ADAPTIVE COMMUNICATIONS AND SIGNAL PROCESSING LABORATORY
CORNELL UNIVERSITY, ITHACA, NY 14853

Distributed Detection of Information Flows: Implementations of Detection Algorithms

Ting He and Lang Tong

Technical Report No. ACSP-TR-07-07-02

July 2007

I. Introduction

In this work, we present pseudo code implementations of several algorithms for detecting information flows based on quantized timing measurements. We refer to [1] for the problem formulation and the definition of the detectors.

II. Pseudo Code Implementations of Algorithms

A. Case I: Slotted Quantization, Full Side-Information

Consider the detector δ_{I} defined under slotted quantization and full side-information. Its pseudo code implementation is presented in Table I. In this implementation, δ_{I} uses BGM with delay bound $T+\Delta$ to compute the number of chaff packets in $(\hat{\mathbf{s}}_1,\, \mathbf{s}_2)$, denoted by C_{I}. Then it makes detection if the fraction of chaff packets is upper bounded by a threshold τ_{I}.

TABLE I

Detector for Case I.

```
δ_I(x^n, s_2, Δ, τ_I):
  i = 1;
  C_I = 0;
  for k = 1 : n
    if s_2(i + x_k − 1) < kT
      C_I = C_I + |S_2 ∩ [max(s_2(i + x_k), Δ), kT)|;
      i = inf{j : s_2(j) ≥ kT};
    else if s_2(i + x_k − 1) > kT + Δ
        C_I = C_I + x_k − |S_2 ∩ [s_2(i), kT + Δ)|;
        i = inf{j : s_2(j) ≥ kT + Δ};
      else
        i = i + x_k;
      end
    end
  end
```

$$\text{return} \begin{cases} \mathcal{H}_1 & \text{if } C_{\text{I}}/(\sum_{k=1}^{n} x_k + |\mathbb{S}_2|) \leq \tau_{\text{I}}, \\ \mathcal{H}_0 & \text{o.w.;} \end{cases}$$

B. *Case II: Slotted Quantization, Equal Capacity Constraints*

The detector δ_{II} defined under slotted quantization of equal slot length T ($T \geq \Delta$) is implemented in Table II. This implementation can be generalized for arbitrary T. In this implementation, δ_{II} computes C_{II}, the number of chaff packets inserted by BGM in the batched processes $(\hat{\mathbf{s}}_1,\ \hat{\mathbf{s}}_2)$ constructed from $(\mathbf{x}^n,\ \mathbf{y}^n)$ with delay bound T (generally, the delay bound should be $\lceil\frac{\Delta}{T}\rceil T$). Then it returns $\mathcal{H}_1$ if the fraction of chaff packets is bounded by a given threshold τ_{II}.

TABLE II

DETECTOR FOR CASE II.

```
δ_II(x^n, y^n, Δ, τ_II):
  i = max(0, y_1 − x_1);
  C_II = 0;
  for k = 1 : n
    if x_k < y_k − i
      C_II = C_II + y_k − i − x_k;
      i = 0;
    else if x_k > y_k − i + y_{k+1}
        C_II = C_II + x_k − y_k + i − y_{k+1};
        i = y_{k+1};
      else
        i = i + x_k − y_k;
      end
    end
  end
```

$$\text{return} \begin{cases} \mathcal{H}_1 & \text{if } C_{\text{II}}/(\sum_{k=1}^{n}(x_k+y_k)) \leq \tau_{\text{II}}, \\ \mathcal{H}_0 & \text{o.w.}; \end{cases}$$

C. *Case III: One-Bit Quantization, Full Side-Information*

Under one-bit quantization and full side-information, the detector δ_{III} is implemented in Table III. Detector δ_{III} computes the number of chaff packets C_{III} inserted by BGM in $(\hat{\mathbf{s}}_1,\ \mathbf{s}_2)$ with delay bound $T + \Delta$. If $x_k = 1$, then we assume that the kth slot contains the number of epochs that minimizes the number of chaff packets among all positive integers. Then δ_{III} estimates the total number of packets and returns $\mathcal{H}_1$ if the estimated fraction of chaff packets is bounded by τ_{III}.

D. Case IV: One-Bit Quantization, Equal Capacity Constraints

The detector proposed for one-bit quantization with equal slot length is δ_{IV}. An implementation of δ_{IV} for the case $T \geq \Delta$ is presented in Table IV. This implementation can be easily amended for other values of T. In the implementation, δ_{IV} uses a variable C_{IV} to count the number of chaff packets inserted by IC-IV, estimates the total traffic size, and then reports $\mathcal{H}_1$ if their ratio is bounded by τ_{IV}.

References

[1] T. He and L. Tong, "Distributed Detection of Information Flows." submitted to IEEE Trans. on Information Forensics and Security, 2007.

TABLE III

DETECTOR FOR CASE III.

```
δ_III(x^n, s_2, Δ, τ_III):
  v = 0;
  u = Δ;
  C_III = 0;
  for k = 1 : n
    if x_k == 0
      if y(u, kT) > 0
        C_III = C_III + y(u, kT);
      end
      v = max(v, kT);
      u = max(u, kT);
    else if y(v, kT + Δ) == 0
        C_III = C_III + 1;
      end
      j' = inf{j : s_2(j) ≥ v};
      if s_2(j') < kT + Δ
        v = max(s_2(j' + 1), kT);
      else
        v = s_2(j');
      end
      u = kT + Δ;
    end
  end
  N = |S_2| − n log(1 − (1/n) Σ_{k=1}^{n} x_k);
  return { H_1  if C_III/N ≤ τ_III,
         { H_0  o.w.;
```

$\delta_{\mathrm{III}}(\mathbf{x}^n,\ \mathbf{s}_2,\ \Delta,\ \tau_{\mathrm{III}})$:

$$N = |\mathcal{S}_2| - n\log\left(1 - \frac{1}{n}\sum_{k=1}^{n} x_k\right);$$

$$\text{return } \begin{cases} \mathcal{H}_1 & \text{if } C_{\mathrm{III}}/N \le \tau_{\mathrm{III}}, \\ \mathcal{H}_0 & \text{o.w.}; \end{cases}$$

TABLE IV

DETECTOR FOR CASE IV.

$\delta_{\text{IV}}(\mathbf{x}^n,\ \mathbf{y}^n,\ \Delta,\ \tau_{\text{IV}})$:

$C_{\text{IV}} = 0$;

$x_0 = 1$;

for $k = 1 : n$

if $(x_k > y_k + y_{k+1})$ or $(y_k > x_{k-1} + x_k)$

$C_{\text{IV}} = C_{\text{IV}} + 1$;

end

end

$N = -n(\log(1 - \frac{1}{n}\sum_{k=1}^{n} x_k) + \log(1 - \frac{1}{n}\sum_{k=1}^{n} y_k))$;

return $\begin{cases} \mathcal{H}_1 & \text{if } C_{\text{IV}}/N \leq \tau_{\text{IV}}, \\ \mathcal{H}_0 & \text{o.w.}; \end{cases}$