Supplemental Materials for
Duet: Helping Data Analysis Novices
Conduct Pairwise Comparisons by Minimal Specification

Contents

Links to Videos Introducing Duet's User Interface

Tutorial Video Used in the User Study:
https://youtu.be/JF8Q-PT3xUY

Analyzing US College Scorecard Data Using Duet:
https://youtu.be/Y4p0h8_EnDU

These videos can also be found in the "Videos" folder of the supplemental materials.

Link to the System

Duet's prototype:
https://duetpaircomp.github.io/

README:

1. Use Google Chrome for better experience.
2. Some datasets are provided in the "Datasets" folder for trying out the system.

Explanations for the "Logistic Regression" Folder

In the following, we explain the materials in the "Logistic Regression" folder. The "Logistic Regression" folder provides details for the model in Sec. 5.3.2 (Multinomial Logistic Regression for Classification) of the paper.

1. 520 Distribution Pairs

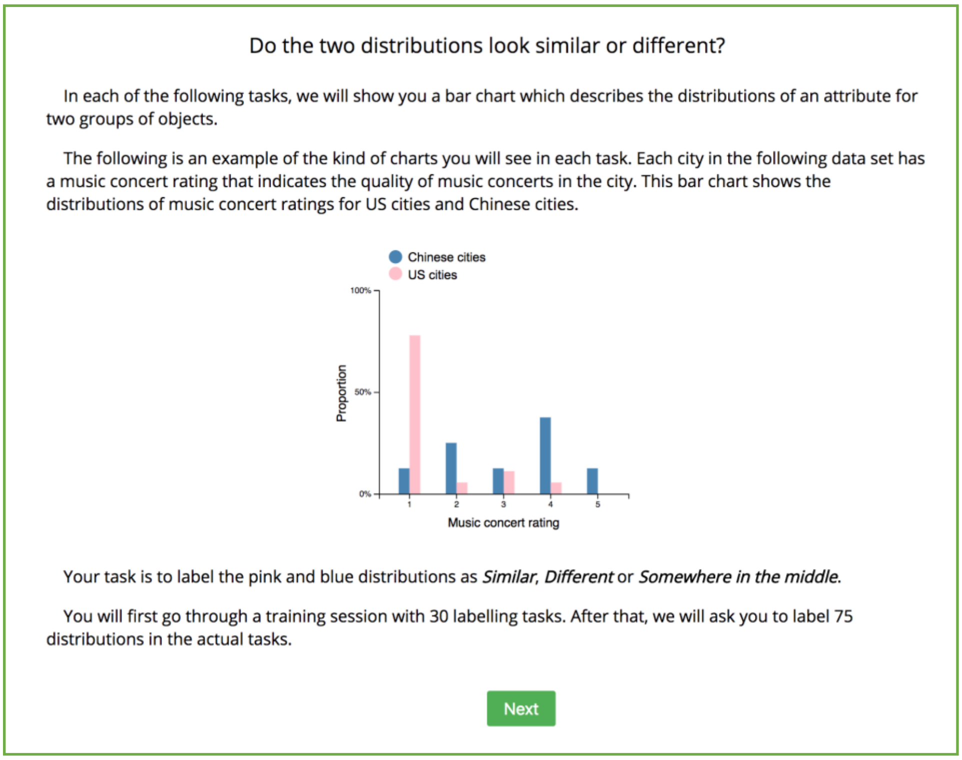
This folder contains 520 distributions pairs we collected from the 83 R datasets (https://vincentarelbundock.github.io/Rdatasets/datasets.html). Each row in a csv file is a data point. There are three important columns in each csv file: "newGroupName", "newAttributeName" and, "attributeValue". "newGroupName" is the name of the group to which a data point belongs, "newAttributeName" is the name of an attribute. "attributeValue" is the value that a data point has for the attribute.

2. Bh Coefficient + Labels.csv

We used SPSS to model the data. This csv file is the input to SPSS for modelling. The "fileName" column is the file name of a distribution pair inside the "520 Distribution Pairs" folder. "BhCoefficicent" is the Bhattacharyya coefficient for a distribution pair and "class" is the label of distribution pair we collected from people.

3. Code for Relabelling 150 Marginal Cases

It is the code of the interface we used for asking 10 subjects to relabel 150 marginal cases. You need Flask to run the interface. To run the code, go to the directory using the console if you are using a Mac and enter "python server.py". For those who have difficulties running the tool, we provide the screenshots of the labelling tool as follows:



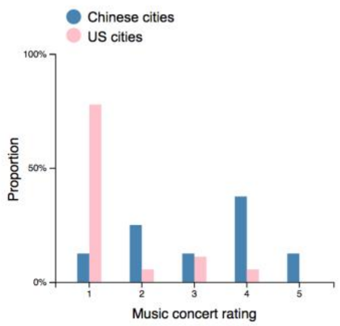
Here's how it'll go...

Imagine that you are analysing a dataset. You would like to tell your supervisor, Bob, some interesting findings after the analysis.

During your analysis, you see the following bar chart. Again, the following bar chart shows the distributions of music concert ratings for US cities and Chinese cities. Try to move your mouse cursor over a bar in the bar chart to see the percentage of cities that have a particular value of musical concert rating.

Would you say that the blue and pink distributions are *similar*, *different* or *somewhere in the middle*?

Choose your answer and click the next button to continue.

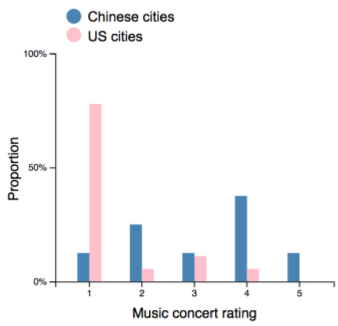


Your answer:

- Similar
- Different
- Somewhere in the middle

Back Next

Here's how it'll go...



As a rule of thumb, if you think that, after your analysis, you will tell your supervisor, Bob that the two distributions look really similar, you should label the distributions as *Similar*.

If you will tell Bob that music concert rating is a distinguishing attribute for US cities and Chinese cities, you should label the two distributions as *Different*.

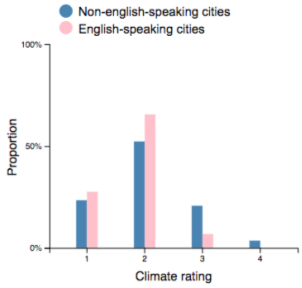
If you can't tell whether the distributions are similar or different enough or if you are not sure whether you will tell Bob about the two distributions, you should label them as *Somewhere in the middle*.

Click the next button below to see some examples.

Back Next

Here are some examples

The following shows the distributions of climate ratings for English-speaking and non-English-speaking cities (each city has a climate rating). You can see that English-speaking cities and non-English-speaking cities have fairly similar distributions of climate ratings. For both groups, many cities get a rating of 2 and fewer cities get a rating of 1 or 3. You probably want to tell Bob about the finding and label the following distributions as *Similar*.



On the other hand, looking at the following distributions, you observe that non-English-speaking cities seem to have a high corruption rating on average and English-speaking cities seem to have a low corruption rating on average, which sounds really interesting. You may want to tell Bob that corruption index is a distinguishing attribute for English-speaking cities and non-English-speaking cities and therefore label the following distributions as *Different*.

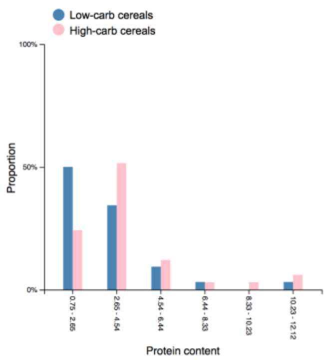
Again, if you are not sure whether the two distributions are similar or different enough for you tell Bob about them, you should label the distributions as *Somewhere in the middle*.

Click the next button to start the training session.

Back Next

Do the two distributions look similar or different?

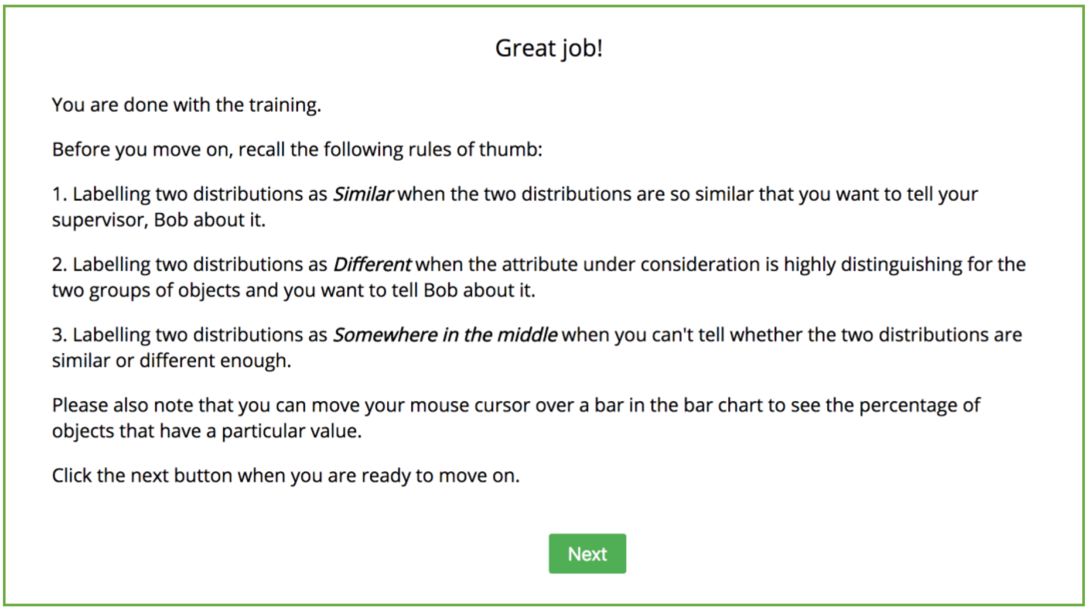
Practice Question 1 of 30

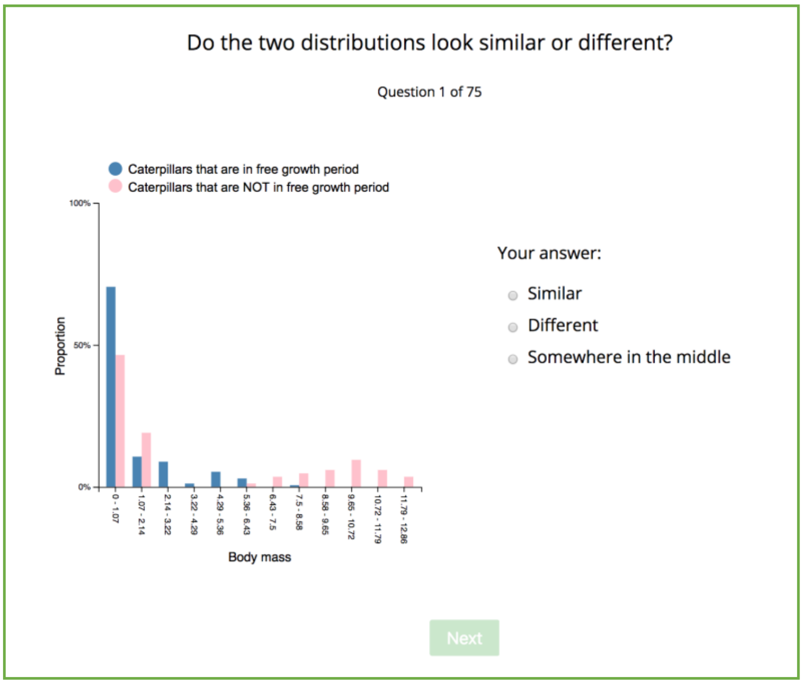


Your answer:

- Similar
- Different
- Somewhere in the middle

Next





4. Data Collected From Relabelling

Each csv file in the folder contains two columns: “filename” that is the file name of a distribution pair in the “520 Distribution Pairs” folder and “class” that is the label provided by a subject.

5. SPSS Modelling Result

It is the screenshot of the output generated by SPSS. “Bh Coefficient + Labels.csv” is used as the input to SPSS. The following explains how the model in Sec. 5.3.2 corresponds to the SPSS output.

Multinomial logistic regression analysis shows that Bh coefficient significantly predicts the label ($\chi^2(2, N = 520) = 596.769, p < .001$). Formally, our logistic regression model is

$$P(S) = \frac{\exp(34.066Bh - 31.408)}{1 + \exp(34.066Bh - 31.408) + \exp(-18.310bh + 15.125)}$$

$$P(S) = \frac{\exp(-18.310Bh + 15.125)}{1 + \exp(34.066Bh - 31.408) + \exp(-18.310bh + 15.125)}$$

$$P(M) = 1 - P(S) - P(D)$$

Model Fitting Information

Model	Model Fitting Criteria: -2 Log Likelihood	Likelihood Ratio Tests: Chi-Square	df	Sig.
Intercept Only	1107.124			
Final	510.354	596.769	2	.000

Pseudo R-Square

Cox and Snell	.683
Nagelkerke	.774
McFadden	.538

Parameter Estimates

class[a]		B	Std. Error	Wald	df	Sig.	Exp(B)	95% Confidence Interval for Exp(B): Lower Bound	Upper Bound
D	Intercept	15.125	1.821	68.970	1	.000			
	BhCoefficient	-18.310	2.173	70.974	1	.000	1.117E-8	1.577E-10	7.907E-7
S	Intercept	-31.408	3.896	64.990	1	.000			
	BhCoefficient	34.066	4.163	66.959	1	.000	6.235E+14	1.783E+11	2.180E+18

a. The reference category is: N.

6. R Code for Computing Model Accuracy.txt

This text file contains the R code for computing the cross-validation accuracy of our logistic regression model using 10-fold cross validation. The input file is “Bh Coefficient + Labels.csv”. The cross-validation accuracy is around 78.1%. We envision that this accuracy can be improved by using more advanced machine learning models and more predictor variables.

Explanations for the “User Study” Folder

The “User Study” folder contains all the materials for the qualitative user study in Sec. 6 (Evaluation) of the paper. The materials inside are described as follows:

1. **Training Session**

   This folder contains the car dataset “cars.csv” we used for the training session, a link to the tutorial video we showed to the participants and the training tasks to get participants familiar with Duet’s interface.

2. **Analysis Session**

   During each analysis session, we first showed the participants “Task Description.pdf”. We then gave them some time to review either “Description for College Dataset.pdf” or “Description for City Dataset.pdf” to get them familiar with the dataset they were about to analyze. The “Datasets” folder contains the city dataset and the college dataset we used for the analysis session.

3. **Interview and Survey**

   At the end of the study, we first showed them “Three Main Features of the Tool.pdf” to ensure the participants know the terminology like “minimal specification” we are going to use in the interview. This folder contains the questions for the semi-structured interview (“Interview Questions.pdf”) and the survey questions (“Survey Questions.pdf”).

4. **Questionnaire Results.pdf**

   It is a summary of the survey result.

Clarification of Literature Review

We drew inspiration from the literature to develop the idea of minimal specification. As described in the paper, there are two high-level considerations in designing minimal specification:

1. To address ***execution barriers***, minimal specification allows users to focus on what they know (the objects of interest in answering a pairwise comparison question) rather than what they might not know (system operations).

2. To address ***interpretation barriers***, the recommendations offered should be explained in order to result in better understanding of the recommendations and stronger feeling of trust.

The following two sections describes the basis of these two components.

Addressing Execution Barrier

This idea of allowing users to focus on *what they know* by shielding them from *what they might not know* is grounded in the following three ideas that have been explored by the HCI community:

Idea	Programming by examples
Reference	1. Kandel, Sean, et al. "Wrangler: Interactive visual specification of data transformation scripts." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011. 2. Mayer, Mikaël, et al. "User interaction models for disambiguation in programming by example." *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015. 3. Yessenov, Kuat, et al. "A colorful approach to text processing by example." *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013.
How does it relate to minimal specification?	The key idea of programming by demonstration is to allow users to provide examples of the text they want to extract (*what users know*) so that do not need to write the scripts for text extraction (*what novice users might not know*).

Idea	Interrogative debugging
Reference	1. Ko, Andrew J., and Brad A. Myers. "Designing the whyline: a debugging interface for asking questions about program behavior." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2004. 2. Ko, Andrew J., and Brad A. Myers. "Finding causes of program output with the Java Whyline." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009.

How it relates to minimal specification?	The key idea of interrogative debugging is to allow programmers to debug their programs by asking why and why not questions about their programs' failure (*what programmers know*) so that they do not need to find strategies for answering these questions using debugging tools' limited capabilities (*what programmers might not know*)

Idea	Natural language for data visualization
Reference	1. Gao, Tong, et al. "Datatone: Managing ambiguity in natural language interfaces for data visualization." *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015. 2. Setlur, Vidya, et al. "Eviza: A natural language interface for visual analysis." *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016. 3. Srinivasan, Arjun, and John Stasko. "Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks." *IEEE transactions on visualization and computer graphics* 24.1 (2018): 511-521. 4. Hoque, Enamul, et al. "Applying Pragmatics Principles for Interaction with Visual Analytics." *IEEE transactions on visualization and computer graphics* 24.1 (2018): 309-318.
How does it relate to minimal specification?	The key idea of these natural language interfaces is to allow data analysts to directly state their questions (*what users know*) so that they do not have to learn the interface or translate their questions into system operations (*what users might not know*).

Addressing Interpretation Barrier

Explaining the recommendations help users understand why they are recommended and inspire users' trust in the system. This idea is grounded in the movement of explainable artificial intelligence (XAI).

Idea	Explainable artificial intelligence (XAI)
Reference	1. Explainable Artificial Intelligence — https://www.darpa.mil/program/explainable-artificial-intelligence. [Accessed: 31th March 2018]. 2. Workshop on Explainable Smart Systems — http://explainablesystems.comp.nus.edu.sg. [Accessed: 31th March 2018]. 3. Lim, Brian Y., Anind K. Dey, and Daniel Avrahami. "Why and why not explanations improve the intelligibility of context-aware intelligent systems." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009. 4. Kulesza, Todd, et al. "Principles of explanatory debugging to personalize interactive machine learning." *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 2015.

How does it relate to minimal specification?	The motivation behind explainable artificial intelligence is that a lot of machine learning models and decisions made by artificial intelligence applications are hard to understand. To promote trust and understanding of artificial intelligence, decisions made by ML models and artificial intelligence should be explained. As novice data analysts might grapple with understanding why Duet recommends certain groups and attributes, we strive to explains these recommendations to reduce interpretation barrier.