CSD-CMAD: Coupling Similarity and Diversity for Clustering Multivariate Astrophysics Data

Xu Teng¹, Thomas Beckler¹, Bradley Gannon¹, Benjamin Huinker¹, Gabriel Huinker¹, Koushik Kumar¹, Christina Marquez¹, Jacob Spooner¹, Goce Trajcevski¹, Prabin Girī¹, Aaron Dotter², Jeff Andrews², Scott Coughlin², Ying Qin²,⁴, Juan Gabriel Serra², Nam Tran³, Jaime Roman Garja³, Konstantinos Kovlakas³, Emmanouil Zapartas³, Simone S. Bavera³, Devina Misra³, Tassos Fragos³

¹Department of ECE, Iowa State University, Ames, USA
²Department of Physics and Astronomy, Northwestern University, Evanston, USA
³Département d’Astronomie, Université de Genève, Versoix, Switzerland
⁴Department of Physics, Anhui Normal University, Wuhu, Anhui 241000, China

{tbeckler,bgannon,bhuinker,ghuinker,kkumar9,cmarquez,jspooner,gocet25}@iastate.edu, aaron.dotter@gmail.com
{jeffrey.andrews,s-coughlin,ying.qin,jgserra}@northwestern.edu, shp344@alumni.ku.dk

ABSTRACT

Traditionally, clustering of multivariate data aims at grouping objects described with multiple heterogeneous attributes based on a suitable similarity (conversely, distance) function. One of the main challenges is due to the fact that it is not straightforward to directly apply mathematical operations (e.g., sum, average) to the feature values, as they stem from heterogeneous contexts.

In this work we take the challenge a step further and tackle the problem of clustering multivariate datasets based on jointly considering: (a) similarity among a subset of the attributes; and (b) distance-based diversity among another subset of the attributes. Specifically, we focus on astrophysics data, where the snapshots of the stellar evolution for different stars contain over 40 distinct attributes corresponding to various physical and categorical (e.g., ‘black hole’) attributes. We present CSD-CAMD – a prototype system for Coupling Similarity and Diversity for Clustering Astrophysics Multivariate Datasets. It provides a flexibility for the users to select their preferred subsets of attributes; assign weight (to reflect their relative importance on the clustering); and select whether the impact should be in terms of proximity or distance. In addition, CSD-CAMD allows for selecting a clusting algorithm and enables visualization of the outcome of clustering.

CCS CONCEPTS
• Information systems → Data management systems; • Computing methodologies → Modeling and simulation.

KEYWORDS
Stellar Evolution, Clustering, Diversity

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGSPATIAL ’21, November 2–5, 2021, Beijing, China
© 2021 Copyright held by the owner/author(s).
https://doi.org/10.1145/3474717.3483989

1 INTRODUCTION AND MOTIVATION

Clustering is considered a canonical problem in data analysis and its main objective is to partition a given set into subsets, in a manner that will ensure that the objects in given subset are “more similar” (modulo certain distance function) to each other than to the objects in the rest of the subsets [10].

In its rich history, many variants to the basic approaches (e.g., K-means [6]; DBSCAN [3]) have been considered. Shortly after tackling the issues due to the high dimensionality of the data were recognized [5], researchers have addressed settings in which peculiar challenges arise due to the heterogeneity of the data. Namely, the elements of the set in question may not only have > 1 dimension, but those dimensions may be incompatible due to their physical properties (i.e., the domain of the attributes) [1, 2]. Another complementary extension was incorporating the diversity in the clustering process, which was more recently considered for improving ranking in recommendation systems [4].

Complementary to this, the scientific discipline of astrophysics studies the properties (and relationships) of various astronomical objects, such as stars, galaxies, interstellar medium, etc. [7]. The advances in optical and other sensing technologies along with the increase of computational and storage capabilities have enabled the creation of large repositories of observational data like, for example, Sloan Digital Sky Survey [1]. Often times, to evaluate the proposed theoretical models against observational data, astrophysicists resort

https://www.sdss.org
to simulation models, relying on tools such as MESA (Modules for Experiments in Stellar Astrophysics) [8].

Our POSYDON2 project also relies on MESA based data and has generated large datasets (5 ～ 10 GB) of stellar evolution. As part of the project, we have already developed a database for storing and querying the stars’ trajectories via web-based User Interface (UI). In addition to the sheer volume, the data (which can be perceived as multivariate time series) is also characterized by a large number of attributes pertaining to different physical phenomena, such as mass, luminosity, He-concentration, temperature, etc.

What motivates this work is the observation that, for the purpose of testing models and hypotheses, the domain scientists may be interested in clustering the instances of the stellar evolution trajectories in a manner that:

- Favor certain attributes more than the others (i.e., their impact on clustering is stronger).
- Insist on diversity for the values of other attributes (i.e., stars within a particular cluster should differ in each of those attributes by a predefined threshold).

As an example, one may want to cluster all the stars that end up in black holes with a stronger influence of mass and temperature (than the other attributes) and with at least 30% of a difference between the corresponding metallicity values.

Towards that end, the main contribution of the CSD-CMAD system prototype is that it enables scientists to focus on a targeted dataset and, once uploaded in the database, provide them with the opportunity to select the corresponding parameters for both classes of preferred attributes (i.e., stronger proximity as well as diversity via separability) – and visualize the outcome of applying a clustering algorithm. In the rest of this paper, in Section 2 we describe the basic architecture of CSD-CMAD and the way we modify the traditionally used distance functions to cater to the two complementary requests for the subsets of attributes. Section 3 lists the steps of the actual demo scenario that the attendees will be able to experience, and Section 4 offers concluding remarks.

2 SYSTEM ASPECTS
We now describe the system architecture of CSD-CMAD and discuss the details of our distance function implementation.

2.1 System Architecture
From a broad perspective, the architecture of CSD-CMAD is the one of a client-server type3. An illustration of the main components is provided in Figure 1.

2.1.1 Backend. As shown, we have a web server in the backend, running on an Ubuntu VM, hosted on Iowa State Cloud. In the VM, the primary systems are nginx, Gunicorn, PostgreSQL, and Memcached.

(1) nginx is our primary web server, and is our initial interaction to incoming requests on port 80.
(2) Gunicorn reacts whenever nginx spawns new processes, to run a WSGI server for the Django instance.
(3) The Django instance houses the logic and functionality of our project, as well as the algorithms that interpret the data. It also interacts with a PostgreSQL database for data storage, as well as a Memcached cache for caching of data to speed up response times.

Since an adjustable metric for the distance function (as well as diversity) was required, we relied on the PyClustering package for implementation of the clustering algorithms. In addition, in order to provide a better user experience with the large files for datasets, the asynchronous package Celery was used, along with its asynchronous service provider RabbitMQ. Lastly, we note that the job of rendering the graphs to be displayed to the users shifted from the client to the backend to reduce data transmission, and we used Matplotlib.

2.1.2 Frontend. Our client side consists of a website that provides the following main categories of functionalities for the users:

- Select a dataset – either an existing one from the database, or a datasets generated by a new simulation (in .csv format) which will firstly be pre-processed with the corresponding scripts to become a new database table (cf. Figure 2). An existing dataset can also be removed.
- Display the attributes of the dataset, and select the ones desired for proximity as well as diversity (cf. Figure 3).
- View the graphs displaying the outcome of the clustering algorithm.

For the sake of the visual appeal of the UI, we set up and styled the wireframe, and the corresponding Django forms and JS-controlled forms. The UI was built with three main parts in mind: a data view/selection part; a distance function selection and an attributes weight section; and the visualization of the clustering outcome – corresponding to the main functionalities. We provide a more detailed description of the UI parts in Section 3.

We close this section with a remark that one of our aims during the design of the CSD-CMAD system was to enable easy extensions in terms of augmenting its functionality with other clustering algorithms – which was part of the reason for selecting PyClustering. Also, we note that the code of our implementation is publicly available at https://github.com/sdmay21-31.

---

Figure 1: Overview of system architecture

---

2https://www.POSYDON.org

3For more detailed discussion, including design-decisions, technology consideration, testing (user/integration/acceptance) etc. – please see the publicly available report at https://sdmay21-31.sd.ece.iastate.edu/docs/DesignDocumentFinal.pdf
2.2 Distance Functions
Since at the core of every clustering algorithm is a calculation of a distance function, and we are attempting at providing clustering capabilities for data with (multiple) heterogeneous attributes, we applied the following pre-processing steps.

1. For each attribute \( a_j \) in the database, we detected the maximal (\( a_j^{\max} \)) and minimal (\( a_j^{\min} \)) values across the entire dataset of stars (at corresponding time instant).
2. For an attribute \( a_j \) corresponding to a particular star \( S_i \), we relativized its value \( S_i.a_j \) by scaling it with respect to \( a_j^{\max} \) and \( a_j^{\min} \), obtaining \( S_i'.a_j = (S_i.a_j - a_j^{\min})/(a_j^{\max} - a_j^{\min}) \).
3. In addition to the numerical scaling (which generates values in the range [0,1]), the ratio also eliminated the different physical dimensions and therefore we can now proceed with using the traditional Euclidean-based calculation of the distances between two points (i.e., stars). Assuming a total of \( n \) attributes, the distance between two stars \( S_i \) and \( S_k \) becomes a numerical quantity:

\[
dist(S_i, S_k) = \sqrt{\sum_{j=1}^{n} (S_i'.a_j - S_k'.a_j)^2}
\]

If more weight is desired for a particular attribute \( a_m \) (i.e., increase the impact of \( a_m \) on the clustering result), the user is allowed to select a desired weight value \( w_m \). In the current implementation of CSD-CMAD, the values are integers. However, in terms of above equation for distances, the implication is that its reciprocal is used to multiply the respective attribute (i.e., \( 1/w_m \times (S_i'.a_j - S_k'.a_j)^2 \)) when calculating the sum. This has the effect to to decrease the impact on the distance for the chosen attribute - i.e., to "bring" the objects closer in the dimension of that particular attributes. The selection of values for \( w_m \)'s is part of the UI.

Now, to incorporate diversity with respect to a desired attribute \( a_d \) in CSD-CMAD we take the following approach:
1. We let the user select a percentage portion (i.e., a value between 0 and 1) \( \alpha \).
2. If the difference \( |S_i.a_d - S_k.a_d| \) is smaller than \( \alpha \times (a_d^{\max} - a_d^{\min}) \), we set the value of that difference to a large constant (e.g., 100). That way, since all the other addends in the sum in the equation for \( dist(S_j, S_k) \) are \( \leq 1 \), the pair \( (S_j, S_k) \) is guaranteed to have a large enough value of \( dist(S_j, S_k) \) so that it is not considered close enough for the respective clustering algorithm.

3 DEMONSTRATION SCENARIO
We now present the details of the demonstration scenarios that the attendees can experience, essentially providing a step-by-step illustration of the functionalities of CSD-CMAD.

We note that in addition to the publicly available source code of the implementation of CSD-CMAD, we also have a video illustrating the functionalities, publicly available at: https://sdmay21-31.sd.ece.iastate.edu/med/demo.mp4.

1. Opening Menu As shown in Figure 2, the landing page offers the option to add a new dataset, or use one of the existing ones (or even edit it). In addition, the opening menu has the option to provide the "User Manual" type of help, as well as administrative privileges (e.g., add a new user) - provided the user currently logged on is a designated administrator. We will show how to create a new dataset.

2. Attributes Selection Subsequently (i.e., after selecting an existing one, or creating a new dataset), the user will be able to select (via check-box) the attributes that should be considered for the chosen clustering algorithm (at this version, CSD-CMAD only has K-means and DBSCAN).
Weights Selection

Once the attributes to participate in the calculation of distance have been selected, as shown in Figure 3, the user can (cf. Section 2.2):

• Select the values for the weight for the attributes that should have heavier impact.
• Select the threshold for exclusion of a particular pair from clustering, on the grounds of not being diverse enough in the values of selected attribute(s).

Visualization

The user is expected to provide a time instant (from the interval of the evolution of the stars) and, upon clicking the “Process” button, CSD-CMAD will start the execution of the processing algorithm. When the algorithm is completed, the 2D or 3D graph will be displayed, showing the clusters (and their centroids), as illustrated in Figures 4 and 5.

4 CONCLUSION AND FUTURE WORK

We presented CSD-CMAD, a system for context-aware clustering of multivariate astrophysics data of stellar evolution tracks. The heterogeneity of the data stems from the fact that the attributes correspond to different physical phenomena. The context awareness comes into play because we allow to select: (1) favorite attributes – i.e., the ones for which the proximity should have greater impact (compared to the rest of the attributes); and diversity attributes – i.e., the ones for which it is of interest that the pair of stars has a distance larger than a certain threshold. As mentioned, the code for implementing CSD-CMAD and extended documentation are publicly available.

There are several extensions to CSD-CMAD that we plan to address in our future work. Firstly, we plan to incorporate additional clustering algorithms and provide a context-aware quality assessment (i.e., which algorithms are better suited for which subsets of attributes). We would also like to add a “spatio”-temporal aspect – i.e., consider the coupling of similarity and diversity over the entire trajectory of stellar evolution[9]. From a broader perspective, we will also work on developing an implementation of CSD-CMAD in distributed environments, that will cater to even larger datasets.

Acknowledgments: The work on this project was supported by the Gordon and Betty Moore Foundation grant GBMF8477 and the Swiss National Science Foundation Professorship grant (project number PP00P2_176868).

REFERENCES


Duration: We expect that completing a run of the scenario in an interactive demonstration of the features of CSD-CMAD should be completed within 4 ~ 5 minutes, with additional 2 minutes to illustrate the help/user manual aspects, as well as the administrative features of the main menu.