Anonymous Data Collection in Sensor Networks

James Horey, Michael M. Groat, and Stephanie Forrest
Department of Computer Science
University of New Mexico
{jhorey, mgroat, forrest}@cs.unm.edu

Fernando Esponda
Department of Computer Science
Yale University
fesponda@cs.yale.edu

***Abstract*—Sensor networks involving human participants will require privacy protection before wide deployment is feasible. This paper proposes and evaluates a set of protocols that enable anonymous data collection in a sensor network. Sensor nodes, instead of transmitting their actual data, transmit a sample of the data complement to a basestation. The basestation then uses the negative samples to reconstruct a histogram of the original sensor readings. These protocols, collectively defined as a negative survey, are computationally simple and do not increase communication overhead. Thus, the negative survey can be implemented efficiently on existing sensor network platforms.**

We analyze the accuracy of the negative survey under a variety of conditions and define a range of parameter values for which it is practical. We also describe an example traffic monitoring application that uses the negative survey to classify traffic behavior. We demonstrate that for reasonable traffic scenarios, the system accurately classifies traffic behavior without revealing private information.

I. Introduction

Many sensor networks in social settings will require new privacy and confidentiality guarantees in order to protect individual participants[20][16][18].Privacy and confidentiality issues have not been adequately addressed in sensor networks for at least two reasons. First, early sensor network deployments, such as environmental monitoring[3], did not have strong privacy or confidentiality requirements. Second, implementing traditional forms of data protection on resource constrained devices such as sensor networks is difficult. Although recent research has demonstrated that encryption is possible within the constraints of these devices[17][21], the relative overhead remains imposing, and other issues such as key propagation remain challenging[6].

We adapt a set of protocols that enable anonymous data collection[8] and evaluate these protocols in the context of sensor networks. By anonymity, we mean that it should not be possible to match sensitive data to a specific sensor node. Anonymity is accomplished by ensuring that sensor nodes, instead of transmitting their actual data, transmit a data value that was *not* collected. The basestation then uses these negative samples to reconstruct a histogram of the actual data. These protocols, collectively referred to as a negative survey, are computationally simple and do not increase communication overhead. Thus, the negative survey could be implemented efficiently on existing sensor network platforms.

In this paper, we describe the protocols that constitute the negative survey and describe their mathematical properties (Section II). We also discuss the computational requirements of the individual sensor nodes. Section III examines various computational tradeoffs and validates the analysis using a *Matlab* simulation. This information is used to predict the circumstances under which the approach is practical. The results of the analysis suggest that the approach can be employed under a wide variety of conditions.

In Section IV, we describe a hypothetical traffic analysis application that highlights how the negative survey could be applied to classify traffic behavior. Initial results suggest that the negative survey can classify traffic accurately for reasonable traffic scenarios. Section VI discusses related work. Finally, Section VII discusses future work, and we summarize our conclusions in Section VIII.

II. Protocols

The proposed system, referred to as a *negative survey*, consists of two protocols. These protocols and the discussion regarding their information-theoretic characteristics were originally detailed in [8] and are replicated here for completeness. Each node in the sensor network runs a *node protocol* that determines what data a sensor node sends back to the basestation. For example, in the traffic monitoring application, sensor nodes embedded in the vehicle would transmit automobile speeds. Once the sensor nodes have propagated the relevant data to the basestation, the basestation then runs a corresponding *basestation protocol* to reconstruct the statistical distribution of the data.

A. Node Protocol: Selecting a Negative Category

The node protocol determines what data the sensor node transmits to the basestation. This protocol selects data from a finite set of discrete data values. For many applications, such as traffic monitoring, these data values represent mutually exclusive and exhaustive categories. For example, categories for traffic monitoring would consist of speed increments ($0-9$ mph, $10-20$ mph, etc). When discussing the node protocol, we use the terms *categories* and *data values* interchangeably.

We assume that every sensor node chooses from the same set of categories. Each node in the sensor network occasionally receives a *query* requesting data. These queries are similar in concept to those used in query-based languages[23][14][15] and are used to trigger the node protocol. We currently assume that each sensor node has exactly one category to report.

Upon receiving a query, the node protocol determines which category the node will transmit to the basestation. The node

first identifies the initial category p. Instead of transmitting p to the basestation, the node selects another category uniformly at random and transmits this category. More precisely, let U be the set of all categories. The protocol then chooses a category uniformly at random from the set $U - \{p\}$. In this way, nodes are said to transmit *negative* values. If the sensor node transmits p, the original category, we say the node is participating in a *positive survey*.

If an adversary intercepts a transmission from a sensor node, he or she learns only a category that the sensor node did not record. Assuming that there are more than two categories, the protocol preserves a high degree of privacy by making it difficult to correctly guess the actual category which was sensed. The node protocol is computationally simple and does not increase communication overhead because the number of messages transmitted remains the same compared to a positive survey.

B. Basestation Protocol: Reconstructing the Histogram

Once the sensor nodes transmit the negative values, the basestation protocol reconstructs the original frequency distribution. This protocol assumes that the basestation knows both the number of sensor nodes and the set of categories used by the nodes. Assume that there are t categories and n sensor nodes. Let R_i be the reported count for category i transmitted by the sensor network. Let A_i be our estimate of the number of nodes that belong in category i. Finally, for a particular category i, let $C_{i,j}$ be the expected number of sensor nodes in category j that report i.

In order to calculate A_i for all i, the basestation protocol uses the equation:

$$A_i = \sum_{j \neq i} (R_j - \sum_{k \neq i,j} C_{j,k})$$

The node protocol dictates that given a category to which the sensor node belongs, the probability of selecting another category is $\frac{1}{(t-1)}$, giving

$$\sum_{k \neq i,j} C_{j,k} = (\tfrac{1}{t-1}) \sum_{k \neq i,j}^{t} A_k$$

Finally, by observing that:

$$\sum_{j \neq i} R_j = n - R_i$$
$$\sum_{j \neq i} A_j = n - A_i$$

we derive:

$$A_i = n - R_i(t-1)$$

Because A_i represents the estimated number of sensor nodes in category i, dividing both sides of the equation by n gives the relative proportion (represented as $\hat{A}_i$). Using this estimate, we next estimate the variance associated with each category:

$$var(\hat{A}_i) = \tfrac{(t-1)^2}{n-1}(\tfrac{R_i}{n})(1 - \tfrac{R_i}{n})$$

Similarly, the covariance with respect to two proportional category estimates is given by:

$$cov(\hat{A}_i, \hat{A}_j) = -\tfrac{(t-1)^2}{n-1}(\tfrac{R_i}{n})(\tfrac{R_j}{n})$$

C. Discussion

Due to the limited computational capabilities of current sensor nodes such as the Mica2[1] and TelosB[2] platforms, it is desirable to shift computational burden to the basestation when possible. This division of labor extends the overall lifetime of the sensor network[11]. The negative survey gives an example of how this can be achieved in a way that enhances the privacy of individual observations.

The basestation protocol itself could be distributed by allowing nodes, other than the basestation, to reconstruct a partial histogram. These partial histograms could then be merged to produce a final histogram. Investigating distributed basestation protocols is a subject of future work.

The node protocol is simple and adds only a small computational overhead: choosing the negative data category requires randomly selecting an index in the array representing all the categories and checking to ensure that the selected value is not equal to the original data category. We assume that choosing a random value takes constant time. The node protocol also does not increase communication overhead. Because the negative answers are drawn from the same pool as the positive answers, there is no increase in message size.

The basestation protocol runs in $O(t)$ time, where t is the number of categories used in the survey, assuming that all the R_i values have been tabulated.

The confidentiality of sensor nodes relies on minimizing the amount of information gained by an adversary. Because an adversary who observes a single transmission from a sensor node learns one category that the node is *not* a member of, the adversary learns a small amount of information about the node. The amount of information can be characterized using Shannon's uncertainty measure, in which the amount of information gained from a *positive* survey can be written as:

$$-\sum_i p_i \log\, p_i \tag{1}$$

where p_i is the probability of category i being true.

The information gained from a negative survey, in which only one category X_s is selected, can be computed as the difference in information of two positive surveys: the information obtained by the positive version of the survey (given in Eq. 1), minus the information gained from the same survey once X_s is no longer an option. This can be written as:

$$-\sum_i p_i log p_i + \sum_{i \neq s} P(X_i = T | X_s = F) \log\, P(X_i = T | X_s = F)$$

where $P(X_i = T | X_s = F)$ is the probability that category i is true in a positive survey after X_s has been removed as an option. It is easy to see that the information gained from a negative survey is at most the quantity obtained from its positive counterpart.

[1] www.xbow.com
[2] www.moteiv.com

III. Evaluation

There is a tradeoff between protecting the confidentiality of a node's data values and the ability of the basestation to reconstruct the data. This tradeoff can be managed for particular applications by varying the number of sensor nodes participating in the survey and varying the number of categories each sensor node transmits.

A. Methods

We simulated the node and basestation protocols in *Matlab* to test the accuracy of the reconstructed distribution and to determine the conditions under which it can be applied. We varied the number of sensor nodes, the number of categories used in the survey, and the distribution of the data (positive) values, while restricting a sensor node to choose only a single category per query.

For our tests, we pre-selected a distribution; each node was assigned a random variable drawn from that distribution, which indicated its *positive* category. The simulation ran the node protocol on each sensor, transmitted the negative data, and then ran the basestation protocol to reconstruct the distribution. This procedure allowed us to compare the results of the histogram reconstructed from the negative data with the actual positive data.

We tested the algorithm on three different distributions: normal, exponential, and uniform. The uniform distribution chooses each category with uniform probability between 0 and 1. Each test was run with twelve categories and 6000 sensor nodes. We normalized and compared the reconstructed histogram with the original using the following root mean-square error (RMSE) test:

$$RMSE = \sqrt{\sum_{i=1}^{n}(positive(i) - negative(i))^2}$$

As Figure 1 shows, the reconstructed histogram matches the original distribution well for all three distributions. However, the negative survey occasionally generates negative solutions for some of the categories. Negative values arise when the expected contribution for a particular category exceeds the actual reported total for that category. This is a statistical artifact; as the number of samples is increased the number of negative solutions will also decrease.

B. Varying the Number of Categories and Samples

In order to understand the effect of the number of categories and samples on error, we conducted tests that varied the number of sensor nodes participating in the survey, and the number of categories from which each sensor node must choose. Intuitively, we expect the error to decrease as the number of samples increases, assuming a constant number of categories. We also expect the error to increase as the number of categories is increased since the number of choices for the sensor node also increases.

For the first test, we used 6000 sensor nodes and varied the number of categories from 4 to 204 in increments of 2. We ran this test independently ten times and took the average error from all ten runs. We ran this test for the normal, uniform, and

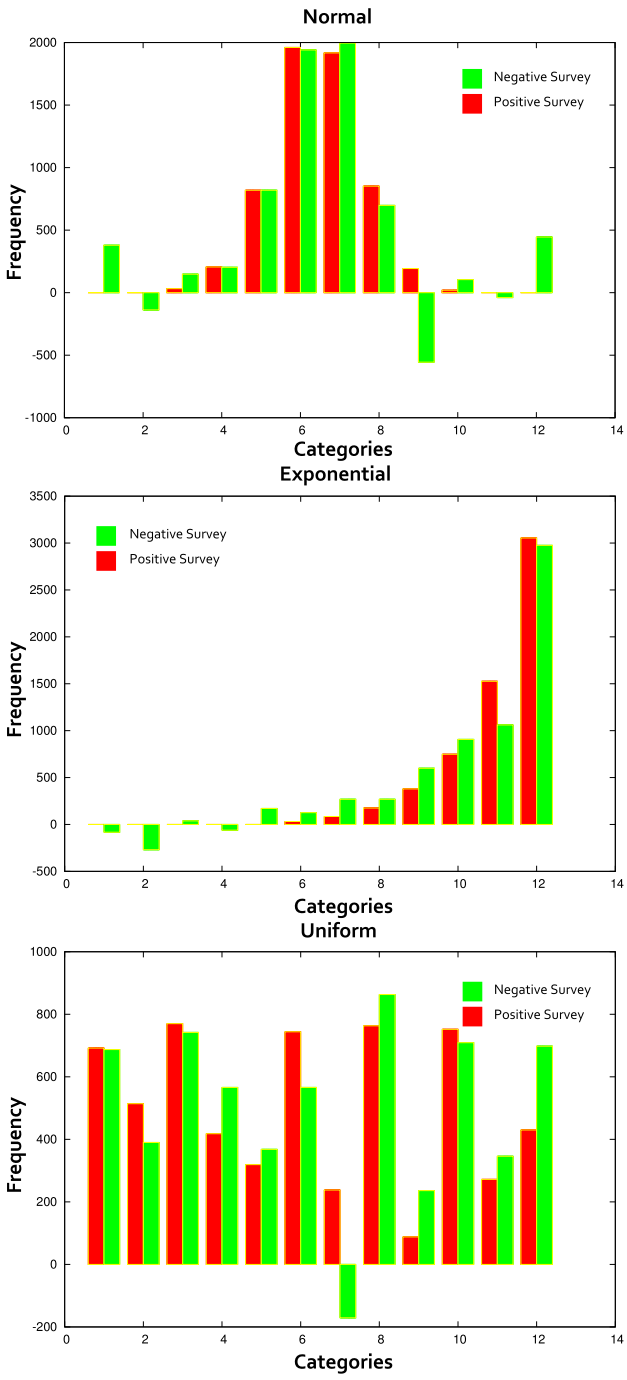

Fig. 1. Each panel shows the reconstructed histogram with the corresponding actual histogram for three distributions. Each trial used twelve categories and 6000 sensor nodes.

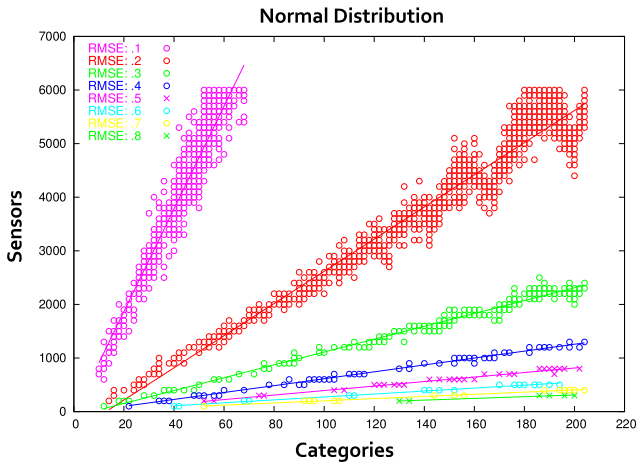

Fig. 3. As the number of categories is increased, the number of sensor nodes needed to maintain a constant RMSE also increases. Each point represents a measured RMSE value that is within a $\pm.008$ range of the tested RMSE values.

exponential distributions. Figure 2 shows that error increases with the number of categories in a near-linear fashion.

For the second test, we tested the effects of varying the number of sensor nodes. We used 14 categories and varied the number of nodes from 100 to 6000 in increments of 100. Again, we ran this test for normal, uniform, and exponential distributions and plotted the average error from ten independent runs. Figure 2 shows that the error falls off quickly as we increase the number of samples and then levels off.

In order to compare the error associated with the negative survey to a baseline sampling error, we ran a simple positive survey with 14 categories and varied the number of sensor nodes from 100 to 6000. We used RMSE to characterize the difference between the positive survey histogram and the actual histogram used for the test. As expected, the error quickly decreases to zero for both the normal and exponential distributions as we increase the number of sensor nodes (data not shown). Although the error associated with the uniform distribution also decreases, it does not reach zero due to the nature of the distribution.

C. Maintaining a Target Accuracy

Given a target RMSE, we tested the relation between the number categories and the number of sensor nodes. This information is useful for applications where a tolerable error threshold is known beforehand. In Figure 3, we plot the number of sensor nodes needed to maintain a target RMSE while increasing the number of categories. We did this for target RMSE values between $0.1 - 0.8$ in increments of 0.1.

As the number of categories increases, we must also increase the number of samples to maintain a constant RMSE value. All three distributions (normal, exponential, and uniform) behave similarly. This implies that we can adopt one method for maintaining a constant accuracy without necessarily knowing the distribution of the positive data ahead of time.

IV. Applications

This section describes how the negative survey could be used in privacy-sensitive applications. The negative survey is appropriate for applications in which the distribution of data is important rather than specific answers from sensor nodes. For example, an application in which users want to know how busy a restaurant is could aggregate discrete location data (such as a city block) provided by individual users' mobile phones. Applications in which data must be associated with specific sensor nodes will likely require other forms of anonymization.

A. Anonymous Traffic Monitoring

Sensor networks have the potential to simplify automobile traffic monitoring[5] and have been used in real-world applications[12]. Traffic monitoring is used in major cities, for example, to make decisions regarding street layouts. It can also be used to identify bottlenecks due to traffic signals. Traffic monitoring could also be useful for individuals. Some road intersections may be congested, while others may be frequented by dangerous drivers. By monitoring traffic conditions, individual drivers could avoid roads with problematic conditions.

Although aggregated information about traffic could be useful, both for individuals and traffic engineers, most drivers would naturally be reluctant to have their driving monitored for fear of legal or insurance repercussions. If, however, the privacy of individuals could be guaranteed, then the larger community could benefit from aggregated information without loss of individual privacy. Similar considerations constrain the collection of health information in epidemiological settings. Although privacy enhancing databases address some concerns, they generally require the individual to trust that his or her information will be sanitized in a way that protects privacy.

In the traffic monitoring example, the negative survey is used to provide end-to-end anonymity to individual drivers. Observers monitoring the traffic would still have access to the real traffic distribution. Note that in the earlier sections, we considered the case in which a single reading is transmitted to the basestation. For the traffic application, sensors might send a periodic tally, say every five minutes.

We assume that each vehicle is equipped with a speed sensor. The speed sensor records the current speed of the host vehicle and the actual speed limit of the road on which the vehicle is traveling. We assume that the speed limit is provided to the sensor, possibly by a basestation located near the road. The basestation collects the sensor data and performs the histogram reconstruction within a locally constrained area, e.g., a single intersection or section of roadway. As explained in Section II, each sensor contains a list of pre-determined categories. For this application, each category represents a set of relative speeds above and below the speed limit. An example with six categories is given below:

1) 10+ mph over the speed limit
2) 5 - 9 mph over the speed limit

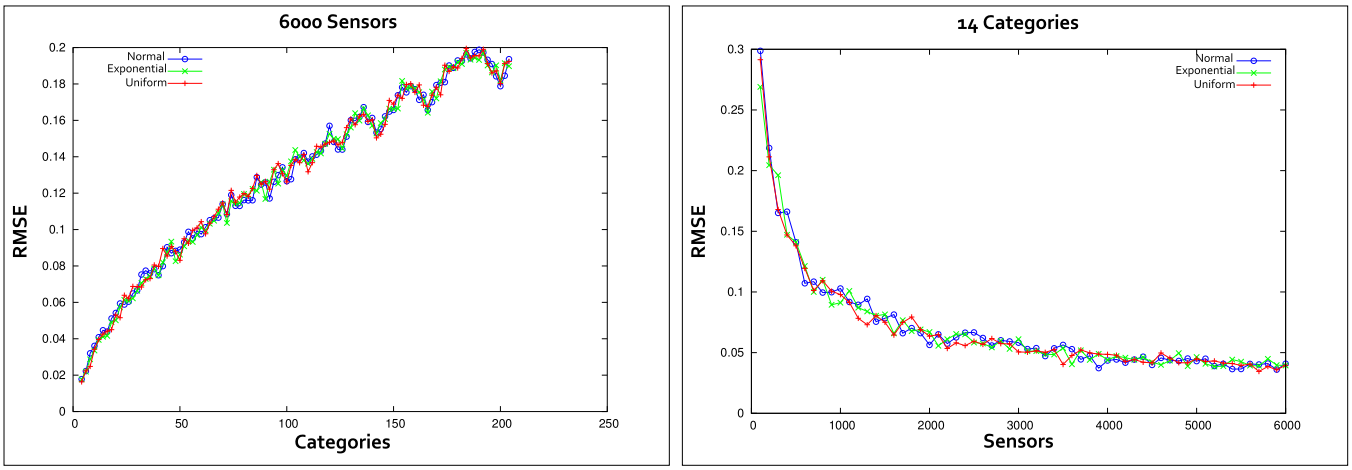

Fig. 2. The error increases with the number of categories in a near-linear fashion. As the number of sensor nodes is increased, the error initially decreases quickly and subsequently decreases at a slower rate.

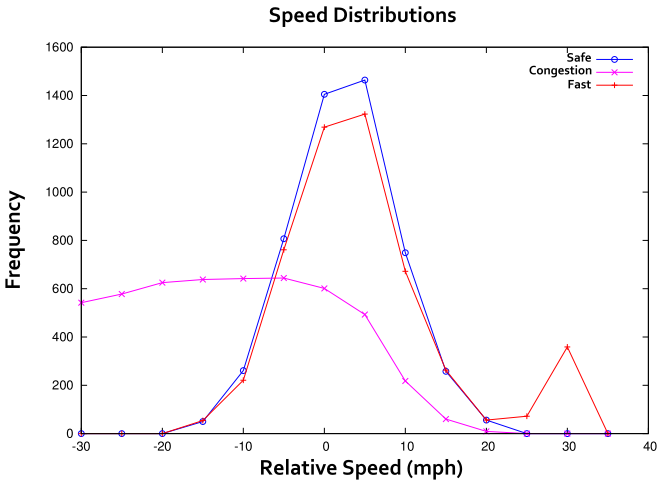

Fig. 4. Three speed distributions characterize different traffic conditions.

3) 0 - 4 mph over the speed limit
4) 0 - 4 mph under the speed limit
5) 5 - 9 mph under the speed limit
6) 10+ mph under the speed limit

In order to determine the proper category, each sensor node takes the difference between its current speed and the known speed limit and chooses a category according the node protocol. The sensor node then transmits this negative value to the basestation.

The basestation, in turn, receives data from all the sensors and reconstructs the histogram. After constructing the histogram, the basestation classifies the histogram into one of three traffic behaviors. Each traffic behavior is distinguished by a canonical speed distribution as illustrated in Figure 4. These speed distributions attempt to capture *congested*, *safe*, and *fast* traffic behaviors. We assume that all traffic obeys one of these three behaviors.

A *safe* speed distribution is characterized by a normal curve centered in the *0 - 4 mph* category. A *congestion* speed distribution is characterized by a skewed-normal curve that leans towards the categories under the speed limit. Finally, the *fast* distribution is a bi-modal curve. The larger mode represents speeds centered near the *0 - 4 mph* category, while the smaller mode is centered near the faster speeds. These speed distributions were derived from real-world patterns[4].

The simulation recorded the average classification accuracy with respect to the number of vehicles participating in the survey. The accuracy was measured as the ratio of the number of correct classifications to the total number of classifications over ten independent runs. We varied the number of vehicles from 100 to 10000 in increments of 100. Each sensor had access to 12 speed categories. We ran the experiment once for each of the speed distributions. Within a single experiment, the actual speed distribution was assumed to remain constant.

After the basestation constructed the histogram, the negative survey results were compared to the three canonical speed distributions using a modified RMSE test. The comparison that yielded the lowest RMSE value was chosen as the actual speed distribution. Results of this test are shown in Figure 5.

In order to validate the algorithm, we also ran the test using a positive survey protocol. 100% accuracy was observed using the positive histogram for all settings. Therefore, we conclude that any error is due to the inaccuracy of the reconstructed histogram.

The classification scheme performed well for all three speed distributions. On average, classification accuracy reaches 80% with 3000 readings. Increasing the number of vehicles increased accuracy to over 90% and eventually to 100%. More complex classification algorithms could increase the accuracy (or the number of categories could be reduced) to improve accuracy in settings with a low number of vehicles. However, $4000-6000$ vehicles in a traffic area is consistent with typical highway and interstate flow[3]. These results illustrate the kind

[3]http://www.mrcog-nm.gov/maps_on-line.htm

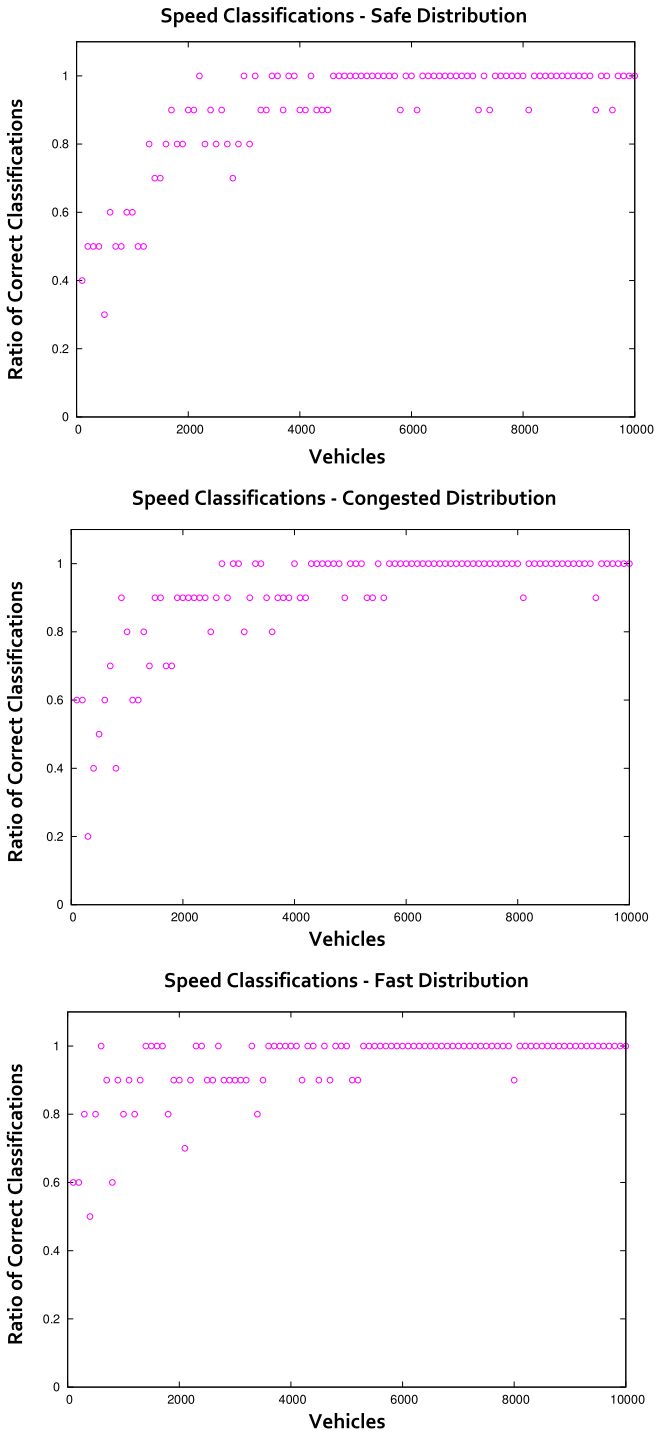

Fig. 5. Classification accuracy for three traffic conditions. Each point is the percent correct classifications in ten trials.

of data rates that would be appropriate for a negative survey approach.

The purpose of this application scenario is not to demonstrate a real-world classification algorithm for traffic monitoring. Real-world deployments would likely include more kinds of traffic behaviors and use more sophisticated classification algorithms. The application does show, however, that a negative survey could supplement existing applications to increase anonymity.

V. Comparison to Random Data Perturbation

The negative survey resembles the data perturbation algorithms proposed by Agrawal *et al*[1] and more recently by Zhang *et al*[24]. Their technique perturbs the original data set with random noise drawn from a known distribution. This perturbed data is then used to reconstruct the original distribution using an iterative algorithm based on Bayes Theorem.

The data perturbation algorithm assumes that the data and the additive noise are both drawn from a continuous domain. However, our work assumes the opposite: the data are drawn from a set of discrete categories. To compare the two approaches, we modified the data perturbation algorithm to handle data and noise drawn from a discrete set of categories. With our modification, it is possible that the perturbed value will lie outside the discrete data domain. In that case, we can either perform the modulo operation on the perturbed data or allow the boundary domain values to subsume all extreme data values. For our tests, we used the former method.

We compared the negative survey and the data perturbation algorithm on two datasets. Each dataset consists of a unimodal Gaussian distribution. For the perturbation algorithm, the noise values were drawn from the set of discrete categories according to a Gaussian distribution. The first dataset contained 12 categories and 6000 sensor nodes. The second dataset contained 64 categories and 32000 sensor nodes. We chose interval sizes of three and four categories for the perturbation algorithm and assumed that the distribution within an interval was uniform.

Figure 6 shows that both algorithms perform well and are able to identify the distribution for 12 categories. However, the negative survey reconstructs the original distribution more accurately. This is because, for a low number of categories, the data perturbation algorithm produces perturbed values that exceed the data domain set. The modulo operation modifies the noise distribution, making reconstruction prone to error.

For 64 categories, the Agrawal algorithm is able to accurately reconstruct both the flat tail and the mode of the distribution. The negative survey, on the other hand, is able to reconstruct the mode of the distribution but fails to accurately reconstruct the tail. We conclude that the negative survey is advantageous for applications with a relatively small number of discrete categories. The random data perturbation algorithm is more applicable for scenarios involving a large number of discrete categories or a continuous data domain.

Finally, reconstructing the histogram using the Agrawal algorithm is relatively computationally intensive. The algo-

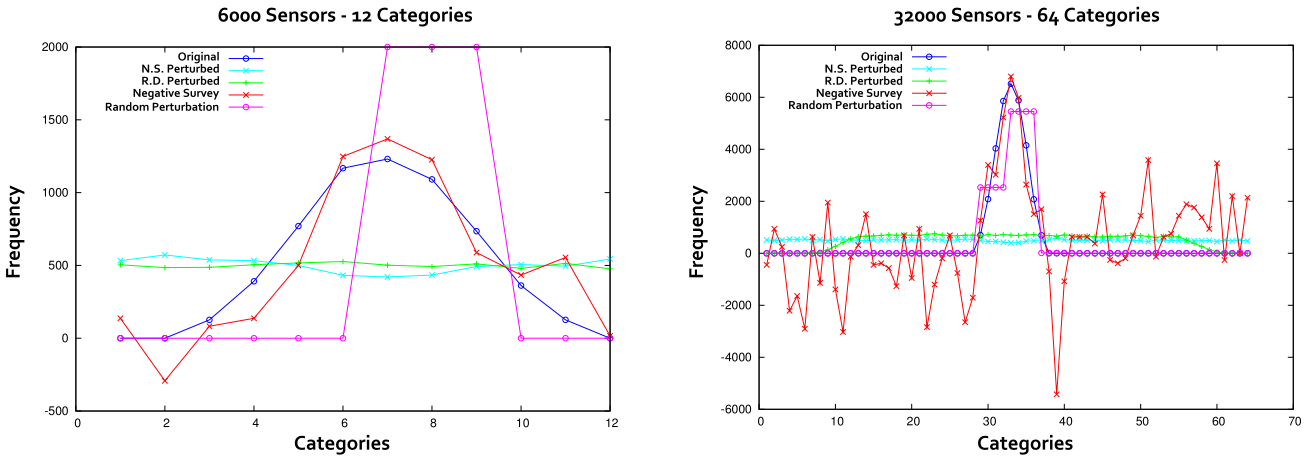

Fig. 6. Comparison of the negative survey and random data perturbation on a unimodal distribution for a low and high number of categories.

rithm is quadratic[1] with respect to the number of intervals and is run multiple times until the algorithm converges. Our algorithm, however, is strictly linear with respect to the number of categories. This implies that the negative survey can be run on less capable basestations.

VI. Related Work

The negative survey[8] is an extension of *negative databases*[10][9]. The negative database, similar to the negative survey, stores a compressed form of the complement of a data set instead of the actual data. The negative database is able to answer certain queries, such as SELECT, in reasonable time. Reconstructing the entire positive database, however, is known to be NP-Hard. Unlike negative databases, our methods do not explicitly store the set of all strings representing the data complement.

Randomized response techniques (RRTs)[22] are an alternative surveying method designed to estimate the proportion of a population that belongs to a particular group while protecting the privacy of individuals participating in the survey. It does this by offering surveyors two possible questions in lieu of a single question. For example, the interviewee might be asked:

Q1: Do you belong in Group A?

Q2: Do you belong in Group B?

Individuals are asked to select one of the two questions to answer given a randomizing device provided by the interviewer. Individuals give a yes or no answer to one of these questions, but do not reveal which question was answered. In this way, the results of the survey combined with the characteristics of the randomizing device provides enough information to reconstruct the proportion of population members in each group.

Zhu et al[25] propose using silence to communicate useful information between sensor nodes. Although this method is in a similar spirit to the work reported here, it does not attempt to provide anonymity. Instead the protocol attempts to reduce communication overhead while ensuring confidentiality between two nodes. This is accomplished via start and stop tokens; the amount of time between the start and stop tokens represents the data to be transmitted. This method has the advantage of low communication overhead, although transmission bandwidth may suffer. Currently, a complete protocol based on this unique idea does not exist.

Data privacy can be ensured using cryptographic techniques and recent work has shown that it is possible to use encryption techniques on existing sensor platforms[21][13]. However, the computational costs are still large compared to our protocol, and key distribution remains a challenge[6].

Our work has similar characteristics and goals as secure election algorithms based on cryptographic methods. However, secure election algorithms generally involve the exchange of private and public keys to sign and encrypt the relevant election results. Using public and private keys, accurate election results can be tabulated with zero or more central tabulating facilities. For more information regarding this class of algorithms, the reader is referred to [19].

Secure multiparty computation algorithms allow nodes to compute any function of many variables without each node knowing the inputs of the other nodes. For instance, secure multiparty algorithms can be used to calculate the average salary of a group of people without the individuals learning the actual salary of each person. These algorithms, unlike the negative survey, require cryptographic methods and often require synchronized communication between a known number of participants. Recent work on collaborative filtering with privacy could also be adapted for the scenarios we are addressing[2].

VII. Future Work

Social aspects of negative surveys remain an interesting avenue of research. Users may be more willing to participate in negative surveys and may be more honest while taking such surveys This could reduce the amount of bias in the data. Further analysis of the negative survey with respect to surveys conducted by humans, as opposed to sensor networks, is explored by in[7][8].

A major assumption in the current work is that the data from different sensor nodes are not correlated. We assume that each measurement made by a sensor node is independent of other sensor nodes. However, this may not be the case in all applications. Two sensor nodes placed in close geographic proximity may sense correlated temperature values. Therefore, by knowing the locations of these nodes, more information could be gained from a single negative reply and the level of anonymity decreased. Similarly, data reported by a sensor node may be correlated with past data. This additional information could also be used to our advantage to estimate individual data values.

We are interested in using the negative database framework more directly on sensor networks. Assuming that the sensor network is a massively distributed database[15][23], with each node containing a small portion of the entire database, each sensor node could then store its own portion as a negative database. This would increase security in the face of node capture; even if intruders are able to obtain the sensor node physically, it would be difficult for an intruder to reconstruct the entire positive database.

VIII. Conclusion

This paper describes the application of novel methods for sensor networks to collective sensitive data. Our methods, collectively described as a negative survey, do not require any form of encryption, and have low computational overhead. Because the negative survey transmits the same amount of data as a corresponding positive survey, there is no additional communication overhead.

A detailed performance study was conducted, in simulation, with a wide range of parameter values. An interesting result suggested by the simulations is that the accuracy of the reconstructed data is relatively insensitive to the underlying distribution from which the samples are drawn. This property increases the likelihood that we could design a sensor network based on these techniques even when the underlying distribution of data is unknown ahead of time.

Finally, realistic application scenarios, using the negative survey, was described. One of these applications, anonymous traffic monitoring, uses the negative survey protocols to classify traffic behavior. The classification scheme was shown to be accurate for a reasonable number of vehicles in high traffic areas.

The negative survey technique shows how the redundancy of sensor networks can be exploited to enhance privacy. As sensor network applications become more human-centric, we expect that this basic principle of trading redundancy for privacy enhancement will become an important tool in sensor-network design.

IX. Acknowledgments

We thank Eric Trias and Elena Ackley for important suggestions and ideas. The authors gratefully acknowledge the support of the National Science Foundation (grants CCR-0331580, CCR-0311686, CCF 0621900), the Santa Fe Institute, Motorola, and Los Alamos National Laboratory (grant DE-AC52-06NA25396).

References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. *SIGMOD*, 29(2):439–450, 2000.

[2] J. Canny. Collaborative filter with privacy. *IEEE Symposium on Security and Privacy*, pages 45–57, 2005.

[3] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001.

[4] P. P. Dey, S. Chandra, and S. Gangopadhaya. Speed distribution curves under mixed traffic conditions. *Journal of Transportation Engineering*, 132, 2006.

[5] S. C. Ergen, S. Y. Cheung, P. Varaiya, R. Kavaler, and A. Haoui. Demonstration: Wireless sensor networks for traffic monitoring. In *IPSN*, 2005.

[6] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *CCS*, 2002.

[7] F. Esponda. *Negative Representations of Information*. PhD thesis, University of New Mexico, 2005.

[8] F. Esponda. Negative surveys. *ArXiv Mathematics e-prints*, Aug 2006.

[9] F. Esponda, E. S. Ackley, P. Helman, H. Jia, and S. Forrest. Protecting data privacy through hard-to-reverse negative databases. In *Information Security Conference*, 2006.

[10] F. Esponda, S. Forrest, and P. Helman. Enhancing privacy through negative representations of data. Technical report, University of New Mexico, 2004.

[11] R. Govindan, E. Kohler, D. Estrin, M. Vieira, J. Paek, O. Gnawali, A. Joki, K.-Y. Jang, and B. Greenstein. The tenet architecture for tiered sensor networks. In *SenSys*, 2006.

[12] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: A distributed mobile sensor computing system. In *SenSys*, 2006.

[13] C. Karlof, N. Sastry, and D. Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *SenSys*, 2004.

[14] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The Design of an Acquisitional Query Processor for Sensor Networks. In *SIGMOD*, pages 491–501, 2003.

[15] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Transaction Database Systems*, pages 122–173, 2005.

[16] A. Parker, S. Reddy, T. Schmid, K. Chang, G. Saurabh, M. Srivastava, M. Hansen, J. Burke, D. Estrin, M. Allman, and V. Paxson. Network system challenges in selective sharing and verification for personal, social, and urban-scale sensing applications. In *HotNets*, 2006.

[17] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. Spins: Security protocols for sensor networks. In *MobiCom*, 2001.

[18] S. Reddy, A. Parker, J. Hyman, J. Burke, M. Hansen, and D. Estrin. Image browsing, processing, and clustering for participatory sensing: Lessons from a dietsense prototype. June 2007.

[19] B. Schneier. *Applied Cryptography Second Edition*. John Wiley and Sons, Inc., 1996.

[20] E. Shi and A. Perrig. Designing secure sensor networks. *IEEE Wireless Communications*, 2004.

[21] H. Wang, B. Sheng, and Q. Li. Elliptic curve cryptography-based access control in sensor networks. *Int. J. Security and Networks*.

[22] S. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[23] Y. Yao and J. Gehrke. The Cougar Approach to In-Network Query Processing in Sensor Networks. In *SIGMOD*, 2002.

[24] S. Zhang, J. Ford, and F. Makedon. Deriving private information from randomly perturbed ratings. In *Siam Conference on Data Mining*, 2006.

[25] Y. Zhu and R. Sivakumar. Challenges: Communication through silence in wireless sensor networks. In *MobiSys*, 2005.