EMS Motion Data Visual Analytics

Deirdre Scully
Vanderbilt University
2301 Vanderbilt Place
Nashville, TN 37235

Abstract

This project analyzes motion data collected during emergency medical services (EMS) in order to predict which medical procedure occurred during that time. The motion data collected consists of many different types and attributes along with several classes of procedures. The goal of this project is Mixed-Initiative Visual Exploration, to learn information from the collected data through visual and machine learning exploratory analysis.

Introduction

This project analyzes motion data collected during emergency medical services (EMS) in order to predict which medical procedure occurred during that time. The GitHub repository for this project can be found here: `https://github.com/dmscul11/EMS-Prediction`

Description

The motivation behind EMS motion data collection is to solve the problem of information loss during patient handoff after an emergency transport, as well as send such information in advanced. The receiving medical facility usually is only given a one paraphrase description of what happened to the patient, such as ”a gunshot wound to the leg”. It would improve patient care if in addition the receiving medical facility was given a comprehensive list of all procedures performed on the patient before arrival, such as ”insert an IV, then CPR for 5 minutes followed by intubation”. This project aims to explore if the EMS motion data is a good classifier of the procedures occurring in real time, as well as which data types and data attributes are necessary and sufficient to do so. In order to answer these questions I will explore the data in depth to find trends, similarities, differences, and prediction powers. Visual analytics will be the key component in this data exploration, by implementing useful and interactive data summary and machine learning visualizations of the data and models. The idea is to gain insight into how well every and which data types predict the various procedures.

Background

The main importance of this project is that it has the potential to help save peoples' lives in emergency situations. However, the importance of exploring with visual analytics in this project is to determine the best visual encodings for the different combination of data types and machine learning algorithms of this time series motion data. The main difference in this project between other time series data studies is combining and comparing many different types of data into one analysis. Normally in time series data one data item will have many attributes recorded over a period of time. In this data however, one sample is comprised of a few different types of data items that are recorded at different rates per second over a period of time. This means that there is not a direct 1-to-1 correlation within one sample across the types of data items over time. In addition each type of data item has its own distinct set of attributes, so that direct comparison of the different data types becomes difficult and meaningless. The challenge is then not only how to explore a large amount of data over a series of time points, but also how to compare and analyze across types of data items that are inherently different representations, formats, and with unequal time points. Specifically the Open-Pose spacial output is an interesting new type of data items and application of visual analytics.

Data

All data collected for this project is time series data. Each section of time is labelled with a categorical class. One data type is the video recordings that were collected for most of the data items on somewhere between 0 to 4 camera angles. The different cameras used record at various Hz per second, starting at 24. The video data attributes are the time series image matrices. In addition, the videos are analyzed with the open source Open-Pose limb detection software, which outputs 18 body points by their x and y locations and confidence intervals for each person identified.

Another data type is recorded by apple watches. The apple watches record various data points at 60 Hz per second. The attributes of this data are either the left or right wrist watch, X, Y, and Z acceleration, raw, pitch, and roll.

The last data type is recorded by Myo armbands. These armbands record acceleration data at 60 Hz per second and can detect small muscle movements of the arms (EMGs) which are recorded at 200 Hz per second separately. The attributes of this data are either the left or right arm band, X, Y, and Z acceleration, and 8 different EMGs.

Most of the data is already stored in a PostgreSQL database and csv files, except for the videos which are on AWS in avi and json format. There will be some effort needed to quality control the data to make sure everything is labelled correctly as apple watch or Myo, left or right, time stamps include the correct procedure class, separate instances of different patients and procedures are distinguished, the time stamps of each recording device and procedures are following the same wall time, etc. In addition, it will likely take some time to pull all of the separate data types into similar formats, but should allow for reading in and combing the data to be more streamlined during analysis.

Baseline

One interesting visualization approach to time series data uses Temporal Multidimensional Scaling (TMDS), (Jackle et al, 2016). The main usage for this method is to examine time series data that does include many dimensions, or data attributes. It performs dimensionality reduction (DR) on the multidimensions, but still maintains the time series dimension so that the user can explore patterns over time of the large data set. "The x-axis represents the time, and the y-axis represents the MDS similarity value." (Jackle et al, 2016). The time axis will also be continuous instead of discrete, so unequal timestamps will not be an issue. An example of this is shown in Figure 1.

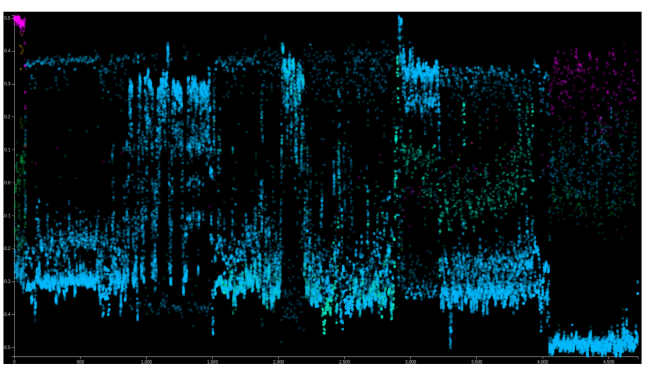
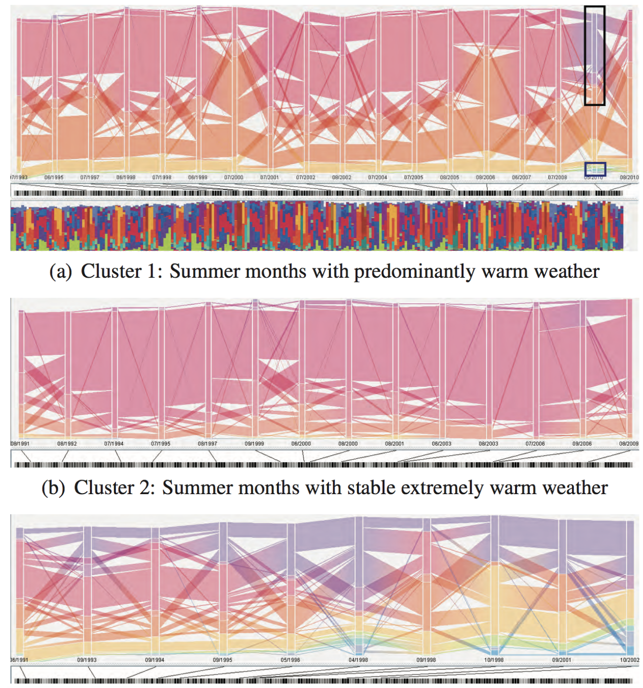


Figure 1: TMDS (Jackle et al, 2016)

Another approach I am considering works similarly with categorical time series data. One algorithmic method von Landesberger et al (2012) implement is clustering the time chunks together based on weather data and attributes. The clusters are then visually encoded as shown in Figure 2. This could apply to my data in an interesting reverse engineering way by clustering the different types of data items into time chunks, and the user could then explore if these clusters of time correspond at all to the already time labelled procedures. To do so, some visual encoding of the procedure classes must be included.

In order to implement either of the ideas mentioned, some



(a) Cluster 1: Summer months with predominantly warm weather

(b) Cluster 2: Summer months with stable extremely warm weather

(c) Cluster 3: Spring and autumn with changing weather conditions

Figure 2: Clustering (von Landesberger et al, 2012)

changes must be made to work with my data set. The main addition I plan to implement to help visualize my data specifically is a couple filtering methods. The first filter will allow the user to select one or more of the different types of input data items, and follow those specifically throughout the dimensionality reduction and machine learning processing. This selection will still keep the context of the other types of input data items, so the user can easily compare which type of data item inputs are creating what kinds of changes to the models. Another filter, which is really optional labeling, is the addition of a class distinguishing label in order to follow the data of each procedure class throughout the dimensionality reduction and machine learning processing. This will help the user compare differences and similarities in the time series trends across procedures. These filters when used together can provide many insights into which combination of types of data items are most similar to each other, and predict which procedures that are most dissimilar from the other procedures.

The language used for this project will either be Python or R, and I am already well practiced in both. If Python is implemented D3, Bokeh, or NVD3 (through Django web-app) will likely be the visualization package used. D3 and NVD3 have a larger learning curve, although I am already familiar with NVD3 through Django. The machine learning package used with Python will likely be TensorFlow or scikit-learn, or both for the neural networks and clustering algorithms respectfully. If instead R is implemented for the project the visualizations will be created in a Shiny app. The machine learning algorithms will be used with packages such as nnet for neural networks, cluster for clustering, or other built-in algorithms.

Schedule

Baseline

February 4th Research and decide which language to use, Python or R, and which image visualization package. Also determine if including video data will be too processing heavy and just use the open pose output or only apple watch and Myo data instead. Begin pulling all of the data from storage locations into similar formats for reading in.

February 11th Choose final language and visualization packages, and learn the packages that I have not implemented before. Finish collecting and formatting the different data types. Have the data read in, some pre-processing, and begin practicing using visualization and machine learning packages on the different data types. Continue research into visualization methods for time series data. Create a short presentation for updates since proposal presentation.

February 18th Use visualization package or a more simple one to create basic visualizations of the data to explore aspects and trends in the data, before implementing machine learning techniques. Continue research into visualization and machine learning algorithmic methods for time series data.

February 25th Choose and iron out details of the final baseline approach from researched methods and experimentation. Begin to outline the code implementation and ideal web-app layout of this baseline. Describe this baseline approach in a document on GitHub. Create a presentation of the baseline and related works to demonstrate the approach.

Project Updates

March 6th Begin implementing the baseline approach by looking at the researched similar methods. Sketch out templates of the ideal web-app layout.

March 13th Have at least some visualizations of the data in working condition, even if not yet interactive. Begin thinking about the machine learning approach and respective visualizations. Formulate how to make these pieces interactive and connected.

March 20th Implement some machine learning algorithms and create basic outputs from those results. Make sure the data exploration visualizations are implemented with good depth. Create a short presentation of updates since the baseline.

March 27th Experiment with the machine learning approaches and visualizations. Choose final or multiple final machine learning algorithms to include in the visualization web-app. Finalize the respective visualizations.

April 3rd Have mostly working interactive visualizations of the machine learning algorithms in addition to the finalized input data visualizations. Create a short presentation of updates since the last update.

Project Submission

April 10th Do final experimentation with the data visualizations, machine learning parameters and respective interactive visualizations in order to finalize web-app. Make sure the web-app is in good working order with all ideas implemented, only small changes remaining. Compare back to baseline chosen to evaluate the interactive visualizations so far.

April 17th Finalize code and visualization web-app. Brainstorm any last features, edits, or next steps. Compare again to the baseline chosen to make sure the visualizations reach the intended goal. Create presentation for class. Begin writing paper based on observations described in the presentation.

April 26th Implement any last features or edits into the code. Make sure the code is in presentable condition. Finish writing the paper, including all sections.

References

T. von Landesberger, S. Bremm, N. Andrienko, G. Andrienko and M. Tekuov. 2012. Visual analytics methods for categoric spatio-temporal data. *IEEE Conference on Visual Analytics Science and Technology (VAST), Seattle, WA, 2012*. pp. 183-192.

D. Jackle, F. Fischer, T. Schreck and D. A. Keim. 2016. Temporal MDS Plots for Analysis of Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics.* vol. 22, no. 1. pp. 141-150.