System Design

Modern data visualization often has to deal with massive amounts of data, which may easily lead to scalability issues of different nature. The task of interacting with sensor-generated time-series data and with its associated activity predictions, for instance, can be limited by factors such as data retrieval (e.g. fetching from servers), computational time, memory constraints and visual cluttering.
For this reason, the design of our system is based on the idea of combining database, machine learning and visualization methods in order to offer a better user experience.
We first define a standardized way to write classification models, so that they can be automatically ran on subsets of the sensor data (stored in a centralized database). These classification results are saved on the same server and then compressed in a single file, which can be fed as input to the Xplorer user interface.

Analytics Pipeline

We define a classifier as function that takes a list of data samples as an input and outputs a single probability or scoring value, along with some optional key-value pairs (``attributes''). We call the input data ``window'' and its length is classifier-specific. The data contained in the window depends as well on the classifier, that specifies which sensors (e.g. IMU) and measures (e.g. linear acceleration, rotational acceleration, magnetometer) it requires.
For instance, a walking classifier based on a wrist-worn sensor may consider sliding windows of two seconds and output for each of them a normalized probability for walking, plus some attributes such as stride-length and speed.

We define ``session'' a time range corresponding to a meaningful subset of the data collected from all sensors. We assume sensor data is timestamped and has already been synchronized among different devices. A possible session could consist in a period of two hours in which a user wears the sensors and can correspond to a variable number of data windows.

Our standardized definition of classifier enables us to make them modular and automate their execution. Given a session and a set of classifiers, sensor data is fetched from the remote server based on classifier requirements and session time interval. Depending on the classifier specifications, a sliding window is used to iterate over the sensor data and fed selectively as input to the classifiers, whose output is stored back in the centralized server.

Classifiers automation guarantees a quick output generation in presence of new sensor data, while storing classification results in a database avoids their recomputation.

Data Storage

Each classifier is identified in the database by the name of its produced label (e.g. ``Walking''), by its author and by its version. Depending on the granularity of the predictions, the classification of time-series data can produce a very large amount of information. Instead of accessing these classification results directly from the centralized server, we define a simple compression method to generate a “classification summary” for each session. For each different label produced by a classifier, we generate a “track”, meaning a list of predictions each characterized by an output value, by a start and end timestamp and by optional attributes. If multiple predictions are very close to each other and their output values do not differ more than a certain threshold, they can be unified in a single prediction, which will have a longer duration and averaged attributes. Optionally, predictions with a low output value can be discarded. In relation to the use case presented later in paper, we noted that compressing classification results yielded an average of 50% less memory without any significant change in their interpretation.

On top of sensor data and classifier results, the database can store annotations and video files associated to each session. We define annotations as labels having a time range, often inserted manually by a human.
While tracks generated from a classifier have a probability or score value, tracks generated from ground-truth labels are only characterized by start and end times, plus optional attributes.

Classifier tracks, label tracks and additional information about the session are stored inside a JSON-based .bsx file, which can be distributed even to data scientists who may not have access to the centralized database. This way, BSX files remain unaffected by later changes to the database.

If a timestamped video is available for a specific session, only one video frame per label is included in the BSX file in order to minimize its size. A reference to the full video is nonetheless stored in the file, making the video available for streaming.

User Interface

We deploy the user interface of Xplorer as a web application, developed by leveraging the *React* Javascript library. Our choice is mostly based on the need of having an easily deployable, multi-platform application. While the interface is publicly accessible on the internet, data privacy and confidentiality are preserved through the use of BSX files, which are distributed through a private channel and are only processed locally by the application.