Unsupervised Learning: Similarities and distance functions for IoT data

G. Casolla, V. Schiano, S. Cuomo, F. Giampaolo, F. Piccialli

University of Naples Federico II

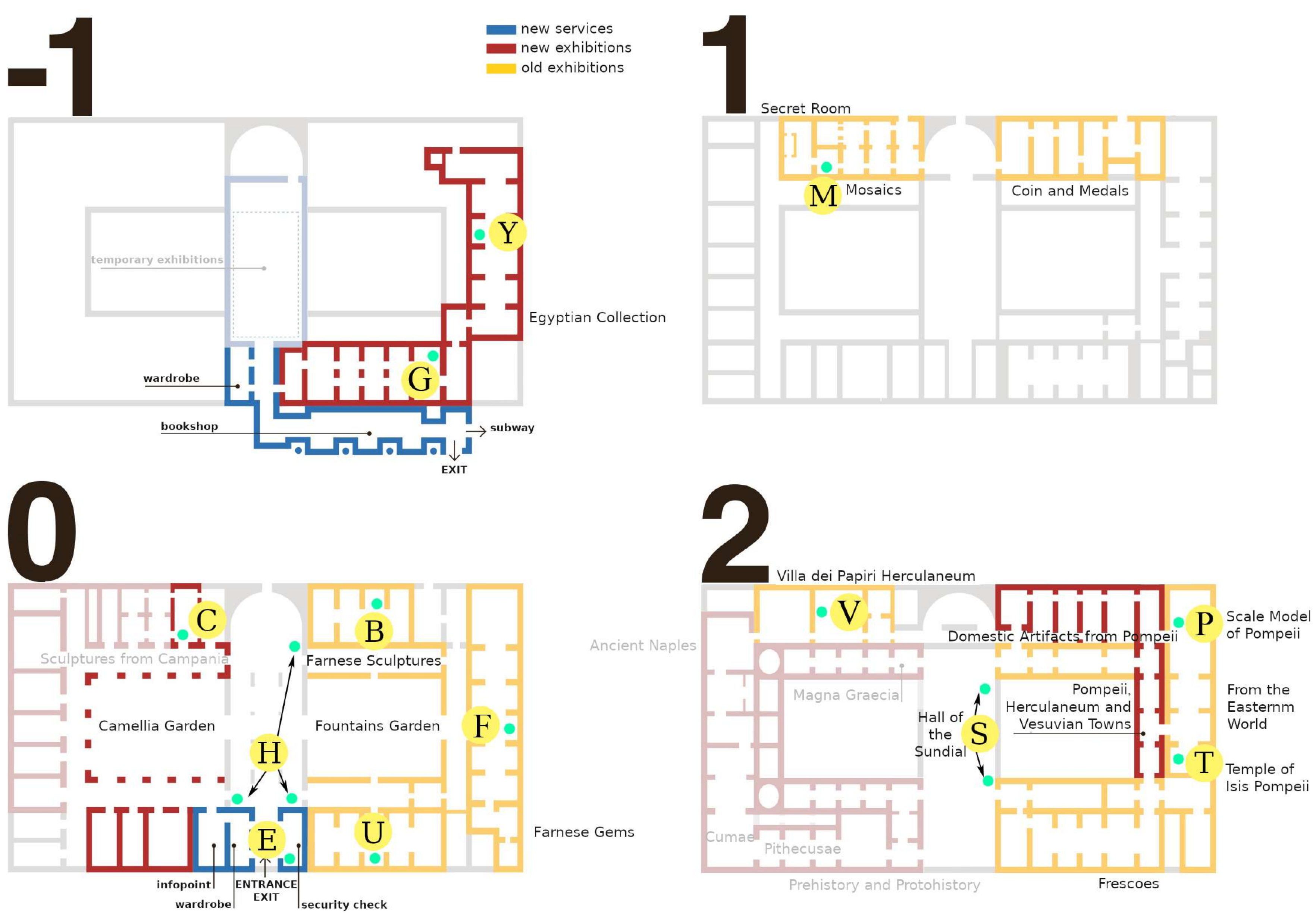
Mathematical mOdelling and Data AnaLysis research group
http://www.labdma.unina.it

Introduction (1)

In our research we studied visitors' behaviours inside one of the most important museum in Italy, the National Archaeological Museum of Naples, through a deployed IoT framework. It is composed by smart sensor boards with Bluetooth and Wi-Fi capabilities. The IoT system is able to track a visitor path by collecting his position and the related time-of-stay inside the museum rooms [1]. Our dataset relies on about 19 000 unique users behavioural data composed by two features: (i) the visiting *Path* (non-numerical data) and (ii) the *Time* spent (numerical data).

Classifying unstructured and unlabelled data is a key challenge dealing with three main research tasks:

- the study and selection of the appropriate similarity and distance functions,
- the selection of the number of clusters in which partitioning the data,
- the choice of the most suitable algorithm to achieve an accurate data clustering.



Theory (2)

The Clustering algorithms we considered are:

- Hierarchical Clustering
- K-Medoids (PAM)

A similarity function is a *tool* that gives the strength of the relationship between two or more data items.

To deal with the non-numerical *Path* feature of our dataset, that can be treated as a *string*, we have analyzed four strings similarity functions. A string can be defined as sequence of finite characters from a finite alphabet. A q-gram is a string with q consecutive characters. The q-grams from a string S are realized by collecting in sequences q characters from S and saving the appearing q-grams. To deal with the numerical *Time* feature we have pre-computed the distance matrix by using the well-known *Euclidean distance* and then merged with the other strings' distances with the formula defined as follows:

$$D_{sum} = \sum_{i=1}^{K} \omega_i D^{(i)}$$

where $D^{(i)}$ is a single distance matrix, K is the total number of distances that we want to merge and ω_i is the inverse of the maximum element of the i-th distance matrix $D^{(i)}$. It's shown that, to improve the performance of a classification, it is better to combine multiple distances than to use a single distance matrix [2].

References (5)

[1] F. Piccialli, Y. Yoshimura, P. Benedusi, C. Ratti, and S. Cuomo, Lessons learned from longitudinal modeling of mobile-equipped visitors in a complex museum. Neural Comp. and App., pp. 1-17, 2019

[2] A. Ibba, R. Duin, and W.-J. Lee, A study on combining sets of differently measured dissimilarities. ICPR'10, pp. 3360-3363, 2010

Similarities and distance functions (3)

(a) - **Cosine similarity**

$$s_{cos}(S,T;q) = \frac{v(S;q)\cdot v(T;q)}{\|v(S;q)\|_2\,\|v(T;q)\|_2}$$

(b) - **Jaccard similarity**

$$s_{\mathrm{Jac}}(S,T;q) = \frac{|Q(S;q)\cap Q(T;q)|}{|Q(S;q)\cup Q(T;q)|}$$

(c) - **Longest Common Substring**

$$d_{LCS}(S,T) = \frac{LCS(n,m)}{n+m}$$

where

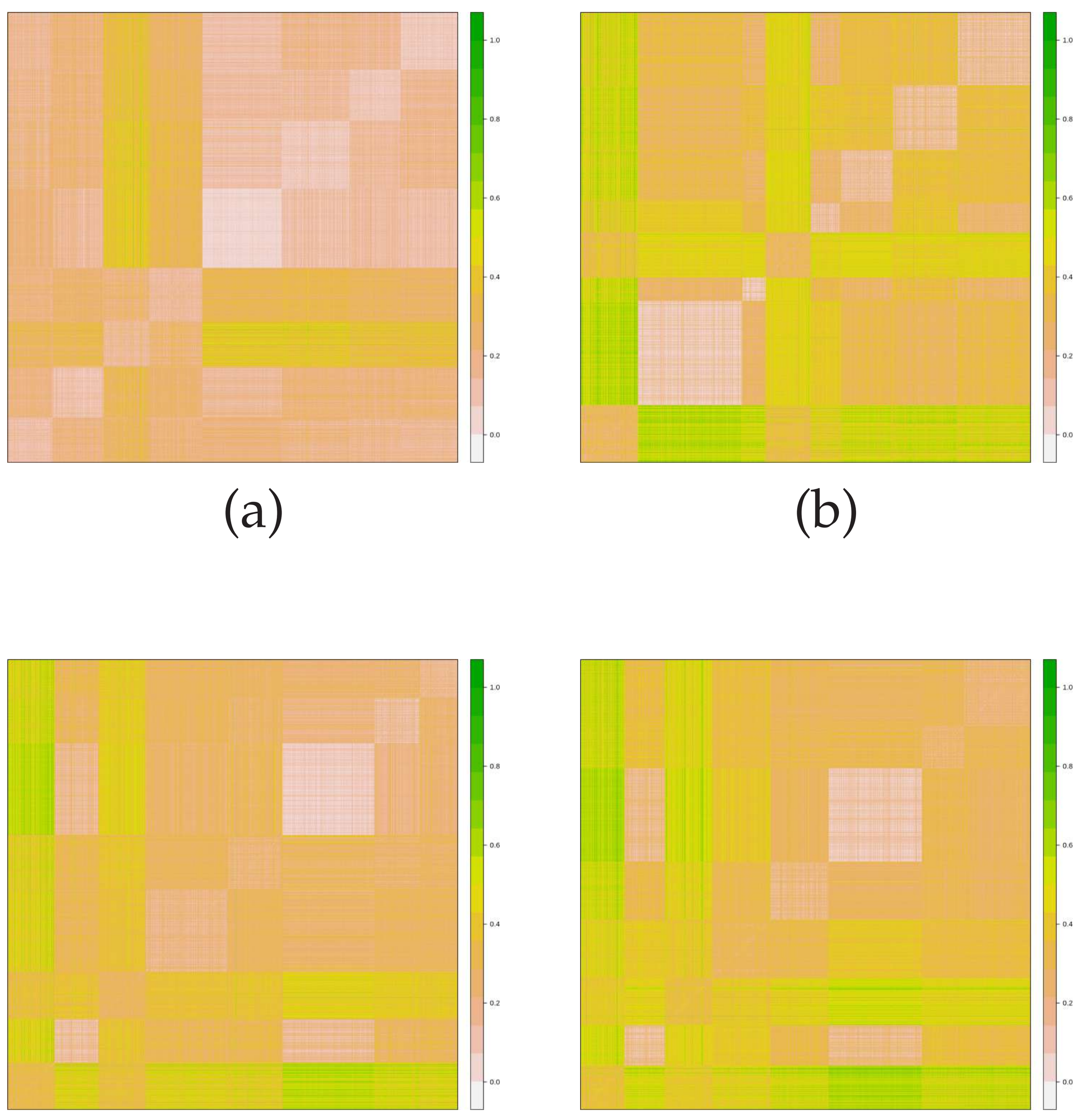
$$LCS(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j)=0,\\ LCS(i-1,j-1) & \text{if } S(i)=T(j),\\ 1+\min\{LCS(i-1,j), LCS(i,j-1)\} & \textit{otherwise} \end{cases}$$

(d) - **Generalized Levenshtein distance**

$$d_{Lv}(S,T) = \frac{Lv(n,m)}{\max\{n,m\}}$$

where

$$Lv(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j)=0,\\ \min\begin{cases} Lv(i-1,j)+1\\ Lv(i,j-1)+1\\ Lv(i-1,j-1)+1_{(S_i\neq T_j)}\end{cases} & \text{otherwise.}\end{cases}$$



(a) (b) (c) (d)

More in detail, the q-grams based distances are calculated from q-grams based similarities with the formula:

$$d_x(S,T) = 1 - s_x(S,T).$$

Comparison (4)

An ordered F-measure heatmap representing the comparison among all the experimented clustering methodologies.