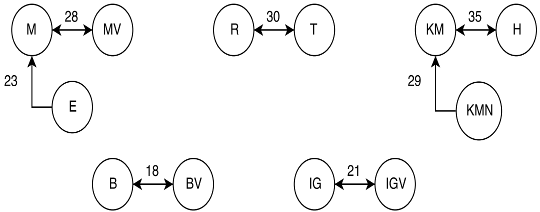
- Each connected component represent one cluster of type C'.
- Nodes represent locations.
- Extraverted edges shows how many times locations L_x and L_y appeared with each other in clusters of type C. Locations M and MV appeared the most (28) times with each other in clusters of type C
- Directed edges from location L_x to location L_y show how many times location L_x appeared in the same cluster of type C with location L_y. Lcation E appeared the most (23) times with location M.
- Due to the transitive property explained above, locations M, MV and E make one cluster of type C'.
- An application of ABCD Clustering algorithm with detailed data set description can be found in Miloŝ Bjelić's PhD thesis, *Analiza ugaone raspodele incidentne energije spoljaŝnje buke primenom mikrofonskog niza*, section 5.4.

ABCD Clustering algorithm - Results



- Each connected component represent one cluster of type C'.
- Nodes represent locations.
- Extraverted edges shows how many times locations L_x and L_y appeared with each other in clusters of type C. Locations M and MV appeared the most (28) times with each other in clusters of type C
- Directed edges from location L_x to location L_y show how many times location L_x appeared in the same cluster of type C with location L_y. Lcation E appeared the most (23) times with location M.
- Due to the transitive property explained above, locations M, MV and E make one cluster of type C'.
- An application of ABCD Clustering algorithm with detailed data set description can be found in Miloŝ Bjelić's PhD thesis, *Analiza ugaone raspodele incidentne energije spoljaŝnje buke primenom mikrofonskog niza*, section 5.4.

ABCD Clustering algorithm - Results



- Each connected component represent one cluster of type C'.
- Nodes represent locations.
- Extraverted edges shows how many times locations L_x and L_y appeared with each other in clusters of type C. Locations M and MV appeared the most (28) times with each other in clusters of type C
- Directed edges from location L_x to location L_y show how many times location L_x appeared in the same cluster of type C with location L_y. Lcation E appeared the most (23) times with location M.
- Due to the transitive property explained above, locations M, MV and E make one cluster of type C'.
- An application of ABCD Clustering algorithm with detailed data set description can be found in Milos̑ Bjelić's PhD thesis, *Analiza ugaone raspodele incidentne energije spoljas̑nje buke primenom mikrofonskog niza*, section 5.4.

Further work

- *Generalize ABCDepth Clustering algorithm (unsupervised)*
- *Introduce ABCDepth Classification algorithm (supervised)*
- *Median-based dimensionality reduction*
- *Outlier detection*

Further work

- *Generalize ABCDepth Clustering algorithm (unsupervised)*
- *Introduce ABCDepth Classification algorithm (supervised)*
- *Median-based dimensionality reduction*
- *Outlier detection*

Further work

- *Generalize ABCDepth Clustering algorithm (unsupervised)*
- *Introduce ABCDepth Classification algorithm (supervised)*
- *Median-based dimensionality reduction*
- *Outlier detection*

Further work

- *Generalize ABCDepth Clustering algorithm (unsupervised)*
- *Introduce ABCDepth Classification algorithm (supervised)*
- *Median-based dimensionality reduction*
- *Outlier detection*

References

[1] Bogićević, M., and Merkle, M. ABCDepth: efficient algorithm for Tukey depth. *arXiv:1603.05609v2* (2016.).

[2] Bogićević, M., and Merkle, M. Approximate calculation of Tukey's depth and median with high-dimensional data. *Yugosl. J. Oper. Res. 28* (2018), 475–500. doi:10.2298/YJOR180520022B.

[3] Chakraborty, B., and Chaudhuri, P. On a transformation and retransformation technique for constructing an affine equivariant multivariate median. *Proc. Amer. Math. Soc. 124*, 8 (1996), 2539–2547.

[4] Donoho, D. L. *Breakdown properties of multivariate location estimators.* PhD thesis, Harvard University, Cambridge, Massachusetts, USA, 1982.

[5] Dutta, S., Ghosh, A. K., and Chaudhuri, P. Some intriguing properties of Tukey's half-space depth. *Bernoulli 17* (2011.), 1420–1434.

[6] Ghosh, A. K., and Chaudhuri, P. On data depth and distribution-free discriminant analysis using separating surfaces. *Bernoulli 11*, 1 (2005.), 1–27.

[7] Liu, X., Mosler, K., , and Mozharovskyi., P. Fast computation of Tukey trimmed regions in dimension $p > 2$. *arXiv:1412.5122* (2014.).

[8] Liu, X. H., and Luo, S. H. The limit of finite sample breakdown point of Tukeys halfspace median for general data. *Acta Math. Sin. (Engl. Ser) 34* (2018), 1403–1416.

[9] Liu, X. H., Zuo, Y., and Wang, Q. Finite sample breakdown point of tukeys halfspace median. arXiv:1604.07039v1, 2018.

[10] Merkle, M. Jensen's inequality for medians. *Stat. Prob. Letters 71* (2005.), 277–281.

[11] Merkle, M. Jensen's inequality for multivariate medians. *J. Math. Anal. Appl. 370* (2010.), 258–269.

[12] Small, C. G. A survey of multidimensional medians. *Internat. Statist. Inst. Rev. 58* (1990.), 263–277.

[13] Tukey, J. Order statistics. In Mimeographed notes for Statistics 411, Princeton University., 1974.