Collaborative Processing in Sensor Network

Lecture 4 - Distributed In-network Processing

Hairong Qi, Associate Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.eecs.utk.edu/faculty/qi
Email: hqi@utk.edu

Lecture Series at ZheJiang University, Summer 2008
Research Focus - Recap

• Develop energy-efficient collaborative processing algorithms with fault tolerance in sensor networks
  – Where to perform collaboration?
    – Computing paradigms
  – Who should participate in the collaboration?
    – Reactive clustering protocols
    – Sensor selection protocols
  – How to conduct collaboration?
    – In-network processing
    – Self deployment
A Signal Processing and Fusion Hierarchy for Target Classification

Multi-sensor fusion
Mobile-agent Middleware

Multi-modality fusion

Temporal fusion

Local processing

modality 1 $x_1^1(t_1)$

node 1

... ...

node i

node N

Temporal fusion

Local processing

modality 1 $x_1^1(t_r)$

... ...

modality M $x_1^M(t_k)$
Signal - Acoustic/Seismic/IR

Events Time Series for PIR

Events Time Series for Acoustic

Events Time Series for Seismic
Local Signal Processing

- Power Spectral Density (PSD)
  - Shape stat.
- Amplitude stat.
- Peak selection
- Coefficients

Time series signal

Wavelet Analysis

Feature vectors (26 elements)

Feature normalization, Principal Component Analysis (PCA)

Target Classification (kNN)

Classifier design
A Signal Processing and Fusion Hierarchy for Target Classification

Multi-sensor fusion

Multi-modality fusion

Temporal fusion

Local processing

modality 1 $x_1^1(t_1)$

modality 1 $x_1^1(t_r)$

modality M $x_1^M(t_k)$

node 1 node i node N
Temporal Fusion

- Fuse all the 1-sec sub-interval local processing results corresponding to the same event (usually lasts about 10-sec)
- Majority voting

\[
\varphi_i^j = \arg \max_c \omega_c, \quad c \in [1, C]
\]

- number of local output c occurrence
- number of possible local processing results
A Signal Processing and Fusion Hierarchy for Target Classification

Multi-sensor fusion

Multi-modality fusion

Temporal fusion

Local processing

modality 1 \( x_1^1(t_1) \)

modality 1 \( x_1^1(t_r) \)

modality M \( x_1^M(t_k) \)

node 1

node i

node N
Multi-modality Fusion

- Difference sensing modalities can compensate each other’s sensing capability and provide a comprehensive view of the event.
- Treat results from different modalities as independent classifiers – classifier fusion.
- Majority voting won’t work.
- Behavior-Knowledge Space algorithm (Huang&Suen)

Assumption:
- 2 modalities
- 3 kinds of targets
- 100 samples in the training set

Then:
- 9 possible classification combinations

<table>
<thead>
<tr>
<th>$c_1$, $c_2$</th>
<th>samples from each class</th>
<th>fused result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>10/3/3</td>
<td>1</td>
</tr>
<tr>
<td>1,2</td>
<td>3/0/6</td>
<td>3</td>
</tr>
<tr>
<td>1,3</td>
<td>5/4/5</td>
<td>1,3</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>3,3</td>
<td>0/0/6</td>
<td>3</td>
</tr>
</tbody>
</table>
A Signal Processing and Fusion Hierarchy for Target Classification

node 1

node i

node N
Value-based vs. Interval-based Fusion

- Interval-based fusion can provide fault tolerance
- Interval integration – overlap function
  - Assume each sensor in a cluster measures the same parameters, the integration algorithm is to construct a simple function (overlap function) from the outputs of the sensors in a cluster and can resolve it at different resolutions as required.

![Overlap Function Diagram]

Crest: the highest, widest peak of the overlap function
Fault Tolerance Comparison of Different Overlap Functions

\( n \): number of sensors
\( f \): number of faulty sensors
\( M \): to find the smallest interval that contains all the intersections of \( n-f \) intervals
\( S \): return interval \([a,b]\) where \( a \) is the \((f+1)\)th left end point and \( b \) is the \((n-f)\)th right end point
\( \Omega \): overlap function
\( N \): interval with the overlap function ranges \([n-f, n]\)

Overlap function for \( \{I_1, I_2, I_3, I_4\} \)

Overlap function for \( \{I_1, I_2, I_3, I_4\} \)
Multi-sensor Fusion for Target Classification

- Generation of local confidence ranges (For example, at each node i, use kNN for each $k \in \{5, \ldots, 15\}$)

<table>
<thead>
<tr>
<th></th>
<th>Class 1</th>
<th>Class 2</th>
<th>...</th>
<th>Class n</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k=5$</td>
<td>3/5</td>
<td>2/5</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>$k=6$</td>
<td>2/6</td>
<td>3/6</td>
<td>...</td>
<td>1/6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$k=15$</td>
<td>10/15</td>
<td>4/15</td>
<td>...</td>
<td>1/15</td>
</tr>
</tbody>
</table>

\{2/6, 10/15\} \{4/15, 3/6\} ... \{0, 1/6\}

smallest \ largest in this column

- Apply the integration algorithm on the confidence ranges generated from each node to construct an overlapping function
Distributed Integration Scheme

- Integration techniques must be robust and fault-tolerant to handle uncertainty and faulty sensor readouts
- Provide progressive accuracy
- A 1-D array serves to represent the partially-integrated overlap function
- Mobile-agent-based framework is employed
Mobile-agent-based Multi-sensor Fusion for Target Classification

Node 1

Confidence range CR1

- [2/6, 10/15] Class 1
- [3/15, 3/6] Class 2

Node 2

Confidence range CR2

- [3/6, 4/5] Class 1
- [0, 5/15] Class 2

Node 3

Confidence range CR3

- [3/5, 10/12] Class 1
- [1/15, 2/8] Class 2

Mobile agent carries CR1

At node 2, fusion code generates partially integrated confidence range CR12

- [3/6, 10/15] Class 1

Fusion result at node 2: class 1 (58%)

Mobile agent carries CR12

At node 3, fusion code generates partially integrated confidence range CR123

- [3/5, 10/15] Class 1
- [3/15, 2/8] Class 2

Fusion result at node 3: class 1 (63%)

Mobile agent carries CR123
An Example of Multi-sensor Fusion

\[ c = h \times w \times \text{acc} \]

h: height of the highest peak
w: width of the peak
acc: confidence at the center of the peak
Example of Multi-sensor Integration
Result at Each Stop

<table>
<thead>
<tr>
<th></th>
<th>stop 1</th>
<th>stop 2</th>
<th>stop 3</th>
<th>stop 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>1</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>acc</td>
<td>0.2</td>
<td>0.125</td>
<td>0.125</td>
<td>0.125</td>
</tr>
<tr>
<td>class 1</td>
<td>1</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>class 2</td>
<td>2.3</td>
<td>4.55</td>
<td>0.35</td>
<td>0.75</td>
</tr>
<tr>
<td>class 3</td>
<td>0.7</td>
<td>0.5</td>
<td>3.3</td>
<td>3.45</td>
</tr>
</tbody>
</table>

Confidence integration

<table>
<thead>
<tr>
<th></th>
<th>stop 1</th>
<th>stop 2</th>
<th>stop 3</th>
<th>stop 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>1</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>acc</td>
<td>0.2</td>
<td>0.125</td>
<td>0.125</td>
<td>0.125</td>
</tr>
<tr>
<td>class 1</td>
<td>1</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>class 2</td>
<td>2.3</td>
<td>4.55</td>
<td>0.35</td>
<td>0.75</td>
</tr>
<tr>
<td>class 3</td>
<td>0.7</td>
<td>0.5</td>
<td>3.3</td>
<td>3.45</td>
</tr>
</tbody>
</table>

Confidence integration

<table>
<thead>
<tr>
<th></th>
<th>stop 1</th>
<th>stop 2</th>
<th>stop 3</th>
<th>stop 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>1</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>acc</td>
<td>0.2</td>
<td>0.125</td>
<td>0.125</td>
<td>0.125</td>
</tr>
<tr>
<td>class 1</td>
<td>1</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
</tr>
<tr>
<td>class 2</td>
<td>2.3</td>
<td>4.55</td>
<td>0.35</td>
<td>0.75</td>
</tr>
<tr>
<td>class 3</td>
<td>0.7</td>
<td>0.5</td>
<td>3.3</td>
<td>3.45</td>
</tr>
</tbody>
</table>

Confidence integration
Performance Evaluation of Target Classification Hierarchy

North-South

East-West

Center
SITEX02 Scenario Setup

- Acoustic sampling rate: 1024Hz
  Seismic sampling rate: 512 Hz
- Target types: AAV, DW, and HMMWV
- Collaborated work with two other universities (Penn State, Wisconsin)
- Data set is divided into 3 partitions: q1, q2, q3
- Cross validation
  - M1: Training (q2+q3); Testing (q1)
  - M2: Training (q1+q3); Testing (q2)
  - M3: Training (q1+q2); Testing (q3)
Confusion Matrices of Classification on SITEX02

<table>
<thead>
<tr>
<th></th>
<th>AAV</th>
<th>DW</th>
<th>HMV</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAV</td>
<td>29</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>DW</td>
<td>0</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td>HMV</td>
<td>0</td>
<td>2</td>
<td>23</td>
</tr>
</tbody>
</table>

Acoustic (75.47%, 81.78%)

Seismic (85.37%, 89.44%)

Multi-modality fusion (84.34%)

Multi-sensor fusion (96.44%)
Result from BAE-Austin Demo

Ford 250

Harley Motocycle

Ford 350

Suzuki Vitara
Acoustic

Seis

Multi-modal
Demo

- Mobile-agent-based multi-modal multi-sensor classification
  - Mobile agent migrates with a fixed itinerary (Tennessee)
  - Mobile agent migrates with a dynamic itinerary realized by distributed services (Auburn)
run 1:
  car
  walker
run 2:
  car
Target Detection in Sensor Networks

- Single target detection
  - Energy threshold
  - Energy decay model: 
    \[ E_{obs} = \frac{E_{source}}{d^\alpha} \]

- Multiple target detection
  - Why is multiple target detection necessary?
  - Requirements: energy-efficient & bandwidth efficient
  - Multiple target detection vs. source number estimation

\[ S = WX \]

- Target separation
- Speaker separation

Assumption: 
\[ size(S) = size_f \]
Source Number Estimation (SNE)

• Task: \( m = \arg \max_m P(H_m \mid X) \)

• Algorithms for source number estimation
  – Heuristic techniques
  – Principled approaches: Bayesian estimation, Markov chain Monte Carlo method, etc.

• Limitations
  – Centralized structure
  – Large amount of raw data transmission
  – Computation burden on the central processing unit
Distributed Source Number Estimation Scheme

System architecture

Algorithm structure
Centralized Bayesian Estimation Approach

\( \mathbf{X} \): sensor observation matrix
\( \mathbf{S} \): source matrix
\( \mathbf{A} \): mixing matrix, \( \mathbf{X} = \mathbf{A} \mathbf{S} \)
\( \mathbf{W} \): unmixing matrix, \( \mathbf{S} = \mathbf{W} \mathbf{X} \) and \( \mathbf{W} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \)
\( H_m \): hypothesis of the number of sources
\( \mathbf{a} \): latent variable, \( \mathbf{a} = \mathbf{W} \mathbf{X} \) and \( \mathbf{S} = \phi(\mathbf{a}) \)
\( \phi \): non-linear transformation function
\( R_n \): noise, with variance \( 1/\beta \)
\( \pi(\cdot) \): marginal distribution of \( \mathbf{a} \)

\[
L(m) = \log p(\mathbf{x}^{(t)} | H_m) = \log \pi(\mathbf{a}(t)) + \frac{1}{2} (n - m) \log(\frac{\hat{\beta}}{2\pi}) - \frac{1}{2} \log \left| \mathbf{A}^T \mathbf{A} \right| - \frac{\hat{\beta}}{2} (\mathbf{x}(t) - \mathbf{A} \mathbf{a}(t))^2
- \left[ \frac{mn}{2} \log(\frac{\hat{\beta}}{2\pi}) + \frac{n}{2} \left( \sum_{j=1}^{m} \log \hat{a}_j \right) + mn \log \gamma \right]
\]

Progressive Bayesian Estimation Approach

- Motivation
  - Reduce data transmission
  - Conserve energy

Local estimation at cluster 1
- Centralized
- Progressive

Local estimation at cluster L
- Centralized
- Progressive

Posterior probability fusion

\( m \) targets present in sensor field

\( L_1(m) \)

\( L_L(m) \)
Progressive Bayesian Estimation Approach

\[ L(m) = \log p(x^{(t)} | H_m) \]
\[ = \log \pi(\hat{a}(t)) + \frac{1}{2} (n-m) \log\left(\frac{\beta}{2\pi}\right) - \frac{1}{2} \log |\hat{A}^T \hat{A}| - \frac{\beta}{2} (x(t) - \hat{A} \hat{a}(t))^2 \]
\[ - \frac{mn}{2} \log\left(\frac{\beta}{2\pi}\right) + \frac{n}{2} \left(\sum_{j=1}^{m} \log \hat{a}_j^2\right) + mn \log \gamma \]

\[ I_i \text{ includes} \]
\[ [A]_i \]
\[ [a^{(t)}]_i \]
\[ \varepsilon_i \]
\[ [\text{Term1}]_i \]
\[ [\text{Term2}]_i \]
\[ \ldots \]
\[ [\text{Term7}]_i \]

Transmitted information
Progressive Update at Local Sensors

Progressive update of log-likelihood function:

\[ L_i(m) = L_{i-1}(m) + f(x_i^{(t)}) \]

\[ L(m) = \log \pi(\hat{a}(t)) + \frac{1}{2} (n - m) \log \left( \frac{\beta}{2\pi} \right) - \frac{1}{2} \log \left| \hat{A}^T \hat{A} \right| - \frac{\beta}{2} (x(t) - \hat{A} \hat{a}(t))^2 \]

\[ - \left[ \frac{mn}{2} \log \left( \frac{\beta}{2\pi} \right) + \frac{n}{2} \left( \sum_{j=1}^{m} \log a_j \right) + mn \log \gamma \right] \]

\[ [\text{Term1}]_i = [\text{Term1}]_{i-1} - \frac{1}{\alpha} \exp(-2\alpha[a_k]_{i-1})[\exp(-2\alpha w_{k,i} x_i^{(t)}) - 1] - w_{k,i} x_i^{(t)} \]

\[ [\text{Term2}]_i = \frac{i - m}{i - 1 - m} [\text{Term2}]_{i-1} + \frac{i - m}{2} \log \left( \frac{i - 1 - m}{i - m} \right) + \frac{i - m}{2} \cdot \left( x_i^{(t)} - \sum_{k=1}^{m} A_{i,k} \hat{a}_k^{(t)} \right)^2 \]

\[ [\text{Term3}]_i, \text{ only depends on the updating rule of mixing matrix} \quad \hat{A} \]

\[ [\text{Term4}]_i = \frac{i - 1 - m}{i - m} [\text{Term4}]_{i-1} + 2\varepsilon_{i-1} (x_i^{(t)} - \sum_{k=1}^{m} A_{i,k} \hat{a}_k^{(t)}) + (x_i^{(t)} - \sum_{k=1}^{m} A_{i,k} \hat{a}_k^{(t)})^4 \]
Progressive Update at Local Sensors

\[ L(m) = \log \pi(\hat{a}(t)) + \frac{1}{2} (n - m) \log \left( \frac{\beta}{2\pi} \right) - \frac{1}{2} \log \left| \hat{A}^T \hat{A} \right| - \frac{\beta}{2} (x(t) - \hat{A} \hat{a}(t))^2 \]

\[-\left[ \frac{mn}{2} \log \left( \frac{\beta}{2\pi} \right) + \frac{n}{2} \left( \sum_{j=1}^{m} \log a_j \right) + mn \log \gamma \right] \]

[Term5]_i = \frac{i}{i-1} [Term5]_{i-1} + \frac{im}{2} \log \left( \frac{i-1-m}{i-m} \right) + \frac{im}{2} \cdot \frac{(x^{(t)}_i - \sum_{k=1}^{m} A_{i,k} \hat{a}^{(t)}_k)^2}{\varepsilon_{i-1}}

[Term6]_i = \frac{i}{i-1} [Term6]_{i-1} + \frac{i}{2} \sum_{k=1}^{m} \left( w_{k,i} x^{(t)}_i \right)^2 + 2w_{k,i} x^{(t)}_i \hat{a}^{(t)}_k \left[ a_k \right]_{i-1} \left( \hat{a}^{(t)}_k \right)_{i-1} \left( a_k \right)_{i-1}^2

[Term7]_i \text{ only depends on the updating rule of mixing matrix } \hat{A}

Progressive update of mixing matrix \( \hat{A} \): BFGS method (Quasi-Newton method)

Progressive update of error: \( \varepsilon_i = \varepsilon_{i-1} + (x^{(t)}_i - \sum_{k=1}^{m} A^{(t)}_{i,k} a_k)^2 \)
An Example

$$\pi(\hat{a}_k^{(t)}) = \frac{1}{Z(\alpha)[\cosh(\alpha\hat{a}_k^{(t)})]^{\frac{1}{\alpha}}}$$

$$[Term1]_{i-1} = [\log \pi(\hat{a}_k^{(t)})]_{i-1} = -\log Z(\alpha) - \frac{1}{\alpha} \log[\cosh(\alpha\hat{a}_k^{(t)})]$$

$$= -\log Z(\alpha) - \frac{1}{\alpha} \log\left[\frac{\exp(\alpha\hat{a}_k^{(t)}) + \exp(-\alpha\hat{a}_k^{(t)})}{2}\right]$$

$$= -\log Z(\alpha) - \frac{1}{\alpha} \log[\exp(\alpha\hat{a}_k^{(t)})(1 + \exp(-2\alpha\hat{a}_k^{(t)})]] + \frac{1}{\alpha} \log 2$$

$$\log[1 + \exp(-2\alpha\hat{a}_k^{(t)})] = \sum_{p=1}^{\infty} \frac{(-1)^{p-1}}{p}[\exp(-2\alpha\hat{a}_k^{(t)})]^p$$

$$[Term1]_{i-1} = -\log Z(\alpha) - \hat{a}_k^{(t)} - \frac{1}{\alpha} \sum_{p=1}^{\infty} \frac{(-1)^{p-1}}{p}[\exp(-2\alpha\hat{a}_k^{(t)})]^p + \frac{1}{\alpha} \log 2$$

$$[Term1]_i = [\log \pi(\hat{a}_k^{(t)})]_i = -\log Z(\alpha) - \frac{1}{\alpha} \log(\cosh(\alpha \sum_{j=1}^{i} w_{k,j} x_j^{(t)}))$$

$$\approx -\log Z(\alpha) - \sum_{i=1}^{i} w_{k,i} x_i^{(t)} - \frac{1}{\alpha} \exp(-2\alpha \sum_{i=1}^{i} w_{k,i} x_i^{(t)}) + \frac{1}{\alpha} \log 2$$

$$[Term1]_i = [Term1]_{i-1} - w_{k,i} x_i^{(t)} - \frac{1}{\alpha} \exp(-2\alpha[\hat{a}_k^{(t)}]_{i-1})[\exp(-2\alpha w_{k,i} x_i^{(t)}) - 1]$$
Mobile-agent-based Implementation of Progressive Estimation

Sensor 1
$x_1^{(t)}$
Initialization

Sensor 2
$x_2^{(t)}$

Sensor 3
$x_3^{(t)}$

Output $m$

Update $A$
Update terms in log-likelihood function $L(m)^{(t)}$
Compute estimated latent variable $a$
Update estimation error $\varepsilon$
Distributed Fusion Scheme

- **Bayes theorem**

\[
P(H_m \mid x^{(i)}) = \frac{p(x^{(i)} \mid H_m)P(H_m)}{p(x^{(i)})} = \prod_{l=1}^{L} \frac{p(x_l^{(i)} \mid H_m)P(H_m)}{p(x_l^{(i)})}
\]

\[
\log P(H_m \mid x^{(i)}) = \sum_{l=1}^{L} c_l \log p(x_l^{(i)} \mid H_m)
\]

where

\[
c_l = \frac{1}{K_l} \sum_{k=1}^{K_l} E_k = \frac{1}{K_l} \sum_{k=1}^{K_l} \frac{1}{d_k^2}
\]

- **Dempster’s rule of combination**
  - Utilize probability intervals and uncertainty intervals to determine the likelihood of hypotheses based on multiple evidence

\[
P(H_m \mid X) = \sum_{H_f \cap H_g = H_m, i \neq j} \frac{P(H_f \mid X_i) \ast P(H_g \mid X_j)}{1 - \sum_{H_f \cap H_g = \phi, i \neq j} P(H_f \mid X_i) \ast P(H_g \mid X_j)}
\]
Performance Evaluation

Sensor nodes clustering

Signals: 1-second acoustic, 500 samples each

Target types

Experiment 1

Experiment 2

Experiment 3

Experiment 5

Source number estimation

$m$ targets present in sensor field

Local estimation

Centralized

Progressive

$L_t(m)

Posterior probability fusion

Bayesian

Dempster’s rule

$m$ targets present in sensor field

Local estimation

Centralized

Progressive

$L_t(m)$
Average Log-likelihood Comparison

Experiment 1
(Centralized)

Experiment 2
(Centralized + Bayesian)

Experiment 3
Centralized + Dempster’s rule

Experiment 4
(Progressive)

Experiment 5 & 6
(Progressive (cluster 1) vs. fusion)

Experiment 5 & 6
(Progressive (cluster 2) vs. fusion)
Output Histogram Comparison

**Experiment 1**
(Centralized)

**Experiment 2**
(Centralized + Bayesian fusion)

**Experiment 3**
(Centralized + Dempster’s rule)

**Experiment 4**
(Progressive)

**Experiment 5**
(Progressive + Bayesian fusion)

**Experiment 6**
(Progressive + Dempster’s rule)
Performance Comparison

Kurtosis

Detection probability

Data transmission

Energy consumption
Discussion

- Develop a novel distributed multiple target detection scheme with
  - Progressive source number estimation approach (first in literature)
  - Mobile agent implementation
  - Cluster-based probability fusion
- The approach (progressive intra-cluster estimation + Bayesian inter-cluster fusion) has the best performance

<table>
<thead>
<tr>
<th></th>
<th>Kurtosis</th>
<th>Detection probability (%)</th>
<th>Data transmission (bits)</th>
<th>Energy consumption (uW*sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>-1.2497</td>
<td>0.25</td>
<td>144,000</td>
<td>486,095</td>
</tr>
<tr>
<td>Centralized + Bayesian fusion</td>
<td>5.6905</td>
<td>0.65</td>
<td>128,160</td>
<td>433,424</td>
</tr>
<tr>
<td>Progressive + Bayesian fusion</td>
<td>4.8866</td>
<td>0.55</td>
<td>24,160 (16.78%)</td>
<td>89,242 (18.36%)</td>
</tr>
</tbody>
</table>
Reference

