sustainability

MDPI

Correction

Correction: An Efficient Grid-Based K-Prototypes Algorithm for Sustainable Decision Making Using Spatial Objects. Sustainability 2018, *10*, 2614

Hong-Jun Jang [1], Byoungwook Kim [2], Jongwan Kim [3] and Soon-Young Jung [1,*]

1 Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea; hongjunjang@korea.ac.kr
2 Department of Computer Engineering, Dongguk University, Gyeongju 38066, Korea; bwkim@dongguk.ac.kr
3 Smith Liberal Arts College, Sahmyook University, Seoul 01795, Korea; kimj@syu.ac.kr
* Correspondence: jsy@korea.ac.kr; Tel.: +82-2-3290-2394

Received: 20 March 2019; Accepted: 21 March 2019; Published: 25 March 2019

check for updates

The authors would like to make the following corrections to this paper [1]:

(1) We corrected the subscript error in Definition 2.

Replacing:

Definition 2. The minimum distance between a cell g_i and a cluster center c_j for numeric attributes, denoted $d_{min}(g_i, c_j)$, is:

$$d_{min}(g_i,\ c_j) = \sqrt{\sum_{i=1}^{m_r} |o_i - r_i|^2}, \tag{5}$$

where $r_i = \begin{cases} s_i & if\ o_i < s_i \\ t_i & if\ o_i > t_i \\ o_i & otherwise \end{cases}.$

with:

Definition 2. The minimum distance between a cell g_i and a cluster center c_j for numeric attributes, denoted $d_{min}(g_i, c_j)$, is:

$$d_{min}(g_i,\ c_j) = \sqrt{\sum_{i=1}^{m_r} |c_{ji} - r_i|^2}, \tag{5}$$

where $r_i = \begin{cases} s_i & if\ c_{ji} < s_i \\ t_i & if\ c_{ji} > t_i \\ o_i & otherwise \end{cases}.$

(2) We corrected the subscript error in Definition 3.

Replacing:

Definition 3. The maximum distance between a cell g_i and a cluster center c_j for numeric attributes, denoted $d_{max}(g_i, c_j)$, is:

$$d_{max}(g_i,\ c_j) = \sqrt{\sum_{i=1}^{m_r} |p_i - r_i|^2}, \tag{6}$$

where $r_i = \begin{cases} t_i, & p_i \le \frac{s_i+t_i}{2} \\ s_i, & otherwise \end{cases}$

with:

$$d_{max}(g_i,\ c_j) = \sqrt{\sum_{i=1}^{m_r} \left|c_{ji} - r_i\right|^2}, \tag{6}$$

where $r_i = \begin{cases} t_i, & c_{ji} \le \frac{s_i+t_i}{2} \\ s_i, & otherwise \end{cases}$

(3) We corrected the order of some commands in Algorithm 1.

Replacing:

C[ ]← Ø // *k* cluster centers
Randomly choosing *k* object, and assigning it to *C*.
while *IsConverged*() **do**
  dmin[], *dmax*[] ← *Calc*(*g*, *C*)
  dminmax ← *min*(*dmax*[])
  for each cell *g* **in** *G*

with:

C[ ]← Ø // *k* cluster centers
Randomly choosing *k* object, and assigning it to *C*.
while *IsConverged*() **do**
  for each cell *g* **in** *G*
    dmin[ ], *dmax*[ ] ← *Calc*(*g*, *C*)
    dminmax ← *min*(*dmax*[ ])

(4) We corrected the order of some commands in Algorithm 2.

Replacing:

C[*k*]← Ø // *k* cluster center
Randomly choosing *k* object, and assigning it to *C*.
while *IsConverged*() **do**
  dmin[], *dmax*[] ← *Calc*(*g*, *C*)
  dminmax ← *min*(*dmax*[])
  for each cell *g* **in** *G*

with:

C[ ]← Ø // *k* cluster centers
Randomly choosing *k* object, and assigning it to *C*.
while *IsConverged*() **do**
  for each cell *g* **in** *G*
    dmin[ ], *dmax*[ ] ← *Calc*(*g*, *C*)
    dminmax ← *min*(*dmax*[ ])

(5) We corrected the analysis of complexity.

Replacing:

However, our proposed algorithms based on heuristic techniques (KCP and KBP) can reduce the number of objects to be computed, $n' \le n$, and the number of dimensions to be computed, $d' \le d$, respectively. Therefore, the time complexities of our proposed algorithms are O($n'kd'i$), $n' \le n$ and $d' \le d$.

For space complexity, KCP requires $O(nd)$ to store the entire dataset, $O(gd)$ to store the start point vector S and the end point vector of each cell, $O(km_c)$ to store the frequency of categorical data in each cluster and $O(kd)$ to store cluster centers, where m_c the number of categorical attributes. Additionally, KBP requires $O(gm_c)$ to store the frequency of categorical data in each cell, where g is the number of cells. Therefore, the space complexities of KCP and KBP are $O(nd + gd + km_c + kd)$ and $O(nd + gd + km_c + kd + gm_c)$, respectively.

with:

However, our proposed algorithms based on heuristic techniques (KCP and KBP) can reduce the number of cluster centers to be computed, $k' \leq k$, and the number of dimensions to be computed, $d' \leq d$. Therefore, the time complexities of our proposed algorithms are $O(nk'd'i)$, $k' \leq k$, and $d' \leq d$.

For space complexity, KCP requires $O(nd)$ to store the entire dataset, $O(gm_r)$, where g is the number of cells, to store the start point vector S and the end point vector T of each cell, $O(ktm_c)$, where t is the number of categorical data, to store the frequency of categorical data in each cluster and $O(kd)$ to store cluster centers. KBP requires $O(gtm_c)$ to store the bitmap index in each cell, where g is the number of cells. Therefore, the space complexities of KCP and KBP are $O(nd + gm_r + ktm_c + kd)$ and $O(nd + gm_r + ktm_c + kd + gtm_c)$, respectively.

(6) We corrected the description of Figure 12.

Replacing:

Figure 12. Effect of the number of clusters (numeric data and categorical data are on uniform distribution).

with:

Figure 12. Effect of the number of clusters (numeric data are uniformly distributed and the distribution of categorical data is skewed).

The authors would like to apologize for any inconvenience caused to the readers by these changes. The manuscript will be updated and the original will remain available from the article webpage.

Reference

1. Jang, H.-J.; Kim, B.; Kim, J.; Jung, S.-Y. An Efficient Grid-Based K-Prototypes Algorithm for Sustainable Decision-Making on Spatial Objects. *Sustainability* **2018**, *10*, 2614. [CrossRef]