Towards Enabling Probabilistic Databases for Participatory Sensing

Nguyen Quoc Viet Hung [1], Saket Sathe [2], Duong Chi Thang [1], Karl Aberer[1]

[1] École Polytechnique Fédérale de Lausanne

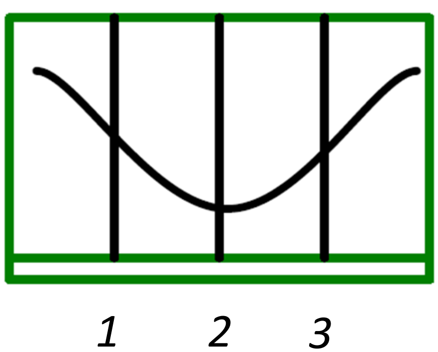
[2] IBM Melbourne Research Laboratory

Participatory sensing

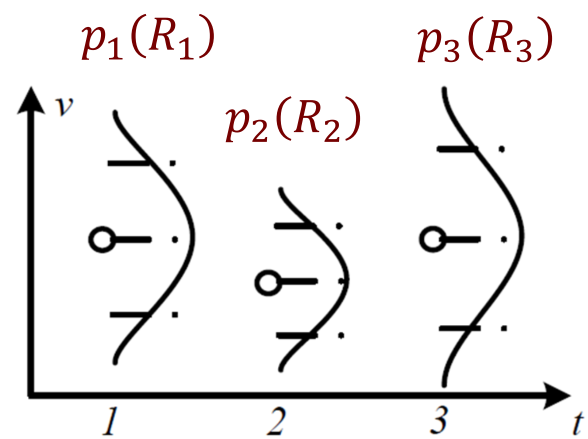
- A concept of communities in which participants proactively report sensory information
  - Sensors might be humans or their mobile devices.
  - Enable harnessing the wisdom of the crowd to collect a huge amount of data for various applications: geo-tagging, environmental monitoring, and public health.
- CrowdSense scenario: http://opensense.epfl.ch
  - Goal: collect high-resolution urban air quality
  - Approach: community-based sensing in which sensors are attached on vehicles and personal devices.
  - Advantages: real-time data gathering, high resolution, and improving data quality

Probabilistic Database

- Traditional database:
  - Sensor data: a series of values $S = \langle r_1, \ldots, r_m \rangle$ at timestamps $1, \ldots, m$
- Probabilistic database:
  - Sensor data: a series of distributions $\mathrm{p}S = \langle p_1(R_1), \ldots, p_m(R_m) \rangle$
    - R_i is a random variable of data value at timestamp i
    - $p_i(R_i)$ reflects a probability distribution, e.g. $N(2,1)$



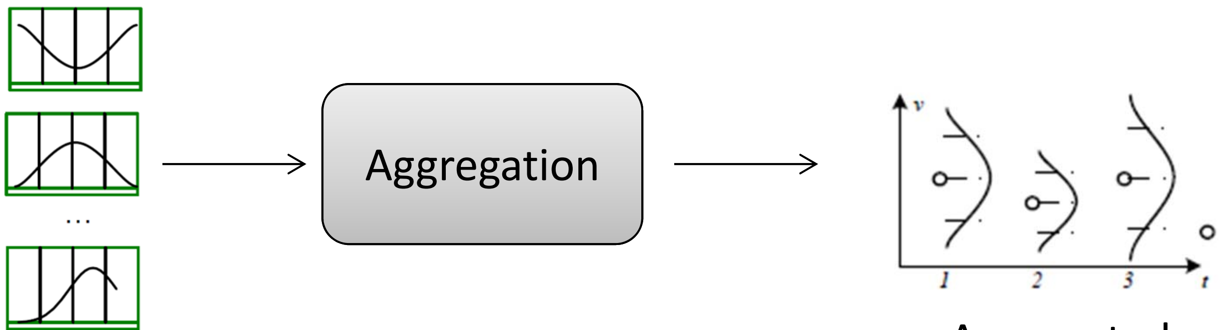
Traditional database



Probabilistic database

Probabilistic Database for Participatory Sensing

❖ Setting:



Multiple sensor data

Aggregated probabilistic data

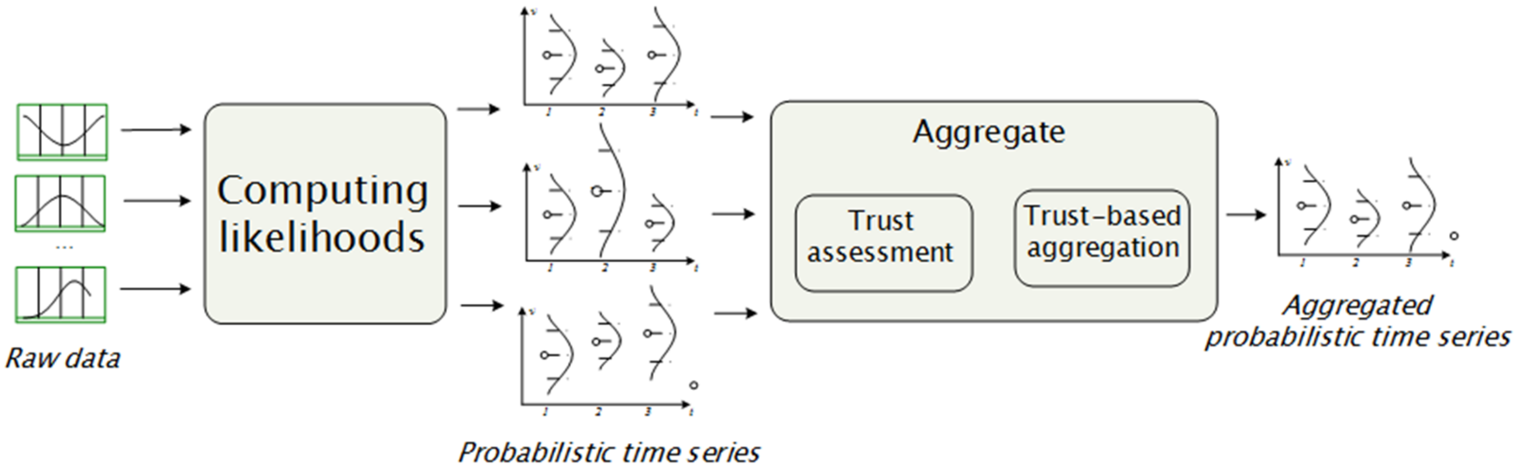
❖ Motivation: data collected from individual participants and their devices are inherently uncertain due to noise factors:

- ➢ Low sensor quality
- ➢ Unstable communication channels

❖ Challenges:

- ➢ Sensor data irregularly depend on time → dynamic computation
  - Temperature changes dramatically around sunrise and sunset, but changes only slightly during the night.
- ➢ Each sensor has different quality → trust model

Approach overview



- Computing likelihoods: compute probabilistic sensor data from raw sensor data
  - Infer probability distribution at each timestamp
- Aggregate: combines all probabilistic sensor data into single probabilistic sensor data
  - Trust assessment: compute trust scores for each sensor and its data
  - Trust-based aggregation: sensors with high trust scores have high impact on aggregated data

Outline

- Model
- Computing likelihoods of sensor data
- Aggregating multiple sensor data
- Experimental results
- Conclusion and future work

Model

- Sensor data: $S_i = \langle r_1^i, \dots, r_m^i \rangle$
  - r_j^i is a reading collected by sensor i at time j
- Probabilistic sensor data: $pS_i = \langle p_1(R_1^i), \dots, p_m(R_m^i) \rangle$
  - R_j^i: random variable
  - $p_j(R_j^i)$: probability density function
- Problem statement: given a set of time series $D = \langle S_1, \dots, S_n \rangle$ of n sensors, compute an aggregated probabilistic sensor data $pG = \langle p_1(G_1), \dots, p_m(G_m) \rangle$.
  - $p_j(G_j)$ is the probabilistic distribution at timestamp j combined from $p_j(R_j^1), \dots, p_j(R_j^n)$.

Computing likelihoods of single sensor data

- Input: a sensor data $S_i = \langle r_1^i, \dots, r_m^i \rangle$
- Output: a probabilistic sensor data $pS_i = \langle p_1(R_1^i), \dots, p_m(R_m^i) \rangle$
- Requirements:
  - R1: The currently value dynamically depends on past values
  - R2: Uncertainty range varies over time
- Solution: model $p_t(R_t)$ by a Gaussian probability distribution $\mathrm{N}(\hat{r}_t, \sigma_t^2)$
  - Estimate the expected value $\hat{r}_t$: using Auto Regressive Moving Average [1] – ARMA (p,q) with p autoregressive terms and q moving-average terms:

$$\hat{r}_t = \delta_0 + \sum_{j=1}^{p} \delta_j r_{i-j} + \sum_{j=1}^{q} \gamma_j \, a_{i-j}$$

satisfy (R1)

  - Compute the variance σ_t^2: using Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) [1] model:

$$\sigma_i^2 = \theta_0 + \sum_{j=1}^{h} \theta_j a_{i-j}^2 + \sum_{j=1}^{k} \varphi_j \sigma_{i-j}^2$$

satisfy (R2)

[*] R. H. Shumway and D. S. Stoffer, Time series analysis and its applications. Springer-Verlag, 2000

Aggregating multiple sensor data

- Input: a set of probabilistic sensor data $pD = \langle pS_1, \dots, pS_n \rangle$
- Output: an aggregated sensor data $G = \langle p_1(G_1), \dots, p_m(G_m) \rangle$
- Method: use trust-based approach
  - Trust assessment
  - Probability aggregation

Outline

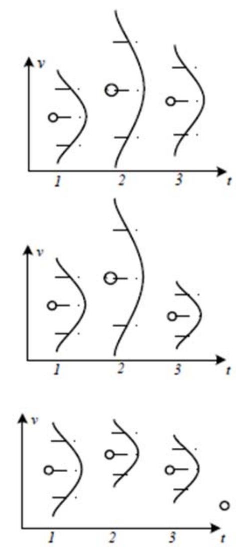
- ~~Model~~
- ~~Computing likelihoods of sensor data~~
- Aggregating multiple sensor data
  - Trustworthiness assessment
  - Probability aggregation
- Experimental results
- Conclusion and future work

Trustworthiness assessment

- Assess the trustworthiness based on two factors:
  - Similarity of distributions: distributions which are similar to each other should have higher trust scores
    - The first distribution of sensor 1 is similar to the first distribution of sensor 2 and sensor 3 → should have high trust score
  - Reliability of sensors: a reliable sensor tends to provide correct information
    - If sensor 1 is reliable, even the third distribution of sensor 1 is different from the others → this distribution should nevertheless have high trust score.

$$\alpha_t^i = \frac{\sum_{j=1..n, j\neq i} \beta_j s_t^{i,j}}{\sum_{j=1..n} \beta_j}$$

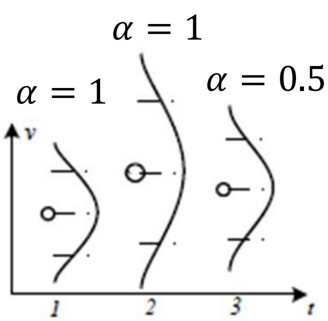
α_t^i: trust score of t-th distribution of sensor i
β_j: reliability of sensor j
$s_t^{i,j}$: similarity between t-th distributions of sensor i and j



Trustworthiness assessment (cont'd)

- Compute the reliability for each sensor:
  - A sensor that provides more correct data tends to be more reliable
  - → Compute the reliability of each sensor based on the trust scores of its data:

$$\beta_i = \frac{\sum_{k=1}^{m} \alpha_k^i}{m}$$



$\longrightarrow$ $\beta \approx 0.83$

Trustworthiness assessment (cont'd)

- There is a mutually reinforcing relationship between sensors and data they provide.
  - Trust score of sensor data:

$$\alpha_t^i = \frac{\sum_{j=1..n, j\neq i} \beta_j s_t^{i,j}}{\sum_{j=1..n} \beta_j}$$

  - Reliability of sensor:

$$\beta_i = \frac{\sum_{k=1}^{m} \alpha_k^i}{m}$$

→ Propose an iterative algorithm to concurrently update the reliability of sensor and trust scores of sensor data until convergence.

Trustworthiness assessment (cont'd)

Algorithm 2 Iterative Algorithm to Compute Trust Scores.

Input: A set of probabilistic time series $p\mathcal{D} = \{pS_1, pS_2, \cdots, pS_n\}$, a termination condition Δ

Output: A set of trust scores α_t^i, β_j.

```
// Initialization
β_1^0 = 0.5;...;β_n^0 = 0.5
q = 1
while Δ do
    for l = 1..m do
        for i = 1..n do
            α_l^{i,q} = (Σ_{j=1..n, j≠i} β_j^{q-1} m_l^{i,j}) / (Σ_{j=1..n} β_j^{q-1})
    for j = 1..n do
        β_j^q = (Σ_{k=1}^m α_k^{j,q}) / m
    q = q + 1
```

$$\alpha_l^{i,q} = \frac{\sum_{j=1..n, j\neq i} \beta_j^{q-1} m_l^{i,j}}{\sum_{j=1..n} \beta_j^{q-1}} \qquad \beta_j^q = \frac{\sum_{k=1}^{m} \alpha_k^{j,q}}{m}$$

- ❖ Initialize trust scores as 0.5 (maximum entropy principle)
- ❖ Iterate until termination condition Δ is satisfied:
  - ➢ Compute reliability of each sensor based on current trust scores
  - ➢ Compute the trust scores based on current reliability.

Probability aggregation

- Compute the final random variable from multiple sensor data weighted by their trust scores:

$$G_t = \frac{\sum_{i=1}^{n} \alpha_i R_t^i}{\sum_{i=1}^{n} \alpha_i}$$

  - where t is timestamp, n is the number of sensors, R is the random variable representing the probability distributions of sensor i at timestamp t.

- Applying moment-generating function [2]:

$$p_t(G_t) = N\left(\sum_{i=1}^{n} \alpha_i \hat{r}_i, \sum_{i=1}^{n} \alpha_i \hat{\sigma}_i^2\right)$$

[2] C. M. Grinstead and J. L. Snell, Introduction to probability. American Mathematical Soc., 1998

Experiment – Dataset and Setting

- Datasets:
  - Real data:
    - Campus: temperature readings collected from a real sensor network deployed on university campus
    - Moving-object: GPS readings collected from 192 moving objects.

TABLE II: Summary of Datasets

	campus-data	*car-data*
Monitored parameter	Temperature	GPS Position
Number of data values	18031	10473
Sensor accuracy	$\pm$ 0.3 deg. C	$\pm$ 10 meters
Sampling interval	2 minutes	1-2 seconds

  - Synthetic data:
    - Fix a true distribution for each timestamp
    - Generate probability distributions from the true distribution by randomly adding differences to the mean value.
- Evaluation measures: computation time, effects of outliers, effects of heterogeneity level

Computation time

- Setting:
  - Vary the length of time series
  - Metrics: computation time
- Observations:
  - Computation time is reasonable w.r.t. data size
  - GARCH model is suitable for online and real-time applications.

TABLE III: Running time of the GARCH method ($\log_2(s)$)

Time series length	campus-data	car-data
30	0.1314	0.1205
60	0.1543	0.1419
90	0.182	0.1653
120	0.2092	0.1874
150	0.2379	0.2104
180	0.2634	0.2333

Effects of outlier

- Setting:
  - Vary the percentage of outliers (i.e. sensors whose data are completely different from normal sensors) in real data.
  - Measure the average trust scores of normal sensors vs. outliers
- Observations:
  - The difference between normal sensors and outliers is clearly separated.
  - Our approach can distinguish between normal sensors and outliers

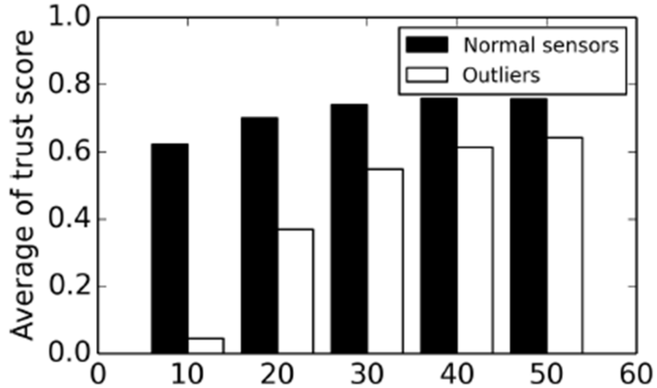
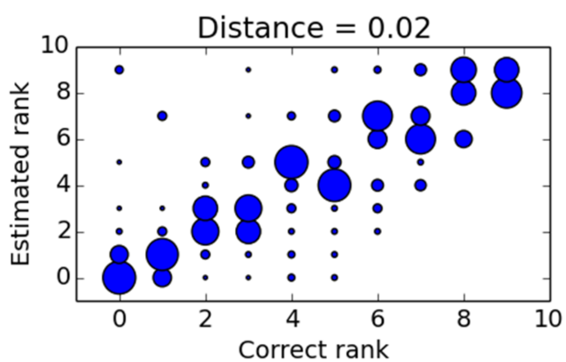
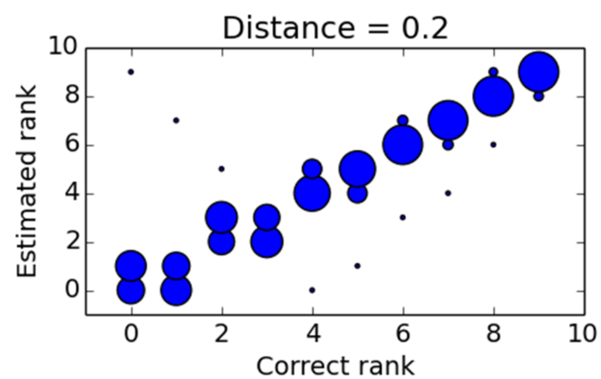


Fig. 2: Accuracy of the algorithm

Effects of heterogeneity level

- Setting:
  - Generate synthetic data by randomly adding some differences (i.e. distance) to the mean value of true distribution
  - Compute the trust scores for the synthetic distributions
  - Metrics: compare the rank of data by distance to true distribution vs. by the trust scores
- Observations:
  - The two ranking orders are similar
  - Our trust scores can reflect the correctness of data





Effect of distance between distributions on the accuracy of computing trust
(size of the circles reflects the number of data with the same ranks)

Conclusions

- We built a systematic model to manage uncertain data from participatory sensing.
- Our probabilistic model captures the dynamic and uncertain nature of sensor data.
- We combined multiple probabilistic data by evaluating the trust scores of data and aggregate based on these trust scores.

Applications

- Information about uncertainty of sensor data can be used as/in:
  - Guidance for data repair:
    - Sensor data is inherently uncertain → need human knowledge to repair data
    - Minimize the repair effort by suggesting the data with most information gain (i.e. the amount of uncertainty reduction of knowing the true value)
  - Adaptive data acquisition in resilience systems:
    - Abnormal readings may reflect unexpected situations: network loss, battery discharge, etc.
    - Detect such situations by analyzing the probability distributions of consecutive timestamps (e.g. sudden changes of variances and mean values).

THANK YOU

Q&A

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE