The 5th International Workshop on Information Quality and Quality of Service for Pervasive Computing 2013, San Diego (22 March 2013)

SPOTLESS: Similarity Patterns Of Trajectories in Label-lEss Sensor Streams

Vasanth Iyer, S. Sitharama Iyengar, Niki Pissinou, Shaolei Ren
School of Computing and Information Sciences
Florida International University, Miami, FL 33199

viyer002@fiu.edu, iyengar@cis.fiu.edu, pissinou@fiu.edu, sren@fiu.edu

Abstract—**The process of inversion, estimation and reconstruction of the sensor quality matrix, allows modeling the precision and accuracy, and in general the reliability of the model. When the sensor data ranges are not known *a priori*, current systems do not train on new data samples, rather they approximate based on the parameter's global average value, losing most of the spatial and temporal features. The proposed model, which we call SPOTLESS, checks the spatial integrity and temporal plausibility of streams generated by mobility patterns due to varying channel conditions. We define a minimum quality of the measured sensor data as local stream (QoD) requirements to give high precision by using distributed labeled training. In our SPOTLESS data-cleaning steps, to account for packet errors due to varying channel conditions, a soft-phy based decoding is selected for various Bit Error Rates (BER), minimizing packet loss at the mobile receiver. Numerical experiments for Rayleigh fading channels and mobile BER model examples are compared with large deployment of ground sensor collecting static data streams and Data MULE collecting multi-hop temporal data from the sensor to provide hypothetical parameter accuracy. Our results were obtained in the context of provisioning a minimum precision and accuracy stream (QoD) required for 802.15.4 mobile services. SPOTLESS data-cleaning algorithm coding provides 90% precision for static streams, and increases the plausible relevance of multi-hop mobile streams by 85% for task-based learning.**

Index Terms—**Sampling sensors, Data mining, Event Modeling, Temporal Patterns, Stream Learning, QoD, QoI and QoS.**

I. INTRODUCTION

Unsupervised data collected from sensors applications are transmitted in short streams, and with the advent of Data-mining and Machine Learning algorithms, Stream Learning has justified its need in improved preprocess and provides measurements with a context-based stream QoD metric. Stream Learning is a fast growing concept primarily due to increasing availability of location-aware functionality in mobile applications, which generate large scale machine generated data in everyday from wireless, mobile GPS and sensors. Unlike databases, these categories of information in streams are real-time, constantly changing and probabilistic in nature. In a survey in 2009, Gartner published that the data will grow over 650%, which may lead to an information overload for existing computation and bandwidth standards. When estimating and predicting labels, current methods need to bridge the gap between the contrasting small amounts of static training samples, compared with vast amounts of label-less mobile sensor data generated.

Due to inherently high redundancy in sensor data streams and the cost of transmission, many of the data stream values are averaged to a single estimate of the measure parameter. In a traditional deployment, for example, to increase the sensitivity of data-streams, typically the sampling period is increased or the density of the number of sensors deployed in a cluster is increased in the region of query interest. The important aspect of all these techniques is to increase the data accuracy and dissemination of the sensor streams to provide enough samples of the evolving patterns of interest. The current gap is to help reliably deliver spatial and temporal variations of the time series data without data packets being corrupted due to multi-hop routing in wireless outages or hight measurements seen due to a few bad sensors.

With the availability of smartphones with inexpensive GPS in them, samples can be accurately tagged with location information using Data MULES when collecting data from static ground sensors. The trajectory sampling used by Data MULES has more variance in the parameter's temporal range compared to when using WSN data aggregation (e.g. very few training samples to estimate). The learning is reduced by repeatedly sampling the same GPS tagged trajectories over time.

The streams' data can be organized into two categories (1) is obtained by spatial sampling periodically from static sensors. The other which is obtained by sampling temporal variations by collecting data from ground sensors using a mobile Data MULES, which have applications in the area of road traffic-related probes and Military Unmanned Autonomous Ground Vehicles (UAGV). The first category uses Identically Independently Distributed (i.i.d.) local sampling and the second category uses a time-stamp with global GPS coordinates to tag the location and position for the complete end-to-end path. In our simulation, we assume the Random Way-point [9] mobility model to collect data. Here, we are not interested in the instantaneous sampling, as in the case of other mobile applications, but in studying temporal overlaps in data streams, which can be further classified by using stream QoD-based labels. When training samples are few during the learning of the

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

overlapping ensemble of trajectory streams, we can re-sample those trajectory segments which have sparse intervals to get a finer bound of the temporal sensitivity of the sensor network deployed.

II. Motivation

Datacleaning algorithms consist of pre-processing raw data and filling in missing values, smooth noisy data, identifying removing outliers, and resolving inconsistencies. We can broadly classify the data transformation issues of Datacleaning algorithms into sensor data normalization and network packet aggregations.

A. Prior and Related Work

As a technical contribution to this paper, we are interested in generating stream QoD based labels of unsupervised data, SENSORML [10] streams and event data [1] present in logs of remote sensing and mobile applications. Most of the current techniques rely on local sampling at the sensor nodes to provide the precision and accuracy for the sensor network application. With the advent of mobile sensors and inexpensive GPS receivers, innovative sampling methods and mobile applications are available to provide locations-specific and temporal granularity in stream learning. Our normalization method uses recent advances in matrix completion [2] to improve the precision of the spatially sampled data streams with mixed real values, and uses a classification [1,2] approach to learn application-level features. In addition to static datacleaning mobile wireless [3] trajectory-based streams are studied with respect to fading and shadowing effects at the receiver. Our method uses data sampled from mobile trajectories which have spatial and temporal values to learn the hidden latent features. These featureset are used in context with stream QoD labels to further classify real-time streams.

B. Outline

The paper is outlined as follows: in Section III, stream QoD is discussed in terms of spatial and temporal stream learning. Section IV describes the matrix completion methods, which deal with normalization. Section V describes the complete datacleaning steps for stream learning applications with emphasis on physical layer network aggregation. Section VI outlines result of the data stream gathered from 10 to 20 Data MULES, measuring samples simultaneously using Random Waypoint trajectories, over an area consisting of 100 randomly placed sensors with a limited communication range. The wireless channel simulation uses Rayleigh fading and varying inter-symbol noise statistics seen at the mobile node's decoder causing dropped packets (observed variable), to estimate its optimal rate. The raw data measurements were compared using two different channel noise conditions, one with static data streams with known labels and the other with trajectory-based multi-hop label-less data streams. Various effects of error correction techniques are studied by estimating the channel states at the receiver provided by soft-phy [3] observed measurements, such as fading, shadowing, and Packet Error (PER). Finally, Section VII summaries the benefits of stream learning (SL) and datacleaning techniques in mobile sensing application.

III. Stream QoD Evaluation

The underlying SPOTLESS processing assumes a full-rank [2] matrix removing redundant columns present in square or skinny matrices, where k is the maximum number of observed features (columns) and we assume (k=1,2,3) to separate the observed features from the missing ones. Our cluster classification method uses a rule based on combining fully observed local attribute k weights, by which it learns the missing features by using gradient-decent methods.

A. Precision of Data Stream Using Spatial Features Learning

The SPOTLESS framework uses some of the data cleaning methods, as described below:

Definition 1 *Selecting k sensors from m possible measurements reduces the error of estimation. The brute force method uses $\binom{m}{k}$ permutations, which is computationally exhaustive, when k and m are large. We select a minimum set k, which reduces the training error. The computation can be efficiently accomplished by matrix factorization, which takes $O(m^3)$ operations and is invariant to the number of attributes in the dataset.*

Measurements taken from distributed sensor networks may contain faulty measurements. The data-cleaning algorithm needs to compute the accuracy of the estimated measurement from all the observed sensor patterns. This correlated pattern matching the training set makes it possible to predict the missing sample's estimate even when some sensor samples are erroneous or faulty.

1) Problem Definition: We define the problem of distributed sensor network sensor selection in the following way. The sensor measurement for a set of I placements with J attributes is needed. A non-faulty sensor can be represented as (i,j,x_{ij}) denoting the i.i.d. sample, where $i \in 1,...,I, j \in 1,...,J, x_{ij} \in \chi \in O$. We assume that the subscript ij belongs to the same set of the i.i.d. sample dataset. In a typical sample dataset, there will be a finite set of observed i.i.d. samples from non-faulty sensors, T, which are used for training. All the non-faulty observed samples are in (i,j) as O. In a typical WSN cluster $O \ll |I|.|J|$, we are interested in the lower-bound of the measured range. In this data-cleaning framework, we use a Machine Learning based algorithm on sensor streams data to minimize the root mean square (RMS)

error, which is defined as:

$$RMSE(T) = \sqrt{\frac{1}{|T|}\sum_{(i,j)\in T}(\hat{x}_{ij} - x_{ij})^2}$$

The sensor data values are organized as (i,j) in matrix X, as shown in Figure 1(a), where the missing or above range values are in $(i,j) \notin O$, as shown in Figure 1(b). In distributed sensor networks, the training samples (non-faulty) and the faulty samples are part of the same dataset collected in time.

B. Classifying of Data Stream Using Temporal Relevance

Definition 2 *Select t trajectories from d possible datasets to reduce the error of label estimation. The brute force method uses* $\binom{d}{t}$ *permutations, which is computationally exhaustive, when* t,d *are large. We select a minimum set t probes heuristically, which reduces the transductive training error. The computation can be efficiently accomplished by a random tree classifier, which takes* $O(t^2 \times d)$ *operations.*

IV. Numerical Modeling of Sensor Data Matrix

The observed values which are i.i.d. samples, can be represented as a system of equations, as described in Figure 1 (a). Its corresponding factorization matrix is derived in Figure 3, describing the training and the faulty sets (random mask). The missing value estimation is accomplished by using the least mean square error calculation of the observed non-faulty sensor stream. The approximating matrix steps are as follows:

The goal of matrix factorization is to represent a larger matrix, such as in Figure 1(a), with two approximately smaller once, which have standard forms, such as Cholesky [2], QR [2] and SVD [2] (Singular Value Decomposition has higher precission). The matrix coefficients, which are real valued constants, pose serious numerical stability issues, such as bias, while estimating the coefficients. Several techniques like regularization, use a damping factor to represent larger coefficients. Here, we show a simple technique to factorize a sensor stream representation [1,2,10]. Figure 1(a) shows a general matrix consisting of all feature values. Figure 1(b) has some missing spatial features denoted by a mask of the same matrix in Figure 1(a).

A. Matrix Completion

Definition 3 *Let* $X_1....X_n \in R^d$ *be feature vectors associated with n measurements. Let* $X = [x_1...x_n]$ *be a* $d \times n$ *feature matrix whose columns are the sensor measurements. Let there be t binary classification range query tasks,* $y_1...y_n \in -1,1^t$ *be the label vector, and* $Y = [y_1...y_n]$ *be the* $t \times n$ *label matrix. Due to the wrong calibration of sensors, entries in X or Y can be missing at random. Let* Ω_x *be the index set of observed labels in* Ω_y*. Our main goal is to predict the missing labels* $y_{i,j}$ *for* $(i,j) \notin \Omega_y$*. This is unsupervised, and without labels we can incorporate transductive learning to find the labels.*

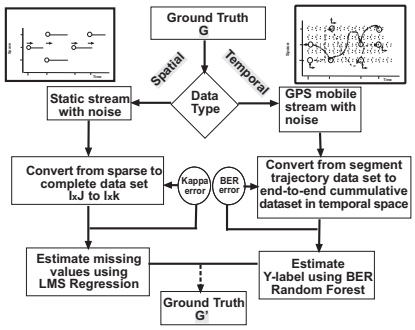


Fig. 2. The prototype of mobile probe sensor dataset being validated with SensorML's static ground truth estimates.

We like to decrease the training error (RMS)[1] of each label to the observed labels and de-noise the observed features in X*.*

To optimally represent the matrix X, which has many missing spatial features, we use $M \in \Re^{I\times K}$ and $S \in \Re^{K\times J}$, as shown in Figure 3, which are denoted as M- *Measurement* and S- *Spatial* of the k- *known features* of the original data matrix in Figure 1(a). Let m_{ik} denote the elements of the approximate matrix observed feature measurements of M, and s_{kj} denote the elements of matrix S, representing the i.i.d. samples of the spatial deployed neighboring sensors.

$$\hat{x}_{ij} = \Sigma_{k=1}^{K} m_{ik} mkj = m_i^T m_j$$
$$e_{ij} = x_{ij} - \hat{x}_{ij} \quad for(i,j) \in \Re$$
$$e_{ij}^{'} = \frac{1}{2} e_{ij}^2$$
$$\Sigma Squared \quad Error = \Sigma_{i,j} \in \Re e_{ij}^2$$
$$RMSE = \frac{SSE}{|\Re|}$$
$$(M^*, S^*) = \arg min \quad RMSE$$

$\hat{x}_{ij}$ denotes how close the i^{th} feature of the j^{th} sensor stream measurement is as shown in general matrix equation, with random mask, as shown in Figure (1). e_{ij} denotes the training error on the (i,j)-th observed measurement, and SSE denotes the sum of the squared training errors.

V. Regression and Data Cleaning

In mobile wireless sensor networks (mWSN in matrix in Figure 3), the data samples are gathered using spatial clustering and mobile trajectories to sample temporal values. We will define the reconstruction of sensor streams for both the cases, as described by the dataflow chart in Figure 2.

A. Error 1: Missing Data

STEP 1: Use model assumptions of sensor selection from definition 1, and the matrix factorization technique to increase precision, as described in definition 3, when

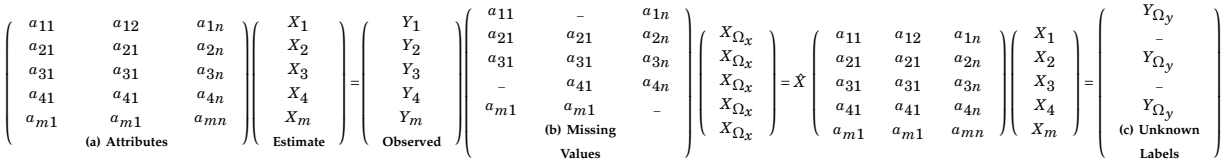


Fig. 1. Empirical evaluation of the accuracy in the proposed hypothesis.

Best Fit	Temp.	Humidity	Light	Kappa
Spatial Sampling	0%	0%	0%	17.78462
Error after cleaning (ω)	(1-0.95)=5%	(1-0.95)=5%	0=100%	13.0278
Real error $\frac{Kappa_{exact}}{Kappa}$	-	-	-	$(1-0.73) = 27.0\%$

TABLE I
MISSING FEATURES (KAPPA)

$$x_{lbound} = \begin{pmatrix} 19.9 & 38.8 & 45.08 \\ \_ & 29.25 & \_ \\ 19.4 & 38.6 & 48.76 \\ 30.0 & 40.5 & 50.76 \\ 19.8 & 39.07 & \_ \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_m \end{pmatrix} = y$$

$$M = \begin{pmatrix} 3.15612279 & 3.32055619 & 2.8046053 \\ 1.94200791 & 3.25275995 & 1.79754018 \\ 3.36247006 & 3.13477718 & 3.75306667 \\ 2.72765219 & 5.74597238 & 1.14535816 \\ 3.21062777 & 3.28008688 & 2.86255375 \end{pmatrix}$$

$$S = \begin{pmatrix} 1.56913487 & 4.46313597 & 0.04110127 \\ 4.28057593 & 4.21979554 & 4.00696943 \\ 4.58024309 & 5.86156674 & 3.98526762 \end{pmatrix}$$

$$\hat{x}_{k=3} \approx \begin{pmatrix} 19.88774894 & 38.76005916 & 45.096574 \\ 17.63866341 & 29.24158276 & 35.12481645 \\ 19.42136163 & 42.65985059 & 48.73261097 \\ 29.97218582 & 40.51216606 & 50.73826944 \\ 19.79503629 & 39.05479727 & 45.33994652 \end{pmatrix}$$

$$\hat{x}_{k=2} \approx \begin{pmatrix} 19.89157822 & 38.76705248 & 45.08909335 \\ 15.40147753 & 29.24164018 & 34.16729664 \\ 19.4175203 & 42.75810799 & 48.73479405 \\ 29.97080536 & 40.50774752 & 50.74237403 \\ 19.79697602 & 39.05335721 & 45.32668706 \end{pmatrix}$$

$$\hat{x}_{k=1} \approx \begin{pmatrix} 21.40909576 & 37.47090348 & 45.44734371 \\ 16.70096795 & 29.23058335 & 35.45290465 \\ 22.33654593 & 39.09415727 & 47.41613992 \\ 24.31880099 & 42.56356526 & 51.62408162 \\ 21.7031853 & 37.98562867 & 46.07163858 \end{pmatrix}$$

Fig. 3. QR Factorization of a sample static measurement matrix X_{lbound} and its covariance matrices M and S is shown for $k = 3$ sensors. The estimated values of $\hat{x}$ are shown for k=3,2,1 with decreasing precision.

applying data cleaning Step 1, which computes matrix coefficient weights of the sensor stream [1,4,10] with missing attributes' values. The transformed sensor data matrix has approximate coefficients for the unknown prior attributes, which gives the best approximation using the observed feature measurements. Figure 3 shows varying estimates of the original sensor data matrix when learning from features k=1,2 and 3. The higher the number of observed features, the better the performance of the learning function to approximate the missing values.

Best Estimate	Temp.	Humidity	Light
Missing Data	16.5%	12.9%	66%
Transductive Label error	(1-0.95%)=5%		

TABLE II
MISSING LABELS (UNSUPERVISED LABEL-LESS LEARNING).

B. Error 2: Network Aggregation

STEP 2: When dealing with streams from mobile trajectory issues, which are related to fading and shadowing, the data cleaning algorithm needs to account for errors induced by high BER due to multi-hop routing. The Data MULE needs to compensate for packet-level MAC errors and provide a minimum required stream QoD to help label and classify the training samples in Step 2. When simulating the various aspects of our mobility model, the percentage of probes are varied, and the communications between the ground sensors are simulated with different network topologies. Wireless simulation of mobile routing allows one to study how Waypoint mobility models and the node's crosslayer BER losses vary in large distributed sensing mobile applications.

C. Error 3: Transductive Label Learning

STEP 3: The feature induction-based learning method does not use training labels; instead the training sample's attributes which best fit the data distribution are induced. Unsupervised feature extraction using random attributes selection used in transductive label learning makes it possible to learn from hidden features (higher order) not obvious in the distribution. The task-based learning approach used by Data MULES, does not use labeled training data from the spatial streams but rather uses unlabeled samples and induces features which reducing the overall training error. Data MULE's can be used in many applications, which do not need any communication infrastructure, such as a

base station. Two such example, are roadside weather monitoring using mobile sensors to alert the driver, and post-safety measures. These stream data values sampled are close to the actual ground truth. For example the road condition seen by the driver could be a real-time update of the local observation affecting safety, such as dense fog, as well as slippery and wet road surfaces. In this step of datacleaning, for the stream to be assigned a safety label, the observed samples need to have contextual parameter relevance from the normal ranking order, or we call the data stream plausible.

D. Datacleaning and Mobility

The main research we undertake here is the adaptive channel coding to be able to distinguish sensor streams using relevance stream QoD labels, learned in the context of channel conditions at the mobile decoding receiver. The receiver needs to accurately predict the corrupted bits representing the current measurement during decoding. Data corruption can be due to packet collision from neighboring nodes, low data-link quality in terms of fading, and shadowing due to mobility segments used by Data MULES.

VI. Results

We select a noisy dataset [1,2,3], which has measurements for temperature, humidity, and light found across a lab area measured using 100 wireless sensors over a period of several weeks. The sensor stream dataset collected from each iteration (epoch) of the sensor network is further divided into spatial clusters. This localized selection allows to estimate the measurements from non-faulty sensors (m) among many faulty sensors by using the best feature set (N), which minimizes the measurement error. Parameters measured locally are shown in Table I, humidity has fewer missing values (12.9%) compared to temperature (16.9%) and light (66%) in our example dataset.

A. Kappa-based Stream QoD

The data cleaning, which is performed in Step 1, handles missing data values' normalization by using the kappa function [1] available in R [6] to label the stream quality. After the matrix completion step, the RMS error of the stream containing best estimates for the missing values is compared with the performance of the original noisy dataset with missing values. From the table in Figure 4(i), the quality of the sensor stream shows an improvement of 27%.

B. BER-based Stream QoD

When analyzing the BER model, we use a QualNet [9] mobile simulator, which measures wireless fading and shadowing losses at the receiver. The decoder performance at the receiver is compared with BPSK and Turbo BPSK error correction codes with block lengths supported by the 802.15.4 protocol. Interference seen at the receiver due to collision [1,3] has been a major source of re-transmission and cause of energy drain. The results which support this are shown in Figures 4(a) and Figure 4(b).

When studying the aspects of fast fading and slow shadowing, we use Data Mule probes in the simulation, which account for 20% of the total static ground sensors. Simulator uses a Random-way-point mobility model. The results of Step 2 data cleaning are shown in Figures 4(b) and Figure 4(c). When using Data MULES, the mobile sensor aggregation errors are reduced by up to 6 times when using BPSK error correction coding, compared to standard uncoded modulation. This is due to the BPSK decoder not able to handle symbol interference, which we have earlier discussed in this section.

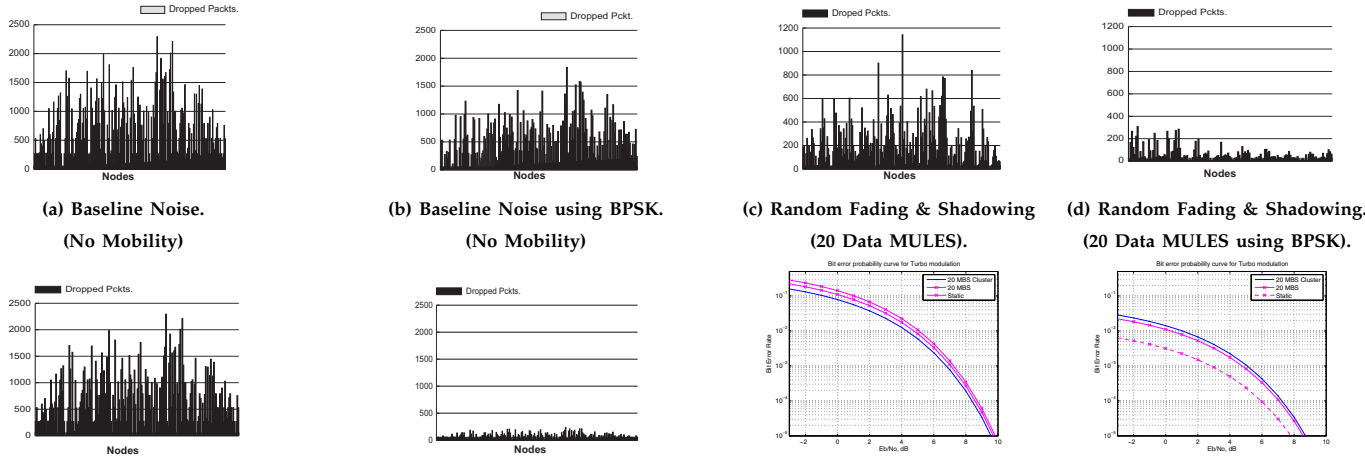
The same simulation is performed using turbo-codes [8]. The simulation results show that tubo codes are more robust to mobile affects of fading and shadowing. During fast-fading, tubo codes are able to use the interleaver to keep the SNR constant during the symbol period while decoding. The shadowing due to slow mobility patterns is completely eliminated with cluster formation, as shown in Figures 4(e) and Figure 4(f), when RSSI (Receiver Signal Strength Index) is optimal within a given wireless coverage area. Figures 4(e) and Figure 4(f) compare the coding gain in MATLAB [8] with varying block sizes of 1000 and 512 symbols at the decoder. The results show that even though the BER do not differ significantly, the fading and shadowing are better handled when ground sensors are clustered and encoding uses a longer code length.

C. Task-based Stream QoD label learning

We measure samples using the mobility framework m*WSN*. The sampling of sensor data are highly probabilistic. The Intel static dataset [5] used earlier is not suitable for unsupervised label learning. Therefore, we use a Monte Carlo simulation method to generate values for the inputs (temperature, humidity and light). The number of random trajectories taken by the Data MULE nodes represents the number of evolutions in the Monte Carlo method. For each iteration of the random inputs, the observed variable Y is saved, and the complete set of sensor samples from the multi-hop path forms a unique trajectory each time. As the distribution is unknown for our dataset attributes, which are temperature, humidity and light, we cannot easily estimate the label error of Y. We use R [6] to define sampling of the observed measurement by using *sample with replacement*, and the command shown below, where t is the number of times mobile sampling is performed.

$$ystar = sample(y, replace = T)$$

The transductive classifier uses random forest algorithm's [7] sampling and replacement boosting to es-



(a) Baseline Noise. (No Mobility)

(b) Baseline Noise using BPSK. (No Mobility)

(c) Random Fading & Shadowing (20 Data MULES).

(d) Random Fading & Shadowing. (20 Data MULES using BPSK).

(e) Random Fading & Optimal Shadowing (Clustering & 20 Data MULES).

(f) Random Fading & Optimal Shadowing (Clustering 20 Data MULES using BPSK).

(g) All Trajectories - BPSK

(h) All Trajectories - Turbo BPSK.

Fig. 4. Stream QoD Fading and Shadowing effects of Data MULES, Clustering with Data MULES and without mobility. Figure 4(g) is a comparison of 802.15.4 using BPSK. Figure 4(h) shows the robustness of Turbo codes with block length of 256.

timate finer temporal variations in the given dataset. We use the WEKA [1] attribute analyzer and set option to 20 fold cross validation. The results on the m*WSN* label learning from a simulated temporal dataset using temprature, humidity and light are shown in Table II. The transductive label error is reduced to 5% when using induced attributes learning utilizing decision trees. The results reflect that when temperature is used as a dependent attribute, the algorithm is able to predict the labels well, even in a small training dataset which has 16.5% of the labels missing.

VII. Summary

Our key findings are that SPOTLESS framework parameter estimates have better precision as they use lower bound of QoD and QoI thresholds, when it comes to correcting spatial measurement errors. The computation complexity of the two categories are given in definitions 1 and 2. Matrix completion always uses a full rank (corrects the sensor local variations) matrix. Therefore, estimates are optimal and closer to the ground truth, even when the data is sparse due to many missing features. The SPOTLESS data cleaning is robust due to random data-link channel errors, which are caused by MAC layer corruption during network packet aggregation. When simulated using SPOTLESS service, the normalized data stream was error-free and its matrix coefficient was close to 95% accurate, compared to original raw sensor data, a 27% increase in spatial consistency. Channel states are monitored, and appropriate variable rate adaption is used at the Data MULE transmitter to minimize the packet aggregation at the network layers. The SPOTLESS data-cleaning service uses soft-phy channel feedback information to estimate BPSK symbol decoding rates to avoid fast fading errors. SPOTLESS service adapts with changing wireless coverage during mobility, which in turn helps reduce decoding errors by 50% and conserves the sensor node's energy by avoiding packet retransmissions. The SPOTLESS service classifies end-to-end GPS location context streams using a task-based multiple tree boosting learning algorithm. Each task is a trajectory taken by the Data MULES from source to sink. The higher the PER for a Data MULES trajectory, the lower is its stream QoD label making it redundant. The simulation results in Figures 4(a-f) support the theoretical estimate of static and mobile BER results using MATLAB, shown in Figures 4(g-h). The SL classifier works off-line to categorize these unsupervised streams into good or redundant samples. The Monte Carlo simulation of the temporal task-based label classifier shows that 95% accuracy is achieved in processing multi-hop data streams.

References

[1] Vasanth Iyer, S.S. Iyengar. Using F-measure attribute performance with test samples from low-cost sensors. In Proceedings- IEEE ICDM, 2011.
[2] Andrew B. Goldberg, Xiaojin Zhu. Transduction with Matrix Completion: Three Birds with One Stone. NIPS, 2010, pp 757-765.
[3] Binbin Chen, Ziling Zhou, Yuda Zhao, Haifeng Yu. Efficient Error Estimating Coding: Feasibility and Applications. SIGCOMM, 2010.
[4] S. Grosanic, A. Dinkel, and S. Piszczek. Optimized traffic control with benchmarked road weather data. SIRWEC 2012, Helsinki, 23-25 May 2012.
[5] Intel senor lab dataset *http : //db.csail.mit.edu/labdata/labdata.html* [Accessed May 15th, 2012].
[6] http://www.r-project.org/ [Accessed May 20th, 2012].
[7] Celine Vens, and Fabrizio Costa. Random Forest Based Feature Induction. In Proceedings- IEEE, ICDM, 2011.
[8] Yuan Jiang. A Practical Guide to Error-Control Coding Using MATLAB.
[9] QualNet is a commercial version of GloMoSim used by (SNT) for their defense projects.
[10] http://www.ogcnetwork.net/SensorML [Last accessed Dec 12/1/2012].