University College London

The Metropolitan Museum of Art Explorer

Website: http://dev.spatialdatacapture.org/~ucfnhke/
Public repository: https://github.com/ucfnmyo/SDC_MuseumsProject

By Terry Lines, Hugh Kelley, Carlos Mestre del Pino, Mohammad Younes, and Jason Masurovsky

Spatial Data Capture, Storage and Analysis

Steven Gray and Huanfa Chen

May 22nd, 2019

Word Count: 5,232

1. Project Context and Aims

The project had two aims. Firstly, to allow the general public to engage with the Metropolitan Museum of Art (MET) collection with a more sophisticated level of interactive visualization than available currently. Secondly, to use data mining techniques to create an alternative taxonomy of artworks, challenging traditional concepts of museum interaction.

The Metropolitan Museum of Art, of New York City, contains artwork from all around the world spanning over 5,000 years. It is the third most visited art museum in the world, having 6,953,927 annual visitors, as well as being the largest art museum in the United States. Its permanent collection contains over two million works of art. The museum has three locations in the City: The Met Breuer, The Met Cloisters, and its largest gallery The Met Fifth Avenue. The MET's Open Access program allows users to view the collection digitally. The public domain is automatically updated as the MET digitizes artworks, and anyone can download the collection dataset and manipulate it without copyright restrictions. The museum encourages the public to build new knowledge with this data, for example the MET collaborated with masters students at The Parsons School of Design to visualize the data. In particular, one example considers the most common materials used in the museum's collection (https://3milychu.github.io/met-erials/). The user can explore a material through representative artworks. Inspired by this example, this project approaches the MET collection through materials: examining how common materials and methods used in one country of origin have influenced artwork made by similar materials and methods in other countries throughout history.

1.1 The digital museum as concept

The development of Web 2.0 has allowed museums to present their artwork digitally. Museums now receive more online than physical visitors (Axiell, 2016). Applications typically are searchable databases which narrow records down through search fields to produce a list of records which can then be used to obtain detailed information about the pieces of art. However artwork has a spatiotemporal context that can be used to provide a member of the general public with new insights based on exploration of a large dataset and this approach has been used by Dumas et al. to create ArtVis, an interaction tool based on 28,000 artworks from the Web Gallery of Art (Dumas et al 2014).

Interactive visualisation can be considered as "Overview first, zoom and filter, then details on demand", along with the ideas of "linking and brushing", to select portions of data and have those elements highlighted in other datasets (Dumas et al, 2014).

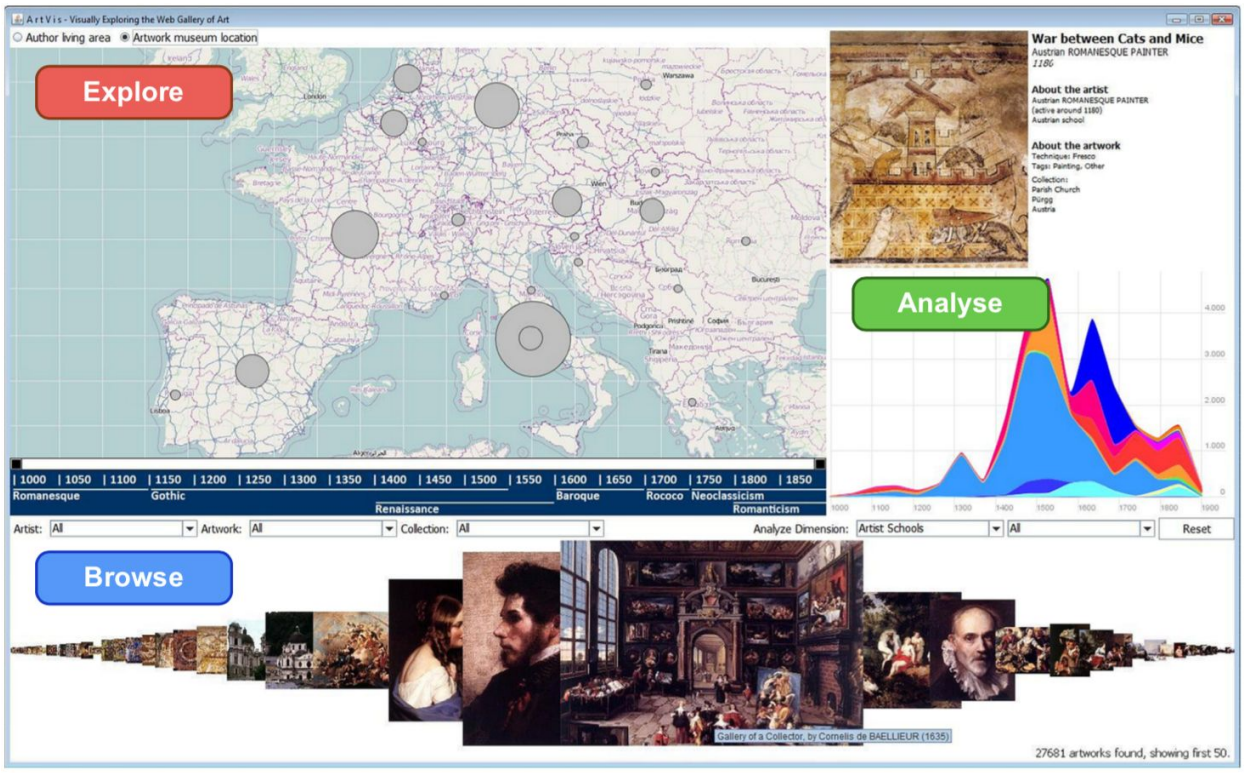

Figure 1: Graphical ArtVis user interface consisting of the Explore, Analyse and Browse panels (Dumas et al, 2014)

In essence, ArtVis displays artwork locations at a city level, along with graphs of activity over time, broken down by selectable dimensions such as school of artist. Dimensions are selectable by category to filter data further. Lastly, 50 selected artworks are displayed with tooltip interface providing further details on demand (Dumas et al, 2014).

The limitations of ArtVis are that the analysis is one-way from using the analysis tool to update the explore and browse elements, with an inability to filter by location, or to go from a specific artwork in the browse area to a related category.

1.2 A critical interpretation of digital practices

Museum practice has emerged from a taxonomical, empirical approach which persists today (Cameron, F. and Robinson, H., 2007). The approach taken by ArtVis reproduces existing practices by replicating the taxonomy of the physical collection. However, critical theory into the use and practices of emerging digital technologies by cultural institutions highlights the

possibility of utilising alternative knowledge systems (Cameron, F. and Robinson, H., 2007). Ethnogeography is an interpretive framework that is used in museum research to understand the interaction between society, cultural practices, and its spatial representations to thicken an understanding of the power of place (Boogaart, 2001). Geography contributes to a deeper understanding of how the environment can cultivate cultural/ritual practices that relate to the latter interactions. Pierre Bourdieu coined the concept of a cities "cultural capital", as moulded by the artistic productions of its inhabitants. Cultural capital can be measured in an objectified manner that is supported by materials to express and transmit these forms into symbolic appropriation; the idea gaining knowledge of other, often minority cultures, through their symbolic materials (Boogaart, 2001).

Specific to digital technologies ,the potential has been identified of co-creation of knowledge and a changed relationship between experts and non-experts., for example social media tagging of objects can provide alternative classifications (Cairns, S., 2013).

However the possibility for users to use digital technologies to systemically reclassify museum collections, as opposed to individual objects, is underexplored. In contrast, web 2.0 has had a profound effect on mapping, with the widening ability to engage in cartography with disregard for existing practices viewed as an extension of critical cartography.

2. Project Rationale

The aim of this project is to build a visualisation tool which explores the over 400,000 items in the MET digital collection within their spatiotemporal context. This expands on the work of ArtViz by utilizing modern web development tools to provide the reflexive browsing functionality that Dumas et.al highlight as a missing component. Additionally we will consider the ability to disregard existing taxonomies and rely on alternative methods of classification through data-mining techniques to find sub collections of work that provide additional contextual interest through their group characteristics.

3. Data Collection

The museum's dataset presents over 490,000 records of art objects, describing the artwork through 44 attribute fields. It is available from their GitHub repository as a c. 250mb csv file (https://github.com/metmuseum/openaccess). It is continually updated, our version is dated 25 April 2019. It is licenced for use under Creative Commons Zero, albeit this repository does not include artwork images, only some of which are covered by Creative Commons licencing. Additional artwork information beyond the repository is available from the MET API, including links to artwork images, however this API only permits the information for a single object to be returned in one call.

The artworks are categorised by country. To visualise this on a map, we used a world countries geojson file to provide the country polygons. A base layer map was used for reference, provided by Stamen design, again under Creative Commons.

4. Data Cleaning

The dataset fields are summarised in Table 1. Information was primarily text string. Of particular interest for us were the fields relating to date, location, medium (a description on the materials and methods of the artwork creation), and classification (the MET taxonomy). The range of objects in the collection includes a large number with no artist, either due to its age or the nature of the artwork (which includes historical artifacts), and this was not considered for further analysis.

Thus a large proportion of our project was to contextualise the text data to prepare it for quantitative analysis, and this was an iterative process of analysing, cleaning and re-analysing. Pandas library in Python was used to clean up and improve the dataset. Finally, this cleaning process was iterative, with analysis needs and problems driving the cleaning decisions discussed above.

Data Field	Description	Problematic Values	Type
Object ID	Unique ID	0	Int
Object Name	Describes the physical type of the object	4406	String
Title	Title, identifying phrase, or name given to a work of art	31264	String
Artist Display Name	Artist name in the correct order for display	206978	String
Artist Display Bio	Nationality and life dates of an artist, also includes birth and death city when known.	256612	String
Object Begin Date	Machine readable date indicating the year the artwork was started to be created	0	Int
Object End Date	Machine readable date indicating the year the artwork was completed (may be the same year or different year than the objectBeginDate)	0	Int

Medium	Materials and techniques used to create the artwork	7604	String
Credit Line	Text acknowledging the source or origin of the artwork and the year the object was acquired by the museum.	794	String
City	City where the artwork was created	462543	String
Country	Country where the artwork was created or found	417978	String
Classification	General term describing the artwork type.	56422	String

Table 1. *Relevant data fields from raw MET dataset, their description, number of missing values and data type are presented.*

4.1. Country and City Data

A necessary process was matching the country fields to the geoJSON being used for country polygons. Initially, only ~76,000 objects had a country filled in the field which matched the geoJSON. Three steps were undertaken to clean and contextualise the data: firstly, removal of any undesired string text and any key fields that were not associated to the name of a country e.g. 'probably China', 'possibly from', repeating and other string characters; secondly, extraction of keywords using the flashtext python library as detailed further in section 6, with manual cleaning using python string functions to fix evident mismatches; and lastly, where no direct country information was present, using the country associated to the artist's nationality, the artist display bio, and the culture.

This last process required printing a list of all text values in the 'artist nationality', 'artist display bio', and 'culture' fields, and creating an individual line of code for each country we could identify (Best, 2018). Transformations such as 'American' to 'United States' or 'Swedish' to 'Sweden' were easy, but few texts such as 'Flemish' to 'Belgium' or 'Babylonian' to 'Iraq' required fact-checking. Finally, there were some countries that had to be renamed to avoid any duplicates such as 'Holland' to 'Netherlands' or 'Greek' to 'Greece'. This increased the number of art objects associated with a country of origin from 76,000 to ~430,000 art objects, very significantly improving the overall dataset. It is noteworthy that the MET's official website for exploring their database (https://www.metmuseum.org/art/collection/) only uses their limited matching when searching by country.

A similar process was started for city level data, but the group ran out of time to finalise this and utilise the data.

4.2 Classification Field

The classification field contained a general term describing the artwork type (e.g. paintings, sculptures). All artwork had a value, although in some cases it was classified as unknown (referred to as missing values above). Our challenge was a lack of generalised categories covering a wide range of artworks, to illustrate, where artworks were classified as 'Membranophone-single-headed / frame drum' and 'Membranophone-single-headed / kettle drum' should be agglomerated into one category: 'membranophone'. To contextualise this data, we used POS tagging with TextBlob, a Natural Language Processing Library (https://textblob.readthedocs.io/en/dev/), to extract noun phrases from the text data, these fed into a keyword extraction function by flashtext, which enabled us to extract relevant artwork categories from the data. This process reduced 1,114 classifications into 205 usable classes, allowing us to group artworks in a comprehensive and efficient manner.

4.3 Medium Field

The cleaning and analysis process for the Medium field was more complex and is referred to in section 6.

5. Data Handling

After cleaning and structuring the data locally with Python, it was loaded onto the CASA server. There, a node.js API was used to interact with the MYSQL database in order to deliver the right combination of data to the various front end content. For the map, every row for a subset of columns was delivered while for the charts, specific premade tables were used, and other features use the endpoints detailed in the API documentation to select a subset or summary of the relevant data. Care was taken to escape all endpoint input before using the input to mitigate sql injection.

6. Data analysis methods and results

Two approaches were used to create classifications. The first was language processing of textual data relating to the artwork medium. The second was a subsequent clustering analysis taking into account variously medium, classification, date and location.

6.1. Text analysis

The aim of the analysis of medium was to understand how certain materials and techniques were employed within artworks across classification, time and space. The challenge was that the raw medium data contained inconsistent formatting and concatenated strings that

prevented categorical grouping. For example, a typical medium description would be “Belleek porcelain, with overglaze enamel decoration and gilding”. In this situation, desired materials would be ‘porcelain’ and ‘enamel’ (nouns), while techniques would involve be ‘overglaze’ and ‘gilding’ (verbs). Extracting meaningful, contextual information from these strings required Natural Language Processing (NLP), namely Point of Speech (POS) tagging. TextBlob (https://textblob.readthedocs.io/en/dev/) is a NLP library, which enabled a simple extraction of verbs (VBG, VBN, VBP) and nouns (NN, NNP) from the dataset. Following manual inspection of the results, less relevant words were removed or edited to optimized keyword matching. For example, ‘engrav’ was set as the keyword to capture both ‘engrave’ and ‘engraving’. Results of this exercise are displayed below, with the distribution of counts shown.

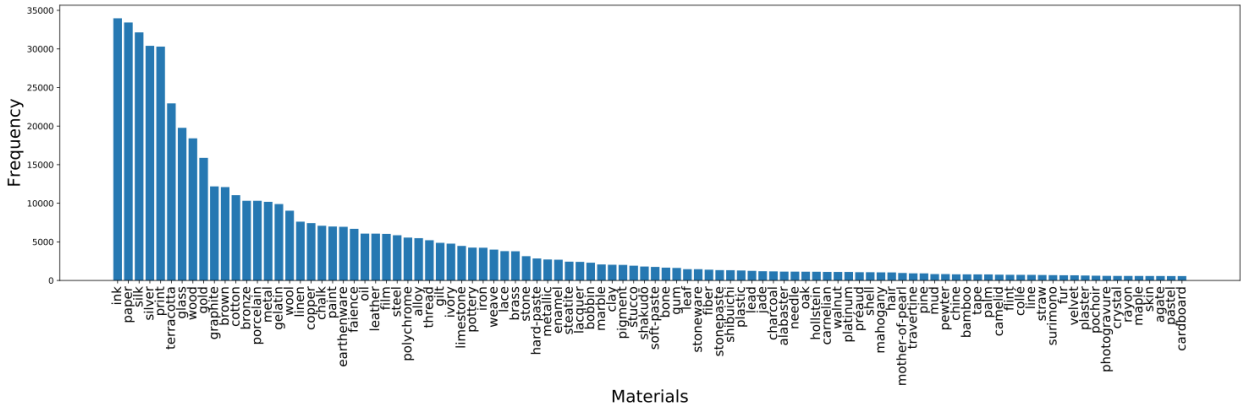

Figure 2. *Extracted material keywords, sorted by frequency of instances in medium data.*

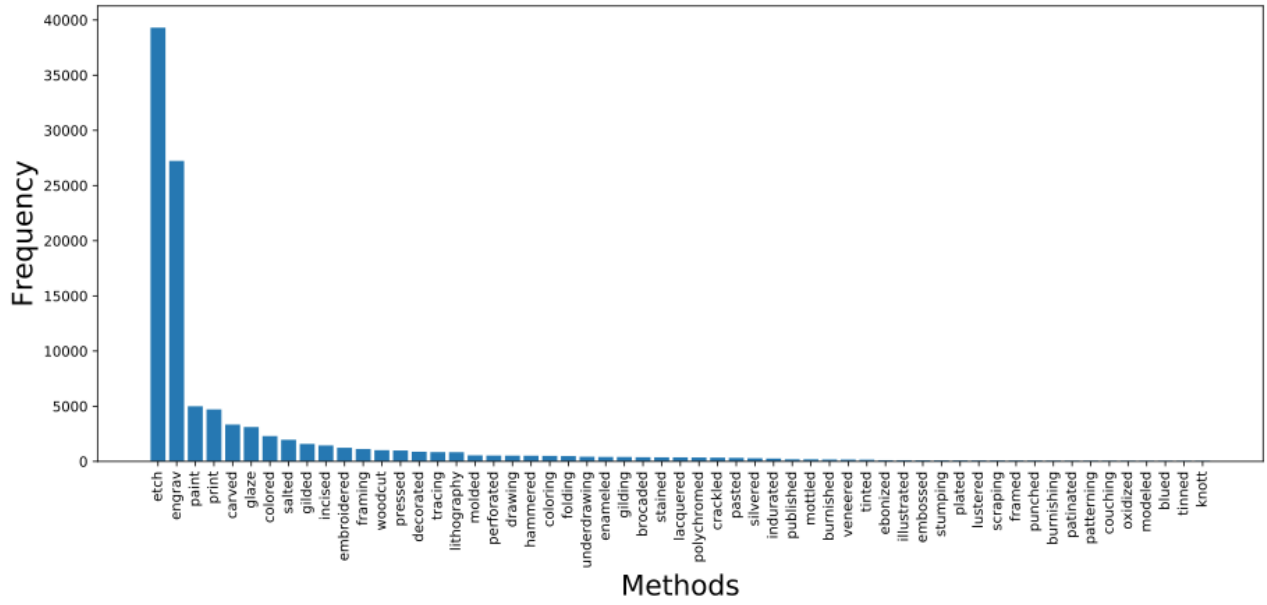

Figure 3. *Extracted method keywords, sorted by frequency of instances in medium data.*

In itself, this provided an alternative method to the official classification used by the MET, however we additionally utilised the data along with our other contextualised fields to perform a clustering analysis.

6.2. Clustering Methodology

The second approach to categorizing data was using clustering algorithms, which was difficult with our underlying dataset. We have no insight to the number of expected clusters and we anticipate a large variance of cluster size based of the significance of the cluster. Additionally, we expect high levels of background noise relating to sparsity of data, for example older artworks or relating to underrepresented cultures. Consequently, partitional clustering, such as K-means clustering, which performs quickly on large datasets, would be inappropriate. Furthermore, as described above, there was a high proportion of missing or unclean data, and we obtained only 4 consistent artwork variables: date of production, country of production, MET classification and the results of the medium keyword analysis above.

To cluster categorical variables, a standard approach is one-hot encoding: creating a binary parameter for each potential category value. However our artwork could have multiple values for our Medium category, and both Medium and Classification could be missing or unknown for artworks. As one-hot encoding does not deal well with varying information levels, the Gower distance was considered as an alternative (Gower, 1971), however in general if partitional methods are not suitable, hierarchical clustering for mixed datasets has $O(n^3)$ runtimes (Ahmad and Khan, 2019), which was unfeasibly long for our 430,000 records. As a consequence DBSCAN was chosen as it is able to run in O(n log n) which equated to c. 40 minutes on our dataset. This required further data preparation: reducing each artwork to be associated with one Medium category (choosing the most frequent keyword in each case); reducing the number of categories for Medium and Classification from over 200 to only the top 40 (to reduce runtimes); and lastly removing artworks with null values for Medium or Classification to avoid the creation of an artificial cluster. This process reduced the number of records from 430,000 to c 280,000.

Runtime concerns also came into our calculation of spatial distance. For this we are using the centroids of the countries. Whilst an accurate distance can be calculated between 2 latitude/longitude points, the implementation of DBSCAN in python's scikit-learn library is far slower when a non-standard distance function is used. In order to work round this we converted the coordinates to a projection, using a Lambart Conformal Conic projection processed using QGIS. Its advantages are that straight-lines in the projection equate to great-circles, the shortest path between two points, and it minimises the distortion of distance in the northern hemisphere, which is where the majority of the artwork is from. The disadvantage is the overstatement of distance in the southern hemisphere. Furthermore by using a projection, we miscalculate distance

where the shortest path lies across a projection boundary, e..g between USA and Japan the shortest path goes via the Pacific, not Europe.

Lastly the artwork dates were distributed with a Zipf's law distribution and were transformed to the logarithm of their age to create a normal distribution for clustering purposes. All data processing was undertaken in python using pandas, numpy and scikit-learn libraries.

Results undertaken with this comprehensive methodology were inconclusive, and therefore a more detailed clustering analysis was undertaken on the medium only, where the full keyword set of methods and mediums were utilised, and records were one-hot encoded with multiple keywords. This helped located interesting results for further analysis, although the results overlapped heavily with the original keyword categorisation.

6.3. Clustering Results

The results of DBSCAN when run with the space, time and categorical variables, in general provided low silhouette scores with clusters primarily spatiotemporal clusters within a single classification and medium. In comparison we also ran DBSCAN using just space and time, however again it was difficult to distinguish significant versus trivial clusterings using silhouette scores, and we had insufficient time to fully parameterise and explore the model. Fundamentally, the data was insufficient to undertake a clustering: years were often rounded to the nearest 100 years; country level data was not sufficiently granular (our original intention was to use city level data); and both the country and categorical data had Zipf's law distributions which made it difficult to parametrise models at a level produced interesting clusters without the larger categories swamping the dataset. A hierarchical versus DBSCAN approach may have yielded results that would be easier to interpret but ultimately the runtime concerns made this unfeasible. In hindsight, clustering mixed data is very difficult without comprehensive data and the time to undertake an iterative workflow of testing and refining the model, and was too ambitious for this project.

The DBSCAN undertaken on purely categorical data returned 413 clusters with a positive silhouette score of 0.68. 89,341 number of artworks were outliers, resulting in a sum of ~340,000 number of artworks in the clusters. Within the clusters, the classification and material fields had some clusters that were unknown or not specified. Therefore, to filter out interesting clusters, we excluded clusters that did not specify the classification of the art object and/or the material(s) it was made out of, producing a final set of 76 clusters.

6.4 Clustering results context

Given the clustering methodology, the relative value doesn't lie in the silhouette score but in the ability to use the obtained clusters to obtain contextual knowledge into the original dataset.

The clusters created are associated with certain groups of artworks sharing common characteristics (classification, method and medium), without constraint to a geographical area or time period, allowing an investigation into how methods and materials spread in influence, by using the visualisation tool on cluster subsets. Such a detailed investigation on many clusters is beyond the scope of this project, however example thematic studies are undertaken below.

It is possible to identify the countries in which certain currents have historically had an influence on another country and global dynamics. A cluster with a Marble medium and Stone Sculpture classification reflects Ancient Greece's influence on the Roman period [and india later on? Need to cluster links]. Another cluster with a medium of gold and classification of gold reflects the same influence Ancient Greece had on the Roman empire.

Different mediums of copper and bronze and similar classification of metalwork reflects early war weaponry (e.g. Harpoon, Ax, Sword) in India and China beginning in -1500 BC. The same medium and classification is used later on in AD history in other countries such as Indonesia, Iran, Japan, Mexico, Netherlands, and others. The difference is technological advances in weaponry did not use copper and bronze, therefore the same materials and techniques were used to make more peaceful items for example jewelry, plates, vases, a tobacco box since, and other aesthetic object works.

Just looking at clusters with classification of ceramics alone are examples of China's influence on the Middle East and parts of Europe. One ceramics cluster reflects the fact Iraq, in the 9th century, admired Chinese pottery and had a wave of imported goods from China to construct their own ceramics. The following century, these techniques spread from Iraq to Iran and eventually Egypt. Another cluster of ceramics shows China influenced Europe, predominantly the UK.

7. Data Visualization

7.1 Visualisation tool description

The visualization tool is a choropleth map with linked charts for time and selectable menus for classification and medium (a reduced set of c.40 for each with minor categories concentrated into unknown/other). The controls provide the ability to select different clusters of art., filter by time range, medium, classification and country, and obtain summary statistics on different countries. Sample record data which matches the filters is provided in a table with a short description of records. Each record has a click functionality to obtain full details of the record, by using the Met API, which avoided additional data being hosted. Each record also has a link to open its associated cluster, completing the project objective of creating an interaction tool with multi-way analysis between map, charts and object browser.

When selecting clusters, the first two have example investigations included as part of our creation of an alternative taxonomy, and further work could included extending this across more clusters.

7.2 Visualisation tool technical notes

To allow immediate interaction, the filtering of data occurs within browser using the Crossfilter javascript library, which is designed for web exploration of multivariate datasets supporting "extremely fast (<30ms) interaction with coordinated views, even with datasets containing a million or more records" ("Crossfilter," n.d.). The charts and map are produced from the filtered data using dc.js, a javascript library which creates charts in SVG format similar to the widely used d3.js library but built to work with Crossfilter. The choropleth map was created as a layer within a Leaflet.js map, providing both basemap as context, and view controls (panning and zooming). Leaflet also provides additional information and control boxes.

Country boundary GeoJSON data is hosted on the web server, with the artwork data provided through an API. The scaling of the map colour is dynamic to deal with the varying artwork ranges over different subcategories. A log scale is used to provide meaningful choropleth breaks, given the typical distribution of artwork distributions.

In general crossfilter performs well for this size of dataset, however the initial data load is significant. We initially reduced the number of fields returned by the API as well as loading up a small cluster initially with full dataset in the background, but it was decided the timesaving was not significant enough as its a mix of server response, crossfilter indexing and creating the DOM elements.

To deal with the asynchronous calls, nested functions are used. The first call uses the geoJSON data to create the leaflet map and objects, but not finalised as they depend on data (colouring of polygons and infoboxes). The artwork data is then requested from the server and used to create charts and table. Finally the map is completed by recolouring polygons and adding infoboxes to the map.

While dc.js charts are designed to update each other as filters are applied, by extending this to the record table, cluster filtering and map required some careful thought to produce the necessary update functions. DC.js does have a choropleth map and in hindsight this should have been used, however at one stage in development we anticipated clicking on polygons to zoom to data at a city scale, and this seemed easier if the svg was created directly by leaflet versus in dc.js and then reprojecting as a layer in leaflet.

7.2 Highcharts

Highcharts, an interactive javascript chart library, were visualizations used to give the user an engaging way to look at the data from a descriptive angle. The bubble chart is used to display the total number of artworks per country, grouped by the continent they fall into. The user can filter through the different classes of artwork in order to have a more specific look at the distribution of the artwork (e.g. metals, prints). The idea here would be to expand to more classifications.

Streamgraph highcharts were used to displays the volume of artworks by country of origin on a timeline. The first streamgraph visualizes artwork donated to the MET by its country of origin, since the MET was founded in 1870. The other two streamgraphs visualize the total number of artworks coming from an individual country, by when the artwork was just starting to be created, split into two time periods: BC and AD.

8. Technical integration between elements

Used single master table because we didn't need to record multiple rows for a given Object_ID so data redundancy wasn't an issue. If we had data on the display of an object in multiple locations at different times this type of data could have gone in a different table to avoid repeating the object characteristic data each time we had an entry for display. In the cases where we did need to join new data into the table, the data was formatted in python and uploaded using SQLAlchemy. Joining the new data into the existing data was done once and saved as a new "master" table replacing the old one. This was done for simplicity and computation speed. The API is fairly slow and adding a JOIN command to any of the queries would have slowed the response even more. The Object_ID serves as the primary key. In cleaning the data, the NaN value from Numpy was used to replace blank values. This is converted to SQL's NULL when uploaded to the database.

Highcharts requires a very specific data structure for each of its charts. In the case of the donation timeline, this was a series of values for each year for each country. It was clear from the outset that organizing the data in this way was much simpler to do in SQL than in Javascript. We selected the distinct countries and years from those columns in the master table and joined them together into a temporary table with a row for every year and every country. We then used a left join that pulled in the count of the objects from the master data table where the country/year pair matched and if the count was NULL used the value 0. This created a table of 135 countries X 145 year for about 19,000 rows, which was a small amount of memory to use in exchange for not having to compute the values each time the chart was built.

8.1 API

The API used to tie the MYSQL database into the front end website began as three simple endpoints, an end point that delivered the full data set to to map, an endpoint that delivered a summary, and an endpoint that delivered a subset. The full data endpoint was used to load the map data. This was done so that after waiting for the map to load initially, all of the data could be manipulated locally and therefore much faster, allowing the user a more smooth experience navigating the artwork. The summary endpoint used a column name supplied in the GET request from the USER and returns json data with a count of the number of artworks for each distinct value of the column the user indicated. Thus for Object ID, the primary key, this would be a count of 1 for each row of the database and for `/county` this would return the number of countries in the database with the number of artworks the museum possesses that originate in that country. The subset endpoint takes a key value pair form the user and returns data for rows that match the value in the column specified by `key`. This can be used to get a single specific record if the Object ID is used or all of the rows where the country of origin was the US. This endpoint evolved into the `specific` endpoint, that could take values for country, material, and a year range so that the map tool could retrieve exactly the data specified by the user. In this API `no` was used to indicate that the data should not be subset by a given key type.

Finally, as mentioned in the database section, endpoints were built to deliver exactly the data necessary for some of the charts. This was done because of the ease of managing data with SQL in the backend compared to using javascript in the front end and to increase speed , demonstrating the power of SQL for managing data.

8.2 Frontend collaboration and integration

Group work was done fairly independently and integrating independently built parts of the site proved to be a significant challenge. Our group used a github repo to track work and while usage was difficult in some situations, reconciling conflicting changes, that arose as team members edited the same website files for different purposes and features, was more straightforward. It is not clear how this challenge could have been managed without git. In a few instances, merging a branch into master undid changes that had been made to files in master and not reflected in the branch being merged but because we had a full record of changes this could be fixed in a matter of minutes. Overall, it seemed that the problems we had using git could mostly have been solved by using more git! It’s worth noting that using Github desktop created a number of problems compared to using git at the terminal plus looking at the status of remote on Github.com. It’s sort of unfortunate that the Github desktop application even exists. Lastly, git was very useful for managing local v hosted development of the website. Being able to clone the repo onto the server, change some file paths and get the website running without worrying about WinSCP or Cyberduck was amazing. Without it, we would have been editing files on our own versions of the server and trying to reconcile changes locally across a large number of files emailed back and forth.

Another key challenge was reconciling differences between the CSS files of various parts of the site. We had a least three independent sites to integrate and because class names tend to be fairly similar, adding the material from one site into the material from another had some pretty wild effects on the website's styling. This was in particular a result of the styling for highcharts and leaflet compared to the styling for the W3 template we used for the website homepage.

Finally, the most challenging part of managing the integration of the various parts of the site where all of the group members lack of experience with the packages used. We simply didn't know at the beginning how different things interacted and what the best combination of tools would be to create a cohesive website. This type of chicken and egg problem was clear in the API, where most of the endpoints that were built initially did not work to provide the exact data required by the specific uses that frontend features had for the data.

9. Conclusion

The visualisation tool broadly fits the aim set out, and the group added significant contextual data to the original dataset through word analysis. In general we struggled to finish the project in time, which is reflected in the overall website styling and the crude user interface of the visualisation tool.

Whilst we had been prepared for work adding the medium and classification data, the cleaning of the countries was unexpectedly difficult and time-consuming and likewise city-level data was a lengthy task that had to be abandoned at a late stage. The spatiotemporal clustering was also far more difficult than expected from a purely computational standpoint, requiring time-consuming programming optimisation. Concentrating on a smaller dataset with as rich data as possible may have been a better approach from the point of view of time management and undertaking a clustering analysis.

With additional time we would primarily improve the visualisation tool by adding city level data and the ability to zoom in to that scale; displaying the classifications and mediums as graphs to use Crossfilter's exploratory nature to its full strength; and improving the UI through having a switchable sidebar from "graphs and filtering" to "records and object browser". We would improve the overall website to have a consistent style which flows into the tool. We would also try to rerun the clustering with richer data (city level data, artist data etc.).

References

Ahmad, A., Khan, S.S., 2019. Survey of State-of-the-Art Mixed Data Clustering Algorithms. IEEE Access 7, 31883–31902.

Axiell., 2016. “Industry Report: Digital Transformation in the Museum Industry.” *Axiell: Archives Library Museums*, 17 Apr. 2019, www.axiell.com/report/digital-transformation-in-the-museum-industry/.

Best, Ryan A., 2018. GitHub source code. https://github.com/ryanabest.

Boogaart, T.A., 2001. The power of place: From semiotics to ethnogeography. *Middle States Geographer*, *34*, pp.38-47.

Cairns, S., 2013. Mutualizing museum knowledge: Folksonomies and the changing shape of expertise. *Curator: The Museum Journal*, *56*(1), pp.107-119.

Cameron, F. and Robinson, H., 2007. Digital knowledgescapes: Cultural, theoretical, practical, and usage issues facing museum collection databases in a digital epoch. *Theorizing digital cultural heritage: A critical discourse*, pp.165-191.

Crossfilter [WWW Document], n.d. URL https://square.github.io/crossfilter/ (accessed 5.22.19)

Dumas, B., Moerman, B., Trullemans, S. and Signer, B., 2014, May. ArtVis: combining advanced visualisation and tangible interaction for the exploration, analysis and browsing of digital artwork collections. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (pp. 65-72). ACM.

Gower, J.C., 1971. A general coefficient of similarity and some of its properties. Biometrics 857–871.