ThesisProgress

Darina Dementyeva

March 23, 2017

1 08.02.2017 Step 1

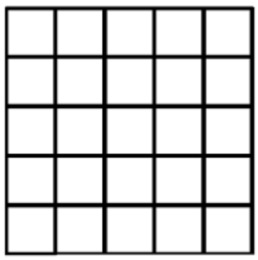

Figure 1: Map of the room.

Let's assume that we have rectangle room with is divided into cells so we have n cells in the row and m cells in the column (for example, as it is shown at Fig. 1). A WiFi access point is placed somewhere in this room (we do not care about its position now).
Task: receiving signal strength x from the user decide in which cell the user is located.
Prior information:

- $(i, j) \forall i = \overline{1, n}; j = \overline{1, m}$ - cells in the room. For further simplicity let's assume that $(i, j) = k$.
- $p(k)$ - the probability of that when user is in the room he will stand at cell k.
- μ_k - the signal strength that can be observed at cell k (at this step we suppose that there is no noise in signal and that we can only observe some particular signal strength in each cell).

Input: x - received signal strength.
Output: $\hat{k}$ - the cell where the user is standing.

Comment: The prior information about cells probability $p(k)$ on this step is gotten by this expression [1]:

$$P_{received}(d) = P_{received}(d_0) - 10 * \alpha * log(\frac{d}{d_0})$$

where $\alpha = 3, d_0 = 4, P_{received}(d_0) = -53$.

Solution:

$$\hat{k} = argmax_k p(x,k) = argmax_k p(k|x)p(x) = argmax_k p(k|x)$$

where $p(k|x)$ can be found as $p(k|x) = \frac{p(x|k)p(k)}{p(x)}$.

2 10.02.2017 Step 2

We should define $p(x|k)$ to find the distribution of signal inside each cell. For beginning, let's take normal distribution $N(\mu_k, \sigma_k^2)$ for each cell k. So, $p(x|k)\tilde{\ } N(\mu_k, \sigma_k^2)$.

Also, for the simplification we will assume that $\mu_k = P_{received}(d)$ and $\forall k$ $\sigma_k^2 = 1$.

Next step: define the way to calculate variance for each cell (obviously, they cannot be equal to 1 and be also equal to each other in common case).

3 12.02.2017 Step 3

Idea - apply particle filtering for $p(x|k)$ approximation.

4 14.02.2017

Particle filter. Key idea: we have some region of start points - our assumptions about start position of the user, defined in some probabilistic approach. Then we have motion activity. We apply this motion to all this points and among them define that have the biggest weights. And only these points will stay in the model. After that just continue to that moment when we will be able exactly define user's position.[5]

In context of the idea current issues: how define start points; what about the motion - how to define or predict it; generally how to apply it to the task.

Other issues: is it the best model? Can be other models found that will be more suitable?

5 15.02.2017

According to "Prioritized Online Positioning Algorithm" described in [3]:

1. Define feasible region.
2. Find possible locations in the feasible region.
3. Apply particle filter to reduce the number of particles on step $t + 1$.

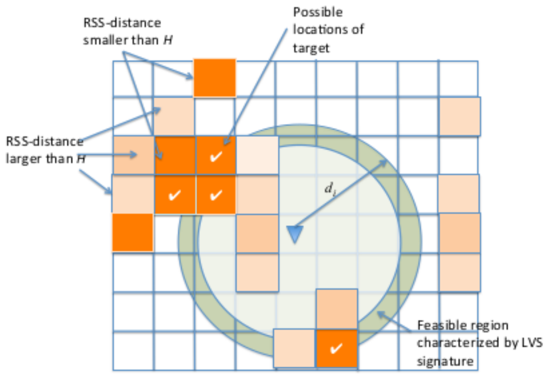

Figure 2: The idea of "Prioritized Online Positioning Algorithm" [3]

Issues:

- Feasible region is defined from some target v. How to define the target? How define start points?
- What is the purpose of feasible region? Does it correlate with motion?
- Test radio-map - how to get it? What information it should bring?

6 16.02.2017 Step 3. Continue.

- t_0: suppose we start from the single user's position $(x(t_0); y(t_0))$ somewhere in the room. Also we have some value of the velocity of the user v_0.
- t_1: after some Δt we get our $RSSI(t_1)$. According to this RSSI we can calculate the probability of user's presence in cell k having such RSSI - $p(k|RSSI(t_1))$.
  Further, we can make an assumption about acceleration of the user - for example, we have $\overrightarrow{a_1} N(0, \sigma)$ - the model of person's motion. We

can make k hypothesis about the supposing value of the acceleration and generate k samples. For each $\overrightarrow{a_1(i)}$ we calculate Δs. Also, we calculate the probability of each hypothesis $p(k|\overrightarrow{a_1(i)})$.

- Using all previous information we calculate the new possible position of the user $(x_i(t_1), y_i(t_1))$. Among them we can choose only with the highest probabilities (particle filtering).
- Apply these three steps further until the error will not change.

Next:

- Look throw the code from [3].
- Python libs for visualization.
- Look for some model of person's motion.

7 22.02.2017

- The calibration data is then often used to estimate the parameters of locally Gaussian models, which have proven to be very effective for localization[2].

- How to distinguish fingerprint twins - generate and use user-motiion database for the location [4]. (Can be useful for further development.)

- For practical positioning systems with limited computational power, e.g., slow processors, particle filter is the best choice. If the density functions are transfered between different nodes, Fourier-based filter should be used because fewer components require smaller bandwidth and communication time [6].

8 27.02.2017.Location Tracking Model.Version 1.

As basic information (at time t_0) we have:

1. Signal propagation map: in each cell $\{c_{ij}\}_{i=\overline{1,n};j=\overline{1,m}}$ signal has propagation with distribution $N(\mu_{ij}, \sigma_{ij}^2)$. (Suppose we work in discrete space.)
2. Start point $s^{t_0} = (x^0; y^0)$.
3. Start velocity v^0.

After some Δt at time t_1 we got $RSSI^{t_1}$. After that:

1. We can calculate for each cell c_{ij} conditional probability $p(c_{ij}|RSSI^{t_1})$ - the probability of staying at cell c_{ij} under condition of getting such signal strength.

2. **Person's motion model**. We make assumption about person's acceleration to predict his possible position. We suppose that acceleration has distribution $N(0; \sigma_a^2)$. Generate K samples $\vec{a_k}$, $k = \overline{1, K}$.

3. Calculate new possible position according to the generated accelerations: $s_k^{t_1} = s^{t_0} + v^0 \Delta t + \frac{a_k \Delta t^2}{2}$.

4. For every cell c_{ij} calculate: $p(RSSI^{t_1}, s_{ij}^{t_1}) = p(RSSI^{t_1}) p(s_{ij}^{t_1})$.

5. Leave particles with max probability with some threshold.

6. Continue.

For this approach put: $n = 100$, $m = 100$, $v^0 = 0$, $K = 100$.
Points to think about / rebuild in model:

- Person's motion model. What parameters of normal distribution should we put? What other description of this model can be (other distribution, etc.)?

- Discrete space or continuous space?

- At point 4 the probability can be decomposed not in such way - the random variables can be dependent.

- Further model improvements - person's rotation element, dependence of current point of time of not one but some previous ones, ...

9 Current approaches. 15.03.2017

For all:
Input: sequence of RSSI measurements: $RSSI_1 ... RSSI_n$.
Output: path estimation $(x_1, y_1) ... (x_n, y_n)$.

1. Simple Bayes approach
   Each step does not depend on any previous step and motion of the user. We get as input RSSI and at each appoach the position as:

$$(x, y) = argmax_{x,y} p(RSSI_i | (x, y))$$

2. Bayes approach with motion
   Here we also add a model of user's motion: $awithN(0,1)$. At each step we generate K samples of possible a and move previous position. Again does not depend on previous step. Position and acceleration will be chosen as:
   $$(x,y,a) = argmax_{x,y,a_k} p(RSSI_i|(x,y))p(a_k)$$
3. Viterbi algorithm
   Parameters: Let's suppose $s = (x, y)$. So, there should be provided information: $p(s_i|s_{i-1}...s_{i-k})$, k = 1, ...

References

[1] Frédéric Evennou and Francois Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing*, 2006:164–164, 2006.

[2] Brian Ferris, Dieter Fox, and Neil D Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007.

[3] Xiongfei Geng, Yongcai Wang, Haoran Feng, and Zhoufeng Chen. Hybrid radio-map for noise tolerant wireless indoor localization. In *Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on*, pages 233–238. IEEE, 2014.

[4] Wei Sun, Junliang Liu, Chenshu Wu, Zheng Yang, Xinglin Zhang, and Yunhao Liu. Moloc: On distinguishing fingerprint twins. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pages 226–235. IEEE, 2013.

[5] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[6] Hui Wang, Andrei Szabo, Joachim Bamberger, Dietrich Brunn, and Uwe D Hanebeck. Performance comparison of nonlinear filters for indoor wlan positioning. In *Information Fusion, 2008 11th International Conference on*, pages 1–7. IEEE, 2008.