The Many Faces of Multivariate Data: Chernoff Faces

Liliana Matos

Advisor: Norm Badler

University of Pennsylvania

ABSTRACT

Humans have the ability to discern subtle differences in facial expressions. What if multivariate data were represented in the form of a facial expression? Would it be easier to compare data sets? That is the argument behind Chernoff faces, an idea conceived by Herman Chernoff. Traditionally, Chernoff faces are a series of 2D shapes and lines that imitate facial features i.e. a triangle for a nose, a circle for an eye etc. However, Chernoff outlined only a concept; he did not specify an algorithm to implement this idea.

Given a data set, each variable would be mapped to a particular facial feature and morphed accordingly. The ultimate goal is to create a web-based application that takes in multiple data sets and generates a Chernoff face for each set in order to juxtapose the data.

Project Blog: lilianasnrdes.wordpress.com

Live Project: lilianamatos.com/chernoff/

1. INTRODUCTION

Bar graphs, pie charts, scatterplots, infographics. Today, data visualization takes many forms and is crucial to our understanding of information. Still, each new representation sports a small, but vital learning curve. How can we visualize data so that its interpretation is practically intuitive to the viewer?

User interface and user experience design go hand in hand in that the success of one depends on the other. From tables to infographics to interactive applications, data visualization has evolved tremendously, both aesthetically and functionally, to reach and inform a larger audience. Pragmatically, if information can be presented in intuitive and interesting fashion, its dissemination occurs effortlessly.

The solution I propose is an application that represents multivariate data in the form of facial expressions—Chernoff faces. Given a dataset, the application will map each variable to a facial feature and morph it according to the value of that variable. The ultimate goal is to represent a collection of data sets that can be interpreted and juxtaposed effortlessly.

This project will seek to implement and contribute the concept of Chernoff faces with live data.

1.1 Design Goals

The target audience consists of artists, students, web enthusiasts, and individuals interested in data visualization.

1.2 Projects Proposed Features and Functionality

The project will be in the form of an interactive web-based application. Features of the application include:

- Change set of data
- Dynamically morph images
- Compare sets of data

2. RELATED WORK

There are various implementations of the traditional Chernoff faces with geometric shapes.

Chernoff H. [Che73] first introduces the concept of what is now known Chernoff faces. This literature discusses the graphical method of representing points in k-dimensional space to represent multivariate data. The reading provides several diagrams displaying iterations of Chernoff faces and provides a baseline for facial feature ranges and parameters.

Ebert D. S., Morris C. J., and Rheingans P. [EMR99] investigates the effective of Chernoff faces in information visualization through the performance of several experiments. The study concluded that the Chernoff faces may not have significant advantage over other visualization techniques. I expect to overcome this obstacle through the use of images of human faces as opposed to geometric shapes arranged to mimic a face.

Flury B., Riedwyl H. [FR81] proposes an alternative to Chernoff faces—asymmetrical faces—that have different face parameters, which correspond to distinct facial features. The study concluded that the choice of graphical method should depend on the nature of the variables.

Research & Development Labs [R&D10] of the *New York Times* created the Cascade project, a data visualization that analyzes the structures that underlie the digital sharing of

news and stories. This visualization was the inspiration for this project and will serve as point of comparison in the evaluation of the final web-based application deliverable.

3. PROJECT PROPOSAL

3.1 Anticipated Approach

The first step of the project will utilize an existing code framework in C++ to implement a working version of the Beier-Neely algorithm.

The second step will manipulate the previously created algorithm to create an application that given an Excel data set will selectively morph certain facial features of an image.

The final step will duplicate the functionality of the previous step in a web-based application that will dynamically morph an image given a dataset from an external web source.

3.2 Target Platforms

I will implement the initial algorithm using the Visual Studio platform in C++ to leverage existing libraries and then transfer it to the web in JavaScript.

3.3 Evaluation Criteria

Each step of the project will be evaluated separately. Subsequent steps are implemented if and only if the previous step is successful.

The first step, the effectiveness of the Beier-Neely image morph, will be evaluated through comparison of prior implementations. The clarity of the data representation of the facial features in the second step will be evaluated by data collection through comparison survey of other forms of data visualization.

The last step will be evaluated through the comparison of the external source data representation with that the web-based application generates.

4. RESEARCH TIMELINE

Project Milestone Report (Alpha Version)

- Completed all background reading

Project Final Deliverables

- Web application
- Documentation

Project Future Tasks

- Perform user-testing of several iterations of Chernoff faces to find most “readable” morph
- Data-visualizations with numerous distinct collections of datasets

5. Method

I decided against using the Beier-Neely algorithm and coding in C++. Instead, I went directly into working in JavaScript. I pulled the data from the Penn Course Review Site and mapped the review parameters to distinct facial features:

Mappings	
Parameter	**Facial feature**
rAmountLearned	Size of head
rReadingValue	Width of head
rCommAbility	Size of mouth
rInstructorQuality	Orientation of mouth
rCourseQuality, rDifficulty	Eyes
rInstructorAccess	Ears
rRecommendMajor	Nose
rRecommendNonMajor	Cheeks
rStimulateInterest	Eyebrows
rWorkRequired	Hair

I utilized the JavaScript drawing library, PaperJS, to draw and morph the head and facial features. Each parameter ranges from 0 to 4, 0 being the worst and 4 being the best. The facial feature morphed accordingly to its respective parameter.

6. RESULTS

I successfully created a web application that displays multivariate data using Chernoff faces.

7. CONCLUSIONS and FUTURE WORK

In conclusion, the application proved to be successful in displaying the data.

In the future, I would like to give the facial features a less rigid look and give them a more organic feel. Furthermore, I would make the changes more drastic from 4 to 0. While 0 is much worse than 1 or 2, students would be averse to taking a course with a rating less than 3.

APPENDIX

References

[Che73] CHERNOFF H.: The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association*, Vol. 68, No. 342. (Jun., 1973), pp. 361-368.

[EMR99] EBERT D. S., MORRIS C. J., RHEINGANS P.: An Experimental Analysis of the Effectiveness of Features in

Chernoff Faces, *University of Maryland Baltimore County*, (1999).

[FR81] FLURY B., RIEDWYL H.: Graphical Representation of Multivariate Data by Means of Asymmetrical Faces, Journal of the American Statistical Association, Vol. 76, No. 376 (Dec., 1981), pp. 757-765.

[R&D10] RESEARCH & DEVELOPMENT LABS: Cascade, http://nytlabs.com/projects/cascade.html, *The New York Times*, (2010).

#	Task Name	Jan	Feb	March	April
1	Data collection				
2	Beier Neely implementation				
3	Algorithm testing				
4	Algorithm evaluation				
5	Application creation				
6	Chernoff implementation				
7	User study				
8	Chernoff iteration				
9	Application evaluation				
10	Web application implementation				
11	External source selection				
12	Web application design				
13	Website design				
14	Evaluation/user testing				
15	Revision				

Figure 1: *Project timeline.*