Machine Learning for the Quantified Self

Chapter 3

Handling Noise and Missing Values in Sensory Data

Overview

- Previously: we collected the data
- Today: Removing noise from the data
  - Removal of outliers
  - Imputation of missing values
  - Transform the data to select the most useful information

Removal of outliers (1)

- What is an outlier?
- An outlier is an observation point that is distant from other observations
- Causes?
  - Measurement error (Arnold with a heart rate of 400)
  - Variability (Arnold trying to push his limits with a heart rate of 190)

Removal of outliers (2)

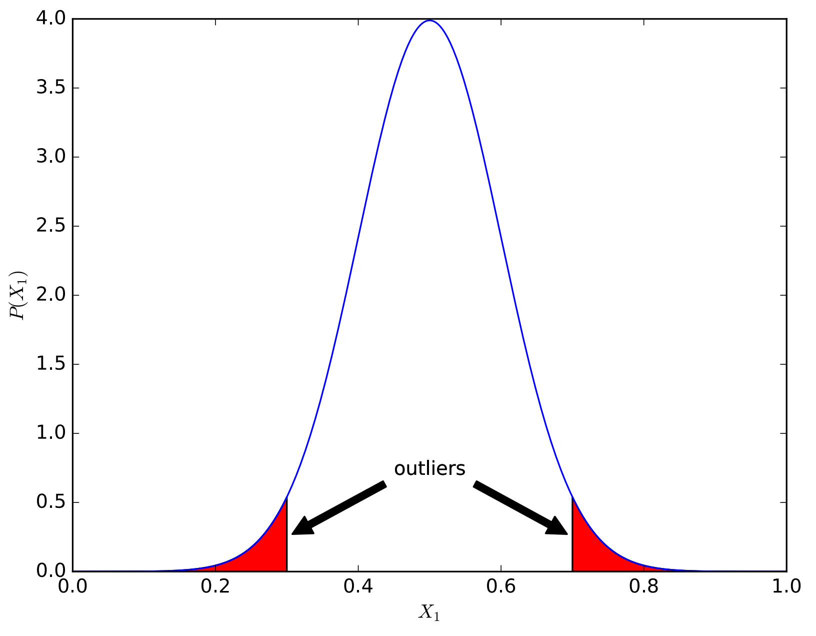
- Difference between measurement and variability outlier?
  - Former generated by *another mechanism*
- How to remove?
  - Domain knowledge (heart rate cannot be over 220)
  - Without domain knowledge (our focus)
- Have to be **cautious** as you do now want to remove valuable information

Removal of outliers (3)

- Two types of outlier detection:
  - Distribution based (we assume a certain distribution of the data)
  - Distance based (we only look at the distance between data points)

Distribution-based outlier detection (1)

- Let us start with Chauvenet's criterion
- Assume a normal distribution, single attribute (X_i)



Chauvenet's criterion (1)

- Take the mean and standard deviation for an attribute *j* in our dataset:

$$\mu = \frac{\sum_{n=1}^{N} x_n^j}{N}$$

$$\sigma = \sqrt{\frac{\sum_{n=1}^{N} (x_n^j - \mu)^2}{N}}$$

Chauvenet's criterion (2)

- Take those values as parameters for our normal distribution
- For each instance *i* for attribute *j* compute the probability of the observation:

$$P(X \leq x_i^j) = \int_{-\infty}^{x_i^j} \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(u-\mu)^2}{2\sigma^2}} \delta u$$

Chauvenet's criterion (3)

- Define it as an outlier when:

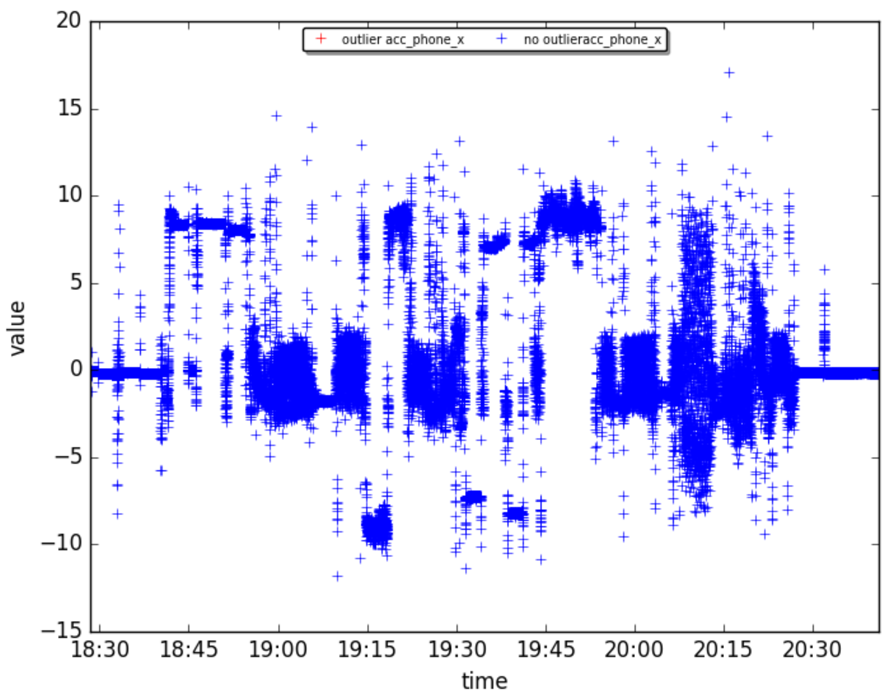
$$(1 - P(X \leq x_i^j)) < \frac{1}{c \cdot N}$$

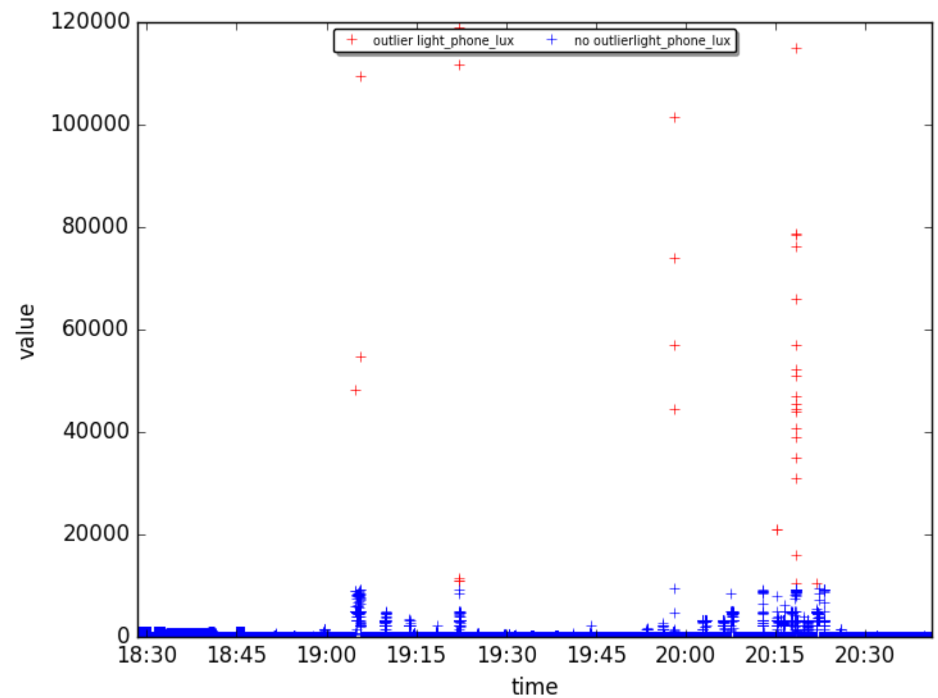
$$P(X \leq x_i^j) < \frac{1}{c \cdot N}$$

- Typical value for *c* is 2

Chauvenet's criterion (4)

- CrowdSignals example (c=2):





Distribution-based outlier detection (2)

- Assuming the data of an attribute to follow a single distribution might be a bit too simple
- We can also use *mixture models*
- Assume the data can be described with *K* normal distributions

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k)$$

with

$$\sum_{k=1}^{K} \pi_k = 1$$

$$\forall k : 0 < \pi_k \leq 1$$

Mixture models (1)

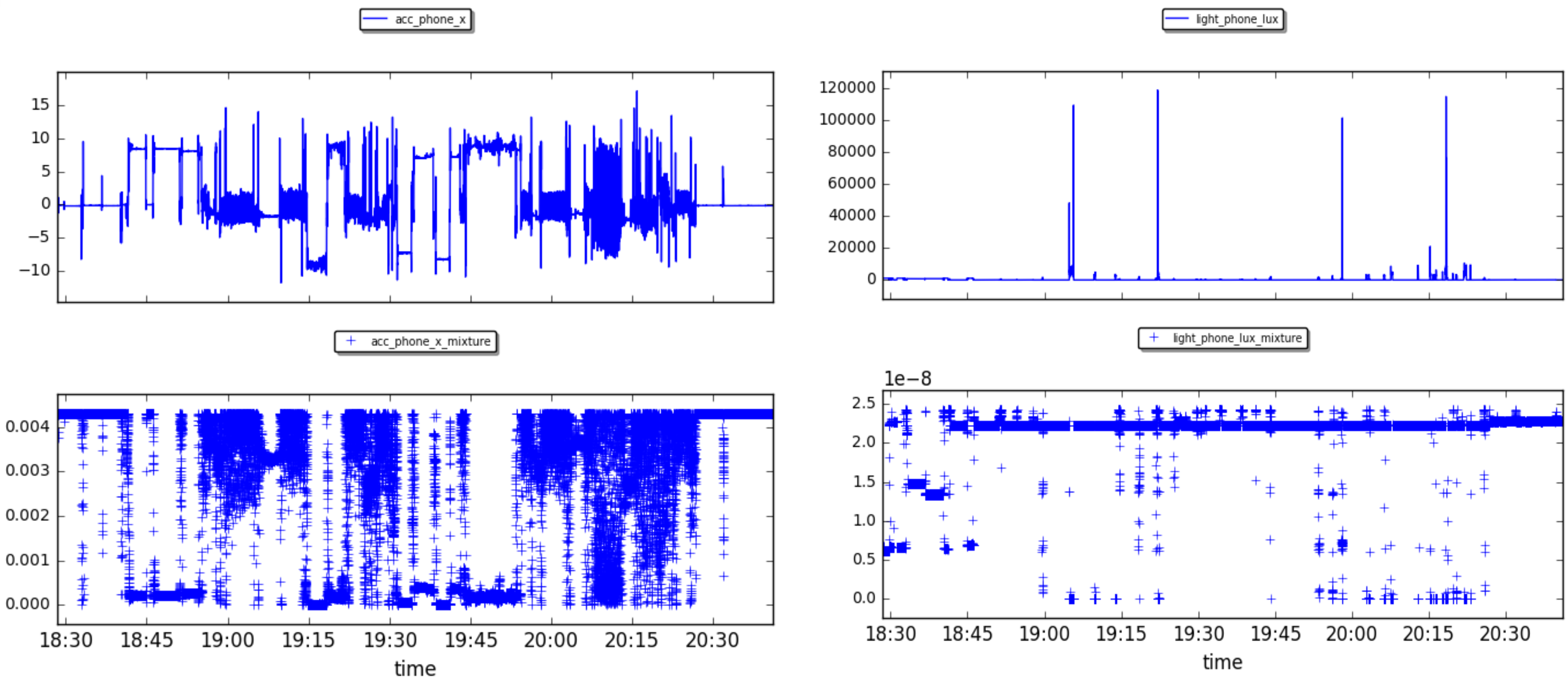
- We can find the best for the parameters by mean of maximizing the likelihood:

$$L = \prod_{n=1}^{N} p(x_n^j)$$

- We can for example use the expectation maximization algorithm
- How many distributions should we use?

Mixture models (2)

- CrowdSignals example (K=3):

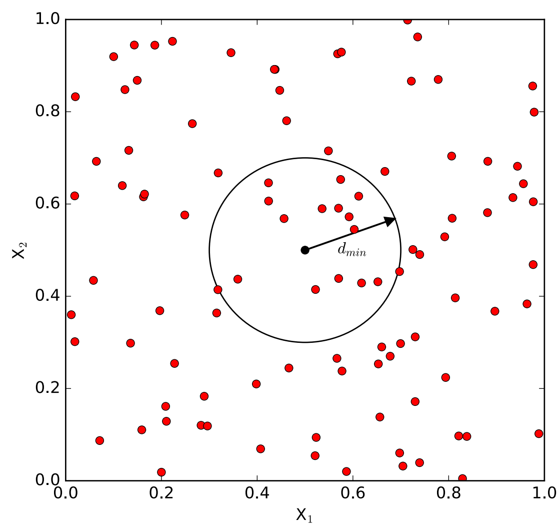


Distance-based outlier detection (1)

- Let us move away from distributions and just consider the distance between points
- We will consider the actual distance metrics later (Chapter 5), but e.g. think of Euclidean distance
- We will use $d(x_i^j, x_k^j)$ to represent the distance between two values of an attribute *j*

Simple distance-based approach (1)

- We call point close if they are within distance d_{min}
- Points are outliers when there are more than a fraction f_{min} far away (i.e. outside of d_{min})



Simple distance-based approach (2)

- Formal:

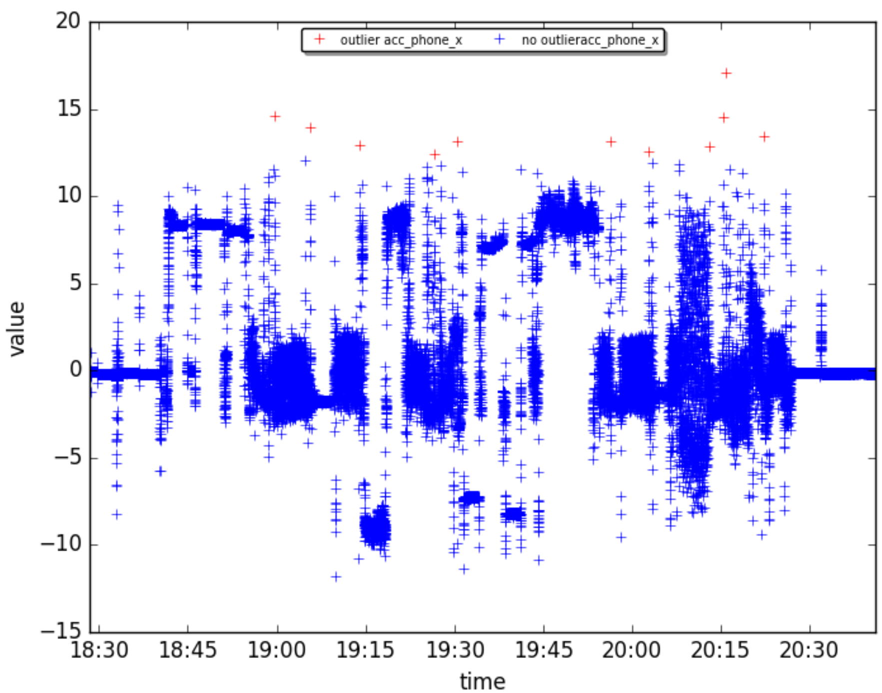
$$outlier(x_i^j) = \begin{cases} 1 & \frac{\sum_{n=1}^{N} d\_over(x_i^j, x_n^j, d_{min})}{N} > f_{min} \\ 0 & otherwise \end{cases}$$

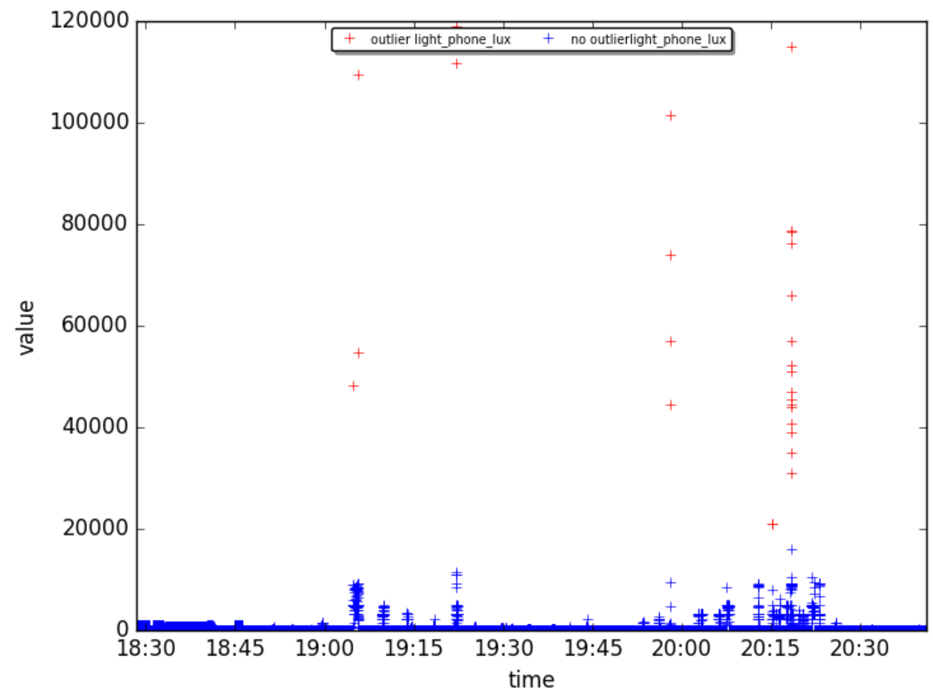
with

$$d\_over(x, y, d_{min}) = \begin{cases} 1 & d(x, y) > d_{min} \\ 0 & otherwise \end{cases}$$

Simple distance-based approach (3)

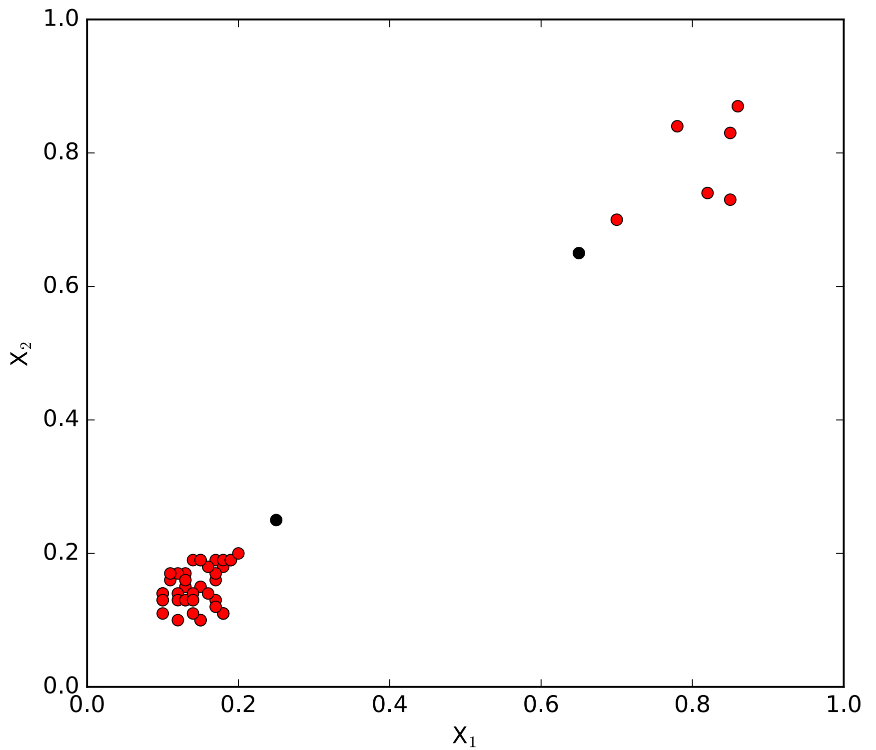
- CrowdSignal example (d_{min}=0.1, f_{min}=0.99)





Distance-based outlier detection (2)

- The previous approach did not take the local density into account, imagine:



Local outlier factor (1)

- Local outlier factor does take this density into account
- We first define the distance k_{dist} for a point x_i^j as the largest distance to one of its k closest neighbors:

$$|\{x|x \in \{x_1^j, \ldots, x_{i-1}^j, x_{i+1}^j, \ldots, x_N^j\} \wedge d(x, x_i^j) \leq k_{dist}(x_i^j)\}| \geq k$$

$$|\{x|x \in \{x_1^j, \ldots, x_{i-1}^j, x_{i+1}^j, \ldots, x_N^j\} \wedge d(x, x_i^j) < k_{dist}(x_i^j)\}| \leq (k-1)$$

Local outlier factor (2)

- The set of neighbors of x_i^j within k_{dist} is called the k-distance neighborhood k_{dist_nh}

$$k_{dist\_nh}(x_i^j) = \{x | x \in \{x_1^j, \ldots, x_{i-1}^j, x_{i+1}^j, \ldots, x_N^j\} \wedge d(x, x_i^j) \leq k_{dist}(x_i^j)\}$$

- We define the reachability distance of x_i^j to x as (we remove small distances within k_{dist})

$$k_{reach\_dist}(x_i^j, x) = max(k_{dist}(x), d(x, x_i^j))$$

Local outlier factor (3)

- Now we define the local reachability distance of our point x_i^j:

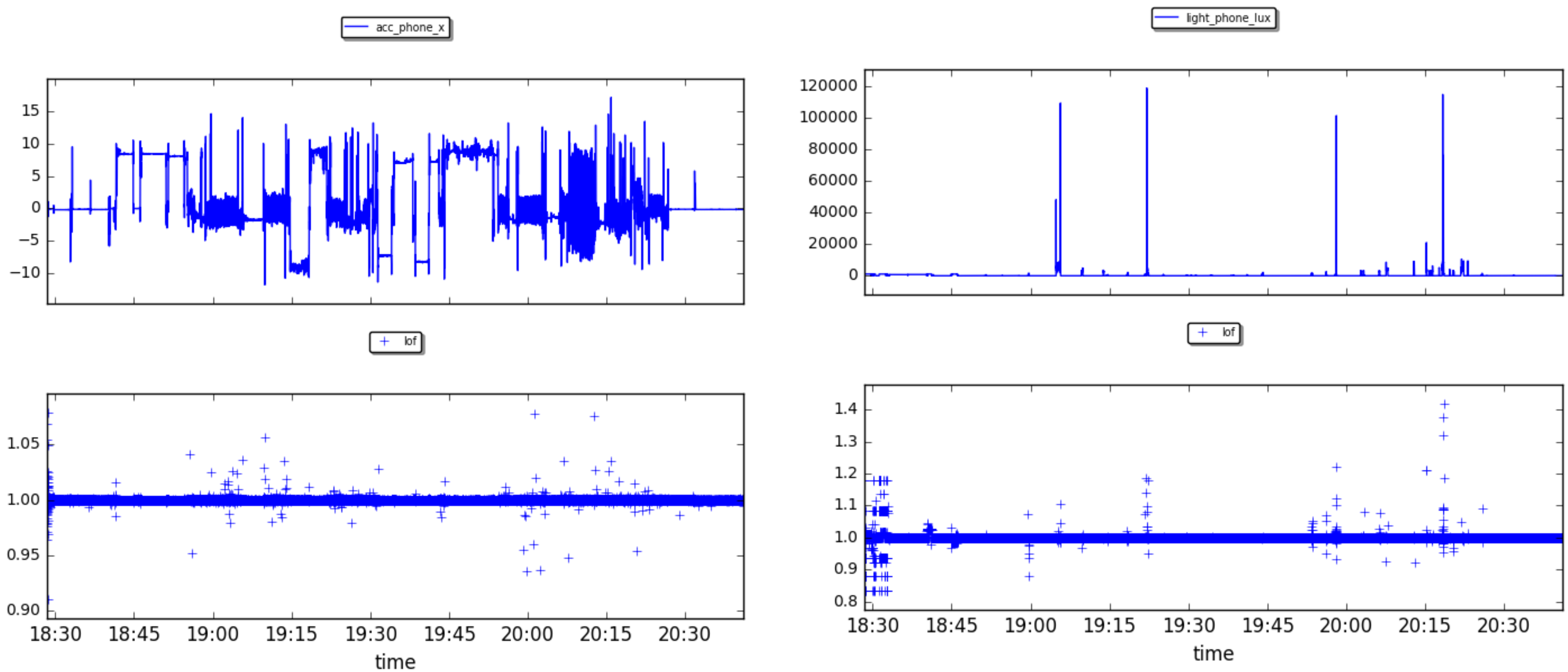
$$k_{lrd}(x_i^j) = 1 / \left(\frac{\sum_{x \in k_{dist\_nh}(x_i^j)} k_{reach\_dist}(x_i^j, x)}{|k_{dist\_nh}(x_i^j)|} \right)$$

- And we compare this to the neighbors:

$$k_{lof}(x_i^j) = \frac{\sum_{x \in k_{dist\_nh}(x_i^j)} \frac{k_{lrd}(x)}{k_{lrd}(x_i^j)}}{|k_{dist\_nh}(x_i^j)|}$$

Local outlier factor (4)

- CrowdSignals case (k=5):



Outlier detection

- We remove all elements we have considered to be an outlier
- We replace them with the value missing

Missing values (1)

- We naturally move to missing values
- We can replace missing values by a substituted value (*imputation*)
- What should these values be?
  – mean (numeric)
  – mode (categorical and numeric)
  – median (numeric)

Missing values (2)

- We can also take more advanced approaches:
  - Use other attribute values in the same instance (Chapter 7):
    $x_i^1, \ldots, x_i^{j-1}, x_i^{j+1}, \ldots, x_i^p \rightarrow x_i^j$
  - Use values of the same attributes from other instances (need a ordered/temporal attribute):
    $x_1^j, \ldots, x_{i-1}^j, x_{i+1}^j, \ldots, x_N^j \rightarrow x_i^j$

Missing values (3)

- Some examples of the second case in case of a single missing measurement:

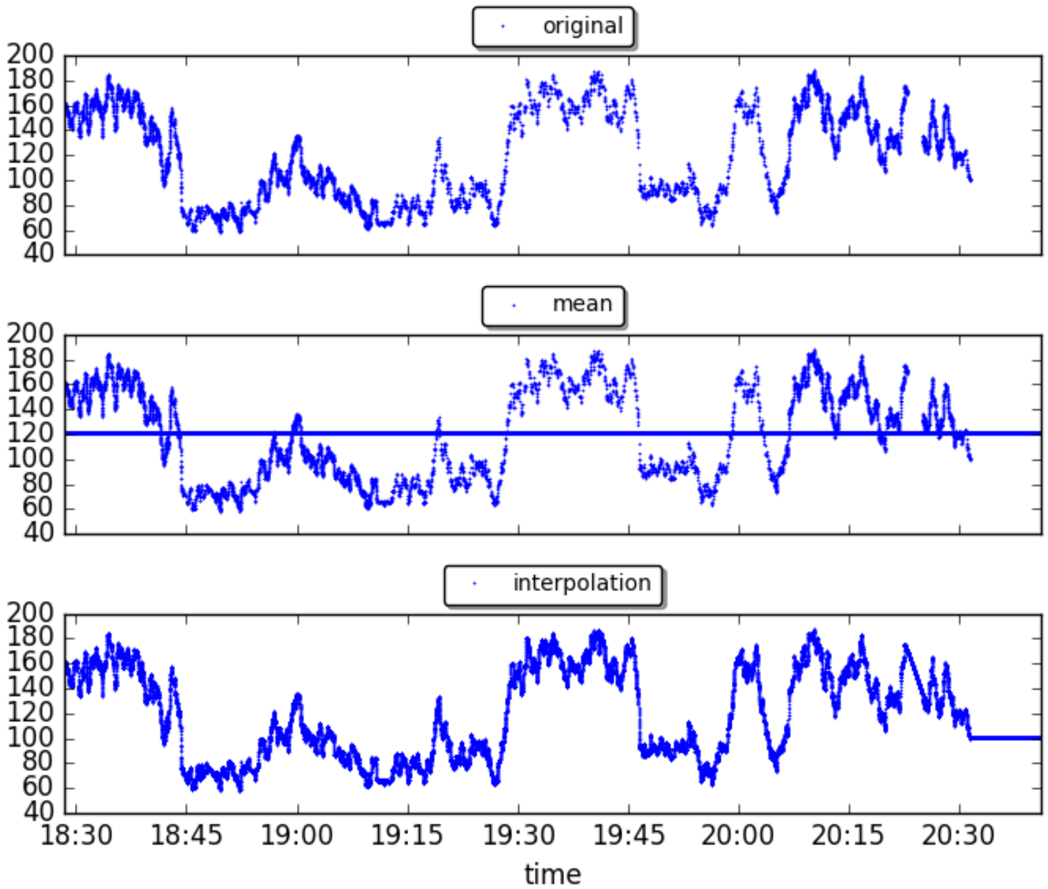
$$x_i^j = \frac{x_{i-1}^j + x_{i+1}^j}{2}$$

- In case of multiple missing measurements:

$$x_i^j = x_{i-k}^j + k \cdot \frac{x_{i+l}^j - x_{i-k}^j}{(k+l)}$$

Missing values (4)

- CrowdSignal example:



Outlier detection + imputation

- Approaches that combine outlier detection and value imputation exist as well
- The Kalman filter is a well known one:
  - it estimates expected values based on historical data
  - if the observed value deviates too much (i.e. an outlier) we can impute with the expected value

Kalman filter (1)

- Assume some latent state s_t which can have multiple components
  - Our quantified self data x_t performs measurements about this state
- For example:
  - s_t is Arnold's presence at a position and velocity
  - x_t is the GPS data and step counter

Kalman filter (2)

- The next value of a state is defined as:

$$s_{t+1} = F_t s_t + B_t u_t + w_t$$

  - u_t is a control input state (e.g. sending a message)
  - w_t is white noise
  - F_t and B_t are matrices
- The measurement associated with s_t is:

$$x_t = H_t s_t + v_t$$

  - v_t is white noise

Kalman filter (3)

- For the noise we assume that: $$w_t = \mathcal{N}(0, Q_t)$$ $$v_t = \mathcal{N}(0, R_t)$$
- Now let us try to predict a next state (denoted by a hat):

$$\hat{s}_{t|t-1} = F_t \hat{s}_{t-1|t-1} + B_t u_t$$

- And let us also estimate our prediction error (matrix of variances and co-variances):

$$P_{t|t-1} = \mathbb{E}[(s_t - \hat{s}_{t|t-1})(s_t - \hat{s}_{t|t-1})^T]$$

Kalman filter (4)

- Based on our prediction of the state, let us look at the error:

$$e_t = x_t - H^T \hat{s}_{t|t-1}$$

- Given this error we come with an updated prediction of our state

$$\hat{s}_{t|t} = \hat{s}_{t|t-1} + K_t e_t$$

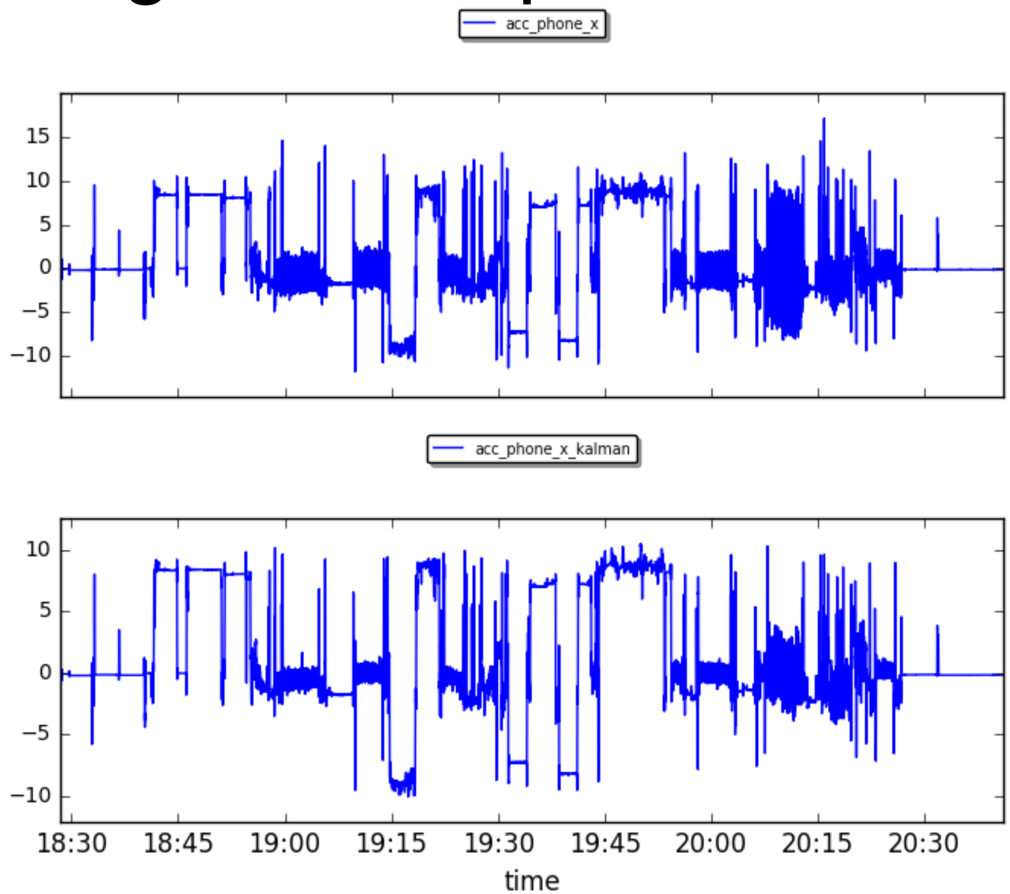
- K_t is a matrix derived based on an algorithm for which $P_{t|t-1}$ is used

Kalman filter (5)

- Of course the matrices contain models (that we might not always know) but you can even do without (one measurement directly related to one latent state)
- Once we observe a large error, our x_t might be of and we can substitute it by the expected value
- Nice more extensive explanation: http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

Kalman filter (6)

- CrowdSignal example:



Transforming the data (1)

- Even though we have removed the outliers and imputed missing values we could still suffer from noise in our dataset that could distract the learning process
- Approaches exist that filter our this more subtle noise
  - Lowpass filter
  - Principal Component Analysis

Lowpass filter (1)

- Main idea: some data has periodicity (e.g. walking, running)
- You can decompose a series of values into different periodic signals:
  - come with their own frequency
  - we will see about this in Chapter 4
- Some frequencies might be more interesting than others

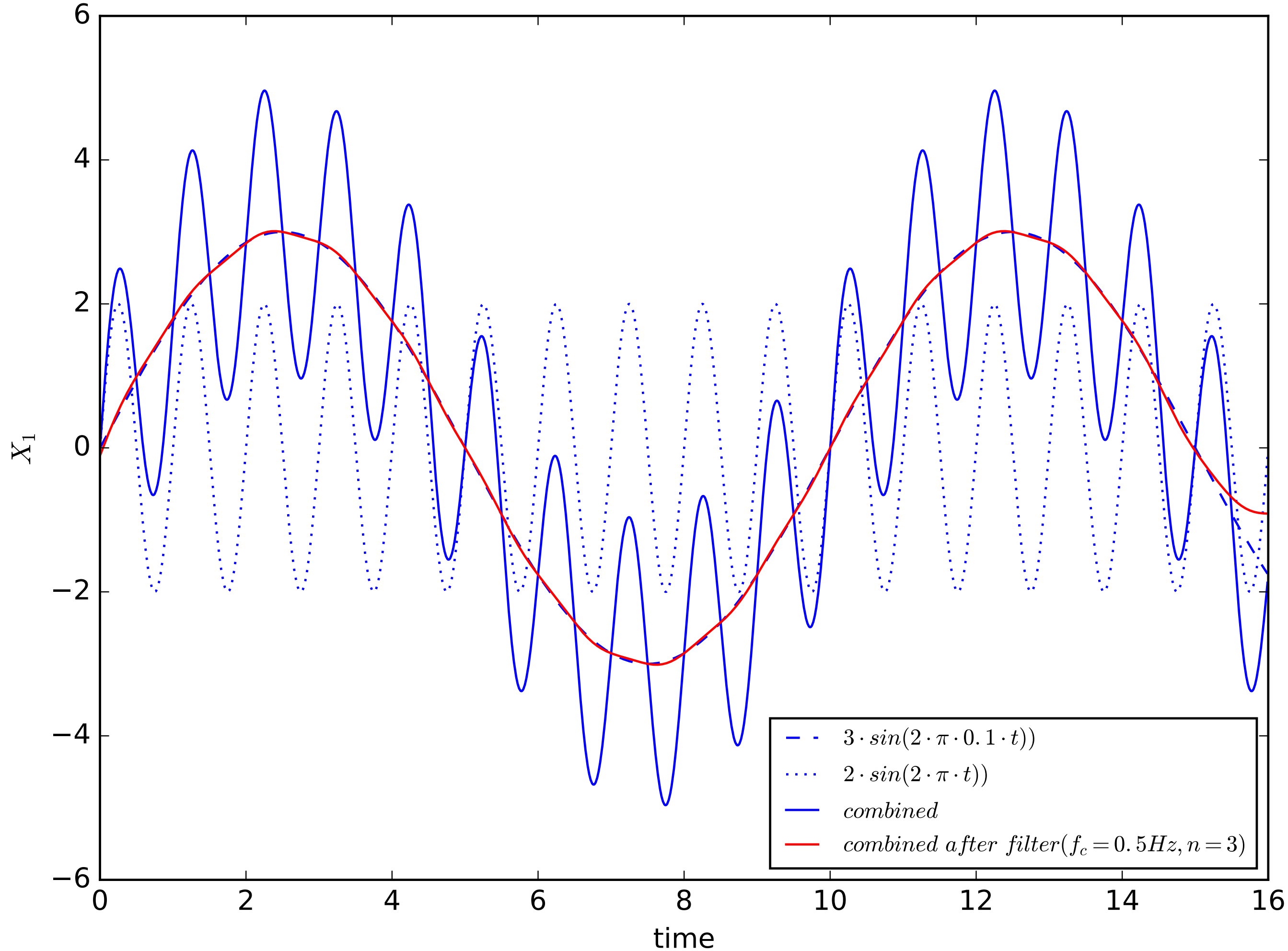
Lowpass filter (2)

- For example: we do not care about running (higher frequency), but we do care about walking
- We can filter out the higher frequency data
- The lowpass filter does exactly this:

$$|G(f)|^2 = \frac{1}{1 + (f/f_c)^{2n}}$$

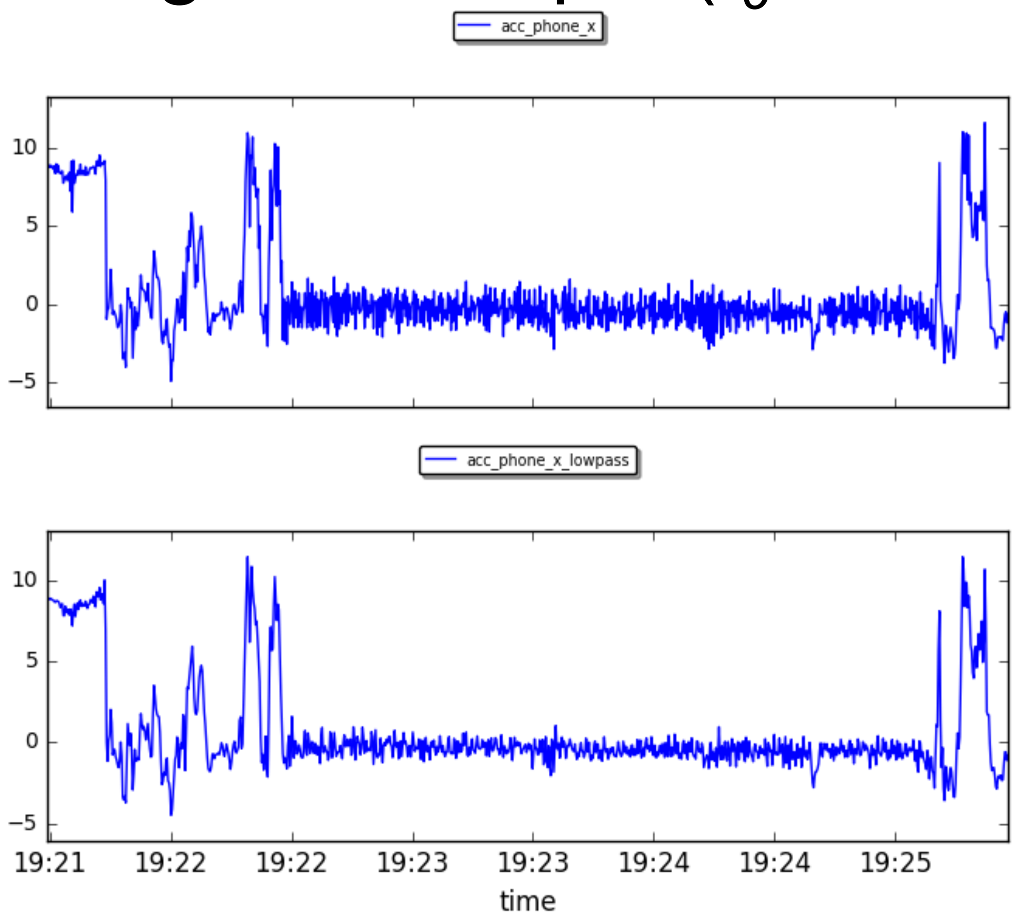
  - $|G(f)|$ is the magnitude of the filter
  - f_c is the cutoff frequency
  - n is the order of the filter

Lowpass filter (3)



Lowpass filter (4)

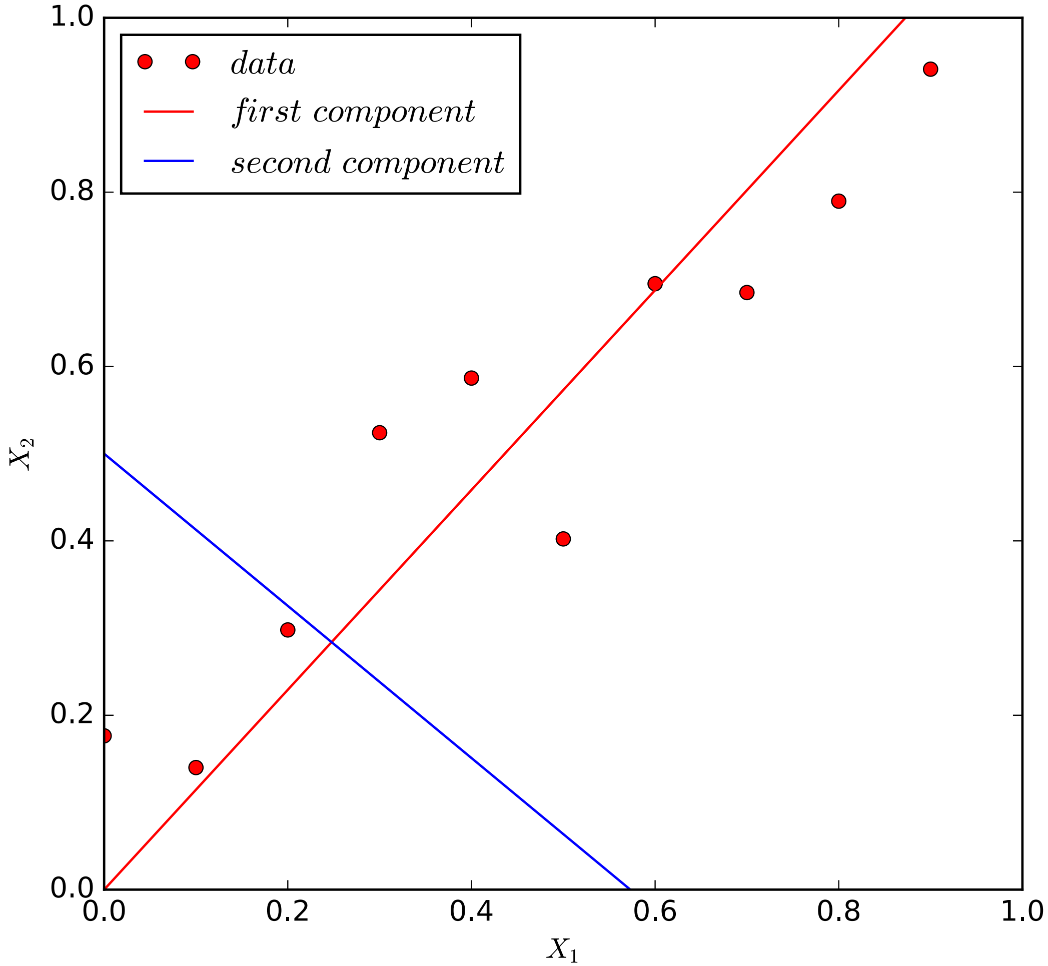
- CrowdSignal example (f_c=1.5Hz, n=20):



Transforming the data (2)

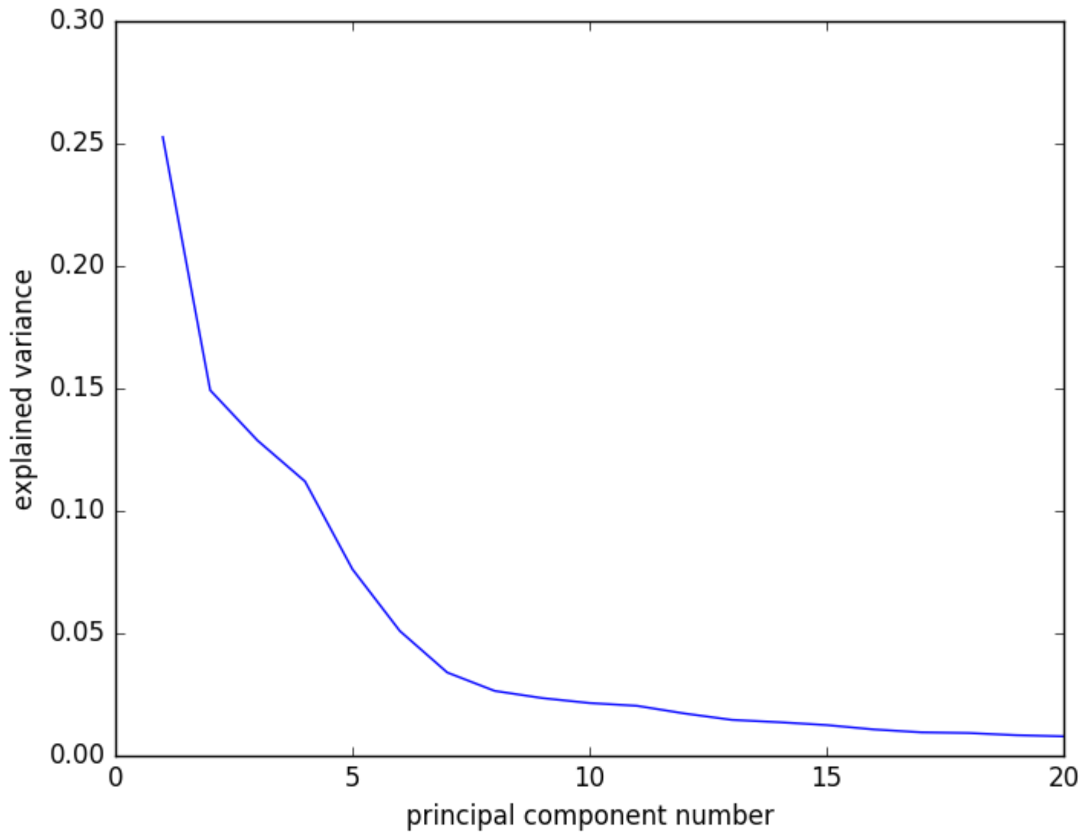
- We can also apply principal component analysis:
  - find new features that explain most of the variability in our data
  - select the number of components based on the explained variance
  - since most are familiar, I will not provide all details, see the book

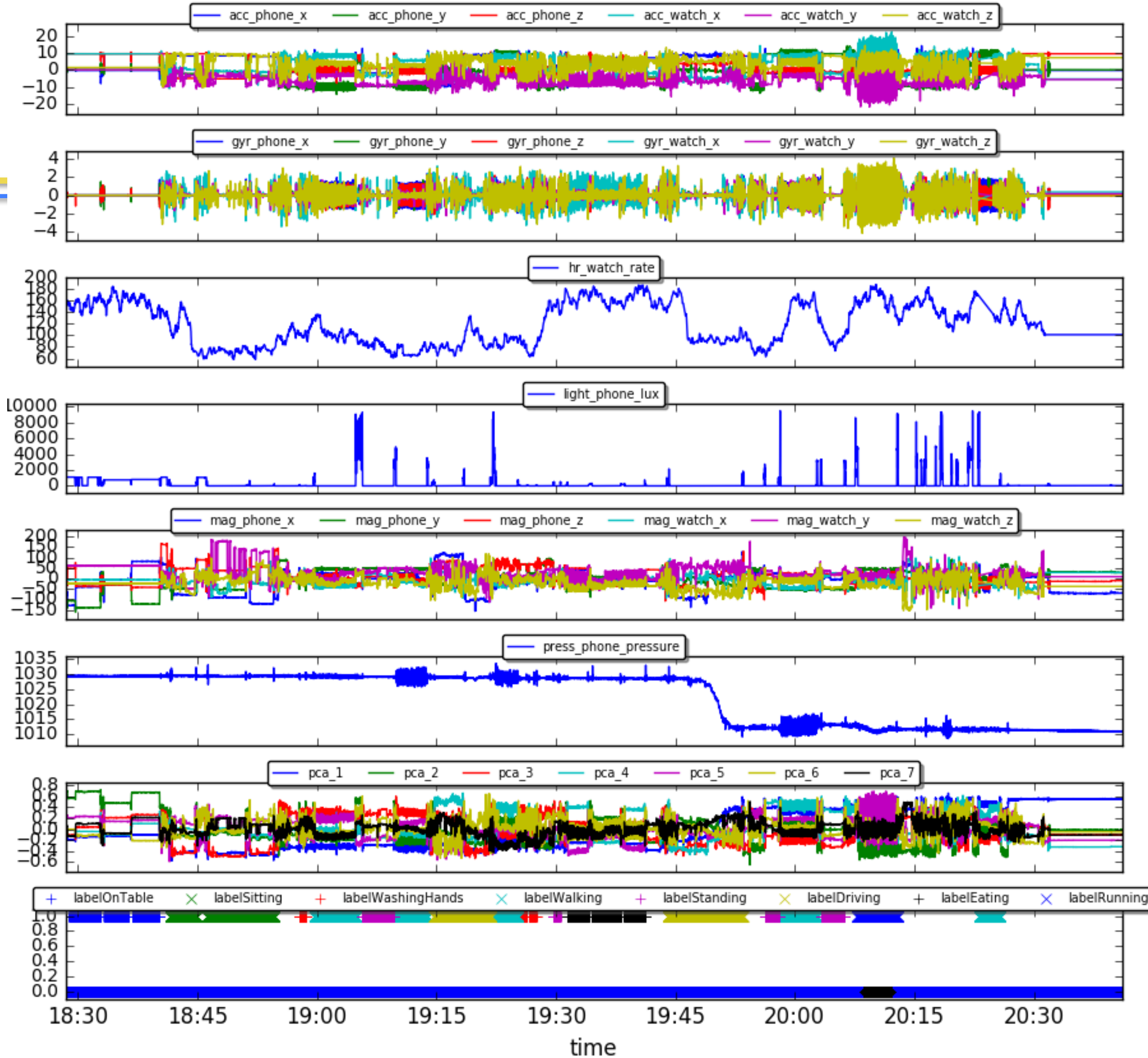
Principal component analysis (1)



Principal component analysis (2)

- CrowdSignals:





acc_phone_x acc_phone_y acc_phone_z acc_watch_x acc_watch_y acc_watch_z
gyr_phone_x gyr_phone_y gyr_phone_z gyr_watch_x gyr_watch_y gyr_watch_z
hr_watch_rate
light_phone_lux
mag_phone_x mag_phone_y mag_phone_z mag_watch_x mag_watch_y mag_watch_z
press_phone_pressure
pca_1 pca_2 pca_3 pca_4 pca_5 pca_6 pca_7
labelOnTable labelSitting labelWashingHands labelWalking labelStanding labelDriving labelEating labelRunning
18:30 18:45 19:00 19:15 19:30 19:45 20:00 20:15 20:30
time

Summary

Approach	Purpose	$\mathbf{X}^{\mathcal{T}}$ specific?	Number of attributes considered	Brief summary
Chauvenets criterion	Outlier detection	No	1	Identify values for an attribute that are unlikely given a single normal distribution to describe the data.
Mixture model-based outlier detection	Outlier detection	No	1	Identify values for an attribute that are unlikely given a combinations of distributions to describe the data.
Simple distance-based outlier detection	Outlier detection	No	$1,\ldots,p$	Identify instances or attribute values at a great distance from other points.
Local outlier factor	Outlier detection	No	$1,\ldots,p$	Identify instances or attribute values who are more distant from other points than other close by points are to their closest points.
Mean imputation	Missing value imputation	No	1	Impute the mean value for an attribute for an unknown value or outlier.
Median imputation	Missing value imputation	No	1	Impute the median value for an attribute for an unknown value or outlier.
Mode imputation	Missing value imputation	No	1	Impute the mode value for an attribute for an unknown value or outlier.
Interpolation-based imputation	Missing value imputation	Yes	1	Impute the value for an attribute by extrapolating the previous and next measurement.
Model-based imputation	Missing value imputation	No	1	Impute the value for an attribute by creating a model to predict it.
Kalman filter	Outlier detection & Missing value imputation	Yes	$1,\ldots,p$	Create estimations of expected values based on historical observations and impute with estimated value when values are too deviant.
Lowpass Butterworth filter	Transformation	Yes	1	Remove periodic irrelevant data of a single attribute over time.
Principal Component Analysis	Transformation	No	p	Condense most of the variability of the data in a set of new features.

Summary

- Have learned how to handle all the noise in our data using various approaches
- This will help our machine learning algorithms later
- Next time: feature extraction