Sensors networks

Introduction & detection

Manuel A. Vázquez
Joaquín Míguez
Jose Miguel Leiva

January 30, 2024

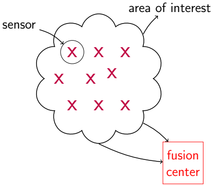
Index

1. Introduction
   - Overview
2. Structure/Topology
   - Types
3. Routing
   - Djikstra
4. Detection
   - Local tests
   - Global test
   - Example
   - Neyman-Pearson

Index

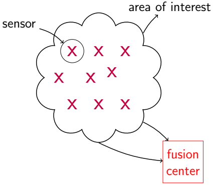
1. Introduction
   - Overview
2. Structure/Topology
   - Types
3. Routing
   - Djikstra
4. Detection
   - Local tests
   - Global test
   - Example
   - Neyman-Pearson

Sensors networks



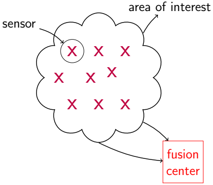
- WSN (wireless sensor network): collection of sensor nodes deployed in a certain area of interest to monitor a physical phenomenon.

Sensors networks



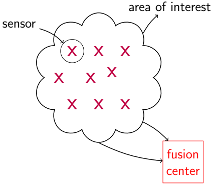
- WSN (wireless sensor network): collection of sensor nodes deployed in a certain area of interest to monitor a physical phenomenon.
- Sensor/Node: device with sensing, processing, data storing and communication capabilities.

Sensors networks



- WSN (wireless sensor network): collection of sensor nodes deployed in a certain area of interest to monitor a physical phenomenon.
- Sensor/Node: device with sensing, processing, data storing and communication capabilities.
- Skills: cooperation, adaptability, self-organization, robustness.

Sensors networks



- WSN (wireless sensor network): collection of sensor nodes deployed in a certain area of interest to monitor a physical phenomenon.
- Sensor/Node: device with sensing, processing, data storing and communication capabilities.
- Skills: cooperation, adaptability, self-organization, robustness.
- Fusion center: device which integrates the data produced by the sensor nodes.

Applications

- Military: command, control, communications, surveillance, exploration, detection/tracking of targets

Applications

- Military: command, control, communications, surveillance, exploration, detection/tracking of targets
- Health/medicine: monitorization, diagnosis, assistance, e.g.,
    - CodeBlue and Vital Dust, devices that monitor heart beat, oxygen levels, electrocardiogram, and send them to a smartphone

Applications

- Military: command, control, communications, surveillance, exploration, detection/tracking of targets
- Health/medicine: monitorization, diagnosis, assistance, e.g.,
  - CodeBlue and Vital Dust, devices that monitor heart beat, oxygen levels, electrocardiogram, and send them to a smartphone
- Civil engineering, e.g.,
  - Smart Buildings
  - Smart Cities.

Applications

- Military: command, control, communications, surveillance, exploration, detection/tracking of targets
- Health/medicine: monitorization, diagnosis, assistance, e.g.,
    - CodeBlue and Vital Dust, devices that monitor heart beat, oxygen levels, electrocardiogram, and send them to a smartphone
- Civil engineering, e.g.,
    - Smart Buildings
    - Smart Cities.
- Environmental monitoring, e.g.,
    - Princeton's Zebranet Project
    - meteorological sensors network deployed in Big Island, Hawaii.
    - sensors network monitoring infrasonic waves in Tungurahua volcano, Ecuador.

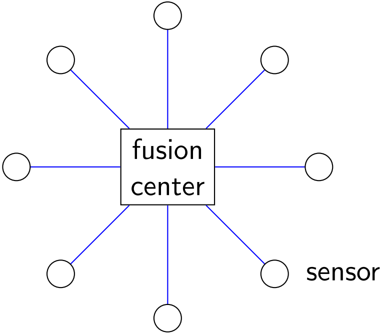
Applications

- Military: command, control, communications, surveillance, exploration, detection/tracking of targets
- Health/medicine: monitorization, diagnosis, assistance, e.g.,
  - CodeBlue and Vital Dust, devices that monitor heart beat, oxygen levels, electrocardiogram, and send them to a smartphone
- Civil engineering, e.g.,
  - Smart Buildings
  - Smart Cities.
- Environmental monitoring, e.g.,
  - Princeton's Zebranet Project
  - meteorological sensors network deployed in Big Island, Hawaii.
  - sensors network monitoring infrasonic waves in Tungurahua volcano, Ecuador.
- Agriculture, e.g.,
  - Intel's Wireless Vineyard.

Index

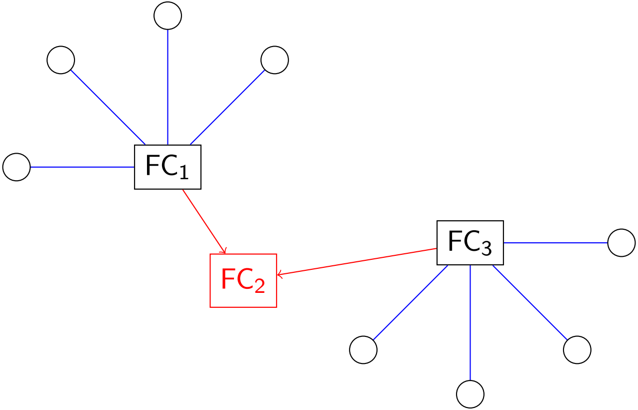
1. Introduction
   - Overview
2. Structure/Topology
   - Types
3. Routing
   - Djikstra
4. Detection
   - Local tests
   - Global test
   - Example
   - Neyman-Pearson

Star topology



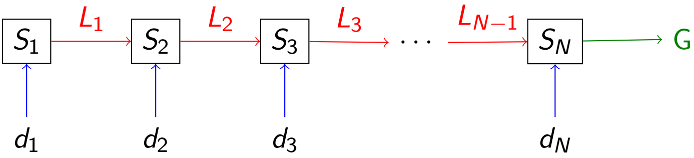
- Communications: broadcast and multicast.
- Control commands from the FC to the sensors. Data from the sensors to the FC.

Hierarchical structure



Point-to-point communications between FCs.

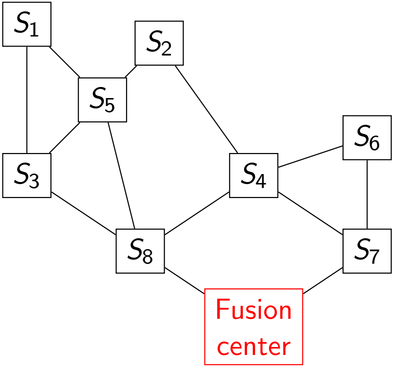
Structure in series



- $S_i \equiv i$-th sensor, $d_i \equiv$ data/measure from the i-th sensor, $L_i \equiv$ output from local processor in the i-th sensor, $G \equiv$ global output
- Recursive processing: $L_i = \phi(L_{i-1}, d_i)$

Mesh structure

Multi-hop communication



Point-to-point communications and routing techniques.

Index

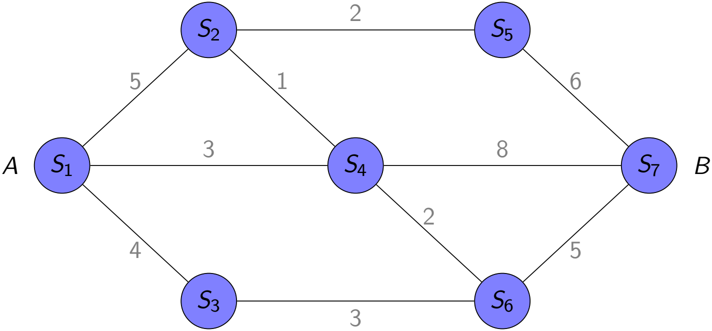
1. Introduction
   - Overview
2. Structure/Topology
   - Types
3. Routing
   - Djikstra
4. Detection
   - Local tests
   - Global test
   - Example
   - Neyman-Pearson

Finding the optimal path

- In an ad-hoc network, the path that is going to be used to communicate any couple of nodes must be known beforehand.
- Due to energy constraints, nodes A and B should communicate with each other following the least costly route.
- Problem with combinatorial complexity.
- Djikstra algorithm.

Djikstra algorithm

- We start with a fully connected graph in which every edge has a certain cost.



- The cost of connecting A and B is given by the sum of the costs of all connections in the path that joins them.

Djikstra algorithm

- Three kinds of nodes: the current node (CN), the inspected nodes, and the non-inspected nodes.

Djikstra algorithm

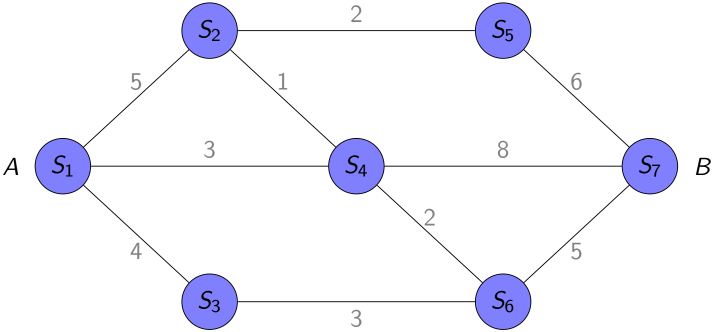
- Three kinds of nodes: the current node (CN), the inspected nodes, and the non-inspected nodes.
- Initialization:
  - the starting node, A, is set as the current node (CN), and its *accumulated* cost is 0
  - every other node is marked as non-inspected, and the overall cost of reaching it is ∞ (not indicated explicitly, but left blank to avoid clutter)

Djikstra algorithm

- Three kinds of nodes: the current node (CN), the inspected nodes, and the non-inspected nodes.
- Initialization:
    - the starting node, A, is set as the current node (CN), and its *accumulated* cost is 0
    - every other node is marked as non-inspected, and the overall cost of reaching it is ∞ (not indicated explicitly, but left blank to avoid clutter)
- Repeat:
    - we compute the distance from the CN **to every one of its neighbors** that is non-inspected; if for a certain node the computed distance is smaller than the current accumulated distance (overall cost of reaching it), the latter is replaced.
    - we mark the current node as inspected
    - the new CN is that node among the non-inspected ones with the smallest cost.

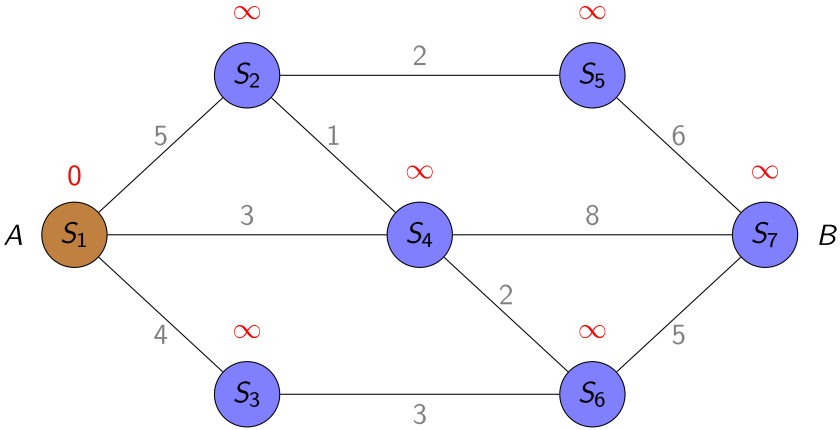
while B has not been marked as inspected.

Example



- The goal is to find the smallest-cost route between A (sensor S_1) and B (sensor S_7).

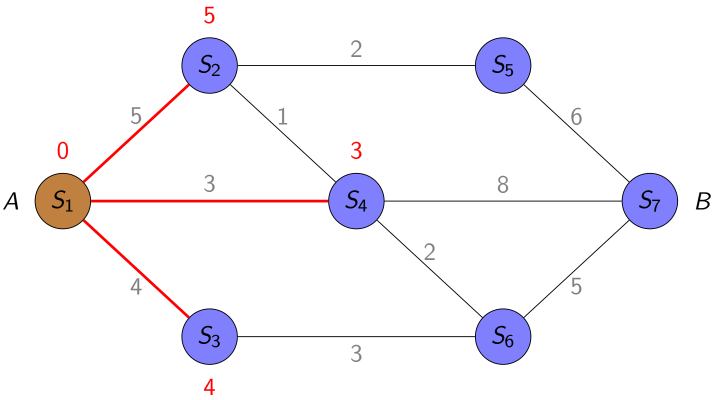
Example



Initialization

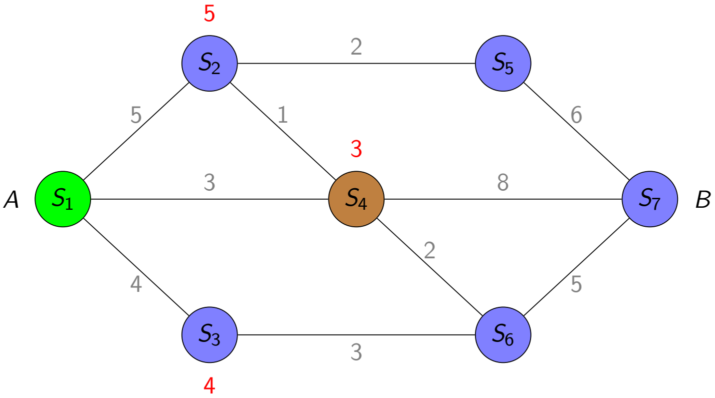
- We designate the starting node S_1 as current node (CN).
- The overall cost of reaching any other node is ∞.

Example



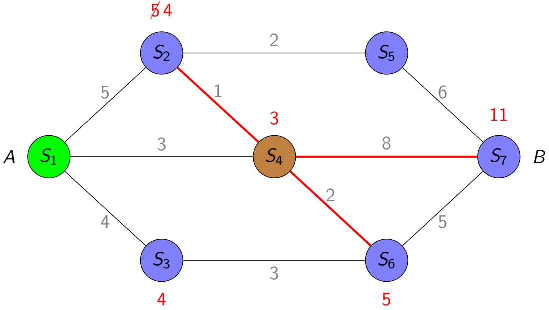
- We compute the distances to its neighbors.

Example



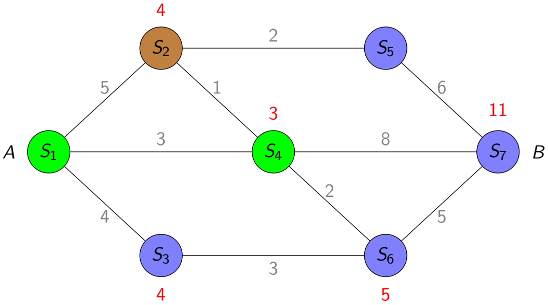
- We mark S_1 as inspected.
- Among the non-inspected nodes, S_4 is the one with smallest cost: it is the next CN.

Example



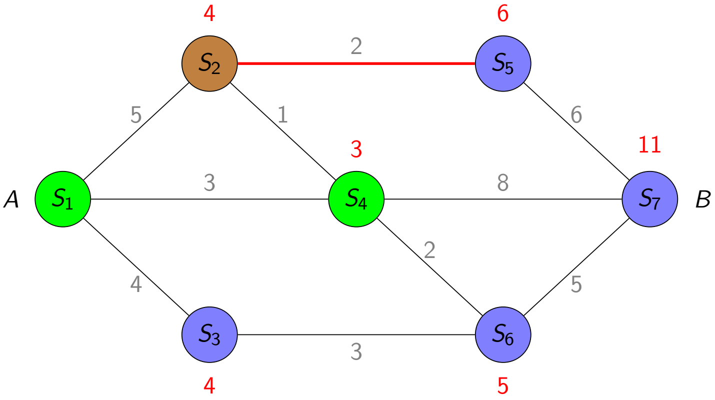
- We compute the accumulated distances up to S_2 $(3 + 1 = 4)$, S_6 $(3 + 2 = 5)$ and S_7 $(3 + 8 = 11)$.
- Since the new distance to S_2 is smaller than the one previously obtained (the path $S_1 \rightarrow S_4 \rightarrow S_2$ has a lower cost than $S_1 \rightarrow S_2$), the latter is *overwritten*.

Example



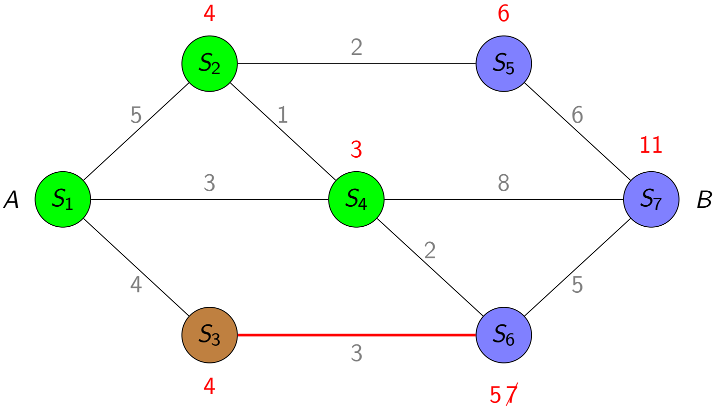
- Nodes S_2 and S_3 are the ones with the smaller cost among the non-inspected.
- We choose either one of them as the next CN, S_2 for instance.

Example



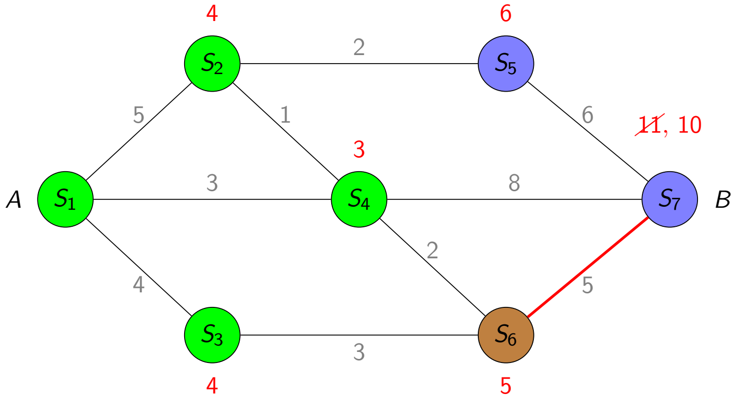
- We compute the distance up to S_5, which is the only non-inspected neighbor of S_2.

Example



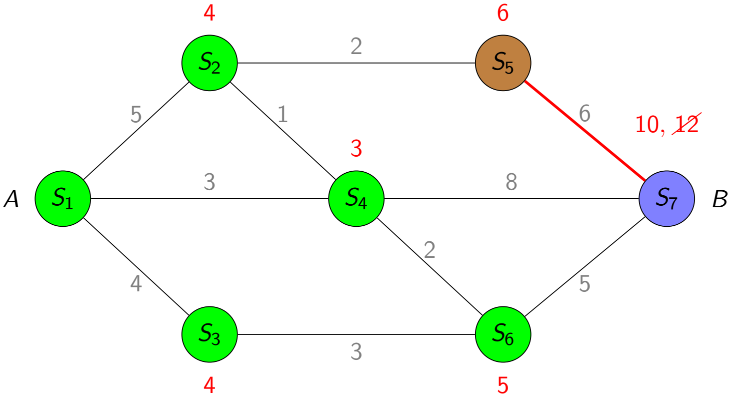
- We set S_3 as CN and compute the distance accumulated up to S_6. Since the resulting cost is larger than the one previoulsy obtained, we ignore it.

Example



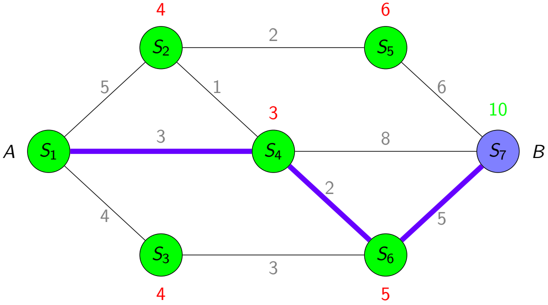
- We set S_6 as CN.
- We update the distance up to S_7, which is smaller than the one previously obtained.

Example



- We set S_5 as CN.
- The distance up to S_7 is no smaller than the previous one.

Example

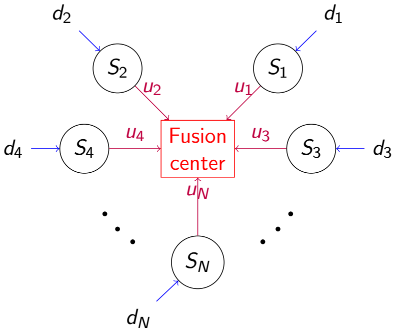


- In the following step, node S_7 would be marked as inspected, and the algorithm ends here.
- The path with the smallest cost is $S_1 \rightarrow S_4 \rightarrow S_6 \rightarrow S_7$.

Index

1. Introduction
   - Overview
2. Structure/Topology
   - Types
3. Routing
   - Djikstra
4. Detection
   - Local tests
   - Global test
   - Example
   - Neyman-Pearson

Centralized detection network



Basic diagram for a wireless sensors network.

Structure of a network with N sensors $(i = 1, \ldots, N)$

- $S_i \equiv i$-th sensor
- $d_i \equiv$ observation in the i-th sensor
- $u_i \in \{0, 1\} \equiv$ decision in the i-th sensor
- $u_0 \in \{0, 1\} \equiv$ decision in the FC.

Local processing

- Sensor S_i, $i \in \{1, ..., N\}$, records observation d_i and must decide between the hypothesis
  - H_0: the phenomenom of interest is *not* present
  - H_1: the phenomenom of interest is *indeed* present

  $$u_i = x \Leftrightarrow S_i \text{ "believes" } H_x \text{ is the correct hypothesis}$$
- H_0 is the "null hypothesis", H_1 is the "alternative hypothesis"

Local processing

- Sensor S_i, $i \in \{1, ..., N\}$, records observation d_i and must decide between the hypothesis
  - H_0: the phenomenom of interest is *not* present
  - H_1: the phenomenom of interest is *indeed* present

  $$u_i = x \Leftrightarrow S_i \text{ "believes" } H_x \text{ is the correct hypothesis}$$

- H_0 is the "null hypothesis", H_1 is the "alternative hypothesis"
- From d_i, statistic $t_i(d_i)$ is computed to make a decision. Observation d_i is random, whereas t_i is a deterministic function of d_i.

Local processing

- Sensor S_i, $i \in \{1, ..., N\}$, records observation d_i and must decide between the hypothesis
  - H_0: the phenomenom of interest is *not* present
  - H_1: the phenomenom of interest is *indeed* present

$$u_i = x \Leftrightarrow S_i \text{ "believes" } H_x \text{ is the correct hypothesis}$$

- H_0 is the "null hypothesis", H_1 is the "alternative hypothesis"
- From d_i, statistic $t_i(d_i)$ is computed to make a decision. Observation d_i is random, whereas t_i is a deterministic function of d_i.
- The output of the **binary** test is $u_i \in \{0, 1\}$:

$$u_i = \begin{cases} 0, & \text{if} \quad t_i(d_i) < \beta_i \\ 1, & \text{if} \quad t_i(d_i) > \beta_i \end{cases}$$

where β_i is the threshold of the test. If $t_i(d_i) \in \mathbb{R}$, then the probability of $t_i(d_i) = \beta_i$ is 0.

Local processing: parameters of interest

The parameters of interest for the *i*-th test (in the corresponding sensor) are:

- the probability of false alarm

$$\alpha_i = \mathbb{P}\{u_i = 1 | H_0\} = \mathbb{P}\{t_i(d_i) > \beta_i | H_0\},$$

Local processing: parameters of interest

The parameters of interest for the *i*-th test (in the corresponding sensor) are:

- the probability of false alarm

$$\alpha_i = \mathbb{P}\{u_i = 1 | H_0\} = \mathbb{P}\{t_i(d_i) > \beta_i | H_0\},$$

- the probability of detection

$$\gamma_i = \mathbb{P}\{u_i = 1 | H_1\} = \mathbb{P}\{t_i(d_i) > \beta_i | H_1\},$$

Local processing: parameters of interest

The parameters of interest for the i-th test (in the corresponding sensor) are:

- the probability of false alarm

$$\alpha_i = \mathbb{P}\{u_i = 1|H_0\} = \mathbb{P}\{t_i(d_i) > \beta_i|H_0\},$$

- the probability of detection

$$\gamma_i = \mathbb{P}\{u_i = 1|H_1\} = \mathbb{P}\{t_i(d_i) > \beta_i|H_1\},$$

- the probability of missing

$$\varepsilon_i = \mathbb{P}\{u_i = 0|H_1\} = \mathbb{P}\{t_i(d_i) < \beta_i|H_1\}.$$

Data fusion I

- Sensors transmit their local decisions to the fusion center (FC) in the network. In the FC, the "observation" is the vector

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \{0,1\}^N.$$

Data fusion I

- Sensors transmit their local decisions to the fusion center (FC) in the network. In the FC, the "observation" is the vector

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \{0,1\}^N.$$

- The FC processes vector **u** to obtain a global output $u_0 \in \{0,1\}$.

Data fusion I

- Sensors transmit their local decisions to the fusion center (FC) in the network. In the FC, the "observation" is the vector

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \{0,1\}^N.$$

- The FC processes vector $\mathbf{u}$ to obtain a global output $u_0 \in \{0,1\}$.
- Let $\mathbb{P}\{H_k|\mathbf{u}\}$ be the probability of hypothesis H_k $(k \in \{0,1\})$ being true given the decision vector $\mathbf{u}$. The optimal Bayesian test is:

$$\mathbb{P}\{H_0|\mathbf{u}\} \underset{u_0=1}{\overset{u_0=0}{\gtrless}} \mathbb{P}\{H_1|\mathbf{u}\}$$

or, equivalently,

$$u_0 = \begin{cases} 0 & \text{if } \mathbb{P}\{H_0|\mathbf{u}\}/\mathbb{P}\{H_1|\mathbf{u}\} > 1 \\ 1 & \text{if } \mathbb{P}\{H_0|\mathbf{u}\}/\mathbb{P}\{H_1|\mathbf{u}\} < 1 \end{cases}$$

Data fusion II

- Using Bayes theorem, we can write the posterior probability of hypothesis H_k as

$$\mathbb{P}\{H_k|\mathbf{u}\} = \frac{\mathbb{P}\{\mathbf{u}|H_k\}\mathbb{P}\{\mathbf{H}_k\}}{\mathbb{P}\{\mathbf{u}\}},$$

where $\mathbb{P}\{H_k\}$ is the posterior probability of hypothesis H_k.

- If we define the threshold β_0 as the ratio of prior probabilities

$$\beta_0 = \mathbb{P}\{H_0\}/\mathbb{P}\{H_1\},$$

and statistic $T(\mathbf{u})$ as the ratio of likelihoods

$$T(\mathbf{u}) = \mathbb{P}\{\mathbf{u}|H_1\}/\mathbb{P}\{\mathbf{u}|H_0\}$$

then we can rewrite the optimal Bayesian test as

$$u_0 = \begin{cases} 1 & \text{if } T(\mathbf{u}) > \beta_0 \\ 0 & \text{if } T(\mathbf{u}) < \beta_0 \end{cases}$$

Data fusion III

In summary, in order to make the *global* decision we need

- a *global* threshold, β_0, which depends on:
    - the (prior) probability of H_0, $\mathbb{P}\{H_0\}$
    - the (prior) probability of H_1, $\mathbb{P}\{H_1\}$

  (notice they are complementary, i.e., $\mathbb{P}\{H_0\} + \mathbb{P}\{H_1\} = 1$)

Data fusion III

In summary, in order to make the *global* decision we need

- a *global* threshold, β_0, which depends on:
    - the (prior) probability of H_0, $\mathbb{P}\{H_0\}$
    - the (prior) probability of H_1, $\mathbb{P}\{H_1\}$

  (notice they are complementary, i.e., $\mathbb{P}\{H_0\} + \mathbb{P}\{H_1\} = 1$)
- the statistic $T(\mathbf{u})$, which depends on
    - the likelihood of H_0, $\mathbb{P}\{\mathbf{u}|H_0\}$
    - the likelihood of H_1, $\mathbb{P}\{\mathbf{u}|H_1\}$

Data fusion III

In summary, in order to make the *global* decision we need

- a *global* threshold, β_0, which depends on:
  - the (prior) probability of H_0, $\mathbb{P}\{H_0\}$
  - the (prior) probability of H_1, $\mathbb{P}\{H_1\}$

  (notice they are complementary, i.e., $\mathbb{P}\{H_0\} + \mathbb{P}\{H_1\} = 1$)
- the statistic $T(\mathbf{u})$, which depends on
  - the likelihood of H_0, $\mathbb{P}\{\mathbf{u}|H_0\}$
  - the likelihood of H_1, $\mathbb{P}\{\mathbf{u}|H_1\}$

If the local decisions are (conditionally) independent, then the statistic $T(\mathbf{u})$ can be written in terms of the parameters of the local tests

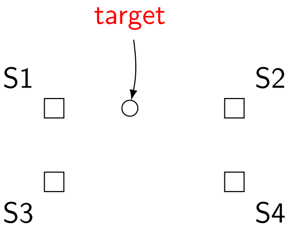
$$T(\mathbf{u}) = \frac{\mathbb{P}\{\mathbf{u}|H_1\}}{\mathbb{P}\{\mathbf{u}|H_0\}} = \frac{\prod_{i=1}^{N} \mathbb{P}\{u_i|H_1\}}{\prod_{i=1}^{N} \mathbb{P}\{u_i|H_0\}}$$

Data fusion: parameters of interest

- The parameters of the test are

$$
\begin{aligned}
\alpha_0 &= \mathbb{P}\{u_0 = 1 | H_0\} \equiv \text{ probability of false alarm} \\
\gamma_0 &= \mathbb{P}\{u_0 = 1 | H_1\} \equiv \text{ probability of detection} \\
\varepsilon_0 &= \mathbb{P}\{u_0 = 0 | H_1\} \equiv \text{ probability of missing}
\end{aligned}
$$

Example



- All the sensors are identical and hence have the same probability of false alarm, $\alpha_i = \alpha = 10^{-3}$.
- Sensors at different distances from the target, and hence $\gamma_1 = 0.9$, $\gamma_2 = 0.7$, $\gamma_3 = 0.5$ and $\gamma_4 = 0.3$.
- Let us assume $\mathbb{P}\{H_1\} = 10^{-3}$.

Example I

- Threshold is given by

$$\beta_0 = \mathbb{P}\{H_0\}/\mathbb{P}\{H_1\} = 0.999/0.001 \approx 10^3.$$

- Assuming $\mathbf{u} = [1, 0, 0, 1]^T$, let us determine the global decision u_0,

$$\begin{aligned} T(\mathbf{u}) &= \frac{\mathbb{P}\{\mathbf{u}|H_1\}}{\mathbb{P}\{\mathbf{u}|H_0\}} = \frac{\prod_{i=1}^4 \mathbb{P}\{u_i|H_1\}}{\prod_{i=1}^4 \mathbb{P}\{u_i|H_0\}} \\ &= \frac{\mathbb{P}\{u_1|H_1\}\mathbb{P}\{u_2|H_1\}\mathbb{P}\{u_3|H_1\}\mathbb{P}\{u_4|H_1\}}{\mathbb{P}\{u_1|H_0\}\mathbb{P}\{u_2|H_0\}\mathbb{P}\{u_3|H_0\}\mathbb{P}\{u_4|H_0\}} \\ &= \frac{\gamma_1}{\alpha_1} \frac{\epsilon_2}{(1-\alpha_2)} \frac{\epsilon_3}{(1-\alpha_3)} \frac{\gamma_4}{\alpha_4} \\ &= \frac{0.9}{10^{-3}} \frac{(1-0.7)}{0.999} \frac{(1-0.5)}{0.999} \frac{0.3}{10^{-3}} \\ &= 4.05 \times 10^4 > \beta_0 \Rightarrow u_0 = 1. \end{aligned}$$

Example II

- Assuming $\mathbf{u} = [0, 1, 0, 0]^\top$, let us determine the global decision u_0,

$$
\begin{aligned}
T(\mathbf{u}) &= \frac{\epsilon_1}{(1-\alpha_1)} \frac{\gamma_2}{\alpha_2} \frac{\epsilon_3}{(1-\alpha_3)} \frac{\epsilon_4}{(1-\alpha_4)} \\
&= \frac{0.1}{0.999} \frac{(0.7)}{10^{-3}} \frac{(1-0.5)}{0.999} \frac{(1-0.3)}{0.999} \\
&= 24.57 < \beta_0 \rightarrow u_0 = 0.
\end{aligned}
$$

Neyman-Pearson lemma

Lemma: Neyman-Pearson

Let us consider the problem of choosing between two hypothesis H_0 and H_1 using a collection of data D. The test evaluating the ratio of likelihoods

$$T(D) = \frac{\mathbb{P}\{D|H_1\}}{\mathbb{P}\{D|H_0\}},$$

with constant probability of false alarm α (associated with a decision threshold β), maximizes the probability of detection γ.

Neyman-Pearson lemma

Lemma: Neyman-Pearson

Let us consider the problem of choosing between two hypothesis H_0 and H_1 using a collection of data D. The test evaluating the ratio of likelihoods

$$T(D) = \frac{\mathbb{P}\{D|H_1\}}{\mathbb{P}\{D|H_0\}},$$

with constant probability of false alarm α (associated with a decision threshold β), maximizes the probability of detection γ.

- The optimal Bayesian test belongs to the Neyman-Pearson class, and hence it maximizes the probability of detection γ.

Neyman-Pearson lemma

Lemma: Neyman-Pearson

Let us consider the problem of choosing between two hypothesis H_0 and H_1 using a collection of data D. The test evaluating the ratio of likelihoods

$$T(D) = \frac{\mathbb{P}\{D|H_1\}}{\mathbb{P}\{D|H_0\}},$$

with constant probability of false alarm α (associated with a decision threshold β), maximizes the probability of detection γ.

- The optimal Bayesian test belongs to the Neyman-Pearson class, and hence it maximizes the probability of detection γ.
- For a fixed probability of false alarm $\alpha_i = \alpha \ \forall i$, the probability of global detection γ is maximized using optimal local test in every sensor.