Computing for Mental Health

Ji Won Chung, Isha Raut, Subashini Sridhar, Ji Young Yun, Kelly Pien

Abstract - Historically, the fields of Human Computer Interaction (HCI) and Human Services have been disparate studies. Professor Crouser collaborated with mental health clinicians at the Justice Resource Institute (JRI) in order to integrate technology to optimize the work of JRI. One of the problems identified by the clinicians was a diagnostic bias stemming from a clinician's specialization in a field. Another bias is because of the DSM-5's organization. Our system, based on symptoms and diagnoses taken from the DSM-5, aims to reduce this bias through a set of binary questions about a patient's symptoms. The system is divided into two parts: (1) a clinician interface that serves as a tool to suggest to the clinician alternative diagnoses via utilization of an interactive filtering system in D3.js and (2) a research interface to change the machine learning models via a Shiny app in R.

I. Introduction

The Diagnostic and Statistical Manual of Mental Disorders (5th ed.; *DSM–5*; American Psychiatric Association, 2013) classifies mental health conditions and the criteria for each diagnosis. The DSM-5 is a book that aids mental health clinicians in making diagnoses and determining treatments. It is organized into 20 chapters, each of which lists possible diagnoses of a particular type, such as neurodevelopmental disorders, depressive disorders, and anxiety disorders. However, the DSM-5 originally was not created to diagnose people but list a set of symptoms. In addition, the DSM-5 was created part by part. Select and separate groups focused on specific parts of the chapter, not across chapters. Therefore, this inherent, organizational bias of the DSM-5 is one problem we try to resolve.

Clinicians are also subject to specialization and diagnostic bias. A clinician may have a tendency to look at a certain chapter depending on their field of expertise. For example, if a clinician is trained in the field of bipolar disorder, the clinician is inclined to look for a diagnosis in that particular chapter. As a result, the clinician is most likely to diagnose a condition from that chapter. However, sometimes a diagnosis in a different chapter might be more appropriate. For instance, if a clinician's patient had difficulty sleeping, the clinician might turn to the "Sleep-Wake Disorders" section of the DSM-5 and assign a diagnosis from that chapter. Yet "persistent reluctance or refusal to sleep away from home or to go to sleep without being near a major attachment figure" is also a symptom of separation anxiety disorder, which is in the "Anxiety Disorders" chapter. But because the clinician did not turn to the "Anxiety Disorders" chapter, the clinician would not have been presented with all possible disorders - and thus might make a nonoptimal diagnosis. Clinicians are also more likely to turn to the chapter of the area in which they specialize.

Our system helps mental health clinicians more accurately diagnose mental health conditions without diagnostic or specialization bias.

II. Data Collection

The data came from the online version of the DSM-V. Using the Python package BeautifulSoup, a Python program scraped the DSM-5. We then manually collected 207 symptoms and the diagnoses each were associated with. We printed them and cut them up into small pieces of paper. With the help of clinicians from the Justice Resource Institute, we put similar symptoms in piles. For example, all symptoms relating to sleep, such as insomnia, fatigue, and lethargy, were grouped in one pile. Each pile became a category of symptoms, such as “sleep”.

A survey, which was sent to clinicians around the country, refined the resulting 29 categories. The survey asked clinicians if the symptoms in each categories were correctly classified. If they were not, the clinician was asked to suggest the correct placement in a different category or in a new one of their own creation.

We determined whether or not each of the 29 categories was a criteria for each diagnosis. 1s indicated that the patient should exhibit a symptom from this category to be diagnosed with the condition, and 0s indicated the symptom should not be present in a patient with this condition. These 0s and 1s are concatenated into a binary code, representing the category criteria assigned to each diagnosis.

Each diagnosis is identified by a combination of letters and numbers beginning with F, G, L, or R and ending with an underscore and the DSM-5 chapter it can be located in. The diagnosis code and corresponding binary code were written out to the pivot.csv file. Not much data cleansing was done during the implementation of the interface. The only changes made were to the “160720 Pivot.csv” (renamed “data.csv”) in which the names of the headings were slightly modified so they would be easily compatible with d3 and JavaScript (i.e. self-harm was changed to selfharm).

III. Data Analysis

RepeatChapters.py scans the pivot.csv file, collects all of the diagnoses with the same binary code (or those diagnoses that our decision tree is unable to differentiate), and puts these in a dictionary. For each DSM-5 chapter, RepeatChapters.py counts the number of times diagnoses from that chapter appear in the non-distinguishable diagnoses dictionary. These results are written out to outputfile.csv.

The visualization of the clinician interface did reveal some fascinating insights. For example, it was not apparent on the csv file that concentration did not have a role in splitting the data. However, in the visualization the concentration bubble is gray from the program’s launch. Therefore, it indicates that there is no diagnosis determined by concentration.

IV. Visualization Techniques

Our interface has two components. One is designed for clinicians and the other for researchers. The clinician interface features an interactive bubble chart that allows the clinician to filter data at their discretion. The researchers' interface is for those who would like to learn more about the system's limitations and development process. Both interfaces are intended to be user-friendly. The former interface is easy to maneuver through with little to no training. However, the latter must require some technical expertise in machine learning in order to interpret the data.

A circle is an intuitive shape that is easy to understand for users without a technical computer science background. A layperson possesses an inherent schema of a circle. This interface assimilates the inherent characteristics of a circle such as its area (size) and color to represent data. In addition to these intrinsic qualities, the interface adds operations to the circle such as clicking and hovering to interact with and demonstrate the data. These functions do not compromise the native attributes of a circle but are additions to it. Inspired by the power of a circle, this interface implements a bubble chart, a data visualization represented by circles.

The bubble chart was implemented for the following reasons: (1) it is an intuitive type of visualization that requires little to no user training, (2) it is a compact way of showing a variety of categorical and quantitative information, (3) it enables different data types to be grouped, compared, and contrasted.

Each bubble is labeled with one category of symptoms a clinician might observe in a patient. The area of each bubble is representative of the frequency that symptom appears over the entire filtered set of diagnoses. To represent the frequency, a form of quantitative data, numerically would be uninformative to the user because it would provide too much information. Thus, the area was used to represent frequency because it concisely and intuitively communicates that a larger area corresponds to higher importance.

The bubble is also connected to simple, intuitive handlers such as hovering, double clicking, and right-clicking. Upon hovering over a bubble, the user can see two features: (1) a black pop-up on the bubble containing the clinician-generated description of the category and (2) a question that appears on the top left of the screen. In response to the question, the user can right-click a bubble to answer yes or no. The selection of 'yes' will make the circle turn red and fix its area to 10000*1. The selection of 'no' will turn the circle blue and change the area to 10000*frequency, where frequency = (number of diagnoses with that category present)/(the number of total filtered diagnoses). Note that 10000 is a scale for visualization. A user can double click a circle to deselect it. The user's selections are automatically shown in the box at the bottom right corner. Upon clicking the "What are the potential diagnoses so far?" button, the user can see the list of filtered possible diagnoses with their corresponding chapters and descriptions.

The types of circles are divided into the following three categories by color: (1) red indicates that the user has selected the bubble, (2) blue indicates an unselected bubble, (3) gray indicates there is no data to be filtered based on that symptom. The blue and red bubbles look more transparent than the matte, gray ones. By design, this contrast in perceived opacity highlights the difference between a circle that the user can interact with and one the user cannot. Because red commonly denotes importance and sharply contrasts with blue, it was chosen to indicate a selection. To facilitate the user's navigation of the interface, a legend is located on the top left of the bubble chart to indicate what the three colors represent.

The layout of the website was structured such that qualitative data could be viewed in an easy, clean manner. The user can see all components of the interface without scrolling down pages. In addition, the text box that lists the potential diagnoses has a scroll option so the user does not have to have a clutter of qualitative descriptions that extend further down the page. The text boxes surround the bubble chart in an orderly manner to not distract from the data visualization. The text box's initial default setting contains minimal text with descriptions of their purposes in order to not overload a user with data and information for his or her initial use. The question box was placed on the top left side in alignment with the bubble chart so the user could focus on the left side of the screen during the interaction with the bubble chart. The rest of the two boxes were placed on the right to minimize number of times the user must look across the screen and to facilitate the process of comparing the qualitative data with the visualization on the bubble chart. We chose a modern font so the data would not look cluttered. The light sky blue headings was designed to prevent the user from being distracted from the actual interface. The black tabs were created such that there colors were distinct enough to indicate their purpose and not be comparable to the colors used in the bubble chart.

Another part of the clinician interface is a collapsable decision tree that consists of all the symptoms. A collapsable decision tree was used mainly because the final diagnosis depends on the chosen symptoms. All the links in the decision tree represent the different paths made up of the symptoms, which vary depending on the user's choices. As the user selects certain symptoms and answers 'yes' or 'no,' the final diagnosis results are taken from the child of this decision tree.

The researchers' interface explores and analyzes the data behind the clinician interface. The clinician interface is broken down into three key parts. A Data Filter tool allows researchers to filter and select the data desired. The Interactive Decision Tree allows researchers to tune, prune, and view the results of the decision tree model from the Rpart and Party package. The Symptom & Chapter Similarity sections shows the results of a Random Forest and our own chapter similarity algorithm to show which symptoms or chapters are often likely to get misclassified.

The homepage or the About page of the interface takes users to a brief description about the DSM-5 and the Mental Health Interface. The tab for the Chapters of DSM on the About page directs researchers to the list of 19 chapters and their titles in the DSM. The tab for the Symptom

Categories on the About page leads researchers to the list of 32 symptoms and their definitions. Both these pages have a search box that allows users to filter by keywords.

Under the DSM-5 tab, researchers can explore the pivot.csv file in detail. The interface provides the option to select multiple columns from the pivot file which include the diagnostic code, chapter in DSM-5, and various symptoms. On updating the selection, users will see all rows from the original dataset with only the subset of columns selected. In addition, a search box is provided to further filter the data that has just appeared.

The Interactive Decision Tree section first plots a recursive partitioning and regression decision tree using the rpart package. This tree updates when researchers modify parameters of the rpart and party tree model including cp, minimum split, and maximum depth. These three options add additional controls to the rpart and ctree tree algorithm. The "minsplit" specifies the minimum number of observations that need to exist for a node to split the tree. "Max depth" specifies the maximum depth of any node of the final tree, and the root node is counted as 0. Finally "cp" is the complexity parameter. Any split that does not increase the fit of the model by the factor of the cp is not attempted. This can be thought of as a pruning parameter. The ranges for all these controls were set after tuning the model several times. The values are colored through the rpart.plot package according to final probability of chapter on the leaf. The colors randomly change with the size of the tree. The RPart Model tab shows a plot for the variance explained by the rpart tree. The variance plot changes with the controls set by the user.

One key thing to note is that the Conditional Inference tree is nonparametric. Hence, the change in cp will not affect its fit but the other controls will. A conditional inference tree using the ctree package is fit to predict the chapter of the diagnosis based on the given symptoms. Conditional inference trees estimate a regression relationship by binary recursive partitioning in a conditional inference framework, and avoid variable selection bias. The only parameter that can be controlled by the researcher is depth. The maximum depth of this tree is six. All leaf nodes of the tree provide a bar plot showing the probability of the particular symptom occurring in each of the 19 chapters.

On Expanding the Feature Exploration tab, the first subsection is a barplot for showing the problems surrounding non distinguishable diagnosis in the pivot dataset. As mentioned in our data analysis, we collected all diagnostic codes with the same binary code for symptoms and grouped them by chapter. Thus, this barplot shows the number of non-distinguishable diagnoses per Chapter. Hovering over a bar provides the chapter number and the value for non-distinguishable diagnoses for the chapter. Diagnoses that have very similar symptom occurrences are almost non-separable to the computer. For instance, Chapter 16 has 30 diagnoses that have the same symptoms.

The second section of the Feature Exploration tab is the results of fitting a random forest. The random forest was fit on the chapters in the pivot file as modeled by the symptoms. The barplot shows the variable importance for the random forest as measured by the mean decrease in accuracy. Hovering over a bar provides the chapter number and the value for mean decrease in accuracy. This plot shows how important the symptoms are in classifying the chapters; thus, the symptoms in the plot are ordered in ascending order based on their mean decrease in accuracy.

The results from the confusion matrix of the random forest are projected to a heat map to show the importance of each symptom in classifying a chapter. The chapter numbers are along x-axis and the symptoms are labeled on the y-axis of the heat map. A legend on the right has a continuous color scale. The intensity increases from blue to yellow, while the corresponding values increase from 0 to 30. Upon hovering over a square on the heat map, the user can see the symptom, chapter, and value of importance of the symptom to the chapter. Researchers can also zoom into the heat map to view the details. For instance, in chapter 12, the symptom "sleep" is colored yellow with an importance value of 32.62. These results are intuitive, as chapter 12 is titled Sleep-Wake Disorder and we would expect the symptom "sleep" to be the most important.

The last bar plot from the Random Forest results displays the class error in predicting each chapter. In fitting the random forest, 500 trees were made and at each split five variables were tried. The total error of the model is 35.27%. The height of each bar in the plot shows the error rate for each chapter. The bars are colored by the number of occurrences of the chapter in the pivot file. According to the legend on the right, chapters with fewer occurrences in the pivot file are in darker blue and those with more are in lighter blue. Hovering over a bar provides the chapter number, the value classification error, and number of occurrences of the chapter in pivot file. From the bar plot we can interpret that only three diagnoses were in chapter three according to the pivot file, and the chapter was incorrectly predicted 100% of the time based on the binary symptoms.

V. System Evaluation

The clinician interface has some flaws. It does not deactivate the "contextmenu", or the pop-up "yes" and "no" selections that appear once the user right-clicks. The interface, at its current state of development, does not deal with this case. The assumption is that the user does not click the gray buttons, but it would be better if our interface did deal with it. By trial, it seems that this assumption is unsafe to make for users attempt to right click on the gray bubbles.

By trial, it seems that the right click button on a mac is user unfriendly depending on the settings of the mac and the user's experience with a mac. It is advised that a simple mouse is connected to the computer to better the user experience.

The interface also does not automatically update the possible diagnoses. The user must manually click the "What are the potential diagnoses so far?" button to see the list of diagnoses. This was initially designed as such in anticipation that updating the data constantly may be confusing. However, in retrospect, this does not seem to be a problem.

The diagnoses list also does not list the name of each diagnosis. Each is only identified by its code.

Another problem with the system is the way in which the bubbles collide may be too distracting for some users. Whether force collision is necessary for the rearrangement of the bubbles is questionable.

In addition, the interface does not allow the users to compare the various sizes of the bubbles in an efficient manner. It is hard to distinguish which bubbles have higher frequency for those that are far apart with similar areas.

The bubbles are not color-blind friendly. Different opacities should be associated with the colors or a different selection of colors is advised.

There are times where the labels on the edges of the decision tree both become “No”. This is a bug by design and should be fixed. A possible way to solve this is to label only the top two edges with “Yes” and “No”.

VI. Future Directions

We would first like to refine the questions asked in clinician interface and integrate the research. Secondly, we would like to reduce the number of indistinguishable diagnoses that our system produces. In the binary decision tree, there are slight differences within each category that should be able to help us distinguish two different (similar) diagnoses. Additionally, some of the symptoms in a category seem quite different: we are lumping together self-centered and low self-esteem, for instance, which seem to be quite different.

Therefore, we believe there should be an additional option to filter the categories more finely if there are multiple possible diagnoses when the user gets to the bottom of the tree. For example, the user could specify that within the attachment category, the symptom is that the patient has too much trust in strangers, rather than a feeling of abandonment. Hopefully, this should reduce the number of diagnoses with the same binary code.

We imagine this to be something like when one searches for something on Amazon and too many results pop up, one can filter through the results by specifying certain categories (e.g. seller, product category, shipping option).

We would also like to get the opinions of more domain experts, such as clinicians and potentially the Smith psychology department.

As suggested by a test user, we could compare overlap rates in chapters of the DSM-5 to chapters in the ICD-10 book for medical doctors. How muddy diagnoses are or are they more objective? The domain expert predicted the ICD-10 would be less muddy. If, however, the muddiness is comparable, perhaps the way we group symptoms is flawed.

VII. Acknowledgments

The Justice Resource Institute was instrumental and helped us with their domain expertise. D3.tip was created by Justin Palmer and adapted by Constantin Gavrilete and David Gotz. d3.contextmenu was created by Patrick Gillespie. Finally, we are very grateful for Jordan Crouser’s support and guidance.