



Activity Classification Using Android Phone Sensors

ALON AMSTER & YARDEN ZUCKERMAN

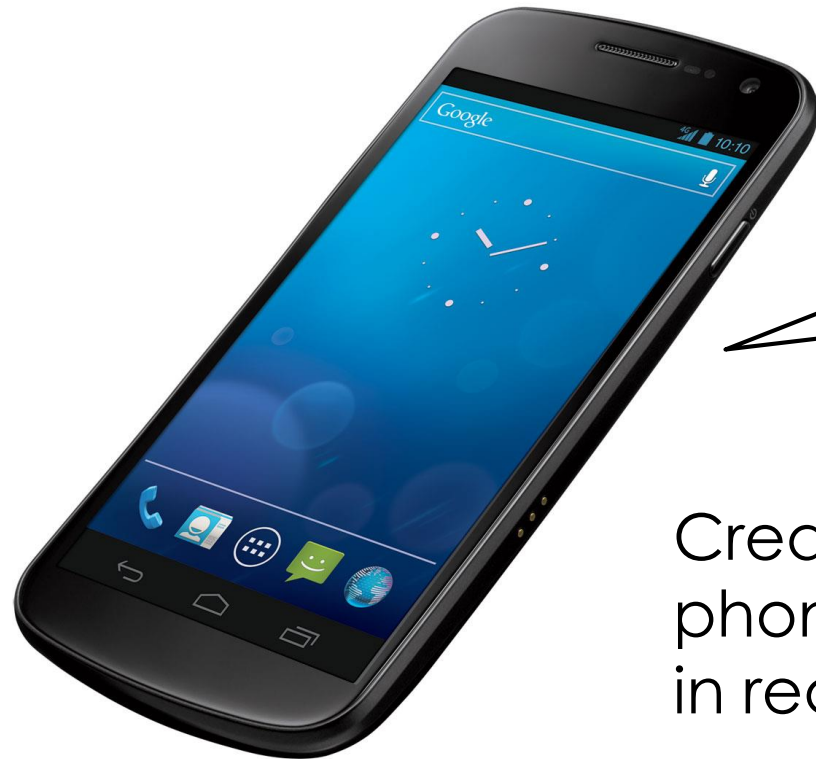
SUPERVISED BY ITAMAR KATZ

Signal and image processing lab, Technion, spring 2014

Motivation

- ▶ Reminder of last parking spot
- ▶ Medical monitoring
- ▶ Automatic actions based on activity
 - ▶ Automatically switch to driving mode

Project Goal



I know what
you did last
summer

Create an Android application that uses the phone's sensors to determine human activity in real time.



Background

Smartphones

- ▶ Processing power

- ▶ Sensors

 - ▶ Accelerometer

 - ▶ Gyroscope (orientation)

50Hz

3-axis

Challenges

Algorithmic

- ▶ Noisy signal
- ▶ Performance

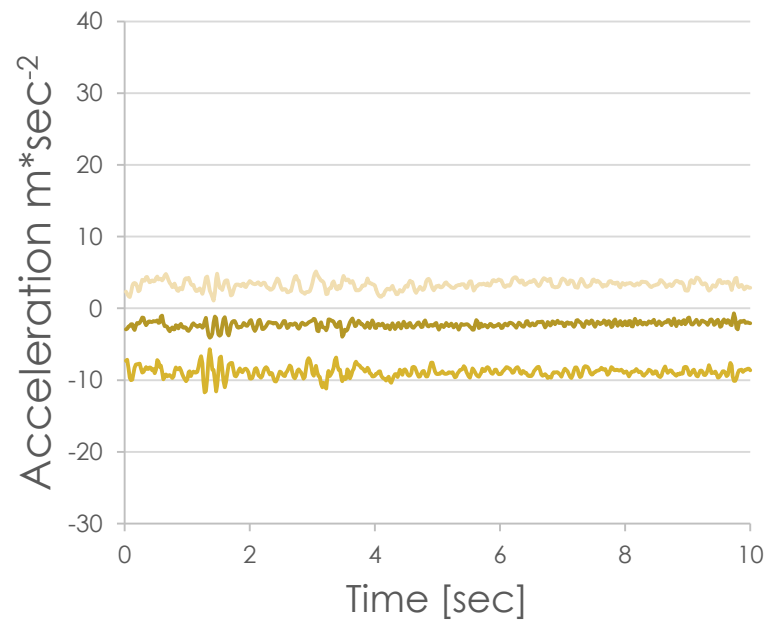
Implementation

- ▶ Concurrency
- ▶ Battery drain

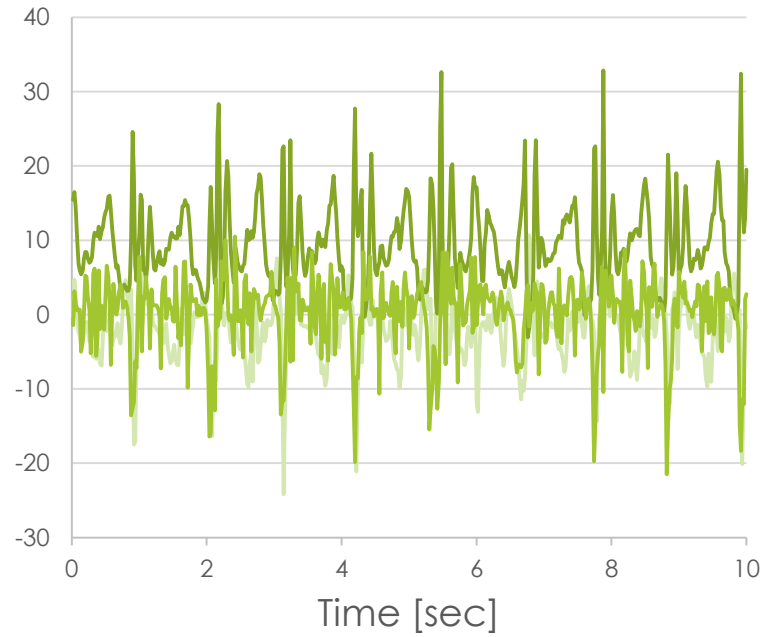
Preparing for Demo

Raw Data - Accelerometer

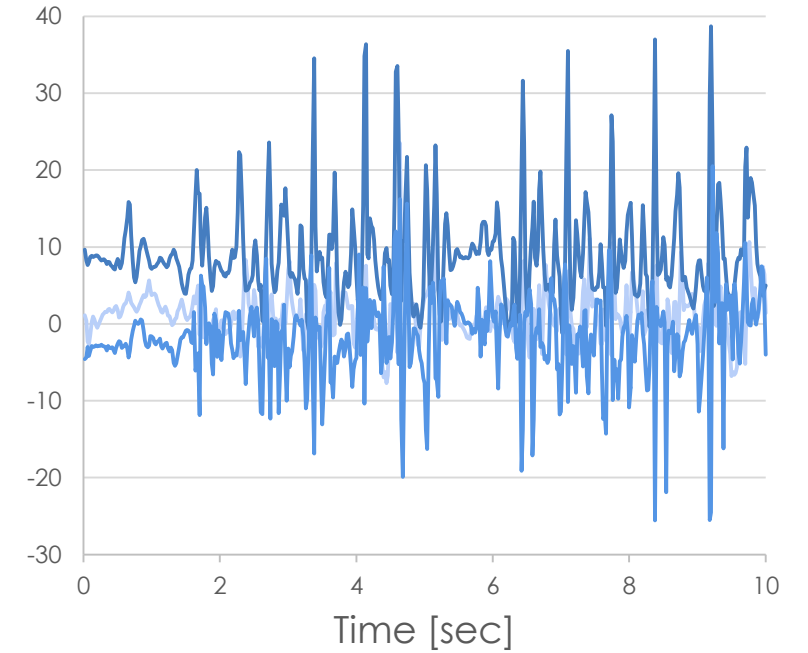
Drive



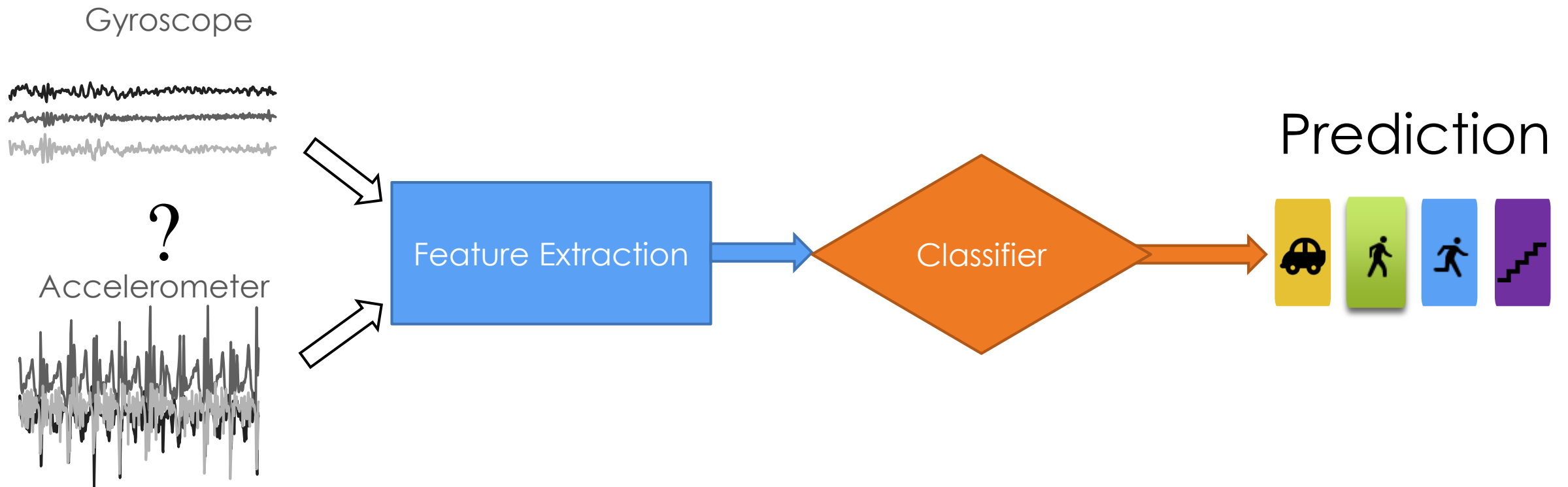
Walk



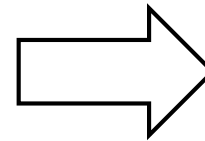
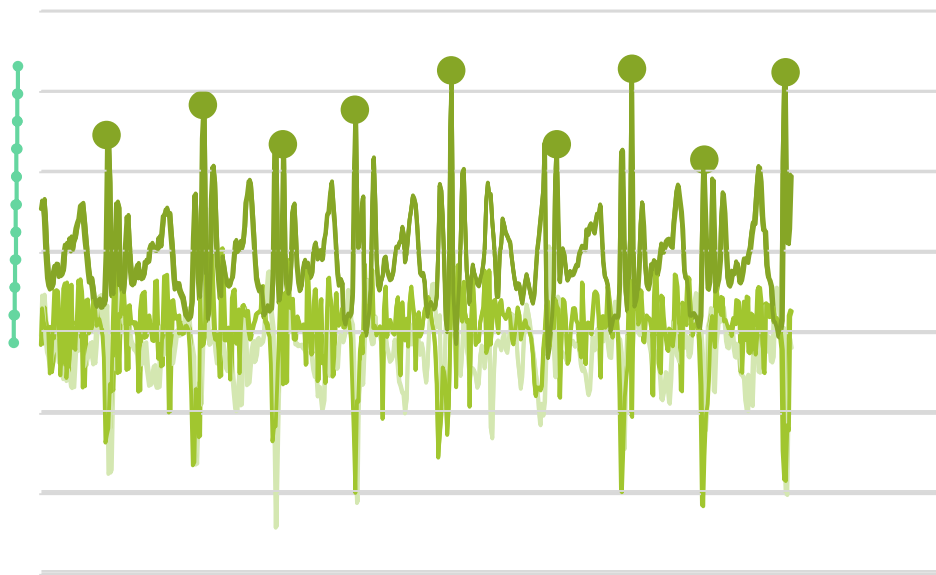
Stairs



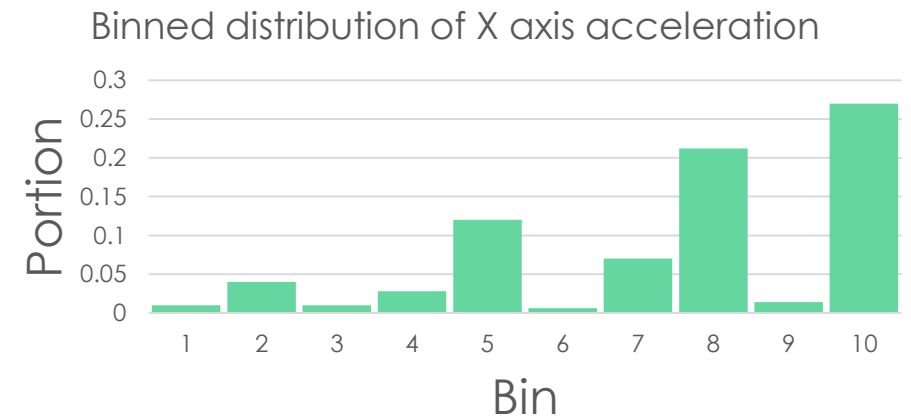
Process Overview



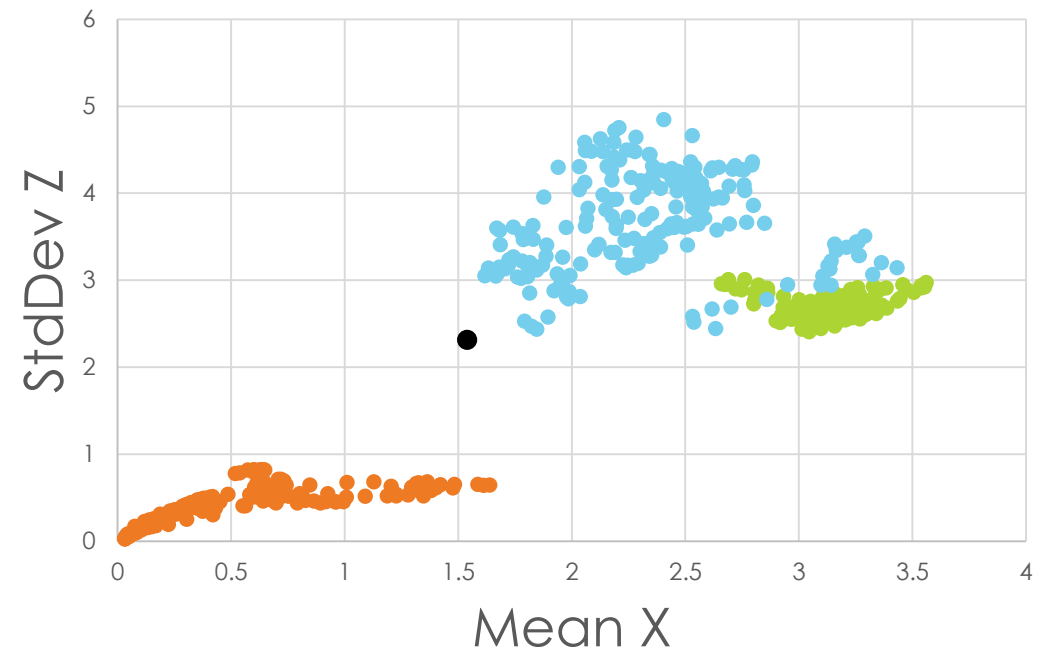
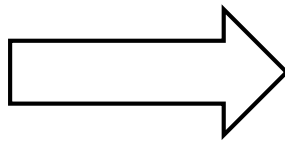
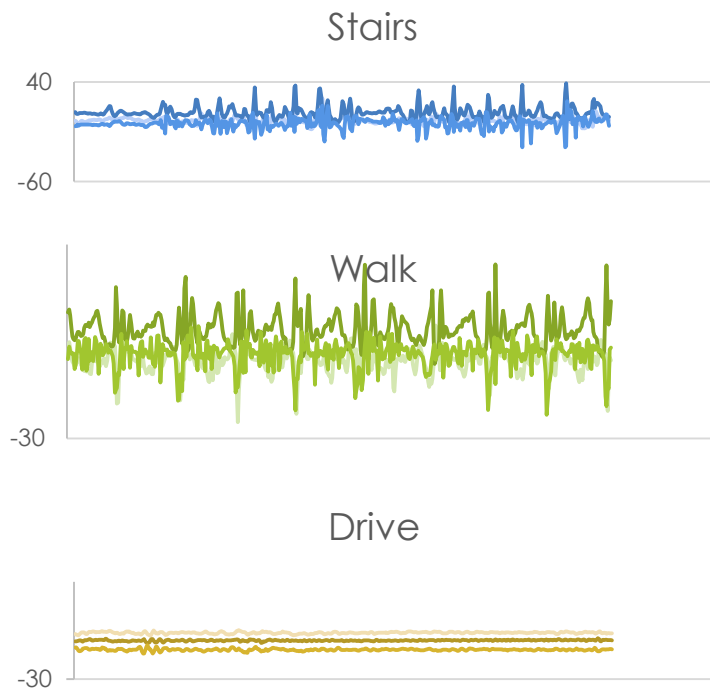
Feature Extraction



- average acceleration
- standard deviation
- peak distance
- binned distribution

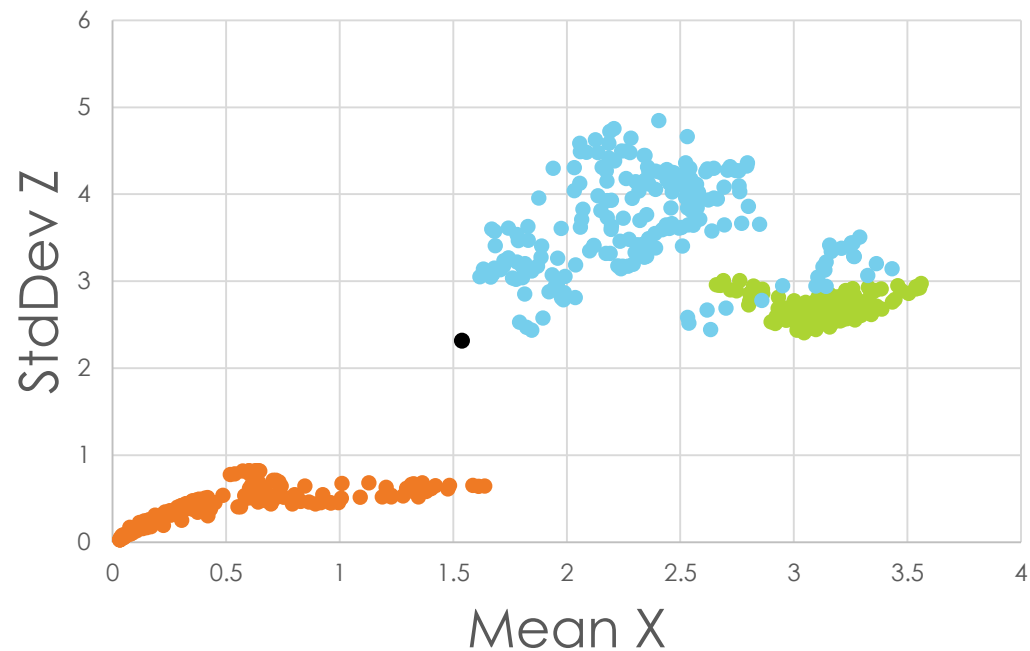


Feature Extraction



Classification

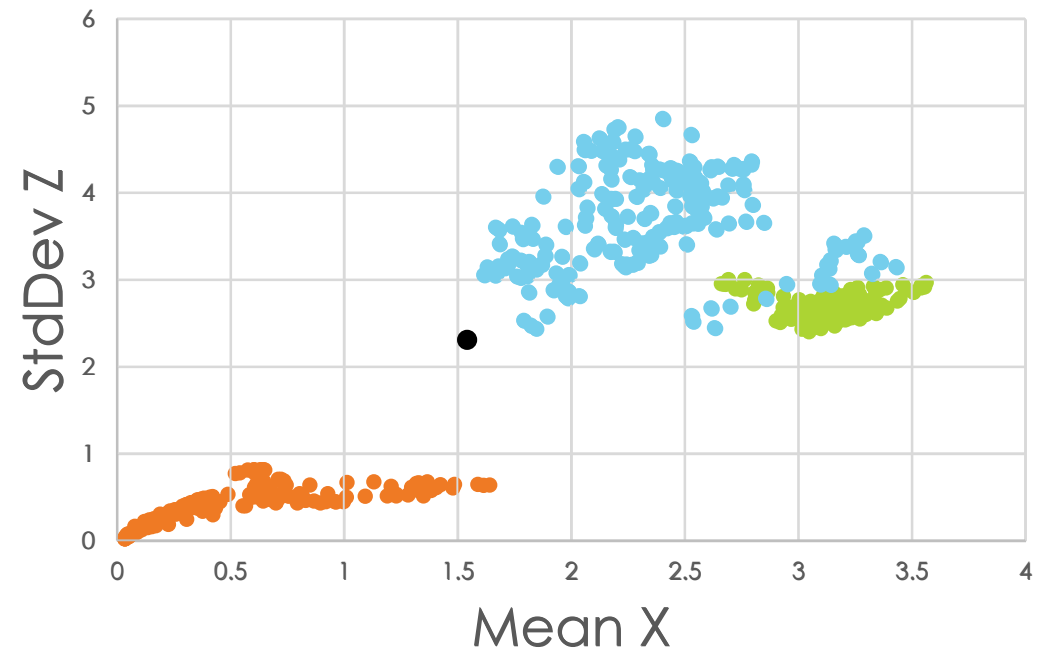
- ▶ Goal: prior data + feature vector = prediction
- ▶ Our classifier: Naïve Bayes
 - ▶ Quick
 - ▶ Small training set
 - ▶ Handles overlapping



Classifiers – Naïve Bayes

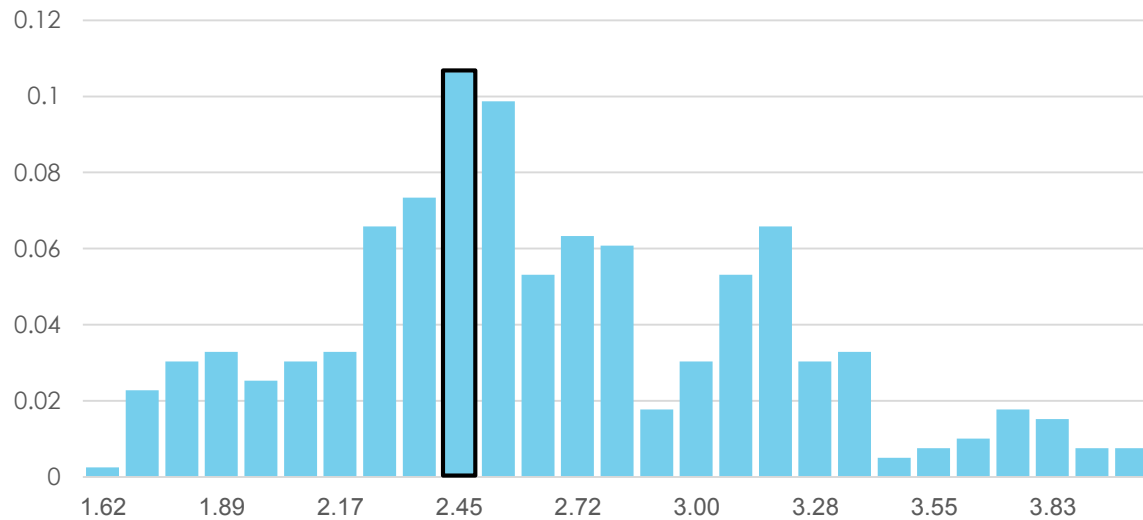
- ▶ Return most likely class, c , given a feature vector, \vec{f} .
- ▶ Assumption: features are independent.

- ▶
$$P(c_i|\vec{f}) = P(c_i)P(\vec{f}|c_i) = P(c_i) \prod_{j=1}^n P(f_j|c_i)$$



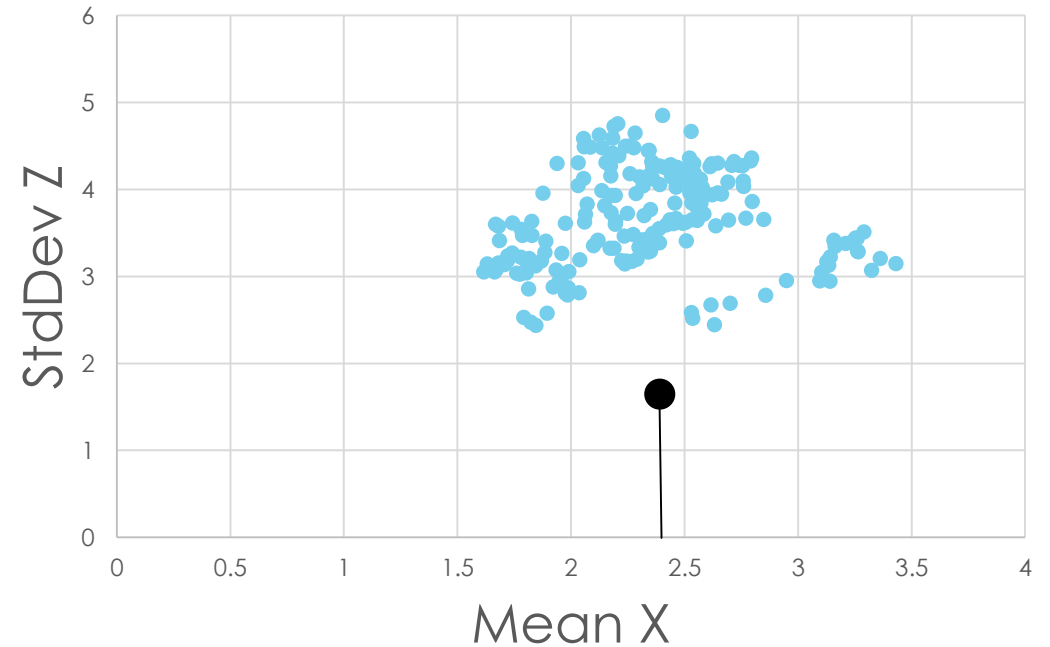
Classifiers – Naïve Bayes

Mean X acceleration distribution of walk data



Mean: 2.61
Std. dev.: 0.51

$$P(f_j | c_i)$$



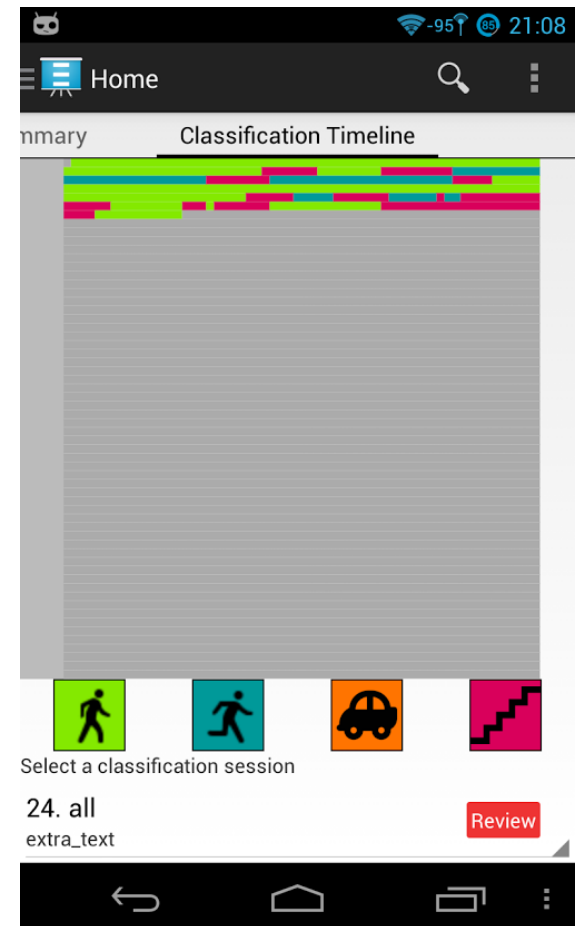
Classifiers – Naïve Bayes

$$P(c_i|\vec{f}) = P(c_i) \prod_{j=1}^n P(f_j|c_i)$$

$$\text{classify}(f_1, \dots, f_n) = \underset{c_i}{\operatorname{argmax}} P(C = c_i) \prod_{j=1}^n P(F_j = f_j|C = c_i)$$

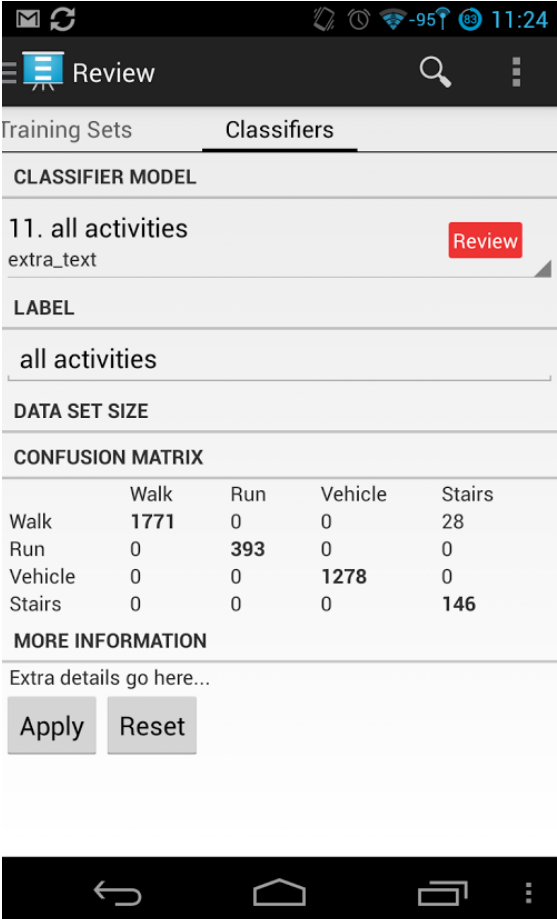
Android Implementation

- ▶ Used a Machine learning lib called *Weka*.
- ▶ Used services that run in separate threads.
- ▶ Asynchronous communication with the UI.
- ▶ UI design and implementation takes a lot of time.



Results and conclusion

- ▶ We created an application that determine human activity in real time using the phone's sensors.
- ▶ Typical accuracy – 94~96%.
- ▶ Sometimes confuses between stairs and walk.



The screenshot shows a mobile application interface with a dark header and a light background. The header includes a search icon and a menu icon. Below the header, there are two tabs: 'Training Sets' and 'Classifiers'. The 'Classifiers' tab is active. The main content area displays the following information:

- CLASSIFIER MODEL**: 11. all activities (with a red 'Review' button) and extra_text.
- LABEL**: all activities.
- DATA SET SIZE**: (Not explicitly shown, but implied by the context).
- CONFUSION MATRIX**: A table showing the relationship between predicted and actual activities.
- MORE INFORMATION**: Extra details go here... (with 'Apply' and 'Reset' buttons).

	Walk	Run	Vehicle	Stairs
Walk	1771	0	0	28
Run	0	393	0	0
Vehicle	0	0	1278	0
Stairs	0	0	0	146

Future suggestions

- ▶ More extensive training set
- ▶ More features
- ▶ Use more sensors
- ▶ Add activities
- ▶ Try other classifiers

Demo time