Minimum-Cost Channel Selection in Wearables

Body Sensor Networks (BSN) Conference 2024

Ramesh Kumar Sah, Nooshin Taheri Chatrudi, Stephanie Marita Carpenter, Hassan Ghasemzadeh

Presenter: Nooshin Taheri

10/2/2024

Background

- **Wearable Sensor Systems**
  - Operate in **environments** with limited computational power and memory.
  - Increasingly used in **digital health applications**.

- **Sensor Channel Selection**
  - A key optimization problem in **resource-constrained wearable systems**.

  - Channel selection involves the **identification and removal** of channels that provide a **negative or negligible contribution** to the task.
    - $C(n,k) = \frac{n!}{\mathrm{k!\times(n-k)!}}$

  - A channel selection algorithm combines a **search technique** to find new channel subsets and an **evaluation method** to assess the performance of the selected subset.

Background

Gap in Current Research

- Prior methods only consider **performance criteria** and ignore the **cost** of the channel subset in decision-making.

Novelty

- Present two backward search algorithms that address this gap by **minimizing cost** while ensuring performance meets a **lower bound** of the performance function.

MINIMUM COST CHANNEL SELECTION (MCCS)

- **select the best combination of sensors** that keeps the system **cost-efficient** while still performing the task at an acceptable level (better than λ).
- λ is the threshold for the minimum acceptable performance level.

- n sensor channels $C_n = \{c_1, c_2, \ldots, c_n\}$
- Cost of each channel: $W_n = \{w_1, w_2, \ldots, w_n\}$
- The MCCS problem is to minimize the total cost:

$$\text{Minimize} \sum_{i=1}^{n} w_i a_i \qquad a_i \in \{0, 1\} \qquad f(c_1 a_1, c_2 a_2, \ldots, c_n a_n) \geq \lambda$$

- w_i is the normalized cost of selecting channel c_i.
- Normalized cost is obtained for all channels given:
  - $\sum_{i=1}^{n} w_i = 1.$

Branch and Bound Channel Selection

How It Works:

- **Start with all channels** and progressively remove one channel at a time.
- Each subset of channels is evaluated based on **performance** and **cost**.

- The channel subset selection problem is to find the subset $c_1^*, \ldots c_s^*$ to discard such that:

$$f(C_{\bar{s}}^*) = \max\ f(C_s^*)$$
$$\text{and } W(C_{\bar{s}}^*) = \min\ W(C_s^*)$$

- **Monotonicity Assumption**:
    - **Removing channels reduces performance**.

$$f_n(c_1, c_2, \ldots, c_n) \geq f_{n-1}(c_1, c_2, \ldots, c_{n-1}) \geq \cdots \geq f_1(c_1)$$

- If a subset performs below **λ**, no further subsets with fewer channels are considered → **Prune suboptimal subsets**

Challenges of Branch and Bound Algorithm

The branch and bound algorithm assume monotonicity in performance, **performance decreases as channels are removed**, which may not always be true.

In the **worst case**, the algorithm must evaluate **all possible channel subsets**, which leads to an **exponential increase in computation time**.

Due to these limitations, a **greedy algorithm** is proposed to provide a **faster, sub-optimal solution** that balances performance and cost without the exhaustive search.

Greedy Channel Selection

- **Goal**:
  - To **find a suboptimal subset of channels** that balances **performance** and **cost**.
  - The focus is on **quickly** finding a good solution, even if it's not the absolute best (optimal).
- **How It Works**:
  - Starts with the **full set of channels**.
  - **Removes one channel at a time**, choosing the channel that results in the smallest impact on performance and cost.
  - The algorithm calculates a **penalty** for each potential removal based on two factors:
    - **Performance Loss**: How much performance drops if a channel is removed.
    - **Cost**: The cost saved by removing that channel.

$$penalty = \alpha \times (1 - f(C - c)) + (1 - \alpha) \times W(C - c)$$

Differences of two methods

Branch and Bound:

- **Global Search**: It searches **all possible subsets** of channels, and it **prunes** (cuts off) branches that don't meet the performance threshold.
- **Optimal Solution**: Because it explores a larger part of the solution space (even if some branches are pruned), it guarantees finding the **best possible subset** of channels.
- **Time-Consuming**: Since it has to explore a large number of combinations (or subsets), even with pruning, it can take a lot of time and computational resources for large sets of channels.

Greedy Channel Selection:

- **Local Search**: It doesn’t look at all possible subsets. Instead, it removes **one channel at a time** based on the penalty function, choosing the best channel to remove **at each step**.
- **Suboptimal Solution**: Because it doesn’t explore all subsets and makes decisions step by step, it might not find the absolute best subset, but it finds a **good enough solution** quickly.
- **Faster**: The greedy method is faster because it doesn’t need to explore the entire set of possible channel combinations. It simply makes **local decisions** at each step (removing one channel), instead of trying to evaluate many combinations or prune branches like in branch and bound.

Cost Model

- Cost model defines the cost of a channel based on some input parameters such as:

  1) computation and memory requirements that are directly related to sampling frequency
  2) power requirement
  3) sensing requirement
  4) usability and interpretability cost
  5) manufacturing cost
  6) other cost

- generate the cost for each channel using a simple heuristic based on the **sampling frequency** of the sensor channel. Sensor channels with **higher sampling frequency** are assigned **a larger cost** and vice-versa.

Dataset

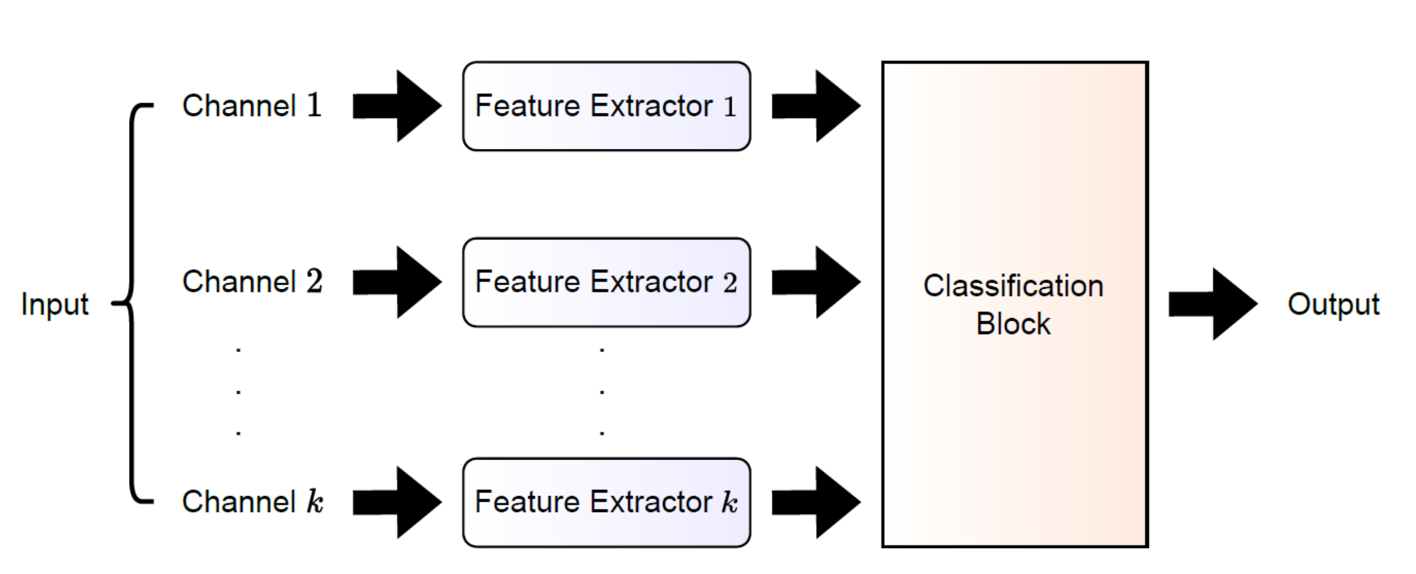
Dataset	EEG Mental Task Dataset	PAMAP2 Human Activity Recognition Dataset
Purpose	Detect mental arithmetic task using EEG signals	Recognize human activities from wearable sensors
Number of Channels	23 EEG channels	27 sensor channels (9 from each body location: chest, wrist, ankle)
Sampling Frequency	500 Hz	100 Hz
Tasks/Activities	Binary classification of mental arithmetic tasks	Multi-class classification (7 classes)
Data Format	60-second artifact-free EEG segments subdivided into 10-sec windows	Signals subdivided into 30-sec windows with 15-sec overlap
Sensors Used	Neurocom EEG system	Accelerometer, Gyroscope, Magnetometer
Number of Participants	36 participants	9 participants

Model Architecture

- The model uses a **1D CNN architecture** to evaluate the selected channel subsets.
- Each sensor channel in the subset is assigned to a **separate feature extraction block (two 1D convolutional layers)**.
- The outputs from all feature extractors are combined in the classification block(two **fully connected layers**).

Training Details:

- The **ReLU activation function** is used
- The Softmax **activation** function is used in the output layer
- The model is trained for **100 epochs** using the **Adam optimizer**.
- The **learning rate** is set to **0.001**.
- The **cross-entropy loss function** is used to evaluate classification performance.



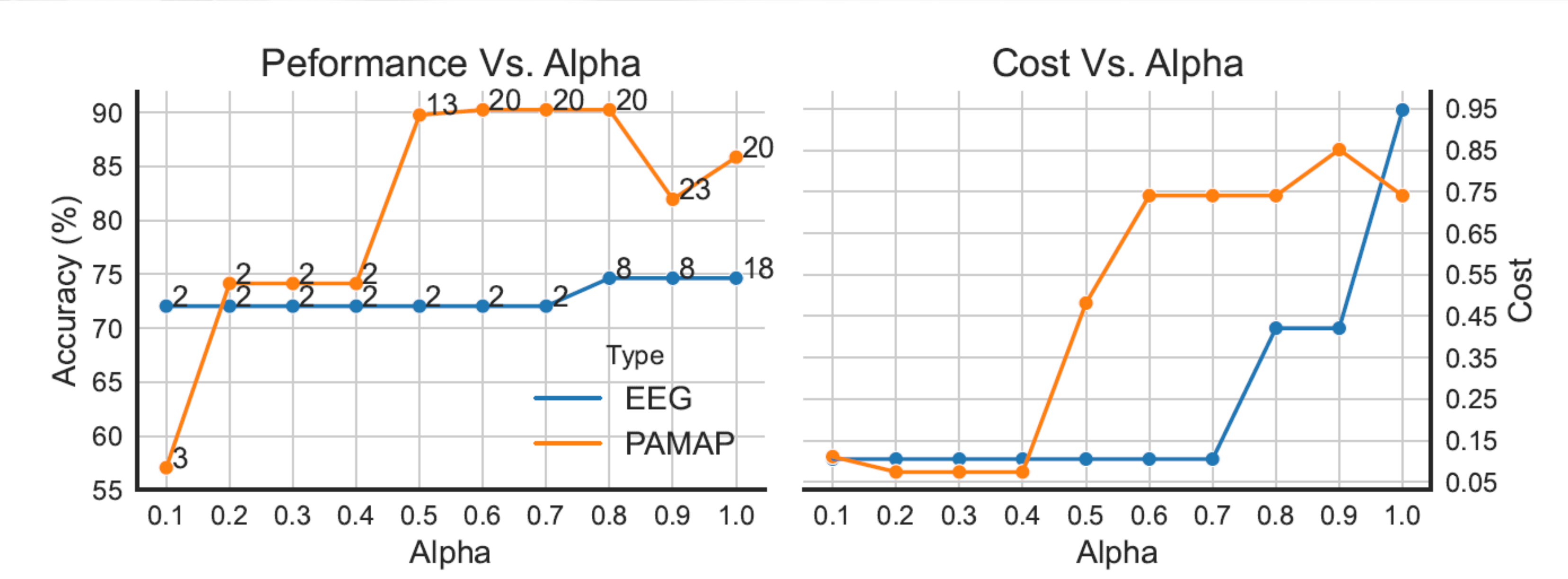
Channel Subset Selection

Dataset	Method	Selected Subset	Accuracy (%)	Cost	Penalty	Cost Savings	Baseline Performance Accuracy (%)	Performance Threshold (λ)
EEGMAT	B&B	FP1	70.31	0.043	0.169	95.7%	73.48	0.7 (70%)
	Greedy	(C3, F3)	72.33	0.086	0.191	91.4%	73.48	0.7 (70%)
PAMAP2	B&B	2 channels	51.22	0.074	0.282	92.6%	59.02	0.5 (50%)
	Greedy	13 channels	89.75	0.4814	0.2919	51.8%	59.02	0.5 (50%)

α = 0.5 for greedy channel search and **measured the performance in terms of accuracy of the trained model**.

Effects of Alpha

- In practice, minimizing cost might be more important than maximizing performance and vice-versa.



- **Higher α values** lead to **better performance** but at the expense of **higher costs**.
- **Lower α values** reduce **costs** by using fewer channels, but the **accuracy** might drop.
- The **number of selected channels** increases as **α** increases, which is why accuracy improves with larger α values.

Conclusion

- Proposed and validated **two channel selection algorithms** for finding the optimal subset of channels with **minimum cost**.
- Algorithms can be applied to **real-life applications** for optimizing sensor systems while maintaining **performance guarantees**.
- **Branch and Bound** enables **dynamic channel selection** during runtime:
  - If channels in the optimal subset become unavailable, the **next best subsets** can be used to keep the system operational.
- The **evaluation scheme** is **model-agnostic**, meaning it can work with any learning algorithm, not just CNNs.