Nonparametric Filter

Quan Nguyen

November 16, 2015

Outline

1. Hidden Markov Model
2. State estimation
3. Bayes filters
4. Histogram filter
5. Binary filter with static state
6. Particle filter
7. Summary
8. References

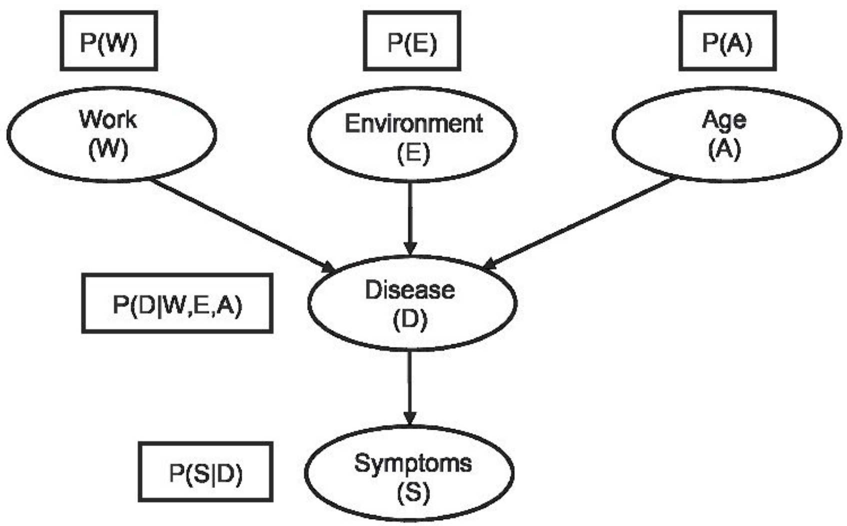
Bayesian Network

- **Graphical model of conditional probabilistic relation**
- **Directed acyclic graph (DAG)**

$$G = (V, E)$$

V: set of random variables

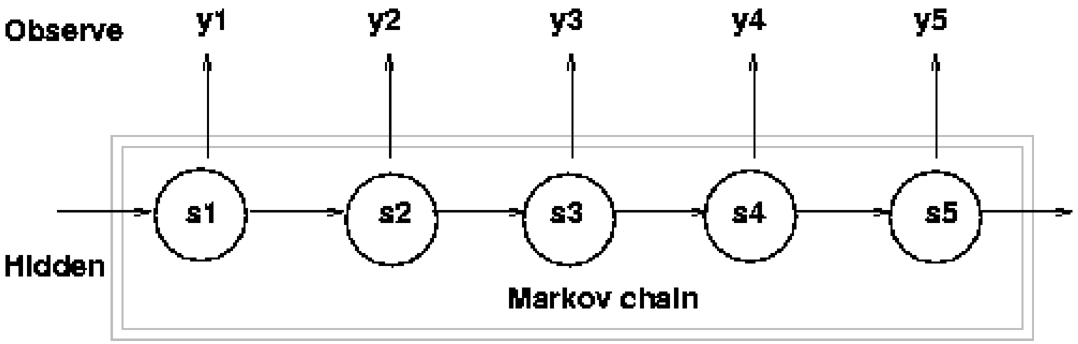
E: set of conditional dependencies



http://www.intechopen.com/books/current-topics-in-public-health/from-creativity-to-artificial-neural-networks-problem-solving-methodologies-in-hospitals

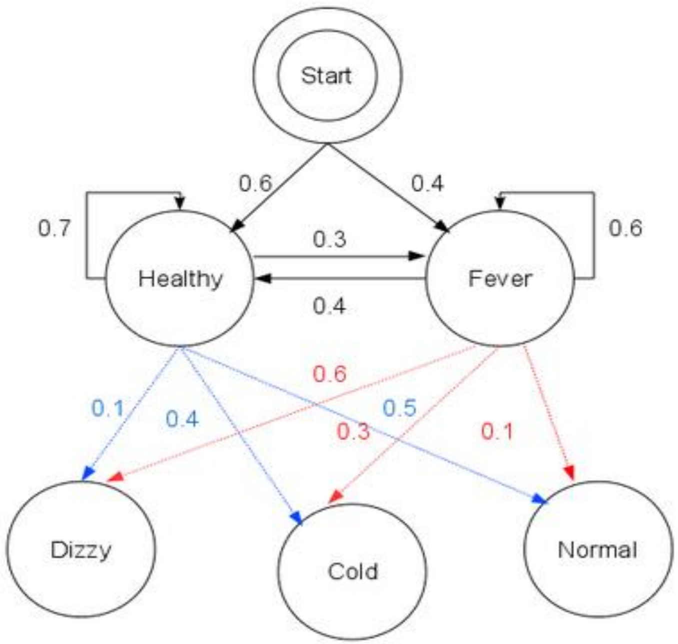
Hidden Markov Model

- Particular kind of Bayesian Network
- Modelling time series data



http://sites.stat.psu.edu/~jiali/hmm.html

Hidden Markov Model



https://en.wikipedia.org/wiki/Viterbi_algorithm#Example

Hidden Markov Model

Observing a patient for 3 days:

+ Day 1: Cold

+ Day 2: Normal

+ Day 3: Dizzy

Question:

1) Most likely sequence of health condition of the patient in last 3 days ?
2) Most likely health condition of the patient in the 4^{th} day ?

State space

- Quantities that cannot be directly observed but can be inferred from sensor data
- Examples: position and direction of robot in a room
- Notation:

$$X = \{x_1, x_2, \dots x_t\}$$

$$P(X = x_t): probabilty\ of\ sate\ equals\ to\ x\ at\ time\ t$$

Measurement (Observation)

- Environment data provided by robot sensor
- Examples: distance to ground, camera images
- Notation:

$Z = \{z_1, z_2, \dots, z_t\}$

$P(Z = z_t): probabilty\ of\ measurement\ equals\ to\ z\ at\ time\ t$

Control data

- Information about the change of state in the environment
- Examples: velocity of robot, temperature of a room, an action of robot on environment objects
- Notation:

$$U = \{u_1, u_2, \dots, u_t\}$$

$$P(U = u_t): probabilty\ of\ measurement\ equals\ to\ z\ at\ time\ t$$

Probabilistic Generative Laws

- State can be constructed on all past states, measurements and controls:

$$P(X = x_t) = P(X = x_t | x_{0:t-1}, z_{0:t-1}, u_{0:t-1})$$

- Markov assumption:

$$P(X = x_t) = P(X = x_t | x_{t-1}, u_t)$$
$$P(Z = z_t) = P(Z = z_t | x_t)$$

Belief distribution

- Belief:

- Internal knowledge of the robot about the true state
- Represent probability to each possible true sate
- Notation:

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

- Prediction:

$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t})$$

- Correction:

$$bel(x_t) = \mathrm{F}(\overline{bel}(x_t))$$

Bayes Filter algorithm (continuous case)

1: $Func\_continous\_Bayes\_filter\ (bel(x_{t-1}), u_t, z_t)$

2: $\quad for\ all\ x_t\ do$

3: $\qquad \overline{bel}(x_t) = \int p(\ x_t | u_t, x_{t-1}) bel(x_{t-1}) dx$

4: $\qquad bel(x_t) = normalizer\ * p(z_t | x_t)\ \overline{bel}(x_t)$

5: $\quad end$

6: $\quad return\ bel(x_t)$

Bayes Filters algorithm (discrete case)

$Func\_discrete\_Bayes\_filter(p_{k,t-1}, u_t, z_t)$

$\quad for\ all\ k\ do$

$\quad\quad \overline{p_{k,t}} = \sum p(\,x_t\,|\,u_t, X_{t-1} = x_i)p_{i,t-1}$

$\quad\quad p_{k,t} = normalizer\ * p(z_t\,|\,x_t)\overline{p_{k,t}}$

$\quad end$

$\quad return\,\overline{p_{k,t}}$

Histogram Filter

- Discrete Bayes filter estimation for *continuous* state spaces

- State space decomposition:

- $Range(X_t) = \{x_{1,t} \cup x_{2,t} \cup \dots x_{M,t}\}$
- $For\ every\ i\ \neq k: x_{i,t} \cap x_{k,t} = \emptyset$

- In each region the posterior is a piecewise constant density:
- $For\ every\ state\ x_t\ belongs\ to\ k^{th}\ region$:

$$p(x_t) = \frac{p_{k,t}}{|x_{k\ t}|}$$

Histogram filter

- **Problem:** prior information is defined for individual states, not for region !

- Refer to line 3, 4 of discrete Bayes filter algorithm

- **Solution:** approximating density of a region by a *representative* state of that region.

$$\widehat{x_{k,t}} = \frac{\int_{x_{k,t}} x_t \, dx_t}{|x_{k,t}|}$$

Histogram filter

- Approximation of density values for regions:

$$p(z_t|x_{k,t}) \approx p(z_t|\widehat{x_{k,t}})$$

$$p(x_{k,t}|u_t, x_{i,t-1}) \approx normalizer \ * \ p(\widehat{x_{k,t}}|u_t, \widehat{x_{i,t-1}})$$

- Precondition: all regions must have the same size.

- Now discrete Bayes filter algorithm is applicable !

Binary Bayes filter with Static State

- Belief is a function of measurement:

$$bel_t(x) = p(x|z_{1:t}, u_{1:t}) = p(x|z_{1:t})$$

- **General algorithm:**

1: $Func\_binary\_Bayes\_filter(l_{t-1}, z_t)$

2: $$l_t = l_{t-1} + \log\frac{p(x|z_t)}{1 - p(x|z_t)} - \log\frac{p(x)}{1-p(x)}$$

3: $return\ l_t$

- **Log odds ratio**

$$l(x) = \log \frac{p(x)}{1 - p(x)}$$

- Avoids truncation problems when probabilities close to 0 or 1

- **Inverse measurement model:**

- Reduce complexity by using probability of state given measurement data

- Example: infer state of a door in an image is much easier than infer an image from all other images of a close/open door.

Example of Binary filter: Occupancy grid mapping

- **Estimate (generate) map from (noisy) sensor measurement data and robot position**
- **General algorithm:**

$$\boldsymbol{p(Map = M|z_{1:t}, x_{1:t}) = \prod_{c} p(Cell = c\ is\ occupied|z_{1:t}, x_{1:t})}$$

$\boldsymbol{p(Cell = c\ is\ occupied|z_{1:t}, x_{1:t})}$ **is a binary estimation problem**

Particle filter algorithm

- **Represent the posterior density by a set of weighted random particles**
- **General algorithm:**

$Func\_Particle\_filter(X_{t-1}, u_t, z_t)$

$\quad \overline{X_t} = X_t = \emptyset$

$\quad for\ i = 1\ to\ M\ do$

$\qquad sample\ x_t^i \sim p(x_t | x_{t-1}^i)$

$\qquad w_t^i = p(z_t\ | x_t^i)$

$\qquad \overline{X_t} = X_t + (x_t^i, w_t^i)$

$\quad endfor$

$\quad for\ i = 1\ to\ M\ do$

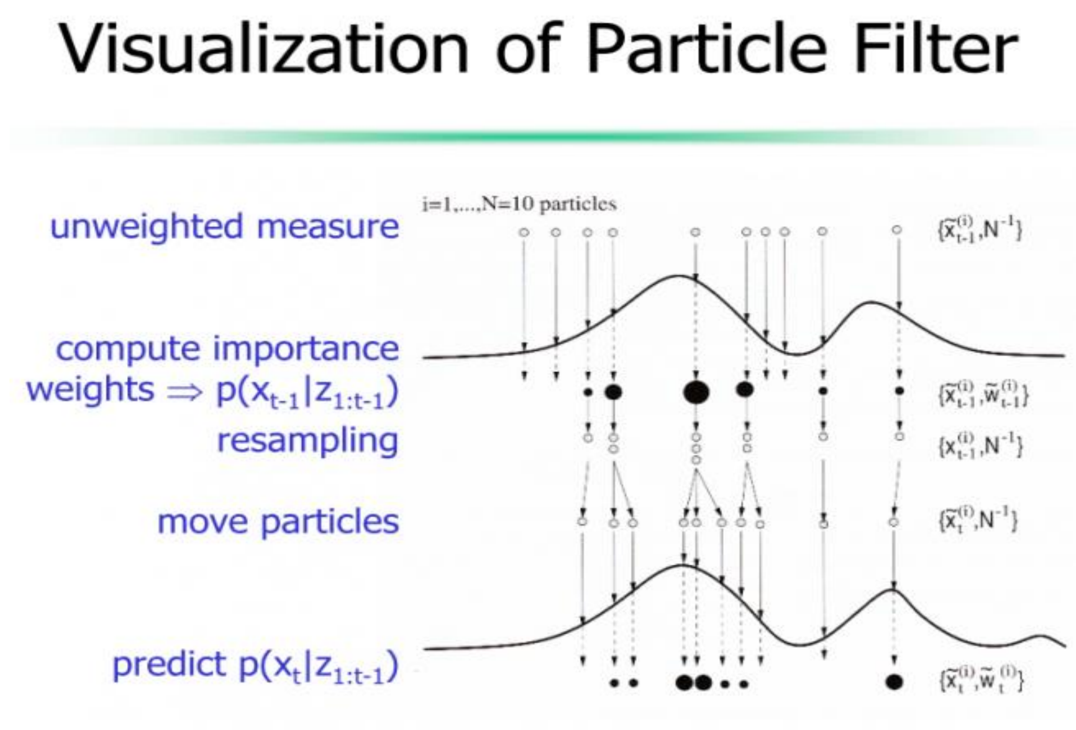
$\qquad draw\ i\ with\ probability\ \propto w_t^i$

$\qquad add\ x_t^i\ to\ X_t$

$\quad endfor$

$\quad return\ \ X_t$

Particle filter algorithm



http://www.juergenwiki.de/work/wiki/doku.php?id=public:particle_filter

Properties of Particle filter algorithm

- **Degree of freedom:**

- Because of normalization we lost one degree of freedom:

$$\deg = M \ - 1$$

- **Identical particles after resampling phase**:

- Resampling with probability proportional to weight: after every iteration we failed to draw one or more state sample

Properties of Particle filter algorithm

- **Deterministic sensor:**

- Sensor with noise-free range: measurement data is zero for most of state !

$\Rightarrow$All weights become zero.

- **Particle deprivation problem:**

- Resampling can wipe out all particles near the true state

$\Rightarrow$incorrect states have larger weight !

Application of Particle filter

- **Tracking the state of a dynamic system modeled by a Bayesian Network:** Robot localization, SLAM, robot fault diagnosis.

- **Image segmentation:** by generating a large number of particles and gradually focus on particle with desired properties

$\Rightarrow$**Image processing, Medial image analysis**

Summary

- Nonparametric filters represent posterior state as a function of previous poster state
- Nonparametric filters does not rely on a fixed functional form of posterior
- Histogram filter and Particle filter represent state space and posterior as a finite set of data
- There is usually a trade-off between efficiency and level of detail of data

References

- Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.
- Kaijen Hsiao, Henry de Plinval-Salgues and Jason Miller. Particle filters and their applications, *Cognitive Robotics*, April 11, 2005.
- Dwyer, P. S.. (1967). Some Applications of Matrix Derivatives in Multivariate Analysis. *Journal of the American Statistical Association*, *62*(318), 607–625. http://doi.org/10.2307/2283988
- Pietro Manzi, Paolo, Barini. Current Topics in Public Health. 2013-05-15