Explaining Misclassified Instances in Machine Learning during Training to Support Informed Debugging of Data

Binod Thapa Chhetry, Farnaz Irannejad Bisafar, Irene De La Torre - Arenas
Northeastern University

Abstract

Machine learning users need to train their learning algorithms and perform informed troubleshooting while doing so. Ways to assist these users include tuning model parameters of the learning algorithm, as well as visualization of prediction errors in terms of raw data and features. Having effective visualization of data is vital for quick and thorough troubleshooting of these learning systems. We present a prototype visualization of raw accelerometer data to facilitate comparison of data between different classes of activities. We also present the findings of the domain expert evaluation study we did based on our prototype. We show that the explanation of prediction errors using raw data comparison is received as a vital feature sought during the training process of learning systems.

Keywords: Machine Learning, Data Visualization, Confusion Matrix, debug of algorithms.

1 Introduction

Typically, machine learning (ML) algorithms are trained using massive data sets. The algorithms learn which features are relevant, often with help from an ML expert that tweaks parameters. However, in user-oriented systems that must recognize highly-individualized behaviors or preferences from relatively small amounts of labeled data, the non-expert, end-user must be involved in training data collection and labeling, and in algorithm correction. 'Black-box' systems, with output that is not interpretable by a non-expert, will frustrate end users if they become dependent upon the systems, because they will need to tailor and fix the systems when these break. Users must understand the system in order to guide it with helpful feedback that leads to tangible improvement in performance. We propose an interactive visualization system that aims to solve this problem. It is a visualization that explains the raw data and extracted features of the misclassified instances by facilitating a comparison against the raw data and features of correctly classified instances. Compared to the complex learning algorithms, the raw data and features are two entities that the users are mostly familiar with and have a better understanding of. We believe that our proposed solution imparts 'actionable' explanation of the misclassified instance. This is because the explanation should enable users to make appropriate informed decisions for debugging and training the learning system, as the users gain better understanding of the validity and richness of data/features for the classification task in hand.

Guiding algorithm training and adaptation using a user's intuitive common sense has to be easy and fast for typical people who are not trained in computers or algorithms. If the user assumes the algorithm works one way, but it actually does something else, only confusion will result. The user's mental model of the learning system must appropriately map onto the behavior of the actual system to facilitate an iterative-feedback loop between the system and the user. The phrase interactive machine learning was popularized by Fails et al. while describing how a train-feedback-correct cycle allowed users to correct the mistakes of an image segmentation system [1]. Since then, researchers have explored this cycle to enable better model selection by end-users [2], and to elicit labels for the most important instances [3]. ModelTracker [4] is an interactive visualization designed to support direct error examination and debugging in ML, as it subsumes information contained within numerous summary statistics while displaying example-level performance. Recent work has also focused on building intelligible ML algorithms so that end-users can understand, validate, edit and trust a learned model [5]. Explanatory debugging [6] incorporates both machine-learned knowledge representation and user-driven learning simultaneously. Kapoor et al. proposed ManiMatrix that allows users to directly interact with confusion matrix and to view the implications of incremental changes to the matrix via a real-time interactive cycle of re-classification and visualization [7]. Confusion matrix is a specific table layout that allows visualization of supervised learning. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class. To our knowledge, most studies have focused on visually explaining the learning algorithm but not enough focus has been given to build intuitive interactive visualizations of the actual data and features and evaluate its significance in training of the system.

Our visualization is an explanation framework rather than interactive ML tool, which could be a possible future extension. The proposed solution would visualize the actual raw data and the extracted features associated with misclassified instances based on users' interaction with the confusion matrix. This should help users build sound (i.e.: correct) mental model of the process. We hypothesize that this will enable users to have more correct mental model of the system and be more efficient and accurate in debugging and training the learning system

2 Related Work

Visualization in machine learning have primarily focused on the learning algorithm and not so on the features and raw data. Recent works have focused on visually explaining additive classifiers[8], presenting a graphical view of confusion matrices to help users understand relative merits of various classifiers[9]. Kapoor et al. proposed ManiMatrix that allows users to directly interact with confusion matrix and to view the implications of incremental changes to the matrix via a real-time interactive cycle of re-classification and visualization[7]. Confusion matrix is a specific table layout that allows the visualization of supervised learning. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.

ExplainD[6] is another promising design framework for representing learned algorithm to the end-user. For various linear classifiers (naïve Bayes, Support Vector Machine, Logistic Regression), the framework provides visual explanation of decision, relative contribution of features, capability to interactively change feature value and visually audit the change in classification decision, ranks of evidence, source of evidence (assisting user in exploring the reasoning and data behind the classifier). ModelTracker[4] is an interactive visualization designed to support direct error examination and debugging in ML, as it subsumes information contained within numerous summary statistics while displaying example-level performance. Bykov and Wang[10] proposed an interactive visualization tool that provides information about each result of the algorithm and allows an exploration of each classification. The visualization (Fig 2), structured as a dashboard, is coded in D3.js.

Despite recent works in visualization supporting interactive machine learning and user-in-the-loop learning, there has been little study on the interactive visualization facilitating comparison of raw data and features of classified and misclassified instances and evaluate its significance in training of the learning system.

In a more different spectrum, but important to mention, is the generalist visualization A Visual Introduction to Machine Learning[11] by Stephanie Yee and Tony Chu, a website that has won many awards, both from scientific publications and design websites, and that has obtained a lot of attention from communities in data science, data visualization and machine learning. Although it is not a tool for debugging algorithms, we consider it a good example of how a visualization can explain this topic effectively to a non-expert audience. With a very simple style and by using different animations and transitions, the visualization guides the viewer through the different steps of statistical learning and modelling of features.

For us, the perfect outcome would be having the same degree of complexity as the visualizations mentioned at the beginning of this section, with the user-friendliness and clarity of this visualization.

To finish this segment, it is interesting to notice the importance that the Chu and Yee give to decision trees, as they consider that seeing the data flow is the best way to see when a model overfits. None of the other mentioned works include the decision flow in their visualizations.

3 Process

Our visualization was designed following specific steps to optimize our work functionality. Data clean-up was not needed, although, during the design process it had to be filtered to a small set. We also did interviews with two different potential users to identify possible opportunities or weaknesses in our visualization. Finally, after the coding part we proceed to an user evaluation.

3.1. Data

Acquisition and clean-up: Our project uses a dataset (courtesy of mHealth lab at Northeastern) of acceleration data tagged with activity type that was previously acquired for a study in activity recognition. Fifty participants took part in this study after providing written informed consent. Tri-axial accelerometers (ActiGraph) were attached using Velcro bands to each participant's dominant and nondominant wrist, dominant and nondominant ankle, dominant and nondominant waist and dominant thigh. The participants also had LG Urbane smartwatch on their non-dominant wrist and carried their phone with them (on hand or in pocket) during the protocol. The protocol included the research assistant instructing participants to perform specific physical activities (e.g.: walking on a treadmill, biking outdoors, unloading books from shelf, etc.) and annotating the activity label as well as start and end time on a tablet, while the tri-axial accelerometers (from standalone sensors as well as smartwatch and phone) were continuously collecting data. All the devices collected data with 80 Hz sampling rate, with exception of smartwatch and smartphone for which sampling rate was 20 Hz.

The raw sensor data is measured in g (m/s2) units. The acceleration data was downloaded using the ActiLife software in the form of compressed/gzipped csv format. These devices are widely used and extensively validated, so the acquired raw sensor data was clean and did not require any cleaning. The metadata/annotation files were revisited to make sure the activity labels and their start and end time were free from human errors. Hence, the collected data consisted of two files: (i) accelerometer file containing time stamp and acceleration in x, y, and z axes, and (ii) annotation file containing start and end time for each activity.

Derived attributes: In our case, each item is a five second window in time and attribute for that item is raw accelerometer data and its associated activity label(s) contained in that window. Other attributes from the window are the derived features from the raw data, which include for each axes (x, y and z): mean, variance, zero-crossing rate, mean distance from the mean, area under the curve, absolute max, dominant frequency, dominant frequency energy and spectral entropy. Correlation between of acceleration between two sensors and sensor orientation based on roll and pitch are also the derived features or the attributes.

3.2. Interviews

In this section, we will give a brief overview of two of the interviewees we initially chose for our task analysis. At this stage, we wanted to understand the needs of not only experts in ML, but also of people who have basic knowledge about ML, but might still encounter the use of it. We therefore chose one expert in ML and one non-expert in ML.

3.2.1. Insights from Non-experts in ML

Our interview with Interviewee-1 had the goal of identifying how non machine-learning experts but users of algorithms and ML techniques could, first, understand our visualization and, secondly and more important, be interested in a possible debugging tool. We approached Interviewee-1 because he is a user of products that use sensors to capture movements or behaviors (such as fit bands) and is also a researcher whose job is to generate data that later will be used by other people to create ML algorithms.

Asked about his knowledge in the topic, Interviewee-1 stated to be familiar with ML and understanding its fundamentals. His interest in the subject is influenced by his current job, and he has been doing machine learning courses to know how to prepare better datasets for future algorithms. He is aware that many times the raw data obtained from 'real life' does not match the algorithm making it fail.

When asked about visualizations about ML or visualization tools for it, Interviewee-1 indicated that he had not seen any about the ML algorithms. However, for him, being able to see how the decisions are made would be more useful than actually knowing more about the algorithm. This approach is represented in the sketch that we asked him to do about a possible ML visualization, where for him the important element was the decision flow in order to see where the error in the algorithm happened.

Interviewee-1's feedback about our initial sketches was really useful: he remarked that the visualization needed to have a very good hierarchy and very good labels to avoid confusing the user, especially in the items names; and highlighted that the visualization needed to allow comparisons of peaks between the different charts at the same time. Overall, he stated that our idea would save a lot of time and would allow two types of use:

- tagging of instances, which would be made by interpreting the charts and could be done by people who are not so proficient in ML
- and fixing the algorithm by experts through the tagging done previously

Our idea of interviewing a more general user, who is not an expert machine learning user, was to understand how we could make our visualization more intuitive and clear to a novice user, nor necessarily to impact our motivating questions related to training/debugging of the algorithm.

3.2.2. Insights from Experts in ML

Our expert interview was done to Interviewee-2, a researcher at Northeastern that uses machine learning algorithms in her work. We approached her with the goal of understanding how ML users generate / re-use / correct or change algorithms in order to get more accurate results. In her opinion, ML needs to be a bigger part of the curriculum for those interested in doing research since, at some point, everyone is going to need using it. Things that are not currently included and are a very important part in her work is data cleaning, filtering and noise cancellation.

Her expertise in ML makes her aware of the important task that is generating and gathering the data. In fact, she remarked how important taking notes while collecting data is. For Interviewee-2, this step is vital in identifying why an algorithm does not work in a specific moment since, in her opinion, the problem usually happens in the gathering of that data, when something occurs different than it should be.

Asked about how she chooses or implements ML, she explained to us that the election of one algorithm instead of another depends mainly on the type of results that the researcher wants: best precision or best accuracy algorithms usually require different nests of features, while algorithms in mobile phones or similar devices need to be fast and, therefore, simple.

Interviewee-2 declared to not be a usual user of sensors or products that collect data in her daily life. She checks the activities categorization that some smartphones' apps do from time to time. However, her main concern about these products is that they cannot be used for specific tasks because they are not accurate, although she agreed that they are good for having overview insights.

Asked if she uses visualization tools in her work, she remarked that sometimes visualizing data is not an easy task, especially when managing big data. Interviewee-2 stated that in fact most of the times there are not tools to visualize those results. Right now, she studies which features show more correlation with the results by using R and visualizations but she agrees that a tool for the raw data and the feature would be a good instrument to have in order to view what is happening in the algorithm and improving her knowledge of this process.

Her feedback about our initial sketches was really useful to identify strengths, weaknesses and opportunities in our idea. Interviewee-2 thought that our visual exploration was a little simplistic and that we give too much importance in the features, while the process of selecting them tends to be done at the very beginning of the process and not at the end, when our visualization is supposed to be used. She did not see the functionality in the frequency feature histograms for this very same reason.

However, she stated that a visualization that highlighted outliers and would help her summarizing and filtering data would be a great implementation.

Aida was not sure that the visualization that we proposed on comparing the features would help her, since she mentioned she would do scatter plots of features ahead of running machine learning algorithm on them. However, she encouraged if there is a way to display these multidimensional features in isolation or combination that could potentially shed more insight of their richness and relevance.

The interview was very helpful and gave us ideas where we can improve our design. However, we feel our motivating questions were validated by this interview.

2.3 Task Analysis

Domain	Analytic tasks *	Technical tasks
Summarize the performance of machine learning algorithm		Visualize confusion/adjacency matrix with true vs. predicted labels for multiple classes
Search and identify misclassified and correctly classified instances	Filter	Use confusion matrix to select cells that belong to correctly or incorrectly classified instances
Compare the raw data and features of the misclassified instances with the correctly classified instances to discover any errors they could fix to improve performance	Correlate, find anomalies	First, visualize raw time-series data of correctly classified and misclassified instances juxtaposed together to check for noisy labels. Second, visualize features used of correctly classified and misclassified instances to check if there is bug in feature computation or if the features are not rich enough to capture the pattern in the data

* Based on Amar et al [8]

1 Design

The key point of our design is to give answer to a current necessity in ML: being able to see what the ML algorithm is actually predicting and how. Therefore, our visualization was created with a very functional aspect. Although aesthetics were important as a way to improve the user experience, and facilitate the interaction; the main functions of our design are to summarize the performance of the algorithm, to search and identify instances and to facilitate comparisons between the data that was correctly and incorrectly classified.

4.1. Design Process

Our design process was driven by a very specific goal: facilitate the debugging of machine learning algorithms. For us, it was very important to design and code a visualization that was intuitive and, especially, easy to use and interact with. From the very beginning, we knew what the different parts of our visualization would be: a confusion matrix, an item list, three line charts and a frequency exploration. However, it was through interviewing our potential users that these ideas got refined, and more focused on the UX experience.

With the answers' of our interviewees, we started sketching the four fragments. The differences between those drafts could be seen not in the high-level goals of those visualizations -the tasks that each of them should implement-, but in the extra functionalities: tooltips, brushings, buttons, etc. The final sketches were chosen based on them.

Most of the difficulty of this project does not lie in the visual aspect, but in how the users interact with the different elements and, especially, how those parts are interconnected. In the back-end, we expected filtering to be the central point of the code, as it finally was. Furthermore, it influenced the size of our dataset since we had major difficulties with it. At the beginning, our goal was to plot data from five different participants but the speed of the filtering –zero, since the browser broke every time we tried to do it- made us reformulate it. At the end, we plotted data from two participants. The loading speed was still very slow, but the visualization worked.

Evaluations with users highlighted problems in the website flow. Their feedback allowed us to identify our visualization's weaknesses: a confusing flow, need of more and better legends and titles, and nonsufficient identifications of what each section was. We compiled the feedback received from our user testing into the following categories:

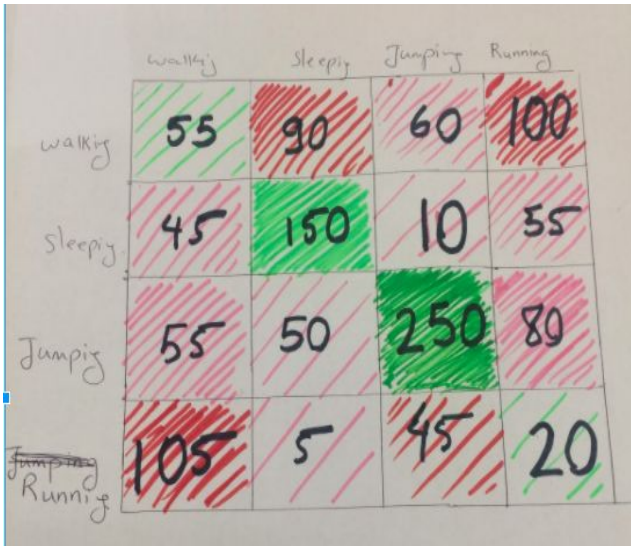
4.1.1. Positive feedback

- Choice of confusion matrix: P1 thought that the confusion matrix made sense when there are more than two classes of activities. Therefore she thought the visualization was a good choice. P2 also expressed that the matrix is exactly what he needs for his work. Both users were quick to grasp what the matrix was intended for.
- Color scheme: P1 mentioned that the color of red and its intensity based on the number of instances was a good choice.
- Visualization of item List: P1 and P2 found the layout and listing of the items intuitive. P2 thought that visualizing the items as a list is very useful.
- Line charts: P1 thought that the line charts will be useful when there are major issues such as missing data.

4.1.2. Needed Improvements

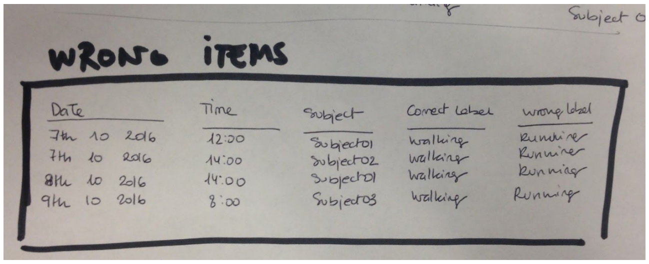
- Titles: P1 mentioned that the titles are not intuitive. She also pointed out that we need to put title on the axis (predicted class and actual class). In the next iteration we will improve our descriptions and titles based on her feedback.
- Explanation: She was confused about the explanation at the beginning, but this was mainly because it is still not completed. Therefore we asked her what explanation could be useful for her to see at the beginning. She mentioned that it would be good to give an overview of how the visualization works.
- Listing of samples: P1 thought that using left and right sliding buttons for left and right will be difficult to work with, in case there is large number of samples. For now we are showing the first 10 samples. But after discussing with P1, we realized that it's important to know what samples should be chosen. But at the same time having a lot of samples would be hard to handle. We will discuss about improvement to this in our discussion section for future work.

4.2. Sketches



	walking	Sleeping	Jumping	Running
walking	55	90	60	100
Sleeping	45	150	10	55
Jumping	55	50	250	80
~~Jumping~~ Running	105	5	45	20

Figure 1: Final sketch for the confusion matrix



WRONG ITEMS

Date	Time	Subject	Correct Label	wrong label
7th 10 2016	12:00	Subject01	walking	Running
7th 10 2016	14:00	Subject02	walking	Running
8th 10 2016	14:00	Subject01	walking	Running
9th 10 2016	8:00	Subject03	walking	Running

Figure 2: Final sketch for the item list

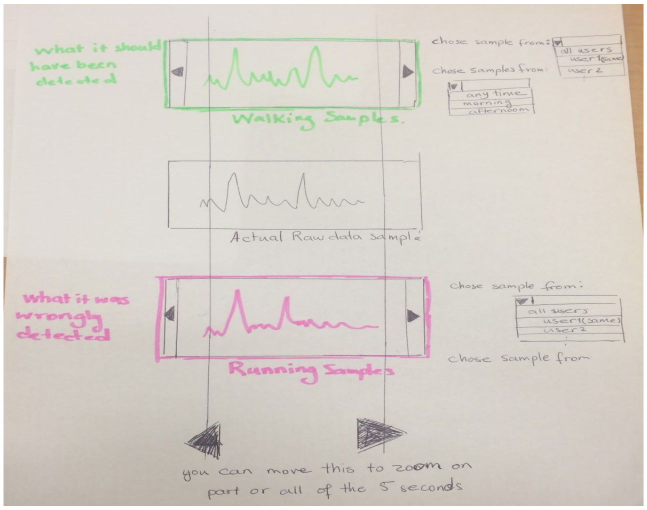


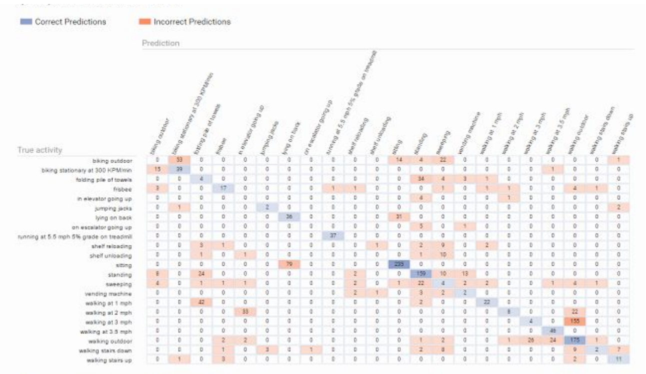
Figure 3: Final sketch for the raw data comparison

4.3. Final design decisions

The final version of our design consists of two main sections. In the first section, we provide a high-level explanation of machine learning algorithms. This explanation serves as an introduction and brief background for interpreting the visualizations and data. The second part of the design consists of the main features of the application developed through an adjacency matrix, an item list and line charts. We first discuss our design choices from a user interaction perspective and then discuss about how our design decisions follow visualization principles.

Through our interviews, we realized that one the things users want is the ability to make comparisons at different levels. This is very important for sense-making and discovery. Therefore, while our visualization is designed as a guided step-by-step process (where each of the visualizations only appear once the users interacts with the visualization), users can also see everything as a single main view. This decision was mainly made to better assist users in comparing different levels information. For instance, if they discover something through the lines charts about a specific activity and need to compare it with another activity in the matrix, they can easily see the matrix as it is in the same page. We also wanted to complement our visualization with a tutorial tour. However, since some of the features of the page were first non-existent and only appeared after the user demanded them, we tried to make provide as much information as possible at each step, to better assist users through the process.

We now discuss the three main visualizations. Our first visualization is a confusion matrix. We chose the adjacency matrix (Figure 1) to show the prediction results of the machine learning algorithm. As previously mentioned, rows represent the classes of actual activities and the columns represent the classes of activities that have been predicted by the ML algorithm. The value in each cell is equal to the number of all samples with a specific actual class and predicted class of activity. Therefore, the number inside the cells that are on the diagonal - for which the actual class is equal to the predicted class - represent the number of samples which have been correctly predicted for a specific class. Figure 4, illustrates the visualization of the confusion matrix.



🟦 Correct Predictions 🟧 Incorrect Predictions

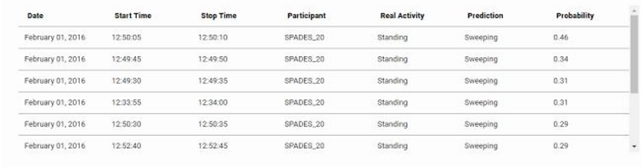
Prediction

True activity	biking outdoor	biking stationary at 300 KPM/min	folding pile of towels	frisbee	in elevator going up	jumping jacks	lying on back	on escalator going up	running at 5.5 mph 5% grade on treadmill	shelf reloading	shelf unloading	sitting	standing	sweeping	vending machine	walking at 1 mph	walking at 2 mph	walking at 3 mph	walking at 3.5 mph	walking outdoor	walking stairs down	walking stairs up
biking outdoor	0	33	0	0	0	0	0	0	0	0	0	14	4	22	0	0	0	0	0	0	0	1
biking stationary at 300 KPM/min	15	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
folding pile of towels	0	0	4	0	0	0	0	0	0	0	0	0	34	4	3	1	0	0	0	0	0	0
frisbee	3	0	0	17	0	0	0	0	1	1	0	0	0	1	0	1	1	0	0	4	1	0
in elevator going up	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	1	0	0	0	0	0
jumping jacks	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
lying on back	0	0	0	0	0	0	36	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0
on escalator going up	0	0	0	0	0	0	0	0	0	0	0	0	5	0	1	0	0	0	0	0	0	0
running at 5.5 mph 5% grade on treadmill	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0
shelf reloading	0	0	3	1	0	0	0	0	0	0	1	0	2	9	0	2	0	0	0	0	0	0
shelf unloading	0	0	1	0	1	0	0	0	0	0	0	0	1	10	0	0	0	0	0	0	0	0
sitting	0	0	0	0	0	0	79	0	0	0	0	235	0	0	0	0	0	0	0	0	0	0
standing	8	0	24	0	0	0	0	0	0	2	0	0	159	10	13	0	0	0	0	0	0	0
sweeping	4	0	1	1	1	0	0	0	0	2	0	1	22	4	2	2	0	0	1	4	1	0
vending machine	0	0	0	0	0	0	0	0	0	2	1	0	3	2	2	0	0	0	0	0	0	0
walking at 1 mph	0	0	42	0	0	0	0	0	0	0	0	0	2	0	0	22	0	0	0	0	0	0
walking at 2 mph	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0	0	8	0	0	22	0	0
walking at 3 mph	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	155	0	0
walking at 3.5 mph	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	0
walking outdoor	0	0	0	2	2	0	0	0	0	0	0	0	1	2	0	0	1	26	24	175	1	0
walking stairs down	0	0	0	1	0	3	0	1	0	0	0	0	2	8	0	0	0	0	0	9	2	7
walking stairs up	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	11

Figure 4: Final confusion matrix

Despite the use of matrices might slightly be burdensome for users, we chose a matrix, for two main reasons. First because confusion matrices are concepts that most users in the area of machine learning are already familiar with and second because they are easily scalable for large number of classes activities. For instances, alternative designs such as tree structure would have been hard to manage given we have more than 20 classes. The users can click on any cell of their interest on the adjacency matrix to get the details on demand. For representing numbers in each cell we initially thought of creating other versions of the visualization. For example, one of our ideas was the use of circles with different sizes at the intersection of each column and row (bigger sizes of circles for higher cell numbers). However we realized that size cannot be easily scalable for large numbers in the cells. Therefore we chose to use color intensity instead of size to represent numbers. We thought that colors are a good choice for helping users to detect anomalies. To better assist users in detecting anomalies, we decided to chose the diverging color map (consisting of two colors of blue for correctly classified samples and red for incorrectly classified samples). Our choice of diverging color map is guided by the fact that we have quantitative data and are interested in extreme values. We used the aforementioned colors for two main reasons. According to Madden et al. [12], the color blue is usually associated with the attribute of "high quality". So we chose this color for the samples that are correctly classified. On the other hand, as red is a stimulating color we used it to grab users' attention to cells that have been incorrectly classified samples. Secondly, we wanted two colors that support color-blind design. Aside from using colors, we also included the actual number of instances within each cell so that users can quickly search and identify any misclassification conditions they want to work on. Colors are therefore an intentional redundancy in choice in our design. Here is a summary of the marks and channel in the matrix and attributes:

- **Marks:** area
- **Channels:** color (color saturation to show magnitude) and position (both horizontal and vertical)
- **Attributes**: class of activities (categorical data), number of samples (quantitative)



Date	Start Time	Stop Time	Participant	Real Activity	Prediction	Probability
February 01, 2016	12:50:05	12:50:10	SPADES_20	Standing	Sweeping	0.46
February 01, 2016	12:49:45	12:49:50	SPADES_20	Standing	Sweeping	0.34
February 01, 2016	12:49:30	12:49:35	SPADES_20	Standing	Sweeping	0.31
February 01, 2016	12:33:55	12:34:00	SPADES_20	Standing	Sweeping	0.31
February 01, 2016	12:50:30	12:50:35	SPADES_20	Standing	Sweeping	0.29
February 01, 2016	12:52:40	12:52:45	SPADES_20	Standing	Sweeping	0.29

Figure 5: Final item list

Based on the cell that the user chooses from the adjacency matrix the users pick, they are provided with the list of samples represented by the cell. Those samples could either be a list of misclassified or correctly classified instances for which a list of attributes such as participant ID, start and end time, true activity and predicted activity are shown. This list is sorted based prediction confidence (the algorithm decision is probabilistic, so it gives probability score for each decision it makes) of the algorithm. For the list of items, we chose a simple table to list all of those instances. Users can scroll and see them. We think that while tables are very simple, they can depict the list of instances and their properties very well. The interactions that is available is the selection of each row. Figure 5 illustrates our visualization for the list of items.

After selecting a specific row, users will see three line charts which are the visualization of raw data. We chose line charts to plot raw accelerometer data, as they are apt to show trends as well as patterns in the time-series data. The line chart of the misclassified instance is in the middle while the line chart for instance of the actual class and predicted class are on the top and bottom, respectively. We believe this will facilitate easier and quick comparison of raw data. We have also added a left and right button for the true/predicted instance, so that the users can compare the misclassified instance with more than one correctly identified instance from both the classes. We have also used three different colors for each accelerometer direction (x, y and z). We have used the categorical color map, with three distinct hues. Here is a summary of channels and marks in the line charts:

- **Marks**: lines, color
- **Channels**: tilt, length
- **Attributes**: acceleration data (quantitative).

Figure 6, illustrates our visualization of the line charts.

Throughout our whole design, we have followed many visualization principles:

- We have used clear labeling of the data.
- While we haven't measured the data ink ratio parameter for our visualizations. However, we have tried to perceptually increase this number as much as possible. For example, even the use of dotted lines on the line chart were carefully chosen and debated. We realized that those lines could actually help users to better follow the lines. Therefore we decided to use them.
- We only have justified redundancy (colors)

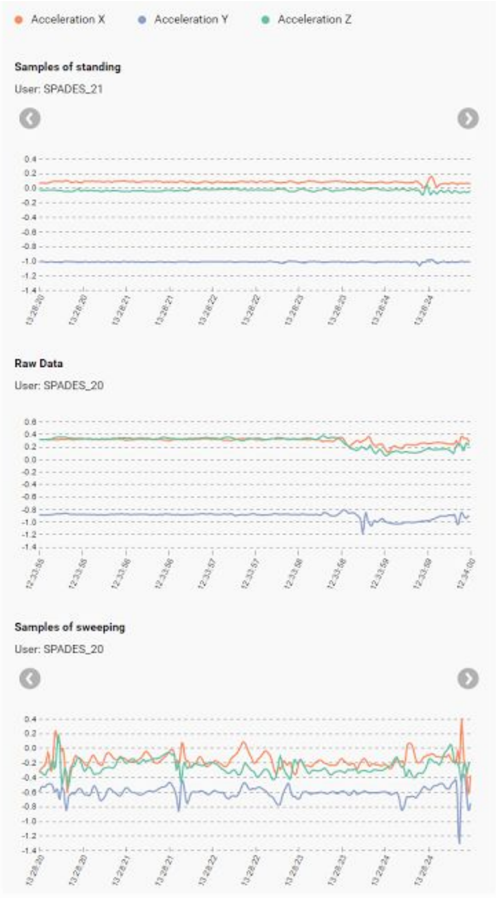


Figure 6: Final raw data comparisons

2 Discussion

The findings of our user study demonstrate some design aspects of our visualization prototype was more intuitive than others. Some of which could be attributed to the fact that few key explanations had not been fully implemented at the time of the user study. Despite a relatively short time of exposure with our visualization, both the users commented that the use of confusion matrix to show the performance of multiclass machine learning algorithms was apt given its familiarity in this domain. The users were able to understand the significance of color choices to represent correctly labelled cells and incorrectly labelled cells. The users also thought that the layout and the listing of information about the instances was intuitive and helpful, especially while troubleshooting prediction errors in machine learning.

These strong findings based on our user study is evidence of the importance of explaining prediction errors while training machine learning algorithms. Given the complexity of implementing the visualization as well as users' availability, we were able to finish implementing the line charts for raw data comparison after our user-study. Hence, we were not able to test our hypotheses of whether going through the troubleshooting process using our visualization could actually help users with building sound mental model of the algorithm, and that they could train machine learning algorithms with better performance faster than users that did not use our visualisation.

Based on user testing, we realized that one of the issues with our design was that users might have issues working with large number of samples. At the current state of our visualization, the user has to manually peruse to individual instances during the decision process, which has potential to increase mental burden and decision fatigue. We think that the design could be improved by either grouping samples or filtering them.

Based on the web performance, and coding complexity, we have realized an important limitation on our project, which is the size of our dataset. Due to the size of our CSVs because of the high sampling rate(80 Hz), we had to limit our prototype's data to only two participants as it was the only way, not only to charge the website, but also to avoid the browser crashing. For further and bigger uses of data, we would need a more robust language and machine.

3 Conclusion & Future Work

We have developed a visualization for raw data exploration for physical activity recognition through our qualitative user study. The results of our user evaluation imply that combination of confusion matrix, instance list and line charts for raw data comparison can be effective way of troubleshooting machine learning errors while training. Since, it is not yet integrated to a machine learning system, we realize that our visualization could add potential burden to the users as they have to work with the actual machine learning system and our visualization.

We are continuing to develop our visualization system for helping users train machine learning algorithms. We will investigate the most effective way to visualize features to compare features between the correctly classified instances and the incorrectly classified instance, so users can make judgement on the relevance of features being used. Finally, we plan to scale our visualization such that it can take massive dataset and be helpful to users of big data.

The work presented here can also be applied to other domains that use machine learning for training purposes. This visualization also has the potential to be integrated to a user-in-the-loop interactive machine learning system, where users are both learning about the machine learning system and also fixing it using their new found knowledge about the machine learning system.

Acknowledgment

We would like to thank mHealth lab at Northeastern University for kindly providing us with the data we used. We would also like to acknowledge our users and domain experts that used our system and provided us with helpful feedback to inform our design and documentation.

References

[1] Fails, J. A., Olsen, D. R., and Jr. Interactive machine learning. In Proceedings of the 8th International Conference on Intelligent User Interfaces (2003), 39–45

[2] Amershi, S., Fogarty, J., Kapoor, A., and Tan, D. Examining multiple potential models in end-user interactive concept learning. In Proceedings of the ACM Conference on Human Factors in Computing Systems (2010),1357–1360

[3] Cakmak, M., Chao, C., and Thomaz, A. L. Designing interactions for robot active learners. IEEE Transactions on Autonomous Mental Development 2, 2 (2010), 108–118

[4] Amershi, S.,Chickering, M.,Drucker, S.,Lee, B.,Simard, P.,Suh, J. ModelTracker: Redesigning performance analysis tools for machine learning. ACM HFCS 2015

[5] Caruana, R.,Lou, Y.,Gehrke, J., Koch, P., Sturm, S.,Elhadad, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. KDD 2015

[6] Kulesza, T.,Stumpf, S.,Burnett, M.,Weng-Keen, Wong,Riche, Y.,Moore, T.,Oberst, I., Shinsel, A., McIntosh, K.Explanatory debugging: Supporting end-user debugging of machine learned programs VL/HCC 2010

[7] Kapoor, A.,Lee, B.,Tan, D., Eric Horvitz, E. Interactive optimization for steering machine classification,Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 10-15, 2010, Atlanta, Georgia, USA

[8] [8] Poulin, B., Eisner, R., Szafron, D., Lu, P., Greiner, R., Wishart, D.S., Fyshe, A., Pearcy, B., MacDonell, C., Anvik, J. Visual explanation of evidence in additive classifiers, Proceedings of the 18th conference on Innovative applications of artificial intelligence, p.1822-1829, July 16-20, 2006, Boston, Massachusetts

[9] [9] Talbot, J., Lee, B., Kapoor, A., Tan, D.S. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 04-09, 2009, Boston, MA, USA

[10] Bykov, A., Wang, S. Interaction visualizations for supervised learning. University of Washington, 2014. http://cse512-14w.github.io/fp-abykov-snwang/final/paper-abykov-snwang.pdf (Accessed on November 7th, 2016)

[11] http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

[12] Thomas J. Madden, Kelly Hewett, Martin S. Roth (2000) Managing Images in Different Cultures: A Cross-National Study of Color Meanings and Preferences. Journal of International Marketing: Winter 2000, Vol. 8, No. 4, pp. 90-107.