Optimizing Quality-of-Information in Cost-sensitive Sensor Data Fusion

Dong Wang, Hossein Ahmadi, Tarek Abdelzaher
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Harsha Chenji, Radu Stoleru
Texas A&M University
College Station, TX 77843

Charu C. Aggarwal
IBM Research
Yorktown Heights, NY 10598

***Abstract*—This paper investigates maximizing *quality of information* subject to cost constraints in data fusion systems. We consider data fusion applications that try to estimate or predict some current or future state of a complex physical world. Examples include target tracking, path planning, and sensor node localization. Rather than optimizing generic network-level metrics such as latency or throughput, we achieve more resource-efficient sensor network operation by directly optimizing an *application-level* notion of quality, namely prediction error. This is done while accommodating cost constraints. Unlike prior cost-sensitive prediction/regression schemes, our solution considers more complex prediction problems that arise in sensor networks where phenomena behave differently under different conditions, and where both ordered and unordered prediction attributes are used. The scheme is evaluated through real sensor network applications in localization and path planning. Experimental results show that non-trivial cost savings can be achieved by our scheme compared to popular cost-insensitive schemes, and a significantly better prediction error can be achieved compared to the cost-sensitive linear regression schemes.**[1]

I. Introduction

This paper investigates the trade-off between data collection cost in sensor networks and application-level quality of information. We are interested in sensor networks, where the mission is to predict or estimate some current or future state of the physical world. Target tracking (estimation of target locations in the physical world), localization (estimation of node locations in the physical world), and minimum exposure routing (estimation of minimum-threat routes for mobile entities through the physical world) are examples of possible application goals.

Our work complements previous research in sensor networks, where base-stations estimate *sensor measurements*. Previous work developed mechanisms to reduce the number of measurements that need to be reported while still being able to accurately estimate missing measurements [29], [10], [5], [1], [2], [32], [39]. In contrast, this paper considers estimation of variables that are not directly measured. For example, the location of a target, computed from multi-lateration, may not be directly measured by any single sensor. Rather, it results from processing individual sensor measurements (such as estimated distance from the target). In such applications, a *model* exists to derive the variable of interest. We shall call it the *prediction model* (using this term loosely to cover estimation and regression models as well). Raw sensor measurements, or features derived from such measurements, are inputs to that model. Hence, the input parameters of the model determine the cost of data collection. Models that have more parameters (or parameters that are more expensive to measure or compute) increase data collection cost [40]. This observation motivates our work on models that minimize cost.

Our paper substantially improves the inherent trade-off between modeling accuracy and cost of data collection in sensor networks. Normally, more accurate models involve more input parameters, which makes them more expensive. By judiciously replacing a complex general model with collections of simpler specialized models for different sub-cases, our scheme can do better both in terms of accuracy and cost; specialization may increase accuracy, while at the same time reducing the number of model parameters needed in the special case at hand, hence reducing cost. The main challenge in achieving such an improved trade-off lies in appropriately defining the special cases and the simpler models that apply in each case, which is the contribution of this paper.

Our scheme builds a tree whose leaves are regression models, each applies to a subspace of the input data space (i.e., a special case corresponding to a tree branch). It explicitly looks for a break-down that results in accurate yet simple (i.e., low-cost) models at branches. This is in contrast, for example, to using a single complex one-size-fits-all model that takes all possible parameters into account at all times [15], hence requiring more expensive data collection. Our modeling scheme is *hybrid* in that it exploits both ordered prediction attributes (those that have ordered values, such as integers) and unordered prediction attributes (e.g., labels such as "Nissan" or "Toyota") in modeling. Our experimental results show that non-trivial cost savings are achieved by our scheme compared to cost-insensitive schemes, and significant improvements are achieved in prediction error compared to cost-sensitive schemes. Prediction costs always stay within budget.

We restrict our analysis to models that do not change quickly. For example, the average accuracy and energy cost of a particular localization algorithm in a given deployment

[1] This research was sponsored by NSF grant CNS 1040380 and the Army Research Laboratory, and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

environment is not likely to change much over time [20]. Similarly, the relation between estimated fuel consumption of a car and various road, trip, and vehicle parameters is likely to remain the same, governed by laws of physics [14]. Once these models are learned, they can be exploited for a long time before re-learning is necessary. Hence, we do not consider the cost of model learning itself (although this would be a straightforward extension). While the models we consider are static, the environment need not be. Since our models are defined at leaves of a tree, different branches may be used for prediction under different environmental conditions.

The rest of this paper is organized as follows: we present the related work in Section II. The proposed hybrid cost-sensitive prediction scheme is discussed in Section III. Implementation and evaluation results are presented in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

Our work is complementary to two important directions in sensor network literature. The first describes techniques for minimizing the sensor data sent to a base-station while estimating missing sensor measurements [10], [5], [28]. In contrast, we concern ourselves with estimation of parameters that are not directly sensed. Hence, ours is the more general problem of developing prediction models for data fusion outputs (that are both accurate and cheap), as opposed to prediction models for individual sensor values. The second relevant body of work in sensor networks develops in-network protocols that take advantage of data models in order to optimize network communication cost [29], [1], [2], [32], [39]. This includes reliability [1], congestion control [2], and data suppression [32], [39] protocols. The problem we address is orthogonal to the above. Rather than being concerned with how the sensory data are aggregated and processed in the network, we discuss which sensors or data attributes should be collected or computed in the first place. Eliminating some data attributes from the model may reduce communication requirements or reduce the amount of processing required to calculate model inputs.

An advantage of our technique is that it produces models that are cost-sensitive (i.e., cost does not exceed a predefined budget). Several previous efforts on cost-sensitive classification and active feature-value acquisition addressed some notion of cost [37], [40], [33], [19], [25], [26], [31]. They differ in the type of costs considered, the modeling approach applied, and the nature of the training set. For example, Tan [37] selects sensors on a robot to control grasping, taking into account execution costs. Turney accommodates both the cost of attributes and misclassification [40]. A partially observable Markov decision process is used in [19]. Melville and Saar-Tsechansky study the active feature-value acquisition problem in the context of incomplete training data [25], [26], [31]. A common property of the above cost-sensitive schemes is that they build classifiers that predict *discrete class labels*. In contrast, our work focuses on a cost-sensitive regression problem, where the leaves of the classification tree hold complete *regression models*, not class labels. In principle, attribute discretization can convert regression problems into classification problems. However, the accuracy and scalability of this approach may vary with scheme [13], [9]. Our scheme is the first to build classification trees with cost-sensitive regression models at the leaves.

Our work also bears resemblance to active learning; a useful technique from the machine learning community in which the learner has the freedom to choose the most informative training set when the resources to obtain data samples are limited [30], [33]. In contrast, our work focuses on the problem of cost-sensitive data attributes selection which is orthogonal to the sample selection problem mentioned above.

Non-cost-sensitive classification and regression techniques were used to extract models embedded in datasets and predict future data trends [18], [6], [38]. The literature on such algorithms is quite mature [22], [11], [4]. Attribute selection and tree pruning are two key techniques used to choose the right prediction attributes that best separate a given dataset into individual subspaces and build up a reliable regression model in each subspace [23], [34], [6], [7]. Our work is different in that it addresses cost concerns when data must be collected and processed over a resource-scarce environment such as a sensor network. Moreover, unlike decision trees that split the dataset in a way that maximizes an information gain metric [18], [7], [40], we use prediction accuracy of regression models as the measure to construct the tree.

An approach that comes close to ours is that of cost-sensitive regression by Geoetschalckx [15]. However, it assumed a *single* linear regression model that jointly minimizes a weighted function of prediction error and feature cost. As shown in the evaluation, by using a tree of regression models (automatically customized to different cases), we are able to achieve a better trade-off between modeling accuracy and data collection cost. Data cubes are another common techniques to handle large data sets with multiple dimensions efficiently for aggregation or prediction purposes [8], [12]. However, current cubes organize data by unordered (or categorical) attributes. We show in the evaluation that removing this restriction we offer a better trade-off between accuracy and cost.

III. Cost-sensitive Prediction

Consider a sensor network that measures or computes multiple data attributes $x_1, x_2, ...x_d$, from the physical world to estimate or predict an output, y, of interest (e.g., the location of a target). We call d the dimension of the dataset. The ith sample of attribute x_j is denoted by $x_{i,j}$, and the ith sample of output y is denoted by y_i. Further, let $\hat{y}$ denote the estimate of y. A cost c_j is associated with measuring or computing data attribute x_j per unit time. The goal is to build a model that minimizes prediction error of y, while keeping the cost of obtaining its input within a budget constraint, $\sum_{j \in used} c_j \leq C_B$. In general, there may be more than one resource to consider, in which case C_B is a vector. The prediction error is given by:

$$Err = f(y_i - \hat{y_i}) \qquad (1)$$

where f is monotonically increasing. The model is to be built given a set of N samples $\mathcal{S}_N = (Y_N, \mathcal{X}_N)$, where each data sample $i (1 \leq i \leq N)$ is a tuple $(y_i^{sense}, x_{i,1}, x_{i,2}, ...x_{i,d})$, and where y_i^{sense} is an actual measurement of y.

Below, we first introduce the multi-model linear regression methods for prediction in complex non-linear data spaces. We then propose a hybrid cost-sensitive prediction model to solve the problem discussed above.

A. Hybrid Model Tree

Consider building a tree where intermediate nodes represent decision attributes, and leaves correspond to appropriate regression models. More formally, let the whole data space consist of L data subspaces $(S_1, S_2, ...S_L)$. In each subspace, the output variable is related to the data attributes by a linear model given by:

$$Y_k = X_k \eta_k + \epsilon_k \tag{2}$$

where Y_k and X_k are the output variable and data attributes of data samples in the k^{th} subspace respectively, and η_k represents the linear model of the subspace, ϵ_k is a zero mean noise with variance σ^2 that is not correlated with X_k.

For each subspace, the linear model (2) can be estimated by applying a standard regression method on training data to predict future output given data attributes in the subspace [17]:

$$\begin{aligned} \hat{Y_k} &= X_k \hat{\eta_k} \\ \hat{\eta_k} &= (X_k^T X_k)^{-1} X_k^T Y_k \end{aligned} \tag{3}$$

where the $\hat{Y_k}$ is the predicted value of output variable Y_k and $\hat{\eta_k}$ is the estimated regression parameter vector for subspace S_k.

We use the mean square error of a regression model as the measure of accuracy of the model similar to the traditional regression methods. However, we provide a particular reliability measure for the calculated mean square error later in Section III-C. Formally, the residual sum of squared errors for a subspace k is defined as follows:

$$Err_k = \sum_{i \in k} (y_i - \hat{y_i})^2 = (Y_k - X_k \hat{\eta_k})^T (Y_k - X_k \hat{\eta_k}) \tag{4}$$

Note that, a data attribute x_j, used in some subspace S_k, can either belong to the attribute set used for regression modeling in this subspace, called the prediction set D_p^k, or used along the decision tree to decide that subspace S_k is the one to use for prediction in the first place. The set of decision attributes on nodes leading to the subspace leaf is called the splitting attribute set D_s^k. The cost of prediction at subspace S_k, called C_k, therefore satisfies:

$$C_k = \sum_{j \in D_p^k \cup D_s^k} c_j \leq C_B \quad k = 1, 2...L \tag{5}$$

In order to optimize the prediction accuracy of the output variable, two problems need to be solved: 1) How to divide the whole data space $\mathcal{S}$ into an appropriate set of subspaces using only information from data attributes? 2) In each subspace, which data attributes are the best (in terms of accuracy) to use to build up the linear regression model of Equation (3)?

A simplified hybrid model tree is shown in Figure 1. Note that, both ordered attributes (i.e., x_1, x_2, x_5) and unordered attributes(i.e., x_3, x_4) are used to split the data into corresponding terminal nodes. Each terminal node represents a subspace, where a linear regression model exists to predict the output variable. Hence, we need to identify both the splitting and prediction attribute sets D_s^k and D_p^k defined above to predict at the terminal node T_k. There are two key problems to be solved in order to build up such a hybrid model tree: 1) which attribute to use to split the data at each intermediate node (e.g., $A_1 \sim A_5$) of the tree, and what termination condition to use at branches of the tree (e.g., $T_1 \sim T_8$)?

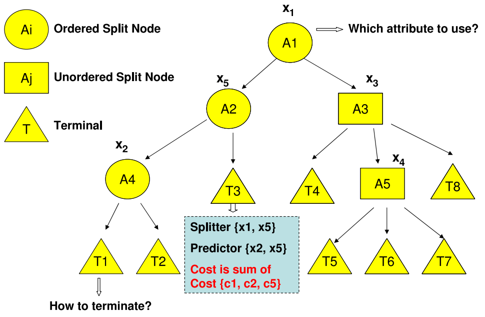

Fig. 1. A simple hybrid model tree

The search space of the optimal solution for the first problem is exponential [16]. Instead, we apply a greedy suboptimal solution. The main idea is to find the best splitting attribute in each step in the sense of maximizing the error reduction between a parent node and all its direct children. Therefore, we first search through all possible data attributes available at an intermediate node i, try to use each of them to split the dataset of node i, then calculate the error reduction $\Delta Err_i = Err_i - \sum_{j \in i's\, child} Err_j$ and select the one with the maximum ΔErr_i as the splitting attribute of node i.

Both ordered and unordered attributes are considered as the splitting attribute at an intermediate node. For unordered attributes, they are usually categorical, we split data according to their categories. For ordered attributes, we categorize their values into ν bins, the midpoint in each bin is considered as a possible split-point. Given ν categorized bins of x_s, ν possible splits are evaluated.

For the question of proper termination condition, we use a two-fold condition to solve the problem. First, a node is claimed as a terminal if no more error reduction occurs by splitting it. In other words, if $\Delta Err_i \leq 0$, we claim node i as a terminal node. Second, it is a terminal if too few samples remain in a subspace for reliable regression model construction to occur after splitting. This is due to the high dimensionality of the data space. A reliable model is the one that remains sufficiently accurate over the input range. We define a reliability condition given by (6) in order to have enough data for building reliable regression models at terminal nodes.

$$Pr[||\hat{\eta_i} - \eta_i|| > \delta] < 0.05 \tag{6}$$

where $\hat{\eta_i}$ and η_i are the estimated and actual parameters of the regression model at node i, δ is the confidence interval and $||x||$ denotes the l_2 norm of vector x. In our scheme, we use a probability threshold of 0.05 to identify whether a terminal node is reliable or not. To capture the effect of scaling in η_i, we set $\delta = \parallel \hat{\eta_i} \parallel$. An upper bound can be easily computed by using Markov inequality for the probability defined in the reliability condition above. The upper bound is given by:

$$Pr[||\hat{\eta_i} - \eta_i|| > \delta] \leq \frac{k\sigma^2}{\delta^2 \lambda_{min}(X_i^T X_i)} \tag{7}$$

where k is the number of data attributes, σ^2 is the estimation error variance that can be estimated from the mean square error of the regression [24], λ_{min} denotes the minimum eigenvalue of a matrix and the X_i is the matrix formed by data attributes of node i. Observe that using only data information contained in a node, the above upper bound can be easily computed efficiently. The details of derivation and computation of this upper bound are discussed in [17]. If we cannot ensure that all of node i's children are reliable, we stop at node i and claim it as a terminal node.

B. Hierarchical Cost Pruning

Given a cost budget for data attributes available for prediction, as discussed in Section III, it is important to ensure that the prediction cost at all terminal nodes of the tree always stays within budget. Briefly, first, we try to reduce the number of prediction attributes used at leaves. We call it *intra-node* cost pruning. It reduces the prediction cost by removing redundant and less important prediction attributes from terminal nodes of the tree. Classical algorithms are available for attribute reduction in a single linear regression model. Examples are forward/backward attribute selection [16] and decision tree methods [7]. The intra-node cost pruning at a terminal node stops when we find a valid reduced set of prediction attributes that moves the terminal into the cost budget while still keeping the regression model reliable. Alternatively, we stop when we are not able to find such a reduced prediction set after the prediction set becomes empty or the prediction model at the terminal node becomes unreliable. One constraint in choosing prediction attributes to remove is that it does not help to remove attributes that are also used for splitting since we would have to collect them anyway (for splitting purposes).

When a terminal node cannot meet the cost budget after the intra-node cost pruning stops, we consider reducing the number of splitting attributes. We call it *inter-node* cost pruning, and is achieved by simplifying the tree (by pruning leaves). A terminal node T_i will first try to prune itself in the sense that any prediction previously done at T_i will be done at T_i's parent after pruning. The parent node is one level up the tree and so has fewer splitting attributes and less cost. At the same time, a more general regression model exists at the parent node with less prediction accuracy which is the price of reduced cost.

It is possible that a parent itself cannot meet the cost budget. Hence, inter-node and intra-node cost pruning are carried out iteratively, first reducing prediction cost of leaves then cutting leaves out altogether until each leaf falls within budget. The pseudocode of the above process is shown in Algorithm 1.

Algorithm 1 Hierarchical Cost Pruning Algorithm

```
Notation
$Root$: The root of the hybrid model tree
$S_t$: The set of terminals that cannot meet the cost budget
$D_s$: The set of splitting attributes for tree construction
$A_{exp}$: The most expensive data attribute in $D_s$
$T_k$: Terminal $k$ in $S_t$
$T_{pk}$: Prediction node of $T_k$ after inter-node cost pruning
$C_k$: Cost to predict at $T_k$
$C_{pk}$: Cost to predict at $T_{pk}$
$C_B$: Cost budget

while $S_t$ is not empty && $D_s$ is not empty do
    pick a terminal $T_k$ from $S_t$
    $intra\_node\_cost\_prune(T_k)$;
    if $C_k > C_B$ then
        $T_{pk} = T_k$
        while $T_{pk} \neq Root$ && $C_{pk} > C_B$ do
            $T_{pk} = inter\_node\_cost\_prune(T_{pk})$;
            if $C_{pk} < C_B$ then
                break;
            end if
            $intra\_node\_cost\_prune(T_{pk})$;
            if $C_{pk} < C_B$ then
                break;
            end if
        end while
        if $C_{pk} > C_B$ then
            exclude $A_{exp}$ from $D_s$;
            rebuild the hybrid model tree;
        else
            exclude $T_k$ from $S_t$
        end if
    end if
end while
if $S_t$ is empty && $D_s$ is not empty then
    Hierarchical cost pruning succeed
else
    Cost budget $C_B$ is too small to predict reliably
end if
```

C. Reliable Bound on Prediction Error

The last step in the process is to estimate a bound on prediction error for nodes on the tree. For any prediction task given to a terminal node T of the hybrid model tree, the expected prediction error can be derived as follows:

$$\begin{aligned} Err_E &= E[(y - \mathbf{x}\hat{\eta_T})^2] = E[(\mathbf{x}\eta_T + \epsilon - \mathbf{x}\hat{\eta_T})^2] \\ &= E[(\mathbf{x}(\eta_T - \hat{\eta_T}) + \epsilon)^2] \end{aligned} \tag{8}$$

where $\mathbf{x}$ is the vector of predictors and y is the actual output value of the attribute we are predicting, $\hat{\eta_T}$ is the estimated linear model of the terminal node T. Considering ϵ and $\mathbf{x}$ are independent and ϵ has a zero mean, an upper bound of prediction error at terminal node T can be derived by using similar approaches in [3]:

$$Err_E \leq E[\parallel \mathbf{x} \parallel^2]\delta^2 + \sigma^2 \tag{9}$$

From the reliability condition defined in (6) to ensure reliable models at terminals, the regression coefficients are expected to be in the confidence interval of δ with at least 95% probability. Therefore, with 95% probability, the prediction error of a terminal node T is bounded by (9), and we call this bound 95% confidence bound on the prediction error.

D. Run-time Operation

For a real system deployment, the cost-sensitive hybrid model tree algorithm is implemented at the base station and it operates in two phases: an offline training phase that constructs a modeling tree and an online prediction phase that uses the model and the continuous stream of sensing and fusion data. For the offline phase, we assume that a sensor network is deployed with the capability to collect all data attributes for prediction. A sufficiently large training sample set is obtained for building the hybrid model tree at the base station. A cost profile configuration file keeps the costs of collecting different attributes. The application specifies the cost budget and the hierarchical pruning algorithm is applied on the hybrid model tree to prune all terminal nodes into the budget.

In the online prediction phase, the modeling backend interacts with the data collection and fusion component to estimate the output attribute. Each prediction task follows the cost-pruned model tree and selects the splitting attributes along the tree until it reaches its predicting terminal node, where all prediction attributes needed for the query are collected from the network. The prediction can be done periodically to build a time-series estimate of the output attribute or on-demand in response to user queries for up-to-date predictions. Remote procedure calls create an interface between the base station and the remote users who consume the predictions.

Upon changes in the cost model or the budget, the configuration file can be updated to initiate a tree re-pruning using the hierarchical algorithm in the online phase. For nodes deployed in the field, they just follow the instructions generated by the base station to collect relevant data attributes without specific hardware or software reconfigurations.

IV. Evaluation

A. Composable Localization Case Study

In this section, we apply the cost-sensitive prediction scheme to the design of a cost-sensitive composable localization protocol for sensor networks. Composable localization aims are localizing nodes in realistic, complex, outdoor environments of sensor networks [36] using the "best-of-breed" protocol for each environment, or the best mix of different protocols. Running multiple localization protocols, rather than one, on sensor nodes provides robustness of localization to protocol-specific inefficiencies. Observe that different localization protocols estimate node locations with different accuracies and they run at different costs. For example, a GPS localization scheme usually locates a node with a low localization error (e.g., 1~2 m) but at a high cost of power consumption (attributed to the GPS device), while DV-Hop is a distributed localization scheme that yields less accurate node location estimates (e.g., 4~7 m), but at a much lower cost [20], [27]. Hence, if we use outputs of different localization protocols and conditions under which they work as data attributes (features) to predict the actual location of a sensor node, we end up with the same cost-sensitive prediction problem presented in Section III. We compare the new scheme to several baseline schemes: i) a cost-sensitive single linear regression model that that uses all data attributes available within budget to build a single linear regression model for the whole data space [15]. ii) regression tree prediction scheme that uses heterogeneous attributes to build up a similar tree as the hybrid model tree, but does not perform regression at the terminal and only uses average value of a node as the model [6]. iii) data cube prediction scheme that only exploits the unordered attributes to split data space and performs regression at data cell [3]. Note that, both regression tree and data cube are cost-insensitive schemes.

Data attributes	Type	Cost(mW)	Localization error (m)
GPS Protocol Result	Ordered	36	1~2
Spotlight Protocol Result	Ordered	37.8	0.3~1
DV-Hop Protocol Result	Ordered	8.64	4~7
Centroid Protocol Result	Ordered	0.51	8~10
GPS Connectivity	Unordered	12.92	N/A
Line of Sight Availability	Unordered	0	N/A

TABLE I
Data Attributes Used in Composable Localization

The mapping from outputs of different localization protocols and conditions that affect their performance to data attributes defined in Section III is shown in Table I. We treat outputs of four localization protocols as the ordered data attributes since their orders indicate node's relative position to each other. GPS connectivity (whether a node has a GPS signal connection or not) and line of sight availability (whether a node has the line of sight to the Spotlight device or not) are taken as two binary unordered attributes (e.g., no explicit order between *on* and *off* state). The costs (i.e., average power consumption) of GPS and Spotlight localization protocols are computed from averaging the energy consumption of nodes equipped with corresponding devices over the localization period, we use numbers reported in [20], [35]. For DV-Hop and Centroid localization protocols, we implemented them in TOSSIM, and the power consumption is obtained from averaging both the energy of running localization algorithms on nodes and the overhead of communication (i.e., sending/receiving packets and idle listening) over the localization period [27], [21]. The cost for GPS connectivity is obtained by averaging the energy of the GPS device working in hot start mode to get coordinated with GPS signals over the localization period [20]. For line of sight availability, we assume such information can be obtained at the deployment time of Spotlight system thus has no further cost after deployment. More detailed parameters used in the above power consumption computation are listed in Table II. Meanwhile, estimations of accuracies of four localization protocols reported in [36] also listed.

We consider a distributed node localization scenario in which a set of anchor nodes, with accurate position knowledge, are distributed in the network. A set of four representative localization protocols, (namely, GPS [20], Spotlight [35], DV-Hop [27] and Centroid [21]) are available to run on the nodes

Name	Explanation	Value
P_{cpu}	CPU power only	11.04mW
P_{gps}	GPS in hot start mode	170.07mW
P_{tx}	Radio in transmit state	78.49mW
P_{rx}	Radio in receive state	74.85mW
P_{ls}	Radio in listen state	21.95mW
P_{laser}	Diode Laser of Spotlight	35mW
T_{hot}	Time for GPS in hot start mode	3.42s
T_{gps}	GPS localization period	45s
T_{sp}	Spotlight localization time	40s
R_{bw}	CC2420 transmit bandwidth	250kbps
B_{hop}	DV-Hop message size	15 Bytes
B_{hd}	Hop distance message size	11 Bytes
B_{hl}	Help message size	1 Byte

TABLE II
FACTORS AFFECTING POWER CONSUMPTION OF LOCALIZATION PROTOCOLS ON GPS-EQUIPPED MICAZ MOTES

in the network. Each anchor node is able to run a subset or full set of four localization protocols depending on its hardware configuration, power budget and location. We assume that nodes in the neighborhood of an anchor run the same set of protocols. A node in the vicinity of multiple anchors belongs to the neighborhood of the nearest one. The anchor node will collect results from different localization protocols and build up a cost-sensitive hybrid model tree for location estimation to share with all non-anchor nodes in its neighborhood, taking into account the latter's cost budget. The cost budget is taken to be a cap on energy consumption of the protocol, computed from the node's battery capacity and desired lifetime. Non-anchor nodes use the computed cost-sensitive model to predict their locations.

The cost-sensitive composable localization system was implemented in TOSSIM/TinyOS-2.1.1. We simulate a network topology with 100 nodes (20 are anchor nodes) deployed in a $125m \times 125m$ area. The nodes were randomly distributed in the topology. The radio range was set to be $30m$ with a sensitivity of $-75dBm$. The standard deviation of Additive White Gaussian noise for radio links was set to $4dB$. We assume 30% nodes are equiped with GPS devices with 40% probability to lose GPS signal during localization period and 50% nodes have a line of sight with the Spotlight device [35]. All nodes can run DV-Hop and Centroid. The system localization period is the same as the period of the GPS protocol.

The first experiment is to show the localization accuracy and cost trade-offs achieved by all composable localization schemes under comparison. We fixed the cost budget (desired average power consumption to run the set of localization protocols on a non-anchor node) at $70mW$. We run 100 experiments and record the average localization error of all non-anchor nodes and the average power consumption of localization process per non-anchor node over the network. As shown in Figure 2, the proposed cost-sensitive hybrid model tree achieves the least localization error compared to all other schemes. The reason is that the hybrid model tree exploits both ordered and unordered attributes to split data into more refined subspaces and build accurate regression models for prediction. The cost-sensitive single regression scheme has the highest localization error due to the fact that localization protocols perform differently under various conditions, a single linear model can not fit all cases. Also note that data cube achieves better localization accuracy than the regression tree scheme by doing regression at data cells. Figure 3 reports the average power consumption per non-anchor node of all schemes under the cost budget of $70mW$. Observe that two cost-sensitive schemes (i.e., single linear regression and hybrid model tree) have less power consumed than the budget while two cost-insensitive schemes (i.e., regression tree and data cube) consume more power than the budget. Therefore, the cost-sensitive hybrid model tree scheme is shown to achieve the least localization error while keeping the cost within budget.

The second experiment is to verify the capability of the cost-sensitive hybrid model tree to always keep the localization cost within budget. We vary the cost budget from $30mW$ to $100mW$, where $30mW$ is the minimum cost budget to have a single reliable prediction model on a node and $100mW$ is the budget to accommodate the total cost of all attributes. We run 100 experiments for each cost budget and report the CDF of localization cost for different localization schemes. As shown in Figure 4, the X-axis is the cost budget and every point on the curve shows the fraction of test cases that cost less than the cost budget. Observe that, the cost-sensitive schemes (i.e., single linear regression and hybrid model tree) always localize within the cost budget. For most cases, the cost-insensitive schemes (i.e., data cube and regression tree) however cost more than a certain budget threshold. Also note that regression tree scheme meets the budget at a lower cost than the data cube scheme due to the fact that regression tree takes average rather than builds regression models at terminal nodes. These results verify the effectiveness of hierarchical cost-pruning algorithm to keep the total cost of prediction always within budget for the hybrid model tree scheme.

The third experiment is to evaluate the prediction reliability characterized by the 95% confidence bound on prediction error discussed in Section III-C. The 95% confidence bound on prediction error is the error value that is guaranteed to be larger than the actual prediction error 95% of the time. To this end, we randomly choose a non-anchor node and predict its location using a model of its nearest anchor node. The cost budget is fixed at $70mW$. We change the size of training set and measure the mean square prediction error obseved and compare it against the analytically expected prediction error and the 95% confidence bound from Equation (9). The results are averaged over 100 experiments. Figure 5 shows that the expected prediction error stays closely around the actual prediction error observed and is well bounded by the confidence bound.

B. Green GPS Case Study

In this section, we apply our cost-sensitive hybrid model tree scheme to a real participatory sensing application, called GreenGPS [14]. GreenGPS gathers participatory sensing data from drivers to predict fuel consumption of different types of cars on different roads. For evaluation purposes, we assume that a future GreenGPS service is installed where users (or

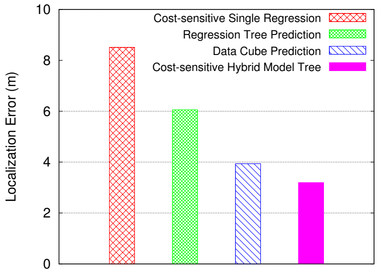

Fig. 2. Comparison of Localization Error

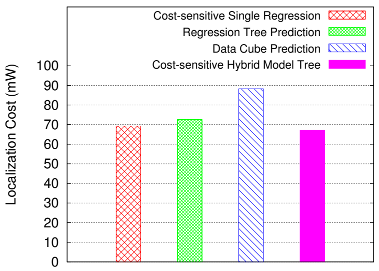

Fig. 3. Comparison of Localization Cost

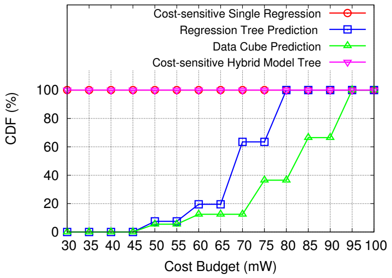

Fig. 4. Comparison of Cost CDF

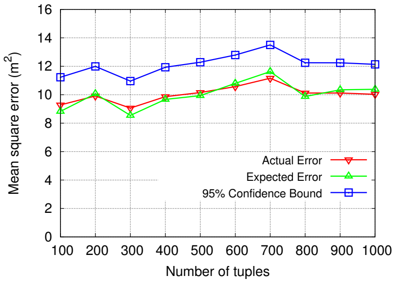

Fig. 5. Prediction Reliability

rather GreenGPS devices acting on their behalf) pay for access to road information that GreenGPS needs to predict fuel consumption. The mapping from different parameters to data attributes defined in Section III are listed in Table III.

Data attributes	Type	Cost (USD)
Number of Stop Signs	Ordered	5
Number of Traffic lights	Ordered	4
Average Traffic Speed	Ordered	10
Traffic Speed Variance	Ordered	8
Road Speed Limit	Ordered	6
Road Slope	Ordered	2
Car's Weight	Ordered	0
Car's Make	Unordered	0
Car's Model	Unordered	0
Car's Year	Unordered	0

TABLE III
DATA ATTRIBUTES USED IN GREENGPS APPLICATION

The first experiment is to show the prediction error and cost trade-offs of all prediction schemes under comparison, we fixed the cost budget to be 25 USD. Figure 6 and Figure 7 show that the cost-sensitive hybrid model tree achieves the best prediction accuracy while keeping its prediction cost within budget. Other baseline schemes either predict with a high error or fail to meet the given budget. For the second experiment to verify the cost-sensitive capability of the hybrid model tree scheme, we vary the cost budget from 16 to 35 USD, where the former is the minimum cost to have a single reliable model while the latter is the total cost of all data attributes. As shown in Figure 8, the cost-sensitive hybrid model tree scheme always predicts within the cost budget, while data cube and regression tree fail to meet the budget when budget is low. The third experiment is carried out to evaluate the prediction reliability, we randomly select samples of a road segment driven by a given car and predict its fuel consumption from data of other cars and segments. The budget is fixed at 25 USD. We change the size of training set and compare the estimated mean square prediction error against the actual prediction error observed and the 95% confidence bound. Figure 9 shows the estimated error is close to the actual error observed and is less than the confidence bound. To make error values meaningful, we have normalized fuel consumption values to be zero mean and between -1 and 1.

V. CONCLUSIONS

This paper described a new data modeling scheme that significantly improves the trade-off between modeling accuracy and cost of data collection. This improvement was achieved by replacing complex general models with groups of simpler cost-sensitive sub-models specialized for different sub-cases. Specialization allowed each sub-model to have a lower data collection cost (e.g., fewer parameters), but a comparable or higher prediction accuracy. A hybrid model tree was built with the cost-sensitive sub-models at leaves to directly optimize application-level prediction accuracy while respecting cost constraints. An accurate confidence bound was computed on prediction error. Experiments with real sensor network applications showed that significant cost savings and prediction error reduction could be achieved.

REFERENCES

[1] H. Ahmadi and T. Abdelzaher. An adaptive-reliability cyber-physical transport protocol for spatio-temporal data. In *Proc. 30th IEEE International Real-Time Systems Symposium (RTSS'09)*, pages 238–247, 2009.

[2] H. Ahmadi, T. Abdelzaher, and I. Gupta. Congestion control for spatio-temporal data in cyber-physical systems. In *Proc. 1st International Conference on Cyber-Physical Systems (ICCPS'10)*, pages 89–98, 2010.

[3] H. Ahmadi, T. Abdelzaher, J. Han, N. Pham, and R. Ganti. The sparse regression cube: A reliable modeling technique for open cyber-physical systems. In *Proc. 2nd International Conference on Cyber-Physical Systems (ICCPS'11)*, 2011.

[4] A.J.Dobson. *An Introduction to Generalized Linear Models (2nd ed.)*. Chapman and Hall, 2001.

[5] D. Blatt and A. Hero. Distributed maximum likelihood estimation for sensor networks. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[6] L. Breima, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

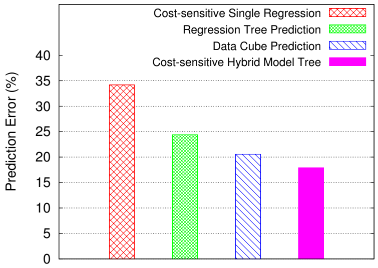

Fig. 6. Comparison of Prediction Error

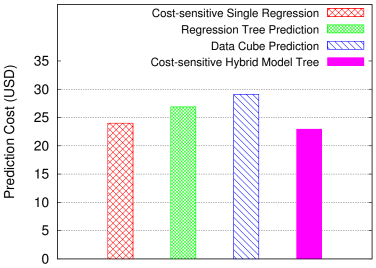

Fig. 7. Comparison of Prediction Cost

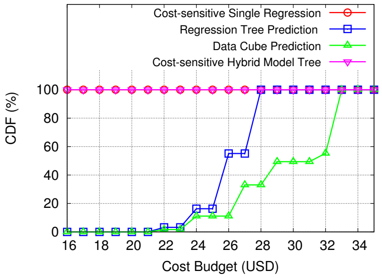

Fig. 8. Comparison of Cost CDF

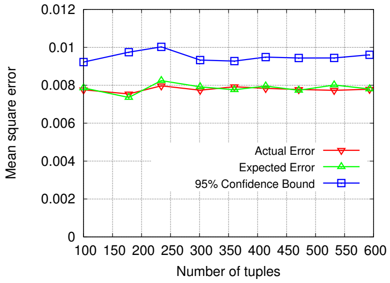

Fig. 9. Prediction Reliability

[7] L. A. Breslow and D. W. Aha. Simplifying decision trees: A survey. *Knowledge Engineering Review*, 12(1):1 – 40, 1997.

[8] B. Chen, L. Chen, Y. Lin, and R. Ramakrishnan. Prediction cubes. In *Proc. VLDB'05*, pages 982–993, 2005.

[9] M. R. Chmielewski and J. W. Grzymala-busse. Global discretization of continuous attributes as preprocessing for machine learning. In *International Journal of Approximate Reasoning*, pages 294–301, 1996.

[10] V. Delouille, R. Neelamani, and R. Baraniuk. Robust distributed estimation in sensor networks using the embedded polygons algorithm. In *Proc. IPSN'04*, pages 405 – 413, 2004.

[11] R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd. ed.)*. John Wiley and Sons, 2001.

[12] Y. C. et al. Regression cubes with lossless compression and aggregation. In *IEEE Trans. on Knowl. and Data Eng.18(12):1585-1599*, 2006.

[13] U. M. Fayyad and K. B. Irani. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proc. 13th International Joint Conference on Uncertainly in Artificial Intelligence(IJCAI'93)*, pages 1022–1029, 1993.

[14] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. GreenGPS: a participatory sensing fuel-efficient maps application. In *Proc. MobiSys'10*, pages 151–164, 2010.

[15] R. Goetschalckx, K. Driessens, and S. Sanner. Cost-sensitive parsimonious linear regression. In *Proc. 8th IEEE International Conference on Data Mining (ICDM'08)*, pages 809–814, 2008.

[16] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[17] C. Heij, A. Ran, and F. van Schagen. *Introduction to Mathematical Systems Theory: Linear Systems, Identification and Control*. Birkhuser Base, 2007.

[18] J.Han and M.Kamber. *Data Mining: Concepts and Techniques, Second Edition*. Morgan Kaufman, 2006.

[19] S. Ji and L. Carin. Cost-sensitive feature acquisition and classification. *Pattern Recogn.*, 40:1474–1485, May 2007.

[20] L. Jiang et al. SenSearch: GPS and witness assisted tracking for delay tolerant sensor networks. In *Proc. 8th International Conference Ad-Hoc, Mobile and Wireless Networks*, pages 255 – 269, 2009.

[21] N. B. John, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7:28–34, 2000.

[22] J.R.Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[23] I. Kononenko and S. J. Hong. Attribute selection for modelling. *Future Generation Computer Systems*, 13(2-3):181 – 195, 1997.

[24] M. Kunter, C. Nachtsheim, J.Neter, and W.Li. *Applied Linear Statistical Models*. McGraw-Hill, 2005.

[25] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. Active feature-value acquisition for classifier induction. In *Proc. 4th IEEE International Conference on Data Mining (ICDM'04)*, pages 483 – 486, 2004.

[26] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An expected utility approach to active feature-value acquisition. In *Proc. 5th IEEE International Conference on Data Mining (ICDM'05)*, page 4 pp., Nov. 2005.

[27] D. Niculescu and B. Nath. Dv based positioning in ad hoc networks. *Telecommunication Systems - Modeling, Analysis, Design and Management*, 22(1-4):267 – 280, 2003.

[28] J. B. Predd, S. R. Kulkarni, and H. V. Poor. Regression in sensor networks: Training distributively with alternating projections. In *CoRR, vol. abs/cs/0507039*, 2005.

[29] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *Proc. IPSN'04*, pages 20–27, 2004.

[30] N. Roy and A. Mccallum. Toward optimal active learning through sampling estimation of error reduction. In *In Proc. 18th International Conf. on Machine Learning*, pages 441–448, 2001.

[31] M. Saar-Tsechansky, P. Melville, and F. Provost. Active feature-value acquisition. *Manage. Sci.*, 55:664–684, April 2009.

[32] S. Santini and K. Roemer. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proc. 3rd International Conference on Networked Sensing Systems (INSS'06)*, 2006.

[33] V. S. Sheng and C. X. Ling. Partial example acquisition in cost-sensitive learning. In *Proc. 13th international conference on Knowledge Discovery and Data Mining (KDD'07)*, pages 638–646, 2007.

[34] Y.-S. Shih. Families of splitting criteria for classification trees. *Statistics and Computing*, 9(4):309 – 315, 1999.

[35] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In *Proc. SenSys'05*, pages 13–26, 2005.

[36] R. Stoleru, J. Stankovic, and S. Son. On composability of localization protocols for wireless sensor networks. *IEEE Network*, 22(4):21 – 25, 2008.

[37] M. Tan. Csl: a cost-sensitive learning system for sensing and grasping objects. In *Proc. 1990 IEEE International Conference on Robotics and Automation*, pages 858 –863 vol.2, 1990.

[38] P. Tan, M. Steinbach, and V.Kumar, editors. *Introduction to Data Mining*. Addison-Wesley, 2005.

[39] D. Tulone and S. Madden. PAQ: time series forecasting for approximate query answering in sensor networks. In *Proc. 3rd European Workshop on Wireless Sensor Networks (EWSN'06)*, 2006.

[40] P. D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, pages 369–409, 1995.