Sorting the Wheat from the Chaff: Adaptive Sensor Selection for Context-aware Applications

Farid Zaid, Johannes Schmitt, Parag S. Mogre, Andreas Reinhardt, Matthias Kropff and Ralf Steinmetz

Multimedia Communications Lab (KOM)
Technische Universität Darmstadt
Darmstadt, Germany
Email: {fzaid, jschmitt, pmogre, areinhardt, mkropff, rst}@KOM.tu-darmstadt.de

***Abstract*—Context-aware applications base their services on contextual information that can be queried from the sensors embedded in the environment. However, when the number of sensors and the applications using them increases, sensor query becomes on one hand a resource hungry task, e.g. for network bandwidth and energy needed to power the sensors, and on the other hand may yield loads of unnecessary information that should be processed by the context-aware application. Therefore, an informed sensor selection in such environments becomes a necessity. This paper proposes an algorithm for a relevance-based sensor query, which adaptively spends the allotted query budget on querying sensors that are most relevant to the user's concept. This relevance is measured using an objective function which combines both expected query cost and expected sensor utility, as observed from the the sensor query history. The results of our evaluation show the potential of our approach to approximate the user's concept with best accuracy while preserving the query budget.**

***Keywords*-**

I. Introduction

Users of ubiquitous computing environments are envisioned to benefit from a multitude of applications that tailor their services to the user's context. For example, in a shopping center a public display can provide shoppers standing nearby with personalized advertisements about articles of interest to them. To achieve this behaviour, the display needs initially to detect if a shopper is in its proximity, e.g. by scanning its Bluetooth neighbourhood. Secondly, the display should infer the preferences of the shopper, e.g. by measuring the time she spends close to different articles, and then displaying the articles near which the user spent most of the time.

Obviously, context-aware applications need basically to collect contextual information by querying sensors embedded in the environment (e.g. Bluetooth proximity sensors in the aforementioned example), and then to process this information to infer meaningful contexts (e.g. 'standing nearby' in the aforementioned example) which can be used to make decisions according to the user concept, i.e. how the user expects the application to behave in different contexts, e.g. 'to display an ad for an interesting article' in the above example. However, given the amount of available sensors and the applications using them, sensor query may on one hand turn into a resource hungry task, e.g. for network bandwidth and energy needed to power the sensors, and on the other hand may yield loads of unnecessary information that should be processed and filtered by the context-aware application. This can be unacceptable if the application executes a crucial task that does not tolerate processing delays, or if the application is executed on a resource-constrained device like a mobile phone, which is very common in ubiquitous computing environments. Therefore, it is crucial to enable informed sensor selection in such environments.

This paper proposes the *Relevance-based Query* with following major contributions:

- the algorithm is adaptive as it allows the queries to be modified at runtime to accommodate changing user concept or changing sensor space, and to fulfil the cost constraints that may change at runtime,
- the algorithm is dynamic as it hides the query complexity from the user, who is only required to give feedback whether the application's behaviour aligns with her concept.

Results of the sensor queries and the corresponding user's feedback are maintained in a history which is used as a basis for computing the expected query cost and the expected sensor utility. Our algorithm is able to confine queries to sensors that promise to contribute most significantly to the context-aware task while maintaining the query cost within budget.

The rest of this paper is organised as follows: Section II gives a scenario to illustrate our approach. Section III highlights the major background for our work. Section IV provides a formulation for the problem. Section V presents our relevance-based query. Section VI presents our test environment and the evaluation results. Finally, Section VII concludes the paper and gives an outlook on future work.

II. Illustrative Scenario

We apply our approach to the Profile Manager application which enables context-aware management of the mobile phone profile (i.e. ringing and tone settings, etc.), therefore the application serves in reducing disruptions caused by improper phone alerts. The application works by querying a set of sensors, like Bluetooth to sense proximate users, chair sensor to sense if a user is occupying her seat, and a PC task sensor which tells the kind of task a user is doing on her PC (e.g. paper writing, skyping, etc.).

In this scenario, some sensors may provide similar information (e.g, both Bluetooth and chair sensors can be seen to provide location information). This on one hand helps the application to make decisions with finer granularity, and on the other hand increases the robustness of the decision against sensor failures. In contrast to this sensor redundancy, some sensors may not look relevant to deciding the proper profile (e.g. the PC task sensor of users working in other rooms). Querying these sensors can lead to unwise use of the available resources (e.g. processing power and battery on the mobile phone, sensor battery and network bandwidth). To handle these situations, the user has the opportunity to give feedback to adjust the automatically selected profile. The feedback forms in this case a means to identify patterns of sensor readings that contribute to approximating the user's concept.

III. Background

As the radio communication is the most costly task in terms of energy in a wireless sensor network, some approaches attempt to save the communication energy by reducing the amount of data sent by each node. For example, in the *directed diffusion* mechanism [1] nodes flood the network stating the data types they are interested in, and all other nodes in the network send their sensor readings matching the interest to the originating node. It becomes clear that when introducing *user-centricity*, sensors surrounding a user are generally better suited to provide information about the user than sensors in other places [2], thus providing means to narrow down searches to sensors of interest. To reduce the set of sensors that need to be queried even further, Chou et al. show in [3] that correlations between sensor readings can be exploited to decrease the amount of data that needs to be queried. To increase the information density within packets, means towards data compression have been adapted for sensor networks [4], [5], [6].

The aforementioned approaches attempt to prolong the network lifetime without considering the detection quality. In contrast, information driven sensor query (IDSQ) aims at balancing the information contribution of individual sensors with the cost for communicating with them. The main idea is to predict the information utility of a sensor reading before obtaining that reading, thereby avoiding unnecessary communication costs. For this purpose, a sensing task-specific utility measure is usually used. For example, in [7] a heuristic based on an *entropy* measure is used to select the sensor whose reading would yield largest reduction in the distribution entropy of the target location. Similarly, the sensing range of a sensor is used in [8] for prediction. Here, the *Mahalanobis distance*, which is a distance measure normalized by the uncertainty covariance, is used as an approximation for the sensor utility. However, this measure does not perform well if the sensor is not a range sensor.

In [9], the authors aim at achieving a desired *Quality of Inference* of a context variable from a set of sensors while keeping the communication cost with those sensors to a minimum. Besides the sensors readings, the tolerance ranges (uncertainties) of the individual sensors are used as input to the optimization algorithm. Sensors with larger tolerance ranges are considered to be more costly because they are to be sampled more often.

Our approach can be classified as an IDSQ, where we try to query most informative sensors while preserving the allocated query budget. However, in contrast to the other IDSQ approaches, we keep the prediction of the sensor utility independent from any specific sensor characteristic. Actually, prediction in our case is based solely on the current belief state, which we maintain in a form of a history. The history comprises the sensor readings (i.e. query results) and the user's feedback on the decision made using those readings. Together with the predicted cost for a sensor query, the predicted sensor utility is augmented in an objective function which ranks the relevance of the sensor. As a utility measure, we adopt the *information gain*, which is typically used as a metric for attribute selection in classification schemes [10].

IV. Problem Formulation

Before we formulate our problem, we explain the following terms in light of the scenario in Section II:

User Concept: This is the user's expectation with respect to the application behaviour, here the 'active profile of her mobile phone'. A concept $\mathcal{C}$ may have different classes (e.g. 'loud', 'vibrate' and 'silent') which specify different application behaviours. Typically, the user gives feedback to tell the application which concept class to adopt in a given context.

Concept Model: This is an approximation of the user concept that maps the sensor readings to one of the concept classes. A concept model $\mathcal{C}'$ can range from simple data processing (e.g. averaging of collected data) to complex data processing (e.g. building a classification tree). The concept model is adapted through the user's feedback. A good model is said to have a good *Classification Accuracy*.

Sensor Query Cost: Querying a sensor incurs some cost q. Cost can be for example the battery consumption needed

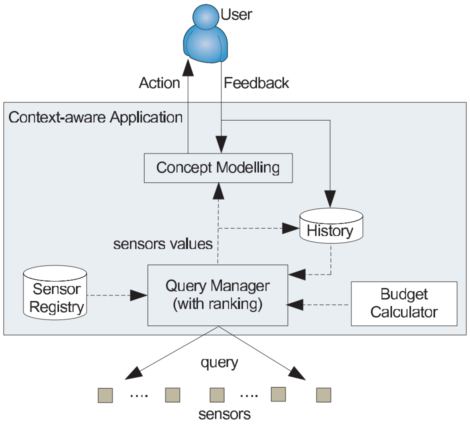


Figure 1. Main functions for the sensor selection.

to query a sensor and process the result, or it can be the time taken to query a sensor and process the result.

Query Budget: A query cost should not exceed an allowed budget. For example, a query is allocated a budget depending on the remaining phone battery (knowing that the mobile phone runs other applications which consume battery as well).

Assuming that the environment is embedded with sensors $\mathcal{S} = \{s_1, \ldots, s_N\}$, where querying sensor s_n is associated with an initial cost q_n^0, the problem we are tackling is to query a subset of sensors $\mathcal{S}_R \subseteq \mathcal{S}$ that are relevant to the user concept. Relevance implies two main requirements:

- querying these sensors lead to best classification accuracy, and
- the cost for querying these sensors does not exceed the allocated budget β

V. Approach

The main blocks of our approach are shown in Figure 1. The *Sensor Registry* element holds registrations for the sensors $\mathcal{S} = \{s_1, \ldots, s_N\}$. The *History* element maintains the sensor query-feedback history $\mathcal{H} = \{h_1, \ldots, h_M\}$. Each history sample h_m contains pairs of sensor value $v_n \in V_n$ and query cost q_n, besides the concept class $c_m \in \mathcal{C}$ as given by the user feedback, i.e. $h_m = (\{(v_{n,\,m}, q_{n,\,m}) : s_n \in \mathcal{S}_N\}\,,\, c_m)$. The *Query Manager* performs a *Relevance-based Query* of the available sensors. The *Budget Calculator* element estimates the available budget β for each query.

A. Relevance-based Sensor Query

The key idea of the relevance-based query is to predict the *expected utility* of a sensor from the query-feedback history before an actual query of that sensor takes place. We adopt the *information gain* as a measure for the sensor utility [8], [7]. In our case, the information gain measures the reduction in the entropy (or uncertainty) in the concept class by knowing the sensor reading. Given the query-feedback history $\mathcal{H}$, the information gain $\mathcal{I}$ of sensor s_n is computed as follows:

$$\mathcal{I}\,(s_n,\, \mathcal{H}) = \varepsilon(\mathcal{C}) - \varepsilon(\mathcal{C} \mid s_n) \tag{1}$$

The term $\varepsilon(\mathcal{C})$ represents the entropy of $\mathcal{C}$, computed as:

$$\varepsilon(\mathcal{C}) = \sum_{\forall c \in \mathcal{C}} \left(-\frac{\mid \mathcal{H}_c \mid}{\mid \mathcal{H} \mid} \cdot \log \frac{\mid \mathcal{H}_c \mid}{\mid \mathcal{H} \mid} \right) \tag{2}$$

where $\mathcal{H}_c = \{h_m : c_m = c\}$. The term $\varepsilon(\mathcal{C} \mid s_n)$ represents the conditional entropy of $\mathcal{C}$ given the values of sensor s_n, computed as:

$$\varepsilon(\mathcal{C} \mid s_n) = \sum_{\forall v \in V_n} \left(-\frac{\mid \mathcal{H}_v \mid}{\mid \mathcal{H} \mid} \cdot \varepsilon(\mathcal{H}_v) \right) \tag{3}$$

where $\mathcal{H}_v = \{h_m : v_{n,m} = v\}$. Similarly, we make an estimation of the sensor cost by taking the mean of the query costs $q_{n,\,m}$ as recorded in the history elements h_m. The estimated cost $\overline{q_n}$ is given as:

$$\overline{q_n} = \frac{\sum_{m=1}^{M} q_{n,\,m}}{M} \tag{4}$$

Now, having established measures for the expected utility and expected cost of the sensor, we express the relevance of sensor s_n to the user concept using the *Objective Function* $\mathcal{O}$ that mixes both utility and cost:

$$\mathcal{O}_{s_n} = \alpha \cdot \mathcal{I}\,(s_n,\, \mathcal{H}) - (1 - \alpha) \cdot \frac{\overline{q_n}}{\beta_r} \tag{5}$$

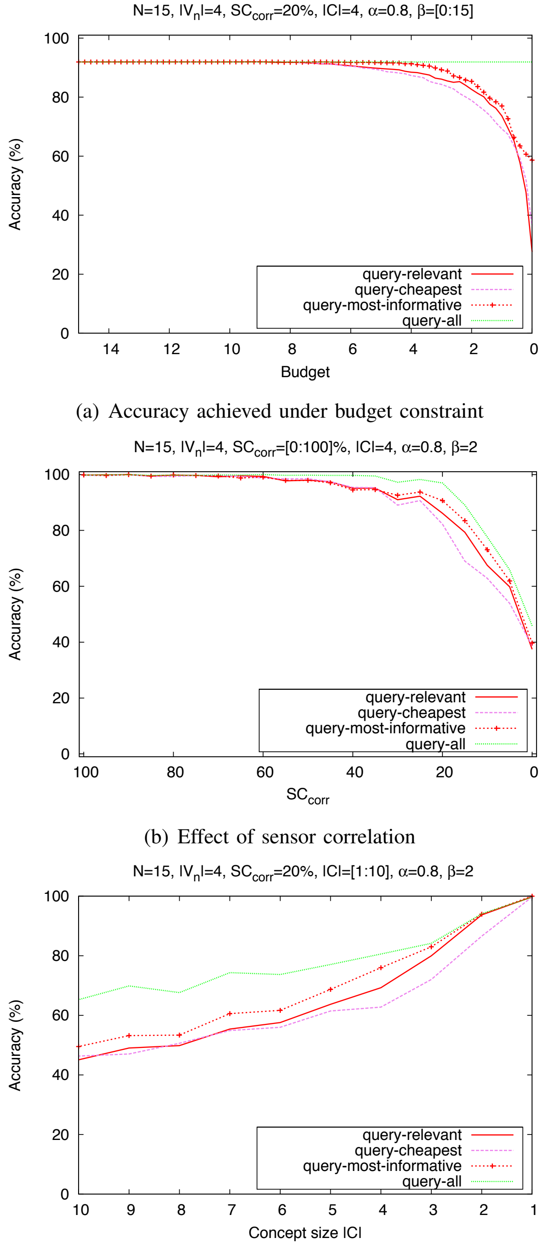
where β_r represents the remaining budget (i.e. budget left to query sensor s_n), and $\alpha \in [0,\, 1]$ is the relative weighting of utility and cost. We choose higher α for applications which value more the utility (i.e. recognition accuracy), while we choose smaller α for applications which are more conservative about the costs. Normalizing the query cost to β_r assures that we select sensors that best fit the currently available budget.

Algorithm 1 depicts the Relevance-based Query algorithm which uses the objective function above to query sensors in the order of their relevance to the user concept. The algorithm accepts a list of registered sensors $\mathcal{S}$ and the query budget β. In each iteration, the BESTSENSOR function selects from $\mathcal{S}$ the sensor s_0 with the highest objective function value. The information gain computation takes into account the values of the already queried sensors $\mathcal{S}'$ to deduce only history elements corresponding to these values. If the expected cost of the s_0 lies within the remaining budget β_r, then s_0 is queried and added to $\mathcal{S}'$, and β_r is decremented with the actual cost incurred by querying s_0. This way, in each iteration we perform updated estimation of the objective functions for the sensors that are still to be queried. At the end of the iteration, the sensor s_0 has

Algorithm 1 Relevance-based Query

```
Input: budget $\beta$, history $\mathcal{H}$, sensors $\mathcal{S}$
  candidate sensor $s_0 = null$
  queried sensors $\mathcal{S}' = \phi$
  remaining budget $\beta_r = \beta$
  while $\mathcal{S} \neq \phi$ do                         ▷ stop condition
      $s_0 = $ BESTSENSOR $(\mathcal{H}, \mathcal{S}', \beta_r)$
      if $q_0 \leq \beta_r$ then
          $query\,(s_0)$
          $\mathcal{S}' = \mathcal{S}' \cup \{s_0\}$
          $\beta_r = \beta_r - q'_0$                        ▷ $q'_0$ is actual cost
          if $\beta_r \leq 0$ then
              break                                        ▷ stop condition
          end if
      end if
      $\mathcal{S} = \mathcal{S} - \{s_0\}$
  end while

  function BESTSENSOR$(\mathcal{H}, \mathcal{S}', \beta_r)$
      $\mathcal{H}' = \{h_m : v_{n,m} = v_n \,\forall\, s_n \in \mathcal{S}'\}$
      $\hat{i} = \underset{\forall S_i \in \mathcal{S}'}{\operatorname{argmax}} \left[\alpha \cdot \mathcal{I}\,(s_i, \mathcal{H}') - (1-\alpha) \cdot \frac{\overline{q_i}}{\beta_r}\right]$
      return $s_{\hat{i}}$
  end function
```



(a) Accuracy achieved under budget constraint

(b) Effect of sensor correlation

(c) Effect of concept size $|\mathcal{C}|$

Figure 2. Simulation results

been checked for relevance and therefore is removed from $\mathcal{S}$ (whether it was queried or not). The algorithm stops when one of the following two conditions holds:

- All sensors have been checked for relevance, i.e. $\mathcal{S} = \phi$
- The remaining budget runs out, i.e. $\beta_r \leq 0$.

We use the $\leq$ in the second stop condition because it may happen that the actual query cost q'_0 exceeds β_r, because q'_0 is higher than the cost estimate $\overline{q_0}$. A possible consequence for this is that some sensors, whose actual cost may lie within the remaining budget, may not be checked for relevance. To mitigate the the effect of discrepancies in the cost estimation, we can consider building a probability distribution function of the query cost, therefore help make better predictions.

The latest issue shows as well how our approach differs from *Integer Linear Programming (ILP)* [11]. While the latest assumes that the constraints are fixed while trying to find a whole solution for the optimization problem, our algorithm iteratively updates the constraint (here, the remaining budget) after finding each partial solution.

VI. Evaluation

A. Simulative Evaluation

We evaluated our approach simulatively by applying it to the scenario described in Section II. The following parameters are used to configure the simulation:

- the sensor space with N sensors,
- each sensor with $|V_n|$ discrete values, and an initial query cost q_n^0 which is normalized between $[0, 1]$,
- the interdependencies among sensors defined by the sensor-to-sensor correlation factor SS_{corr}
- the user concept with $|\mathcal{C}|$ classes,
- the ratio of sensors that are correlated with the user concept defined by the sensor-to-concept correlation SC_{corr},
- the weighting factor α

Initially, we used the above parameters to artificially construct the query-feedback history. Here, for each history element, a concept class is randomly chosen and the sensors correlated with this class are then forced to use values within the range defined by the user concept. The concept-uncorrelated sensors are set to randomly chosen values within their parameter range, however by still conforming to the interdependencies between the sensors. For the moment, we assume that the query cost to be fixed at the initial value.

We used the generated history as a training dataset to build a classifier (a Naive Bayes in this case). We then ran our relevance-based algorithm (shortly query-relevant henceforth) and tested it against other query schemes including:

- *query-all* which simply polls all available sensors,
- *query-cheapest* which uses the available budget to query the least expensive sensors, and
- *query-most-informative* which uses the available budget to query the sensors with highest information gain

For each query scheme, we simulated a sensor query by selecting a number of instances from the training dataset, where in each instance the value of the queried sensor is kept in the instance while the values of the non-queried sensors are set as missing. The selected instances are then fed into the classifier as a testing dataset, and we recorded the achieved classification accuracy. As part of our test environment, we used the Weka tool [10] for classification and information gain computation.

Figure 2(a) compares the query-relevant scheme against the other three schemes for a scenario setup with $N = 15$, $|V_n| = 4$, $SC_{corr} = 20\%$, $|\mathcal{C}| = 4$, $\alpha = 0.8$. The query-all forms a baseline with best achievable accuracy. At low budgets, the accuracy of our query-relevant is higher than that of the query-cheapest, however less than that of the query-most-informative because the latest consumes the budget in querying the most concept-correlated sensors. However, given enough budgets, the different schemes approach the baseline. Figure 2(b) explains the effect of the sensors' utility, i.e. their correlation with the user concept. Clearly, as SC_{corr} increases, the accuracy will increase for the different query schemes. Figure 2(c) shows the effect of the complexity of the user concept, which depends mainly on the number of concept classes. An analogous case in our scenario happens when the user extends the set of feedback classes she gives to the application (e.g. by defining the new class 'Incrementally Increasing Ringing'). Assuming a fixed budget, the increasing concept size will cause the classification accuracy to degrade. From these results, it is evident that our query-relevant offers a very good accuracy-cost trade-off as compared to the extremes of the query-cheapest and the query-most-informative schemes.

B. Emulative Evaluation

To evaluate our approach in a real world setup, we implemented the profile manager application on top of the Nokia N95 8GB mobile phone and tested it with the four query scheme variants. As our focus is on testing the query algorithm itself and not on how fast a user concept can be learned, we again used our aforementioned simulative method to build the query history, including the sensor readings and the corresponding user concept. However, the sensor themselves were emulated on a standalone server, and they were queried by the profile manager using socket communication over WiFi. The sensors returned values based on the generated query history. To emulate the cost for a sensor query, we let each sensor append to its return value a dummy string whose length is proportional to the cost of the sensor. Figure 3 shows the power profiles when using the query-all and query-relevant schemes (for $N = 10$, $|V_n| = 4$, $SC_{corr} = 20\%$, $|\mathcal{C}| = 4$, $\alpha = 0.8$), where sensors are polled every 30 seconds. Basically, in each query the query-all scheme drains power for longer periods of time (for the radio transceiver and the CPU).

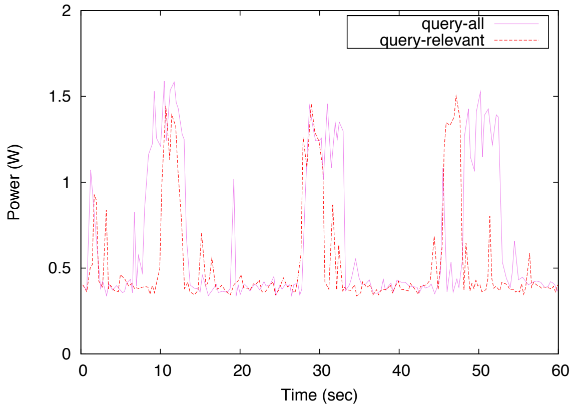


Figure 3. Power consumption of the query-all and query-relevant schemes

We ran our application in combination with the different query schemes, each for 10 minutes, and recorded the average power consumption, while shutting down all other applications on the mobile phone to avoid any influence on the measurements. Table I provides a summary of the test results which were collected using an energy profiling tool. The expected battery time is computed based on the average power consumption and the energy storage of a fully charged battery (equals to 1267 mAh in case of the Nokia N95 8GB). As expected, the query-all scheme leads to highest classification accuracy, however it leads as well to the shortest battery operational time. On the other extreme, the query-cheapest scheme guaranteed longest battery time, but led to a low classification accuracy[1]. The query-most-informative scheme led to a shorter battery time than the query-cheapest scheme, however with a not very significant improvement in the classification accuracy[2]. On the other

[1] By querying the cheapest five sensors

[2] By querying the most informative five sensors

Table I
COMPARISON OF DIFFERENT QUERY SCHEMES

	Query Scheme			
	All	Cheapest	Most-informative	Relevant
Classification Accuracy (%)	92.0	80.0	84.5	86.0
Expected Battery Time (hour:minute)	8:48	10:17	9:00	9:52

hand, our query-relevant scheme (with $alpha = 0.8$) gave an accuracy higher than query-cheapest and query-most-informative schemes, while providing a remarkably longer battery time.

VII. SUMMARY AND FUTURE WORK

This paper presents our relevance-based query algorithm to facilitate adaptive sensor selection for context-aware applications. The main goal is to confine queries to a subset of relevant sensors, i.e. informative sensors that contribute to the classification accuracy and preserve the query budget posed by the application. The complexity of the query is hidden from the user who is only required to give feedback on the application behaviour. Using simulative and emulative evaluation, we showed the advantage of our approach as applied to a specific context-aware computing scenario. The test results show that our algorithm provides a very good accuracy-cost trade-off in comparison with utility-unaware query schemes (like query of cheapest sensors) and cost-unaware query schemes (like query of most informative sensors).

As next step, we will be investigating several issues including the performance of our algorithm, the effect of varying query costs (e.g. when caching schemes are incorporated), different models for estimating the query cost, and the minimum history size that is required for a good enough estimation of the expected utility.

ACKNOWLEDGEMENT

Parts of this research have been supported by the German Research Foundation (DFG) within the Research Training Group 1362 "Cooperative, adaptive and responsive monitoring in mixed mode environments" and the German Federal Ministry of Education and Research (BMBF)

REFERENCES

[1] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 1999.

[2] G. Giorgetti, G. Manes, J. H. Lewis, S. T. Mastroianni, and S. K. S. Gupta, "The Personal Sensor Network: A User-Centric Monitoring Solution," in *Proceedings of the 2nd International Conference on Body Area Networks (BodyNets)*, 2007.

[3] J. Chou, D. Petrović, and K. Ramchandran, "A Distributed and Adaptive Signal Processing Approach to Reducing Energy Consumption in Sensor Networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[4] C. M. Sadler and M. Martonosi, "Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant Networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[5] N. Tsiftes, A. Dunkels, and T. Voigt, "Efficient Sensor Network Reprogramming through Compression of Executable Modules," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2008.

[6] A. Reinhardt, M. Hollick, and R. Steinmetz, "Stream-oriented Lossless Packet Compression in Wireless Sensor Networks," in *Proceedings of the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009)*, 2009.

[7] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based Sensor Selection Heuristic for Target Localization," in *IPSN '04: Proceedings of the 3rd international symposium on Information Processing in Sensor Networks*. New York, NY, USA: ACM, 2004.

[8] F. Zhao, J. Shin, and J. Reich, "Information-driven Dynamic Sensor Collaboration," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, 2002.

[9] N. Roy, A. Misra, S. K. Das, and C. Julien, "Quality-of-inference (QoINF)-aware Context Determination in Assisted Living Environments," in *WiMD '09: Proceedings of the 1st ACM International Workshop on Medical-grade Wireless Networks*. ACM, 2009.

[10] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, 2005. [Online]. Available: http://www.cs.waikato.ac.nz/~ml/weka/book.html

[11] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998.