Power-Aware Computing in Wearable Sensor Networks: An Optimal Feature Selection

Hassan Ghasemzadeh, Member, IEEE, Navid Amini, Student Member, IEEE, Ramyar Saeedi, Student Member, IEEE and Majid Sarrafzadeh, Fellow, IEEE

Abstract—Wearable sensory devices are becoming the enabling technology for many applications in healthcare and well-being, where computational elements are tightly coupled with the human body to monitor specific events about their subjects. Classification algorithms are the most commonly used machine learning modules that detect events of interest in these systems. The use of accurate and resource-efficient classification algorithms is of key importance because wearable nodes operate on limited resources on one hand and intend to recognize critical events (e.g., falls) on the other hand. These algorithms are used to map statistical features extracted from physiological signals onto different states such as health status of a patient or type of activity performed by a subject. Conventionally selected features may lead to rapid battery depletion, mainly due to the absence of computing complexity criterion while selecting prominent features. In this paper, we introduce the notion of power-aware feature selection, which aims at minimizing energy consumption of the data processing for classification applications such as action recognition. Our approach takes into consideration the energy cost of individual features that are calculated in real-time. A graph model is introduced to represent correlation and computing complexity of the features. The problem is formulated using integer programming and a greedy approximation is presented to select the features in a power-efficient manner. Experimental results on thirty channels of activity data collected from real subjects demonstrate that our approach can significantly reduce energy consumption of the computing module, resulting in more than 30% energy savings while achieving 96.7% classification accuracy.

Index Terms—Real-time systems and embedded systems, Emerging technologies, Wearable Computers, Healthcare, Human-centered computing, Signal processing, Ubiquitous computing, Low-power design, Optimization

1 INTRODUCTION

Recent technology advances have led to the development of different sensing, computing, and communication artifacts that are becoming an essential part of our daily lives forming pervasive and mobile sensory platforms. These ubiquitous systems have proved to be effective in a number of domains ranging from medical [1] and well-being [2] to military [3] and smart vehicles [4]. A special class of these platforms is wearable sensor networks [5] whose computational elements are tightly coupled with the human body. These networks are known as enabling technologies for many applications such as remote patient monitoring and personalized healthcare [6], gaming and sports [6], maintenance, production and process support [7], [8].

There are a number of challenges that must be overcome to fully implement wearable sensor networks including high costs, package size and weight limitations, power efficiency and battery lifetime, memory storage, connectivity, ease of use, reliability, application level accuracy, security, and privacy issues [9]. Since wearable sensor networks are battery-operated and may have critical and life-saving purposes, power efficiency is considered the most challenging design consideration in their real life deployment. In medical applications, wearable units are mainly used for remote and continuous patient monitoring [10], and therefore, their power consumption needs to be minimized to guarantee their long term operation and infrequent battery charge or replacement [10].

Existing studies do not address the battery lifetime issue directly, since the length of related pilot studies is relatively short and they are carried out in controlled or semi-naturalistic environments. Tasks performed constantly such as sensing, processing (e.g., classification), and wireless transmissions incur significant energy expenditure in wearable nodes. As such, battery energy, if not properly managed, can be completely used up within hours and recharging is often impractical when the nodes are in-use.

In wearable sensor networks, where raw data is simply streamed to the gateway, the largest energy consumer is the radio subsystem (e.g., wearable ECG monitors) [11], with the processing unit only required for formatting the data according to the utilized communication protocol. On the other hand, for wearable systems with on-node processing (e.g., movement monitoring and wearable EEG monitors), the processing subsystem is the most energy consuming subsystem. In such systems, a signal with a lower bit rate will be transmitted to the gateway after processing. This necessitates further optimization of the computing units’ power...
consumption in order to prolong the lifetime of the entire system. This second group of wearable systems often employ embedded signal processing and machine learning blocks that use sensor data (e.g., acceleration of body segments) to extract relevant information (e.g., types of movements) about their subjects.

Signal processing and machine learning methods are defined by the application and vary in complexity. Current techniques for wearable systems especially those concerning activity recognition aim at using a reduced feature set to characterize the monitored signals in a real-time fashion while meeting wearable systems' memory and processing constraints. Statistical, time/frequency domain, and heuristic features comprise the reduced feature set [12] at the end of the feature selection process. Feature extraction is often time consuming and can deplete the battery if an exhaustive feature set is extracted from the signal. Although classification accuracy is the ultimate measure of relative performance, for wearable platforms, there should be a mechanism to gauge the amount of useful information extracted for a given energy budget. Hence, real-life deployment of wearable platforms necessitates the incorporation of energy components into performance measures.

The traditional feature selection algorithms focus on specific criteria that finds redundancy and relevance in a given feature set. This approach is generally acceptable in conventional algorithms such as image processing and text mining, which run on highly powerful computers. These techniques, however, do not take into consideration computing complexity of individual features. That is, they allocate equal weights to features of varying complexity. In wearable systems this approach is not effective, as these systems have limited processing power and need to operate in real-time. For instance, while computing ‘peak-to-peak’ amplitude of a signal segment demands a linear mathematical function, calculation of Root-Mean-Square (RMS) power is quadratic, and a Fast Fourier Transform (FFT) requires \( (n \log n) \) operations to complete. As the signal segment grows in length, complexity of these functions that need to run in real-time can result in infeasible signal processing and machine learning algorithms. None of the feature selection techniques studied in the past takes into consideration the computing complexity of the selected features, an important measure in designing wearable monitoring systems. Furthermore, none of the existing power-aware schemes in embedded system design has dealt with feature selection algorithms and how the energy saving mechanisms can cleverly prevent some features from being accounted for in classification mechanisms.

This study embodies the innovation of the notion of power-aware feature selection; we model the problem of energy optimal feature set and prove that it constitutes a computational problem that is NP-hard and finally we provide an approximation to find the appropriate minimum cost feature set. Real human motion data sets are used in order to verify the efficacy of our approximation scheme. Our work takes special interest in classification applications, where physiological signals from human body are used to classify different states of a subject [13], [14], [15], [16], [17], [18]. Examples of such applications include human action recognition and fall detection using accelerometer and gyroscope sensors, and arrhythmia detection from ECG signals. In the classification process, a set of representative features, such as ‘signal amplitude’ and ‘root mean square’ power, are typically extracted from the measured signal prior to performing the classification task.

Our approach combines three criteria, namely, feature relevance, feature redundancy, and computing complexity and builds a minimum cost feature set without sacrificing the classification accuracy. Generally, our contributions in this paper can be summarized as follows: 1) we introduce the notion of power-aware feature selection by adding a new design dimension, computing complexity, to the feature selection problem; 2) we propose a graph model that embodies information regarding classical feature selection, relevance and redundancy; 3) we present an Integer Linear Programming (ILP) formulation of our optimization problem and provide a greedy approximation to solve it; 4) we use real data collected from healthy individuals in a controlled experimental setting using a wearable sensor network designed for physical movement monitoring to evaluate the effectiveness of our power-aware approach in managing battery usage without compromising classification accuracy to the extent possible.

2 Related Work

Power efficiency in wearable platforms is usually at odds with all other design objectives such as performance and reliability [19]. Many techniques to improve power efficiency can incur performance, power, or classification accuracy penalties [20].

In the context of real-time computing, Zhao et al. [21] explore the energy-reliability trade-off. Their approach minimizes the system-level energy consumption while satisfying a certain reliability target in the task scheduler. More specifically, their approach specifies the optimal number of recoveries to deploy together with task-level processing frequencies to minimize the energy consumption while achieving the target reliability and meeting the deadline constraints.

Until recently, power awareness and classification accuracy have been studied independently in the context of wireless sensor networks and wearable computing. However, there is an interesting trade-off between a system’s power efficiency and classification accuracy as both goals compete for processing resources. There exists a growing body of related research that implicitly or explicitly deals with such a trade-off.

Zappi et al. [22] proposed a method, called dynamic sensor selection, which trades off classification accuracy for battery lifetime in wireless sensor networks. Dynamic
sensor selection investigates the impact of varying number of sensors on activity classification accuracy and minimizes the number of nodes necessary to obtain a given classification accuracy for activity recognition. The method was tested by recognizing ten different activities of assembly line workers in a car production environment. French et al. [23] propose selective sampling strategies for activity recognition which can adapt dynamically at run-time. Significant savings in energy consumption is achieved via efficient selection of sensor sampling rates for optimal energy-accuracy tradeoff. Similarly, Yan et al. [24] focus on adaptively varying the sampling rate to achieve the optimal energy consumption above the baseline CPU power consumption.

In contrast to the above existing works, we argue that aside from sampling frequency and the number of active sensors, a larger set of features can also lead to significant energy overheads and therefore, it can violate the tight energy constraints of battery-operated wearable systems. Investigators in [20], [24], [25] have developed models to decide whether to perform classification tasks and inference algorithms, locally or remotely. In the first model, such decisions are made in reaction to system changes (e.g., battery levels or foreground processing load). Changes in the user’s activities are determining factors in assessing where and to what extent the classification needs to be executed in the second model. Finally, a multi-criteria decision theory is used by the designers of the third model to distribute the computational tasks between a wearable node and the server. Our work primarily differs from these models in that the feature set they use is selected a priori and has no understanding of power efficiency.

Increasing the number of sensor nodes, the sampling frequency of each sensor, and the number of features to extract will all result in improved classification accuracy at the expense of a larger energy overhead. Our study makes a connection between two broad research topics, feature selection and power-aware design, that are disjointedly explored in machine learning and embedded system design domains, respectively.

Feature selection is of immense importance in various domains such as bioinformatics, signal processing, image processing, text categorization, data mining, pattern recognition, and medical diagnosis. In the context of wearable sensor networks, Palmerini et al. [26] implemented a wrapper feature selection technique, where they evaluate the feasibility and the impact of accelerometer based posturography in evaluation of Parkinson’s disease (PD) [27] and its progression; in this regard, they implement a feature selection technique aimed at identifying a subset of measures that best discriminates between early-mild PD subjects and control subjects. They investigate measures derived from acceleration signals including tremor measures, postural acceleration measures, and postural displacement measures [26]. The tight energy constraints of battery-operated wearable systems call for the development of energy-aware signal processing methods to preserve energy [28], [23]. Depending on the application and signals involved, it may be more useful and energy-efficient to trade off radio communications energy for processing energy, or vice versa. Dynamic sensor selection which trades off classification accuracy for the system’s battery lifetime is another power saving approach. This method, proposed in [22], minimizes the number of nodes necessary to obtain a given classification ratio for activity recognition.

Studies on energy optimization are numerous, and address all levels of design [21], [29], [30]. With the increasing importance of computing in wearable systems, the power and energy consumed by the processing subsystem have become a key consideration. The goal of energy-aware computing is not merely to make algorithms run as fast as possible, but also to minimize energy requirements for computation, by treating energy as a constrained resource like memory or disk. Energy-aware computing is a research area that is concerned with finding new ways on how to save energy in processing subsystems using novel algorithms, applications or architectures.

None of the mentioned approaches tackles challenging issues arising when multiple features need to run simultaneously and share dynamic and scarce energy resources in a single wearable node. In contrast to those work in the literature mostly focusing on adaptive sampling and computation distribution, our approach accounts for the energy cost of individual features that are calculated in real-time.

3 Preliminaries
Before presenting our power-aware approach for feature selection, several basic concepts are reviewed in this section, with an emphasis on the architecture of wearable platforms, their signal processing, and information extraction from physiological signals.

3.1 System Architecture
A wearable sensor network, also called body sensor network, is composed of several body-worn sensor nodes, a gateway, and a back-end server as shown in Fig. 1. Each sensor node is attached to the body to sample
and process physiological signals and transmit partial results to the gateway. A sensor node usually has several sensors for capturing different user’s states (e.g., body acceleration), an embedded processor to perform limited signal processing and information extraction, and a radio for data transmissions. The gateway is a more powerful unit such as a cell phone or a PDA that performs data fusion and makes conclusions about current state of the user (e.g., ‘walking’, ‘running’, and ‘sitting’). The results are further transmitted, through the Internet, to a back-end server for storage, further processing, and clinical decision support. This processing chain can be closed by a feedback loop from the back-end server to the user. For example, a feedback can suggest changes in patient’s medication dosage due to lack of sufficient physical activity or if a Parkinson’s patient is experiencing increased tremor.

The focus of this study is primarily on power optimization of the wearable sensor nodes, where stringent constrained sensor units are used to process physiological signals in real-time. Other elements such as the gateway and back-end server are usually powerful in terms of computing power.

Each sensor node in a wearable system processes sensor readings through a chain of embedded signal processing modules, each of which is intended to extract partial information from the signal and reduces the amount of sampled data. Fig. 2 illustrates a typical signal processing flow for applications targeting classifying physiological signals into user’s states. This processing flow is detailed in Section 3.2. In physical movement monitoring applications, readings from motion sensors such as accelerometers, magnetometers, and gyroscopes undergo signal processing to classify human actions such as ‘walking’, ‘sit to stand’, and ‘jumping’. Signal processing and pattern recognition for wearable sensor networks require a learning phase during which the system is trained based on a training data set. During this phase, parameters of the system used in different signal processing modules are adjusted. Such parameters define the training model and will be used throughout the operation of the system. Feature selection (as specified by dashed lines in Fig. 2) is only part of the learning process.

The signals that are sampled by each sensor node are first passed through a filter to reduce high frequency noise. Although complexity of the required digital filter is highly dependent on the application area of the wearable platform, a simple moving average filter is usually sufficient to alleviate the effect of the noise in movement monitoring applications which is the pilot application for our study.

Segmentation is intended to identify ‘start’ and ‘end’ points of the actions that are being classified. In fact, motion sensors sample capture human movements constantly, streaming continuous actions. Thus, it is essential to partition the signal into segments of interest. Each segment will be further processed for the purpose of action recognition, which maps the signal segment onto a specific action.

Feature extraction module is responsible for calculating statistical and morphological characteristics of the signal segment. Prominent feature are known a priori as they are defined in the learning phase. Features represent different attributes of the signal such as ‘peak-to-peak amplitude’, ‘standard deviation’, and ‘mean value’. Features extracted from different sensors form a feature vector that will be used for classification.

Finally, a classification algorithm is utilized to determine the current state of the user (e.g., type of actions being performed by the user). An example of classification algorithms is k-Nearest-Neighbor (kNN), which has proved effective in movement monitoring applications [31], [32].

### 3.3 Feature Selection Criteria

In general, feature selection [33], [34], [35] aims to find an optimal set of features from an exhaustively extracted set of features. In most cases, the optimality of the solution is defined by two criteria, relevance and redundancy [36]. While relevance criterion focuses on eliminating features that are irrelevant to the classification task, redundancy criterion uses inter-feature correlation measures to eliminate features with high correlation.

By removing most irrelevant and redundant features from the data, feature selection helps to improve the analysis of high-dimensional spaces. This speeds up the learning process by highlighting the important features and understanding how they interrelate. Feature selection methods can be categorized into wrapper, filter, and embedded methods [34]. Filter based methods [37] rank the features as a pre-processing step prior to the learning algorithm, and select those features with higher ranking scores. Wrapper methods [38] score the features using the learning algorithm that will eventually be employed. Embedded methods [39] combine feature selection with learning algorithm and hence, they are specific to a model. Please refer to [35] to learn about the most recent and frequently used methods of each category.

To better present the concept of relevance and redundancy, we give a simple example adapted from [36]. Assume the feature set \( \{f_1, f_2, \ldots, f_n\} \) is given, where a classification decision is made based on a binary function
Fig. 2. Per-node signal processing flow

\[ C = g(f_1, f_2) \]

Further, let \( f_2 = f_3 \) and \( f_4 = f_5 \). Clearly, \( f_1 \) is a required feature as it is necessary for classification and is not redundant with respect to other features. Also, features \( f_4 \) and \( f_5 \) are redundant because they are not used to make a classification decision. We note, however, that \( f_2 \) and \( f_3 \) can be interchanged as they are equally informative. Thus, an optimal feature set is either \( \{f_1, f_2\} \) or \( \{f_1, f_3\} \).

### 3.4 Energy Model

Over the past decade, researchers have proposed various techniques to estimate the processing energy of embedded systems. Such techniques can be classified into the following categories:

1) **Physical level energy analysis** [40]: this methodology investigates the switching activity of all circuit nodes of the processor architecture. A detailed description of the processor architecture at the transistor level is needed in order to pursue physical level energy analysis [41].

2) **Architectural level energy analysis** [42]: this method requires modeling the standard components of the processing unit such as registers, load/store queues, and functional units using higher abstraction levels (i.e. architecture level), which decreases the computational complexity as compared to the previous method [41].

3) **Functional level energy analysis** [43], [44]: functional level analysis effectively combines the low modeling and computational effort of a functional level energy model and the higher accuracy of an instruction level energy model.

4) **Instruction level energy analysis** [45], [46]: At the instruction level, the energy consumption of each instruction from of the instruction set of a processor can be determined by means of low level simulations or physical measurements. The instruction level energy analysis technique allows for accounting for the inter-instruction effects by measuring the energy consumption of various groups of processor instructions.

In this work we chose to perform instruction level energy analysis, since it provides the fundamental information needed to estimate the energy cost of different programs (i.e. features in our battery-operated cyber physical systems). The main shortcomings of instruction level energy analysis techniques include: 1) they do not provide any insight on the energy consumption breakdown within the processor core, which is viewed as a black box, and 2) they usually do not account for the power consumed in the memory system, which often exceeds that of the processor itself. To address the latter shortcoming, we jointly consider the energy consumption due to the memory address mode of operands as well as the energy overhead between two instructions. In doing so, we presume that pipeline stalls and cache misses do not alter the results if the effects of those would be considered in the model. As such, instruction level energy model provides more accurate results for cache-less processors, such as TI MSP430.

As an example to contrast memory’s energy consumption with that of processor, we consider TI MSP430’s ‘mov’ instruction. The ‘mov’ instruction consumes roughly three times more energy in the “absolute to register” mode than in the “indirect to register” mode.

Fig. 3. Proposed approach for optimal power-aware feature selection

### Problem Statement

Fig. 3 illustrates our approach for power-aware feature selection. Given an initial feature set, \( F \), irrelevant features are first eliminated from subsequent processing. A redundancy analysis is then performed to find features that are strongly correlated and can be substituted once power efficiency of the processing is taken into account. To this end, a graph model, called redundancy graph, is constructed based on the correlation information obtained in redundancy analysis. Finally, the graph model is used to solve an optimization problem, called Min-
imum Cost Feature Selection (MCFS), meant for finding the optimal feature set.

4.1 Redundancy Graph

Our relevance and redundancy analyses, which provide inputs to construct the graph model and formulate the problem, are based on the concept of symmetric uncertainty:

Definition 1 (Symmetric Uncertainty): The symmetric uncertainty between two discrete random variables $X$ and $Y$ is given by:

$$ U(X, Y) = \frac{2I(X, Y)}{H(X) + H(Y)}, $$

(1)

where $H(X)$ and $H(Y)$ represent the entropy of random variables $X$ and $Y$, respectively, and $I(X, Y)$ denotes the information gain between the two variables. $I(X, Y)$ is further defined by:

$$ I(X, Y) = H(X) - H(X|Y) $$

(2)

The symmetric uncertainty is actually the normalized information gain and is always between 0 and 1, where $U=1$ means that knowing the value of either variable can completely predict the other variable, and $U=0$ indicates that the two variables are completely independent.

We note that the symmetric uncertainty is a measure of correlation between two random variables. Major advantage of this measure against other measures, such as correlation coefficient, is that the symmetric uncertainty can capture non-linear correlation between variables and therefore, is a safe measure for our feature analysis study.

Definition 2 (Irrelevant Feature): Given an exhaustive set of $n$ features $F={f_1, f_2, \ldots, f_n}$ and a set of human actions $A={a_1, a_2, \ldots, a_k}$ to be classified, a feature $f_i$ is irrelevant to the classification task if

$$ \min_j (U(f_i, a_j)) < \lambda_R, $$

(3)

where $\lambda_R$ (relevance threshold) is a design parameter.

Relevance analysis will eliminate features that are irrelevant to the action recognition. The remaining $m$ features ($m < n$) are subject to redundancy analysis whose main goal is to find strongly correlated features.

Definition 3 (Strongly Correlated Features): Two features $f_i$ and $f_k$ are considered to be strongly correlated if

$$ U(f_i, f_k) > \lambda_D, $$

(4)

where $\lambda_D$ (redundancy threshold) is a design parameter.

The output of the redundancy analysis is a set of feature pairs in the form of $(f_i,f_k)$, which are strongly correlated and either of them can be eliminated according to the correlation analysis. However, these features are further examined for computing complexity using the graph model presented in the following section.

Definition 4 (Redundancy Graph): Given $m$ relevant features introduced by the relevance analysis and a set of feature pairs $\{f_j,f_k\}$ generated according to the redundancy analysis, an undirected graph $G=(V,E,W)$ is called redundancy graph, where $V$ is a set of $m$ vertices, $V = \{u_1, u_2, \ldots, u_m\}$ associated with the $m$ relevant features, $E=\{e_1, e_2, \ldots, e_r\}$ is the set of $r$ feature pairs that are strongly correlated, and $W=\{w_1, w_2, \ldots, w_m\}$ is the set of weights, assigned to the vertices, denoting the computing cost associated with each feature.

4.2 Problem Definition

We now present a simple example to motivate our idea of finding the optimal feature set using MCFS. Assume that ten features construct our exhaustive set of features, represented by $F = \{f_1, f_2, \ldots, f_{10}\}$. Furthermore, assume that the relevance analysis decides to eliminate five features and hence, the redundancy graph will contain five features, i.e. $R = \{f_1, f_2, f_3, f_4, f_5\}$. The redundancy graph with each vertex representing one of the five features is shown in Fig. 4. Note that the processing cost attributed to each feature is represented by the weight of each vertex, e.g., $w_1$ is the processing cost of $f_1$.

Let all the weights be equal to one unit, that is $W = \{w_1, w_2, w_3, w_4, w_5\} = \{1, 1, 1, 1, 1\}$. In this case, MCFS treats all features equally and thus, the optimal feature set consists of two vertices, specifically $f_1$ and $f_3$. However, if we modify the weight set to $W = \{10, 1, 1, 1, 1\}$, MCFS gives more consideration to vertices with lower weights and accordingly, features $f_4$ and $f_5$ will be favored over $f_1$. In the recent scenario the optimal feature set will contain three vertices, i.e. $f_4$, $f_5$, and $f_3$. As such, the computation energy cost will be decreased from 11 units to 3 units.

We present an optimization problem that finds optimal feature set taking into account relevance, redundancy and complexity criterion discussed previously. The problem takes a redundancy graph as input and outputs a subset of relevant features that are optimal in terms of computing complexity and are not redundant.

Problem 1: Given a redundancy graph $G=(V,E,W)$, the minimum cost feature selection problem is to find a subset of vertices that are not dominated by any other vertex in the graph and their total cost is minimized.
4.3 Problem Formulation

Assume that $a_{ij}$ is a given binary that encodes existence of edges in the redundancy graph:

$$a_{ij} = \begin{cases} 1, & \text{if } (u_i, u_j) \in V \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (5)

and $x_i$ is a binary variable which determines whether or not a vertex $u_i$ is chosen as a member of the final vertex set:

$$x_i = \begin{cases} 1, & \text{if vertex } u_i \text{ is chosen} \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (6)

The corresponding ILP formulation for the MCFS problem is as follows:

Minimize $\sum_{i=1}^{m} w_i x_i$ \hspace{1cm} (7)

subject to:

$$\sum_{j=1}^{m} x_i a_{ij} \geq 1 \text{ } \forall i \in \{1, 2, \ldots, m\}$$  \hspace{1cm} (8)

$$x_i \in \{0, 1\}$$  \hspace{1cm} (9)

The objective function in (7) is to minimize the total cost of the selected vertices (i.e. those with $x_i=1$). The constraint (8) guarantees that each selected vertex is adjacent to at least one more vertex and the constraint in (9) ensures that the variable $x_i$ takes only binary values.

4.4 Problem Complexity

The MCFS problem is equivalent to the Minimum Cost Dominating Set (MCDS) problem. The MCDS problem is proved to be NP-hard by reduction from the Weighted Set Cover (WSC) problem.

**Theorem 1:** The MCFS problem is NP-hard.

**Proof:** We prove that the Minimum Cost Feature Selection (MCFS) problem is NP-hard by reduction from Weighted Set Cover (WSC) problem. Let $(S,U,W)$ be an instance of the WSC problem with the universe $U$ and the family of subsets $S = \{S_1, S_2, \ldots, S_n\}$ each associated with a weight $w_i$ from the set $W = \{w_1, w_2, \ldots, w_m\}$. Construct a graph $G=(V,E,W)$ as follows: For each set $S_i \in S$, draw a vertex $u_i$ (associated with feature $f_i$) and draw and edge $(u_i,u_j)$ for every pair of $(u_i,u_j) \in S_i$. This forms the vertex set $V = \{u_1, u_2, \ldots, u_n\}$ as well as the edge set $E$. Furthermore, assign to each vertex $u_i$ (associated with the set $S_i$) the weight value $w_i$ as given by the set $W$. Now if $C = \{S_i : i \in D\}$ is a feasible solution of the weighted set cover problem, then $D$ is also a solution to the MCFS problem. \hfill \□

4.5 Greedy Approach

Our greedy algorithm for solving MCFS problem is presented in Algorithm 1. For each vertex $u_i$ in the redundancy graph, the algorithm first finds all adjacent vertices $(V_i)$. It then finds the best candidate vertex to include in the final vertex set $(\Omega)$. The best candidate is the one with maximum profit. A maximum profit vertex is the one with maximum value of “cardinality of $V_i$ divided by vertex cost $w_i$”. The intuition behind selecting such a vertex is that it has a large number of adjacent vertices and a small cost. Finally, the algorithm adds the candidate vertex $(u_i)$ to $\Omega$ and eliminates $u_i$ and all its neighbors from $V_i$ as well as $V$. The algorithm iterates until there is no more vertex in $V$ indicating that each vertex is either chosen as a final vertex or is dominated by a final vertex.

The greedy algorithm has a time complexity of $O(m \log m)$ where $m = |V|$. In fact, the main loop in Algorithm 1 is the ‘while’ loop which iterates for $O(m)$ time. The main operation inside the loop is to the vertex with maximum profit (maximum value of cardinality of $V_i$ divided by vertex cost $w_i$). This can be done in $O(\log m)$ time using a priority heap. Therefore, the greedy algorithm achieves a time complexity of $O(m \log m)$.

**Theorem 2:** Algorithm 1 achieves an $\ln n$ approximation to the MCFS problem.

**Proof:** For every vertex $u_i$ selected as maximum profit vertex, define $\theta_i$ as $\frac{|V_i|}{w_i}$ at the time that $u_i$ was picked. Essentially, when $u_i$ is picked, it will dominate a number of adjacent vertices. For each vertex $u_j \in V$, let $u_i$ be the first picked vertex that is adjacent to $u_i$ and dominates it. Let define the cost associated with each dominated vertex $u_j$ be $\text{cost}(u_j) = \frac{1}{\theta_i}$. Notice that $\sum_{j=1}^{m} \text{cost}(u_j)$ represents the total cost obtained by the greedy algorithm. Let us order the vertices in the order that they were dominated. At the time that the $k$th vertex (call it $u_k$) was dominated, $V$ contained at least $m - k + 1$ vertices. For example, at the very beginning of the algorithm when the first vertex $u_1$ in $V_i$ is being dominated by the first picked vertex $u_1$, the total number of non-dominated vertices in $V$ is $m$. When the second vertex $u_2$ is about to be dominated by some neighboring vertex, the number of non-dominated vertices in $V$ is $m - 1$. At that point, the “per-vertex” cost of OPT is at most $\frac{OPT}{m-k+1}$. Thus, for at least one of the $u_i$ (call it $U$)
in OPT, we know that

\[ \frac{|U \cap V|}{w_u} \geq \frac{m - k + 1}{OPT} \]  

Thus, for the vertex \( u_i \) picked by the algorithm as the most profit profit vertex, we have \( \theta_i \geq \frac{m-k+1}{OPT} \). Therefore,

\[ \text{cost}(u_k) \leq \frac{OPT}{m-k+1} \]  

Over the execution of the greedy algorithm, the value of \( k \) changes from \( m \) to 1. Thus, the total cost of each vertex that the algorithm removes is at most

\[ \sum_{k=1}^{m} \frac{OPT}{m-k+1} \leq OPT \ln m. \]  

Thus, the greedy algorithm is a \( \ln m \) approximation to the MCFS where \( m \) denotes the number of vertices in the redundancy graph.

\[ \square \]

4.6 Real-Time Feature Selection

Similar to the weighted set cover problem, the MCFS problem belongs to a group of hard problems which are neither approximable in polynomial time nor fixed parameter tractable, under widely believed complexity assumptions [47]. While one can use the ILP approach to find the optimal solution offline (i.e. finding optimal feature set prior to deploying the wearable sensor node system), the ILP may not be feasible for real-time execution on the stringent constrained wearable sensor nodes. Thus, for real-time and dynamic feature selection an approximation algorithm is preferred as long as sufficient accuracy is obtained. The greedy algorithm presented in Algorithm 1 has a logarithmic approximation factor (\( \ln m \)) and yields a time complexity of \( O(m \log m) \). Ideally one would like to devise an algorithm which has an accuracy performance as close as possible to the ILP solution (i.e. exact algorithm), and a time complexity as close as to the greedy approach. The only way to provide the algorithm with a better performance is to sacrifice on the time complexity of the algorithm. The approach presented in [47] suggests that partitioning the universe set (e.g., vertex set in redundancy graph) into \( r \) subsets and running an exact algorithm on each subset will result in an approximation ratio of (\( \ln r \)) and a time complexity of \( O(e^{m/r}) \) where the exact algorithm can for the entire feature set can run in \( O(e^m) \).

An essential aspect of real-time feature selection is providing tradeoffs between the accuracy and time complexity of the feature selection algorithm. In this section, we attempt to present the general approach for formulation of the problem. Without loss of generality, let (\( \ln r \)) be the approximation ratio of a the feature selection algorithm that runs ILP on partitioned feature subsets. Also, assume that such an algorithm runs in \( t = e^{m/r} \) time units. Our objective is to find optimal number of partitions (\( \hat{r} \)) that minimizes the approximation ratio subject to a time budget (\( T \)) for running the feature selection algorithm.

\[ \text{Minimize} \quad \ln \hat{r} \]  

Subject to:

\[ t \leq Tt = e^{m/r} \]  

This optimization problem is equivalent to:

\[ \text{Minimize} \quad \hat{r} \]  

Subject to

\[ r \geq \frac{\ln t}{\ln T} \]  

Thus, the optimal number of partitions (\( \hat{r} \)) is

\[ \hat{r} = \frac{\ln t}{\ln T} = \ln(t - T) \]

Fig. 5 shows how the number of partitions grows as \( t - T \) increases as a result of reduction in the time budget (\( T \)). The figure illustrates \( r \) for the case when \( t - T \) ranges from \( 10^3 \) to \( 10^6 \).

5 Experimental Validation

In this section, we demonstrate the performance of the proposed feature analysis techniques utilizing real data collected from three human subjects using wearable motion sensors. Motion sensors were used to measure acceleration and angular velocity of six different body segments (including upper and lower body limbs) while the subjects were instructed to perform 14 transitional movements. The collected data were partitioned into two disjoint data sets, one for solving the proposed optimization problem and the other for measuring classification accuracy of the system on the selected features.
TABLE 1
Location of wearable sensor nodes

<table>
<thead>
<tr>
<th>ID</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Waist</td>
</tr>
<tr>
<td>2</td>
<td>Right Wrist</td>
</tr>
<tr>
<td>3</td>
<td>Left Wrist</td>
</tr>
<tr>
<td>4</td>
<td>Right Arm</td>
</tr>
<tr>
<td>5</td>
<td>Left Thigh</td>
</tr>
<tr>
<td>6</td>
<td>Right Ankle</td>
</tr>
</tbody>
</table>

TABLE 2
Experimental movements

<table>
<thead>
<tr>
<th>No.</th>
<th>Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stand to Sit</td>
</tr>
<tr>
<td>2</td>
<td>Sit to Stand</td>
</tr>
<tr>
<td>3</td>
<td>Sit to Lie</td>
</tr>
<tr>
<td>4</td>
<td>Lie to Sit</td>
</tr>
<tr>
<td>5</td>
<td>Bend to Grasp</td>
</tr>
<tr>
<td>6</td>
<td>Rising from Bending</td>
</tr>
<tr>
<td>7</td>
<td>Kneeling Right</td>
</tr>
<tr>
<td>8</td>
<td>Rising from Kneeling</td>
</tr>
<tr>
<td>9</td>
<td>Look Back</td>
</tr>
<tr>
<td>10</td>
<td>Return from look back</td>
</tr>
<tr>
<td>11</td>
<td>Turn Clockwise</td>
</tr>
<tr>
<td>12</td>
<td>Step Forward</td>
</tr>
<tr>
<td>13</td>
<td>Step Backward</td>
</tr>
<tr>
<td>14</td>
<td>Jumping</td>
</tr>
</tbody>
</table>

5.1 Experimental Setup

Three healthy subjects volunteered to participate in the data collection for this study. The participants were all graduate students, male, and aged between 25 and 35. They were asked to wear a wearable network by attaching six sensor nodes to the locations specified in Table 1. Each node has a 3-axis accelerometer and a 2-axis gyroscope for sensing human motions. Also, it has an embedded microcontroller (MSP430) for processing and a chipcon radio for data transmission. Our accelerometers are LIS3LV02DQ with 1024 Lsb/g sensitivity and are used in 2g mode for the experiments. We use IDG-300 gyroscopes with 2 mV/°/s sensitivity and 0.014 °/s/√Hz noise performance. Each subject was asked to perform each of the 14 transitional movements shown in Table 2, 10 times. The nodes were programmed to sample the sensors at 50 Hz and transmit the data wirelessly to a Laptop computer. The data acquired on the computer were used for offline signal processing and feature selection.

The data of each movement were segmented to find ‘start’ and ‘end’ points of the movement in the signal. The signal segments then undergo feature extraction. A set of nine statistical features, shown in Table 3, was extracted from each signal segment. Given that each node consists of five embedded sensors and feature vector of length 270 (i.e. $F = \{f_1, f_2, \ldots, f_{270}\}$) was obtained from the six sensor nodes. This feature vector was used for the feature analysis and power-aware selection approaches discussed in Section 4 and Section 4.2. The extracted features were partitioned into two sets, a training data set to find optimal feature set and a test data set to measure the classification accuracy using only optimal features.

5.2 Parameter Estimation

One of the parameters used to solve our optimization problem is the energy consumed for processing each feature ($w_i$). Energy consumption variation for different features stems from different instruction types, circuit states, and memory address modes, as well as the overall complexity of each feature. We consider TI MSP430 processor in order to find each feature’s processing energy. The MSP430 is widely used in battery-operated cyber-physical systems, especially those intended for augmenting human capabilities, and enhancing societal well-being. Battery-operated cyber-physical systems are obviously in need of low power consumption and MSP430 nicely meets such need (594 µW power consumption on average, which yields a performance of 37 µW/MIPS).

MSP430 is a 16-bit RISC CPU that uses the Von Neumann architecture. It has 48KB of Flash memory, 10KB of RAM, and uses an 8 MHz clock. The processor benefits from a 3-stage pipeline with 16 general purpose registers. Twenty-seven instructions comprise the instruction set with 7 memory addressing modes available. As a multiplier is a peripheral and is not implemented in every member of the MSP430 family, we utilized a method based on the Horner’s approach to implement multiplication only by means of shift and add instructions. It is worthy of attention that MSP430 can perform a register shift or add in one instruction cycle.

We calculated energy consumption of the feature extraction block in Fig. 2 for calculating each of the 9 features listed in Table 3 using the MSP430 microcontroller, which is available on the TelosB motes used in our experiments.

The energy cost of each feature has been quantified by summing up the energy consumption for each of its
The accuracy, however, starts decreasing at features being eliminated from the classification process.

5.3 Relevant Features

Fig. 6 shows the number of relevant feature and classification accuracy as a function of the relevance threshold ($\lambda_R$). Clearly, the number of relevant features decreases as the threshold in (3) increases. A very small value of $\lambda_R$ results in all the original 270 features being evaluated as relevant. With the complete feature set, the classification accuracy is 79.5% at the beginning. As the threshold increases, the accuracy improves as a result of irrelevant features being eliminated from the classification process. The accuracy, however, starts decreasing at $\lambda_R = 0.03$ indicating that the threshold is exceeding its optimal value and some relevant features start being eliminated from the list. Thus, we consider $\lambda_R = 0.03$ the optimal threshold value for our relevance analysis and perform the rest of our feature analysis with this value. Note that this threshold results in 51 relevant features.

An interesting observation is that irrelevant features can significantly reduce the performance of the action recognition system. We observe that by eliminating irrelevant features, the accuracy of the classification exhibits more than 21% improvement (i.e. from 79.5% with all 270 features to 96.7% with 51 features as obtained by $\lambda_R = 0.03$ in Fig. 6).

5.4 Optimal Feature Set

Fig. 7 shows the number of optimal features and classification accuracy versus the redundancy threshold ($\lambda_D$) using the ILP approach. It should be noted that according to the redundancy criterion in (4), a larger value of $\lambda_D$ results in less features being considered as strongly correlated and therefore, the redundancy graph will have smaller number of edges. With a small number of edges, more vertices need to be considered to cover all the vertices. This can result in a larger optimal set as Fig. 7 shows. As the threshold increases from 0.05 to 0.15, the
number of selected features grows from 4 to 44 with an average of 20.7 features. The classification accuracy ranges from 80% for $\lambda = 0.05$ to 96.7% for $\lambda = 0.15$. The classification accuracy is 90.1% on average.

The classification accuracy and number of final features reported by our greedy solution are illustrated in Fig. 8. For the greedy approach, the number of selected features ranges from 5 to 47 depending on the design parameter $\lambda_D$. The optimal feature set has a length of 24.2 features on average. The accuracy ranges from 79.5% for the lowest threshold to 96.7% for $\lambda = 0.15$, with an average accuracy of 90.0%. We note that, unlike the ILP solution, the greedy approach does not result in a monotonically increasing accuracy curve. The accuracy curve for the greedy approach has a local minimum at $\lambda_D = 0.13$. This is in fact due to the sub-optimality of the greedy approach, which does not necessarily select the optimal feature set at each step.

Table 5 lists energy savings obtained by applying our power-aware feature selection technique. For this specific table, only ILP results are reported. For greedy approach, however, similar results are achieved. The energy savings for the greedy solution range from 29.6% for the highest accuracy (96.7%) to 98.1% for the lowest performance (79.5%). This results in an average energy saving of 71.6% using the greedy solution.

5.5 Energy Analysis

Fig. 9 shows the total energy consumption of the selected features obtained by the ILP and greedy solutions. For ILP approach, the energy values range from 33$\mu$J for $\lambda_D = 0.05$ to 1184$\mu$J for $\lambda_D = 0.15$, resulting in an energy consumption of 467$\mu$J on average. The energy consumption results for the greedy approach range from 33$\mu$J to 1204$\mu$J with an average of 484$\mu$J.

Table 6 shows the optimal feature set and the corresponding sensor (i.e. x,y,z accelerometer and x,y gyroscope) and location of the node on the body. For visualization, the results are shown only for two cases, one with an accuracy of 80% and the other with 85.7% accuracy. Each entry in the second column
TABLE 6
Selected features and active nodes

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>80.0%</td>
<td>Amp_GyroX_Waist, Mean_GyroX_Waist S2E_AccZ_Waist, Max_GyroX_LeftThigh</td>
</tr>
<tr>
<td>85.7%</td>
<td>Mean_GyroY_Waist, S2E_GyroX_RightWrist S2E_AccY_LeftWrist, S2E_GyroX_RightArm Max_GyroX_LeftThigh, Amp_AccX_RightArm Mean_GyroY_RightAnkle</td>
</tr>
</tbody>
</table>

has three items representing the feature, the sensor, and the node location, respectively. For example, the entry ‘Amp_GyroX_Waist’ specifies ‘signal amplitude’ feature that is calculated from the ‘x’ axis of the gyroscope sensor attached to the ‘waist’. For the first case, only two nodes need to be active (‘waist’ and ‘left thigh’). In the second case, five nodes are required to complete the classification task to achieve a higher accuracy.

6 DISCUSSION AND FUTURE WORK

In addition to the primary advantage of power-aware feature selection, our algorithm provides an implicit method of sensor selection and node activation. Once prominent features are found, only their corresponding sensors need to be maintained in the system. Sensor nodes that do not contribute any features to the optimal feature set can be easily eliminated from the system. Thus, more energy savings can be achieved by including elimination of redundant sensor nodes in the energy saving calculations. For example, for detecting the 14 movements specified in Table 2, it is clear from Table 6 that only two nodes (‘waist’ and ‘thigh’) are sufficient. We anticipate that only one sensor node would be enough for detecting upper-body movements listed in Table 2 and only one sensor suffice for recognition of lower-body actions.

In our problem formulation in Section 4.3, we assumed that the redundancy graph is a connected graph. In reality, there might be cases where a relevant feature is not strongly correlated with any other features, which results in a disconnected graph. Yet, an optimal solution can be obtained in such cases, by running the optimization problem on individual connected sub-graphs and finding optimal feature set for each sub-graph.

Previous studies [51], [52] have demonstrated that continuous activity recognition (on-node processing of accelerometer and other sensor data streams) can rapidly deplete the battery energy in smartphones. Thereby, it is important to reduce the energy overheads of continuous mobile sensing [53], [54]. Even though the hardware in smartphones is not as customized and dedicated wearable sensor platforms, the ubiquitous nature and variety of radio communications and sensors enable them to be easily deployed in activity monitoring applications with a minimal learning curve and without imposing an extra cost on the users. We expect that at the end of a regular day of activity monitoring on a Samsung Galaxy S2, the power-aware feature selection scheme can save up to 50% of battery power as opposed to the power-unaware feature selection scheme. The selected features are computed on the appropriate frames of accelerometer streams. Such savings will decline when power hungry sensors such as gyro and GPS are in use or if we turn on the wireless network interfaces and the display. In general, energy savings for smartphones is lower than those for wearable sensor platforms as the communication subsystem and the display are the dominant power consumers.

In this study, we used an information theory measure (symmetric uncertainty) to quantify correlation between two features or between a feature and a class. Our model-based design and optimization approach, however, is independent of the choice of correlation measurement. One can replace the symmetric uncertainty with any other measure, build our graph model, and apply the proposed algorithms.

The development of low-complexity algorithms for the extraction, selection, and classification of the features of an observed signal at node level often leads to a significant reduction in the energy consumption of the node itself. Designing power-aware signal processing algorithms is challenging as special care needs to be taken to maintain acceptable classification accuracy while minimizing the energy consumption.

In this paper, our primary focus was to tackle the problem of feature selection in wearable sensor networks. For this, we focused on a static feature selection approach where the optimal feature set for a particular application setting is determined prior to execution of the signal processing algorithms in real-time. Although fully dynamic feature selection requires extensive research and is out of scope of this paper, the static feature selection approach presented in this paper can address a semi-dynamic feature selection as well. In fact, dynamic reconfiguration of the signal processing algorithms and modification of optimal feature set in real-time can be accomplished when the context of a user changes. Basically, as the requirements of the system change or new environments (e.g., geographical location) create the need for new feature set, the optimal feature set can be replaced with another predefined feature sets associated with the new context. For example, from monitoring upper extremity movements in an in-home rehabilitation setting, the system may switch to monitoring lower extremity movements. In this case, the optimal feature set may change from features extracted from a node placed on the wrist to a set of features from a node placed on the ankle. In any case, as long as the possible contexts are defined a priori, the system can redefine optimal feature set for each context and activate that feature set during real-time execution of the algorithms.

In the future, we will also investigate dynamic feature selection and node activation based on the contextual information about the subject in real-time. That is, information such as subject’s location and current activity can be used in real-time to further eliminate power-hungry
features and deactivate sensor nodes not contributing to the classification.

7 Conclusion
The accuracy and power trade-offs in wearable sensor networks have been investigated in order to guarantee classification accuracy, while minimizing the system's power consumption. We accounted for energy consumption in the process of feature selection; to achieve that, we first formulated the problem as a weighted minimum set cover approximation, which is one of the oldest and most studied NP-hard problems. We then devised a greedy approach to select the features needed for the identification of activities being performed in a power-efficient manner. Experimental results on inertial data collected from real subjects demonstrated significant power savings.

References


[46] N. D. Lane and T. Campbell, “The jigsaw continuous sensing engine for mobile computing with emphasis on biomedical signal processing and ophthalmic imaging. Navid is currently the recipient of the CTSI Core Voucher Award at UCLA and has previously received the Symantec Outstanding Graduate Student Research Award and Google Outstanding Graduate Student Research Award.


Hassan Ghasemzadeh (M’10) is an Assistant Professor of Computer Science in the School of Electrical Engineering and Computer Science at Washington State University. He received Ph.D. in Computer Engineering from the University of Texas at Dallas, Richardson, TX, in 2010. He was on the faculty of Azad University from 2003–2006 where he served as Chair of Computer Science and Engineering at Damavand branch. The focus of his research is on algorithm design and system level optimization of embedded and pervasive systems with applications in healthcare and wellness. He is a member of the IEEE.

Navid Amini (M’12) is a postdoctoral fellow at the UCLA Stein Eye Institute, where he investigates different sensing technologies and signal processing techniques to quantify the quality of life for people with low vision. Navid received his Ph.D. degree in computer science from UCLA in 2012. His research interests lie broadly in mobile computing with emphasis on biomedical signal processing and ophthalmic imaging. Navid is currently the recipient of the CTSI Core Voucher Award at UCLA and has previously received the Symantec Outstanding Graduate Student Research Award and Google Outstanding Graduate Student Research Award.

Ramyar Saeedi (S’14) received the B.S. degree in computer engineering from University of Isfahan, Isfahan, Iran with honors and the M.Sc. degree in computer architecture from University of Tehran, Tehran, Iran, in 2008 and 2011, respectively. Currently, he is a Ph.D. candidate in electrical and computer engineering at Washington State University. He was working at University of Kurdistan, and University of Tehran from 2011-2013 where he served as lecturer and laboratory instructor. His current research interests include embedded systems, signal processing, low power design, body area networks and wireless sensor networks. He is a student member of the IEEE.

Majid Sarrafzadeh (F’96) received his Ph.D. in 1987 from the University of Illinois at Urbana-Champaign in Electrical and Computer Engineering. He joined Northwestern University as an Assistant Professor in 1987. In 2000, he joined the Computer Science Department at University of California at Los Angeles (UCLA). He is a co-founder and co-director of the UCLA Wireless Health Institute (WHI). His recent research interests include embedded systems, signal processing, low power design, body area networks and wireless sensor networks.

Majid Sarrafzadeh is a Fellow of IEEE. He has published approximately 450 papers, co-authored 5 books, and is a named inventor on many US patents. Prof. Sarrafzadeh has collaborated with many industries in the past 25 years. He co-founded two companies around 2000 - they were both acquired around 2004. He has recently co-founded two companies both in the area of Technology in Healthcare.