Online Learning for Collaborative Spectrum Sensing in Frugal Cognitive Radio Network for IoT

Ph.D. Seminar I

Presented by: Nancy Nayak
Supervisor: Dr. Sheetal Kalyani

February 4, 2024

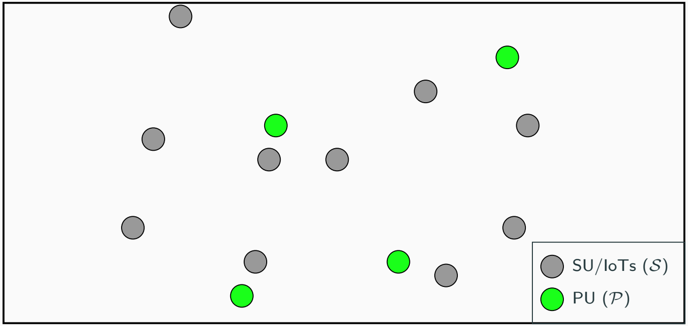
Indian Institute of Technology Madras, India

Outline

1. Background and Motivation
2. Leveraging Online Learning for CSS
3. Reducing false alarm using FDR control
4. CSS in non-stationary environment
5. Experimental validation

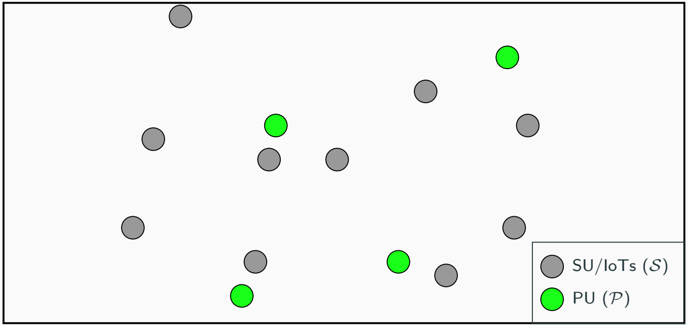
Background and Motivation

Overview



IoT devices in a Cognitive Radio Network

Overview



IoT devices in a Cognitive Radio Network

Assumptions:

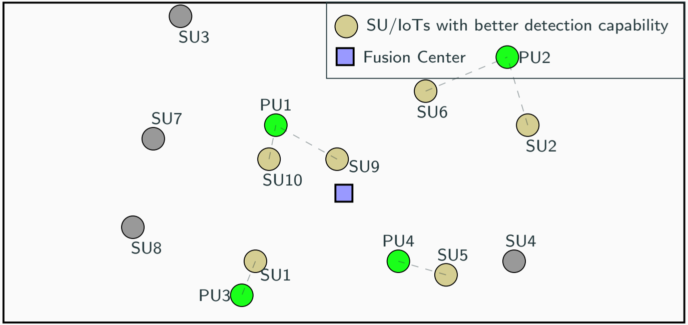
1. Primary users $\mathcal{P}$, secondary users $\mathcal{S}$, $|\mathcal{P}| = P$, $|\mathcal{S}| = S$, and $S \gg P$
2. SUs do not have any time-critical information for transmission

Spectrum Sensing

- Locally sensed data at each SU[1] may be degraded due to
  1. Fading nature of the wireless channel
  2. Hidden PUs
  3. Shadowing

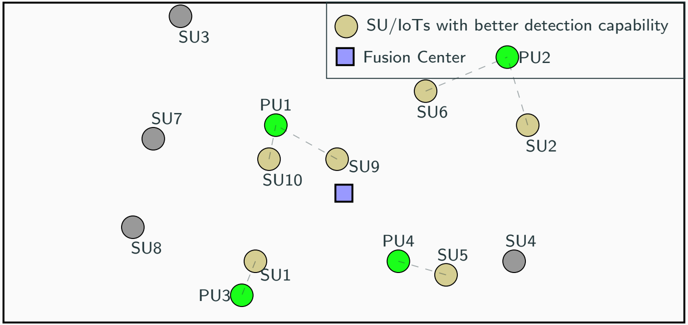
[1] Different robust sensing methods are Energy detectors, Waveform based techniques, Matched filters, Cyclo-stationary based sensing : Yucek, Tevfik, and Huseyin Arslan. "A survey of spectrum sensing algorithms for cognitive radio applications." IEEE communications surveys & tutorials 11, no. 1 (2009): 116-130.

Overview



IoT devices in a Cognitive Radio Network

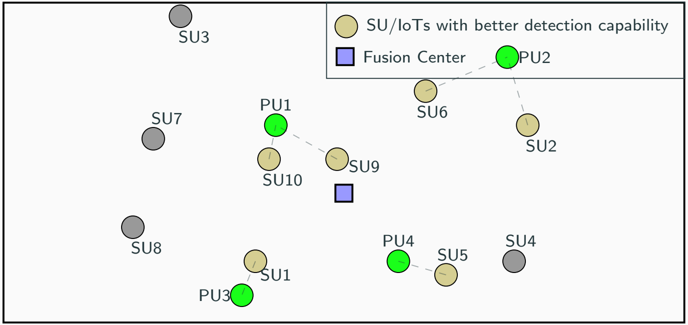
Overview



IoT devices in a Cognitive Radio Network

- SUs simultaneously try to acquire a free channel - collision and data loss

Overview



IoT devices in a Cognitive Radio Network

- SUs simultaneously try to acquire a free channel - collision and data loss
- Information from all the SUs can be fused to identify the state of the channels with high confidence - **Collaborative Spectrum Sensing**

Why Collaborative Spectrum Sensing?

1. Spatial placements of IoT devices
   - perceive the occupancy state of the same channel differently
   - dissimilar detection performance across IoT devices

Why Collaborative Spectrum Sensing?

1. Spatial placements of IoT devices
    - perceive the occupancy state of the same channel differently
    - dissimilar detection performance across IoT devices
2. Dense deployment
    - at least one SU satisfies the SNR condition for correct detection

Traditional CSS schemes:

- Locally sensed observations are combined in different ways.
- Example: AND [2], OR [3], Confidence Voting [4]
- But cannot handle:
  - Widely varying channel conditions of SUs
  - Different detection capability of SUs from different vendors

[2] Visotsky, E., Kuffner, S. and Peterson, R., 2005, November. On collaborative detection of TV transmissions in support of dynamic spectrum sharing. In First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. (pp. 338-345). IEEE.

[3] Ghasemi, A. and Sousa, E.S., 2005, November. Collaborative spectrum sensing for opportunistic access in fading environments. In First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. (pp. 131-136). IEEE.

[4] Lee, C.H. and Wolf, W., 2008, January. Energy efficient techniques for cooperative spectrum sensing in cognitive radios. In 2008 5th IEEE Consumer Communications and Networking Conference (pp. 968-972). IEEE.

Proposed method: Online Learning

- Assuming no prior knowledge about the detection performance of individual SUs, we **weigh the information** from each SUs according to **their relative performance** in an online fashion to arrive at final decision of the channel state
- Why Online Learning?
  - Learns from streaming data
  - Learns the quality of each device

Leveraging Online Learning for CSS

Combining observations in CSS

1. True state of c_j at time step n is $\mathbf{g}(n)$ where $g_j(n) \in \{0, 1\} \quad \forall c_j \in \mathcal{P}$ - **ground truth**

Combining observations in CSS

1. True state of c_j at time step n is $\mathbf{g}(n)$ where $g_j(n) \in \{0, 1\} \quad \forall c_j \in \mathcal{P}$ - **ground truth**
2. Observation $o_{ji}(n)$ is made by the i^{th} SU about the j^{th} channel state at time step n

Combining observations in CSS

1. True state of c_j at time step n is $\mathbf{g}(n)$ where $g_j(n) \in \{0, 1\} \quad \forall c_j \in \mathcal{P}$ - **ground truth**
2. Observation $o_{ji}(n)$ is made by the i^{th} SU about the j^{th} channel state at time step n
3. A **weighted combination of observations $\mathbf{O}$** $\in \mathbb{R}^{P \times S}$ is taken to produce decision $\mathbf{f}(n) \in [0, 1]^P$ where $f_j(n)$ is the decision for state of c_j

Detection method at every SU/IoT device

- Channel detection method: Neyman-Pearson (NP) detector

Detection method at every SU/IoT device

- Channel detection method: Neyman-Pearson (NP) detector
- Let, $H_0 : x(n) = n_0(n)$ and $H_1 : x(n) = s(n) + n_0(n)$ where $s(n)$ is transmitted signal from BS and $n_0(n)$ is the noise

Detection method at every SU/IoT device

- Channel detection method: Neyman-Pearson (NP) detector
- Let, $H_0 : x(n) = n_0(n)$ and $H_1 : x(n) = s(n) + n_0(n)$ where $s(n)$ is transmitted signal from BS and $n_0(n)$ is the noise
- Detect channel state of c_j: $e_{ji}(n) = (1/N)\sum_{s=0}^{N-1} x^2[s] \lesseqgtr \zeta$

Detection method at every SU/IoT device

- Channel detection method: Neyman-Pearson (NP) detector
- Let, $H_0 : x(n) = n_0(n)$ and $H_1 : x(n) = s(n) + n_0(n)$ where $s(n)$ is transmitted signal from BS and $n_0(n)$ is the noise
- Detect channel state of c_j: $e_{ji}(n) = (1/N)\sum_{s=0}^{N-1} x^2[s] \lesseqqgtr \zeta$
- The detection hypothesis can be written as,

$$\frac{e_{ji}(n)}{\sigma^2} \sim \chi_N^2 \quad \text{under } H_0 \tag{1}$$

$$\frac{e_{ji}(n)}{\sigma_s^2 + \sigma^2} \sim \chi_N^2 \quad \text{under } H_1 \tag{2}$$

where σ^2 is noise variance and σ_s^2 is signal variance

Detection method at every SU/IoT device

- According to NP criterion, for a targeted P_{fa}, the threshold to detect a channel

$$\zeta = \sigma^2 . Q^{-1}_{\chi^2_N}(P_{fa}) \tag{3}$$

Detection method at every SU/IoT device

- According to NP criterion, for a targeted P_{fa}, the threshold to detect a channel

$$\zeta = \sigma^2 . Q_{\chi_N^2}^{-1}(P_{fa}) \tag{3}$$

- The prediction of SU s_i about the channel c_j

$$d_{ji}(n) = 1_{[e_{ji}(n) \geq \zeta]} \tag{4}$$

Detection method at every SU/IoT device

- According to NP criterion, for a targeted P_{fa}, the threshold to detect a channel

$$\zeta = \sigma^2 . Q^{-1}_{\chi^2_N}(P_{fa}) \tag{3}$$

- The prediction of SU s_i about the channel c_j

$$d_{ji}(n) = 1_{[e_{ji}(n) \geq \zeta]} \tag{4}$$

- **Combining Hard decisions:** when $o_{ji}(n) = d_{ji}(n)$ and $d_{ji} \in \{0, 1\} \quad \forall i, j$

Detection method at every SU/IoT device

- According to NP criterion, for a targeted P_{fa}, the threshold to detect a channel

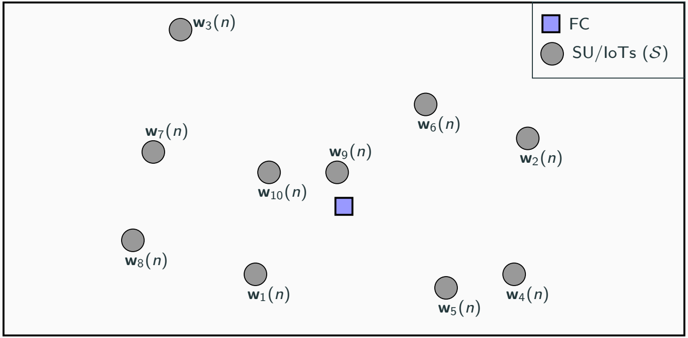
$$\zeta = \sigma^2 . Q^{-1}_{\chi^2_N}(P_{fa}) \tag{3}$$

- The prediction of SU s_i about the channel c_j

$$d_{ji}(n) = 1_{[e_{ji}(n) \geq \zeta]} \tag{4}$$

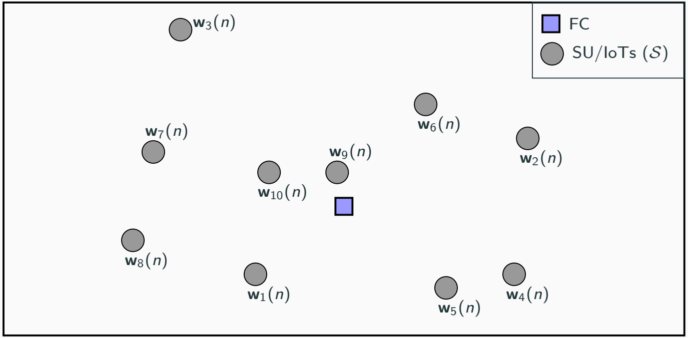
- **Combining Hard decisions:** when $o_{ji}(n) = d_{ji}(n)$ and $d_{ji} \in \{0, 1\} \quad \forall i, j$
- **Combining Soft decisions:** when $o_{ji}(n) = e_{ji}(n)$ and $e_{ji}(n) \in \mathbb{R}^+ \quad \forall i, j$

General framework for online learning for CSS



Weights on the IoT devices based on their performances

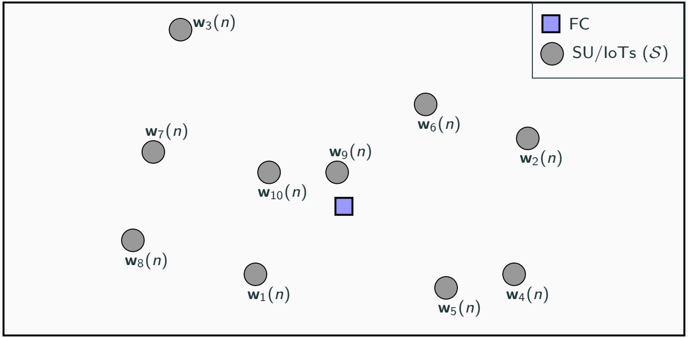
General framework for online learning for CSS



Weights on the IoT devices based on their performances

- Initial weight $\mathbf{w}_i(0)$.

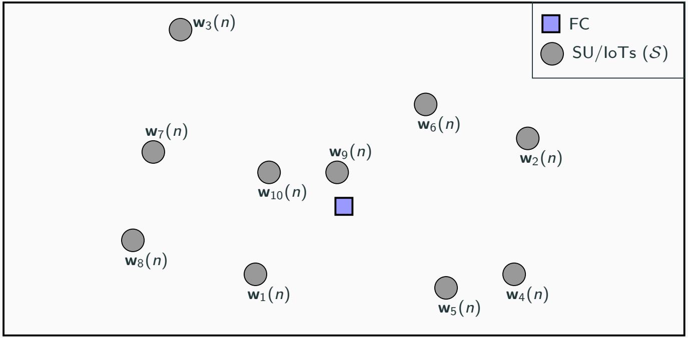
General framework for online learning for CSS



Weights on the IoT devices based on their performances

- Initial weight $\mathbf{w}_i(0)$.
- Weights of the SUs for the PU-channels are stored in $\mathbf{W}(n) \in \mathbb{R}^{P \times S}$

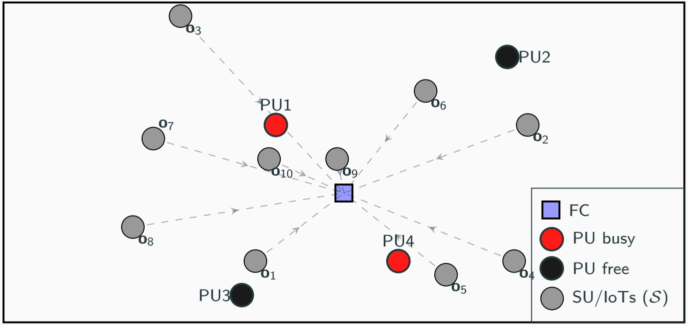
General framework for online learning for CSS



Weights on the IoT devices based on their performances

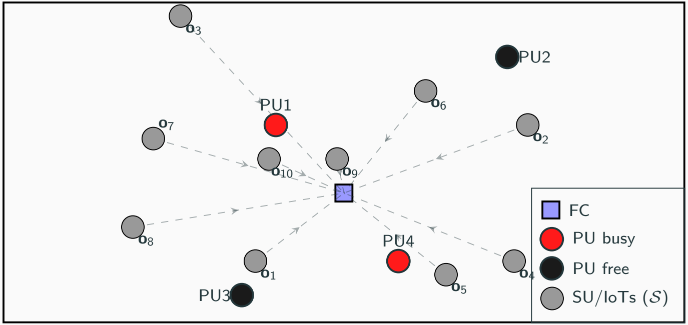
- Initial weight $\mathbf{w}_i(0)$.
- Weights of the SUs for the PU-channels are stored in $\mathbf{W}(n) \in \mathbb{R}^{P \times S}$
- Normalized weight $p_{ji}(n) = \frac{w_{ji}(n)}{\sum_{i=1}^{S} w_{ji}(n)}$.

General framework for online learning for CSS



Combining observations

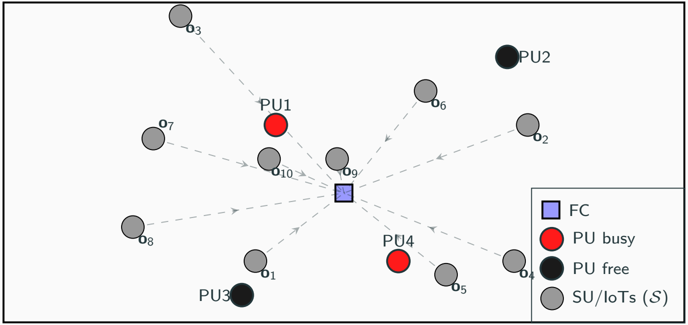
General framework for online learning for CSS



Combining observations

- Let, $\mathbf{g}(n) = [\text{busy, free, free, busy}]$

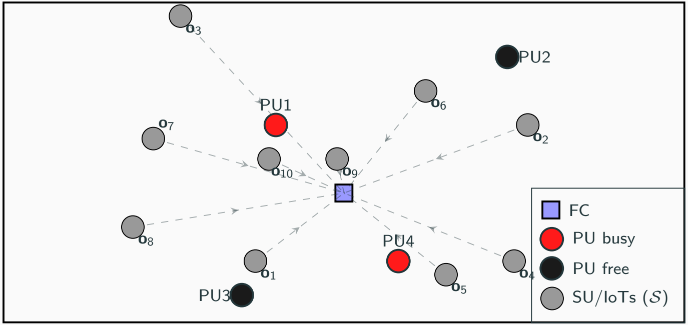
General framework for online learning for CSS



Combining observations

- Let, $\mathbf{g}(n) = [\text{busy, free, free, busy}]$
- Combined information $\tilde{f}_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n)$

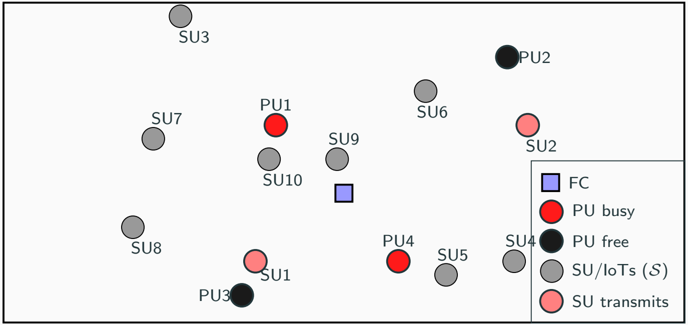
General framework for online learning for CSS



Combining observations

- Let, $\mathbf{g}(n) = [\text{busy, free, free, busy}]$
- Combined information $\tilde{f}_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n)$
- FC's decision $f_j(n)$ is busy if $\tilde{f}_j(n) \geq \gamma_j$ else free

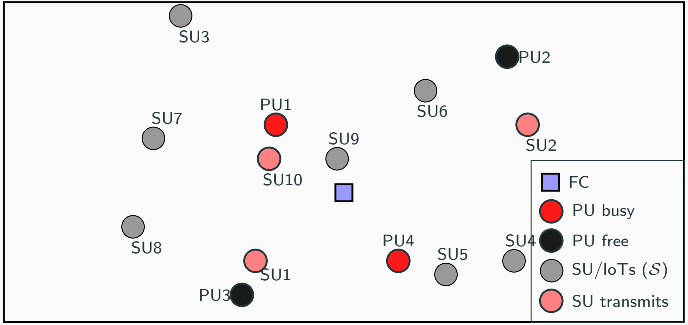
General framework for online learning for CSS



Making a correct final decision

- If $\mathbf{f}(n)$ is **free**, then SUs transmit and if $\mathbf{f}(n)$ is **busy** then they do not
- Let, $\mathbf{f}(n) = [\text{busy, free, free, busy}]$
- If $\mathbf{f}(n) = \mathbf{g}(n)$ then **Correct decision** else **Wrong decision**

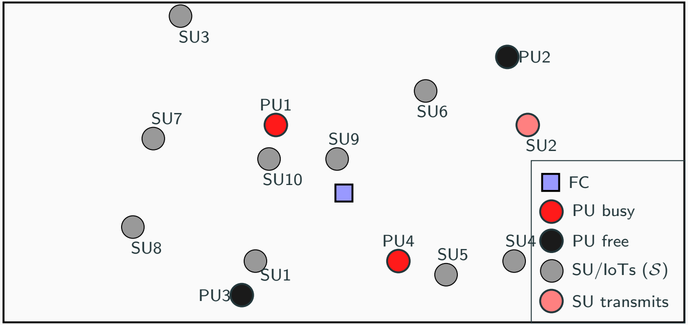
General framework for online learning for CSS



Making a wrong final decision: SU and PU collision

- $\mathbf{g}(n)$ = [busy, free, free, busy], and let, $\mathbf{f}(n)$ = [free, free, free, busy]
- **Collision** between PU1 and SU10

General framework for online learning for CSS



Making a wrong final decision: missed idle slots

- $\mathbf{g}(n)$ = [busy, free, free, busy], and let, $\mathbf{f}(n)$ = [busy, free, busy, busy]
- SU1 **missed the opportunity to transmit** using the channel of PU3
- SUs get to know the ground truth $g_j(n)$ only if they transmit
- **Approximate ground truth** (AGT) when $f_j(n)$ is busy and SU doesn't transmit

Hedge[5] inspired online Learning for CSS

- Find instantaneous loss at SUs using AGT $l_{ji}(n)$ for (s_i, c_j) pair
  $l_{ji}(n) = \mathcal{L}(d_{ji}(n), g_j(n)) = |d_{ji}(n) - g_j(n)| \in \{0, 1\}$

[5] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of computer and system sciences 55, no. 1 (1997): 119-139.

Hedge[5] inspired online Learning for CSS

- Find instantaneous loss at SUs using AGT $l_{ji}(n)$ for (s_i, c_j) pair
  $l_{ji}(n) = \mathcal{L}(d_{ji}(n), g_j(n)) = |d_{ji}(n) - g_j(n)| \in \{0, 1\}$
- The loss is used to update the weights of (s_i, c_j) as $w_{ji}(n+1) \leftarrow w_{ji}(n)\beta^{l_{ji}(n)}$ where $\beta \in (0, 1]$ is the *learning parameter*

[5] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of computer and system sciences 55, no. 1 (1997): 119-139.

Hedge hard combining (Hed-HC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) = \text{busy if } \tilde{f}_j(n) \geq \gamma_j \text{ else free}$

Hedge hard combining (Hed-HC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) =$ busy if $\tilde{f}_j(n) \geq \gamma_j$ else free
- **How to pick γ_j for Hedge Hard Combining (HC)?**

Hedge hard combining (Hed-HC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) =$ busy if $\tilde{f}_j(n) \geq \gamma_j$ else free
- **How to pick γ_j for Hedge Hard Combining (HC)?**
- Individual decisions o_{ji} are either 0 or 1, so γ_j is set to be 0.5

Hedge hard combining (Hed-HC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) =$ busy if $\tilde{f}_j(n) \geq \gamma_j$ else free
- **How to pick γ_j for Hedge Hard Combining (HC)?**
- Individual decisions o_{ji} are either 0 or 1, so γ_j is set to be 0.5
- When all the SUs have equal normalized weight i.e., $\approx \frac{1}{S}$, Hedge-HC behaves same as CV

Hedge hard combining (Hed-HC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) = \text{busy if } \tilde{f}_j(n) \geq \gamma_j \text{ else free}$
- **How to pick γ_j for Hedge Hard Combining (HC)?**
- Individual decisions o_{ji} are either 0 or 1, so γ_j is set to be 0.5
- When all the SUs have equal normalized weight i.e., $\approx \frac{1}{S}$, Hedge-HC behaves same as CV
- Benefit: SUs need to send only one bit of information to the FC - reduces the overhead

Hedge soft combining (Hed-SC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) = \text{busy if } \tilde{f}_j(n) \geq \gamma_j \text{ else free}$

Hedge soft combining (Hed-SC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) = \text{busy if } \tilde{f}_j(n) \geq \gamma_j \text{ else free}$
- **How to pick γ_j for Hed-SC?**

Hedge soft combining (Hed-SC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) =$ busy if $\tilde{f}_j(n) \geq \gamma_j$ else free
- **How to pick γ_j for Hed-SC?**
- Combining this soft information with Hedge

$$\tilde{f}_j(n) = \sum_{i=1}^{S} p_{ji}(n) \eta_{ji}^2 \psi(n), \tag{5}$$

where $\psi(n) \sim \chi_N^2$, $\eta_{ji}^2 = \sigma^2$ under $\mathcal{H}_0$ and $\eta_{ji}^2 = \sigma^2 + \sigma_{sji}^2$ under $\mathcal{H}_1$

Hedge soft combining (Hed-SC) the observations

- Final decision: $f_j(n) = \sum_{i=1}^{S} p_{ji}(n) o_{ji}(n) = \text{busy if } \tilde{f}_j(n) \geq \gamma_j \text{ else free}$
- **How to pick γ_j for Hed-SC?**
- Combining this soft information with Hedge

$$\tilde{f}_j(n) = \sum_{i=1}^{S} p_{ji}(n) \eta_{ji}^2 \psi(n), \tag{5}$$

where $\psi(n) \sim \chi_N^2$, $\eta_{ji}^2 = \sigma^2$ under $\mathcal{H}_0$ and $\eta_{ji}^2 = \sigma^2 + \sigma_{sji}^2$ under $\mathcal{H}_1$

- χ_N^2 is a special case of $Gamma(\frac{N}{2}, 2)$, so $\tilde{f}_j(n) = \sum_{i=1}^{S} p_{ji}(n) \tilde{\psi}(n)$ where $\tilde{\psi}(n) \sim Gamma\left(\frac{N}{2}, 2\eta_{ji}^2\right)$

Hedge soft combining (Hed-SC) the observations

- Weighted sum of gamma random variables follows Confluent Lauricella distribution

Hedge soft combining (Hed-SC) the observations

- Weighted sum of gamma random variables follows Confluent Lauricella distribution
- Thresholds are computationally expensive to calculate online at every step

Hedge soft combining (Hed-SC) the observations

- Weighted sum of gamma random variables follows Confluent Lauricella distribution
- Thresholds are computationally expensive to calculate online at every step
- Approximate $\tilde{f}_j(n)$ under $\mathcal{H}_0$ with another gamma distribution $\Gamma(k_j, \theta_j)$

Hedge soft combining (Hed-SC) the observations

- Weighted sum of gamma random variables follows Confluent Lauricella distribution
- Thresholds are computationally expensive to calculate online at every step
- Approximate $\tilde{f}_j(n)$ under $\mathcal{H}_0$ with another gamma distribution $\Gamma(k_j, \theta_j)$
- By equating first and second moments we get,

$$k_j = \frac{N}{2\sum_{i=1}^{S} p_{ji}^2}, \text{ and } \theta_j = 2\sigma^2 \sum_{i=1}^{S} p_{ji}^2$$

Hedge soft combining (Hed-SC) the observations

- Weighted sum of gamma random variables follows Confluent Lauricella distribution
- Thresholds are computationally expensive to calculate online at every step
- Approximate $\tilde{f}_j(n)$ under $\mathcal{H}_0$ with another gamma distribution $\Gamma(k_j, \theta_j)$
- By equating first and second moments we get,

$$k_j = \frac{N}{2\sum_{i=1}^{S} p_{ji}^2}, \text{ and } \theta_j = 2\sigma^2 \sum_{i=1}^{S} p_{ji}^2$$

- Given a P_{fa}, the threshold is calculated by

$$\gamma_j = Q^{-1}_{\Gamma(k_j,\theta_j)}(P_{fa}) \tag{6}$$

Hedge soft combining (Hed-SC) the observations

- Weighted sum of gamma random variables follows Confluent Lauricella distribution
- Thresholds are computationally expensive to calculate online at every step
- Approximate $\tilde{f}_j(n)$ under $\mathcal{H}_0$ with another gamma distribution $\Gamma(k_j, \theta_j)$
- By equating first and second moments we get,

$$k_j = \frac{N}{2\sum_{i=1}^{S} p_{ji}^2}, \text{ and } \theta_j = 2\sigma^2 \sum_{i=1}^{S} p_{ji}^2$$

- Given a P_{fa}, the threshold is calculated by

$$\gamma_j = Q^{-1}_{\Gamma(k_j,\theta_j)}(P_{fa}) \tag{6}$$

- Benefit: FC can exploit the granularity of the observation to arrive at better decision

Reducing false alarm using FDR control

Controlling False Alarm

- CSS with multiple PU as **multiple hypothesis testing task** [6]

[6] Shaffer, Juliet Popper. "Multiple hypothesis testing." Annual review of psychology 46, no. 1 (1995): 561-584.

Controlling False Alarm

- CSS with multiple PU as **multiple hypothesis testing task** [6]
- Suppose, each of the P channels has a probability of false alarm P_{fa}

[6] Shaffer, Juliet Popper. "Multiple hypothesis testing." Annual review of psychology 46, no. 1 (1995): 561-584.

Controlling False Alarm

- CSS with multiple PU as **multiple hypothesis testing task** [6]
- Suppose, each of the P channels has a probability of false alarm P_{fa}
- The probability of getting at least one false positive, termed as family wise error rate

$$FWER = 1 - (1 - P_{fa})^P \approx P \times P_{fa} \tag{7}$$

[6] Shaffer, Juliet Popper. "Multiple hypothesis testing." Annual review of psychology 46, no. 1 (1995): 561-584.

Controlling False Alarm

- CSS with multiple PU as **multiple hypothesis testing task** [6]
- Suppose, each of the P channels has a probability of false alarm P_{fa}
- The probability of getting at least one false positive, termed as family wise error rate

$$FWER = 1 - (1 - P_{fa})^P \approx P \times P_{fa} \tag{7}$$

- FWER increases with increase in number of channels P

[6] Shaffer, Juliet Popper. "Multiple hypothesis testing." Annual review of psychology 46, no. 1 (1995): 561-584.

Controlling False Alarm

- CSS with multiple PU as **multiple hypothesis testing task** [6]
- Suppose, each of the P channels has a probability of false alarm P_{fa}
- The probability of getting at least one false positive, termed as family wise error rate

$$FWER = 1 - (1 - P_{fa})^P \approx P \times P_{fa} \tag{7}$$

- FWER increases with increase in number of channels P
- Controlling FWER at a level α implies each of the P tests should have $P_{fa} = \alpha/P$ - conservative because this lowers the statistical detection power

[6] Shaffer, Juliet Popper. "Multiple hypothesis testing." Annual review of psychology 46, no. 1 (1995): 561-584.

Controlling False Alarm

- CSS with multiple PU as **multiple hypothesis testing task** [6]
- Suppose, each of the P channels has a probability of false alarm P_{fa}
- The probability of getting at least one false positive, termed as family wise error rate

$$FWER = 1 - (1 - P_{fa})^P \approx P \times P_{fa} \tag{7}$$

- FWER increases with increase in number of channels P
- Controlling FWER at a level α implies each of the P tests should have $P_{fa} = \alpha / P$ - conservative because this lowers the statistical detection power
- The procedure for controlling FWER reduces the probability of getting false positive at the cost of increasing the probability of getting false negative (more collision)

[6] Shaffer, Juliet Popper. "Multiple hypothesis testing." Annual review of psychology 46, no. 1 (1995): 561-584.

False Discovery Rate[7] for FWER control

- **How to control FWER to a low level and still avoid collision?**

[7] Benjamini, Yoav, and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the Royal statistical society: series B (Methodological) 57, no. 1 (1995): 289-300.

False Discovery Rate[7] for FWER control

- **How to control FWER to a low level and still avoid collision?**
- Assume, R of total P channels are declared as occupied among which V of them are falsely declared as occupied.

[7] Benjamini, Yoav, and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the Royal statistical society: series B (Methodological) 57, no. 1 (1995): 289-300.

False Discovery Rate[7] for FWER control

- **How to control FWER to a low level and still avoid collision?**
- Assume, R of total P channels are declared as occupied among which V of them are falsely declared as occupied.
- Then the false discovery rate (FDR) is defined as,

$$FDR = E\left[\frac{V}{R}\right] \tag{8}$$

[7] Benjamini, Yoav, and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the Royal statistical society: series B (Methodological) 57, no. 1 (1995): 289-300.

False Discovery Rate[7] for FWER control

- **How to control FWER to a low level and still avoid collision?**
- Assume, R of total P channels are declared as occupied among which V of them are falsely declared as occupied.
- Then the false discovery rate (FDR) is defined as,

$$FDR = E\left[\frac{V}{R}\right] \tag{8}$$

- FDR controls **the number of false discoveries made only among the total discoveries** - less conservative

[7] Benjamini, Yoav, and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the Royal statistical society: series B (Methodological) 57, no. 1 (1995): 289-300.

False Discovery Rate[7] for FWER control

- **How to control FWER to a low level and still avoid collision?**
- Assume, R of total P channels are declared as occupied among which V of them are falsely declared as occupied.
- Then the false discovery rate (FDR) is defined as,

$$FDR = E\left[\frac{V}{R}\right] \tag{8}$$

- FDR controls **the number of false discoveries made only among the total discoveries** - less conservative
- BH procedure can be used by FC to make the final decision about the state of the channel - helps to reduce the fraction of missed slots

[7] Benjamini, Yoav, and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the Royal statistical society: series B (Methodological) 57, no. 1 (1995): 289-300.

CSS in non-stationary environment

Handling non-stationary environment

- Algorithm should be able to *discount* [8] the past observations in favor of more recent observations

[8] Raj, Vishnu, and Sheetal Kalyani. "An aggregating strategy for shifting experts in discrete sequence prediction." arXiv preprint arXiv:1708.01744 (2017).

Handling non-stationary environment

- Algorithm should be able to *discount* [8] the past observations in favor of more recent observations
- Reduces the importance of distant past observations using an exponential weighing scheme

[8] Raj, Vishnu, and Sheetal Kalyani. "An aggregating strategy for shifting experts in discrete sequence prediction." arXiv preprint arXiv:1708.01744 (2017).

Handling non-stationary environment

- Algorithm should be able to *discount* [8] the past observations in favor of more recent observations
- Reduces the importance of distant past observations using an exponential weighing scheme
- **Update for dHedge:**

$$w_{ji}(n+1) \leftarrow w_{ji}(n)^{\gamma} \beta^{l_{ji}(n)}. \tag{9}$$

where $0 \leq \gamma \leq 1$ is the discounting factor

[8] Raj, Vishnu, and Sheetal Kalyani. "An aggregating strategy for shifting experts in discrete sequence prediction." arXiv preprint arXiv:1708.01744 (2017).

Experimental validation

Three different CRN configurations

1. *Good signal condition* (GSC): An area of 1×1 km^2 with 10 SUs. Approximately 78% of the SUs have $P_d > 0.95$.
2. *Medium signal condition* (MSC): An area of 8×8 km^2 with 50 SUs. Approximately 55% of the SUs have $P_d > 0.95$.
3. *Bad signal condition* (BSC): An area of 8×8 km^2 with 10 SUs. Just about 1% of the SUs have $P_d > 0.95$.

Metrics

$$\text{Fraction of SU collision} = \frac{\sum_{n=1}^{N} \sum_{s \in \mathcal{S}} \mathbb{I}[s \text{ incured a collision at } n]}{\sum_{n=1}^{N} \sum_{s \in \mathcal{S}} \mathbb{I}[s \text{ attempts a transmission at } n]}$$

$$\text{Fraction of PU collision} = \frac{\sum_{n=1}^{N} \sum_{j} \mathbb{I}[\text{collision observed in } c_j \text{ at } n]}{\sum_{n=1}^{N} \sum_{j} \mathbb{I}[c_j \text{ is busy at } n]}$$

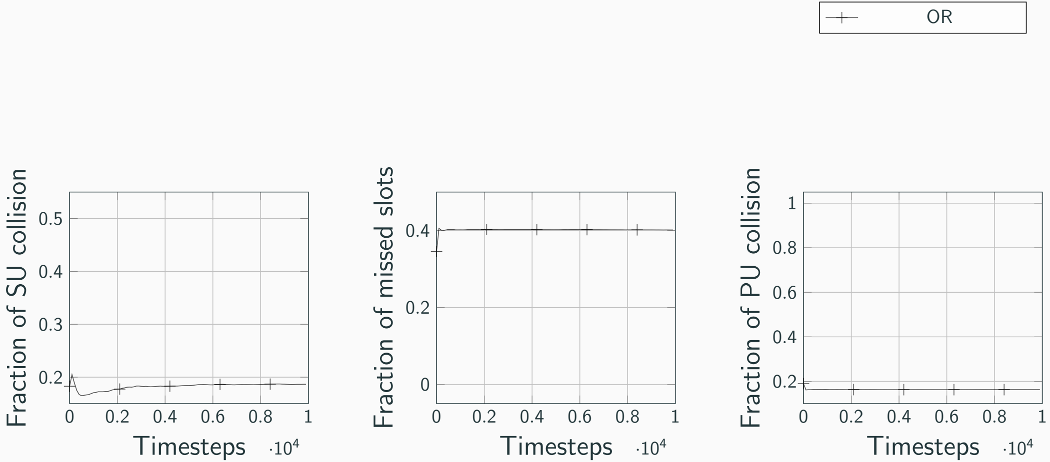
Metrics

$$\text{Fraction of missed slots} = \frac{\sum_{n=1}^{N} \sum_{j} \mathbb{I}_{[\text{Incurred a false alarm at } c_j]}}{\sum_{n=1}^{N} \sum_{j} \mathbb{I}_{[c_j \text{ is idle at } n]}}$$

Number of sensing averaged over all SUs in the network

- comes down when SUs are selectively deactivated to sense the channel
- indicator of energy spent in sensing

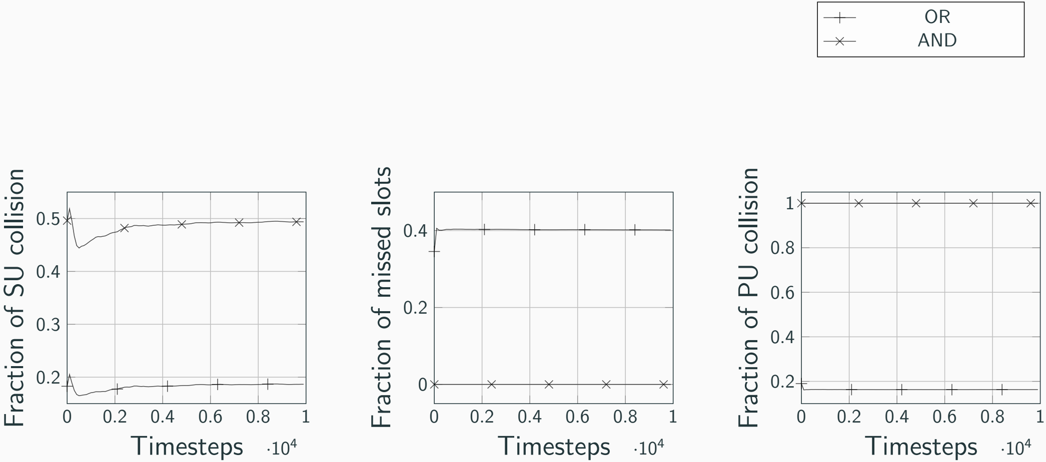
Results on SU collision, PU collision and missed slots in BSC



(1.1) Fraction of SU packet collision **(1.2)** Fraction of missed idle slots **(1.3)** Observed interference at PU

Figure 1: Comparison of proposed Hedge-HC, Hedge-SC and Perceptron-SC with traditional OR, AND and CV for BSC

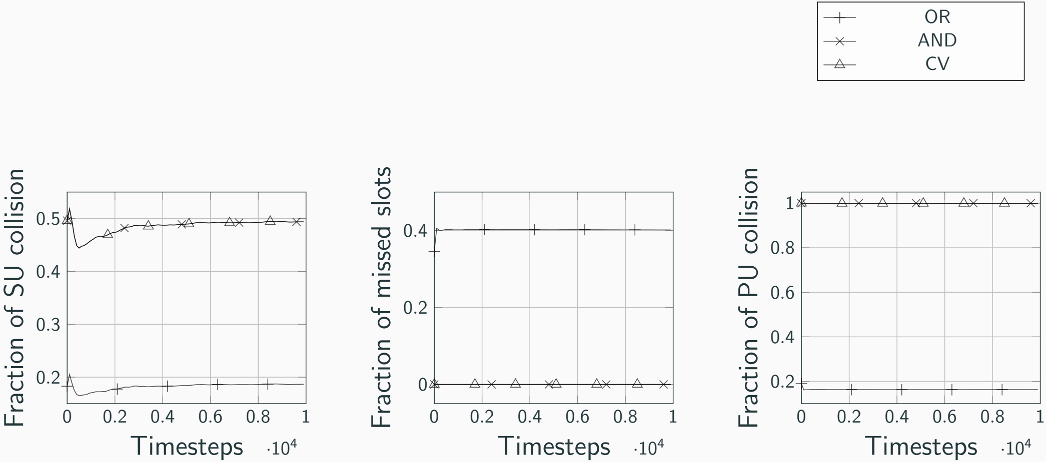
Results on SU collision, PU collision and missed slots in BSC



(1.1) Fraction of SU packet collision **(1.2)** Fraction of missed idle slots **(1.3)** Observed interference at PU

Figure 1: Comparison of proposed Hedge-HC, Hedge-SC and Perceptron-SC with traditional OR, AND and CV for BSC

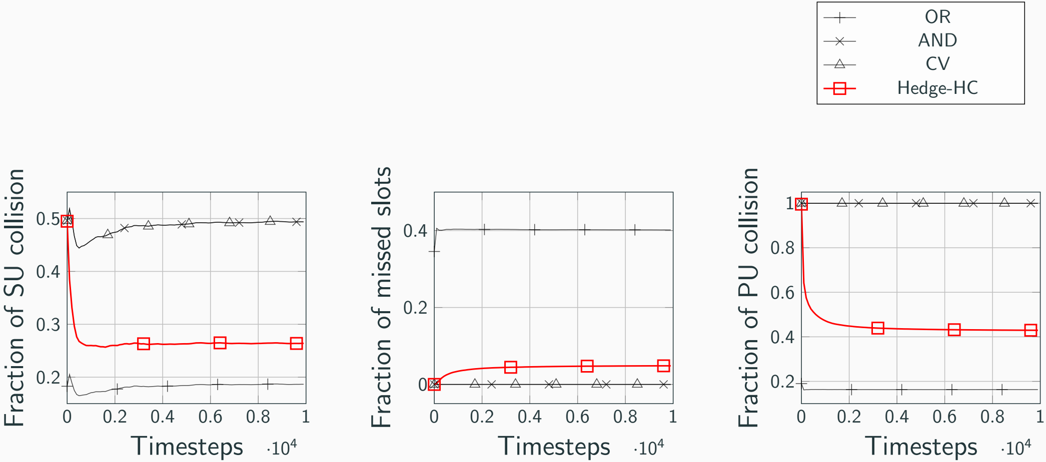
Results on SU collision, PU collision and missed slots in BSC



(1.1) Fraction of SU packet collision **(1.2)** Fraction of missed idle slots **(1.3)** Observed interference at PU

Figure 1: Comparison of proposed Hedge-HC, Hedge-SC and Perceptron-SC with traditional OR, AND and CV for BSC

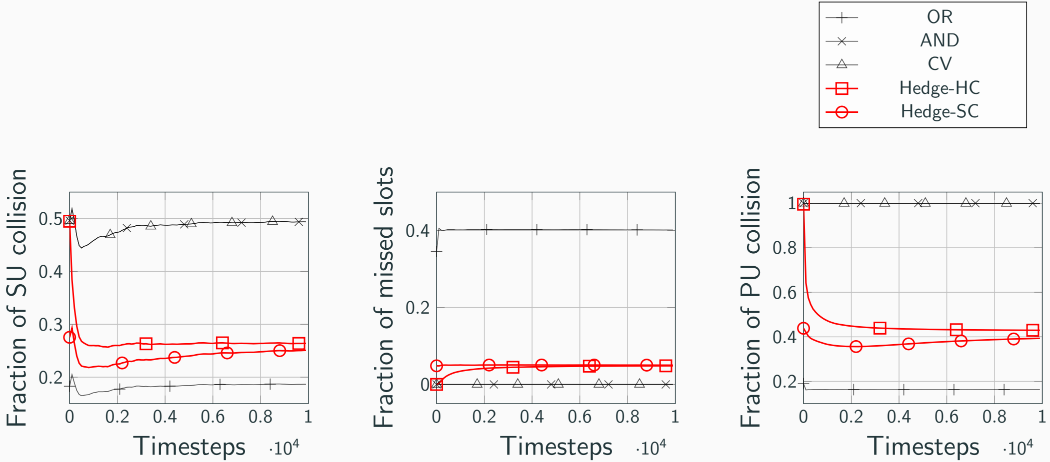
Results on SU collision, PU collision and missed slots in BSC



(1.1) Fraction of SU packet collision **(1.2)** Fraction of missed idle slots **(1.3)** Observed interference at PU

Figure 1: Comparison of proposed Hedge-HC, Hedge-SC and Perceptron-SC with traditional OR, AND and CV for BSC

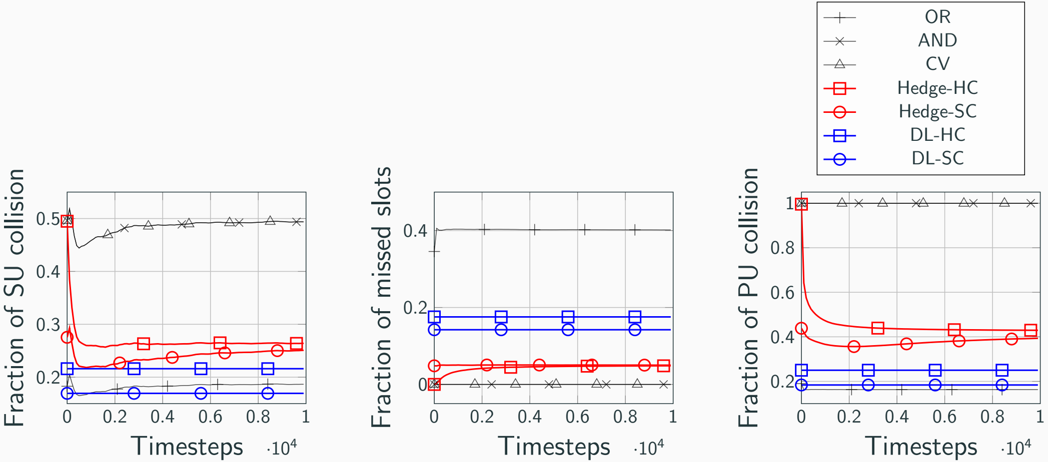
Results on SU collision, PU collision and missed slots in BSC



(1.1) Fraction of SU packet collision **(1.2)** Fraction of missed idle slots **(1.3)** Observed interference at PU

Figure 1: Comparison of proposed Hedge-HC, Hedge-SC and Perceptron-SC with traditional OR, AND and CV for BSC

Results on SU collision, PU collision and missed slots in BSC



(1.1) Fraction of SU packet collision **(1.2)** Fraction of missed idle slots **(1.3)** Observed interference at PU

Figure 1: Comparison of proposed Hedge-HC, Hedge-SC and Perceptron-SC with traditional OR, AND and CV for BSC

Hard Combining vs Soft Combining

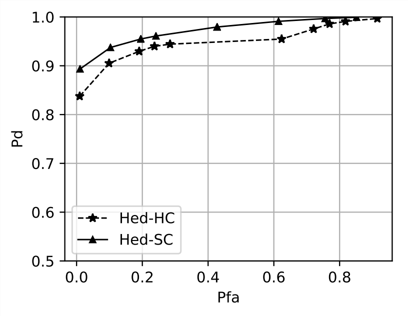
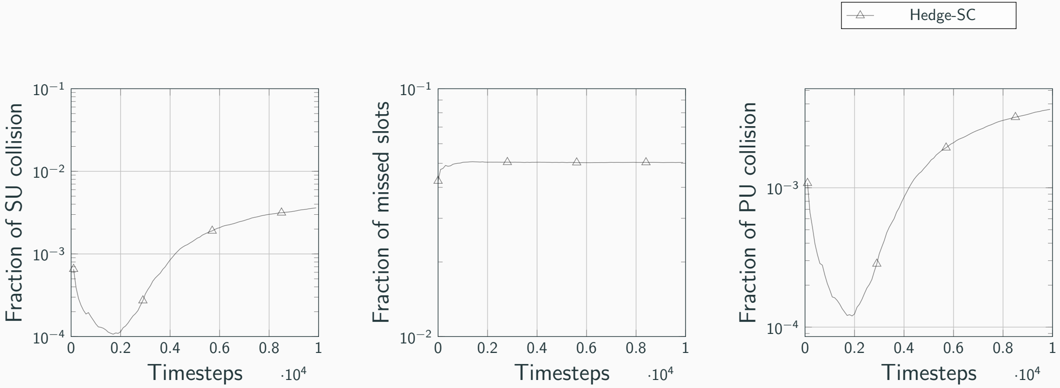


Figure 2: ROC at medium signal condition

- P_d and P_{fa} are empirically calculated at fusion center
- ROC for "Hed-SC" lies above "Hed-HC"

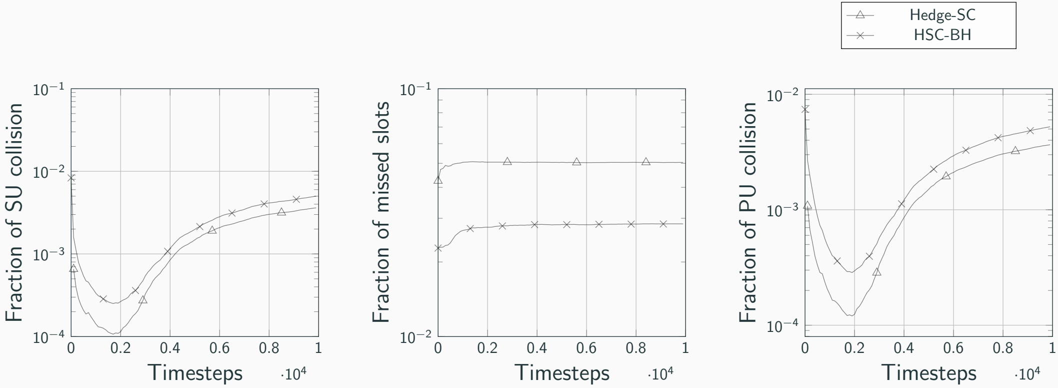
Results on BH method for FDR control in MSC



(3.1) SU packet collision **(3.2)** Fraction of missed idle slots **(3.3)** Observed interference at PU

Figure 3: BH method for FDR control for MSC

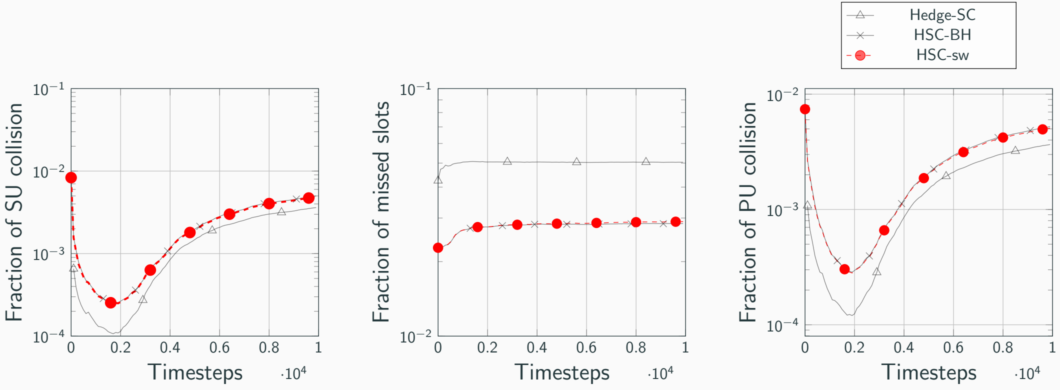
Results on BH method for FDR control in MSC



(3.1) SU packet collision **(3.2)** Fraction of missed idle slots **(3.3)** Observed interference at PU

Figure 3: BH method for FDR control for MSC

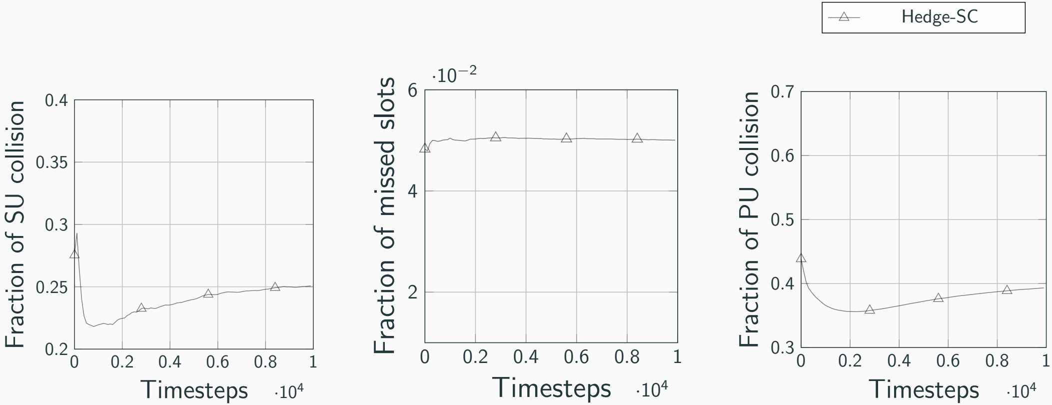
Results on BH method for FDR control in MSC



(3.1) SU packet collision **(3.2)** Fraction of missed idle slots **(3.3)** Observed interference at PU

Figure 3: BH method for FDR control for MSC

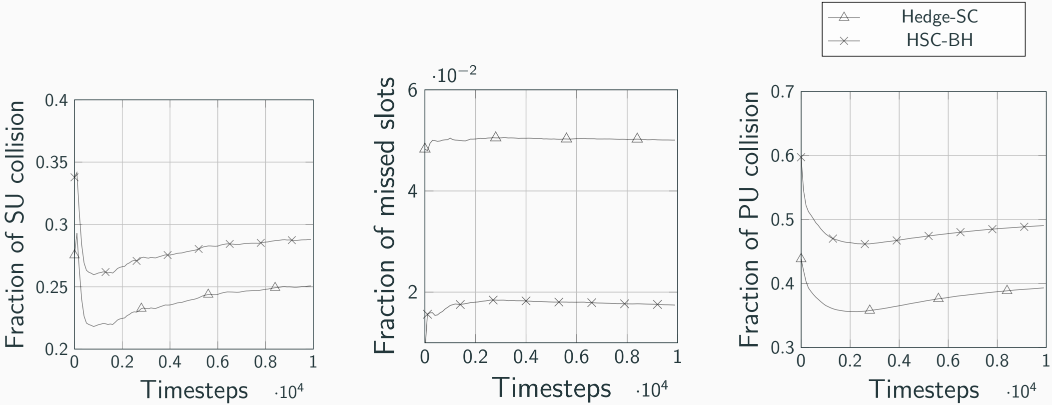
Results on BH method for FDR control in BSC



(4.1) SU packet collision **(4.2)** Fraction of missed idle slots **(4.3)** Observed interference at PU

Figure 4: BH method for FDR control for BSC

Results on BH method for FDR control in BSC



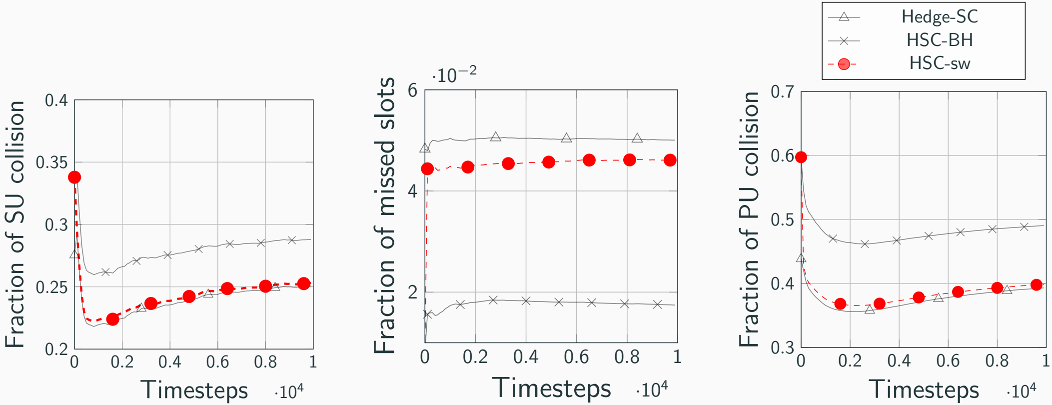
(4.1) SU packet collision

(4.2) Fraction of missed idle slots

(4.3) Observed interference at PU

Figure 4: BH method for FDR control for BSC

Results on BH method for FDR control in BSC



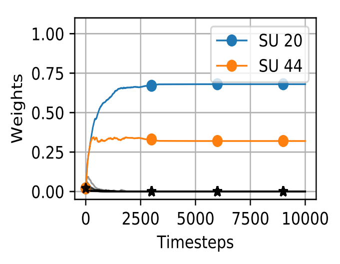
(4.1) SU packet collision

(4.2) Fraction of missed idle slots

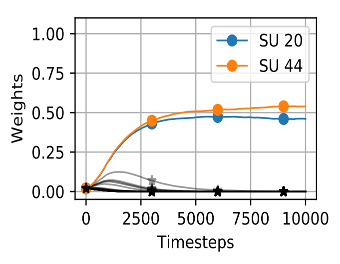
(4.3) Observed interference at PU

Figure 4: BH method for FDR control for BSC

Weight evolution



(5.1) Hedge hard combining

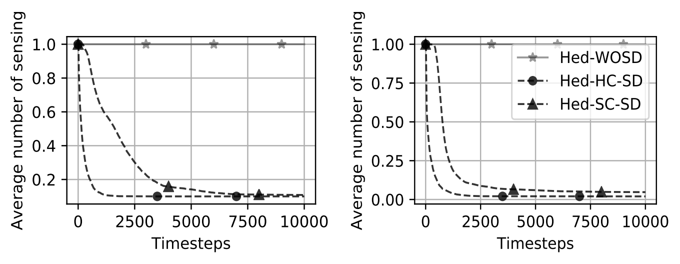


(5.2) Hedge soft combining

Figure 5: Weight evolution in MSC for stationary channel condition

- Selectively deactivate SUs whose observation are not important

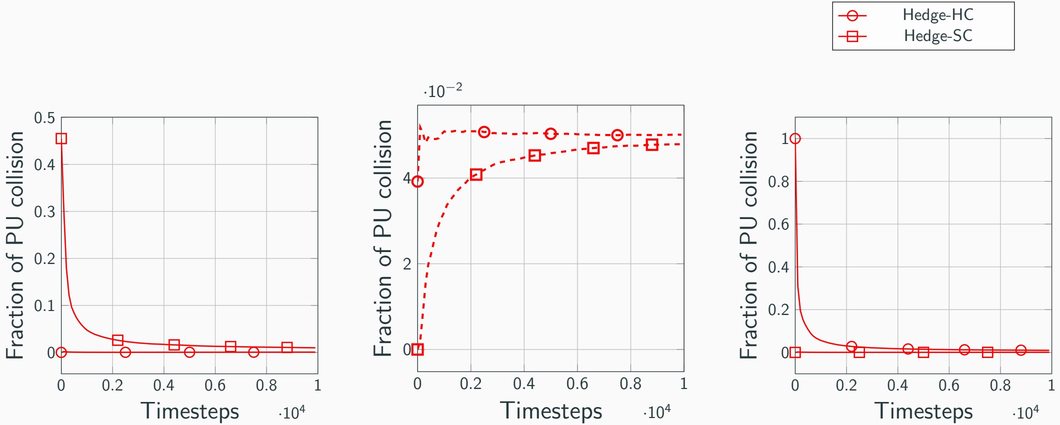
Saving energy by selectively enabling devices



(6.1) BSC **(6.2)** MSC

Figure 6: Average number of sensing per SU per timestep

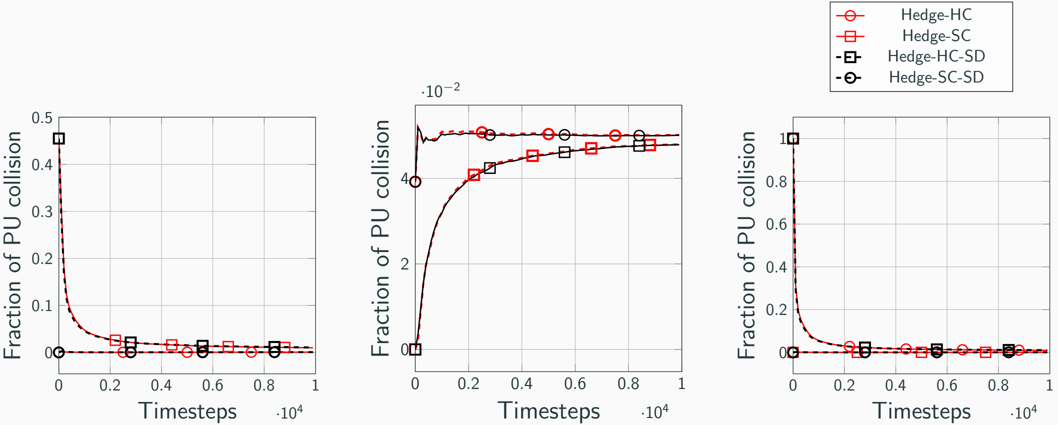
Selective deactivation of poor-performing detectors



(7.1) SU packet collision **(7.2)** Fraction of missed idle slots **(7.3)** Observed interference at PU

Figure 7: Comparison of metrics with selective deactivation of poor-performing detectors in MSC

Selective deactivation of poor-performing detectors



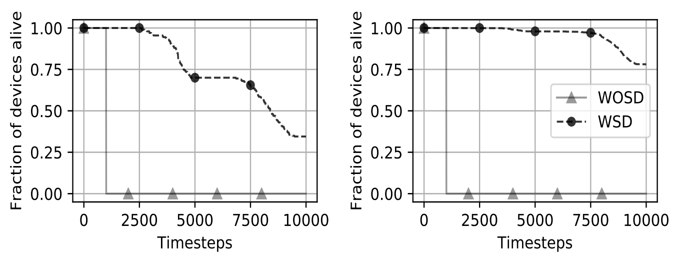
(7.1) SU packet collision

(7.2) Fraction of missed idle slots

(7.3) Observed interference at PU

Figure 7: Comparison of metrics with selective deactivation of poor-performing detectors in MSC

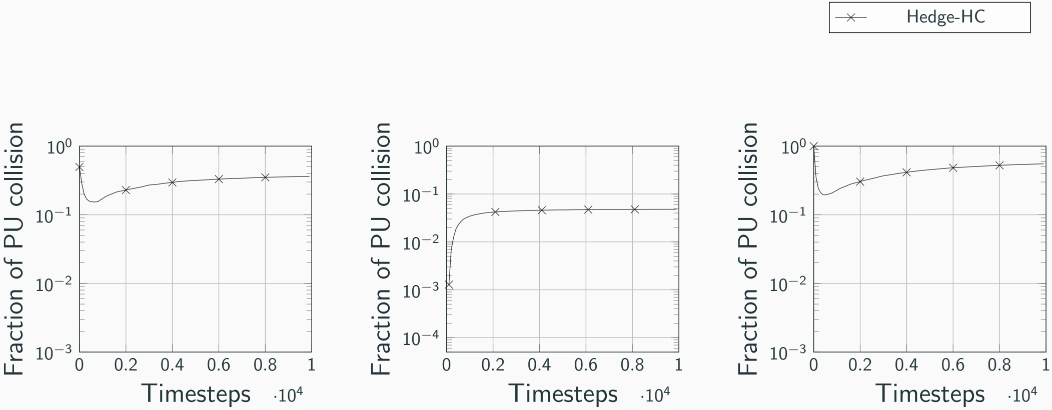
Fraction of IoT devices left with energy



(8.1) BSC **(8.2)** MSC

Figure 8: Comparison of fraction of devices left with energy

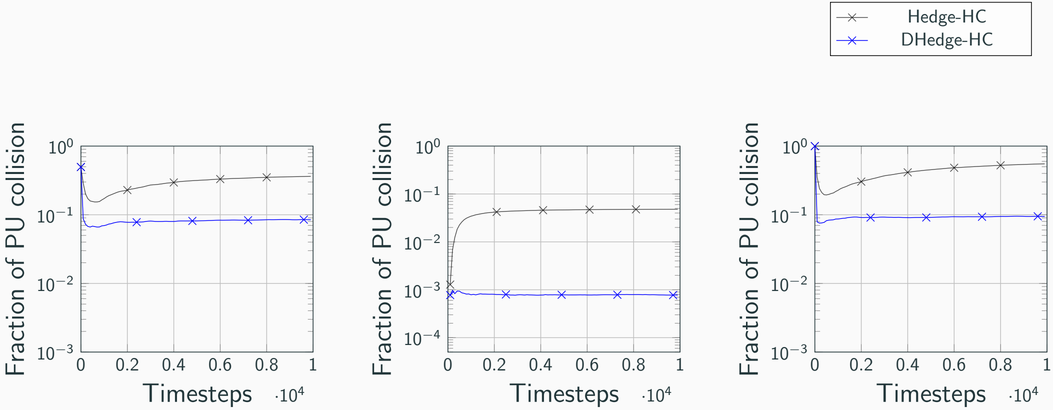
Results on non-stationary environment



(9.1) SU packet collision **(9.2)** Fraction of missed idle slots **(9.3)** Observed interference at PU

Figure 9: Both PUs and SUs are mobile in Medium Signal Condition

Results on non-stationary environment



(9.1) SU packet collision

(9.2) Fraction of missed idle slots

(9.3) Observed interference at PU

Figure 9: Both PUs and SUs are mobile in Medium Signal Condition

Results on non-stationary environment

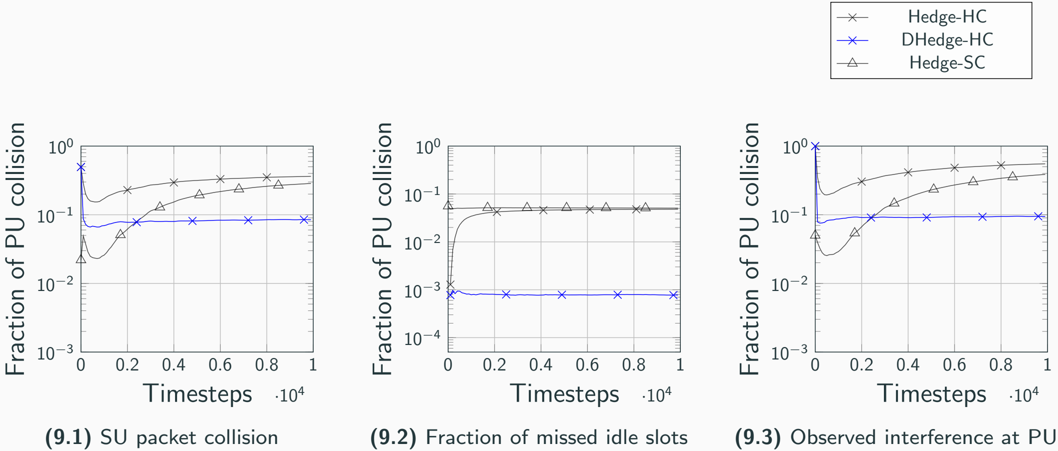
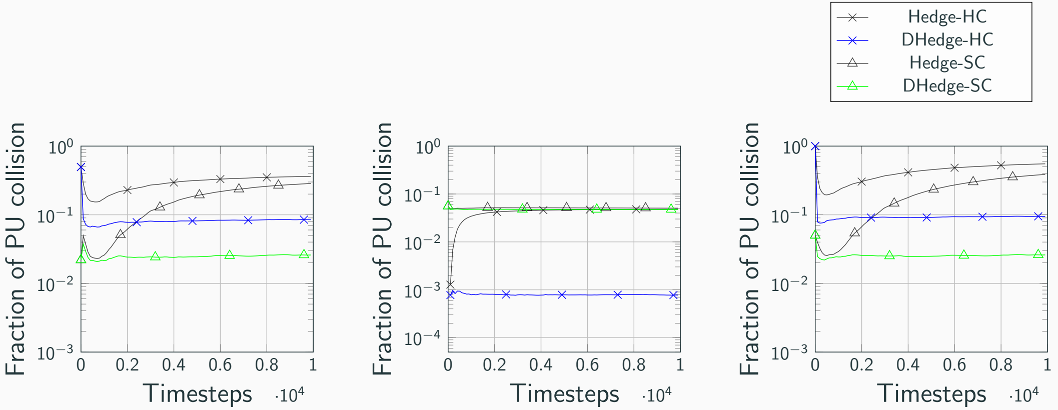


Figure 9: Both PUs and SUs are mobile in Medium Signal Condition

Results on non-stationary environment



(9.1) SU packet collision **(9.2)** Fraction of missed idle slots **(9.3)** Observed interference at PU

Figure 9: Both PUs and SUs are mobile in Medium Signal Condition

Conclusion

We presented an online learning framework for collaborative spectrum sensing.

1. It learns to combine the information based on past performances
2. Extends the battery-life
3. Can be easily scaled to networks experiencing wide variety of signal conditions and large number of devices
4. Handles situations where devices drop-out of the network randomly
5. Equally applicable for non-stationary environments

Publications

Related publication

1. Nayak, Nancy, Vishnu Raj, and Sheetal Kalyani. "Leveraging online learning for CSS in frugal IoT network." IEEE Transactions on Cognitive Communications and Networking 6, no. 4 (2020): 1350-1364.

Publications

Other accepted publications

2. Vikas, Devannagari, Nancy Nayak, and Sheetal Kalyani. "Realizing neural decoder at the edge with ensembled bnn." IEEE Communications Letters 25, no. 10 (2021): 3315-3319.
3. Nayak, N., Raj, V. and Kalyani, S. "[Re] A comprehensive study on binary optimizer and its applicability." ReScience C: 6 pp. #9 (2).
4. Raj, Vishnu, Nancy Nayak, and Sheetal Kalyani. "Deep reinforcement learning based blind mmwave MIMO beam alignment." IEEE Transactions on Wireless Communications 21, no. 10 (2022): 8772-8785.

Publications under-review

1. Nayak, Nancy, and Sheetal Kalyani. "Rotate the ReLU to implicitly sparsify deep networks." arXiv preprint arXiv:2206.00488 (2022).
2. Nayak, Nancy, Sheetal Kalyani, and Himal A. Suraweera. "A DRL Approach for RIS-Assisted Full-Duplex UL and DL Transmission: Beamforming, Phase Shift and Power Optimization." arXiv preprint arXiv:2212.13854 (2022).
3. Shankar, Nitin Priyadarshini, Deepsayan Sadhukhan, Nancy Nayak, and Sheetal Kalyani. "Binarized ResNet: Enabling Automatic Modulation Classification at the resource-constrained Edge." arXiv preprint arXiv:2110.14357 (2021).

Publications

Pre-prints

1. Raj, Vishnu, Nancy Nayak, and Sheetal Kalyani. "Understanding learning dynamics of binary neural networks via information bottleneck." arXiv preprint arXiv:2006.07522 (2020).
2. Nayak, Nancy, Thulasi Tholeti, Muralikrishnan Srinivasan, and Sheetal Kalyani. "Green DetNet: Computation and memory efficient DetNet using smart compression and training." arXiv preprint arXiv:2003.09446 (2020).
3. Sharma, Akshay, Nancy Nayak, and Sheetal Kalyani. "BayesAoA: A Bayesian method for Computation Efficient Angle of Arrival Estimation." arXiv preprint arXiv:2110.07992 (2021).

Thank you!

Detection method at every SU/IoT device

- Channel detection method: Neyman-Pearson (NP) detector
- Let, $H_0 : x(n) = n_0(n)$ and $H_1 : x(n) = s(n) + n_0(n)$ where $s(n)$ is transmitted signal from BS and $n_0(n)$ is the noise
- The statistics $e_{ji}(n) = (1/N) \sum_{s=0}^{N-1} x^2[s]$, sum of square of N IID Gaussian RVs, is compared with a threshold η at each time-step n to detect channel state of c_j
- The detection hypothesis can be written as,

$$\frac{e_{ji}(n)}{\sigma^2} \sim \chi_N^2 \quad \text{under } H_0 \tag{10}$$

$$\frac{e_{ji}(n)}{\sigma_s^2 + \sigma^2} \sim \chi_N^2 \quad \text{under } H_1 \tag{11}$$

where σ^2 is noise variance and σ_s^2 is signal variance

Detection method at every SU/IoT device

- Probability of false alarm

$$P_{fa} = P\left(\frac{e_{ji}(n)}{\sigma^2} > \frac{\zeta}{\sigma^2}; H_0\right). \tag{12}$$

- According to NP criterion, for a targeted P_{fa}, the threshold to detect a channel

$$\zeta = \sigma^2 . Q^{-1}_{\chi^2_N}(P_{fa}) \tag{13}$$

  where N is the number of samples used for energy detector
- The prediction of SU s_i about the channel c_j at n^{th} time step is $d_{ji}(n)$ given by,

$$d_{ji}(n) = 1_{[e_{ji}(n) \geq \zeta]} \tag{14}$$

- When $d_{ji} \in \{0, 1\}$ is sent from SUs to FC, each element in observation matrix is $o_{ji}(n) = d_{ji}(n) \quad \forall i, j$ - **Hard decision combining**
- When *soft* information is sent by SUs to FC then each element in observation matrix is $o_{ji}(n) = e_{ji}(n) \in \mathbb{R} \quad \forall i, j$ - **Soft decision combining**

Hedge soft combining (Hed-SC) the observations

- Approximate $\tilde{f}_j(n)$ with another gamma distribution $\Gamma(k_j, \theta_j)$ using the moment matching technique where $\tilde{f}_j(n)$ can be written as

$$\tilde{f}_j(n) = \sum_{i=1}^{S} p_{ji}(n)\tilde{\psi}(n) = \sum_{i=1}^{S} \Gamma\left(\frac{N}{2}, 2p_{ji}(n)\eta_{ji}^2\right), \tag{15}$$

- By equating first moment

$$k_j \times \theta_j = \sum_{i=1}^{S} \frac{N}{2} \times 2p_{ji}(n)\sigma^2 = N\sigma^2 \text{ as } \sum_{i=1}^{S} p_i = 1 \tag{16}$$

- By equating the variance,

$$k_j \times \theta_j^2 = \sum_{i=1}^{S} \frac{N}{2} \left(2p_{ji}\sigma^2\right)^2. \tag{17}$$

Hedge soft combining the observations

- Comparing (16) and (17), we get $\theta_j = 2\sigma^2 \sum_{i=1}^{S} p_{ji}^2$ and $k_j = \frac{N}{2\sum_{i=1}^{S} p_{ji}^2}$
- Given a probability of false alarm requirement, the threshold for detection at time instant n, $\gamma_j(n)$, for channel c_j can be calculated from,

$$\gamma_j = Q^{-1}_{\Gamma(k_j,\theta_j)}(P_{fa}) \tag{18}$$

- Final decision on j^{th} channel c_j: $f_j(n) = \text{busy if } \tilde{f}_j(n) \geq \gamma_j$ else free

Controlling the FDR at α with BH procedure

- Considering $\tilde{f}_j(n)$ as the observed combined soft information at FC, the corresponding p-value P_j is

$$P_j = Q_{\Gamma(k_j,\theta_j)}(\tilde{f}_j(n)). \tag{19}$$

- Order p-values: $P_{(1)} \leq P_{(2)} \leq ... \leq P_{(P)}$
- $H_{(j)}$: the null hypothesis corresponding to $P_{(j)}$ $\forall j \in \mathcal{P}$
- Let k be the largest j for which,

$$P_{(j)} \leq \frac{j}{P}\alpha \tag{20}$$

  then reject all $H_{(j)}$ for $j = 1, 2, ..., k$
- BH procedure helps to reduce the fraction of missed slots for transmission
- Switch between traditional Hedge and BH procedure to attain best of both the worlds

Perceptron Inspired Online Learning

- A version of perceptron which fits into the need for CSS
- Algorithm maintains a weight vector $\mathbf{w}_j$ of length S for each channel c_j
- At FC, the combined expert decision

$$\tilde{f}_j(n) = \sum_{i=1}^{S} w_{ji}(n) o_{ji}(n). \tag{21}$$

  is compared with a threshold γ_j^p
- The perceptron algorithm in the CSS setting learns the weights $w_{ji}(n)$ and the intercept γ_j^p of the hyperplane

$$\sum_{i=1}^{S} w_{ji}(n) o_{ji}(n) - \gamma_j^p = 0. \tag{22}$$

Perceptron Inspired Online Learning

- Whenever an expert s_i makes a mistake when the actual channel state c_j is *busy*, the weight of that expert is updated as,

$$w_{ji}(n+1) \leftarrow w_{ji}(n) + \rho \cdot o_{ji}(n). \tag{23}$$

- When the actual channel state c_j is *idle* and an expert s_i makes a mistake, the weight of that expert is updated as,

$$w_{ji}(n+1) \leftarrow w_{ji}(n) - \rho \cdot o_{ji}(n). \tag{24}$$

Perceptron Inspired Online Learning

- As the observations come from χ^2_N or Γ distribution, the combined expert decision $\tilde{f}_j(n) \sim \sum_{i=1}^{S} w_{ji}(n)\Gamma\left(\frac{N}{2}, 2\eta^2\right)$ is a weighted sum of Γ distributions.

$$\tilde{f}_j(n) \sim \sum_{i=1}^{S} w_{ji}(n)\Gamma\left(\frac{N}{2}, 2\eta^2\right) \sim \sum_{i=1}^{S} \Gamma\left(\frac{N}{2}, 2w_{ji}(n)\eta^2\right) \tag{25}$$

- Closed form expression or proper approximation for the above distribution is not available - a histogram fitting method
- Let $\mathcal{H}$ be the normalized histogram of samples drawn from (25), for a predefined P_{fa},

$$\gamma_j^p = Q_{\mathcal{H}}^{-1}(P_{fa}). \tag{26}$$

Perceptron Inspired Online Learning

- Hyperplane (22) can be written as,

$$\sum_{i=1}^{S} w_{ji}(n) o'_{ji}(n) = 0 \tag{27}$$

where

$$o'_{ji}(n) = o_{ji}(n) - \frac{\gamma_j^p}{S \times w_{ji}(n)}. \tag{28}$$

- So the update equations are given as

$$\text{On false positive: } w_{ji}(n+1) \leftarrow w_{ji}(n) + \rho \cdot o'_{ji}(n) \tag{29}$$

and

$$\text{On false negative: } w_{ji}(n+1) \leftarrow w_{ji}(n) - \rho \cdot o'_{ji}(n). \tag{30}$$

dPerceptron

- **Update for dPerceptron**
- Channel c_j is *busy*, but s_i makes a mistake:

$$w_{ji}(n+1) \leftarrow \gamma w_{ji}(n) + \rho \cdot o_{ji}(n). \tag{31}$$

- Channel c_j is *idle*, but s_i makes a mistake:

$$w_{ji}(n+1) \leftarrow \gamma w_{ji}(n) - \rho \cdot o_{ji}(n) \tag{32}$$

Deep Learning based approach

- Enough data - train a fully connected network offline - use the trained network to predict the channel occupancy state.
- Input: A flattened vector of $\mathbf{O}(n)$
- Output: A sigmoid layer of dimension $P \times 1$ whose j^{th} element denotes the probability of j^{th} channel being occupied
- Loss function: Mean Square Error between the output vector of the deep network and the actual GT of the channels

Parameter	Value
Number of hidden layers	3
Number of neurons in each hidden layer	$2PS$
Activation	tanh
Learning rate of batch wise GD	0.001
Batch size	20

Table 1: Parameters for offline training

Traffic model

The PU traffic is modeled using the Hyper-exponential distribution (HED) as suggested in [**5506438**]. Both ON time and OFF time of PU are modelled using an M component HED random variable X as

$$f_X^{HED} = \sum_{k=1}^{M} p_k f_{Y_k}(x),$$

where each Y_k is exponentially distributed with rate λ_k, and p_k is the weight given to k^{th} component with $\sum_{k=1}^{M} p_k = 1$. The simulation parameters used are given in Table 2.

Parameter	Value	Parameter	Value
Number of PUs	10	P_{fa}	0.05
Working frequency of PUs	6 GHz	Packet loss	0.05
PU Transmit Power	0 dB	Hedge HC: β	0.88
No.of HED components	3	Hedge SC: β	0.99
λ	$(0, 500]$	Perceptron: ρ	0.80