GSOC 2017 – Numfocus (Matplotlib)

W M D Kanchana N Ranasinghe

1 Categorical Color

1.1 Abstract

Categorical information (non-numerical like gender, favorite food, etc.) is quite common in datasets that we analyze. However Matplotlib support for plotting data is available mostly for numerical data, with few direct APIs built for categorical data. This project attempts to build an API for plotting heat-maps and other scalarMappables using categorical data. It will also work on developing normalization and color-map APIs supporting categorical data as these will be dependencies for the main API.

Currently, users wanting to plot categorical data with color support in Matplotlib must manually map their categorical data to numbers, creating non-existent links between classes and requiring a lot of additional effort.

Intensity	**Involves**	**Mentors**
Intermediate	Python, Data science	[@tacaswell][@story645]

1.2 Technical Details

With the widespread presence and usage of large datasets of varied nature, the analysis of categorical data has received much attention. Most machine learning, big-data analysis, and data-mining research and projects require the datasets available to be visualized at various stages of work. A variety of plotting techniques are utilized for this matter, and Matplotlib plays a prominent role as an essential tool for many researchers. Despite its diverse functionality with regard to numerical data visualization, Matplotlib is quite restricted in the domain of categorical data analysis. In most cases, user are compelled to manually map their categorical data to numerical data, thus introducing unnecessary relationships and patterns to the datasets, creating redundant data, and being withheld from the simplicity and ease of usage inherent to Matplotlib.

This work hopes to tackle this issue to a certain extent. The Categorical Color Project will include two main tasks:

1) working on heat-map and other scalarMappables APIs
2) working on normalization and color-map APIs.

A main aim of this project is to integrate unit support into these APIs without subjecting them to any direct changes. Matplotlib currently has these frameworks set-up with functionality for numerical data. This functionality will be extended to support categorical data. The work of this project will be done solely using Python. Libraries used will include Numpy and Matplotlib. There will be no new (unused in Matplotlib previously) third-party libraries used in the course of this project.

1.3 Schedule of Deliverables

1.3.1 May 1st - May 28th, Community Bonding Period

- Working on understanding categorical data support for imshow function
- Self-implementation of previous work done to support categorical color for better understanding
- Setting up of project blog covering all major technical details
- Working on MEPs in Matplotlib Developers' Guide, and Matplotlib issues related to normalization, color-map, and color-bar code.

1.3.2 May 29th - June 3rd

- imshow basic functionality including tests and documentation

The work of this project will stem from existing development done for categorical color support in Matplotlib issues 6889 and 7383. The starting point for the project will be extending the imshow function (matplotlib.axes._axes.imshow) to support categorical data. This mainly focuses on providing it unit knowledge with regards to non-numerical data. The approach followed here will be the initial basis for this extended support for categorical data. It will be used as a foundation to build the heat-maps API (the imshow function). This will also lead to the normalization and color-map APIs.

1.3.3 June 5th - June 9th

- color-bar support for imshow

The color bar is mapped from numerical data for heat-maps currently, creating boundaries for each class. Due to discrete nature of categorical data, ListedColormaps will be used and the boundaries required for each distinct stage will be extended to support non-numerical values. The string values of categorical data will be directly mapped to each boundary class by the units integration.

1.3.4 June 12th - June 16th

- legend support for imshow

This is an additional improvement attempted. Since heat-maps do not currently have a direct method for implementation of a legend, this will focus on extending the legend support of other scallarMappables. Also the categorical support for legend here will mostly be for the purpose of serving as a foundation for legend support for other scallarMappables.

1.3.5 June 19th - June 23th, End of Phase 1

- testing and debugging of imshow function (heatmap API)

1.3.6 June 26 - June 30th, Begin of Phase 2

- framework outline for norm functionality extensions
- norm basic functionality including tests and documentation

With regards to the normalization function, in a Matplotlib sense, it encompasses a function mapping different data values to different colors (or color levels). Currently, the different data values to be mapped from are expected to be numerical. This limitation will be worked around to enable support for categorical data (string data). Also during the course of the project focus will be given to the currently available discrete

normalization methods (matplotlib.colors.BoundaryNorm) which handle data in a different sense which may be used as a foundation for this additional unit support.

1.3.7 July 3rd - July 7th

- framework outline for cmap functionality extensions
- cmap basic functionality including tests and documentation

Currently, the process of using matplotlib.pyplot for heat-maps and other scalarMappables includes the creation of a color-map followed by normalization of all available data to fit this color-map. Color-maps plotting functionality is threefold: sequential, divergent and qualitative. Considering the intrinsic discreteness of categorical data, the extension of the color-maps API will focus on its qualitative plotting ability. However the fundamental behavior of the color-maps API will not require any changes for handling categorical data as our approach simply involves the integration of unit support. Therefore it is possible to build this extended support for categorical data while keeping the public API intact.

1.3.8 July 10th - July 14th

- integration of norm and cmap into heatmaps

Given the functionality extended for categorical data, the final task is entwining all this functionality to imshow to allow the plotting of heat-maps from non-numerical values. This would be handled here.

1.3.9 July 17th - July 21th, End of Phase 2

- testing and debugging of norm and cmap APIs

1.3.10 July 24th - July 28th, Begin of Phase 3

- identifying possible scalarMappables to extend categorical color support
- framework outline for categorical color support of other scalarMappables

Identification of other scalarMappables like matshow, pcolor, pcolormesh, scatter, etc. will be done here. Possibilities to extend the current work to support categorical data will be analyzed. The work here will not be direct extensions of heat-maps but will require different approaches to extend the units support for these scalarMappables.

1.3.11 July 31st - August 4th

- basic functionality of other scalarMappables

1.3.12 August 7th - August 11th

- overall functionality of other scalarMappables with tests and documentation

1.3.13 August 14th - August 18th

- testing and debugging of other scalarMappables code

1.3.14 August 21st - August 25th, Final Week

- testing of all code and bug fixing
- final review of all code and documentation

1.3.15 August 28th - August 29th, Submit final work

- code submission
- discussion of possible future developments on blog

1.4 Future works

- Improving the APIs built by considering different approaches for categorical data integration

1.5 Development Experience

I have been involved in programming work in python, MATLAB, Lua, Java, C and Micro C over the past three years. Some of my work is on github. With regards to open source development, I have worked on a couple of issues on matplotlib and scikitlearn over the past few weeks. In addition, I was able to get involved in this natural language processing related project for CLTK on Pali Language (mainly because I am familiar with the language). The links of work done are below.

https://github.com/matplotlib/matplotlib/pull/8371#event-1015235492

https://github.com/matplotlib/matplotlib/pull/8357

https://github.com/scikit-learn/scikit-learn/pull/8558#pullrequestreview-26737369

https://github.com/kahnchana/Pali-NLP

1.6 Other Experiences

I have been involved in some machine learning and computer vision related research work, especially regarding activity recognition in videos. Currently I am involved in some work related to dense trajectories. I have also been involved in numerous mathematics related work since an early age, having also represented my country at the International Mathematical Olympiad while in high school.

1.7 Why this project?

I am quite interested in the areas of machine learning, computer vision, and robotics. As part of a machine learning research group at my university, I have constantly used Matplotlib for various purposes, including visualizing of certain categorical data. So I feel that I should try to get involved in contributing to the community as well. Also I feel a lot familiar with this project and find it interesting to work on mainly due to its relevance to data science.

1.8 Appendix

1.8.1 About Me

I am a second-year undergraduate student at the University of Moratuwa, Sri Lanka studying Electronics and Telecommunication Engineering. I have been using python for the last three years for development and

research work and am quite familiar with Numpy and Matplotlib libraries.

1.8.2 Contact Details

W M D Kanchana N Ranasinghe

kahnchana@gmail.com / 150507H@uom.lk

kahnchana@github.com

1.9 Availability

I will be having summer holidays from the first week of June till the first week of September so I will be able to commit completely towards GSoC during that period.

- Time zone: Sri Lanka Standard Time (SLST) - UTC +05:30
- Hours per week: 35 – 40 hours (except for first week of June – roughly 30 hours)