ACM SIGCHI Summer School on User Modeling and Personalization in Urban Computing (Umcit 2019)

kenneth E.I
Unversity Jean Monnet, Saint-Etienne.
{*ifeanyi.ezukwoke@etu.univ-st-etienne.fr*}

1 Day 1

1.1 Opening presentation and key note

MARCELO G. ARMENTANO

- Exploring the importance of sensors in smart devices for urban sensing.
- How to gather data through crowdsensing, mobile sensors and applications.
- Urban computing for solving traffic estimation challenge.

1.2 Mobile Crowdsensing

- Brief history of smartphones and development.
- Gathering data using mobile sensors
  - mobile sensor gathers data of user mobility.
  - gathering information from user like spots/browser history/non-likes etc.
  - purpose is to design recommendation system for large scale use.

1.2.1 Mobile Sensing

- Smartphone sensing
- Sensing via Software APIs
- Mobile Apps

1.2.2 Personal Sensing

- Collecting data for user purpose **ONLY**

1.2.3 Public Sensing

- Collecting data for public purpose. Data can be stored on protected/unprotected cloud servers.

1.3 Discussion on evolution of mobile phones

- discussion on evolution of sensor devices, ranging from first generation smartphones to recently available smartphones e.g. bluetooth, infrared, camera, gyroscope, accelerometer, barometer, light sensor, ambient sensor, GPS.

1.4 Components that enable mobile computing

- Apps (personal sensing apps), surveillance, car and bike sensors etc.

1.5 Major talk on sensors

- Motion sensors
  - Accelerometer (for measuring the acceleration of user. very sensitive)
  - Gyroscope (navigational precision. stable)
  - Magnetometer (direction strength)
  - Proximity (distance with respect to destination)
  - Pedometer (steps movements of the legs)
- Environmental sensors
  - Ambient light
  - Barometer
- Radio
  - GPS
  - Cellular radios (not easily available data e.g cell phone antennas)
  - Wifi
  - Bluetooth
- External sensors
  - Car sensors
  - Bike sensors
  - Wearables

1.6 Crowd Sensing

Brief talk on its meaning and applications. Existing application using the concept of crowd sensing.

1.6.1 Environmental application areas

- Common sensing: pollution monitoring
- CreekWatch: monitor water level and creek quality
- MobGeoSen: local environment pollution.
- NoiseTube: monitor noise pollution.

1.6.2 Infrastructural application areas

- MIT Cartel & Microsoft Nericell
- ParkNet
- Traffic Sense
- PetrolWatch

1.6.3 Social application areas

- BikeNet
- Diet Sense
- Party thermometer
- LiveCompare

1.6.4 Road transport application areas

- Picture analysis
- Road quality
- Traffic re-routing
- Large scale patterns of traffic.

NOTE

- Traffic can be estimated using **accelerometer & gyroscope or GPS location**. Based on the balancing of the car sensors and the average speed of the car.
- Traffic can also be estimated through social media posts e.g twitter report/post (a typical NLP problem.)

1.7 Sensing Classification

1.7.1 Participatory sensing

- Road transport sensing
- Photo & video journalism
- Data sharing on social networks

1.7.2 Opportunistic sensing

- Road
- Traffic
- Parking

2 Day 2

ANTONELA

3 Day 3

3.1 Recommedation system

LUDOVICO
Recommedation based on previous historical pattern. Used by major online business stores.

- From Scarcity to Abundance.
  The reason for the emergence of recommender system was the need to move from shelf space to online space. Hence, recommending or proposing for clients the best buys based on previous buyers history (Amazon recommedation.)
- Editorial and hand curated
- Simple tailored
- Tailored to individuals

3.1.1 Formal representation

Requires a Utility matrix (matrix of ratings) to compute the similarity scores or the pearson correlation coefficient.

3.1.2 Key problems

- Gathering known rating matrix
  - Explicit rating: learning ratings from directly asking users.
  - Implicit rating: learning ratings from user actions/behaviors.
  - NOTE: item-item similarity often works better than user-user similarity.
- Extrapolate unknown ratings from known
- Gathering knownn rating matrix

3.1.3 Types of recommedation

- Content-based recommendation
  - Requires Item profile
  - Requires User profile
- Collaborative filtering
  - **Finding similarity between users using the Jackard similarity, Cosine similarity and Pearson correlation coefficient.**
  - Evaluation for collaborative filtering
    * Root mean square
    * Precision
    * Ranking
- Latent factor recommendation

    - Uses the concept of SVD (Singular Vector Decomposition)
    - We use a latent space to avoid sparsity.

- Point-of-Interest recommendation
    - Influential factors
        * Geographical influence
        * Social influence: uses the concept of collaborative filtering.
        * Temporary influence (periodicity, consecutiveness, non-uniformness.)
        *

    Models from POI recommendation

        * Fused model (adds up all the influences to design a recommendation system.)
        * Joint model (learns from several models and take the best performing model for recommendation)
    - Methodology
    - Task
        * General POI Recommendation
        * Successive POI Recommendation

4 Day 4

4.1 User Modeling and Adaptive systems

Federica Cena
Adapt to user interactivity. Role is to help user with information access and recommendations.

- Text adaptation

    - Canned text: only display text that meet certain conditions.

- Explicit systems
- Implicit systems

Personalized Adaptation.

- Adapative modality
- Personalization of presentation

Navigation Adaption
USer Modeling: schema for describing a user at an exact time.

5 Day 5

5.1 Spirit of the City

5.1.1 Urbanization

- Smart city == Efficiency
- Perception
- Insight

5.1.2 Quantifyng Beauty

- Collective perception of beauty. trueskill algorithm for ranking game players and match making.
- Urban Smellscape. The concept of smell contribute greatly to perception of how people like a urban area or not.
- Finding correlation between features (air quality and street i,e Nature and pollution.)
- City beautification using Deep Generative Adversary networks to convert images of urban areas to much beautiful scenes. Using this machine learning algorithm we can convert old looking pictures into better looking areas.

Extracting data from Dense dataset

- Using a Disparity Filter, we are able to convert/trim random dataset to a better tree based dataset. This can be done using thr **Louvain Algorithm**