Clustering works from the Cleveland Museum of Art - Final Report

Da Vinci's Visuals - William Braga, Harshal Dahake, Kewal Khatiwala,
Hannah Li, Sidney Miller, Aditya Tapshalkar
Profs. Chau, Roozbahani (CX 4242/CSE 6242 - A) - Georgia Institute of Technology (Fall 2021)

Introduction

The Cleveland Museum of Art (CMA) is a Creative Commons Open Access institute, providing online and free access to its collection sporting more than 30,000 works of public-domain art. With this artwork data, CMA has implemented a public API with configurable endpoints and queries for performing network requests to get artworks, artists, or CMA exhibits by customizable parameters.

Our team worked with artwork data provided by CMA's online collection and visualize in a web application relations between works of art with respect to user-queried criteria such as culture, creation date, department, and art medium, using the unsupervised K-means machine learning clustering algorithm with user-defined *k* values for the number of clusters into which users would like to group artworks. We present to viewers a big-picture overview of related artworks and even uncover how past art inspires artists from different cultures or time periods in subsequently replicating art style, medium, or technique in their own art. We hope that our project serves relevant and convenient for curators and art historians and enthusiasts intrigued about how artworks are related.

Problem Definition and Final Deliverable

Fetching and looking at the amount of data stored for each individual artwork entries through sample HTTP requests to the CMA Open Access API inspired us to find a way to organize this data by these specific categorical attributes while simultaneously exhibiting how closely different artworks are related. With our visualization, we are looking to innovate envisioning art data by allowing users to explore artworks that have similar properties to other artworks and discover themes that can be found in artworks organized by user-configured parameters using unsupervised machine learning. This clustering visualization can also be employed by museum-goers to feel more connected to the artworks in a museum.

Survey

Prior Art

Past studies into creating and analyzing interactive museum data visualization displays have yielded mixed results. One team, led by Hinrichs et al. (2008), created a museum visualization display for the Glenbow Museum exhibit on Emily Carr, artistically depicting themes in her autobiography and publications over time, and explored how their visualization rewards both short-term and long-term exploration. Their visualization displayed data metaphorically as a tree but the implementation notably often took focus away from the data itself (Hinrichs et al., 2008).

Another study conducted by Ma et al. (2020) described how using elements like animation to represent change over time and color to categorize data could enhance or hinder comprehension of data in the team's exploration into plankton population visualization. Similarly, researchers Börner et al. (2016) and Styliani et al. (2009) concluded in their studies, after conducting user contextual "think-aloud" interviews, that data trends highlighted by visualizations can be inaccessible and unable to be perceived by those technologically and visualization illiterate or with other handicaps, which we will need to consider in our visualization construction.

Some solutions are proposed in an article written by McGhee (2015), such as avoiding overcomplicating visualizations and utilizing metaphorical presentation to illustrate data in a more familiar manner. McGhee (2015) also brings to attention the need for balancing artistic elements and data representation by providing examples of successful visualizations, of which we will take inspiration in our project. Additionally, Styliani et al. (2009) outlined issues in providing high-resolution digital artwork efficiently with limited bandwidth and storage space, and explained how resorting to preserving "Russian Doll" architectures of images optimizes image retrieval for variable bandwidth availability and therefore considerably addressed server accessibility.

Since GAC is far-and-away the most effective tool for virtual art exploration, it would not make sense to compete with them head-on. In response, we modified our original scope to also focus on unconventional forms of visualization and addressed critical reviews of GAC. Due to Google's size, questions about power imbalance and artistic discourse must be addressed. The papers we found make the following three contentions: (1) GAC imposes its own narratives onto artwork by forcing unintended interpretations: "we are now seeing the things more than we should" (Zhang, 2020). Artists criticize that the super-zoom feature, user tips/directions, and captions introduce a new narrator to the work that intrudes on the artists' intentions (Zhang, 2020). For our project, we will not incorporate these features. (2) GAC implies the value of one piece over another through the varying resolution of exhibits, the emphasis of certain artworks (like on their main page), and the default sorting of art in a collection (like "Street Art" appearing at the bottom of their mediums list) (Bayer et al., 2014). In development, we will be mindful of these power considerations and not make any undue comparisons via default sorting or branding.

Approach

Accounting for previous and current visualizations of data and how digitized artwork can be leveraged to uncover hidden trends and relations, we want to design our visualization of artwork data from CMA's Open Access API such that users can easily interpret similarities in the attributes in the art with minimal loss of data comprehension. As in previous implementations mentioned above, we plan to analogize data with familiar visual metaphors and grant interactivity through applying custom clustering configurations and embedding detailed webpages and information on user-selected works.

Works in the visualization would be depicted as nodes on a planar canvas created in d3. Distances between two artwork nodes on the canvas would correspond to how closely the two works are related, with respect to the preset clustering configuration. The Open Access API includes relevant data about each queried artwork, including type (e.g. "Painting"), department (e.g. “African Art”), culture (e.g. “Arctic”), and creation date (e.g. “c. 1800”); these categories would serve as some criteria for clustering.

In addition to these querying criteria, we initially wanted to develop our own method of querying artworks for the user, in which artworks could also be grouped with respect to subject matter, which would be determined through utilizing Computer Vision services and publicly available deep learning models such as ResNet50. However, due to the uncertainty of generating classifications on artworks in the API with possibly inconsistent image qualities, we opted to instead leverage natural language processing (NLP) to allow users to cluster artwork by title, through the generation of word vectors via Word2Vec.

For this project we used a JavaScript (React) front-end for displaying the visualization in a webpage format and a Python (Flask) back-end for processing raw API data. Frameworks such as Scikit-Learn, OpenCV, and Pandas were regarded and appropriately applied to conduct the K-means algorithm, and an Azure SQL database was used to store fetched data from HTTP request calls to the CMA Open Access API. The cost of fabricating our project was minimal as any Azure cloud service expenses were covered through Azure’s student credit.

Proposed Method

Data Processing, K-means Algorithm, and Back-End Setup

The machine learning backend pipeline had three main stages: encoding, principal component analysis (PCA), and K-means. The pipeline is encapsulated in a Python class, ArtKMeans, that takes in a Pandas DataFrame of artwork information and a list of features to be clustered. Unnecessary columns as omitted by user configuration are culled from the DataFrame, and the Flask server-side script then converts the DataFrame to a JSON dictionary ready to send back to the front-end.

The features are then put in one of three buckets: numeric, categorical, or word_vec. Numerical features require no more pre-processing and are ready for PCA. Categorical and word_vec variables must be numerically represented.

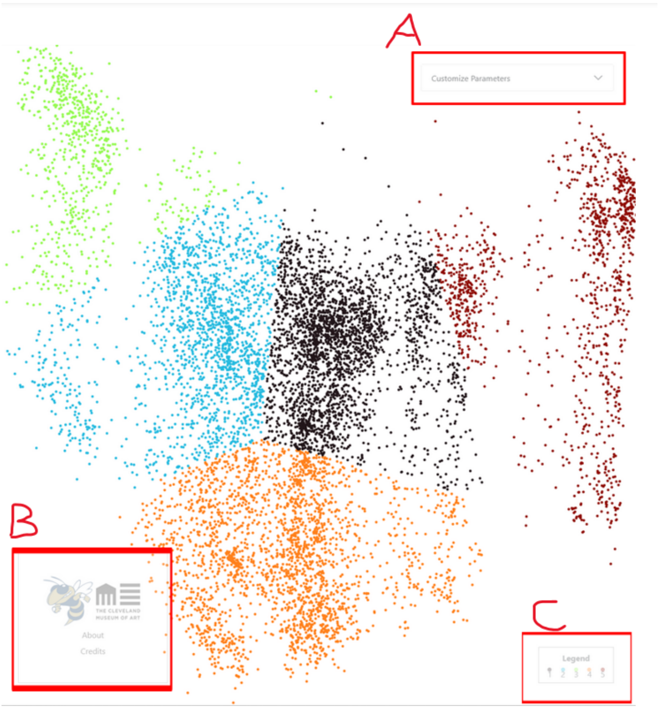
Categorical and word_vec features are both string types, but we define categorical features as those having no correlated values in its range, in contrast to word_vec features whose values may be correlated. For example, for the feature ‘Artwork Type’, values such as ‘painting’, ‘print’, and ‘silver’ are distinct and hold little to no semantic relation.

Taking a word_vec feature like ‘Culture’ for instance, with sample values such as ‘America, 18th century’; ‘America, 20th century’; and ‘Netherlands 17th century’ now have an uneven correlation. Two artworks from America, regardless of the century, should be represented as being more similar in culture to each other than one from America and one from the Netherlands. To make sure these similarities are captured by our K-means process, these two kinds of features are encoded in different manners.

For encoding categorical features, we chose the one-hot encoding strategy in lieu of just converting each unique value in the feature’s range to a consecutive integer. One-hot encoding assigns a value in a consecutive dimension, ensuring that during a run of the K-means algorithm, values are not arbitrarily determined to be less or greater than another; instead, these features are encoded to be orthogonal. Word_vec features are encoded using the spaCy NLP library. Each cell is encoded with its average word embedding.

Once finished with encoding, the data has a significant number of new features (having expanded all the one-hot and embedding vectors to new columns). Conversely, the output must be represented in two-dimensions so that it can be plotted. To handle the dimension reduction, the data is normalized and undergoes PCA. Now, it can be sent to KMeans, where each value will be assigned an id linking it to a cluster. The final output of ArtKMeans is an Nx4 matrix containing each node's primary key id, x coordinate, y coordinate, and cluster id.

D3.js, ReactJS, and Front-End Configuration



We chose to develop our front-end in ReactJS, due to having past experience working with Node and the ReactJS framework. In initial implementations, we had a working D3 where, upon sending a POST request to our Flask server, we were able to output a basic plot of all artworks received by the POST callback JSON data. We then expanded on our front-end layout by modifying our canvas to fill the entire webpage to better immerse users, and building modals with instructions on using the application and a list of credits for the project. One design choice we employed was varying opacity levels for on-screen components, such as the main menu, parameter dropdown component, and the legend component. We wanted to ensure that these components would not be intrusive in the exploration of the plotted data, and the component appropriately comes into view when hovered over. Another feature implemented was the loading spinner; we felt that while the K-means parameters were being configured and run on the data, users would benefit from having visual confirmation that their inputs were successfully fed into our algorithm.

The above figure showcases three components overlaid on the graph. Component (A) is the dropdown menu for users to input parameters for their K-means instance. Component (B) is the main menu where the About and Credits links display modals that detail instructions on how to use the application and listing our team members and responsibilities, respectively.

Component (C) is the legend for the visualization, with each node attributed to a specific cluster ID.

Configurable parameters in the parameter dropdown menu were subject to validation to ensure the algorithm would not run on invalid inputs or time out. We found our upper limit on the number of renderable artworks to be around 10,000 at a time, so we limited users to selecting a number of artworks to render between 1 to 10,000. We also made sure to handle cases where the number of clusters may exceed the number of data points, and eliminated the possibility of users being able to conduct K-means on less than 2 features at a time.

Experiment

Upon starting this visualization project, we sought to answer the following questions:

(1) How can we best describe a relationship between 2 pieces of art? What does this mean in terms of visuals?
   - (a) We evaluated the performance of our K-means algorithm by qualitatively comparing artworks within the same clusters and gauging how similar or distinct pairs of artworks were in the graph depending on the distance between them.
   - (b) Our initial idea with the K-means visualization was to find and display the optimal number of clusters using the Elbow Method or Silhouette Analysis; however, since this would require multiple runs of K-means at different values of *k* and also since users would be manually configuring the number of clusters into which artworks would be grouped, we tabled this idea given the scope and time constraint of the project.

(2) How can we incorporate different text based variables as factors?
   - (a) We used the one-hot encoding system to convert the text based data into numerical features which were then used in the k-means model.
   - (b) Geographical information could be extracted from values in the ‘Culture’ column, and grouping by geographical origin data (i.e. proximity of coordinates) could provide an added dimension to the visualized data.

(3) What variables should be considered?
   - (a) After reviewing all of the different potential variables, we found that title, creation date, culture, collection, type, and technique to be the most insightful variables. We found that most values associated with artists we either already included or inconclusive.

(4) What elements should be inputs for the user?
   - (a) The user must enter the number of datapoints n which must be less than 10,000 and k which is the number of clusters limiting to 25 after which the processing takes longer than usual. The user also needs to select at least 2 of the attributes for clustering.

(5) If we decide to use a K-means cluster, what is the best way to visualize large sets of data?
   - (a) By categorizing the clusters by significantly contrasting colors to distinguish one cluster from another and develop a standard scaling system which clearly tells

what the values signify. A legend telling which cluster represents what would be helpful.

(6) Upon selecting a node, what information needs to be included? What information can be exempted?

   (a) The node must provide all the essential data which would include title, author, any possible and important dates, categories, techniques, time period if available and a small scaled image preview. We can exempt some specific details like descriptions, all details of author other than name which would make the node too cluttered with text and ruin the user experience.

(7) How does our model scale up to larger datasets?

   (a) Our model will significantly slow down as the cluster size increases due to the number of iterations it will have to take. We could set up an algorithm to automatically optimize the k value which would take some control away from users and they would only be allowed to have their custom number of datapoints.

Conclusion

______After accessing the API data from the Cleveland Museum of Art (SMA), we were able to filter and configure the data to be used in our visual. By passing our complete data through our unsupervised K-means machine learning clustering algorithm on a variable user-defined number of clusters and criteria. From here, we were able to present to viewers a big-picture overview of related artworks along with underlying trends from different cultures or time periods in subsequently replicating art style, medium, or technique in their own art.

User Contribution:

All team members have contributed similar amount of effort

- William Braga (back-end, ML engineer, data processing)
- Harshal Dahake (back-end, schema configuration)
- Kewal Khatiwala (back-end, data processing)
- Hannah Li (front-end)
- Sidney Miller (data processing, presentation)
- Aditya Tapshalkar (front-end, back-end)

References

Bayer, A., Farber, C., & Bayer, A. (2014). EVANGELIZING THE ‘GALLERY OF THE FUTURE’: A CRITICAL ANALYSIS OF THE GOOGLE ART PROJECT NARRATIVE AND ITS POLITICAL, CULTURAL AND TECHNOLOGICAL STAKES.

Börner, K., Heimlich, J., Balliet, R., Maltese, A. (2016). Investigating Aspects of Data Visualization Literacy Using 20 Information Visualizations and 273 Science Museum Visitors. Information Visualization 15 (3): 198-213.

Google. (2021) Google Arts & Culture. Retrieved October 16, 2021, from https://artsandculture.google.com/.

Hinrichs, U., Schmidt, H., Carpendale, S. *EMDialog: Bringing Information Visualization into the Museum*. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2008)*, 14(6):1181-1188, November-December, 2008.

Hylland, O. (2017). Even Better than the Real Thing? Digital Copies and Digital Museums in a Digital Cultural Policy. Culture Unbound: Journal of Current Cultural Research. 9. 62-84. 10.3384/cu.2000.1525.179162.

Ma, J. (2013). Engaging Museum Visitors with Scientific Data through Visualization: A Comparison of Three Strategies. Paper presented at the annual meeting of the 2013 American Educational Research Association, San Francisco, California.

Ma, J., Ma, K. -L., Frazier, J. Decoding a Complex Visualization in a Science Museum – An Empirical Study. *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 472-481, Jan. 2020, doi: 10.1109/TVCG.2019.2934401.

Marmitage. (2021, June 18). *Open access*. The Cleveland Museum of Art. Retrieved October 16, 2021, from https://www.clevelandart.org/open-access.

McGhee G. (2015, September 22). Taking Data Visualization From Eye Candy to Efficiency. National Geographic. https://www.nationalgeographic.com/science/article/150922-data-points-visualization-eye-candy-efficiency

Styliani, S., Fotis, L., Kostas, K. Petros, P. (2009). Virtual museums, a survey and some issues for consideration. Journal of Cultural Heritage. 10. 520-528. 10.1016/j.culher.2009.03.003.

Zhang A. (2020) The Narration of Art on Google Arts and Culture. School of The Art Institute of Chicago. Vol 1, Issue 1, Article 149.