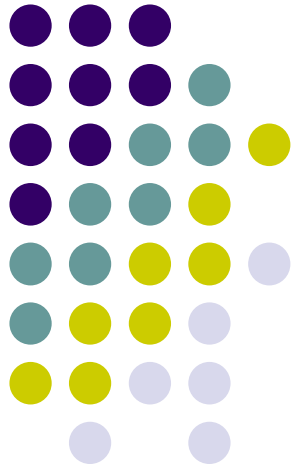
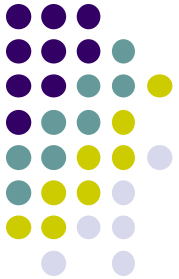
CS 528 Mobile and Ubiquitous Computing

Lecture 7a: Applications of Activity Recognition + Machine Learning for Ubiquitous Computing

Emmanuel Agu

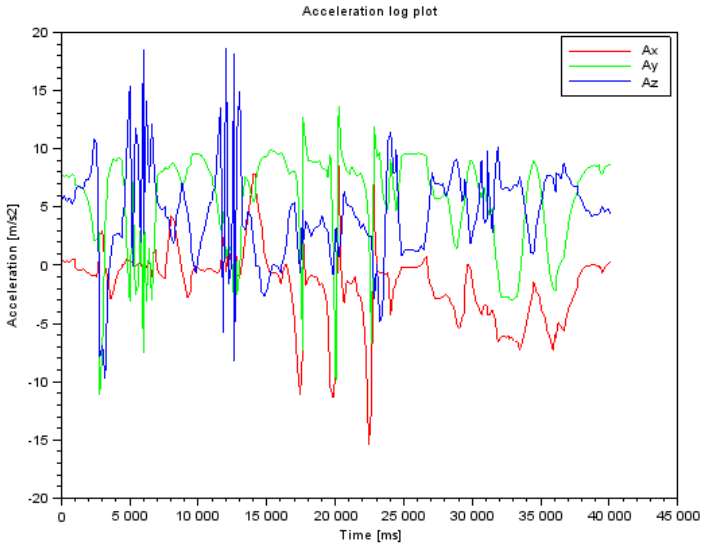




Applications of Activity Recognition

Recall: Activity Recognition

- **Goal:** Want our app to detect what activity the user is doing?
- **Classification task:** which of these 6 activities is user doing?
  - Walking,
  - Jogging,
  - Ascending stairs,
  - Descending stairs,
  - Sitting,
  - Standing

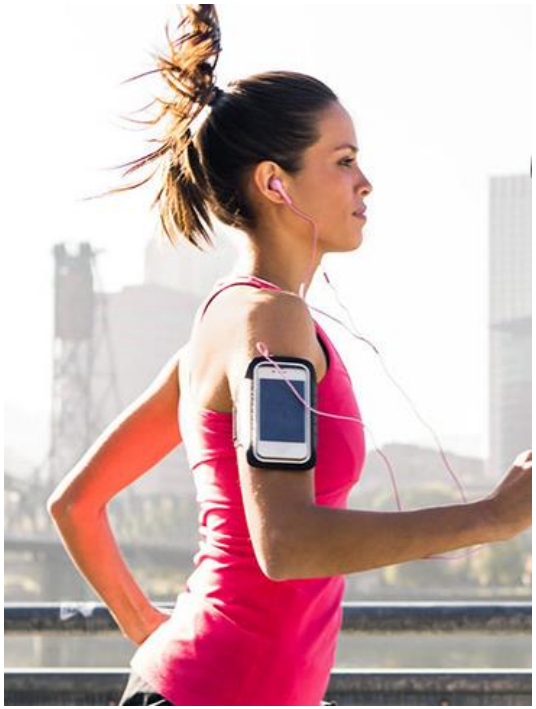


- Typically, use machine learning classifers to classify user's accelerometer signals

Applications of Activity Recognition (AR)

Ref: Lockhart *et al,* Applications of Mobile Activity recognition

- **Fitness Tracking:**
  - **Initially:**
    - Physical activity type,
    - Distance travelled,
    - Calories burned
  - **Newer features:**
    - Stairs climbed,
    - Physical activity (duration + intensity)
    - Activity type logging + context e.g. Ran 0.54 miles/hr faster during morning runs
    - Sleep tracking
    - Activity history



Note: AR refers to algorithm
But could run on a range of devices (smartphones, wearables, e.g. fitbit)

Applications of Activity Recognition (AR)

Ref: Lockhart *et al,* Applications of Mobile Activity recognition

- **Health monitoring:** How **well** is patient performing activity?
- Make clinical monitoring pervasive, continuous, real world!!
  - Gather context information (e.g. what makes condition worse/better?)
  - E.g. timed up and go test
- Show patient contexts that worsen condition => Change behavior
  - E.g. walking in narror hallways worsens gait freeze

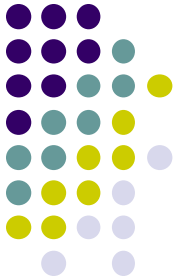
Parkinsons disease
Gait freezing

Question: What data would you need to build PD gait classifier? From what types of subjects?

COPD, Walk tests in the wild

Applications of Activity Recognition

Ref: Lockhart *et al,* Applications of Mobile Activity recognition

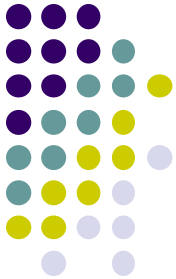


- **Fall:** Leading cause of death for seniors
- **Fall detection:** Smartphone/watch, wearable detects senior who has fallen, alert family
  - Text message, email, call relative

Fall detection + prediction

Applications of Activity Recognition (AR)

Ref: Lockhart *et al,* Applications of Mobile Activity recognition



- **Context-Aware Behavior:**
  - In-meeting? => Phone switches to silent mode
  - Exercising? => Play song from playlist, use larger font sizes for text
  - Arrived at work? => download email

- Study found that messages delivered when transitioning between activities better received

- **Adaptive Systems to Improve User Experience:**
  - Walking, running, riding bike? => Turn off Bluetooth, WiFi (save power)
  - Can increase battery life up to 5x

Applications of AR

Ref: Lockhart *et al,* Applications of Mobile Activity recognition

- **Smart home:**
  - Determine what activities people in the home are doing,
    - **Why?** infer illness, wellness, patterns, intrusion (security), etc
    - E.g. TV automatically turns on at about when you usually lie on the couch

Applications of AR: 3rd Party Apps

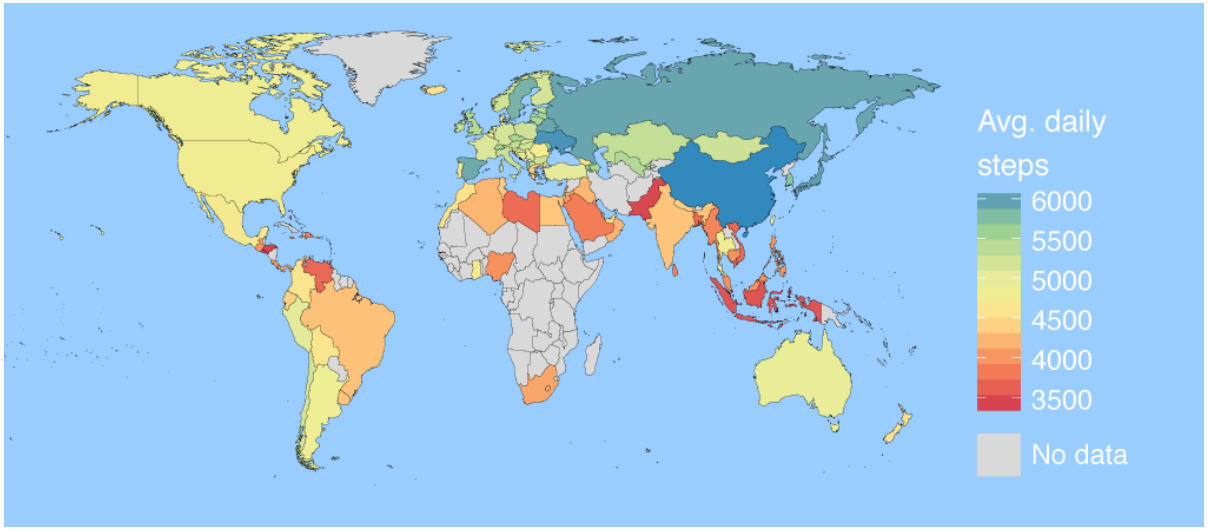
Ref: Lockhart *et al,* Applications of Mobile Activity recognition

- **Targeted Advertising:**
  - AR helps deliver more relevant ads
  - E.g user runs a lot => Get exercise clothing ads
  - Goes to pizza places often + sits there => Get pizza ads

Applications of AR: 3rd Party Apps

Ref: Lockhart *et al,* Applications of Mobile Activity recognition

- **Research Platforms for Data Collection:**
  - E.g. public health officials want to know how much time various people (e.g. students) spend sleeping, walking, exercising, etc
  - Mobile AR: inexpensive, automated data collection
  - E.g. Stanford Inequality project: Analyzed physical activity of 700k users in 111 countries using smartphone AR data
  - http://activityinequality.stanford.edu/



Applications of AR: 3rd Party Apps

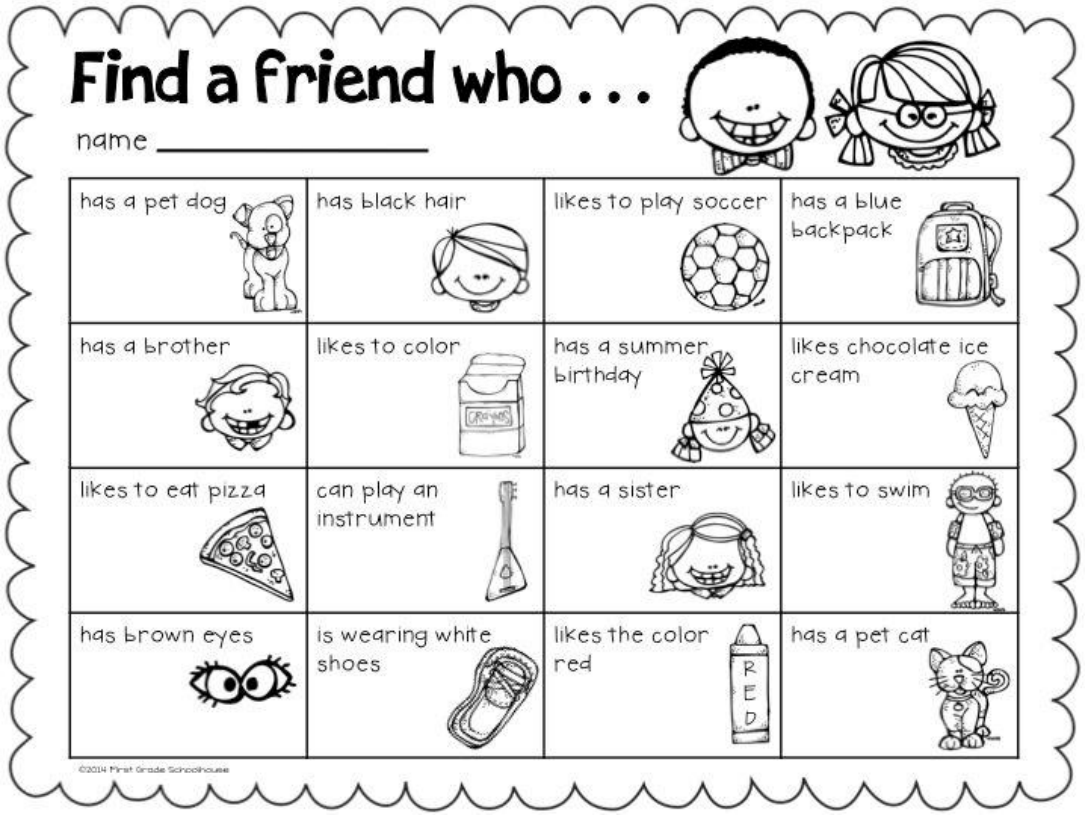
Ref: Lockhart *et al,* Applications of Mobile Activity recognition

- **Track, manage staff on-demand:**
  - E.g. at hospital, determine “availability of nurses”, assign them to new jobs/patients/surgeries/cases

Applications of AR: Social Networking

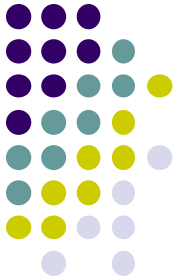
Ref: Lockhart *et al,* Applications of Mobile Activity recognition

- **Activity-Based Social Networking:**
  - Automatically connect users who do same activities + live close together



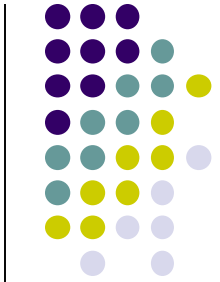
Applications of AR: Social Networking

Ref: Lockhart *et al,* Applications of Mobile Activity recognition



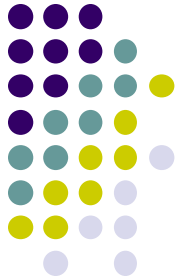
- **Activity-Based Place Tagging:**
  - Automatically “popular” places where users perform same activity
  - E.g. Park street is popular for runners (activity-based maps)

- **Automatic Status updates:**
  - E.g. Bob is sleeping
  - Tracy is jogging along Broadway with track team
  - Privacy/security concerns => Different Levels of details for different friends



Intuitive Introduction to Machine Learning for Ubiquitous Computing

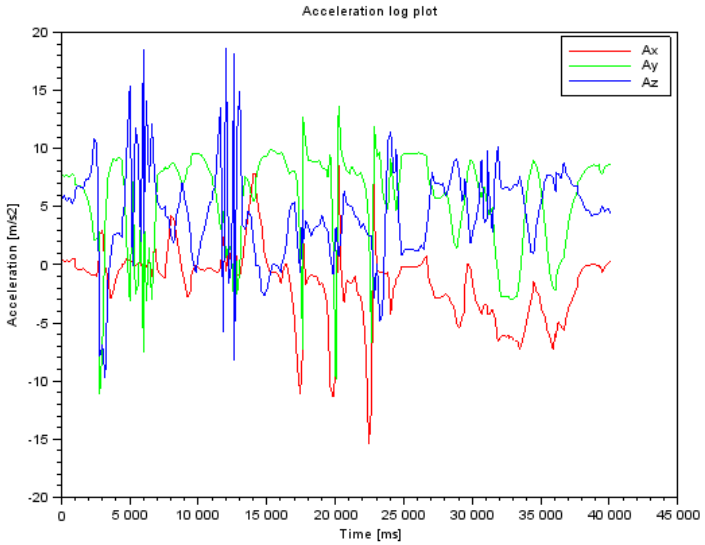
My Goals in this Section



- If you know machine learning
  - Set off light bulb
  - Projects involving ML?
- If you don’t know machine learning
  - Get general idea, how it’s used
- Knowledge will also make papers easier to read/understand

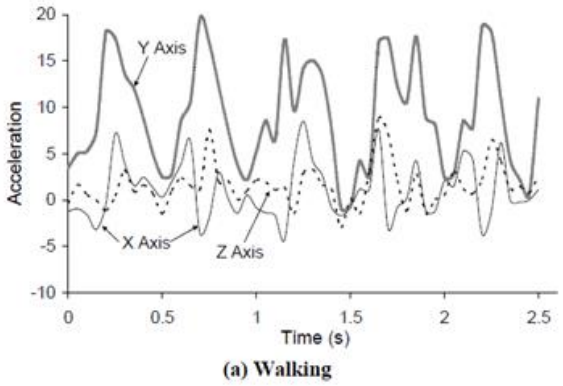
Recall: Activity Recognition

- Want app to detect when user is performing any of the following 6 activities
  - Walking,
  - Jogging,
  - Ascending stairs,
  - Descending stairs,
  - Sitting,
  - Standing



Recall: Activity Recognition Overview

Gather Accelerometer data



(a) Walking

Machine Learning Classifier

Classify Accelerometer data

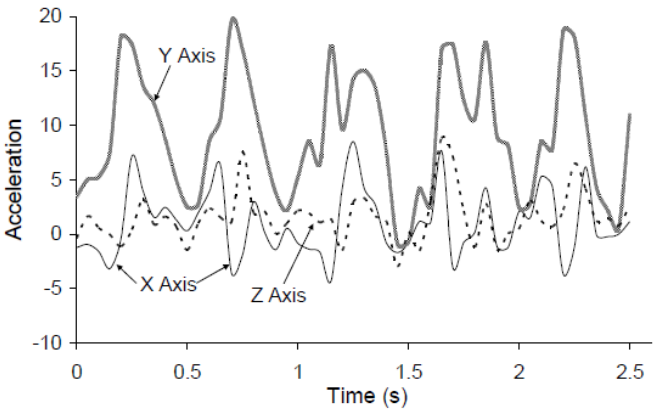
Walking

Running

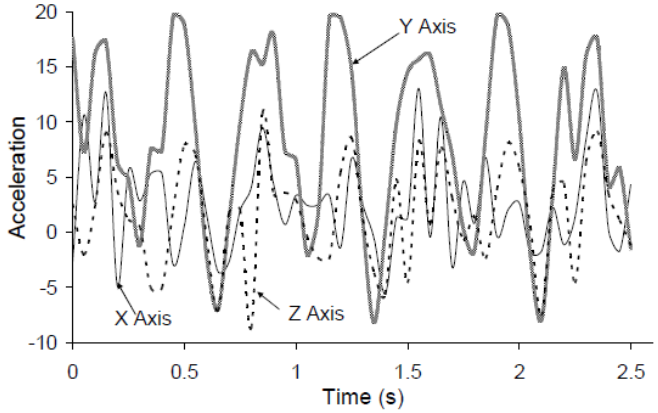
Climbing Stairs

Recall: Example Accelerometer Data for Activities

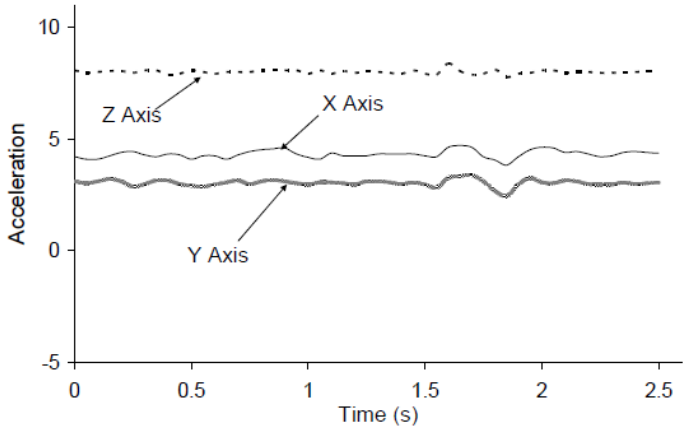
Different user activities generate different accelerometer patterns



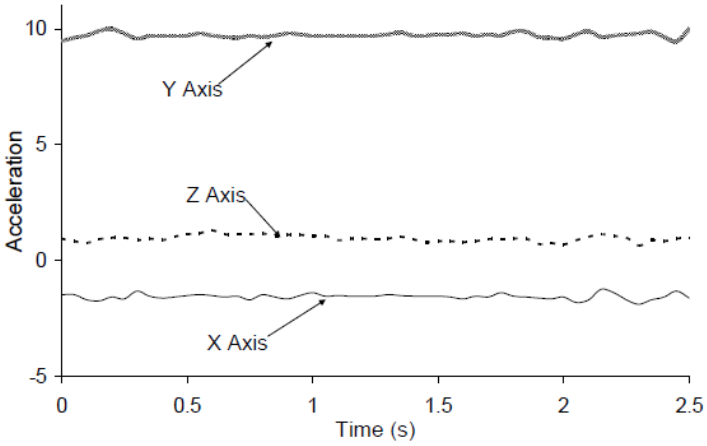
(a) Walking



(b) Jogging



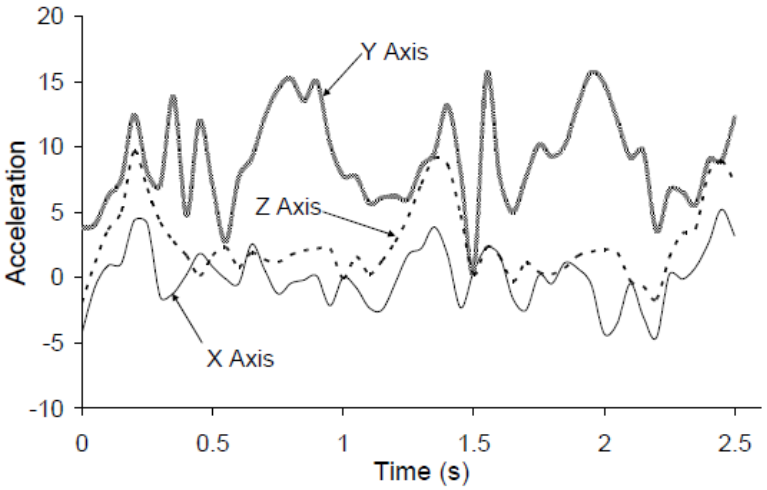
(e) Sitting



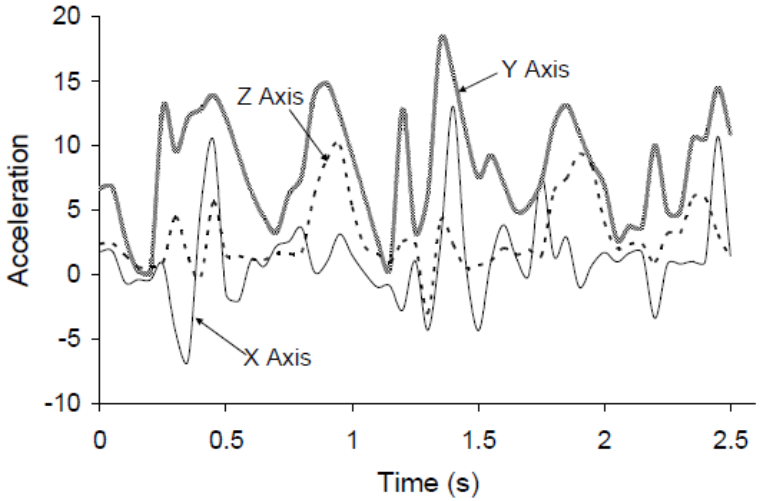
(f) Standing

Recall: Example Accelerometer Data for Activities

Different user activities generate different accelerometer patterns



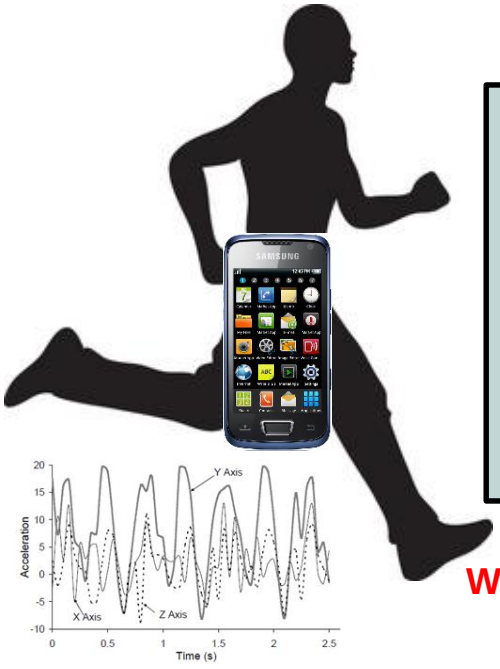
(c) Ascending Stairs

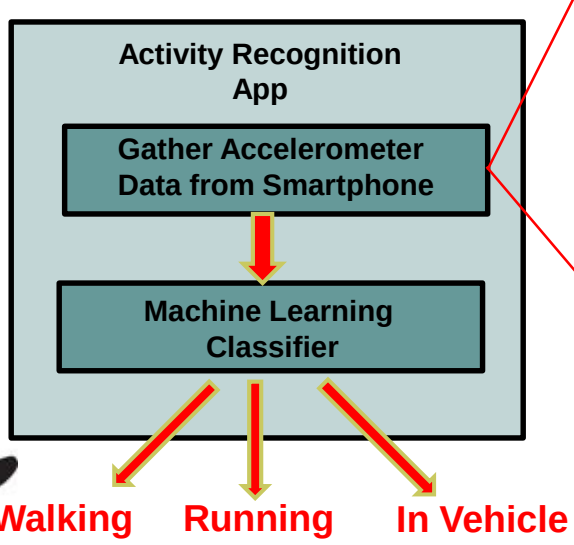


(d) Descending Stairs

DIY Activity Recognition (AR) Android App

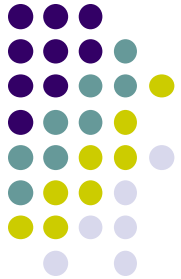
- As user performs an activity, AR app on user's smartphone
  1. Gathers accelerometer data
  2. Uses **machine learning classifier** to determine what activity (running, jumping, etc) accelerometer pattern corresponds to
- **Classifier:** Machine learning algorithm that guesses what activity **class** accelerometer sample corresponds to






```
msensor = (mSensorManager)
    getSystemService(Context.SENSOR_SERVICE)

....

Public void onSensorChanged(SensorEvent event){

....

}
```

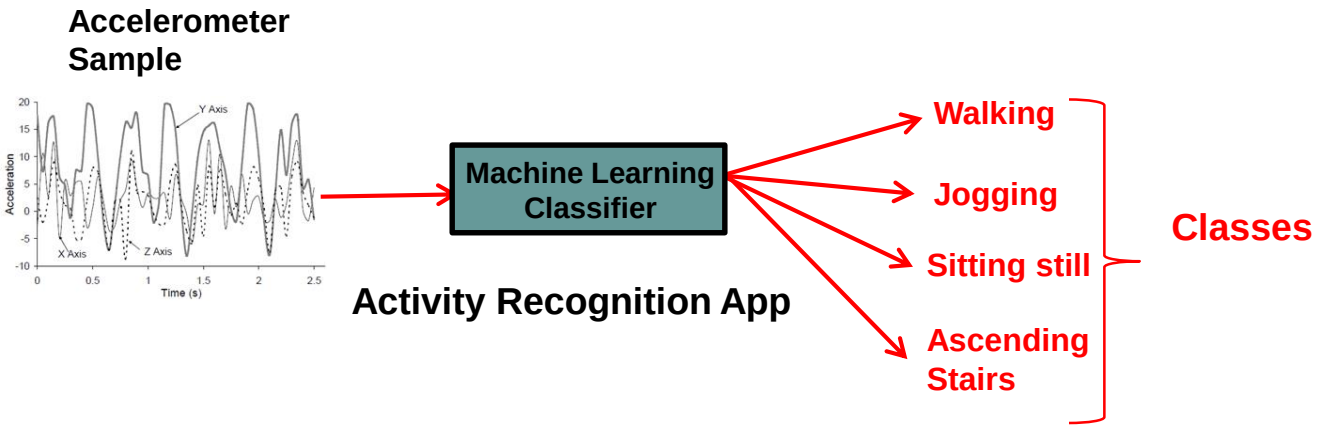
Next: Machine learning Classification

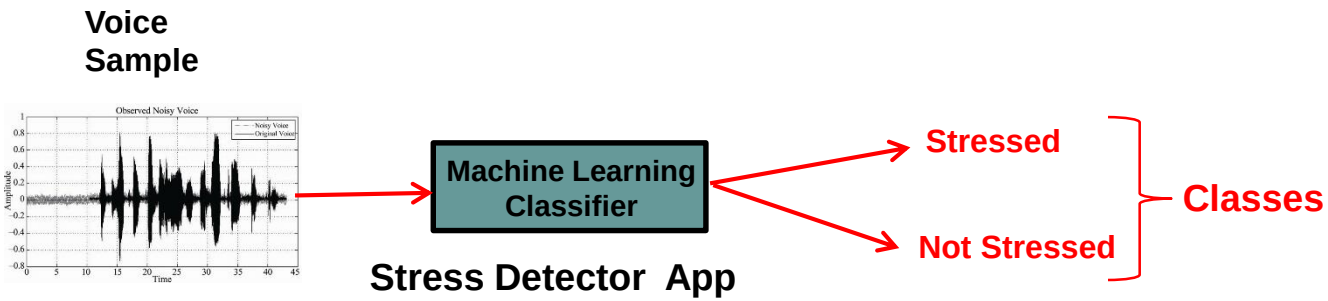


Classification for Ubiquitous Computing

Classification

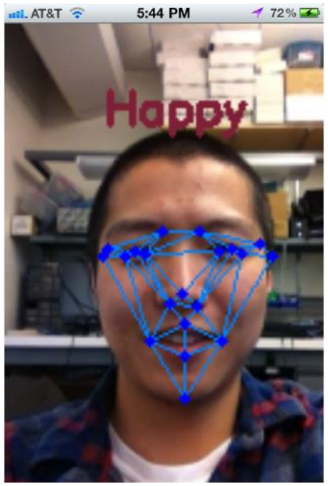
- **Classification** is type of machine learning used a lot in Ubicomp
- Classification? determine which **class** a sample belongs to. Examples:





Classification

Image showing
Facial Expression



Machine Learning Classifier

- Anger
- Disgust
- Fear
- Happy
- Neutral
- Sadness
- Surprise

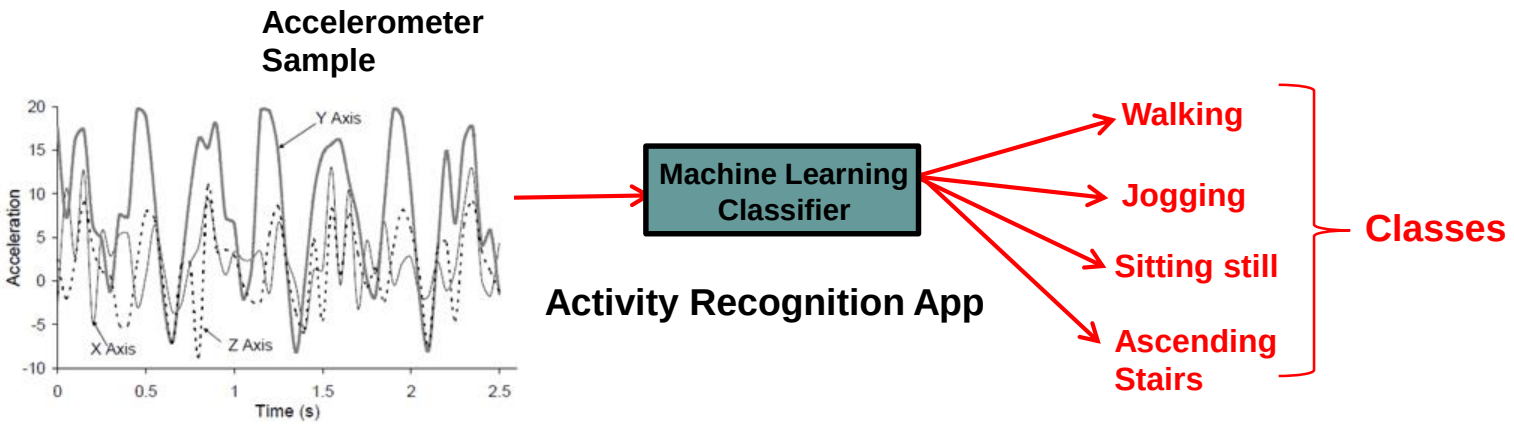
Classes

Facial Interpretation
App

Classifier

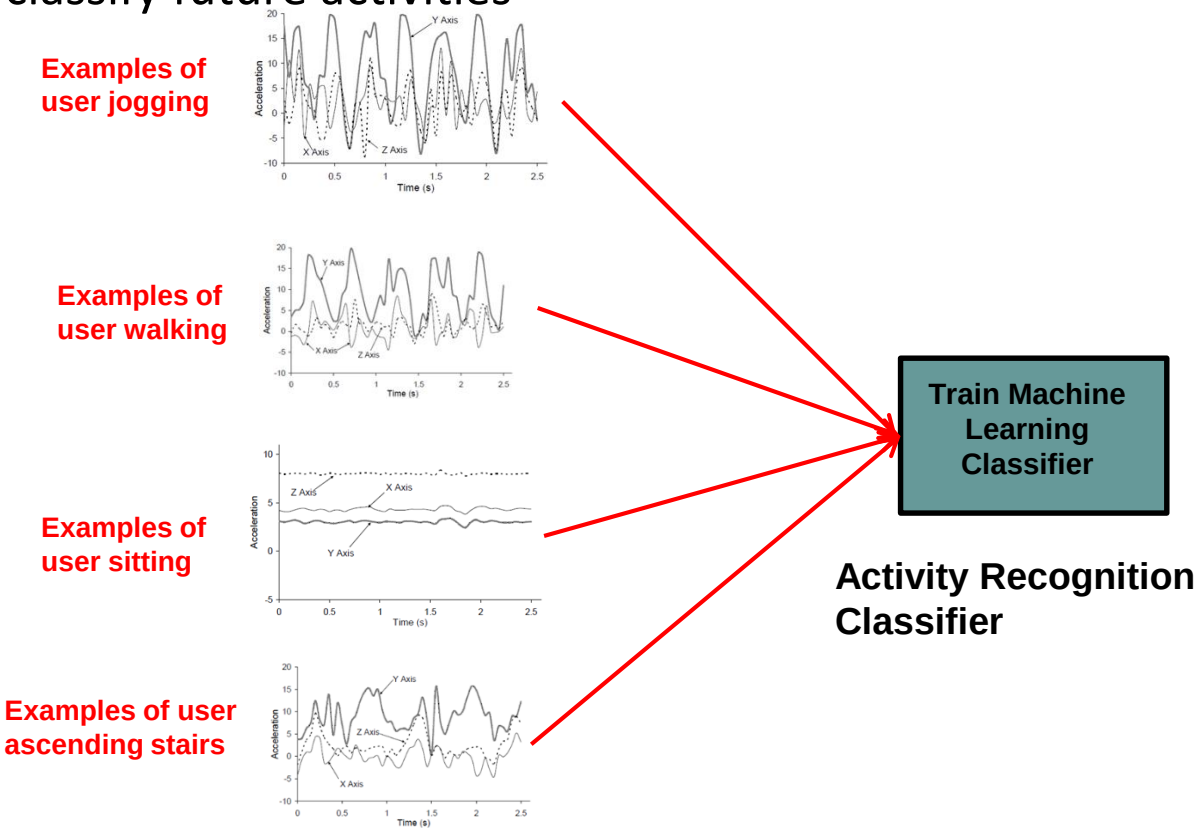
- Analyzes new sample, guesses corresponding class
- Intuitively, can think of classifier as set of rules for classification. E.g.
- Example rules for classifying accelerometer signal in Activity Recognition

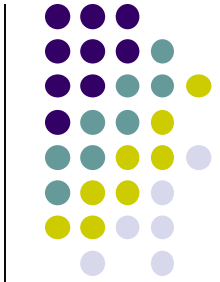
```
If ((Accelerometer peak value > 12 m/s)
      and (Accelerometer average value < 6 m/s)){
            Activity = "Jogging";
}
```



Training a Classifier

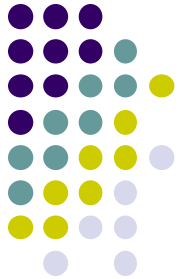
- Created using example-based approach (called training)
- **Training a classifier:** Given examples of each class => generate rules to categorize new samples
- **E.g:** Analyze 30+ Examples (from 30 subjects) of accelerometer signal for each activity type (walking, jogging, sitting, ascending stairs) => generate rules (classifier) to classify future activities





Training a Classifier: Steps

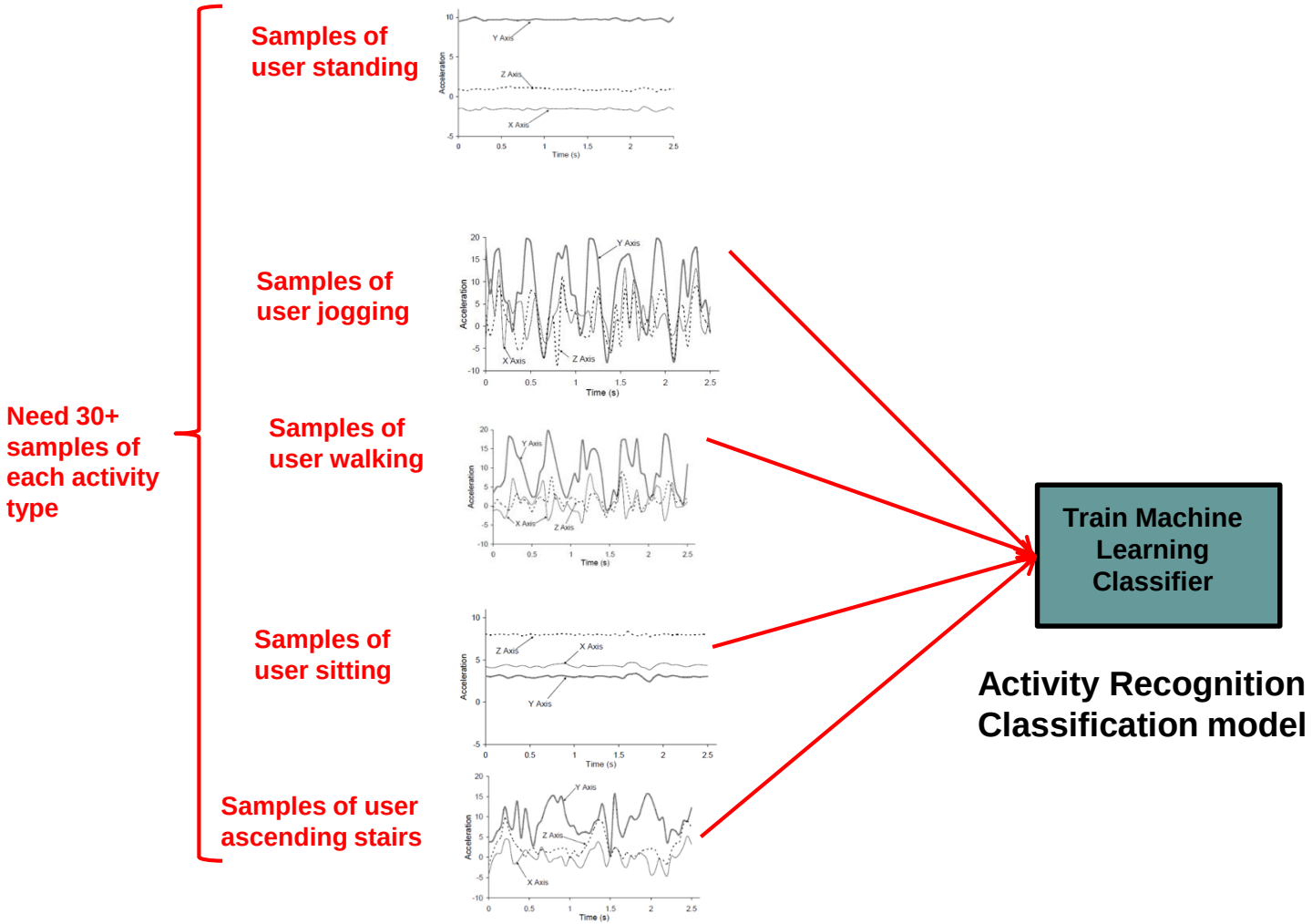
Steps for Training a Classifier



1. Gather data samples + label them
2. Import accelerometer samples into classification library (e.g. Weka, MATLAB)
3. Pre-processing (segmentation, smoothing, etc)
4. Extract features
5. Train classifier
6. Export classification model as JAR file
7. Import into Android app

Step 1: Gather Sample data + Label them

- Need many samples of accelerometer data corresponding to each activity type (jogging, walking, sitting, ascending stairs, etc)



Step 1: Gather Sample data + Label them

- Run a study to gather sample accelerometer data for each activity class
  - Recruit 30+ subjects
  - Run program that gathers accelerometer sensor data on subject's phone
  - Make subjects perform each activity (walking, jogging, sitting, etc)
  - Collect accelerometer data while they perform each activity (walking, jogging, sitting, etc)
  - Label data. i.e. tag each accelerometer sample with the corresponding activity
- Now have 30 examples of each activity

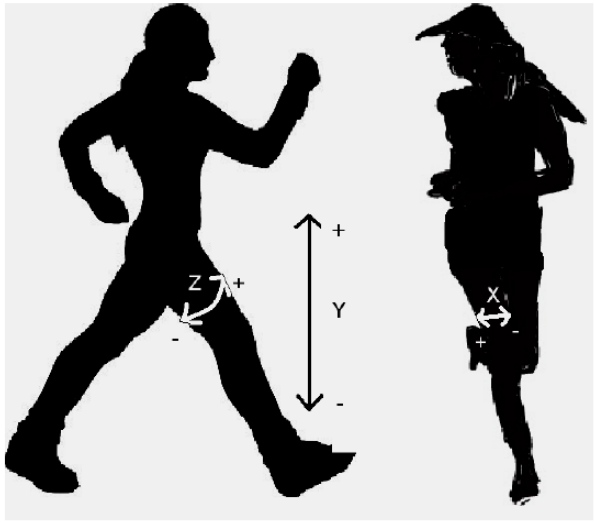
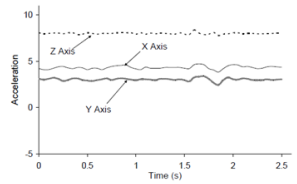
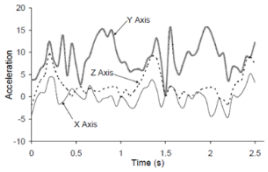


Figure 1: Axes of Motion Relative to User

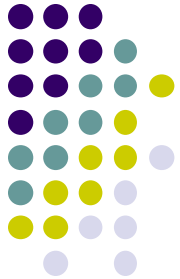
30+ Samples of user sitting



30+ Samples of user ascending stairs



Step 1: Gather Sample data + Label them
Program to Gather Accelerometer Data



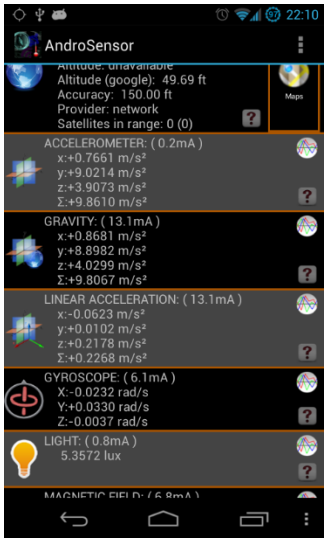
- **Option 1:** Can write sensor program app that gathers accelerometer data while user is doing each of 6 activities (1 at a time)

```
msensor = (mSensorManager)
    getSystemService(Context.SENSOR_SERVICE)

....

Public void onSensorChanged(SensorEvent event){

....

}
```

Step 1: Gather Sample data + Label them
Program to Gather Accelerometer Data

- **Option 2:** Use 3rd party app to gather accelerometer
  - 2 popular ones: **Funf** and **AndroSensor**
  - Just download app,
    - Select sensors to log (e.g. accelerometer)
    - Continuously gathers sensor data in background
- FUNF app from MIT
  - Accelerometer readings
  - Phone calls
  - SMS messages, etc
- AndroSensor

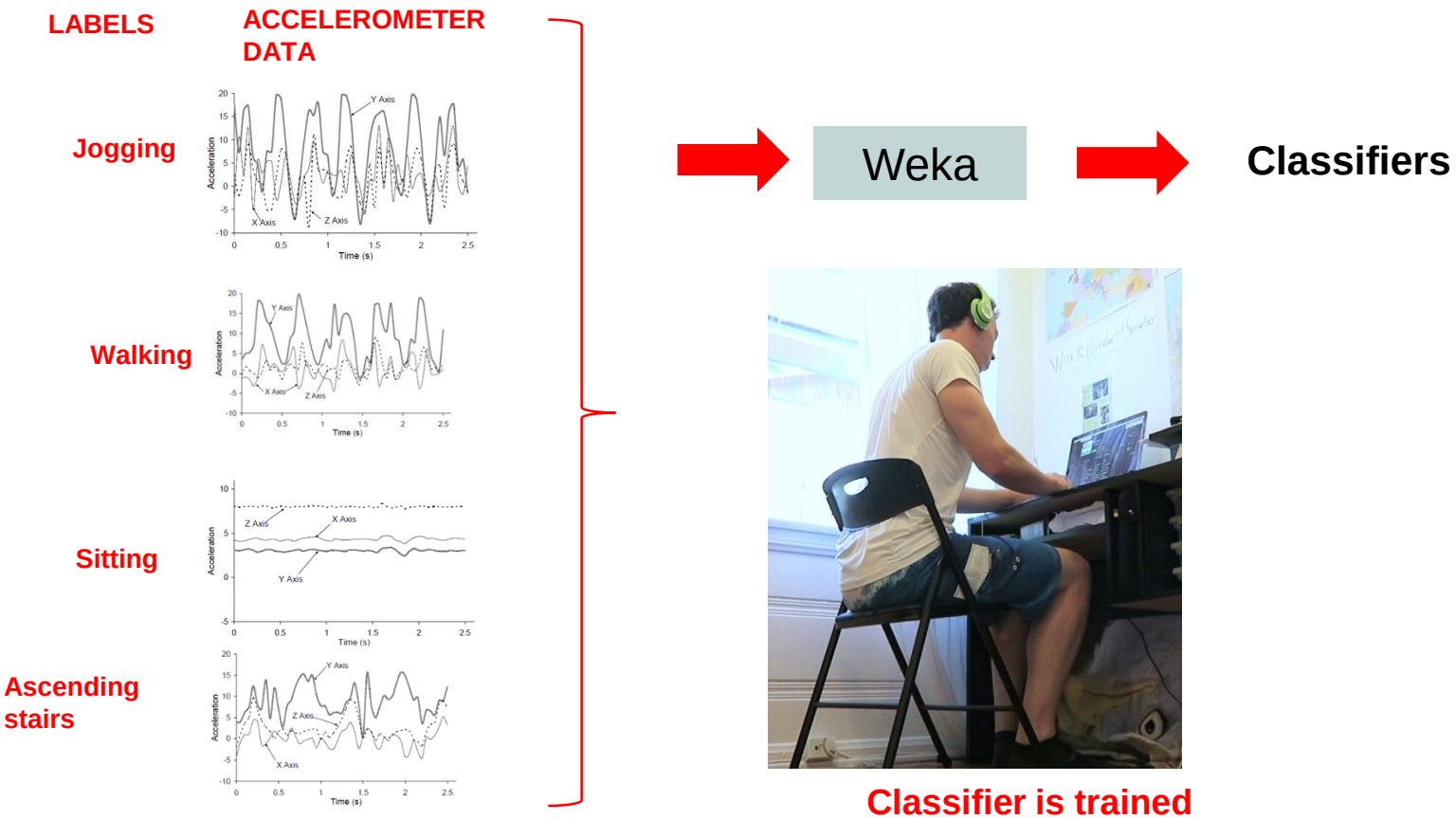
Funf



AndroSensor

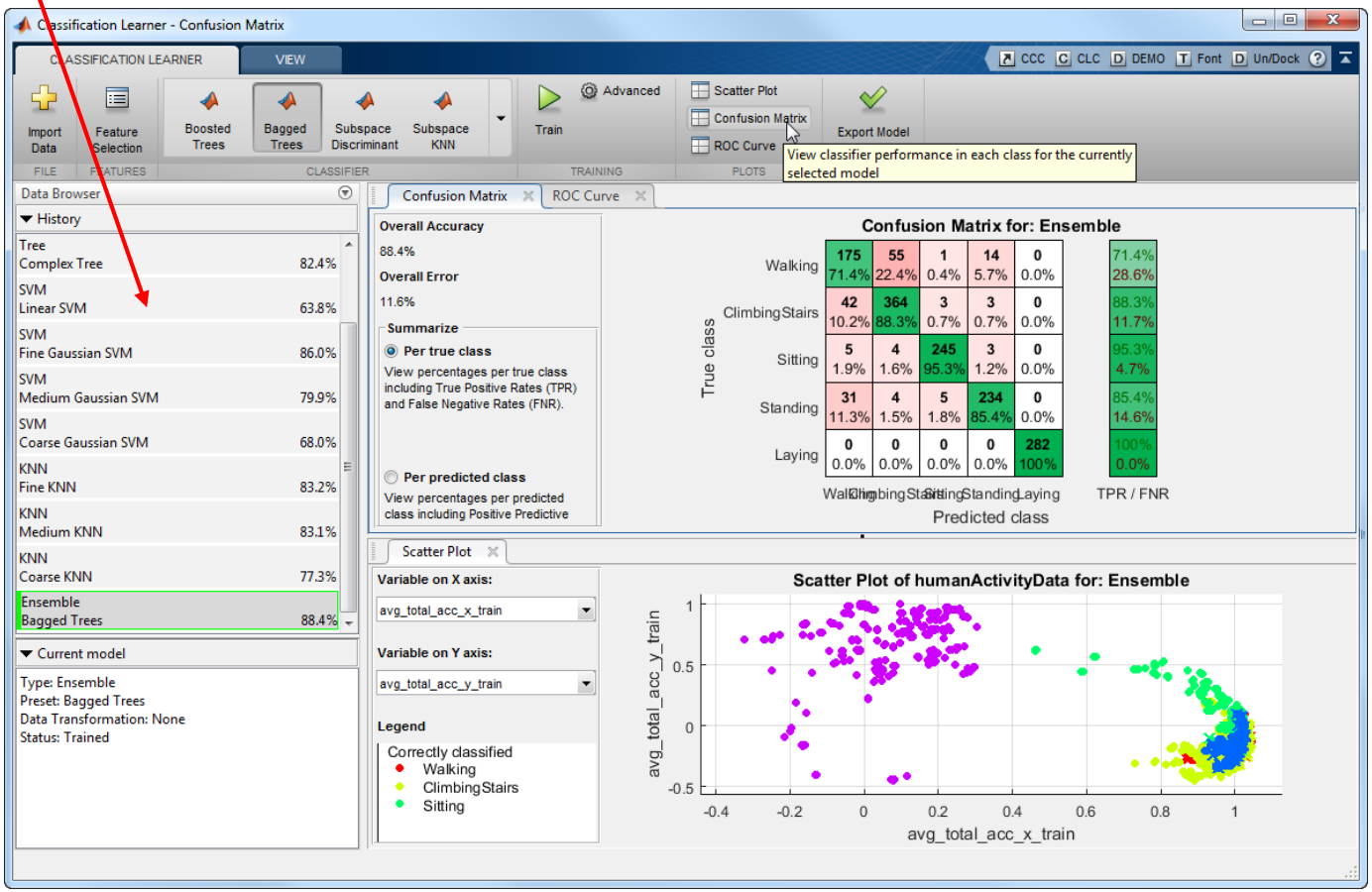
Step 2: Import accelerometer samples into classification library (e.g. Weka, MATLAB)

- Import accelerometer data (labelled with corresponding activity) into Weka, MATLAB (or other Machine learning Framework)



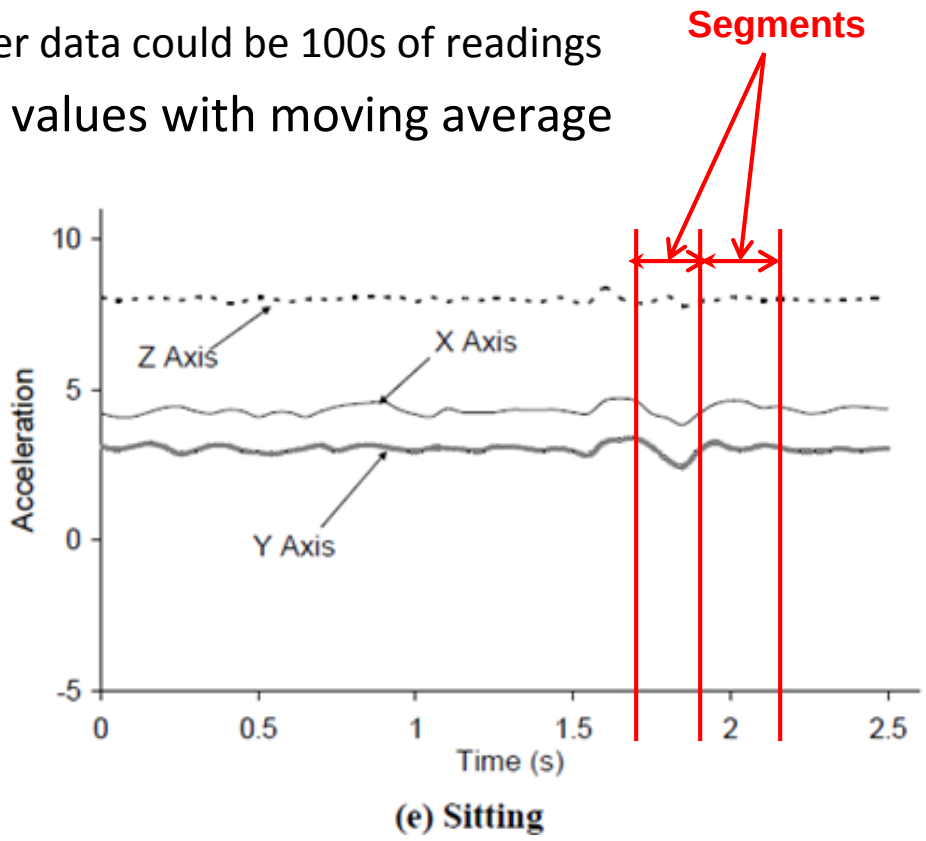
MATLAB Classification Learner App

- Import accelerometer data into MATLAB
- Click and select Classifier types to compare



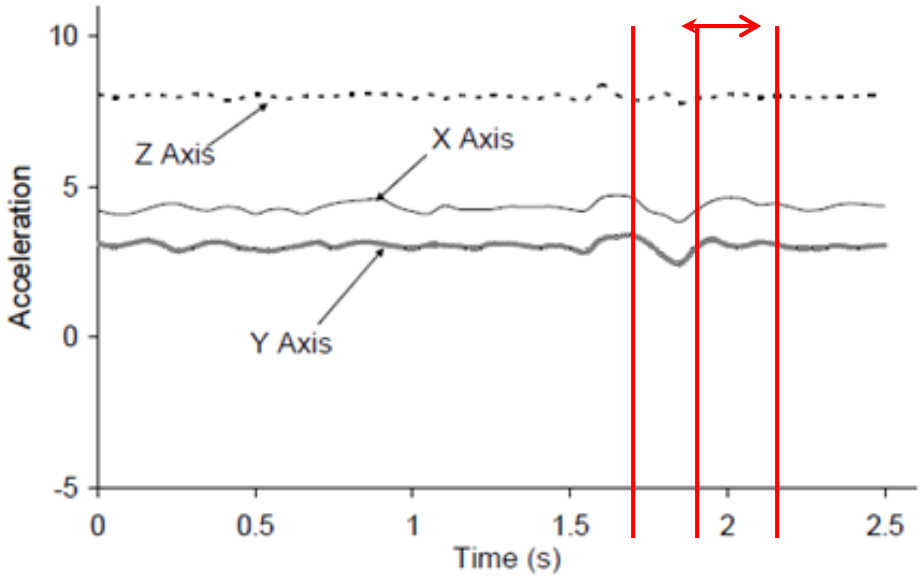
Step 3: Pre-processing (segmentation, smoothing, etc) Segment Data (Windows)

- Pre-processing data (in Weka, or MATLAB) may include segmentation, smoothing, etc
  - **Segment:** Divide 60 seconds of raw time-series data divided into chunks(e.g. 5 seconds)
    - Note: 5 seconds of accelerometer data could be 100s of readings
  - **Smoothing:** Replace groups of values with moving average



Step 4: Compute (Extract) Features

- For each 5-second segment (batch of accelerometer values) compute features (in Weka, MATLAB, etc)
- **Features:** Functions computed on accelerometer data, captures important accelerometer characteristics
- **Examples:** min-max of values, largest magnitude within segment, standard deviation

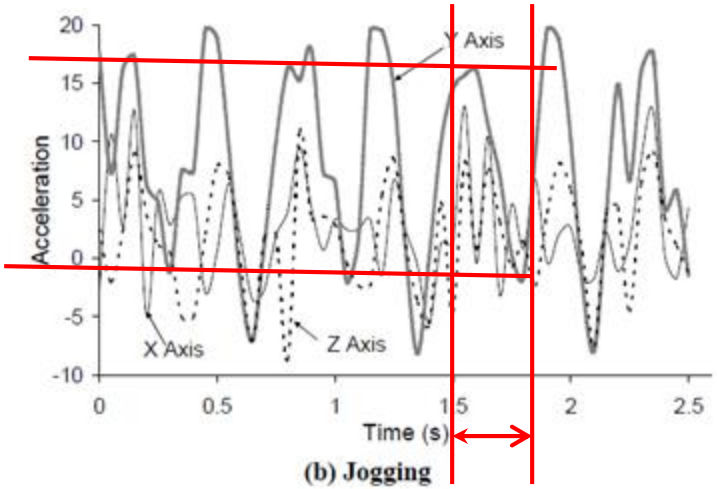


(e) Sitting

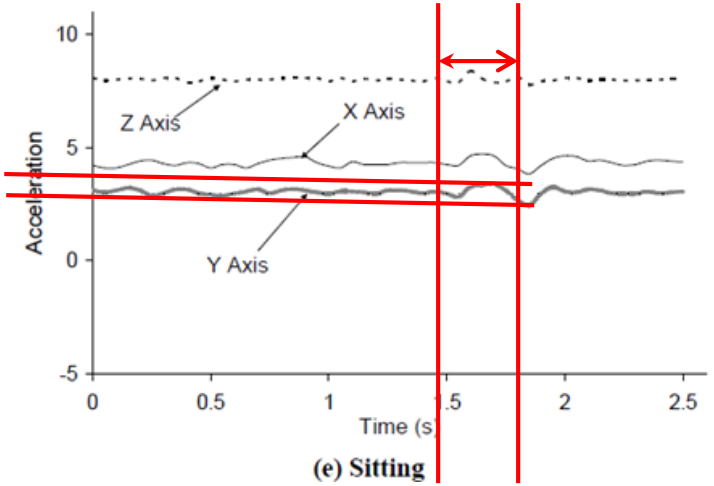
Step 4: Compute (Extract) Features

- **Important:** Ideally, values of features different for, distinguish each activity type
- **E.g:** Min-max range feature

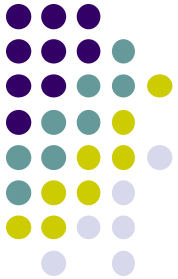
Large min-max for jogging



Small min-max for jogging



Step 4: Compute (Extract) Features

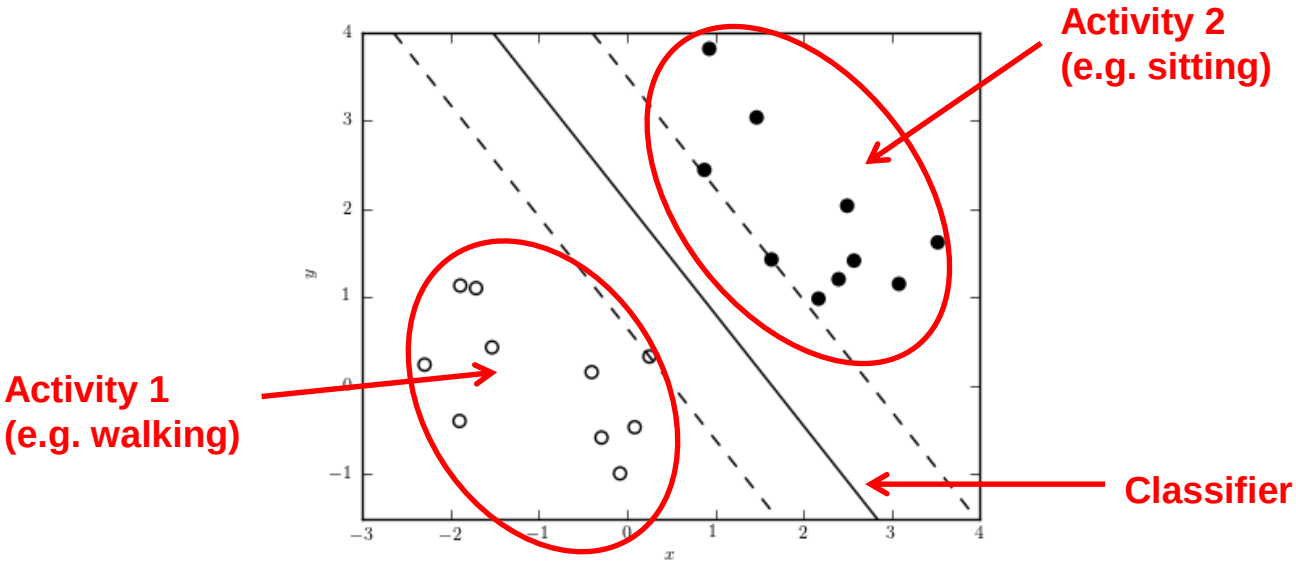


Calculate many different features

- Average[3]: Average acceleration (for each axis)
- Standard Deviation[3]: Standard deviation (for each axis)
- Average Absolute Difference[3]: Average absolute difference between the value of each of the 200 readings within the ED and the mean value over those 200 values (for each axis)
- Average Resultant Acceleration[1]: Average of the square roots of the sum of the values of each axis squared $\sqrt{(x_i^2 + y_i^2 + z_i^2)}$ over the ED
- Time Between Peaks[3]: Time in milliseconds between peaks in the sinusoidal waves associated with most activities (for each axis)
- Binned Distribution[30]: We determine the range of values for each axis (maximum – minimum), divide this range into 10 equal sized bins, and then record what fraction of the 200 values fell within each of the bins.

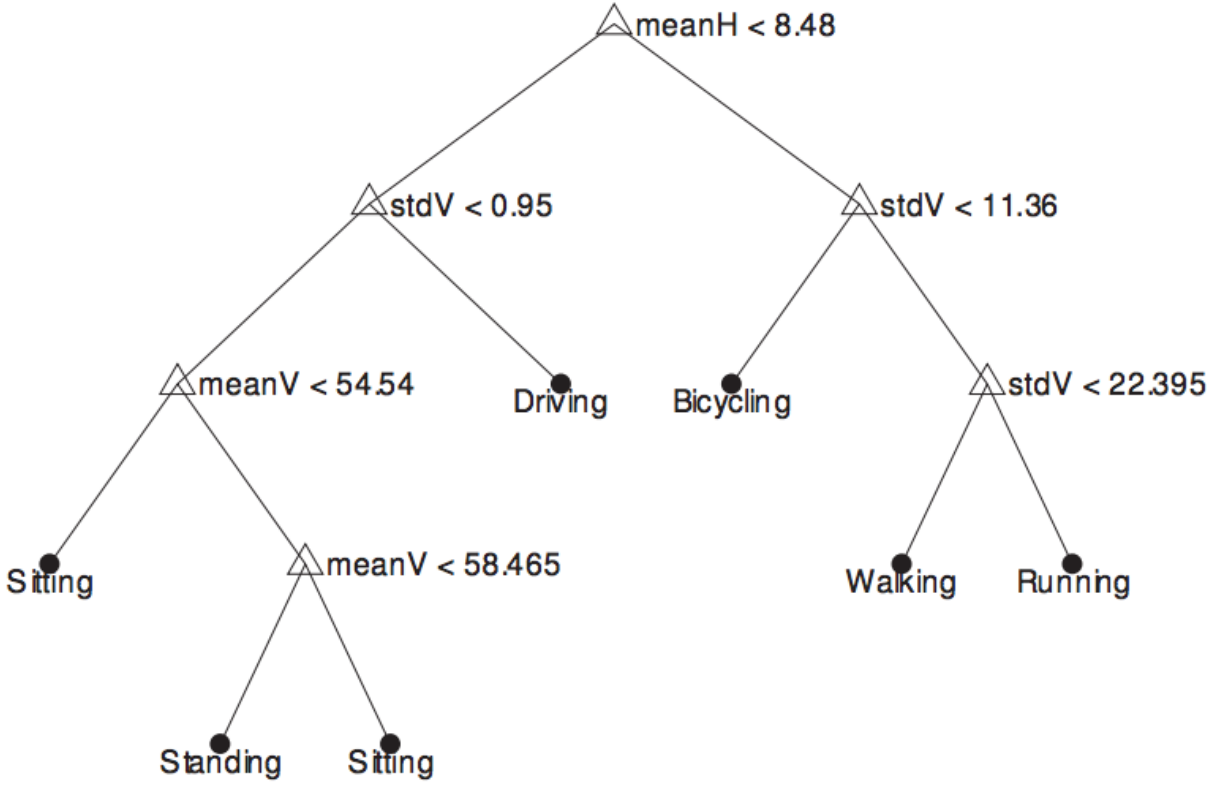
Step 5: Train classifier

- Features are just numbers
- Different values for different activities
- **Training classifier:** figures out feature values corresponding to each activity
- Weka, MATLAB already programmed with different classification algorithms (SVM, Naïve Bayes, Random Forest, J48, logistic regression, SMO, etc)
- Try different ones, compare accuracy
- SVM example

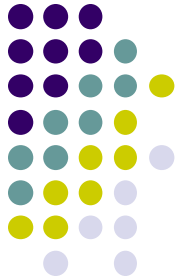


Step 5: Train classifier

- **Example:** Decision Tree Classifier
- Feature values compared against learned thresholds at each node



Step 5: Train classifier
Compare Accuracy of Classifier Algorithms



- Weka, MATLAB also reports accuracy of each classifier type

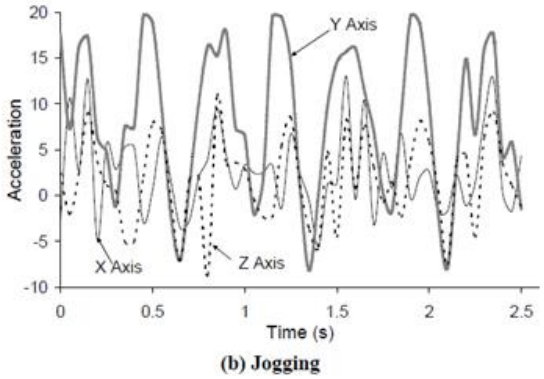
Table 2: Accuracies of Activity Recognition

	% of Records Correctly Predicted			
	J48	**Logistic Regression**	**Multilayer Perceptron**	**Straw Man**
Walking	89.9	**93.6**	91.7	37.2
Jogging	96.5	98.0	**98.3**	29.2
Upstairs	59.3	27.5	**61.5**	12.2
Downstairs	**55.5**	12.3	44.3	10.0
Sitting	**95.7**	92.2	95.0	6.4
Standing	**93.3**	87.0	91.9	5.0
Overall	85.1	78.1	**91.7**	37.2

Compare, pick most accurate classification algorithm

Step 6: Export Classification model as JAR file
Step 7: Import into Android app

- Export classification model (most accurate classifier type + data threshold values) as Java JAR file
- Import JAR file into Android app
- In app write Android code to
  - Gather accelerometer data, segment, extract feature, classify using classifier in JAR file
- Classifies new accelerometer patterns while user is performing activity => Guess (infer) what activity

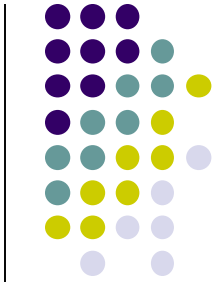


(b) Jogging

New accelerometer Sample in real time

Classifier in Android app

Activity (e.g. Jogging)



Support Vector Machine (SVM)

Scalable Vector Machines (SVM)

- One of the most popular classification algorithms
- If plot example points with features as axes
- Classification problem: Find boundary between classes
- E.g Classify healthy vs unhealthy patients
- 2 Features are strongest predictors
  - Age
  - Maximum exercise rate

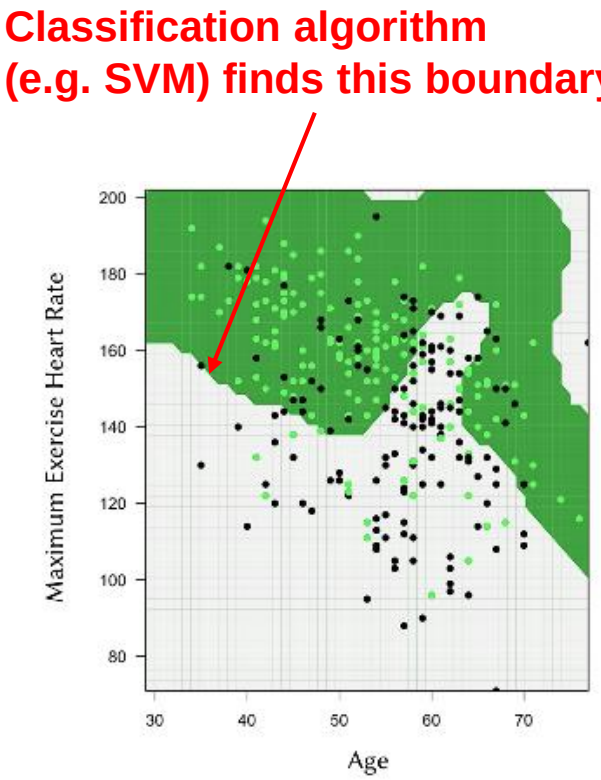


Figure 1. Using SVM to predict the presence of heart disease. The dark green region represents the profile of healthy adults, while the gray region represents the profile of heart disease patients. The light green and black points represent healthy adults and heart disease patients respectively.

SVM: Delineating Boundaries

- Multiple ways to delineate optimal boundary

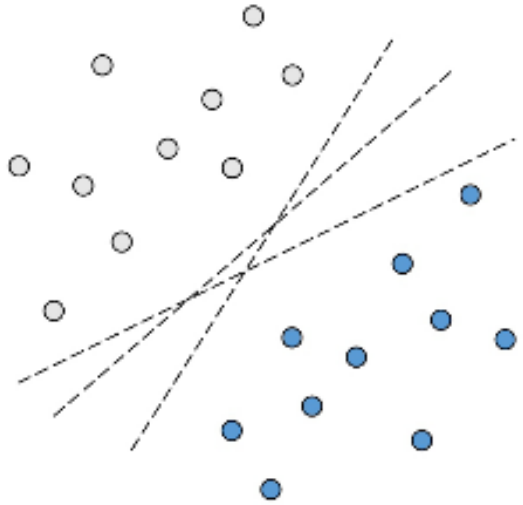


Figure 2. Multiple ways to separate two groups.

SVM: Support Vectors

- SVM first finds peripheral data points in group 1 that are closest to the points in group 2 (called **support vectors**)
- Then draw **optimal boundary** between support vectors of both groups
- Since SVM uses only relatively few data points (support vectors), it is computationally efficient

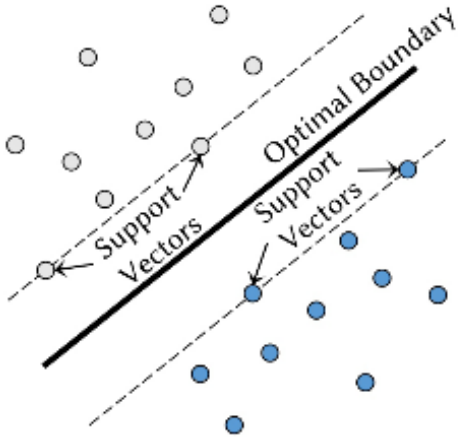
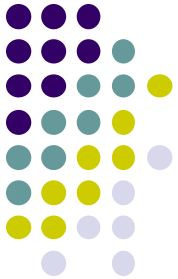


Figure 3. Optimal boundary is located in the middle of peripheral data points from opposing groups.

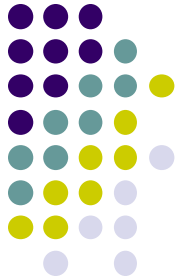
SVM Limitations



- **Inaccurate for small datasets:** Smaller dataset would have fewer points, less likely to find good support vectors

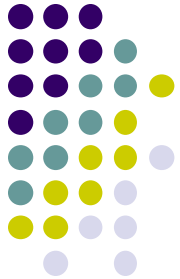
- **Classifying multiple groups:**
  - SVM classifies 2 groups at a time.
  - Multiple groups handled by making multiple 2-group classifications
  - Multi-group SVM: On each iteration, classify 1 group from the rest

- **Overlapping groups:**
  - Since SVM classifies points based on what side of boundary it lies, overlapping groups present a challenge
  - If classes overlap, points close to boundary may be mis-classified



Context Sensing

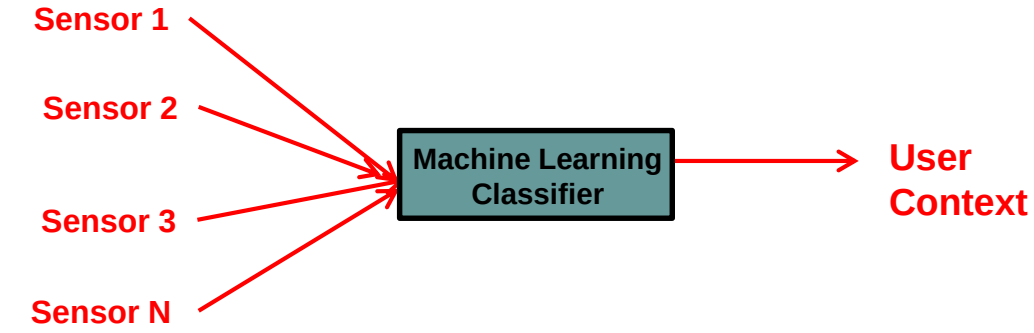
Recall: Ubicomp Senses User's Context

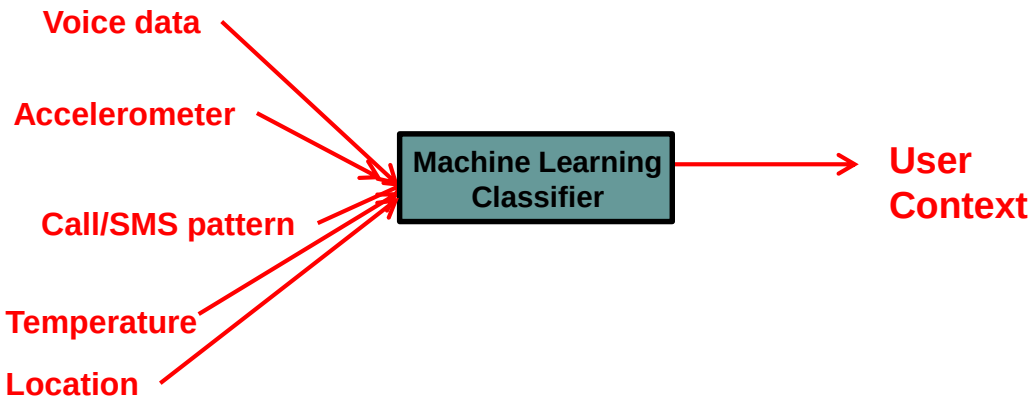


- Context?
  - *Human:* motion, mood, identity, gesture
  - *Environment:* temperature, sound, humidity, location
  - *Computing Resources:* Hard disk space, memory, bandwidth
  - *Ubicomp example:*
    - *Assistant senses:* Temperature outside is 10F (environment sensing) + Human plans to go work (schedule)
    - *Ubicomp assistant advises:* Dress warm!
- Sensed **environment + Human + Computer resources** = ***Context***
- *Context-Aware* applications adapt their behavior to context

Context Sensing

- Activity Recognition uses data from only accelerometer (1 sensor)
- Can combine multiple sensors, use machine learning to learn **user context** that occur to various outcomes (e.g. user's emotion)
- More later





References

- Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore, Activity recognition using cell phone accelerometers, SIGKDD Explor. Newsl. 12, 2 (March 2011), 74-82.
- Deepak Ganesan, Activity Recognition, Physiological Sensing Class, UMASS Amherst