Spatial Patterns Underlying Facility Location Problems

Visualization and Classification

Hannah Bakker
hannah.bakker@kit.edu
Stefan Nickel
stefan.nickel@kit.edu
Karlsruhe Institute of Technology, Institute for Operations Research

I. Motivation

Despite facility location problems being perfectly suited for effective visualizations of instances and solutions, relatively few works make use of this option. This is, not the least, because many of the benchmark instances that are frequently used do not provide locations in terms of coordinates that would allow for visualization in a two-dimensional space. Instead, these instances only provide the input that is required for the mathematical programming formulation, the transport cost matrix $\{D\}_{I\times J}$ with I being the set of candidate locations, J the set of customers, and d_{ij} the unit transportation costs.

In the following, we briefly outline how, based on this sparse information, one can retrieve approximate coordinates in two dimensions for all customers and candidates that allow for an effective visualization of the underlying spatial relationships. In addition, we provide code for classifying the estimated spatial point pattern. We see the following two use cases for our code:

1) Apprehend and visualize spatial relationships between candidates and customers in benchmark instances for which exact locations are not available.
2) Apprehend implied spatial relationships in real-world instances for which exact locations are known, yet, distances other than Euclidean distances are used to derive $\{D\}_{I\times J}$, and thus, the visualization on a map may be misleading.

II. Coordinates via Multi-dimensional scaling

We suggest retrieving coordinates via multi-dimensional scaling (MDS), a dimensionality reduction technique. To the best of our knowledge, Guazelli et al. [1] first used MDS to approximate coordinates of location instances in a plane. MDS is a dimensionality reduction technique commonly used in machine learning. It analyses the similarity or dissimilarity of data in the original high-dimensional space and then attempts to model it in a lower-dimensional space. We implement MDS using the `sklearn.manifold` package, taking the unit transport costs $\{D\}_{I\times J}$ as input features. As there are usually more customers than facilities, we consider each customer as an observation and their distances to the respective facilities as features. This leaves us with a $J \times I$-dimensional feature matrix. Distances among facilities are usually not provided in the transport cost matrix of benchmark instances, so they are estimated according to Algorithm 1. For each pair of facilities i and i', we successively derive upper and lower bounds (UB and LB) on their distance with the help of triangular inequalities. We then approximate their distance as the average between the upper and lower bound. The parameter p determines how many customers closest to facility i are considered to improve UB and LB. The resulting extended transport cost matrix has dimensions $((|J| + |I|) \times |I|)$. With MDS it is projected to a $(|J| + |I|) \times 2$-dimensional matrix. The two columns serve as a basis for the (x, y)-coordinates in a plane.

Algorithm 1 Estimate distances between two facilities i, i'

Require: i, i', d_{ij} and $d_{i'j}$, for $j \in J$, p
Ensure: $\tilde{d}_{ii'}$
1: $UB \leftarrow \infty$, $LB \leftarrow 0$
2: Sort d_{ij} such that $d^1_{ij} \leq d^2_{ij} \leq \ldots d^k_{ij}$.
3: **for** $k \in 1 \ldots, p$ **do**
4: $\quad UB \leftarrow \min\{UB, d^k_{ij} + d_{i'j}\}$,
5: $\quad LB \leftarrow \max\{LB, |d^k_{ij} - d_{i'j}|\}$.
6: **end for**
7: $\tilde{d}_{ii'} \leftarrow (UB + LB)/2$

III. Classification of spatial point patterns

Taking the estimated coordinates of candidates and customers as input, we classify the spatial point pattern underlying a problem instance as either “clustered”, “random”, or “even” with the help of a two-sided t-test on the average nearest neighbor distance. The test measures the degree to which the average distance of any point to its nearest neighbor differs from the expected average distance under spatial randomness, a situation in which any point has had the same chance of occurring on any sub-area as any other point.

Let D_O denote the observed average nearest neighbor distance. D_O is compared to the expected nearest neighbor distance D_E in a randomly dispersed population of $I + J$ points with specified density ρ, known to be

$$D_E = \frac{1}{2\sqrt{\rho}} \tag{1}$$

with a standard deviation σ_{D_E} of

$$\sigma_{D_E} = \frac{0.26136}{\sqrt{N\rho}}. \quad (2)$$

The density ρ is obtained as the number of points divided by the area [3].

Given D_O, D_E, and σ_{D_E}, it is possible to perform a two-sided t-test on the hypothesis that D_O equals D_E and thereby the null hypothesis that the mean distance to the nearest neighbor in the observed distribution cannot be distinguished from the mean distance under spatial randomness. This test evaluates the test statistic

$$z = \frac{D_o - D_E}{\sigma_{D_E}} \quad (3)$$

based on the quantiles of the standard normal distribution. This implies that z-values whose absolute value exceeds 1.96 (2.58) allow one to reject the null hypothesis of spatial randomness with a confidence level of 95% (99%). A detailed description of the hypothesis test for complete spatial randomness can be found in Clark et al. [2].

The test allows one to gain information on which spatial distribution one observes instead of spatial randomness. If the test statistic is negative, then the observed mean distance to the nearest neighbor is smaller than expected under randomness, indicating clusters of points in certain sub-spaces. We classify these patterns as “clustered”. If the test statistic is positive, the distance is larger than expected, indicating a more evenly dispersed distribution of points. We classify these patterns as “even”. If the null hypothesis cannot be rejected, we classify the pattern as “random”.

Considering candidates and customers as data points, we use the coordinates obtained according to the previous section as a proxy for the location of candidates and customers in a hypothetical plane. We determine the Euclidean distances between all pairs of points and take the minimum distance per point as its estimated nearest neighbor distance. We take the area within the convex hull of the estimated points as a proxy for A.

IV. Spatial point patterns in benchmark instances

We approximated coordinates representative of the spatial relationships between candidates and customers. The resulting spatial point patterns were inferred with the t-test described above with a confidence level of 0.99 for several sets of benchmark instances for the CFLP. The results are summarized below.

TABLE I. Overview spatial point patterns underlying benchmark instances

Source	# instances	# clustered	# random	# even
Delmaire et al. [4]	57	-	-	57
Holmberg et al. [5]	71	9	37	25
Yang et al. [6]	20	6	14	-
Beasley [7]*	36	24	12	-
Beasley [7]**	12	-	12	-

(*) 36 small instances; (**) 12 large instances

TABLE II. Benchmark instances presented in Delmaire et al. [4]

subset	name	I	J	z-value	spatial point pattern
C1	p1	10	20	7.76	even
C1	p2	10	20	8.62	even
C1	p3	10	20	8.95	even
C1	p4	10	20	7.63	even
C1	p5	10	20	7.78	even
C1	p6	10	20	6.92	even
C2	p7	15	30	8.44	even
C2	p8	15	30	8.55	even
C2	p9	15	30	8.11	even
C2	p10	15	30	8.26	even
C2	p11	15	30	8.91	even
C2	p12	15	30	5.67	even
C2	p13	15	30	5.67	even
C2	p14	15	30	8.26	even
C2	p15	15	30	9.41	even
C2	p16	15	30	9.41	even
C2	p17	15	30	9.19	even
C3	p18	20	40	8.01	even
C3	p19	20	40	8.86	even
C3	p20	20	40	8.53	even
C3	p21	20	40	7.79	even
C3	p22	20	40	9.36	even
C3	p23	20	40	8.60	even
C3	p24	20	40	8.87	even
C3	p25	20	40	7.24	even
C4	p26	20	50	9.91	even
C4	p27	20	50	9.02	even
C4	p28	20	50	9.78	even
C4	p29	20	50	9.32	even
C4	p30	20	50	8.65	even
C4	p31	20	50	8.69	even
C4	p32	20	50	8.00	even
C4	p33	20	50	7.89	even
C5	p34	30	60	8.54	even
C5	p35	30	60	8.01	even
C5	p36	30	60	9.62	even
C5	p37	30	60	8.61	even
C5	p38	30	60	8.58	even
C5	p39	30	60	9.18	even
C5	p40	30	60	9.08	even
C5	p41	30	60	7.84	even
C6	p42	30	75	9.93	even
C6	p43	30	75	10.28	even
C6	p44	30	75	8.66	even
C6	p45	30	75	11.09	even
C6	p46	30	75	9.85	even
C6	p47	30	75	10.01	even
C6	p48	30	75	9.18	even
C6	p49	30	75	8.99	even
C7	p57	30	90	9.21	even
C7	p50	30	90	10.15	even
C7	p51	30	90	8.81	even
C7	p52	30	90	9.12	even
C7	p53	30	90	9.46	even
C7	p54	30	90	9.37	even
C7	p55	30	90	10.44	even
C7	p56	30	90	9.39	even

References

[1] Guazzelli, C. S., Cunha, C. B.: Exploring K- best solutions to enrich network design decision-making. *Omega* 78:139–164 (2018). doi:10.1016/j.omega.2017.06.009 1

[2] Clark, P. J., Evans, F. C.: Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations. *Ecology* 35(4):445–453 (1954). doi:10.2307/1931034 2

[3] Hertz, P.: Über den gegenseitigen durchschnittlichen Abstand von Punkten, die mit bekannter mittlerer Dichte im Raume angeordnet sind. *Mathematische Annalen* 67(3):1432–1807 (1909). doi:10.1007/BF01450410 2

[4] Delmaire, H. and Díaz, J. A. and Fernández, E. and Ortega, M.: Reactive Grasp And Tabu Search Based Heuristics For The Single Source Capacitated Plant Location Problem. *INFOR: Information Systems and Operational Research* 37(3):194—225 (1999). doi:10.1080/03155986.1999.11732381 2

[5] Holmberg, K. and Rönnqvist, M. and Yuan, D.: An exact algorithm

TABLE III. BENCHMARK INSTANCES PRESENTED IN HOLMBERG ET AL. [5]

name	I	J	z-value	spatial point pattern
p1	10	50	3.70	even
p2	10	50	3.70	even
p3	10	50	3.70	even
p4	10	50	3.70	even
p5	10	50	3.70	even
p6	10	50	3.70	even
p7	10	50	3.70	even
p8	10	50	3.70	even
p9	10	50	3.70	even
p10	10	50	3.70	even
p11	10	50	3.70	even
p12	10	50	3.70	even
p13	20	50	6.03	even
p14	20	50	6.03	even
p15	20	50	6.03	even
p16	20	50	6.03	even
p17	20	50	6.03	even
p18	20	50	6.03	even
p19	20	50	6.03	even
p20	20	50	6.03	even
p21	20	50	6.03	even
p22	20	50	6.03	even
p23	20	50	6.03	even
p24	20	50	6.03	even
p25	30	150	-0.42	random
p26	30	150	-0.42	random
p27	30	150	-0.42	random
p28	30	150	-0.42	random
p29	30	150	-0.42	random
p30	30	150	-0.42	random
p31	30	150	-0.42	random
p32	30	150	-0.42	random
p33	30	150	-0.42	random
p34	30	150	-0.42	random
p35	30	150	-0.42	random
p36	30	150	-0.42	random
p37	30	150	-0.42	random
p38	30	150	-0.42	random
p39	30	150	-0.42	random
p40	30	150	-0.42	random
p41	10	90	-2.80	clustered
p42	20	80	-3.80	clustered
p43	30	70	-2.80	clustered
p44	10	90	-1.63	random
p45	20	80	1.42	random
p46	30	70	0.92	random
p47	10	90	0.98	random
p48	20	80	0.85	random
p49	30	70	2.76	even
p50	10	100	-2.93	clustered
p51	20	100	-4.18	clustered
p52	10	100	-3.27	clustered
p53	20	100	-4.48	clustered
p54	10	100	-1.79	random
p55	20	100	-4.50	clustered
p56	30	200	1.08	random
p57	30	200	1.08	random
p58	30	200	1.08	random
p59	30	200	1.08	random
p60	30	200	1.08	random
p61	30	200	1.08	random
p62	30	200	1.08	random
p63	30	200	1.08	random
p64	30	200	1.08	random
p65	30	200	1.08	random
p66	30	200	1.08	random
p67	30	200	-5.34	clustered
p68	30	200	1.08	random
p69	30	200	1.08	random
p70	30	200	1.08	random
p71	30	200	1.08	random

TABLE IV. BENCHMARK INSTANCES PRESENTED IN YANG ET AL. [6]

name	I	J	z-value	spatial point pattern
30-200-1	30	200	1.10	random
30-200-2	30	200	-0.68	random
30-200-3	30	200	-2.28	random
30-200-4	30	200	-1.70	random
30-200-5	30	200	0.02	random
60-200-1	60	200	-1.39	random
60-200-2	60	200	-1.87	random
60-200-3	60	200	-3.13	clustered
60-200-4	60	200	-1.39	random
60-200-5	60	200	-2.80	clustered
60-300-1	60	300	-1.24	random
60-300-2	60	300	-1.37	random
60-300-3	60	300	-1.40	random
60-300-4	60	300	-3.25	clustered
60-300-5	60	300	-2.14	random
80-400-1	80	400	-3.21	clustered
80-400-2	80	400	-1.70	random
80-400-3	80	400	-2.17	random
80-400-4	80	400	-2.98	clustered
80-400-5	80	400	-3.07	clustered

TABLE V. 36 SMALL INSTANCES PRESENTED IN BEASLEY [7]

name	I	J	z-value	spatial point pattern
cap41	16	50	-1.16	random
cap42	16	50	-1.16	random
cap43	16	50	-1.16	random
cap44	16	50	-1.16	random
cap61	16	50	-1.20	random
cap62	16	50	-1.20	random
cap63	16	50	-1.20	random
cap64	16	50	-1.20	random
cap71	16	50	-1.20	random
cap72	16	50	-1.20	random
cap73	16	50	-1.20	random
cap74	16	50	-1.20	random
cap81	25	50	-3.84	clustered
cap82	25	50	-3.84	clustered
cap83	25	50	-3.84	clustered
cap84	25	50	-3.84	clustered
cap91	25	50	-3.84	clustered
cap92	25	50	-3.84	clustered
cap93	25	50	-3.84	clustered
cap94	25	50	-3.84	clustered
cap101	25	50	-3.84	clustered
cap102	25	50	-3.84	clustered
cap103	25	50	-3.84	clustered
cap104	25	50	-3.84	clustered
cap111	50	50	-13.08	clustered
cap112	50	50	-13.08	clustered
cap113	50	50	-13.08	clustered
cap114	50	50	-13.08	clustered
cap121	50	50	-13.01	clustered
cap122	50	50	-13.01	clustered
cap123	50	50	-13.01	clustered
cap124	50	50	-13.01	clustered
cap131	50	50	-13.01	clustered
cap132	50	50	-13.01	clustered
cap133	50	50	-13.01	clustered
cap134	50	50	-13.01	clustered

TABLE VI. 12 LARGE INSTANCES PRESENTED IN BEASLEY [7]

name	I	J	z-value	spatial point pattern
capa1	100	1000	-2.52	random
capa2	100	1000	-2.52	random
capa3	100	1000	-2.52	random
capa4	100	1000	-2.52	random
capb1	100	1000	-1.75	random
capb2	100	1000	-1.75	random
capb3	100	1000	-1.75	random
capb4	100	1000	-1.75	random
capc1	100	1000	-1.56	random
capc2	100	1000	-1.56	random
capc3	100	1000	-1.56	random
capc4	100	1000	-1.56	random

for the capacitated facility location problems with single sourcing. *European Journal of Operational Research* 113(3):544-–559 (1999). doi:10.1016/s0377-2217(98)00008-3 2, 3

[6] Yang, Z. and Chu, F. and Chen, H.: A cut-and-solve based algorithm for the single-source capacitated facility location problem. *European Journal of Operational Research* 221(3):521—532 (2012). doi:10.1016/j.ejor.2012.03.047 2, 3

[7] Beasley, J. E.: OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society* 41(11):1069—1072 (1990). doi:10.1057/jors.1990.166 2, 3