Sensor Tasking and Control

Leonidas Guibas
Stanford University

CS321

[ZG, Chapters 2, 5]
Sensor systems are about sensing, after all ...
Node Functions
Node Roles in a WSN

- Nodes can play **multiple roles** in a sensor network:
  - sensing
  - routing
  - aggregating information
  - monitoring
  - sleeping

- When tasking sensors, we must be aware of all the potential roles these nodes can play.
System State
Continuous and Discrete Variables

The quantities that we may want to estimate using a sensor network can be either continuous or discrete.

Examples of continuous variables include:
- a vehicle’s position and velocity
- the temperature in a certain location

Examples of discrete variables include:
- the presence or absence of vehicles in a certain area
- the number of peaks in a temperature field
Uncertainty in Sensor Data

Quantities measured by sensors always contain errors and have associated uncertainty – thus they are best described by PDFs.
- interference from other signal sources in the environment
- systematic sensor bias(es)
- measurement noise

The quantities we are interested in may differ from the ones we can measure – they can only indirectly be inferred from sensor data. These hidden variables are also best described by PDFs.
Information Sources

- Prior information, together with knowledge of the physical laws for the system of interest
- Current sensor measurements

Prior knowledge (the prior) + Current measurements → Current knowledge (the posterior)
(Review) Excursion into Sequential Estimation

[From Thrun, Brugard, and Fox]
Recursive State Estimation

State $x$:
- external parameters describing the environment that are relevant to the sensing problem at hand (say vehicle locations in a tracking problem)
- internal sensor settings (say the direction a pan/tilt camera is aiming)

While internal state may be readily available to a node, external state is typically hidden — it cannot be directly observed but only indirectly estimated.

States may only be known probabilistically.
Environmental Interaction

Control $u$: a sensor node can change its internal parameters to improve its sensing abilities

Observation $z$: a sensor node can take various measurements of the environment

Discrete Time $t$: $0, 1, 2, 3, ...$

$x_t, u_t, z_t$  \[ z_{t_1:t_2} \equiv z_{t_1}, z_{t_1+1}, z_{t_1+2}, z_{t_1+3}, \ldots, z_{t_2} \]
Bayes Rule

\[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} \]

(discrete)

\[ = \frac{\sum_{x'} p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')} \]

(continuous)

\[ = \frac{\int_{x'} p(y|x)p(x) p(x') dx'}{\int_{x'} p(y|x')p(x') dx'} \]

\[ p(x|z) = \eta p(z|x)p(x) \]

probability of state \( x \), given measurement \( z \)

probability of measurement \( z \), given state \( x \) (the environment/sensor model)
Probabilistic Generative Laws

State $x_t$ is generated stochastically by

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

Markovian assumption (state completeness)

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

$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t)$$
The Bayes Filter

Belief distributions

\[ b(x_t) = p(x_t \mid z_{1:t}, u_{1:t}), \quad \overline{b}(x_t) = p(x_t \mid z_{1:t-1}, u_{1:t}) \]

Algorithm Bayes_Filter \((b(x_{t-1}), u_t, z_t)\)

for all \(x_t\) do

\[ \overline{b}(x_t) = \int p(x_t \mid u_t, x_{t-1}) b(x_{t-1}) \, dx \] [prediction]

\[ b(x_t) = \eta \, p(z_t \mid x_t) \overline{b}(x_t) \] [observation]

endfor

return \(b(x_t)\)
Gaussian Filters

Beliefs are represented by multivariate Gaussian distributions

\[ p(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) \]

Here \( \mu \) is the mean of the state, and \( \Sigma \) its covariance

Appropriate for unimodal distributions
The Kalman Filter

Next state probability must be a **linear function**, with added Gaussian noise [result still Gaussian]

\[
x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t
\]

Gaussian noise with zero mean and covariance \(R_t\)

\[
p(x_t|u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} (x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\}
\]

Measurement probability must also be **linear** in it arguments, with added Gaussian noise

\[
z_t = C_t x_t + \delta_t
\]

Gaussian noise with zero mean and covariance \(Q_t\)

\[
p(z_t|x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} (z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right\}
\]
Kalman Filter Algorithm

**Algorithm Kalman_Filter** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\)

\[
\begin{align*}
\bar{\mu}_t &= A_t \mu_{t-1} + B_t u_t \\
\bar{\Sigma}_t &= A_t \Sigma_{t-1} A_t^T + R_t
\end{align*}
\]

*belief predicted by system dynamics*

\[
K_t = \frac{\bar{\Sigma}_t}{\bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q)^{-1}} \quad \text{(the Kalman gain)}
\]

\[
\hat{\mu}_t = \mu_t + K_t (z_t - C_t \bar{\mu}_t)
\]

*belief updated using measurements*

\[
\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t
\]
Non-Parametric Filters

- Parametric filters parametrize a distribution by a fixed number of parameters (mean and covariance in the Gaussian case)

- Non-parametric filters are discrete approximations to continuous distributions, using variable size representations
  - essential for capturing more complex distributions
  - do not require prior knowledge of the distribution shape
The Particle Filter

Samples from a Gaussian, passed through a non-linear function
The Particle Filter Algorithm

Algorithm Particle_Filter \((X_{t-1}, u_t, z_t)\)

\[
X_t = \overline{X}_t = \emptyset
\]

for \(m = 1\) to \(M\) do

sample \(x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})\)

\[w_t^{[m]} = p(z_t | x_t^{[m]}); \quad \overline{X}_t = \overline{X}_t \cup \langle x_t^{[m]}, w_t^{[m]} \rangle\]

endfor

for \(m = 1\) to \(M\) do

draw \(i\) with probability proportional to \(w_t^{[i]}\)

add \(x_t^{[i]}\) to \(X_t\)

return \(X_t\)

number of particles of unit weight
stochastic propagation
importance weights
resampling, or importance sampling
Sensor Models
Sensor Models

To be able to develop protocols and algorithms for sensor networks, we need **sensor models**

- To perform **Bayesian inversion**, we need to compute the value of a measurement, given an assumed state
- We need to assess the **importance** of a sensor measurement?

Our state PDF representations must allow expression of the state ambiguities inherent in the sensor data

Need also to be aware of the effect of sensor characteristics on system performance
- cost, size, sensitivity, resolution, response time, energy use, calibration and installation ease, etc.
Sensor Characteristics

Accuracy: The difference between the measured value and the actual value, reported as a maximum.

Precision: The difference between the instrument’s reported values during repeated measurements of the same quantity.

Resolution: The smallest increment of change in the measured value that can be determined from the instruments read out. Usually similar or smaller than precision.

Sensitivity: The change in the output of an instrument per unit change in the input.
Acoustic Amplitude Sensors

- Lossless isotropic propagation from a point source
  
  \[ z = \frac{a}{\| x - \zeta \|} + w \]

- Say \( w \) is Gaussian \( \mathcal{N}(0,\sigma) \), and \( a \) uniform in \([a_{lo}, a_{hi}]\)

\[
p(z \mid x) = \frac{r}{\Delta_a} \left[ \Phi \left( \frac{a_{hi} - rz}{r\sigma} \right) - \Phi \left( \frac{a_{lo} - rz}{r\sigma} \right) \right]
\]

\[ \Delta_a = a_{hi} - a_{lo} \]

\[ r = \| z - \zeta \| \]

error function
DoA Sensors

- Beam-forming with microphone arrays

\[ g_m(t) = s_0(t - t_m) + w_m(t) \]

- Far field assumption

\[ t_m = \frac{d}{c} \sin \theta \]

sound propagation speed
Direction estimates are only accurate within a certain range of distances from the sensor.

\[ p(z \mid \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z - \theta)^2}{2\sigma^2}\right) \]

PDF for beamforming sensor
Performance Comparison and Metrics for Detection/Localization

- Detectability
- Accuracy
- Scalability
- Survivability
- Resource usage

Receiver Operator Characteristic (ROC) curve
## System Performance Metrics and Parameters

<table>
<thead>
<tr>
<th>Performance Metrics</th>
<th>Detection Quality</th>
<th>Spatial Resolution</th>
<th>Latency</th>
<th>Robustness to Failure</th>
<th>Power Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>System, Application Parameters</td>
<td>SNR, Distractors</td>
<td>Target, Node Spacing</td>
<td>Link Delay, Target #, Query #</td>
<td>Node Loss</td>
<td>Active/Sleep Ratio, Sleep Efficiency</td>
</tr>
</tbody>
</table>

- **Detection Quality**
- **Spatial Resolution**
- **Latency**
- **Robustness to Failure**
- **Power Efficiency**
- **SNR, Distractors**
- **Target, Node Spacing**
- **Link Delay, Target #, Query #**
- **Node Loss**
- **Active/Sleep Ratio, Sleep Efficiency**
Value of Information
What is the Value (Importance) of a Sensor Reading?

- Sensing operations cost energy – for both sensing and communication
- To properly task sensors, we need to estimate the cost of sensing ... 
- But also the value of information to be potentially obtained
  - but without making the measurement first!
  - information value must be based on prior beliefs and known sensor characteristics
  - it always carries uncertainty
Information Theoretic Metrics: Conditional Entropy & Mutual Information

- Entropy is a measure of uncertainty.
- Let $H(x)$ be the entropy of $x$ based on previous observations.
- Let $y$ be a new observation on $x$.
- We can measure the uncertainty of $x$ after observing $y$ by using the conditional entropy which is defined as:
  \[ H(x|y) = H(x, y) - H(y) \]
  with the property 0: $H(x|y) = H(x)$

- Here, $H(x,y)$ is the joint entropy of observations $x$ and $y$.
- The conditional entropy $H(x|y)$ represents the amount of uncertainty remaining about $x$ after $y$ has been observed.
- If the uncertainty is reduced, then there is information gained by observing $y$.
- Therefore, we can measure the relevance of $y$ by using conditional entropy.
- Another measure that is related to conditional entropy is mutual information $I(x,y)$ which is a measure of uncertainty that is resolved by observing $y$ and is defined:
  \[ I(x, y) = H(x) - H(x|y) = H(x) + H(y) - H(x,y) \]
Model-Driven Sensor Tasking:

[BBQ, A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, W. Hong ‘04]

Strengths of model-based data acquisition

Observe fewer attributes
Exploit correlations
Reuse information between queries
Directly deal with missing data
Answer more complex (probabilistic) queries
An Example Problem
Distributed State Estimation
An Example Sensor Network Problem

- One or more targets are moving through a sensor field
- The field contains networked acoustic amplitude and bearing (DoA) sensors
- Queries requesting information about object tracks may be injected at any node of the network
- Queries may be about reporting all objects detected, or focused on only a subset of the objects.
The Tracking Scenario

Constraints:
- Node power reserves
- RF path loss
- Packet loss
- Initialization cost
- ...

1. **Discovery**: Node $a$ detects the target and initializes tracking
2. **Query processing**: User query $Q$ enters the net and is routed towards regions of interest
3. **Collaborative Processing**: Node $a$ estimates target location, with help from neighboring nodes
4. **Communication protocol**: Node $a$ may hand data off to node $b$, $b$ to $c$, ...
5. **Reporting**: Node $d$ or $f$ summarizes track data and send it back to the querying node

What if there are other (possibly) interfering targets?

What if there are obstacles?

- Bearing sensors (eg. PIR, beamformer)
- Range sensors (eg. Omni-microphone)
Query must be routed to the node best able to answer it.
Tracking Scenario
Tracking Scenario
Tracking Scenario
Tracking Scenario

Sensor $a$ senses the location of the target and chooses the next best sensor.
Tracking Scenario

Sensor $b$ does the same
Tracking Scenario
Sensor $d$ both chooses the next best sensor and also sends a reply to the query node.
Tracking Scenario
Tracking Scenario

Sensor $f$ loses the target and sends the final response back to the query node.
Key Issues

- How is a target detected? How do we suppress multiple simultaneous discoveries?
- How do nodes form collaboration groups to better jointly track the target(s)? How do these groups evolve as the targets move?
- How are different targets differentiated and their identities maintained?
- What information needs to be communicated to allow collaborative information processing within each group, as well as the maintenance of these groups under target motion?
- How are queries routed towards the region of interest?
- How are results from multiple parts of the network accumulated and reported?
Formulation

- Discrete time $t = 0, 1, 2 ...$
- $K$ sensors; $\lambda_i^t$ characteristics of the $i$-th sensor at time $t$
- $N$ targets; $x_i^t$ state of target $i$ at time $t$; $x^t$ is the collective state of all the targets; state of a target is its position in the $x$-$y$ plane
- Measurement of sensor $i$ at time $t$ is $z_i^t$; collective measurements from all sensors together are $z^t$

$z_i^t$ and $z^t$ denote the respective measurement histories over time
Sensing Model

Back to estimation theory

\[ z_i^t = h(x^t, \lambda_i^t) \]

\[ z_i^t = H_i^t(\lambda_i^t) x^t + w_i^t \]

Assume time-invariant sensor characteristics

Use only acoustic amplitude sensors

\[ \lambda_i = \begin{bmatrix} \zeta_i \sigma_i^2 \end{bmatrix}^T , \quad z_i = \frac{a_i}{\| x_i - \zeta_i \|^{\alpha/2}} + w_i \]
Collaborative Single Target Localization

- Three distance measurements are needed to localize a point in the plane (because of ambiguities)
- Linearization of quadratic distance equations

\[ \| x \|^2 + \| \zeta_i \|^2 - 2 x^T \zeta_i = \frac{a_i}{z_i}, \quad i = 1, 2, 3, \ldots \]

\[-2 (\zeta_i - \zeta_1)^T x = a_i \left( \frac{1}{z_i} - \frac{1}{z_1} \right) - (\| \zeta_i \|^2 - \| \zeta_1 \|^2) \]

\[ c_i^T x = d_i \]

subtract equation 1 from equation \( i \)
Least Squares Estimation

Since the state $x$ has two components, three measurements are needed to obtain two equations.

More measurements lead to an over-determined system -- which can yield more robust estimates via standard least squares techniques.

$$ Cx = d \quad (K - 1) \times 2, \ 2 \times 1 = (K - 1) \times 1 $$

$$ x = \left[ \left( C^T C \right)^{-1} \ C^T \right] \ d \quad \text{Least-squares solution} $$
Bayesian State Estimation

Initial Distribution $p(x_0)$

Prior = Posterior at time $k-1$

Dynamic Model

$$p(x_k | z_k) \propto p(z_k | x_k) \cdot p(x_k | x_{k-1}) p(x_{k-1} | z_{k-1}) dx_{k-1}$$

Posterior at time $k$

Observation at time $k$

Prediction at time $k$
Distributed State Estimation

- Observations $z$ are naturally distributed among the sensors that make them.
- But which node(s) should hold the state $x$? Even in the single target case ($N=1$), this is not clear...

1. All nodes hold the state.
2. A single fixed node holds the state.
3. A variable node holds the state (the leader).
Many, Many Questions and Trade-Offs

- How are leader nodes to be initially selected, and how are they handed off?
- What if a leader node fails?
- How should the distribution of the target state (= position) be represented? parametrically (Gaussian) or non-parametrically (particles)?

Best-possible state estimation, under constraints → Communication, Delay, Power
IDSQ: Information-Driven Sensor Querying
IDSQ: Information-Driven Sensor Querying

Localize a target using multiple acoustic amplitude sensors

**Challenge**
- Select next sensor to query to maximize information return while minimizing latency & bandwidth consumption

**Ideas**
- Use information utility measures
  - E.g. Mahalanobis distance, volume of error covariance ellipsoid
- Incrementally query and combine sensor data
Tracking Multiple Objects

New issues arise when tracking multiple interacting targets

- The dimensionality of the state space increases — this can cause an exponential increase in complexity (e.g., in a particle representation)

The distribution of state representation becomes more challenging

- One leader per target?
- What if targets come near and they mix (data association problem)?
State Space Decomposition

- For well-separated targets, we can factorize the joint state space of the $N$ targets into its marginals.
- Such a factorization is not possible when targets pass near each other.
- Another factorization is between target locations and identities,
  - the former require frequent local communication,
  - the latter less frequent global communication.
Data Association

- Data association methods attribute specific measurements to specific targets, before applying estimation techniques.
  - Even when there is no signal mixing, the space of possible associations is exponential: $N!/K!$ possible associations ($N = \#$ of targets, $K = \#$ of sensors).
  - Signal mixing makes this even worse: $2^{NK}$ possible associations.

- Traditional data association methods are designed for centralized settings.
  - Multiple Hypothesis Tracking (MHT)
  - Joint Probabilistic Data Association (JPDA)

- Network delays may cause measurements to arrive out of order in the nodes where the corresponding state is being held, complicating sequential estimation.
Conclusion

An appropriate state representation is crucial
- Different representations may be needed at different times
- The distribution of state raises many challenges

Information utility:
- Directs sensing to find more valuable information
- Balances cost of power consumption and benefit of information acquisition
The End