Searching for a globally optimal partition

R. Scitovski, I. Vazler

Department of Mathematics, University of Osijek

May 22, 2013

Problem statement

$\mathcal{A} = \{a^i \in \mathbb{R}^n : i = 1, \ldots, m\} \subset \mathbb{R}^n$, $|\mathcal{A}| = m \gg n$ (set of data)

$1 \leq k \leq m$ (number of clusters)

$\Pi(\mathcal{A}) = \{\pi_1, \ldots, \pi_k\}$ (partition)

$$
\begin{array}{ll}
(i) & \bigcup_{i=1}^{k} \pi_i = \mathcal{A}, \\
(ii) & \pi_i \cap \pi_j = \emptyset, \quad i \neq j, \\
(iii) & |\pi_j| \geq 1, \quad j = 1, \ldots, k
\end{array}
$$

$\mathcal{P}(\mathcal{A}; m, k)$ (set of all partitions)

$w_i > 0$ (the weight associated to each data point)

Problem statement

$d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, $\mathbb{R}_+ = [0, +\infty\rangle$ (distance-like function)

J.Kogan, *Introduction to clustering large and high-dimensional data*, Cambridge University Press, 2007

M.Teboulle, Journal of Machine Learning Research, **8**(2007), 65-102

Center of the cluster $\pi_j \in \Pi$:

$$c_j = c(\pi_j) := \underset{x \in \mathrm{conv}(\pi_j)}{\operatorname{argmin}} \sum_{a^i \in \pi_j} w_i d(x, a^i).$$

Objective function $\mathcal{F}\colon \mathcal{P}(\mathcal{A}; m, k) \to \mathbb{R}_+$,

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

Optimization problem:

determine an optimal partition $\Pi^\star$*, such that*

$$\Pi^\star = \underset{\Pi \in \mathcal{P}(\mathcal{A}; m, k)}{\operatorname{argmin}} \mathcal{F}(\Pi)$$

Problem statement

$d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+,\ \mathbb{R}_+ = [0, +\infty\rangle$ (distance-like function)

J.Kogan, *Introduction to clustering large and high-dimensional data*, Cambridge University Press, 2007

M.Teboulle, Journal of Machine Learning Research, **8**(2007), 65-102

Center of the cluster $\pi_j \in \Pi$:

$$c_j = c(\pi_j) := \operatorname*{argmin}_{x\in\mathrm{conv}(\pi_j)} \sum_{a^i\in\pi_j} w_i d(x, a^i).$$

Objective function $\mathcal{F}\colon \mathcal{P}(\mathcal{A}; m, k) \to \mathbb{R}_+$,

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

Optimization problem:

determine an optimal partition $\Pi^\star$, *such that*

$$\Pi^\star = \operatorname*{argmin}_{\Pi\in\mathcal{P}(\mathcal{A};m,k)} \mathcal{F}(\Pi)$$

Problem statement

$d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, $\mathbb{R}_+ = [0, +\infty\rangle$ (distance-like function)

J.Kogan, *Introduction to clustering large and high-dimensional data*, Cambridge University Press, 2007

M.Teboulle, Journal of Machine Learning Research, **8**(2007), 65-102

Center of the cluster $\pi_j \in \Pi$:

$$c_j = c(\pi_j) := \operatorname*{argmin}_{x \in \mathrm{conv}(\pi_j)} \sum_{a^i \in \pi_j} w_i d(x, a^i).$$

Objective function $\mathcal{F}\colon \mathcal{P}(\mathcal{A}; m, k) \to \mathbb{R}_+$,

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

Optimization problem:

determine an optimal partition $\Pi^\star$, such that

$$\Pi^\star = \operatorname*{argmin}_{\Pi \in \mathcal{P}(\mathcal{A}; m, k)} \mathcal{F}(\Pi)$$

Problem statement

$d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, $\mathbb{R}_+ = [0, +\infty\rangle$ (distance-like function)

J.Kogan, *Introduction to clustering large and high-dimensional data*, Cambridge University Press, 2007

M.Teboulle, Journal of Machine Learning Research, **8**(2007), 65-102

Center of the cluster $\pi_j \in \Pi$:

$$c_j = c(\pi_j) := \underset{x \in \mathrm{conv}(\pi_j)}{\mathrm{argmin}} \sum_{a^i \in \pi_j} w_i d(x, a^i).$$

Objective function $\mathcal{F}\colon \mathcal{P}(\mathcal{A}; m, k) \to \mathbb{R}_+$,

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

Optimization problem:

determine an optimal partition $\Pi^\star$, such that

$$\Pi^\star = \underset{\Pi \in \mathcal{P}(\mathcal{A}; m, k)}{\mathrm{argmin}} \mathcal{F}(\Pi)$$

Number of partitions

$$|\mathcal{P}(\mathcal{A}; m, k)| = \frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}\binom{k}{j}j^m$$ (Stirling number of the second kind)

$\lvert\mathcal{P}(\mathcal{A}; m, k)\rvert \approx$	$k = 2$	$k = 5$	$k = 10$
$m = 10$	511	42525	1
$m = 10^3$	10^{300}	10^{697}	10^{993}
$m = 10^6$	$10^{301\,029}$	$10^{698\,968}$	10^{10^6}

It is not possible to search directly for the optimal partition

Number of partitions

$$|\mathcal{P}(\mathcal{A}; m, k)| = \frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}\binom{k}{j}j^m$$ (Stirling number of the second kind)

$\lvert\mathcal{P}(\mathcal{A}; m, k)\rvert \approx$	$k = 2$	$k = 5$	$k = 10$
$m = 10$	511	42525	1
$m = 10^3$	10^{300}	10^{697}	10^{993}
$m = 10^6$	$10^{301\,029}$	$10^{698\,968}$	10^{10^6}

It is not possible to search directly for the optimal partition

k-means problem

$c_1, \ldots, c_k \in \mathbb{R}^n$ (given set of centers)

Minimal distance principle: $\Pi = \{\pi(c_1), \ldots, \pi(c_k)\}$

$$\pi(c_j) = \{a \in \mathcal{A} : d(c_j, a) \leq d(c_s, a), \forall s = 1, \ldots, k\}, \qquad j = 1, \ldots, k,$$

Problem of finding an optimal partition of the set $\mathcal{A}$:

$$\underset{c_1, \ldots, c_k \in \mathbb{R}^n}{\operatorname{argmin}} F(c_1, \ldots, c_k), \qquad F(c_1, \ldots, c_k) = \sum_{i=1}^{m} \min_{1 \leq s \leq k} w_i d(c_s, a^i),$$

F. Aurenhammer, R. Klein, *Handbook of Computational Geometry, Chapter V*. Elsevier Science Publishing, 2000.

M.Teboulle, P.Berkhin, I.Dhilon Y.Guan, J.Kogan, *Clustering with entropy-like k-means algorithms*, J.Kogan, C.Nicholas, M.Teboulle (Eds.) Grouping Multidimensional Data, Springer, 2006, 127-160

k-means problem

$c_1, \dots, c_k \in \mathbb{R}^n$ (given set of centers)

Minimal distance principle: $\Pi = \{\pi(c_1), \dots, \pi(c_k)\}$

$$\pi(c_j) = \{a \in \mathcal{A} : d(c_j, a) \leq d(c_s, a), \forall s = 1, \dots, k\}, \qquad j = 1, \dots, k,$$

Problem of finding an optimal partition of the set $\mathcal{A}$:

$$\operatorname*{argmin}_{c_1,\dots,c_k \in \mathbb{R}^n} F(c_1, \dots, c_k), \qquad F(c_1, \dots, c_k) = \sum_{i=1}^{m} \min_{1 \leq s \leq k} w_i d(c_s, a^i),$$

F. Aurenhammer, R. Klein, *Handbook of Computational Geometry, Chapter V*. Elsevier Science Publishing, 2000.

M.Teboulle, P.Berkhin, I.Dhilon Y.Guan, J.Kogan, *Clustering with entropy-like k-means algorithms*, J.Kogan, C.Nicholas, M.Teboulle (Eds.) Grouping Multidimensional Data, Springer, 2006, 127-160

k-means problem

$c_1, \dots, c_k \in \mathbb{R}^n$ (given set of centers)

Minimal distance principle: $\Pi = \{\pi(c_1), \dots, \pi(c_k)\}$

$$\pi(c_j) = \{a \in \mathcal{A} : d(c_j, a) \leq d(c_s, a), \forall s = 1, \dots, k\}, \qquad j = 1, \dots, k,$$

Problem of finding an optimal partition of the set $\mathcal{A}$:

$$\underset{c_1,\dots,c_k \in \mathbb{R}^n}{\operatorname{argmin}} F(c_1, \dots, c_k), \qquad F(c_1, \dots, c_k) = \sum_{i=1}^{m} \min_{1 \leq s \leq k} w_i d(c_s, a^i),$$

F. Aurenhammer, R. Klein, *Handbook of Computational Geometry, Chapter V*. Elsevier Science Publishing, 2000.

M.Teboulle, P.Berkhin, I.Dhilon Y.Guan, J.Kogan, *Clustering with entropy-like k-means algorithms*, J.Kogan, C.Nicholas, M.Teboulle (Eds.) Grouping Multidimensional Data, Springer, 2006, 127-160

Distance–like functions

Distance–like function

$$d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \qquad \text{(positive definite function)}$$

Least Squares (LS) distance–like function:

$$d_{LS} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_{LS}(x,y) = \|x-y\|_2^2;$$

$$c_j = \operatorname*{argmin}_{x \in \mathrm{conv}(\pi_j)} \sum_{a^i \in \pi_j} d_{LS}(x, a^i) = \frac{1}{W_j} \sum_{a^i \in \pi_j} w_i a^i, \qquad W_j = \sum_{a^i \in \pi_j} w_i$$

Mahalanobis distance–like function:

$$d_M : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_M(x,y) = (x-y)\Sigma^{-1}(x-y)^T \qquad (\Sigma \text{ - covariance matrix})$$

$$c_j = \operatorname*{argmin}_{x \in \mathrm{conv}(\pi_j)} \sum_{a^i \in \pi_j} d_M(x, a^i) = \frac{1}{W_j} \sum_{a^i \in \pi_j} w_i a^i, \qquad W_j = \sum_{a^i \in \pi_j} w_i$$

Distance–like functions

Distance–like function

$$d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \qquad \text{(positive definite function)}$$

Least Squares (LS) distance–like function:

$$d_{LS}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_{LS}(x,y) = \|x-y\|_2^2;$$

$$c_j = \underset{x\in\mathrm{conv}(\pi_j)}{\operatorname{argmin}} \sum_{a^i\in\pi_j} d_{LS}(x,a^i) = \frac{1}{W_j}\sum_{a^i\in\pi_j} w_i a^i, \qquad W_j = \sum_{a^i\in\pi_j} w_i$$

Mahalanobis distance–like function:

$$d_M\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_M(x,y) = (x-y)\Sigma^{-1}(x-y)^T \qquad (\Sigma \text{ - covariance matrix})$$

$$c_j = \underset{x\in\mathrm{conv}(\pi_j)}{\operatorname{argmin}} \sum_{a^i\in\pi_j} d_M(x,a^i) = \frac{1}{W_j}\sum_{a^i\in\pi_j} w_i a^i, \qquad W_j = \sum_{a^i\in\pi_j} w_i$$

Distance–like functions

Distance–like function

$$d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \qquad \text{(positive definite function)}$$

Least Squares (LS) distance–like function:

$$d_{LS}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_{LS}(x,y) = \|x-y\|_2^2;$$

$$c_j = \operatorname*{argmin}_{x\in\mathrm{conv}(\pi_j)} \sum_{a^i\in\pi_j} d_{LS}(x,a^i) = \frac{1}{W_j}\sum_{a^i\in\pi_j} w_i a^i, \qquad W_j = \sum_{a^i\in\pi_j} w_i$$

Mahalanobis distance–like function:

$$d_M\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_M(x,y) = (x-y)\Sigma^{-1}(x-y)^T \qquad (\Sigma \text{ - covariance matrix})$$

$$c_j = \operatorname*{argmin}_{x\in\mathrm{conv}(\pi_j)} \sum_{a^i\in\pi_j} d_M(x,a^i) = \frac{1}{W_j}\sum_{a^i\in\pi_j} w_i a^i, \qquad W_j = \sum_{a^i\in\pi_j} w_i$$

Distance–like functions

Least Absolute Deviations (LAD) distance function:

$$d_{LAD}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_{LAD}(x,y) = \|x-y\|_1$$

$$c_j = \operatorname*{argmin}_{x \in \operatorname{conv}(\pi_j)} \sum_{a^i \in \pi_j} d_{LAD}(x, a^i) = \operatorname*{med}_{a^i \in \pi_j}(w_i, a^i)$$

Distance function on the unit circle $d_K\colon K \times K \to \mathbb{R}_+$:

$$\begin{aligned} d_K(a(\tau_1), a(\tau_2)) &= \begin{cases} |\tau_1 - \tau_2|, & \text{if } |\tau_1 - \tau_2| \le \pi, \\ 2\pi - |\tau_1 - \tau_2|, & \text{if } |\tau_1 - \tau_2| > \pi, \end{cases} \\ &= \pi - \big||\tau_1 - \tau_2| - \pi\big|, \qquad \tau_1, \tau_2 \in [0, 2\pi] \end{aligned}$$

There is no explicit formula for the cluster center

Distance–like functions

Least Absolute Deviations (LAD) distance function:

$$d_{LAD}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+, \quad d_{LAD}(x,y) = \|x-y\|_1$$

$$c_j = \operatorname*{argmin}_{x\in\mathrm{conv}(\pi_j)} \sum_{a^i\in\pi_j} d_{LAD}(x,a^i) = \operatorname*{med}_{a^i\in\pi_j}(w_i, a^i)$$

Distance function on the unit circle $d_K\colon K\times K \to \mathbb{R}_+$:

$$\begin{aligned} d_K(a(\tau_1), a(\tau_2)) &= \begin{cases} |\tau_1-\tau_2|, & \text{if } |\tau_1-\tau_2| \le \pi, \\ 2\pi - |\tau_1-\tau_2|, & \text{if } |\tau_1-\tau_2| > \pi, \end{cases} \\ &= \pi - \left||\tau_1-\tau_2| - \pi\right|, \qquad \tau_1,\tau_2 \in [0,2\pi] \end{aligned}$$

There is no explicit formula for the cluster center

GPAs of successful second-year students majoring in mathematics at the Department of Mathematics, University of Osijek.

Student	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
GPA	2.2	2.35	2.5	2.64	2.85	3.	3.25	3.35	3.4	3.54
Student	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}
GPA	3.54	3.7	3.72	3.72	3.8	3.85	3.95	4.05	4.15	4.2
Student	s_{21}	s_{22}	s_{23}	s_{24}	s_{25}	s_{26}	s_{27}	s_{28}	s_{29}	s_{30}
GPA	4.2	4.3	4.41	4.41	4.54	4.6	4.6	4.65	4.84	5.

We will look at the problem of finding an optimal partition with two clusters.

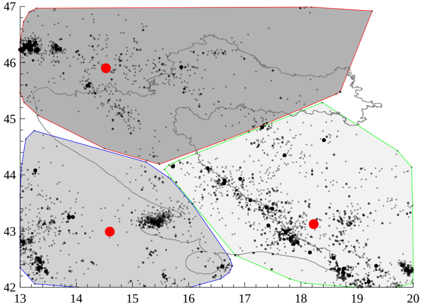
K.Sabo, R.Scitovski, I.Vazler, *Data clustering (in Croatian)*, Osječki matematički list, **10**(2010) 149-178

Earthquake locations of magnitude at least 3 in a wider area of Croatia since 1973

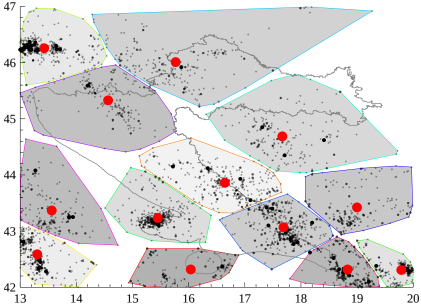
Earthquake locations of magnitude at least 3 in a wider area of Croatia since 1973

$\mathcal{A} = \{a^i = (\lambda_i, \varphi_i) \in \mathbb{R}^2 \colon 13 \le \lambda_i \le 20, \quad 42 \le \varphi_i \le 47, \quad i = 1, \dots, 3184\}$

a) WM-optimal 3-partitions



b) WM-optimal 13-partitions

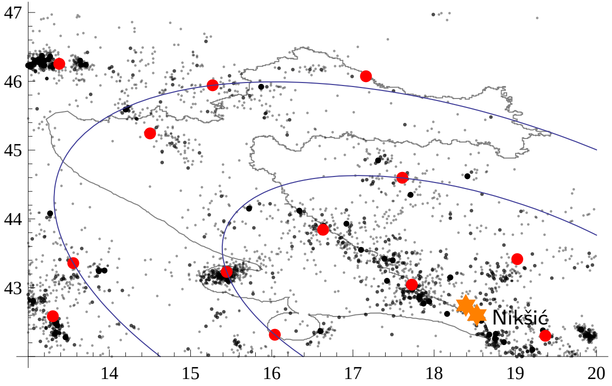


Data source:

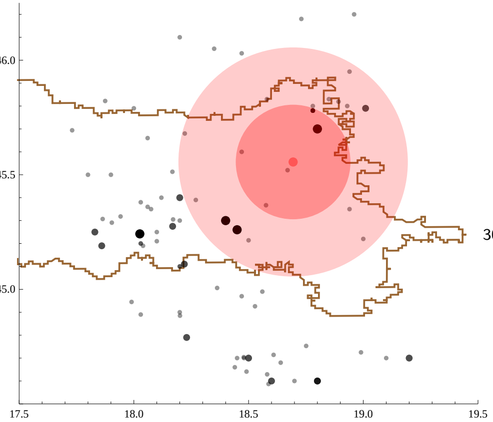
http://earthquake.usgs.gov/earthquakes/eqarchives/epic/

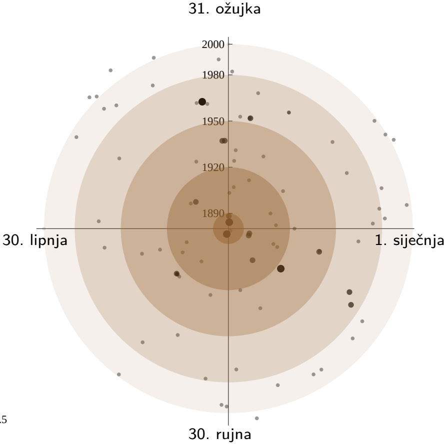
Geometric position of WM-optimal cluster centers

Earthquake locations of magnitude at least 3 in a wider area of Croatia since 1973
$\mathcal{A} = \{a^i = (\lambda_i, \varphi_i) \in \mathbb{R}^2 \colon 13 \leq \lambda_i \leq 20, \quad 42 \leq \varphi_i \leq 47, \quad i = 1, \ldots, 3184\}$



Earthquake locations of magnitude at least 3 in a wider area of Osijek since 1880





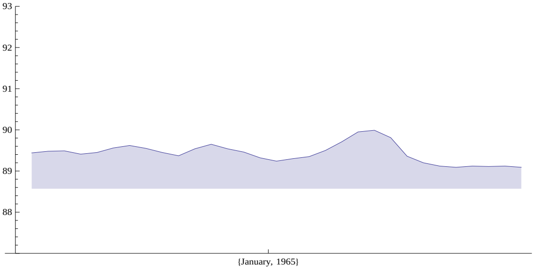
Investigating and forecasting a high water level of the Drava River by D.Miholjac from 1 January 1900 to 1 February 2012

$(T_i, w_i),\ i = 1, \dots, N,\ T_i \in [0, 112]$ (some date between 1900-1-1 and 2012-2-1)

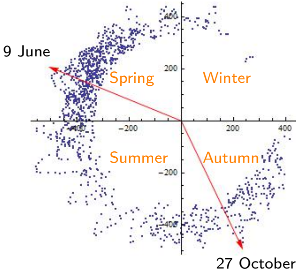
w_i (measured water level value on the day T_i)

$t_i = 2\pi T_i (\mathrm{mod}\ 2\pi) \in [0, 2\pi],$ (transformed dates)

$$\mathcal{A} = \{a^i = w_i(\cos t_i, \sin t_i) \in \mathbb{R}^2 \colon w_i - l(t_i) \geq 100\}$$



Burn diagram



J.Parajka et al., Journal of Hydrology 394 (2010) 78–89
Data source: Water Management Department for the Drava and Danube River Basin District, headquartered in Osijek

Acceptable definition of constituencies

Data: $a^i = (x_i, y_i),\ i = 1, \ldots, m$ (positions of cities or municipalities)

q_i (number of voters in the city a^i)

$Q = \sum_{i=1}^{m} q_i$ (total number of voters)

Problem: The territory of the country should be divided into k constituencies $\pi_1, \ldots, \pi_k$, such that

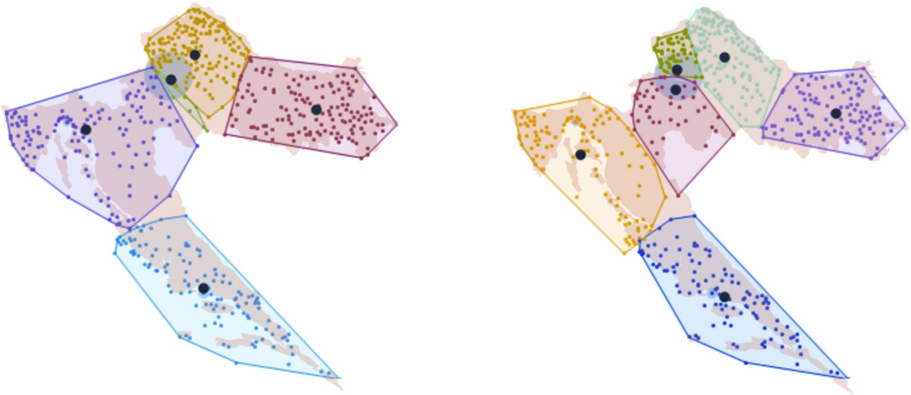
$$\left(1 - \tfrac{p}{100}\right) \tfrac{Q}{k} \leq |\pi_j| \leq \left(1 + \tfrac{p}{100}\right) \tfrac{Q}{k}, \quad j = 1, \ldots, k$$

It is permitted that the voters of a city can be divided into several constituencies (e.g. the city of Zagreb)

K.Sabo, R.Scitovski, P.Taler, *Uniform distribution of the number of voters per constituency on the basis of a mathematical model (in Croatian)*, Hrvatska i komparativna javna uprava 14(2012) 229-249

5 or 6 constituencies in the Republic of Croatia

Optimal/appropriate number of constituencies for the Republic of Croatia



The most appropriate number of clusters in a partition

- The number of clusters in a partition is determined by the nature of the problem itself
  - Student grade point averages
  - Crop rows detection
- The number of clusters in a partition is not given in advance
  - Earthquake locations
  - Acceptable definition of constituencies

The most appropriate number of clusters in a partition

- Calinski-Harabasz Index (CH)
- Davies - Bouldin Index (DB)
- Dunn Index
- Silhouette Width Criterion (SVC)
- Simplify Silhouette Width Criterion (SSC)
- Separability Index

K.Sabo, R.Scitovski, I.Vazler, *Cluster stability in a partition and applications*, Advances in Data Analysis and Classification, Revision process

Searching for a globally optimal partition

$$\underset{\Pi \in \mathcal{P}(\mathcal{A};m,k)}{\operatorname{argmin}} \mathcal{F}(\Pi), \qquad \mathcal{F}(\Pi) = \sum_{j=1}^{k} \sum_{a^i \in \pi_j} w_i d(c_j, a^i)$$

$$\underset{c_1,\ldots,c_k \in \Omega}{\operatorname{argmin}} F(c_1,\ldots,c_k), \qquad F(c_1,\ldots,c_k) = \sum_{i=1}^{m} w_i \min_{1 \le s \le k} d(c_s, a^i)$$

$$\Omega = \prod_{i=1}^{n} [\alpha_i, \beta_i] \subset \mathbb{R}^n$$

Searching for a globally optimal partition

$$\underset{\Pi\in\mathcal{P}(\mathcal{A};m,k)}{\operatorname{argmin}} \mathcal{F}(\Pi), \qquad \mathcal{F}(\Pi) = \sum_{j=1}^{k} \sum_{a^i\in\pi_j} w_i d(c_j, a^i)$$

$$\underset{c_1,\ldots,c_k\in\Omega}{\operatorname{argmin}} F(c_1,\ldots,c_k), \qquad F(c_1,\ldots,c_k) = \sum_{i=1}^{m} w_i \min_{1\leq s\leq k} d(c_s, a^i)$$

$$\Omega = \prod_{i=1}^{n} [\alpha_i, \beta_i] \subset \mathbb{R}^n$$

k-means algorithm

Initialization: $\mathcal{A}$, $\Pi = \{\pi_1, \ldots, \pi_k\}$

Assignment: $c_j = \operatorname*{argmin}_{x \in \mathbb{R}} \sum_{a \in \pi_j} w_i d(x, a), \quad j = 1, \ldots, k;$

$$\mathcal{F}(\Pi) = \sum_{i=1}^{k} \sum_{a \in \pi_j} w_i d(c_j, a)$$

Update: $\nu_j = \{a \in \mathcal{A} : d(c_j, a) \leq d(c_s, a), \, s = 1, \ldots, k\}, \, j = 1, \ldots, k$

V.Volkovich, J.Kogan, C.Nicholas, *Building initial partitions through sampling techniques*, European Journal of Operational Research, 2007

F.Leisch, *A toolbox for k-centroids cluster analysis*, Computat. Stat. & Data Analysis, 2006

Methods of searching for the globally optimal partition

The application of general methods for global search

- genetic algorithms (Auger and Doerr,2011, Yang,2009)
- interval analysis (Hansen and Walster,2004)

C.A.Floudas, C.E.Gounaris, *A review of recent advances in global optimization*, JOGO, 2009

A.Neumaier,*Complete search in continuous global optimization and constraint satisfaction*, Acta Numerica, Cambridge University Press, 2006

A.Auger, B.Doerr, *Theory of Randomized Search Heuristics*, World Scientific, Danvers, 2011

X.S.Yang, *Firefly Algorithms for Multimodal Optimization* Proc. of the 5th international conference on Stochastic algorithms: foundations and applications, 2009

E.Hansen, G.W.Walster, *Global optimization using interval analysis*, Marcel Dekker, New York, 2004

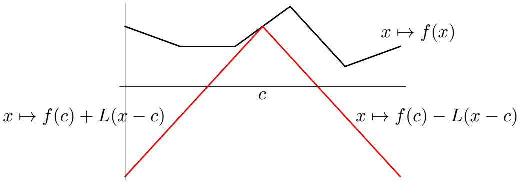
Global optimization for Lipschitz continuous function

Pijavskij, 1972   Shubert, 1972

DIvided RECTangles method (Jones et al., 1993)

Lower bound of the function $f : [a, b] \to \mathbb{R}$

$$\phi(x) = \begin{cases} f(c) + L(x - c), & x \le c \\ f(c) - L(x - c), & x \ge c \end{cases}, \qquad c = \frac{a+b}{2}$$



$$\min_{x \in [a,b]} \phi(x) = f(c) - L\frac{b-a}{2} =: \mathcal{B}$$

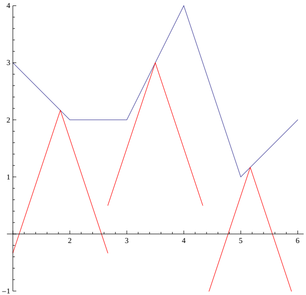
С. А. Пиявский, Один алгоритм отыскания абсолютного екстрмума функции, Ж. вычисл. матем. и матем. физ., 1972

B.Shubert, *A sequential method seeking the global maximum of a function*, SIAM Journal on Numerical Analysis, 1972

D.R.Jones, C.D.Perttunen, B.E.Stuckman, *Lipschitzian optimization without the Lipschitz constant*, Journal of Optimization Theory and Applications, 1993

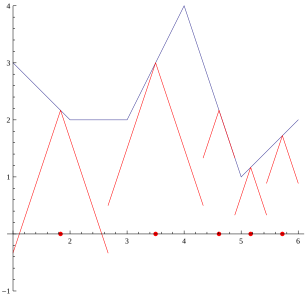
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



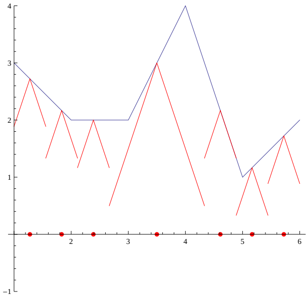
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



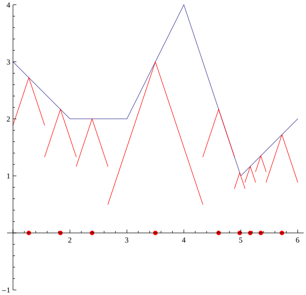
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



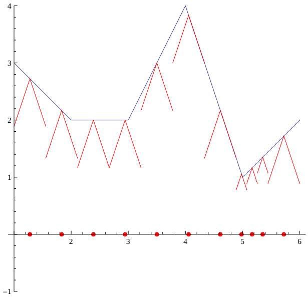
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



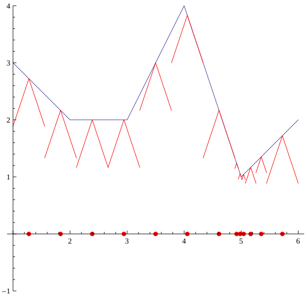
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



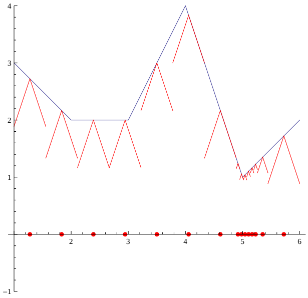
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



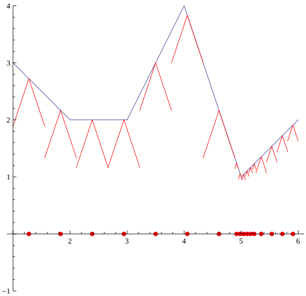
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



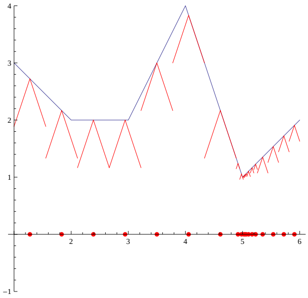
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



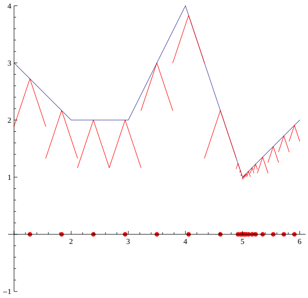
The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.

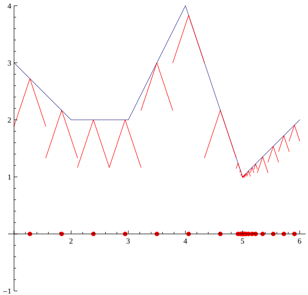


The DIRECT algorithm

- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.

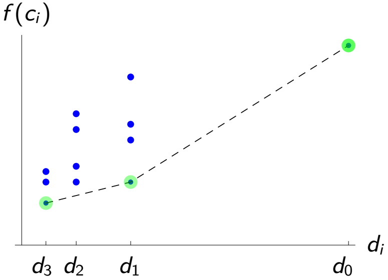


- The interval $[a, b]$ with center $c = \frac{a+b}{2}$ is divided into three equal parts, whereby the center of the middle subinterval is once again the point c.
- For each subinterval, the $\mathcal{B}$-value is determined.
- The subinterval with the least $\mathcal{B}$-value is divided further.
- The global minimum of the function f is then searched for between the points representing the centers of subintervals.



Potentially optimal intervals

$[\alpha_1, \beta_1], \dots, [\alpha_m, \beta_m]$ – subintervals with centers $c_1, \dots, c_m$ and half-widths $d_1, \dots, d_m$



$$[\alpha_i, \beta_i] \leftrightarrow T_i = (d_i, f(c_i))$$
$$i = 1 \dots, m$$

$$\mathcal{B} = f(c) - L\frac{b-a}{2}$$

The straight line with the slope L which passes through the point T_i has an ordinate equal to the $\mathcal{B}$-value

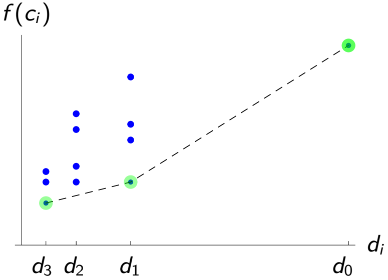
The intervals to be divided further *(potentially optimal intervals):*

lower bound of the convex hull of the points T_i.

It is possible to choose *potentially optimal intervals* without using the Lipschitz constant L.

Potentially optimal intervals

$[\alpha_1, \beta_1], \ldots, [\alpha_m, \beta_m]$ – subintervals with centers $c_1, \ldots, c_m$ and half-widths $d_1, \ldots, d_m$



$$[\alpha_i, \beta_i] \leftrightarrow T_i = (d_i, f(c_i))$$
$$i = 1 \ldots, m$$

$$\mathcal{B} = f(c) - L\frac{b-a}{2}$$

The straight line with the slope L which passes through the point T_i has an ordinate equal to the $\mathcal{B}$-value

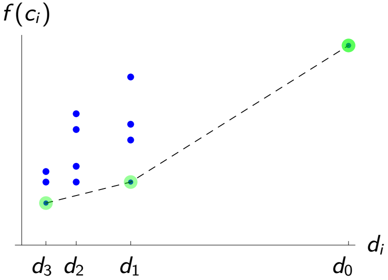
The intervals to be divided further *(potentially optimal intervals):*

lower bound of the convex hull of the points T_i.

It is possible to choose *potentially optimal intervals* without using the Lipschitz constant L.

Potentially optimal intervals

$[\alpha_1, \beta_1], \ldots, [\alpha_m, \beta_m]$ – subintervals with centers $c_1, \ldots, c_m$ and half-widths $d_1, \ldots, d_m$



$$[\alpha_i, \beta_i] \leftrightarrow T_i = (d_i, f(c_i))$$
$$i = 1 \ldots, m$$

$$\mathcal{B} = f(c) - L\tfrac{b-a}{2}$$

The straight line with the slope L which passes through the point T_i has an ordinate equal to the $\mathcal{B}$-value

The intervals to be divided further *(potentially optimal intervals):*

lower bound of the convex hull of the points T_i.

It is possible to choose *potentially optimal intervals* without using the Lipschitz constant L.

Generalization to the function of several variables

$g\colon \Omega \to \mathbb{R}, \quad \Omega = \prod_{i=1}^{n}[\alpha_i, \beta_i] \subset \mathbb{R}^n$ – Lipschitz continuous function

$f\colon [0,1]^n \to \mathbb{R}\ , \qquad f = g \circ T^{-1}$

$[0,1]^n$ with the center $c = (\frac{1}{2}, \ldots, \frac{1}{2}) \in \mathbb{R}^n$ is divided into smaller hyperrectangles, out of which one has again the center in the point c:

$R_i(c_i, (h_1^{(i)}, \ldots, h_n^{(i)})), \quad c_i = (\zeta_1^{(i)}, \ldots, \zeta_n^{(i)}), \quad h_j^{(i)}, j = 1, \ldots, n$

$R_i(c_i, d_i), \qquad d_i = \max\{h_1^{(i)}, \ldots, h_n^{(i)}\}$

D. E.Finkel, C. T.Kelley, *Convergence analysis of the DIRECT algorithm*, CRSC-TR04-28 Center for Research in Sci. Comput., North Carolina State Univ., 2004

D. E.Finkel, C. T.Kelley, *Additive scaling and the DIRECT algorithm*, JOGO, 2006

J. M.Gablonsky, *DIRECT Version 2.0*, Center for Research in Sci. Comput., North Carolina State Univ., 2001

D.R.Jones, C.D.Perttunen, B.E.Stuckman, *Lipschitzian optimization without the Lipschitz constant*, JOTA, 1993

Generalization to the function of several variables

$g\colon \Omega \to \mathbb{R}, \quad \Omega = \prod_{i=1}^{n}[\alpha_i, \beta_i] \subset \mathbb{R}^n$ – Lipschitz continuous function

$f\colon [0,1]^n \to \mathbb{R}\ , \qquad f = g \circ T^{-1}$

$[0,1]^n$ with the center $c = (\frac{1}{2}, \dots, \frac{1}{2}) \in \mathbb{R}^n$ is divided into smaller hyperrectangles, out of which one has again the center in the point c:

$$R_i(c_i, (h_1^{(i)}, \dots, h_n^{(i)})), \quad c_i = (\zeta_1^{(i)}, \dots, \zeta_n^{(i)}), \quad h_j^{(i)}, j = 1, \dots, n$$

$$R_i(c_i, d_i), \qquad d_i = \max\{h_1^{(i)}, \dots, h_n^{(i)}\}$$

D. E.Finkel, C. T.Kelley, *Convergence analysis of the DIRECT algorithm*, CRSC-TR04-28 Center for Research in Sci. Comput., North Carolina State Univ., 2004

D. E.Finkel, C. T.Kelley, *Additive scaling and the DIRECT algorithm*, JOGO, 2006

J. M.Gablonsky, *DIRECT Version 2.0*, Center for Research in Sci. Comput., North Carolina State Univ., 2001

D.R.Jones, C.D.Perttunen, B.E.Stuckman, *Lipschitzian optimization without the Lipschitz constant*, JOTA, 1993

Generalization to the function of several variables

$g\colon \Omega \to \mathbb{R}, \quad \Omega = \prod_{i=1}^{n}[\alpha_i, \beta_i] \subset \mathbb{R}^n$ – Lipschitz continuous function

$f\colon [0,1]^n \to \mathbb{R}\ , \qquad f = g \circ T^{-1}$

$[0,1]^n$ with the center $c = (\frac{1}{2}, \ldots, \frac{1}{2}) \in \mathbb{R}^n$ is divided into smaller hyperrectangles, out of which one has again the center in the point c:

$$R_i(c_i, (h_1^{(i)}, \ldots, h_n^{(i)})), \qquad c_i = (\zeta_1^{(i)}, \ldots, \zeta_n^{(i)}), \qquad h_j^{(i)}, j = 1, \ldots, n$$

$$R_i(c_i, d_i), \qquad d_i = \max\{h_1^{(i)}, \ldots, h_n^{(i)}\}$$

D. E.Finkel, C. T.Kelley, *Convergence analysis of the DIRECT algorithm*, CRSC-TR04-28 Center for Research in Sci. Comput., North Carolina State Univ., 2004

D. E.Finkel, C. T.Kelley, *Additive scaling and the DIRECT algorithm*, JOGO, 2006

J. M.Gablonsky, *DIRECT Version 2.0*, Center for Research in Sci. Comput., North Carolina State Univ., 2001

D.R.Jones, C.D.Perttunen, B.E.Stuckman, *Lipschitzian optimization without the Lipschitz constant*, JOTA, 1993

$f\colon [0,1]^n \to \mathbb{R}$ – symmetric Lipschitz continuous function, i.e.

$$F(c_1, \ldots, c_k) = \sum_{i=1}^{m} \min_{1\le s\le k} w_i d(c_s, a^i)$$

Global optimization problem:

Find the point $x^\star = \operatorname*{argmin}_{x\in\Delta} f(x)$, *such that* $f(x^\star) = \inf_{x\in\Delta} f(x)$,

where

$$\Delta = \{x \in [0,1]^n\colon x_1 \ge \cdots \ge x_n\}.$$

Region Δ represents the n!-th part of the domain of the function f.

In our modifications: those hyperrectangles that are completely or only partially contained in the region Δ will be divided

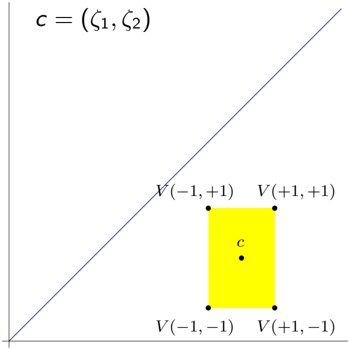
R. Grbić, E. K. Nyarko, R. Scitovski, *A modification of the DIRECT method for Lipschitz global optimization for a symmetric function*, Journal of Global Optimization; Published online: 23 December 2012

Find the point $x^\star = \underset{x\in\Delta}{\operatorname{argmin}} f(x)$*, such that* $f(x^\star) = \inf_{x\in\Delta} f(x)$*,*

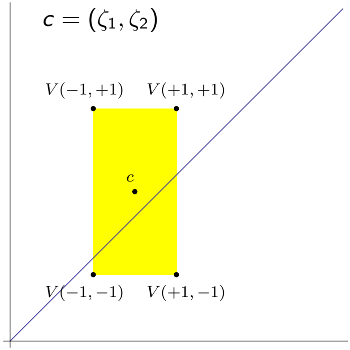
where

$$\Delta = \{x = (x_1, x_2) \in [0,1]^2 \colon x_1 \ge x_2\}.$$

(a) $R \subset \Delta \quad (\zeta_1 - h_1 \ge \zeta_2 + h_2)$



(b) $R \cap \Delta \ne \emptyset \quad (\zeta_1 + h_1 \ge \zeta_2 - h_2)$



$$V(\sigma_1, \sigma_2) = (\zeta_1 + \sigma_1 h_1, \zeta_2 + \sigma_2 h_2), \quad \sigma_1, \sigma_2 \in S = \{-1, +1\}$$

$f\colon [0,1]^3 \to \mathbb{R}$

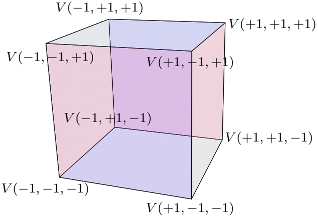
$$\Delta = \{(x_1, x_2, x_3) \in [0,1]^3 \colon x_1 \geq x_2 \geq x_3\} \qquad \text{(tetrahedron)}$$

Lemma 1

$R(c, (h_1, h_2, h_3))$, $c = (\zeta_1, \zeta_2, \zeta_3)$, h_i-half side-lengths in the direction of unit vectors e_i. Then it holds

(i) *$R \subset \Delta$ if and only if $\zeta_1 - h_1 \geq \zeta_2 + h_2$ and $\zeta_2 - h_2 \geq \zeta_3 + h_3$*

(ii) *$R \cap \Delta \neq \emptyset$ if and only if*
$(\zeta_1 + h_1 \geq \zeta_2 - h_2 \geq \zeta_3 - h_3)$ or $(\zeta_1 + h_1 \geq \zeta_2 + h_2 \geq \zeta_3 - h_3)$



$f\colon [0,1]^n \to \mathbb{R}$

$$\Delta = \{(x_1, \dots x_n) \in [0,1]^n \colon x_1 \geq x_2 \geq \cdots \geq x_n\} \qquad \text{(hypertetrahedron)}$$

Theorem 1

Let $R(c,(h_1,\dots,h_n))$ be a hyperrectangle contained in unit hypercube $[0,1]^n$ with center $c = (\zeta_1,\dots,\zeta_n)$, half side-lengths h_i in the direction of unit vector e_i and with vertices $V(\sigma_1,\dots,\sigma_n) = (\zeta_1 + \sigma_1 h_1, \dots, \zeta_n + \sigma_n h_n)$, where $\sigma_1,\dots,\sigma_n \in S = \{-1,+1\}$. Then the following holds

(i) *$R \subset \Delta$ if and only if the following $(n-1)$ conditions hold:*

$$\zeta_i - h_i \geq \zeta_{i+1} + h_{i+1}, \quad \forall\, i = 1,\dots,n-1$$

(ii) *$R \cap \Delta \neq \emptyset$ if and only if there exists $\sigma_2, \dots \sigma_{n-1} \in S$ such that (2^{n-2} possibilities)*

$$\zeta_1 + h_1 \geq \zeta_2 + \sigma_2 h_2 \geq \cdots \geq \zeta_{n-1} + \sigma_{n-1} h_{n-1} \geq \zeta_n - h_n.$$

Numerical experiments: one-dimensional center-based problems

Methods	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
SymDIRECT	0 : 0 : 02	0 : 0 : 06	0 : 00 : 27	00 : 15 : 28	0 : 20 : 45
DIRECT	0 : 0 : 08	0 : 1 : 23	0 : 13 : 57	16 : 47 : 13	34 : 45 : 32
Firefly	0 : 3 : 54	0 : 6 : 23	0 : 06 : 37	00 : 10 : 42	0 : 08 : 27

CPU time (hh:mm:ss)

Methods	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
SymDIRECT	869	3 091	7 513	108 773	214 341
DIRECT	5 097	50 861	160 189	1 142 959	2 012 589
Firefly	610 380	939 000	1 011 120	1 611 160	1 279 980

Number of function evaluations

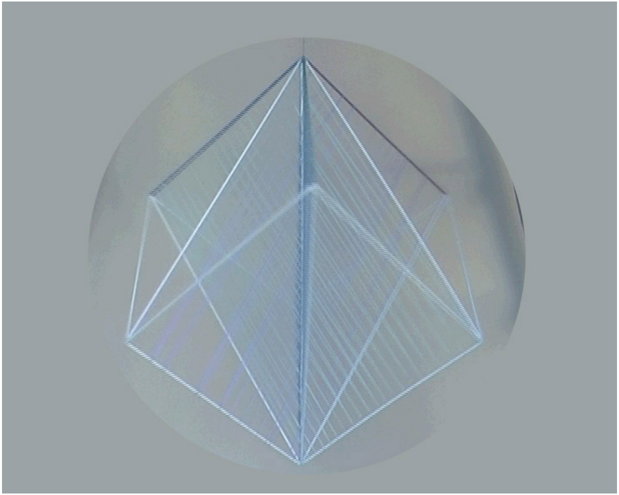
J. M.Gablonsky, *DIRECT Version 2.0*, Center for Research in Scientific Computation, North Carolina State University, 2001

X. S.Yang, *Firefly Algorithms for Multimodal Optimization*, Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications, 2009

Generalization and other possibilities

Dividing the triangle

Data set $\mathcal{A}$ with elements which can have several features



Searching for an approximate globally optimal partition

$\mathcal{A} = \{a^i \in \mathbb{R}^n \colon i = 1, \ldots, m\} \subset [\alpha, \beta] \subset \mathbb{R}^n$ (data points set)

$[\alpha, \beta] = \{x \in \mathbb{R}^n \colon \alpha_i \leq x_i \leq \beta_i\}$

$$F_k \colon \underbrace{\mathbb{R}^n \times \cdots \times \mathbb{R}^n}_{k} \to \mathbb{R}_+, \quad F_k(c_1, \ldots, c_k) = \sum_{i=1}^{m} w_i \min\{d(c_1, a^i), \ldots, d(c_k, a^i)\}$$

For $k = 1$, the function F_1 attains its global minimum at the point $c_1^\star \in [\alpha, \beta]$

A.Likas, N.Vlassis, J.J.Verbeek, *The global k-means clustering algorithm*, Pattern Recognition, 2003

A. M.Bagirov, J.Ugon, *An algorithm for minimizing clustering functions*, Optimization, 2005

A. M.Bagirov, *Modified global k-means algorithm for minimum sum-of-squares clustering problems*, Pattern Recognition, 2008

A. M.Bagirov, J.Ugon, D.Webb, *Fast modified global k-means algorithm for incremental cluster construction*, Pattern Recognition, 2011

Incremental methods

Let $\hat{c}_1, \ldots, \hat{c}_{k-1}$ be the centers obtained in the previous step as an approximation of a global minimizer of the function F_{k-1} and let

$$F_{k-1}(\hat{c}_1, \ldots, \hat{c}_{k-1}) = \sum_{i=1}^{m} w_i \hat{\delta}_{k-1}^i, \quad \hat{\delta}_{k-1}^i = \min\{d(\hat{c}_1, a^i), \ldots, d(\hat{c}_{k-1}, a^i)\},$$

$$\Phi_k(c) := F_k(\hat{c}_1, \ldots, \hat{c}_{k-1}, c) = \sum_{i=1}^{m} w_i \min\{\hat{\delta}_{k-1}^i, d(c, a^i)\}.$$

$$\hat{c}_k \in \operatorname*{argmin}_{c \in [\alpha, \beta]} \Phi_k(c)$$

Global k-means Algorithm

The first possibility

$$\operatorname*{argmin}_{a^j \in \mathcal{A}} \Delta(a^j), \quad \Delta(a^j) := F_{k-1}(\hat{c}_1, \ldots, \hat{c}_{k-1}) - \Phi(a^j)$$

The second possibility: Global k-means Algorithm

$$\operatorname*{argmin}_{a^j \in \mathcal{A}} \Delta(a^j) = \operatorname*{argmax}_{a^j \in \mathcal{A}} \sum_{i=1}^{m} w_i \max\{0, \hat{\delta}^i_{k-1} - d(a^j, a^i)\}$$

The third possibility: Bagirov, 2005

$$\hat{c}_k \in \operatorname*{argmin}_{c \in [\alpha, \beta]} \Phi_k(c) \qquad \text{(Discrete gradient method)}$$

Algorithm 1

Step 1: Let $\hat{c}_1, \ldots, \hat{c}_{k-1}$ be the centers obtained in the previous step as an approximation of a global minimizer of the function F_{k-1} and let

$$F_{k-1}(\hat{c}_1, \ldots, \hat{c}_{k-1}) = \sum_{i=1}^{m} w_i \hat{\delta}^i_{k-1}, \quad \hat{\delta}^i_{k-1} = \min\{d(\hat{c}_1, a^i), \ldots, d(\hat{c}_{k-1}, a^i)\},$$

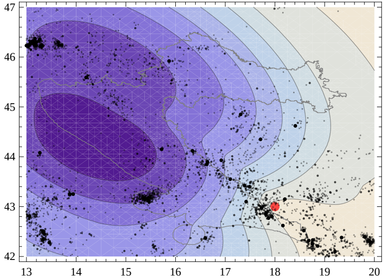
$$\Phi_k(c) := F_k(\hat{c}_1, \ldots, \hat{c}_{k-1}, c) = \sum_{i=1}^{m} w_i \min\{\hat{\delta}^i_{k-1}, d(c, a^i)\}.$$

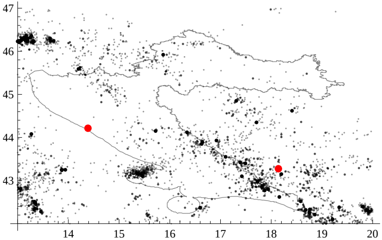
Step 2: By using the `DIRECT` algorithm for global optimization determine

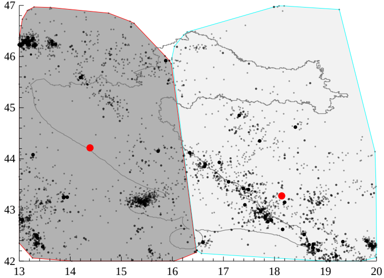
$$\hat{c}_k \in \operatorname*{argmin}_{c \in [\alpha, \beta]} \Phi_k(c)$$

Step 3: By using the k-means algorithm with initial approximations $\hat{c}_1, \ldots, \hat{c}_k$ determine new centers $c_1^\star, \ldots, c_k^\star$.

Example 1: Detection of spatial locations of seismic activity centers





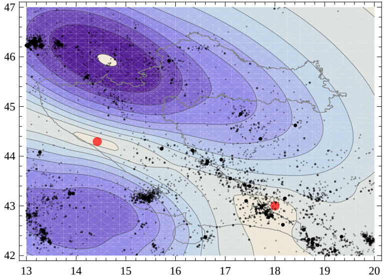


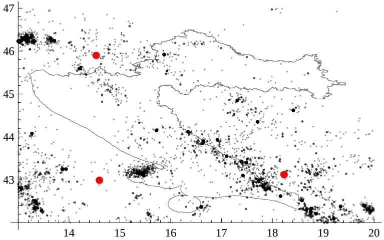
Mahalanobis distance–like function

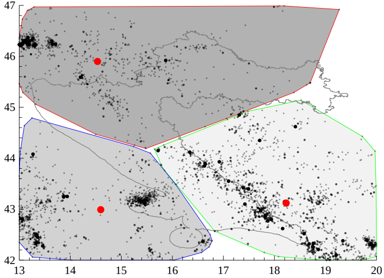
$d_M : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+, \quad d_M(x, y) = (x - y)\Sigma^{-1}(x - y)^T$

$$\Sigma = \begin{bmatrix} 4.6646 & -1.3706 \\ -1.3706 & 1.8571 \end{bmatrix}$$ (covariance matrix)

$\lambda_1 \approx 5.2, \ \lambda_2 \approx 1.3$







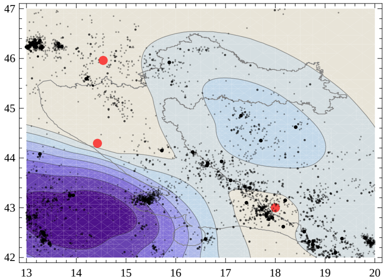
Mahalanobis distance–like function

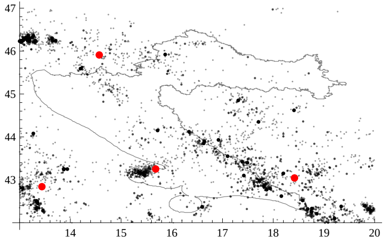
$d_M : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+, \quad d_M(x,y) = (x-y)\Sigma^{-1}(x-y)^T$

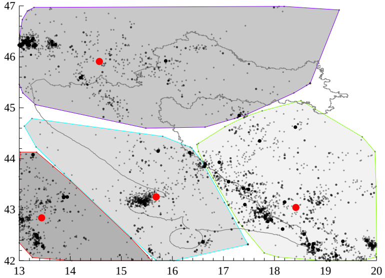
$$\Sigma = \begin{bmatrix} 4.6646 & -1.3706 \\ -1.3706 & 1.8571 \end{bmatrix}$$ (covariance matrix)

$\lambda_1 \approx 5.2, \ \lambda_2 \approx 1.3$

Example 1: Detection of spatial locations of seismic activity centers





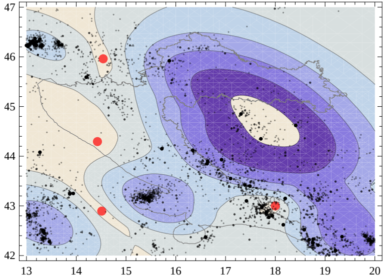


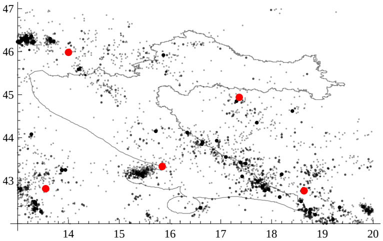
Mahalanobis distance–like function

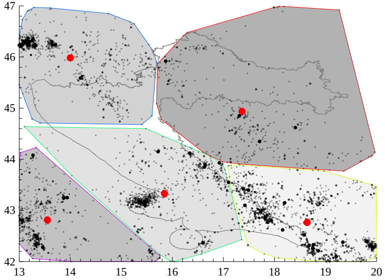
$d_M : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+, \quad d_M(x, y) = (x - y)\Sigma^{-1}(x - y)^T$

$$\Sigma = \begin{bmatrix} 4.6646 & -1.3706 \\ -1.3706 & 1.8571 \end{bmatrix}$$ (covariance matrix)

$\lambda_1 \approx 5.2, \ \lambda_2 \approx 1.3$







Mahalanobis distance–like function

$d_M : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+, \quad d_M(x,y) = (x-y)\Sigma^{-1}(x-y)^T$

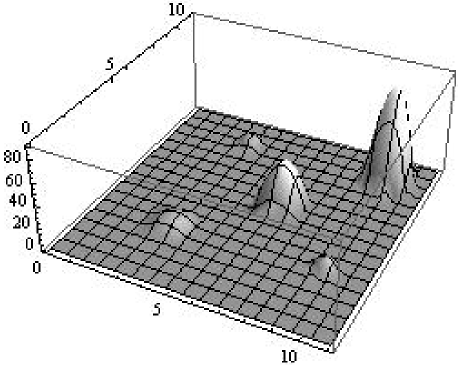
$$\Sigma = \begin{bmatrix} 4.6646 & -1.3706 \\ -1.3706 & 1.8571 \end{bmatrix}$$ (covariance matrix)

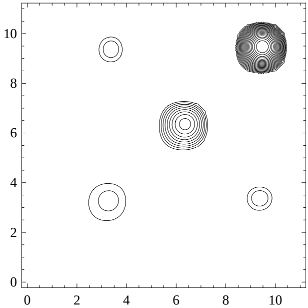
$\lambda_1 \approx 5.2, \ \lambda_2 \approx 1.3$

Example 2: Searching for all local maxima

$$\varphi\colon [0,11]^2 \to \mathbb{R}, \quad \varphi(x_1,x_2) = -\tfrac{1}{5}(x_1^2+x_2^2) + 2x_1x_2\cos x_1\cos x_2$$

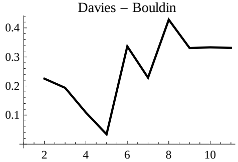
$$f(x_1,x_2) = \begin{cases} \varphi(x_1,x_2), & \text{if } \varphi(x_1,x_2) \geq 0, \\ 0, & \text{if } \varphi(x_1,x_2) < 0. \end{cases}$$

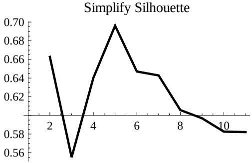




Example 2: Searching for all local maxima

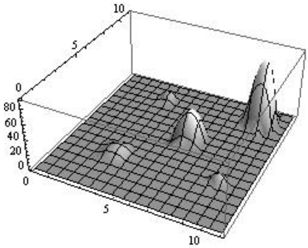
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 : x_i, y_j \in \{\tfrac{11}{m} k : k = 0, 1, \dots, m\}\}, \quad m = 100$$

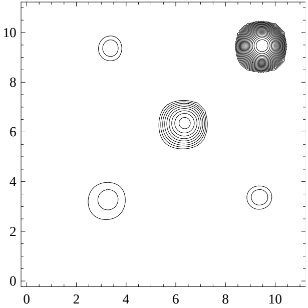




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

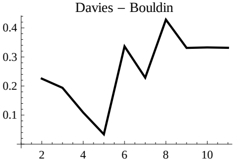
$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$

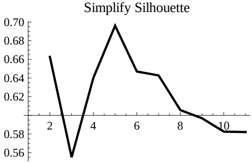




Example 2: Searching for all local maxima

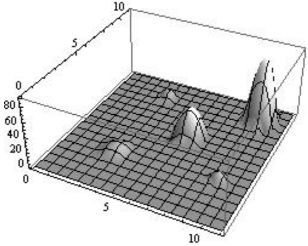
$$\mathcal{A} = \{(x_i, y_j) \in [0,11]^2 \colon x_i, y_j \in \{\tfrac{11}{m}k \colon k = 0,1,\ldots,m\}\}, \qquad m = 100$$

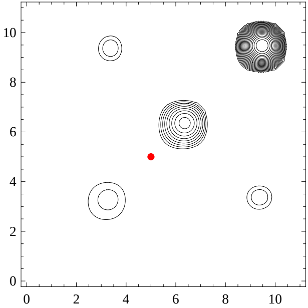




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

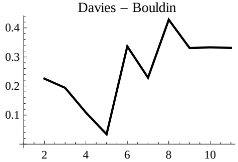
$$w_{ij} = \frac{1}{\nu-\mu} f(x_i, y_j) - \frac{\mu}{\nu-\mu}$$

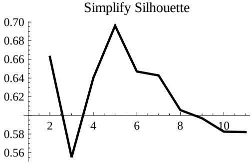




Example 2: Searching for all local maxima

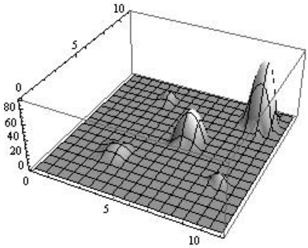
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 \colon x_i, y_j \in \{\tfrac{11}{m} k \colon k = 0, 1, \ldots, m\}\}, \qquad m = 100$$

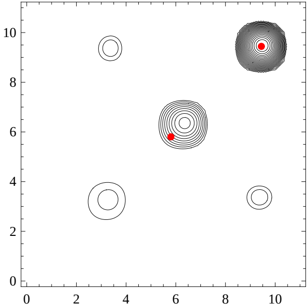




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

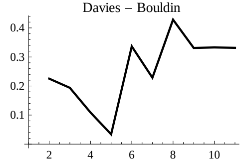
$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$

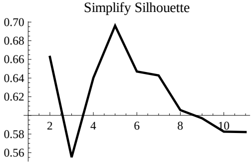




Example 2: Searching for all local maxima

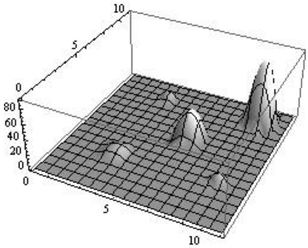
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 \colon x_i, y_j \in \{\tfrac{11}{m}k \colon k = 0, 1, \dots, m\}\}, \qquad m = 100$$

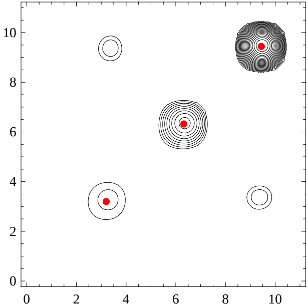




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

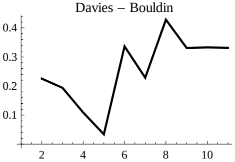
$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$

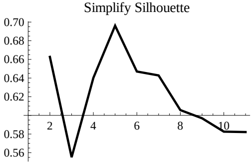




Example 2: Searching for all local maxima

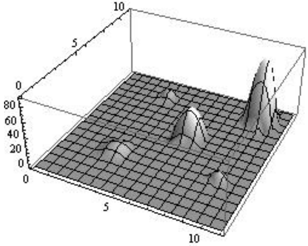
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 \colon x_i, y_j \in \{\tfrac{11}{m} k \colon k = 0, 1, \ldots, m\}\}, \qquad m = 100$$

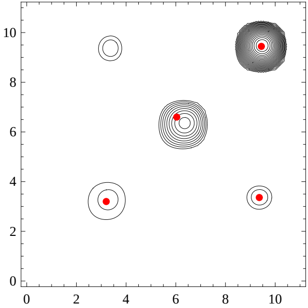




$$\mu = \min_{i,j} f(x_i, y_j), \ \nu = \max_{i,j} f(x_i, y_j)$$

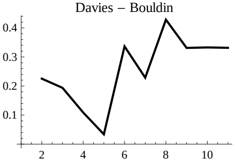
$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$

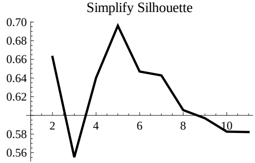




Example 2: Searching for all local maxima

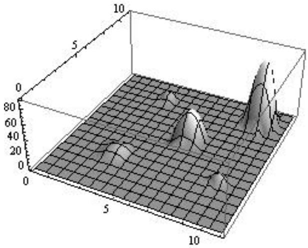
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 \colon x_i, y_j \in \{\tfrac{11}{m} k \colon k = 0, 1, \ldots, m\}\}, \qquad m = 100$$

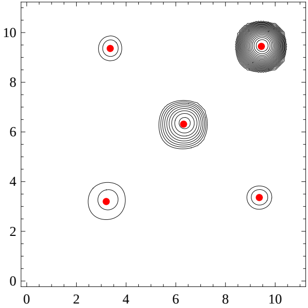




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

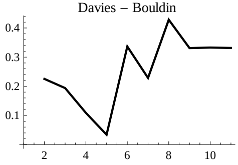
$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$

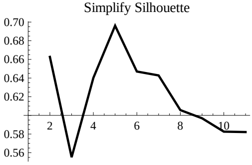




Example 2: Searching for all local maxima

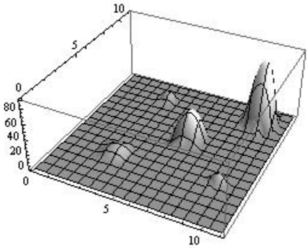
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 \colon x_i, y_j \in \{\tfrac{11}{m} k \colon k = 0, 1, \ldots, m\}\}, \qquad m = 100$$

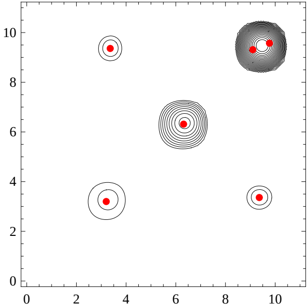




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

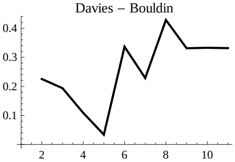
$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$

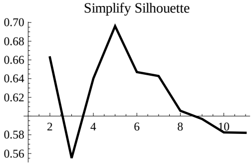




Example 2: Searching for all local maxima

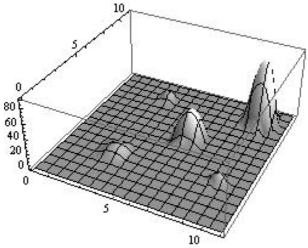
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 \colon x_i, y_j \in \{\tfrac{11}{m} k \colon k = 0, 1, \ldots, m\}\}, \qquad m = 100$$

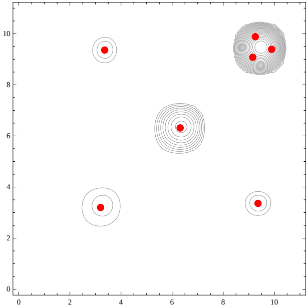




$$\mu = \min_{i,j} f(x_i, y_j), \ \nu = \max_{i,j} f(x_i, y_j)$$

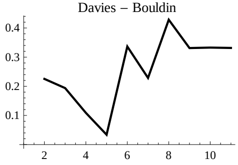
$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$

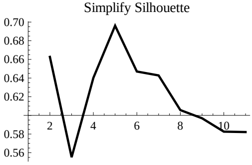




Example 2: Searching for all local maxima

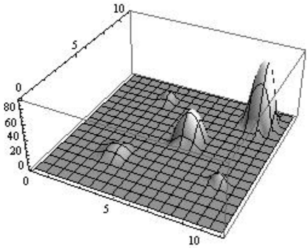
$$\mathcal{A} = \{(x_i, y_j) \in [0,11]^2 \colon x_i, y_j \in \{\tfrac{11}{m}k \colon k = 0,1,\ldots,m\}\}, \qquad m = 100$$

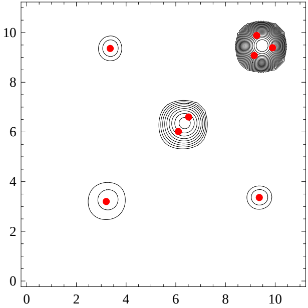




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

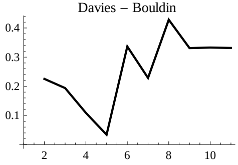
$$w_{ij} = \frac{1}{\nu-\mu} f(x_i, y_j) - \frac{\mu}{\nu-\mu}$$

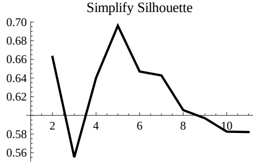




Example 2: Searching for all local maxima

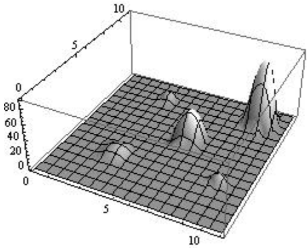
$$\mathcal{A} = \{(x_i, y_j) \in [0,11]^2 \colon x_i, y_j \in \{\tfrac{11}{m}k \colon k = 0,1,\ldots,m\}\}, \qquad m = 100$$

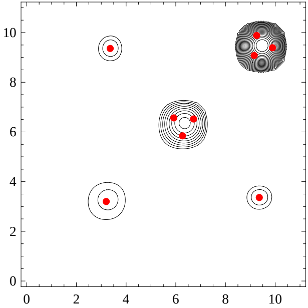




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

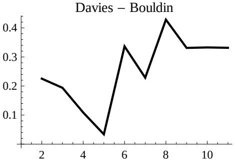
$$w_{ij} = \frac{1}{\nu-\mu} f(x_i, y_j) - \frac{\mu}{\nu-\mu}$$

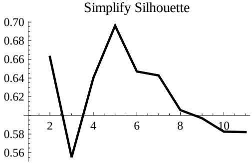




Example 2: Searching for all local maxima

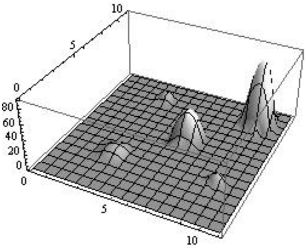
$$\mathcal{A} = \{(x_i, y_j) \in [0, 11]^2 \colon x_i, y_j \in \{\tfrac{11}{m} k \colon k = 0, 1, \ldots, m\}\}, \qquad m = 100$$

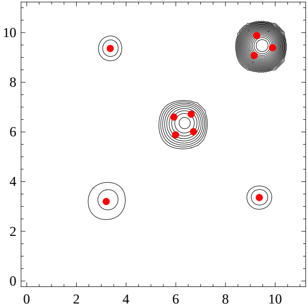




$$\mu = \min_{i,j} f(x_i, y_j), \; \nu = \max_{i,j} f(x_i, y_j)$$

$$w_{ij} = \frac{1}{\nu - \mu} f(x_i, y_j) - \frac{\mu}{\nu - \mu}$$





Line detection problem

$ax + by - c = 0, \quad a^2 + b^2 = 1, \quad c \geq 0$ (line)

$\mathcal{A} = \{T_i = (x_i, y_i) \in \mathbb{R}^2 \colon i \in I\} \subset [x_{min}, x_{max}] \times [y_{min}, y_{max}]$
(data point set $\mathcal{A}$ generated by k lines $p_1, \ldots, p_k$)

Reconstruction:

- Hough transformation
- filter-based methods
- linear regression methods
- cluster-based methods

Ivan Vidović, Rudolf Scitovski, Ivan Vazler, *Center-based clustering for line detection*, Image and Vision Computing (submitted: May 08, 2013)

Line detection problem

$$d(p_j(a_j, b_j, c_j), T(\xi, \eta)) = (a_j\xi + b_j\eta - c_j)^2$$ (distance from the point to the line)

$$\hat{p}_j(\hat{a}_j, \hat{b}_j, \hat{c}_j), \quad (\hat{a}_j, \hat{b}_j, \hat{c}_j) = \operatorname*{argmin}_{a_j, b_j, c_j \in \mathbb{R}} \sum_{T \in \pi_j} d(p_j(a_j, b_j, c_j), T)$$ (center-line)

$$\pi(p_j) = \{T \in \mathcal{A} : d(p_j, T) \le d(p_s, T), \forall s = 1, \ldots, k\}$$ (cluster)

k-means problem (global optimization problem):

$$\operatorname*{argmin}_{\Pi \in \mathcal{P}(\mathcal{A}; m, k)} \mathcal{F}(\Pi), \qquad \mathcal{F}(\Pi) = \sum_{j=1}^{k} \sum_{T \in \pi_j} d(\hat{p}_j(\hat{a}_j, \hat{b}_j, \hat{c}_j), T)$$

$$\operatorname*{argmin}_{a, b, c \in \mathbb{R}^k} F(a, b, c), \qquad F(a, b, c) = \sum_{T \in \mathcal{A}} \min_{1 \le s \le k} d(p_s(a_s, b_s, c_s), T)$$

Adjustment of incremental methods

$\hat{p}_1(\hat{a}_1, \hat{b}_1, \hat{c}_1)$ (the best TLS line)

$\hat{p}_2(\hat{a}_2, \hat{b}_2, \hat{c}_2)$:

$$\underset{\alpha,\beta,\gamma\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{m} \min\{d(\hat{p}_1(\hat{a}_1, \hat{b}_1, \hat{c}_1), T_i), d(p(\alpha,\beta,\gamma), T_i)\},$$

$$\alpha^2 + \beta^2 = 1,\ \gamma \geq 0$$

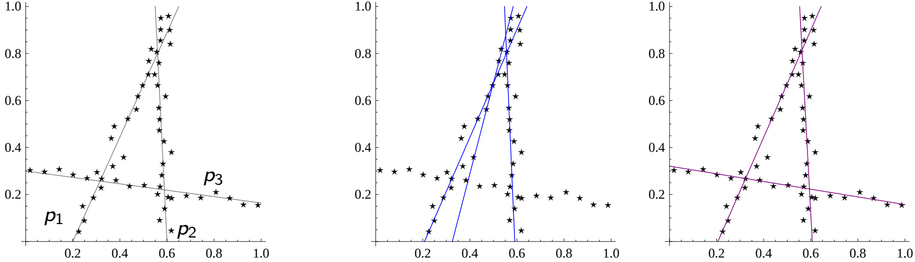
$\hat{p}_1, \ldots, \hat{p}_{k-1}$ (already known lines)

$\hat{p}_k(\hat{a}_k, \hat{b}_k, \hat{c}_k)$: $\underset{\alpha,\beta,\gamma\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{m} \min\{\hat{\delta}^i_{k-1}, d(p(\alpha,\beta,\gamma), T_i)\},$

$$\hat{\delta}^i_{k-1} = \min\{d(\hat{p}_1, T_i), \ldots, d(\hat{p}_{k-1}, T_i)\}$$

Line detection by HTA and IMLD: an exaple

(a) Generated data points (b) HTA (c) IMLD



An application: two crop rows detection

Algorithm 1 (approximate globally optimal 2-partition)

1. Let $\mathcal{A} = \{T_i = (x_i, y_i) \colon i = 1, \ldots, m\}$;
2. Determine the best TLS line p_0;
3. By using line p_0 divide the set $\mathcal{A}$ into two disjoint subsets such that $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$;
4. **for** $j = 1, 2$ **do**
5. For the data point set $\mathcal{A}_j$ determine the best TLS line $\hat{p}_j$;
6. **end for**
7. Apply the k-means algorithm with initial center-lines $\hat{p}_1, \hat{p}_2$

	Complete sowing			Incomplete sowing		
Algorithm	HTA	IMLD	Algorithm 1	HTA	IMLD	Algorithm 1
$\hat{d}_H < 0.05$	31	84	99	30	69	98
$0.05 < \hat{d}_H \leq 0.1$	12	3	1	14	4	2
$0.1 < \hat{d}_H \leq 0.2$	9	12	-	17	27	-
$0.2 < \hat{d}_H \leq 0.5$	48	1	-	39	-	-
CPU-time (sec)	1.25	.24	.04	1.24	.23	.04

Table: Testing of the methods for solving the problem of detecting two crop rows with $\sigma^2 = 0.02$

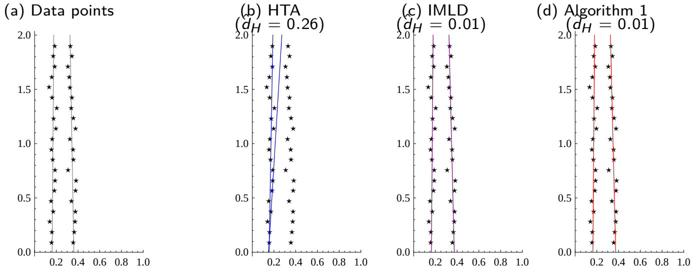


Figure: An example of a numerical test of methods for solving the problem of detecting two crop rows for data with variance $\sigma^2 = 0.02$ that simulate complete sowing

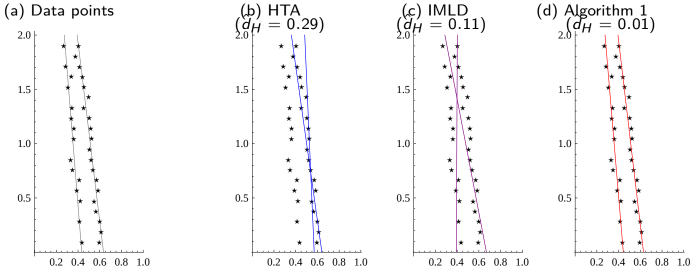


Figure: An example of a numerical test of methods for solving the problem of detecting two crop rows for data with variance $\sigma^2 = 0.02$ that simulate incomplete sowing

An application: three crop rows detection

Algorithm 2 (approximate globally optimal 3-partition)

1. Let $\mathcal{A} = \{T_i = (x_i, y_i) : i = 1, \ldots, m\}$;
2. For the data point set $\mathcal{A}$ determine the best TLS line $\hat{p}_0$;
3. By using line $\hat{p}_0$ divide the set $\mathcal{A}$ into two disjoint subsets such that $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$;
4. **for** $j = 1, 2$ **do**
5. By using the `DIRECT` method solve the following GOP
$$(\zeta_1, \zeta_2, \zeta_3) = \operatorname*{argmin}_{\alpha,\beta,\gamma\in\mathbb{R}} \sum_{T\in\mathcal{A}_j} \min\{d(\hat{p}_0, T), d(p(\alpha,\beta,\gamma), T)\},$$ and set $\hat{p}_j := p(\zeta_1, \zeta_2, \zeta_3)$
6. **end for**
7. Apply the k-means algorithm with initial center-lines $\hat{p}_0, \hat{p}_1, \hat{p}_2$

Algorithm	Complete sowing HTA	Complete sowing IMLD	Complete sowing Algorithm 2	Incomplete sowing HTA	Incomplete sowing IMLD	Incomplete sowing Algorithm 2
$\hat{d}_H < 0.05$	18	74	100	8	67	98
$0.05 < \hat{d}_H \leq 0.1$	5	2	-	5	3	2
$0.1 < \hat{d}_H \leq 0.15$	2	10	-	6	11	-
$0.15 < \hat{d}_H \leq 0.20$	6	14	-	4	19	-
CPU-time (sec)	1.25	.46	.36	1.25	.45	.36

Table: Testing the methods for solving the problem of detecting three crop rows with $\sigma^2 = 0.02$

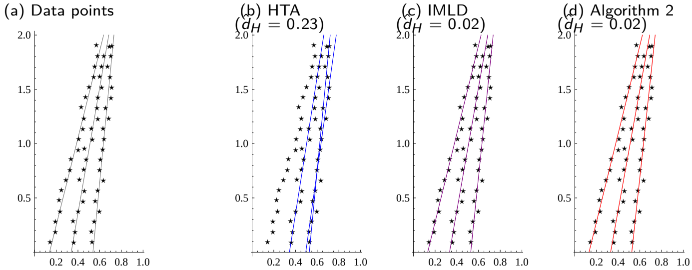


Figure: An example of a numerical test of methods for solving the three crop row detection problem for data with variance $\sigma^2 = 0.02$ that simulate complete sowing

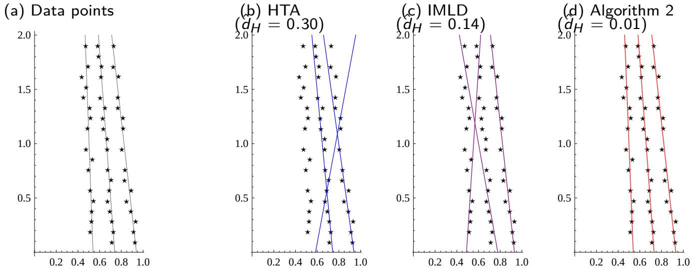


Figure: An example of a numerical test of methods for solving the problem of detecting three crop rows for data with variance $\sigma^2 = 0.02$ that simulate incomplete sowing