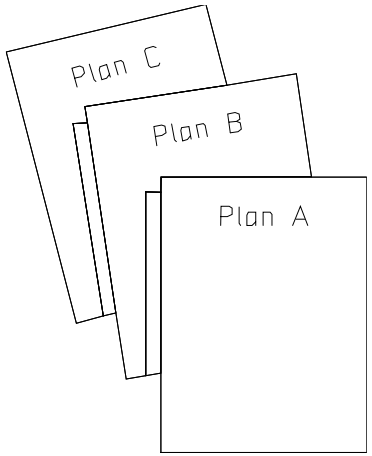
Human Behavior and Context Sensing

It takes many good deeds to build a good reputation, and only one bad one to lose it.

Benjamin Franklin

Overview

- Objective
  - To understand exemplary techniques and challenges for activity sensing and recognition



- Content
  - Overview on human behavior and context sensing
  - Physical activity sensing and recognition

- After this module, you should be able to
  - Understand human behavior and context sensing
  - Understand basics of activity recognition

Overview on Human Behavior and Context Sensing

Life-Immersive Mobile Computing

Sense **real-world** situations and human behavior

↓

Extract and infer useful insights and Knowledge

↓

Provide what people need right on time & place

Pothole Monitoring

Sleep Quality Monitoring

Location-aware Alarms

Physical Activity Diary

Bus Stop Queue Estimation

Proactive Advertisement

Life-Immersive Mobile Computing

- The first step toward realizing Mark Weser's vision for ubiquitous computing

The Computer
for the 21st Century

Mark Weiser

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Key Building Block:
Behavioral Sensing and Analytics

Research Trends

Comprehensive/ detailed behavior

- ✓ Centimeter-level indoor localization

- ✓ Eating
- ✓ Smoking
- ✓ Shopping
- ✓ Dancing
- ✓ Drumming

- ✓ Turn-takings
- ✓ Linguistic contents
- ✓ Emotional expressions

External Behavior

Location

Physical Activity

Conversation

Internal States

- ✓ Heartrate
- ✓ Stress
- ✓ Mood
- ✓ Sleep quality
- ✓ Distractibility
- ✓ Intention
- ✓ Engagement
- ✓ Attention
- ✓ Mindfulness
- ✓ Emotion
- ✓ Anxiety
- ✓ Depression
- ✓ Boredom
- ✓ Fatigue
- ✓ …

Smartphone Sensors

Sensor-rich Environments

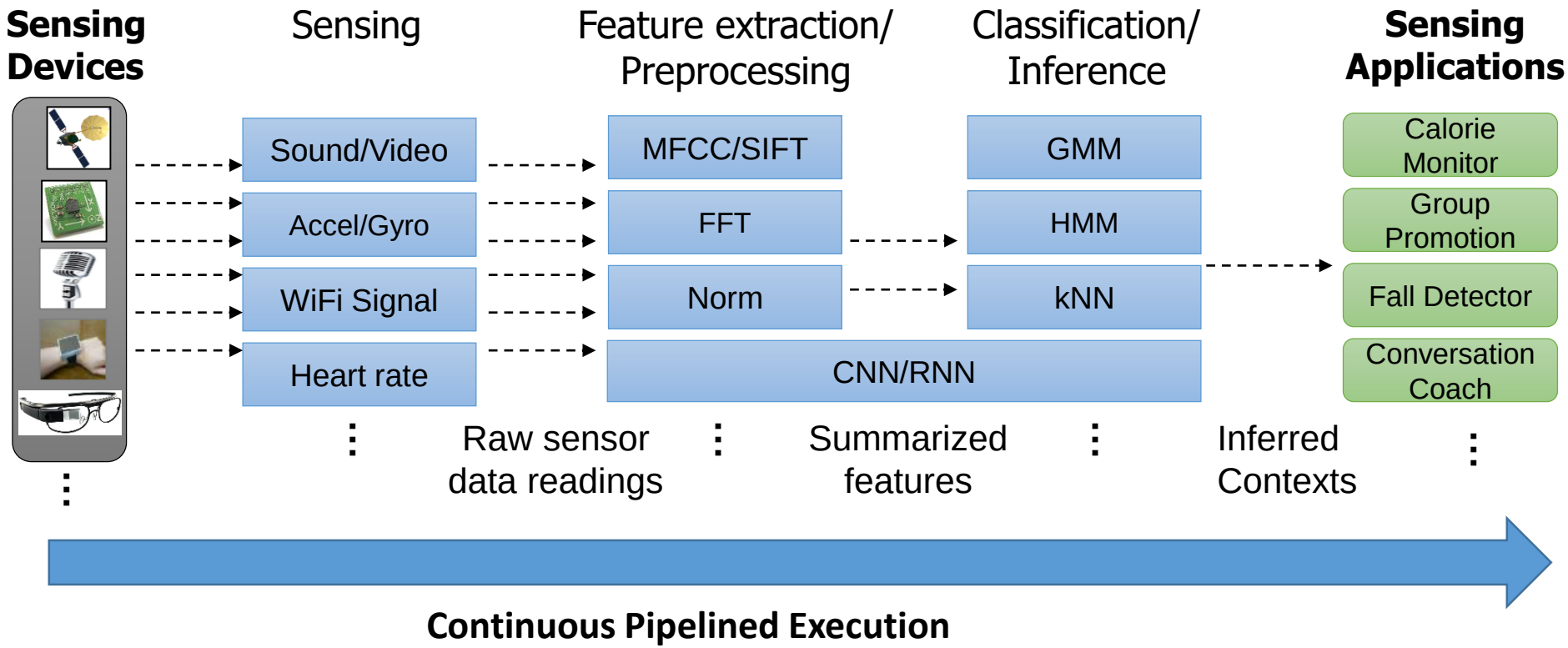
Wearable sensors

Phone-embedded sensors

Space/object-embedded Sensors and other users

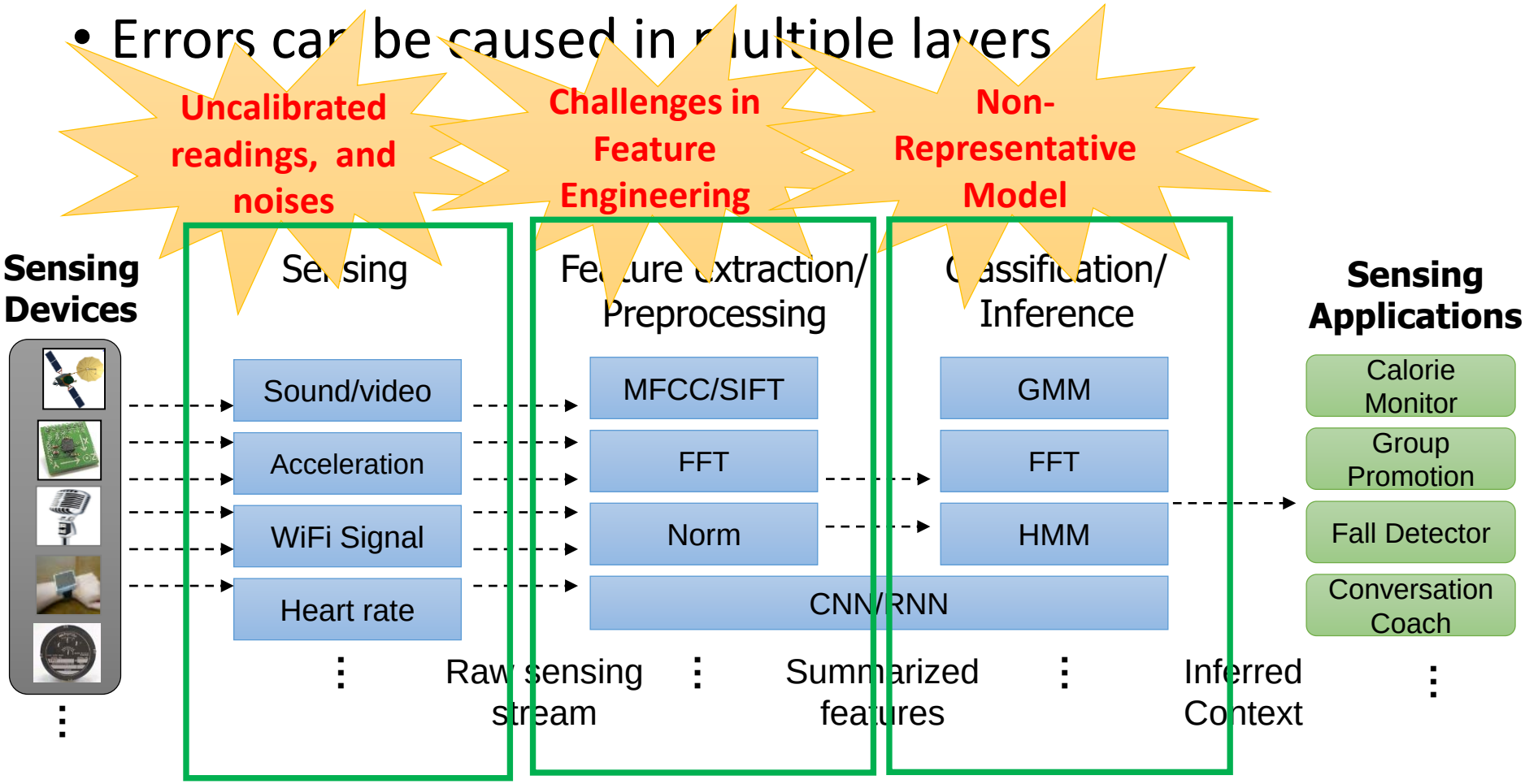
Common Computational Flow

Continuous sensing and analytics of user activities, location, emotions, and surroundings with mobile/IoT/wearable devices



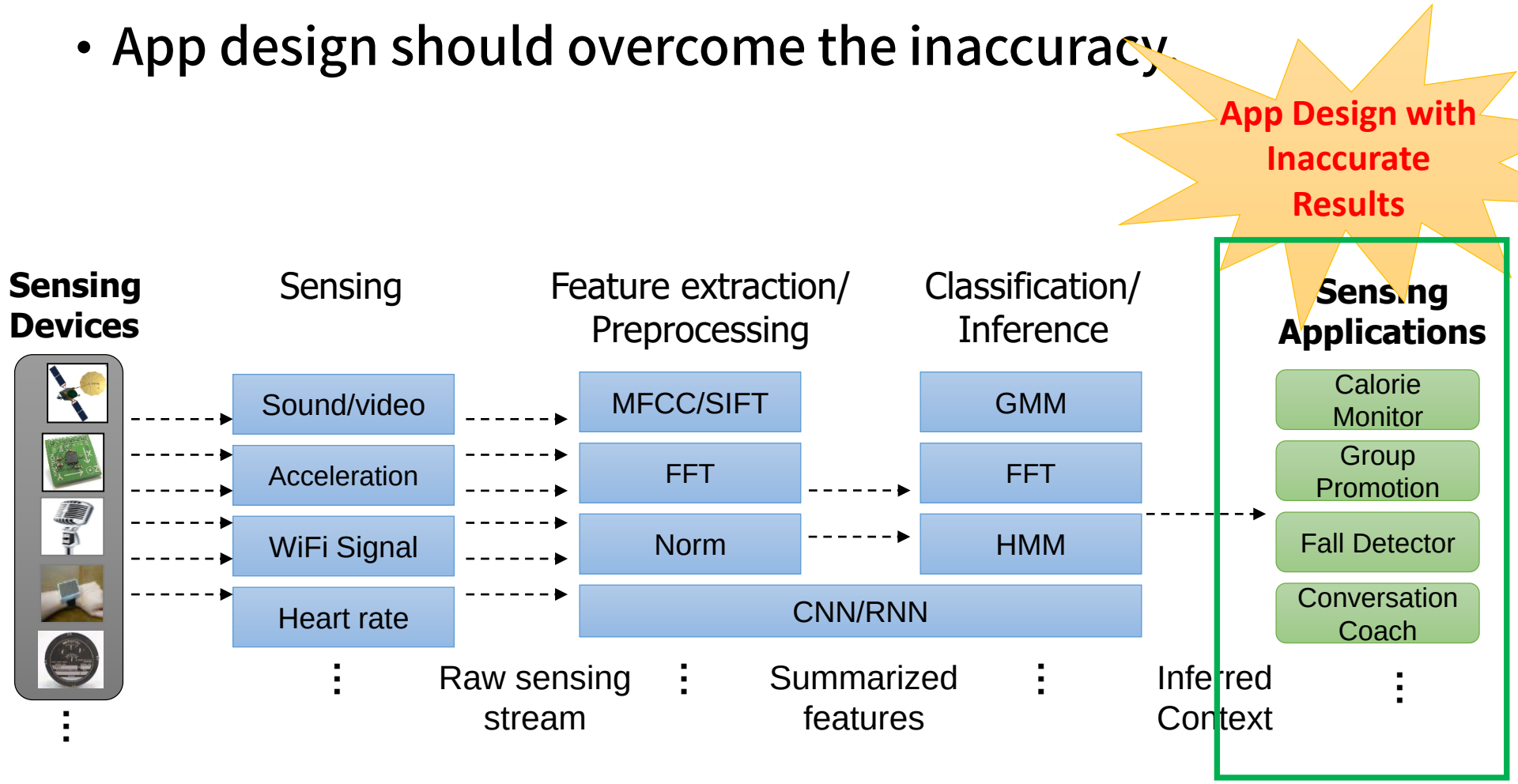
Challenge 1: Inference Accuracy

- > 90% accuracy is extremely challenging.
- Errors can be caused in multiple layers

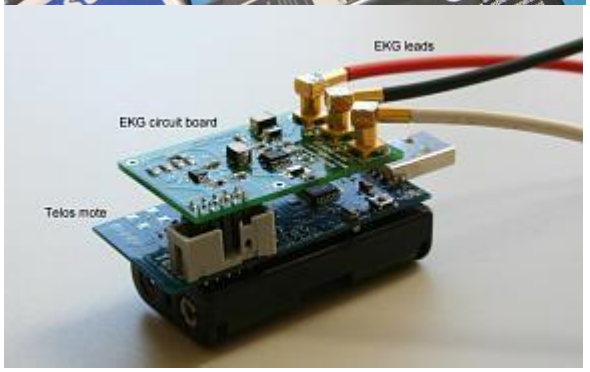


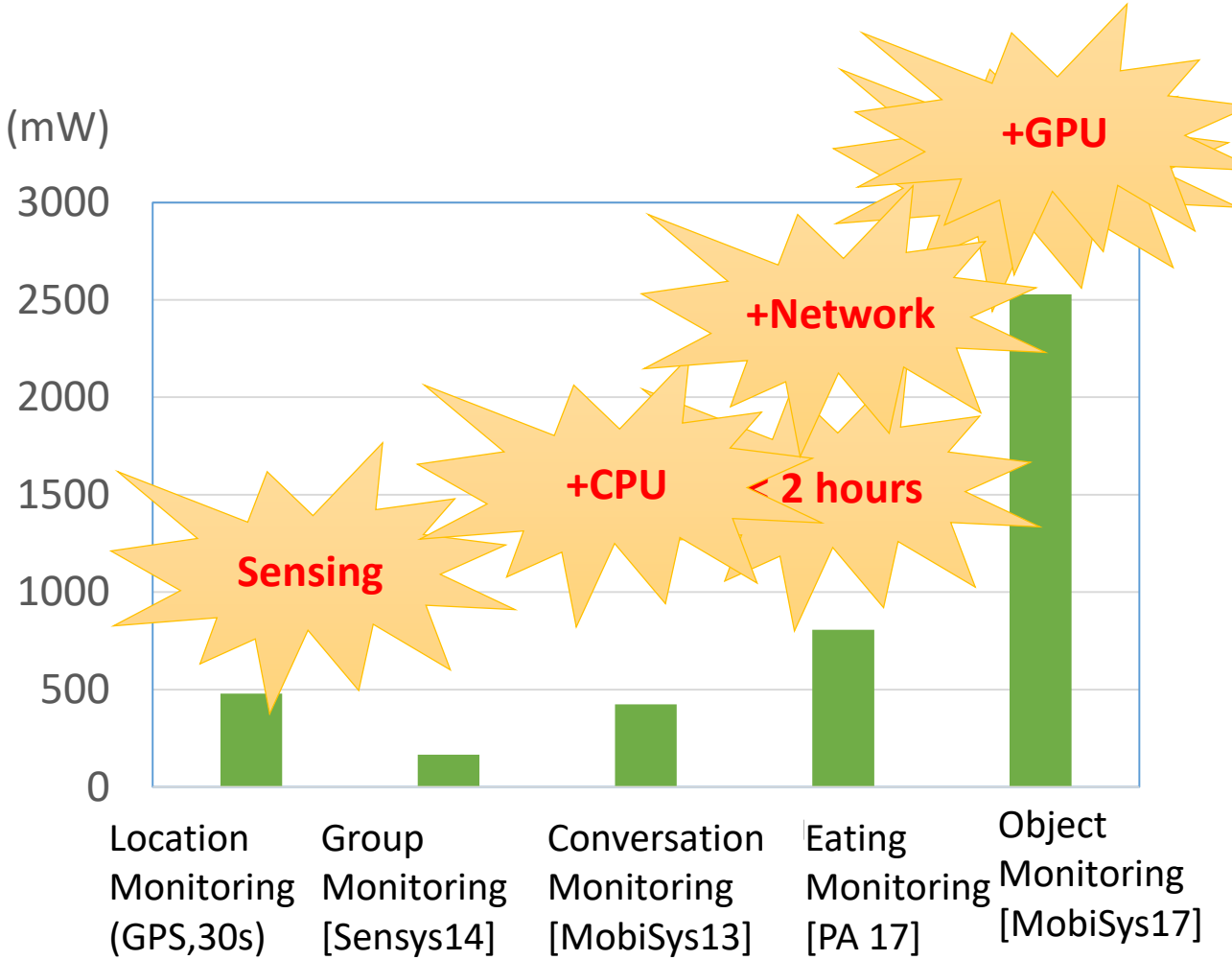
Challenge 2: Application Usability

- **The inference results are not 100% correct.**
- **App design should overcome the inaccuracy.**



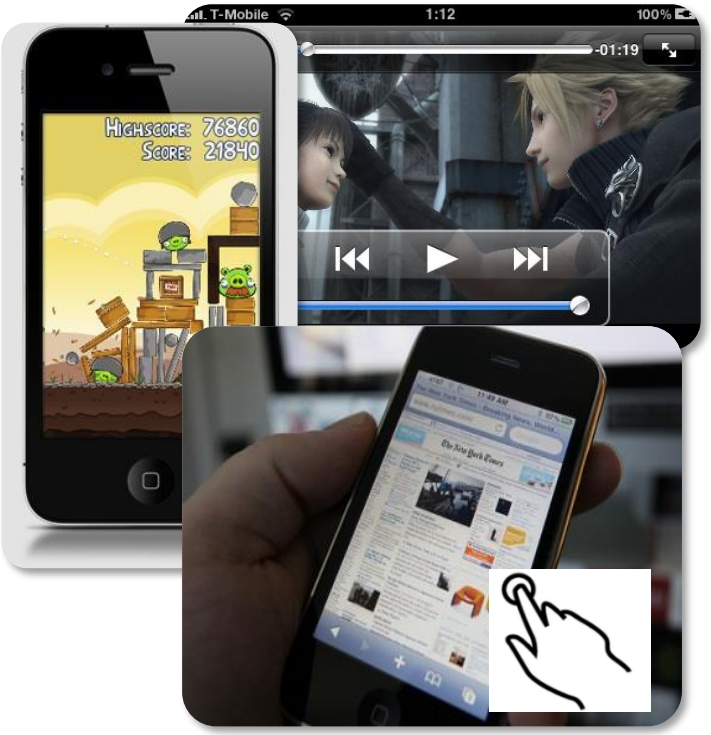
Challenge 3: Power Scarcity





- Measured with Samsung Note 4 (3220mAh battery)
- Used Samsung Gear (315 mAh battery) for Anapruna (eating detection)

Challenge 4: New Operational Mode



Vs.

Small display, user mobility
→ **A single user-interactive application**

Mobile sensing : *autonomous, situation-aware* services
→ **Concurrent background sensing applications**

Challenge 5: Resource Contention

- Continuous vision sensing (with VGG-16 Deep Neural Network Model) and a foreground application concurrently.
- Measured frame rates on Samsung Galaxy S7 (Mali T880 GPU)

Foreground Application	Standalone (w/o Vision Sensing)	Contention (with Vision Sensing)
Video Player	30 FPS	22.14 FPS (Avg) (15 Min.)
Flappy Bird	60 FPS	28.54 FPS (Avg) (13 FPS Min.)

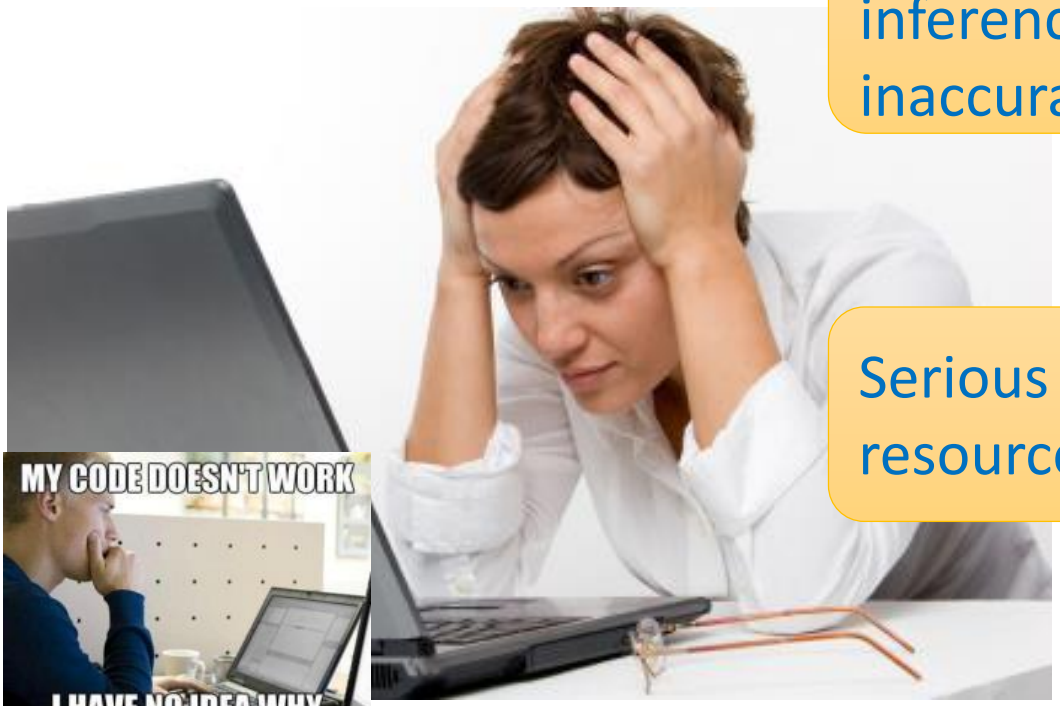
Challenge 6: Poor Scalability

Amazing mobile s
How to test with

Lets test it with lab users and a small number of real users and consider it “real-world”.

Wow! It does not work!
Need access to real venues
With real users on real devices
HOW???

Individual Applications Solve All These?



Complexity to implement **accurate** inference logics and handling inaccuracy in app design

+

Serious **optimization** in battery and resource usage

+

Scalable deployment and testing with a large pool of real users in real-life situations

Behavior and Context Sensing

- Active on-going research area
- Lots of inter-disciplinary challenges across computer systems, human computer interaction, artificial intelligence, and many other domains.
- Huge opportunities for both research and business

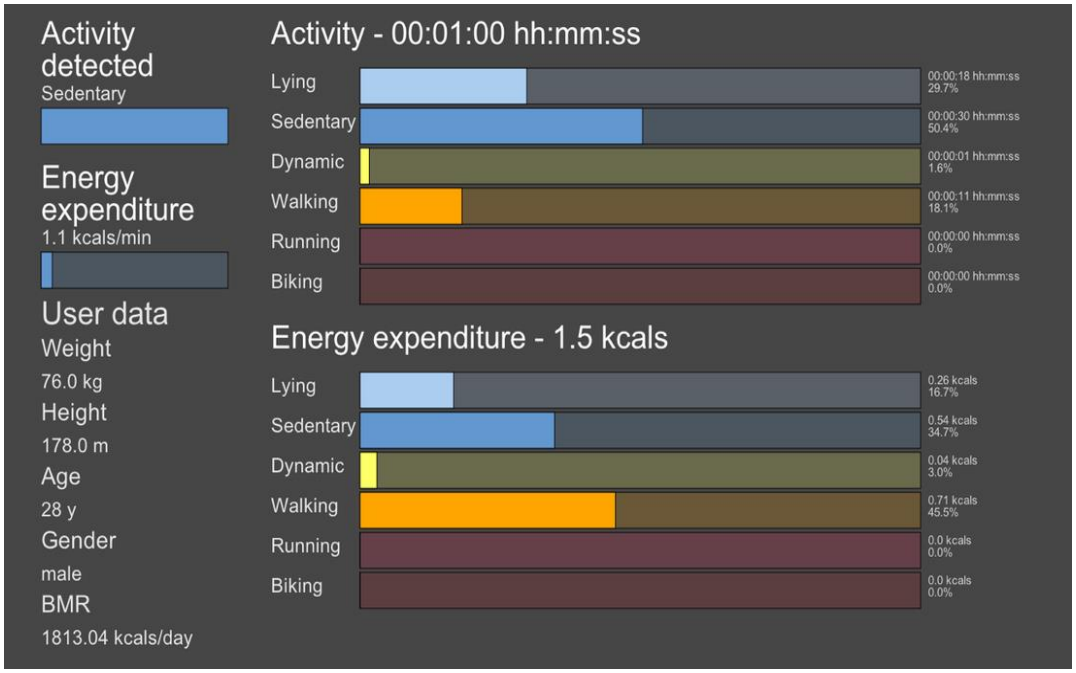
Overview on Activity Sensing and Recognition

Mobile Activity Tracker

- Everyday exercise progress monitor and motivator
- Provide reliable feedback about how much they move. (People often overestimate!)
- Provide instant and constant feedback about activity levels.
- Gamify to encourage individuals to compete in getting fit and losing weight.

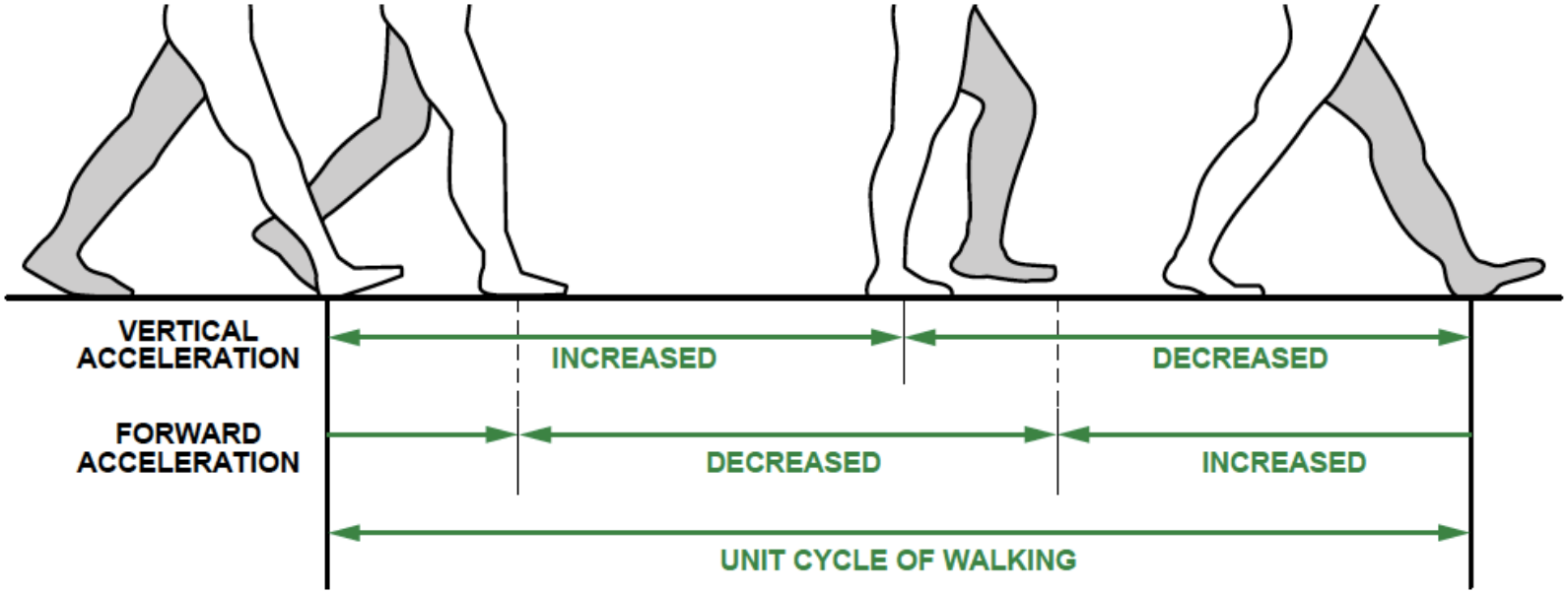
Quantified Self

- Continuously track user activities and objectively summarize and visualize data
- Elicit users behavioral changes in a positive way



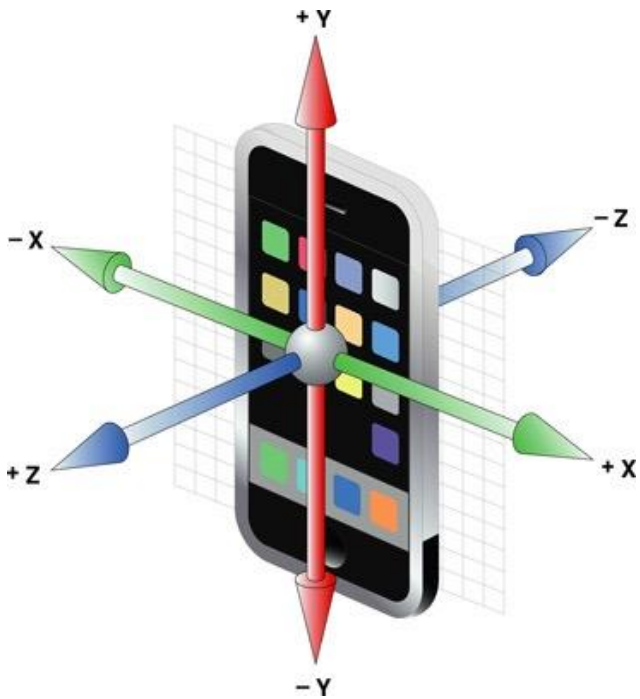
How Do we Monitor Activities?

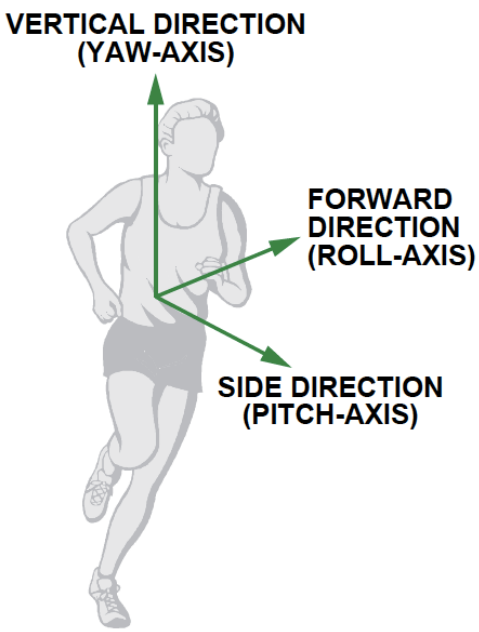
- Activities involve physical movement of body limbs.



Inertial Sensors: Accelerometer

- All commodity smartphones have accelerometers.
- Measure linear acceleration (m/s^2) in three different directions.

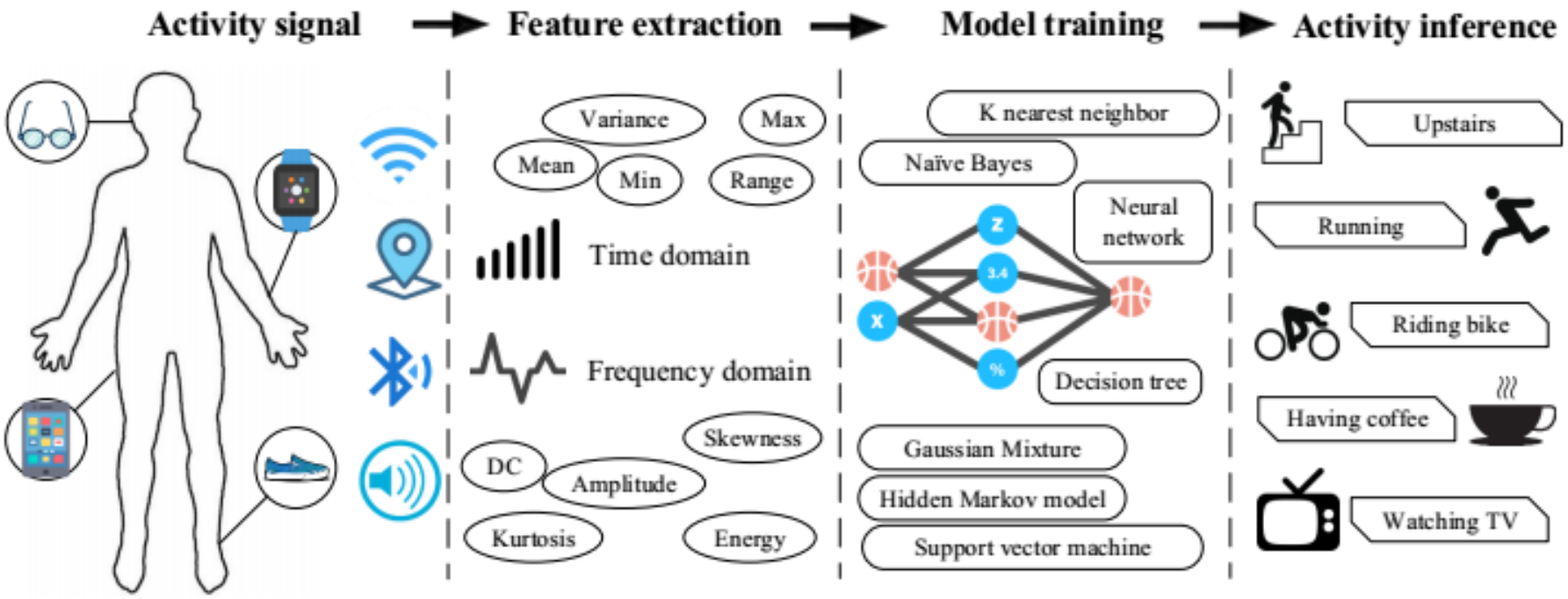




Inertial Sensors: Gyroscope and Compass

- Gyroscope: measure orientation and angular velocity
- Compass: measure the direction on the earth's surface toward the north.

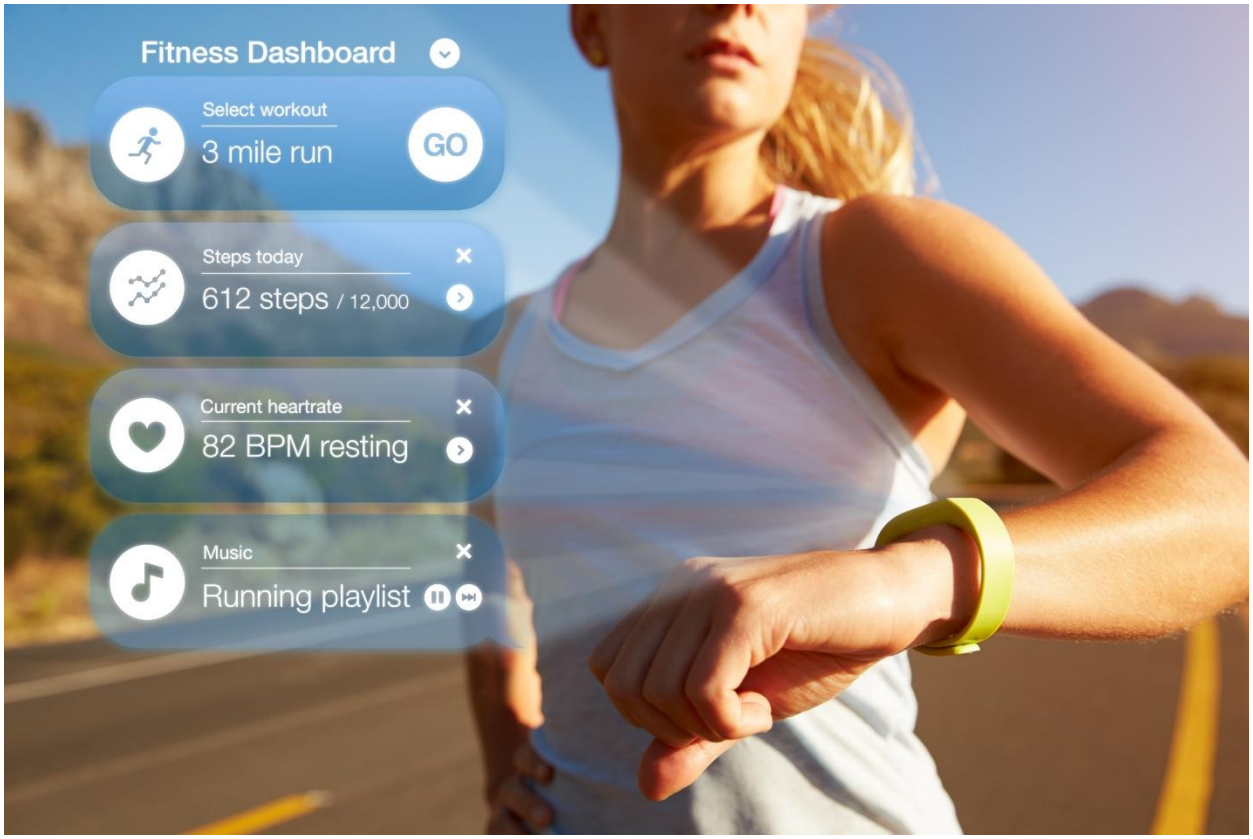
Activity Monitoring Overview



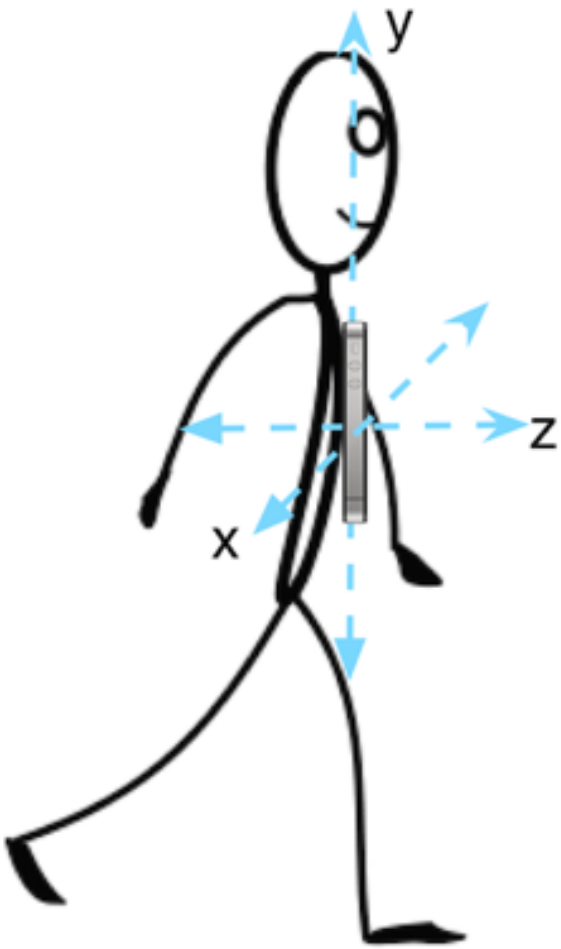
Example 1: Pedometer

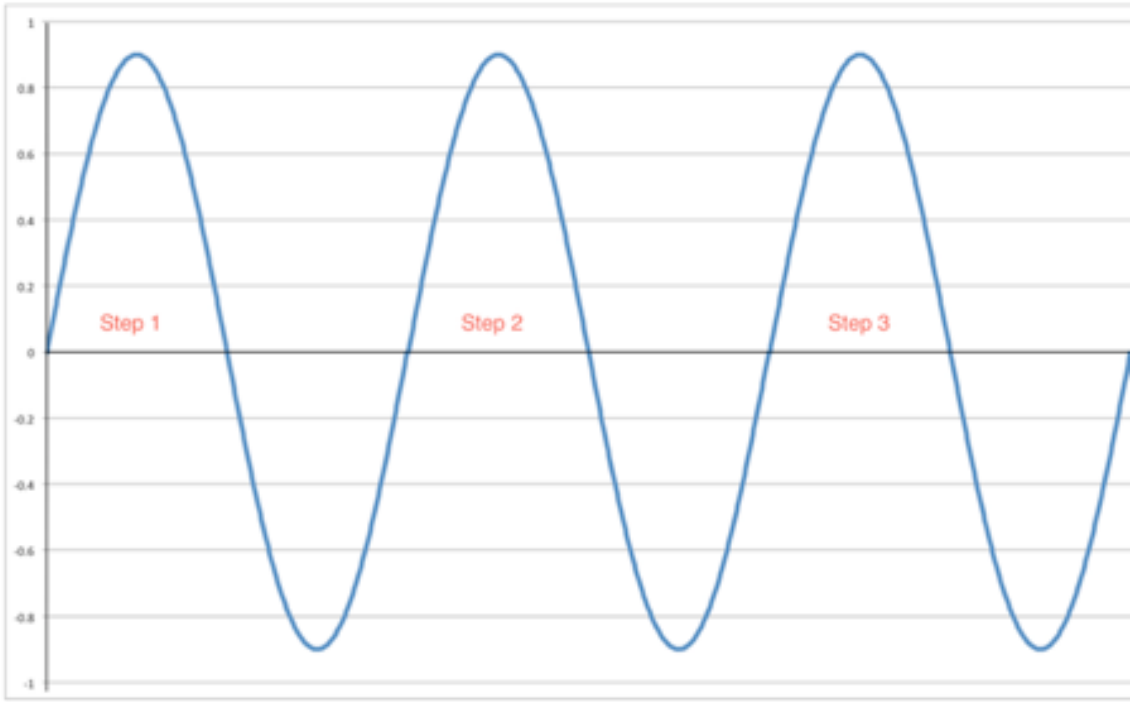
Pedometer

- Detect and Count Steps

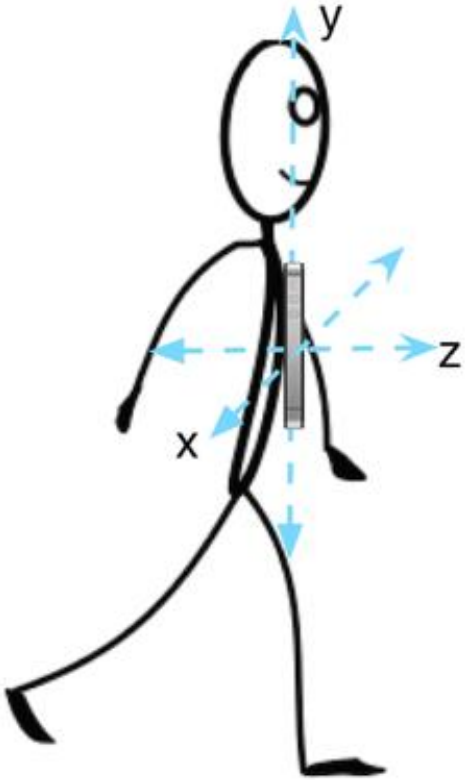


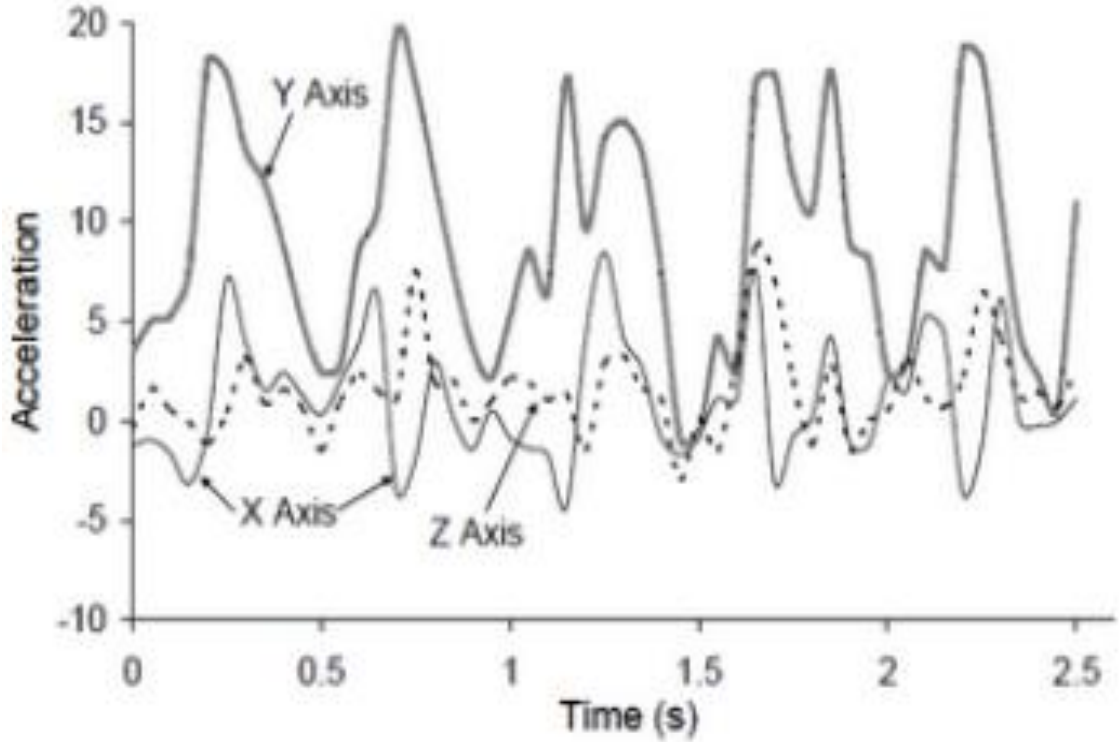
Ideal Acceleration Signal of Walking





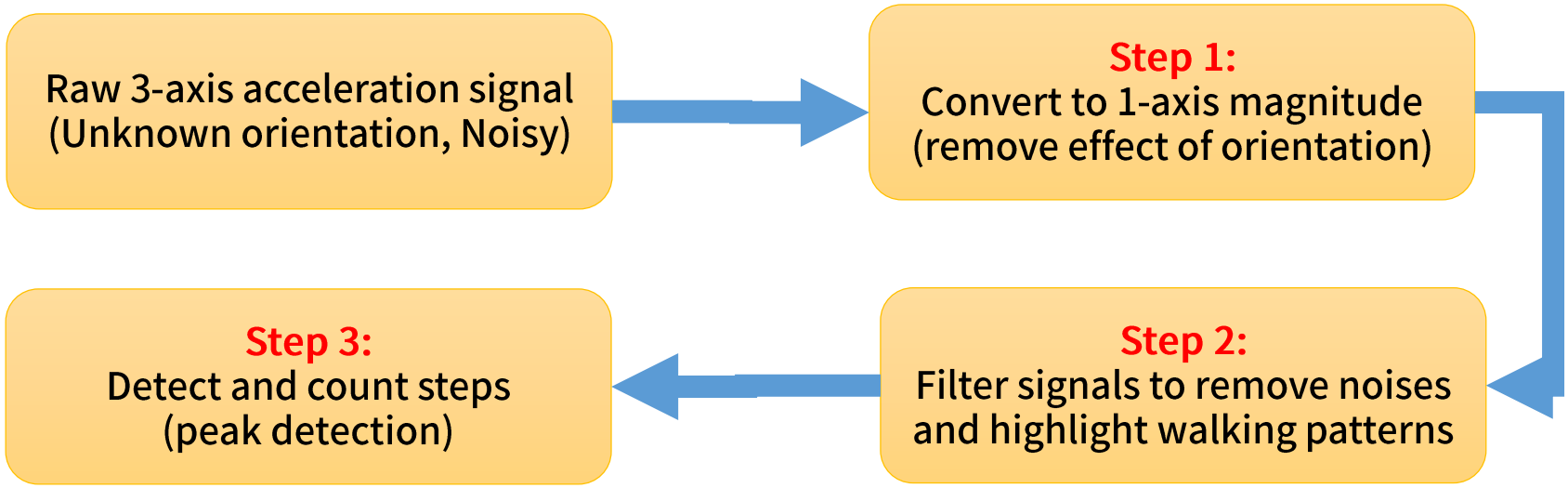
Realistic Signal with Noise





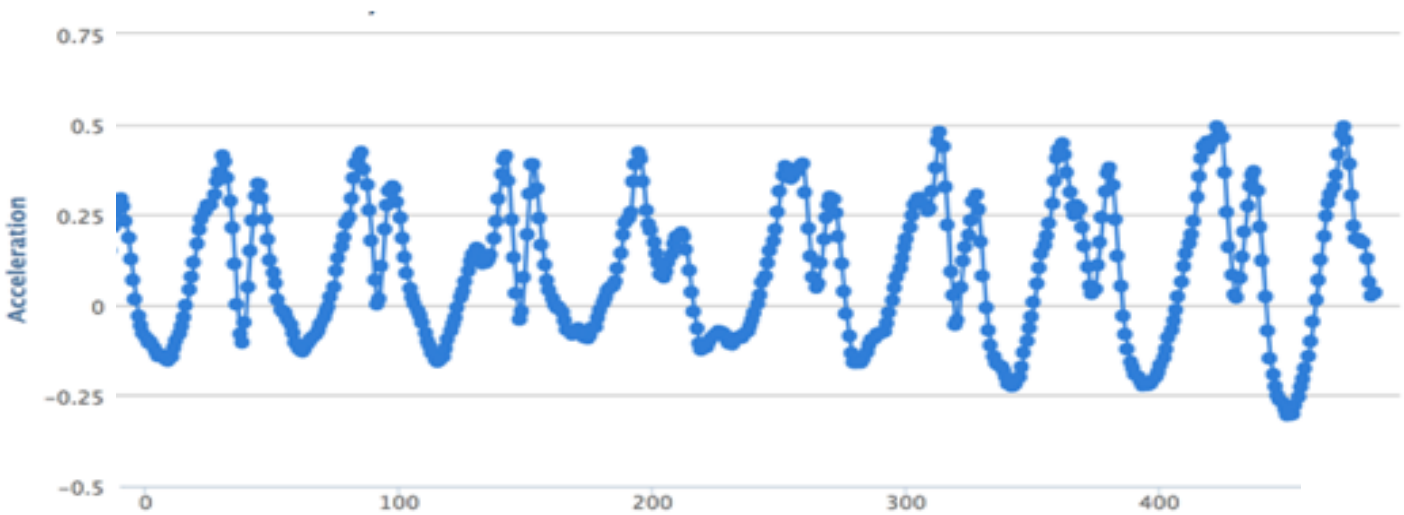
How to Detect Steps?

- There are many ways that we can design a step detection algorithm.
- Initial idea: Convert the 3-axis signal into the 1-axis magnitude signal, and then extract steps from this signal.



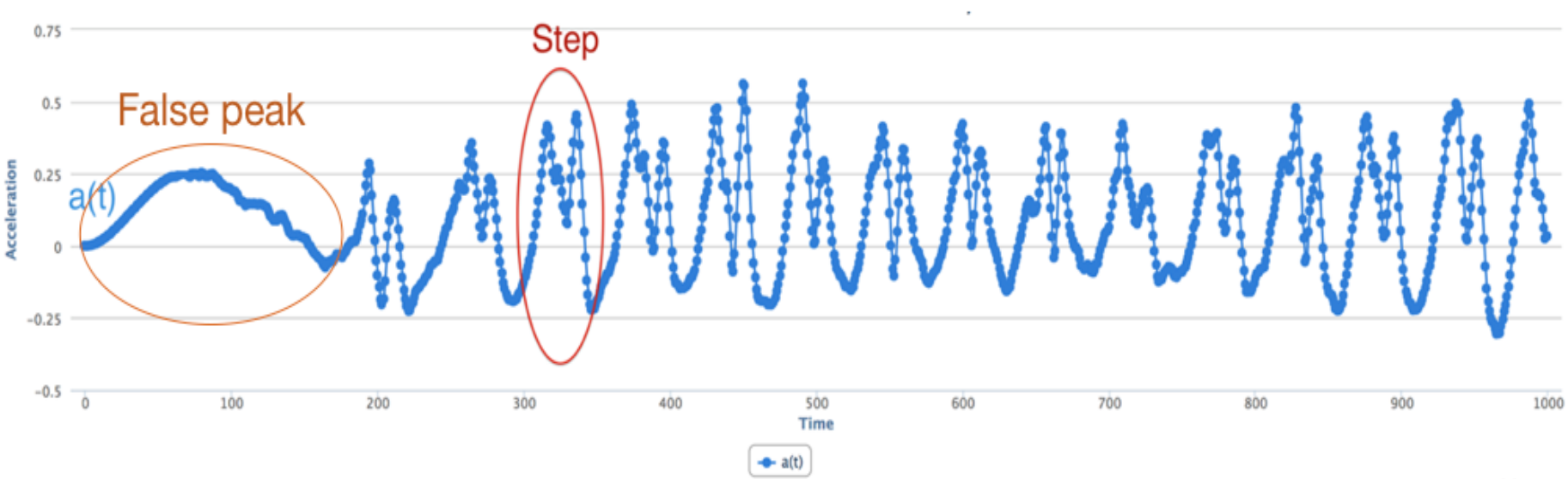
Step 1: Extract Signal Magnitude

- Calculate the maximum acceleration in one direction.
- Take the magnitude of the entire acceleration vector i.e. $\sqrt{(x^2+y^2+z^2)}$, where x, y, and z are the 3-axis accelerometer readings.
- We will see the signals as below.



Step 2: Noise Removal

- Remove noise to extract the specific signal corresponding to walking.
- Various noises:
  - Jumpy peaks: Phone jiggles. Some users have a bounce.
  - Short peaks: Users may use a phone.
  - Slow peaks: Users may move legs while sitting.



Noise Removal Techniques

- Sensor signals often include various noises.
- Need to apply various pre-processing techniques to remove noises and highlight the signals we want to capture
- Common pre-processing techniques
  - Moving average smoothing
  - Exponential smoothing
  - Median filtering
  - Frequency domain filtering

Moving Average Smoothing

- Use average values of multiple adjacent samples.
- Example: Averaging the values for 3 samples
  - Input: $x = x_1, x_2, x_3, \ldots x_n$ where the index is the sample number.
  - The output of the moving average filter, $s = s_1, s_2, s_3, \ldots s_n$,is:
    - $s_1 = (x_1 + x_2 + x_3)/3$
    - $s_2 = (x_2 + x_3 + x_4)/3$
    - $s_3 = (x_3 + x_4 + x_5)/3$
    - …
    - $s_{n-2} = (x_{n-2} + x_{n-1} + x_n)/3$
- The smoothing window size can be different.
- The larger the window is, the cleaner the signal becomes.
- Too large window may smooth out the important characteristics of the signal (e.g., steps for step detection).

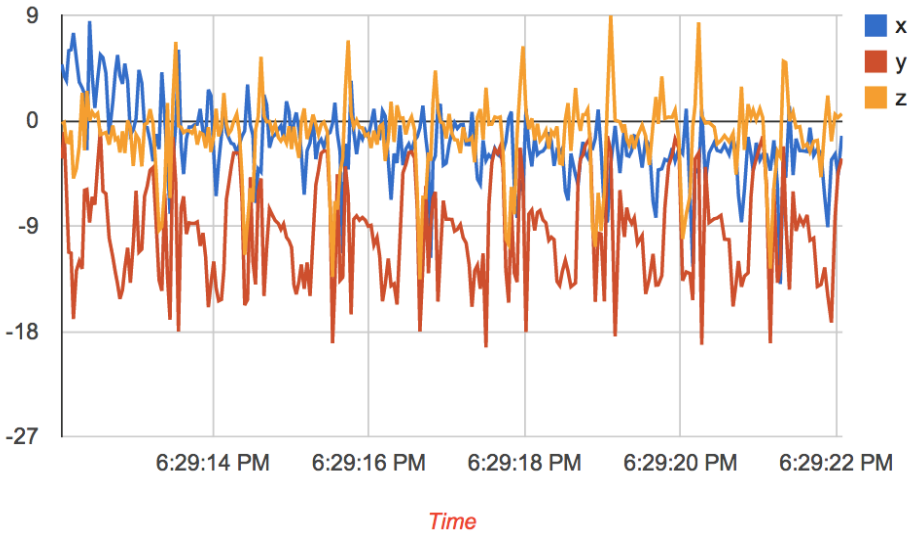
Exponential Smoothing

- What if we want to give more weights to recent values?
- The idea in exponential smoothing is to assign exponentially decreasing weights as the observation get older.

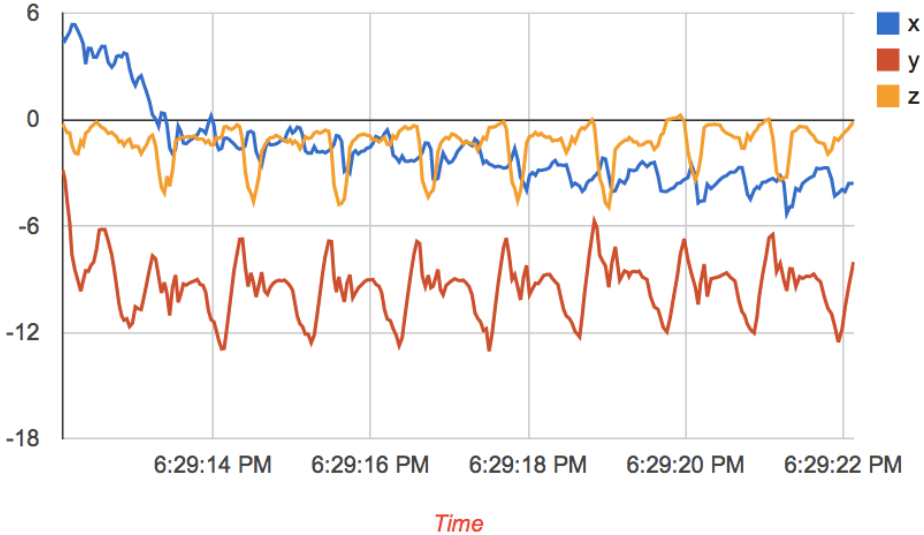
$$s_1 = x_0$$
$$s_t = \alpha x_{t-1} + (1-\alpha)s_{t-1} = s_{t-1} + \alpha(x_{t-1} - s_{t-1}), t > 1$$

- where α is the smoothing factor, and $0 < \alpha < 1$.
- The smoothed output s_t is a simple weighted average of the current observation x_t and the previous smoothed output s_{t-1}.

Effect of Exponential Smoothing



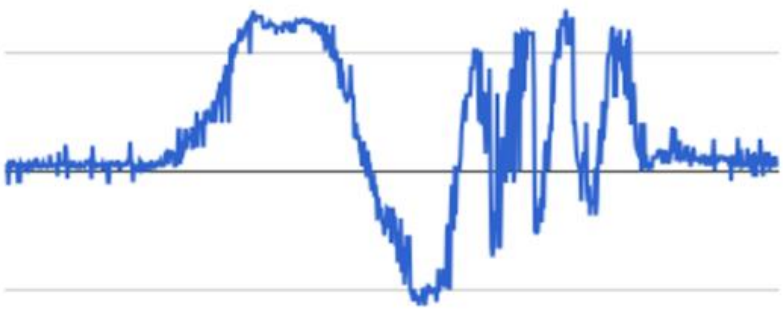
Raw Acceleration Signal



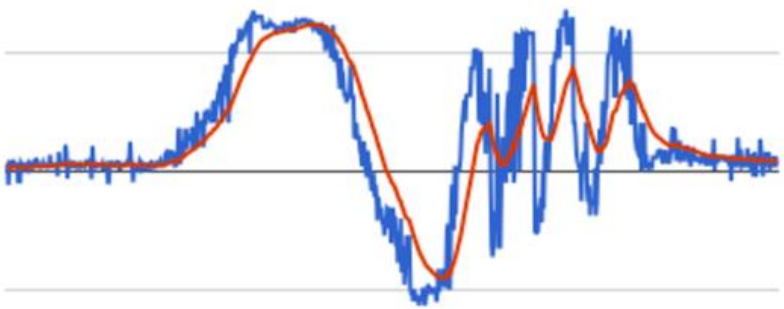
Signal with Exponential Smoothing ($\alpha = 1/6$)

Problem of Exponential Smoothing

- Average out some of the peaks in the data
- Amplitude gets smaller
- Time lag in the peaks, i.e., peaks are slightly sifted to the right.



Raw Acceleration Signal



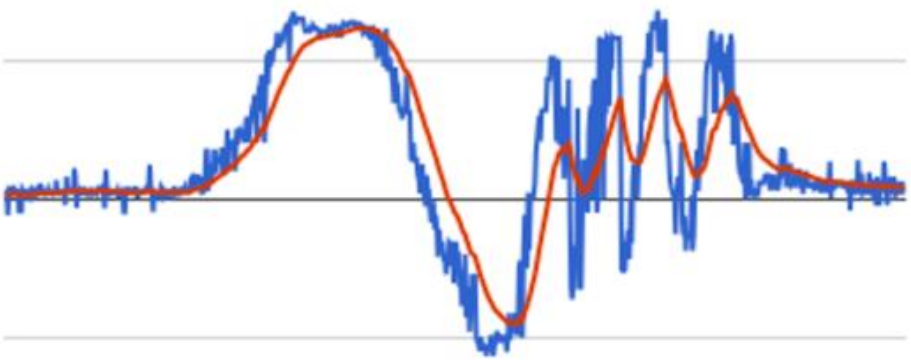
Signal with Exponential Smoothing

A Solution: Median Filtering

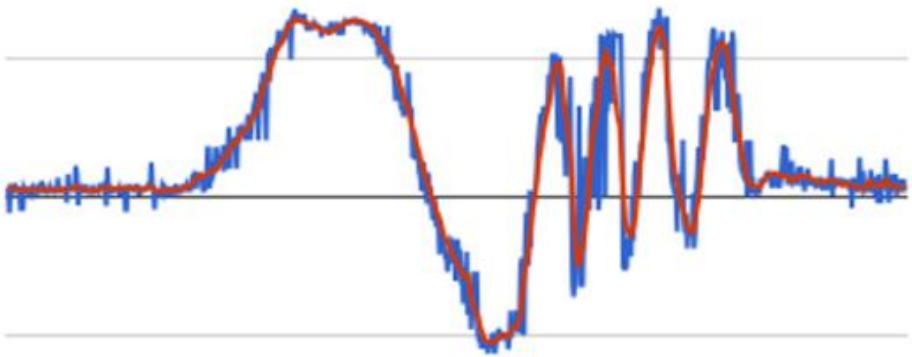
- Given input accelerometer signal: $x = x_1, x_2, x_3, \ldots x_n$,
- The output of the median filter, $s = s_1, s_2, s_3, \ldots s_n$, is:

  - $s_1 = \text{median}(x_1, x_2, x_3)$
  - $s_2 = \text{median}(x_2, x_3, x_4)$
  - $s_3 = \text{median}(x_3, x_4, x_5)$
  - …
  - $s_{n-2} = \text{median}(x_{n-2}, x_{n-1}, x_n)$

Effect of Median Filtering



Exponential Smoothing

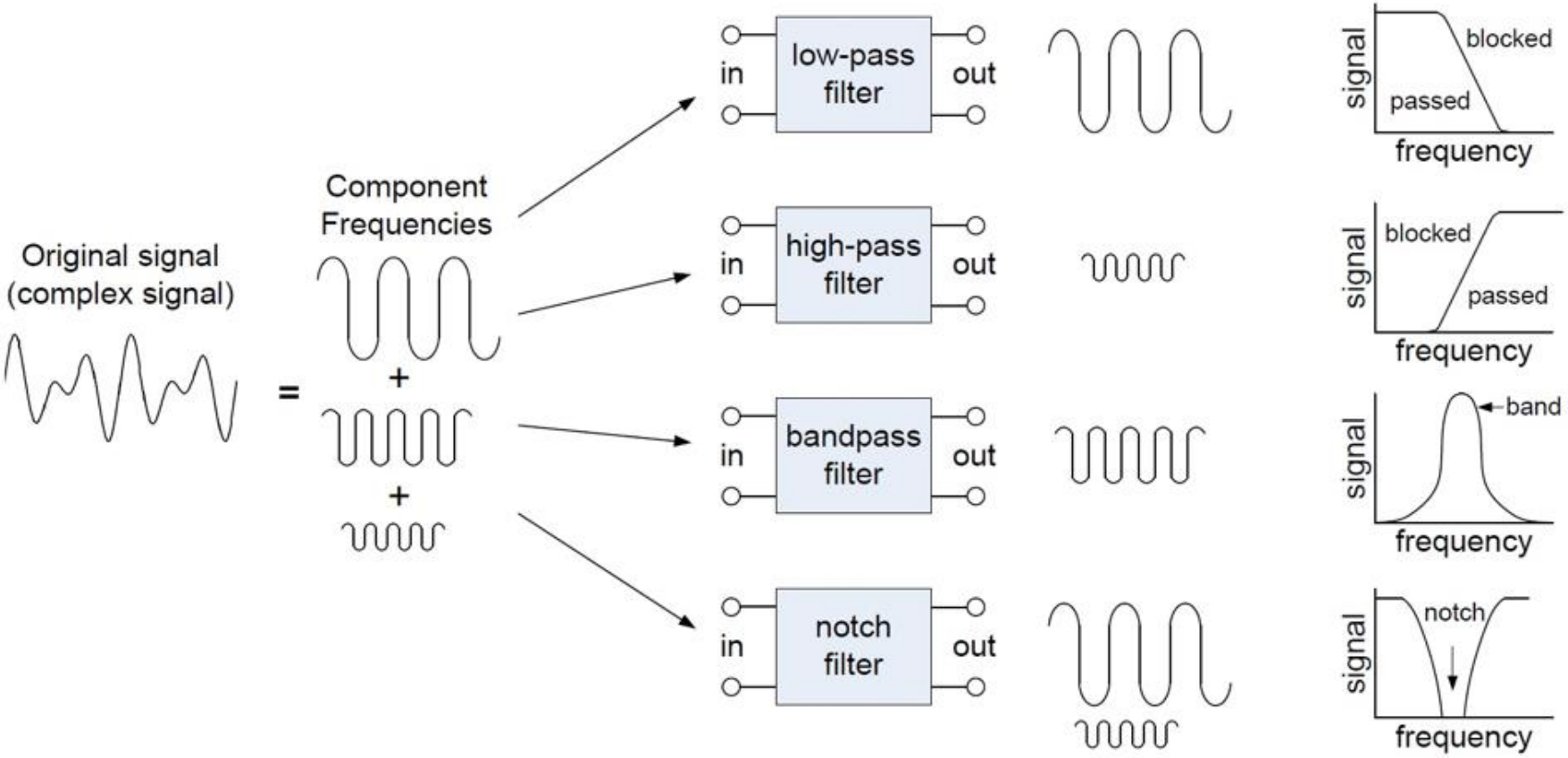


Median Filtering

Frequency Domain Filtering

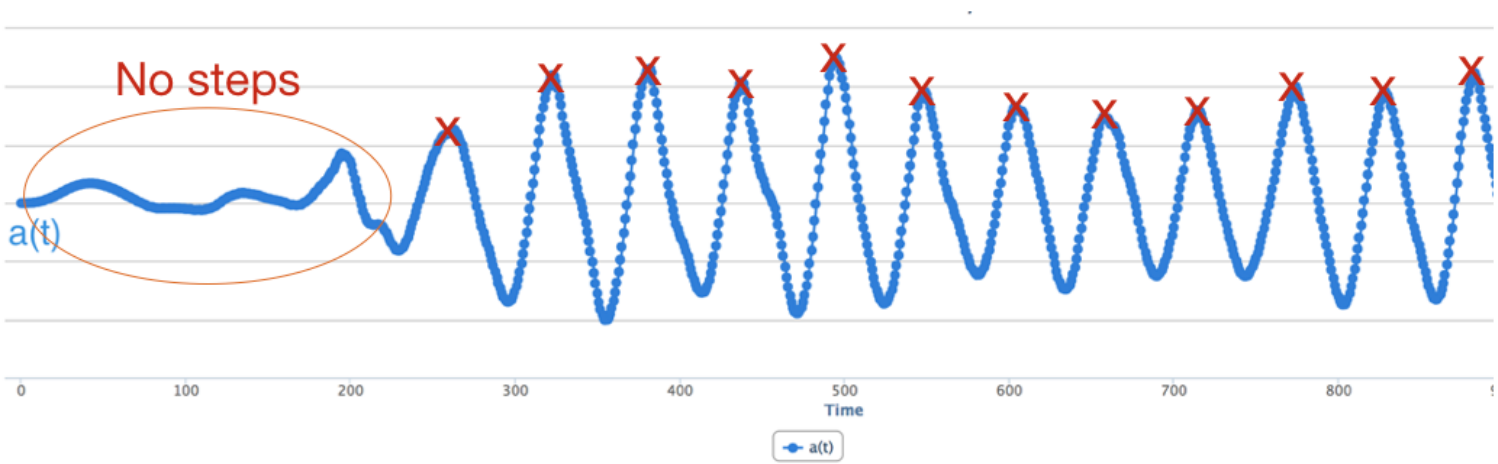
- So far, we have studied the time domain smoothing
  - First method to remove noises
  - Simple and easy to understand
  - Work well in many practical examples
- In some cases, identifying a good time domain-filter is not easy.
- Frequency domain filtering
  - Convert a signal to a weighted sum of sine waves, and remove all the waves whose periods are outside the range that you expect!
  - Jean Baptiste Fourier (1768–1830) proved the mathematical fact that any periodic waveform can be expressed as the sum of an infinite set of sine waves.

Frequency Domain Filters



Step 3: Detecting Steps

- Look for high peaks and use that to detect steps.
- Take the derivative (slope) of the smoothed acceleration signal.
  - The derivative changes from negative to positive when a step occurs.



Filtered acceleration signal and peaks

Calorie Estimation

- Estimating Distance Covered
  - Distance = number of steps × stride
- Estimating Speed: Speed = distance/time.
  - Speed = (steps per y seconds) × stride / y seconds
- Estimating Calories
  - A conventional approximation using speed, distance, and weights.
  - Speed = alpha x speed x distance x weights

Activity Recognition

Activity Recognition

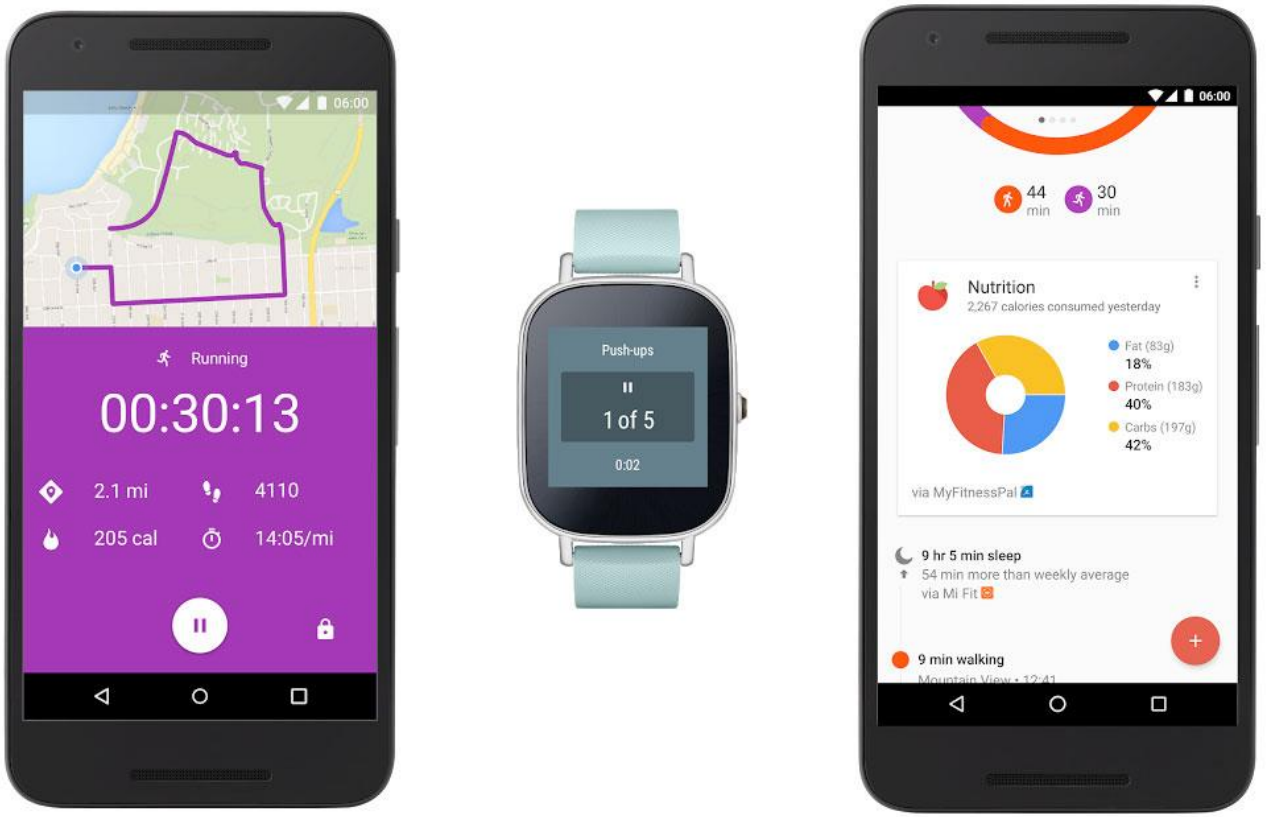
- Identifying the physical activity of a user
  - E.g., jogging, walking, sitting, standing
- Providing useful knowledge about the habits of millions of users passively—just by carrying cell phones.
- Wide range of applications
  - Activity-aware phone configuration (e.g., sending calls directly to voicemail if a user is jogging)
  - Daily/weekly activity profile for daily healthcare.

Detection vs. Classification

- Detection: Single activity type
- Classification: Multiple activity types
- Step detection vs. activity classification
- Classification usually needs a more general approach where it is harder to capture the distinguishing characteristics of each class (walking, sitting, etc.) ahead of time.

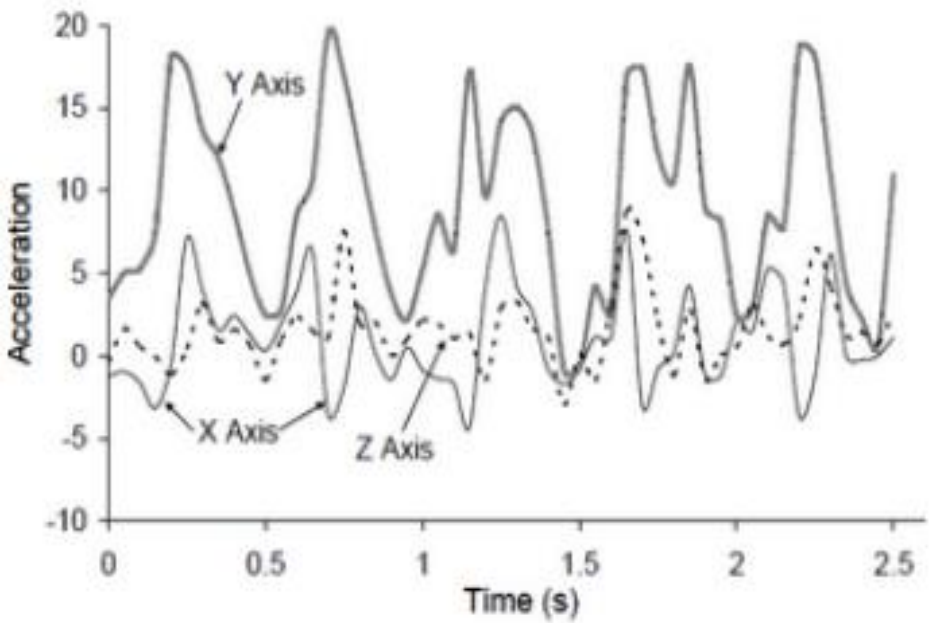
Example: Activity Tracker

- **How do we know if a person is waking or jogging?**

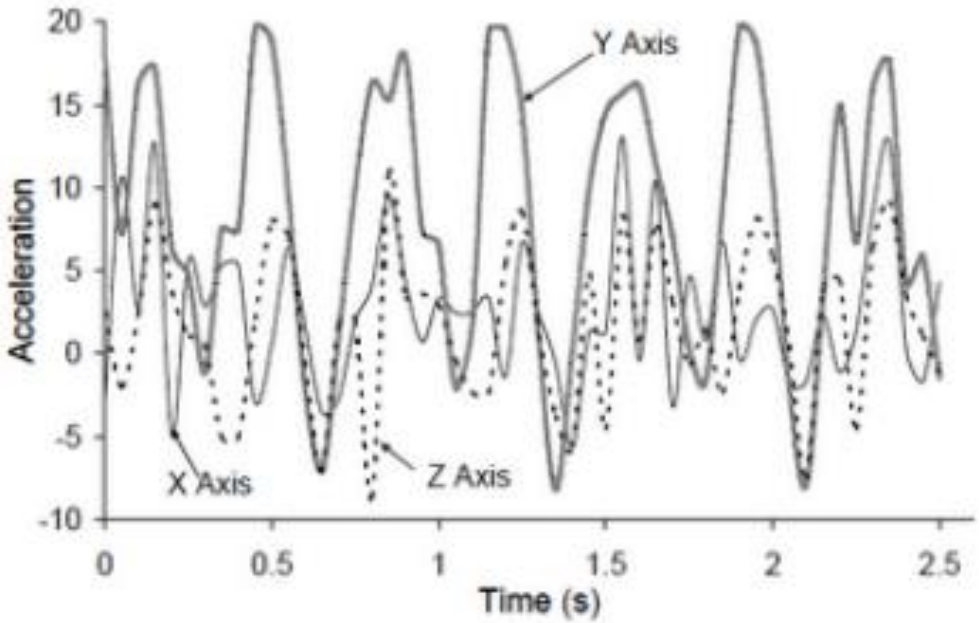


Acceleration Signals

- Assumption: Smartphone is in a front pocket.



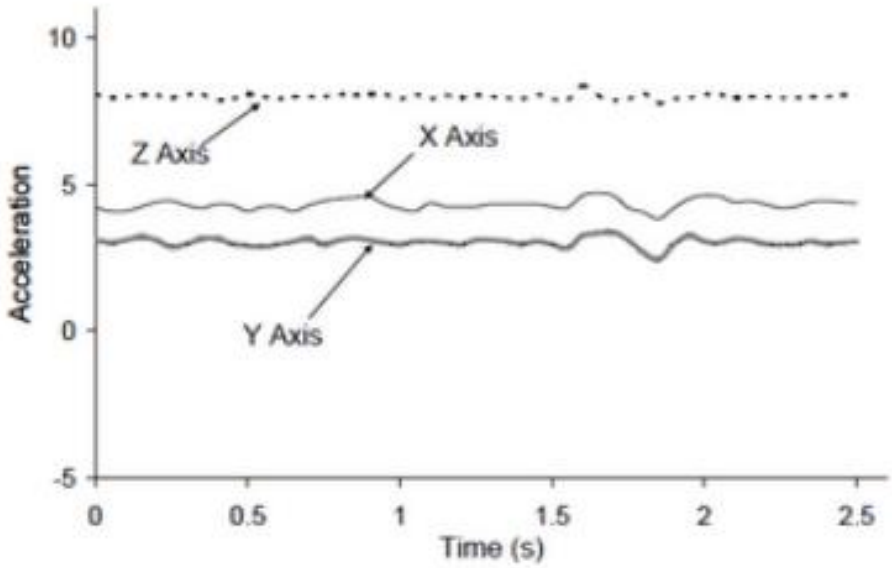
Waking



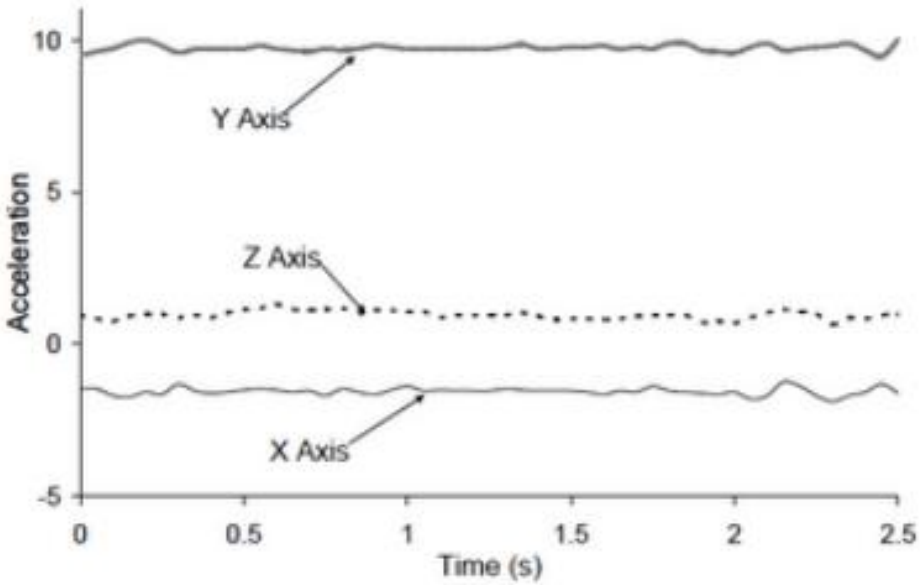
Jogging

Acceleration Signals

- Let's say we also want to know if a person is standing or sitting.

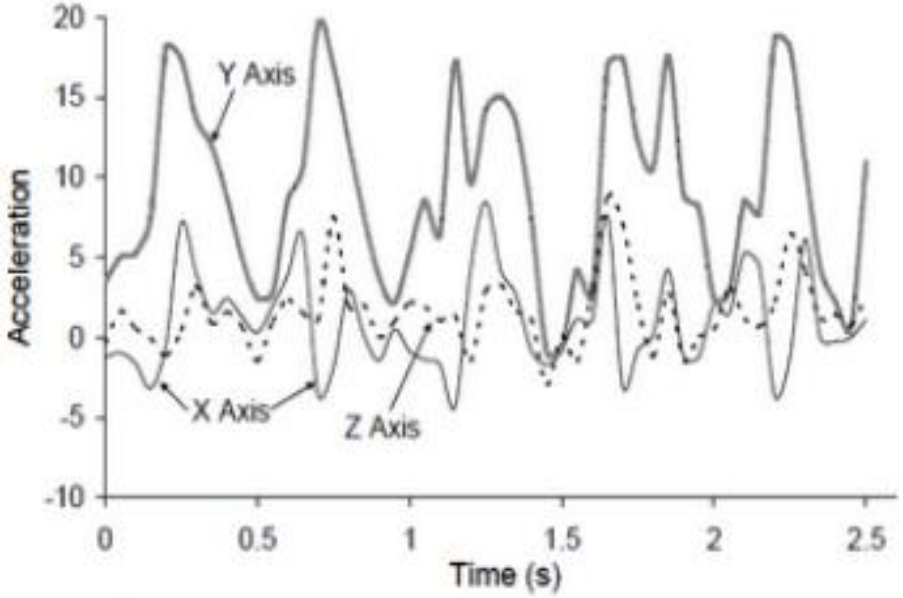


Sitting

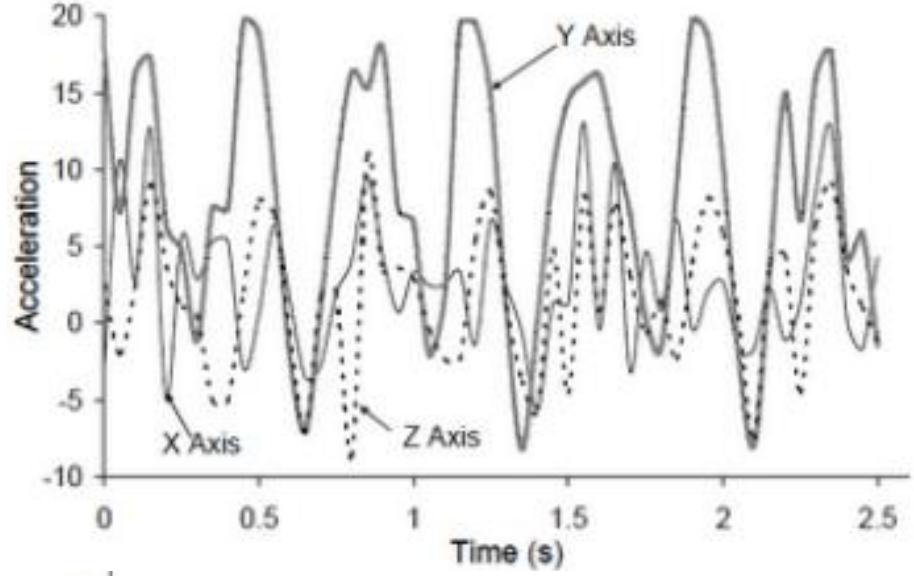


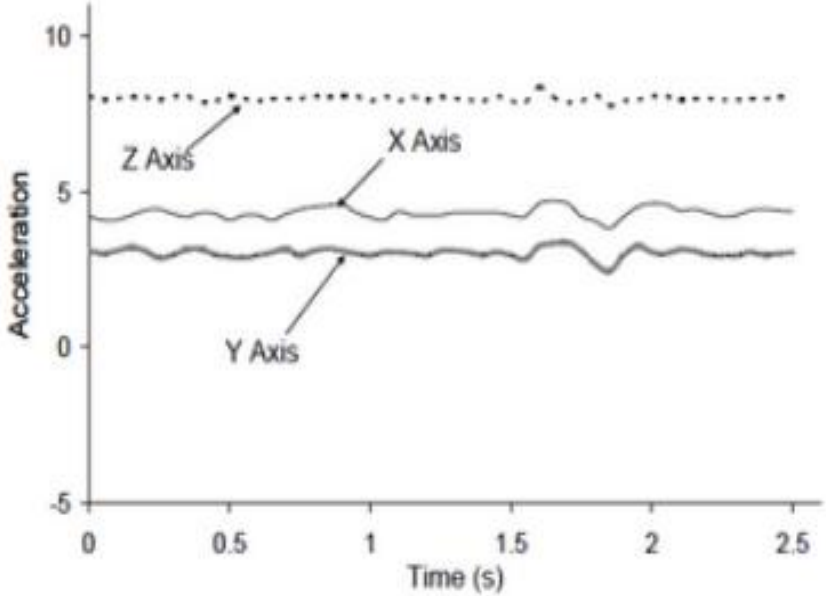
Standing

Waking

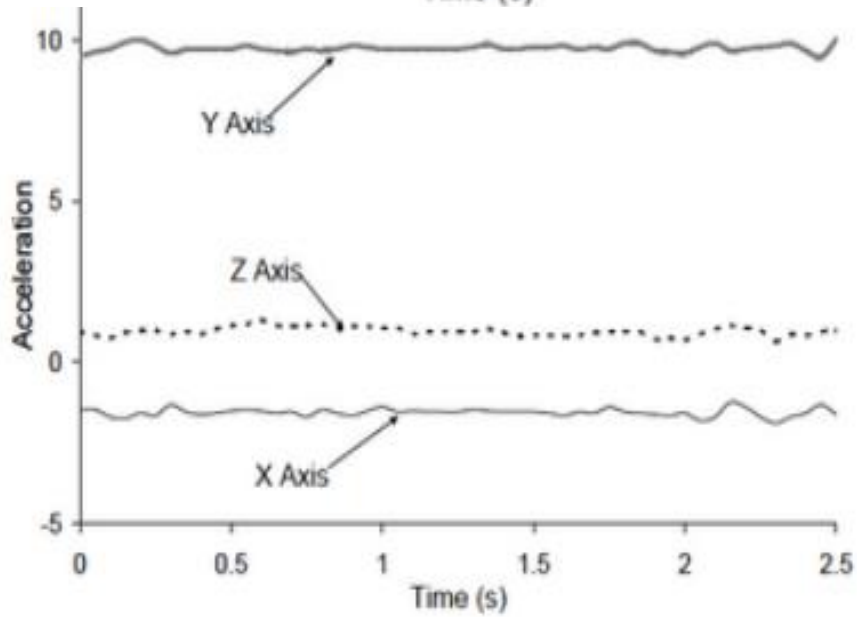


Jogging





Sitting



Standing

How Do We Classify Activities?

- Input: 2.5 seconds of 3-axis acceleration data (sampling rate: 120 Hz)

- Output: User activity (one of sitting, standing, walking, jogging)

A Simple Heuristic?

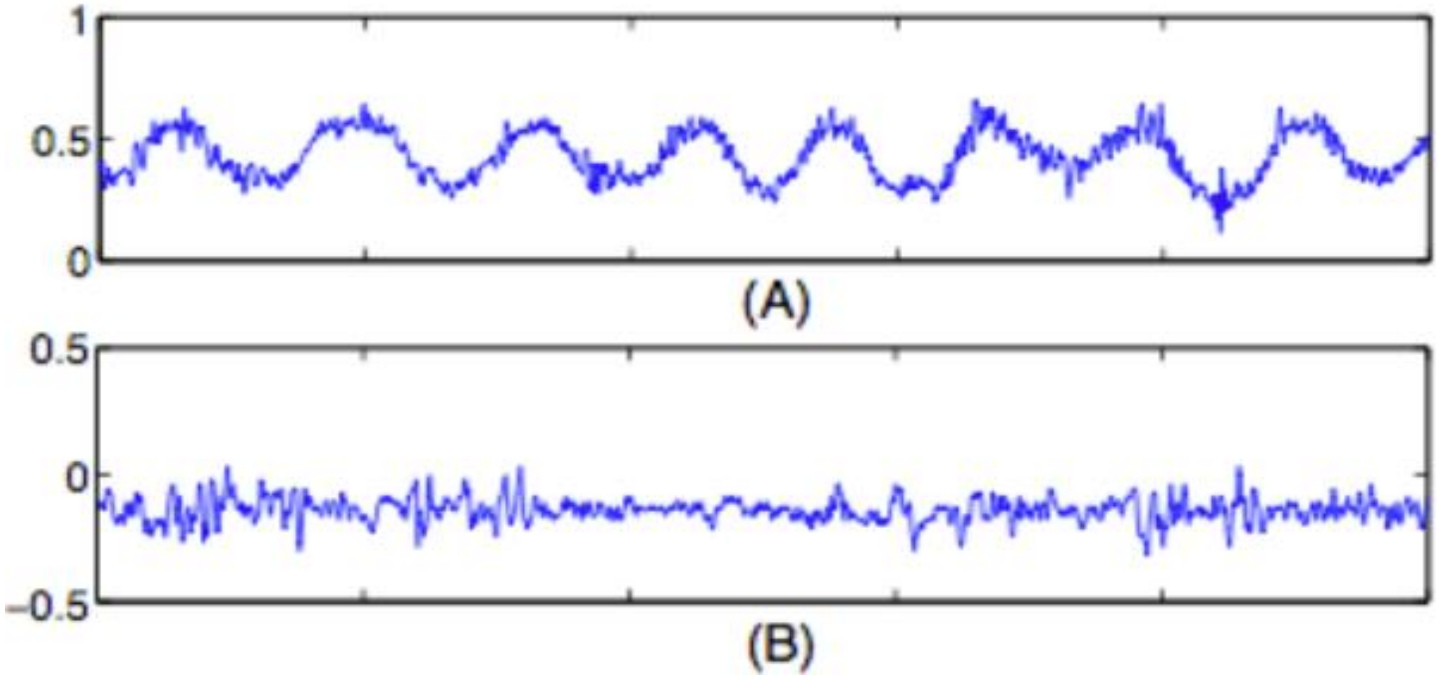
- If STDEV(y-axis samples) < $C_{Threshold1}$
  - If AVG(y-axis samples) > $C_{Threshold2}$
    - output standing
  - Else
    - output sitting
- Else
  - If FFT(y-axis samples) < $C_{Threshold3}$
    - output walking
  - Else
    - output jogging

Problems of The Heuristics

- How do we determine good features and good thresholds?
  - How do we know STDEV is better than MAX?
  - How do we know AVG is better than Median?
  - How do we know the right values for $C_{threshold}$?
- What if a user puts her phone in her bag, not in her front pocket?
  - The Y-axis of the phone is not anymore the major axis of movement.
- How do we solve these problems? A better heuristic?

One Activity, Two Distinct Patterns

Acceleration while cycling

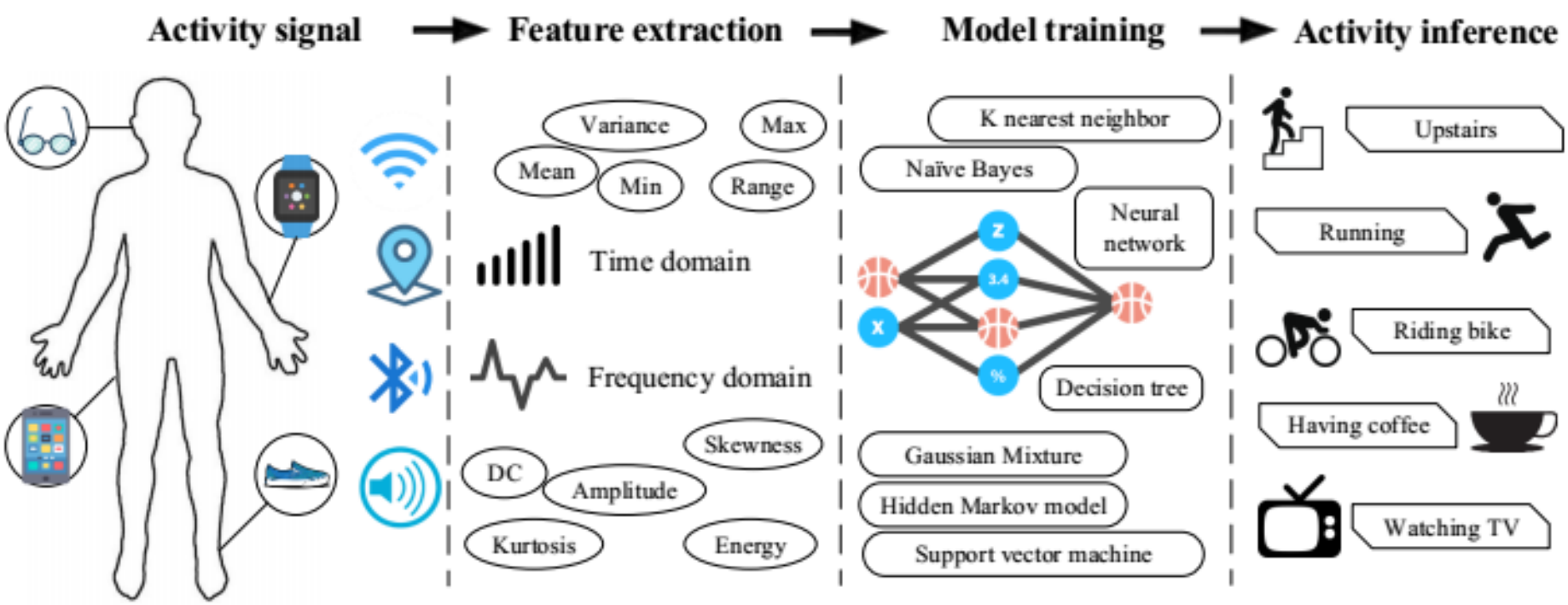


(a) Phone in the pocket

(b) phone in the backpack

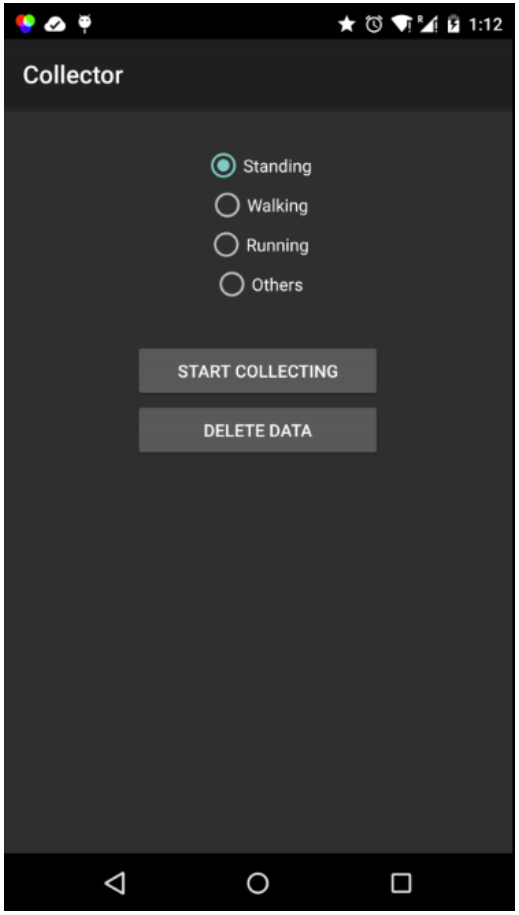
Source: Prof. Andrew Campbell's Lecture Note

Machine Learning Techniques



Step 1: Data Collection

- The first step is to collect labeled data.
- Labels mean that the ground truth corresponding to the raw data.
- E.g.) an hour of the raw accelerometer data from a phone, as well as user-provided labels regarding their state (walking, running, etc.).
- This data is referred to as a training dataset.
- Need to collect sufficient data for each activity to classify.



Step 2: Feature Extraction

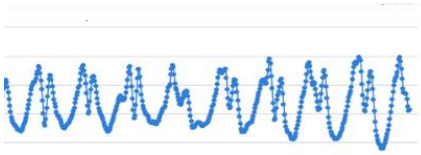
- Identify distinguishing features in the data
- Time-domain features: Aggregate statistics of the data (e.g., avg. stdev.)
- Frequency domain features: Periodic patterns and rhythmic behavior in the signal. (e.g., walking and running have different dominant frequencies)

Time domain features	Frequency domain features
Mean, Median, Variance, Standard deviation, Min, Max, Range, Zero-crossings, Angle, Angular velocity, etc.	Dominant frequency, Signal Energy, etc.

Principal Component Analysis

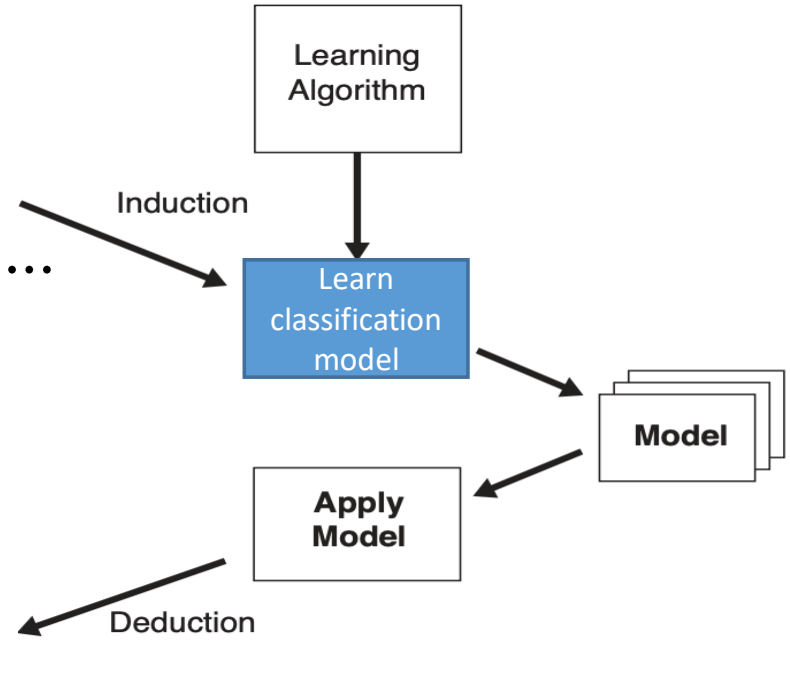
- A classifier identifies which of the features is most useful in distinguishing between the different activities.

Training Data:

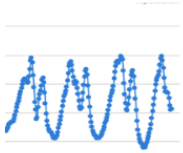


Labels: walking, running, sitting, …

Features to use

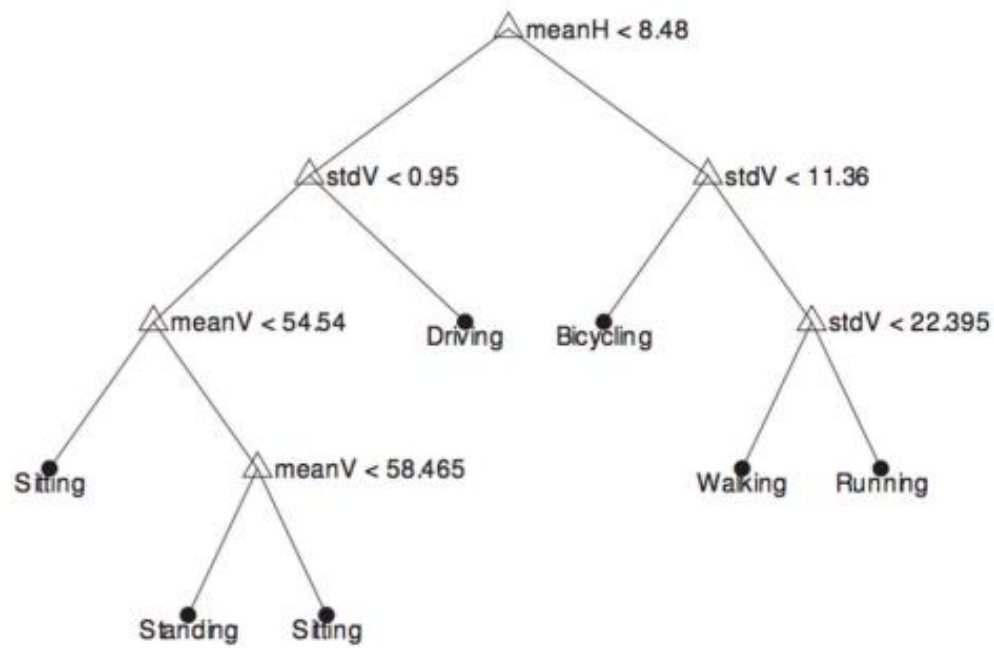


New Data



Decision Tree

- A simple but effective ML classifier.
- Given training data, the algorithm can automatically determine the important features and their thresholds.
- Then, when a new data is given, it is trivial to classify what activity it belongs to



An example decision tree to distinguish 6 activities with 3 features

How to Build a Decision Tree?

- Search for the C4.5 algorithm if you are interested.

- Pseudocode

1. For each feature *f*, find the normalized information gain (a metric to effectively split data into classes) from splitting on *f*
2. Let *f_best* be the attribute with the highest normalized information gain
3. Create a decision node that splits on *f_best*
4. Recurse on the sublists obtained by splitting on *f_best*, and add those nodes as children of node

Other ML Techniques

- Random Forest
- Support Vector Machine
- Naïve Bayes
- Hidden Markov Model
- Gaussian Mixture Model
- Neural Networks
- …

Activity Recognition

- This still is an active on-going research topic.
- To recognize various types of activities (e.g., eating, smoking, exercising, swimming, etc.).
- To recognize activities more accurately using different types of devices, sensors, and machine learning algorithms.
- To recognize various activities in a resource efficient way.

Active Research Directions

- **Use and fuse other sensor data**

- **Use of deep learning**