Applying Dynamic Markov Random Fields for Sensor Data Analysis

Jong-Moon Kim

Abstract

In this paper, we present a probabilistic model to answering queries across a sensor network with limited and stochastic information. This approach is practical as it allows analysis of real-world data that accounts for environmental noise, corruption of data, and loss of integrity over the sensors. We use a Markov Random Field (MRF) that varies over time, also known as a Dynamic Markov Random Field (D-MRF), to model the sensor network and implement Markov Chain Monte Carlo (MCMC) sampling in conjunction with the Value Iteration algorithm to perform inference. We demonstrate our methodology on a specific scenario of analyzing forest fires, where a sensor network is implemented to localize and predict the current and future behaviors of fires. This algorithm provides for the efficient processing of imperfect data, allowing the extraction of information for locales where sensors are not present at all.

Introduction

Every year, tens of thousands of wildfires nationwide decimate forests, not only threatening wildlife, but inflicting human suffering and economic loss as well. Based on the National Fire Information Control (NFIC), the United States spent over $1.3 billion dollars on the suppression of these fires in 2003 alone [1].

The focus of this research was inspired after the 2003 California Wildfire Disaster ravaged over 800,000 acres of Southern California. The conflagration consisted of twelve significant, distinct fires across the region, where flames reached 100 feet tall in some places. Eight thousand firefighters were called upon to put out the fires in the 25,000 acres of San Diego city, from which more than 40,000 people were evacuated[2]. This disaster highlighted the need to accurately locate the beginnings of small fires to earlier detect a larger fire, as well as the importance of formulating an effective firefighting strategy.

With the advent of mobile processing power, the construction of wireless sensor networks has become a reality [3]. Small modules containing a processor, an array of sensors, and a wireless modem have been developed by universities and corporations. However, there are areas that need improvement, in order to enhance practicality in battery life and system miniaturization. Sensor networks have been shown to be viable for a variety of applications including environmental monitoring, target localization and tracking, manufacturing automation, surveillance intelligence, and observing natural phenomena. With a large population of these sensors, they can be distributed across an environment; in the case of a forest, they can form a sensor network. Low-cost environmental sensors with simple functionality can be scattered from the airplane and then implemented to observe a widespread area for any particular event out of the norm by communicating with a central node for processing. As these sensors feed a continuous source of data, the collective sensor network is able to generate a map portraying the current situation [4]. The objective of the deployment of a sensor network can be defined as:

1. to ascertain whether a significant fire of a certain magnitude exists,
2. to determine where exactly the fires are located,
3. to predict where the fires will spread, given a number of time steps, and
4. to understand how to enhance the network through the placement of additional sensors.

In an ideal world, sensors would be distributed densely in a regular fashion in strict grid-like ordinate positions returning perfect deterministic data. In reality, however, distribution of sensors in such a uniform manner proves to be unfeasible, and even in practice, the network will not have the extremely dense coverage that is desired. A particular area of focus may lie outside the observation range, thereby preventing an accurate representation. As with any man-made equipment, it is also inevitable that sensors will malfunction occasionally, returning incorrect data. The key to successful sensor network algorithms is to infer as much coherent information as possible from an evidence base that may be noisy, corrupt, and erroneous.

We present an algorithm that has the ability to incorporate stochastic, limited data to extract accurate renditions. The previously specified concerns are addressed through the implementation of a probabilistic approach to this problem. As each factor is associated with a probability, instead of simply providing a conclusion with questionable reliability, it provides a wider perspective of information, along with an increased accuracy, allowing more complex analysis to predict the current situation.

Problem Definition

Variables

To better understand and analyze our problem, let us establish it in a mathematical framework. First, let the work space W be a P by Q grid over the world with individual elements $W_{p,q}$. This discrete representation effectively approximates a continuous distribution with sufficiently large P and Q. Figure 1 shows an example forest area (left) and its mathematically mapped two-dimensional arrays. In this study, we assume each cell contains certain environmental information to analyze the current situation but not in a complete manner.

Second, each grid cell can either be on fire or not be on fire. Specifically, let $F^t = \{F^t_{1,1} \ldots F^t_{P,Q}\}$ be a set of P by Q Boolean variables where $F^t_{p,q}$ is true if and only if the area at grid cell $W_{p,q}$ is on fire at a given time step *t*. An alternative representation we considered was to presume the existence of a set of fires, each of which existed over one or more grid cells. The current representation, though perhaps counter-intuitive, robustly addresses events such as: a new fire starting, two fires merging, a fire splitting, and a fire dying. We will find that while we do not capture the notion of intensity of a fire, the techniques we use inherently captures and model this notion.

Third, let a subset of the grid cells contain sensors. No grid cell may contain more than one sensor. This scenario could happen from the sensor deployment process, but a simple algorithm can handle this issue. We assume that each existing sensor sends back a single reading, indicating either the detection or no detection of a fire. Low-cost temperature sensors might be sufficient for this role. In reality, some subset of sensors will malfunction, fail to report information, and/or lose its messages in transit. Our algorithm is robust against such failures and treats non-reporting sensors as non-existent sensors. Let S be the set of Boolean variables representing sensor results, and let $S_{p,q}$ specifically represent the result from a sensor at $W_{p,q}$.

Fourth, let each grid cell be one of three basic geographic types – glassland [G], water body [W], or forest [F]. These types will capture how rapidly fire will spread across different sorts of areas. Let $T = \{T_{1,1} \ldots T_{P,Q}\}$ be a discrete variable that takes values G, W, and F.

Relations Between Variables

While some subset of sensors can malfunction by not reporting any information, two other subsets will malfunction by reporting incorrect information. That is, some sensors will report the presence of non-existent fires, and other sensors will fail to report the presence of real fires. We capture this type of malfunction explicitly on our model by linking $F^t_{p,q}$ to $S_{p,q}$ as:

$$F^t_{p,q} = \begin{cases} S_{p,q} & : with\ probability\ \xi \\ 0 & : with\ probability\ \frac{(1-\xi)}{2} \\ 1 & : with\ probability\ \frac{(1-\xi)}{2} \end{cases} \qquad (1)$$

The parameter ξ signifies the probability that the sensor correctly detects the presence of fire. When the sensor malfunctions, with probability 1 - ξ, it returns either the presence or absence of fire with equal probability. Note that the probability of the sensor returning the incorrect answer is (ξ + 1) / 2, which is less than 1 - ξ.


http://www.accem.org/graphics/wildfire_2.jpg

W_{1Q}	W_{2Q}	W_{3Q}	W_{4Q}	W_{5Q}	?	?	W_{PQ}
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
W_{15}	W_{25}	W_{35}	W_{45}	W_{55}	?	?	W_{P5}
W_{14}	W_{24}	W_{34}	W_{44}	W_{54}	?	?	W_{P4}
W_{13}	W_{23}	W_{33}	W_{43}	W_{53}	?	?	W_{P3}
W_{12}	W_{22}	W_{32}	W_{42}	W_{52}	?	?	W_{P2}
W_{11}	W_{21}	W_{31}	W_{41}	W_{51}	?	?	W_{P1}

Figure 1. An example environment and its mathematical representation

Queries

Given this framework, we can also make the kind of queries we seek to answer in a mathematically precise manner. The first class of queries asks whether there are significant fires in the region of interest. Before we answer this query, we must quantify the notion of the existence of a "significant" fire. Let us say that a significant fire exists if and only if there exist N unique grid cells $F^t_1 \ldots F^t_N$ such that $F^t_i = 1$ for all N cells and the cells are contiguous, i.e. $||F^t_i - F^t_i+1||1$ is equal to 1, where $||x||1$ defines an L1 vector norm. Let SIGNIFICANT(F^t) signify that $F^t = \{F^t_1 \ldots F^t_N\}$ meets the two above conditions. Given this definition, the query concerning the existence of a big fire simply becomes: given S, what is P(ξ F^t s.t. SIGNIFICANT(F^t))?

The second class of queries asks where the fires are. There are two different ways to represent this query. The first way would be to ask for a set of fires and a parametric representation of the boundaries of each one. Such an answer would be brittle, suffering from stark discontinuities given different measurements by a small number of sensors. The second way, following the technique used by level set calculations in computer graphics, asks for a posterior probability of fire existing at each grid cell. Specifically, given S, for each p and each q, what is $P(F^t_{p,q} = 1)$ when t is the current time step?

The third class of queries asks a similar question but about the past or future. The query here is, given S, for each p and each q, what is P(Ftp,q = 1) when t is not the current time step? This would be useful in trying to understand where a fire may have started. It is also essential in understanding what areas are mostly likely to be affected next by an expanding fire.

Prior Work

This focus of this study is unique in that there is no precedent for such inference algorithms with sensor data. The traditional approach in this situation is to assume the entire world state is given through sensors and to interpret results directly. Our algorithm adds an additional step, between data collection and interpretation of results, to refine sensor data through probabilistic analysis of sensor reliability and to extract data for unobserved regions by cross-reference neighboring sensors. It is important not to be confused with post-data analysis using collected data; inference is designed to provide more initial information from the sensors before doing any such processes. The application of Markov Chain Monte Carlo (MCMC) to sensor networks is still a relatively nascent field of study. MCMC is best known previously for work in localization and mapping.

In a more general stance, there are two traditional strategies employed by firefighters. The first is the detection through lookout stations using direct visual monitoring to determine the presence of a fire. Second is the use of a helicopter equipped with a GPS tracking device to map out the environment below [1]. While these two methods are effective when a large fire becomes present, they are expensive in terms of manpower and operation costs such that economics becomes the critical factor limiting coverage. As there is no direct way to confirm the validity of reported fires, and a confirmation action must be taken for every instance,, whether it is the deployment of the helicopter or direct visual detection. The problem remains that these reports may be a false alarm, or the fire may be benign or prescribed, thereby rendering the process extremely expensive. Our approach does not merely direct the deployment of an environmental sensor network, but rather improves the algorithmic processes after getting measurement data. Early studies with sensor networks do exist, but they were hampered by the physical deficiencies of the sensors themselves.

Our Approach

Why Probabilistic?

We take a probabilistic stance towards this problem. This approach offers several advantages to a non-probabilistic one. First, a probabilistic technique effectively considers different possibilities for what the state of the world might be and their ratios to the actual chances; it provides insight into the subtleties of the problem and the inherent uncertainty it faces. Non-probabilistic approaches which have no way to cope with their uncertainty instead make strong assumptions that can send them astray. To investigate this difference, we show the stark differences between a common non-probabilistic approach and our probabilistic approach in answering queries of the type proposed in the introduction.

Probabilistic Generative Model

We postulate the relations between the variables presented in the problem definition as a Dynamic Markov Random Field (D-MRF). The first set of nodes in the D-MRF are the T times P times Q Boolean variables F = {F11,1 ... F1P,Q; F21,1 ... F2P,Q; ... FT1,1 ... FTP,Q}, where T is the number of time steps, P is the width of the grid, and Q is the length of the grid. The second set of nodes are the P times Q Boolean variables S = SP,Q. The last node is the query of interest at time step t, Dt. The F and S nodes are identical to the definitions given in the problem. The query Dt is a Boolean variable. Note that the Boolean nature of Dt does not preempt us from asking about the probability of Dt. The node potentials for the F variables encapsulate a prior probability of ξ of Ftp,q being 1. Typically, we use values of ξ around 0.02, indicating that a fire is fairly unlikely to begin with. The threshold value can be fine tuned through the field tests. The other nodes have uniform potentials, i.e. they are primarily dependent on their connections to other variables. Figure 2 shows a D-MRF with five sensor measurements across a 4 by 4 grid. Field structure will vary depending on the actual number of sensors and the size of the grid.

The first set of edges connects the set of variables Ft+1p,q and the sets of variables Ftp,q, Ftp+1,q, Ftp-1,q, Ftp,q+1, Ftp,q-1, and Tp,q. The relation depends on the value of T and is:

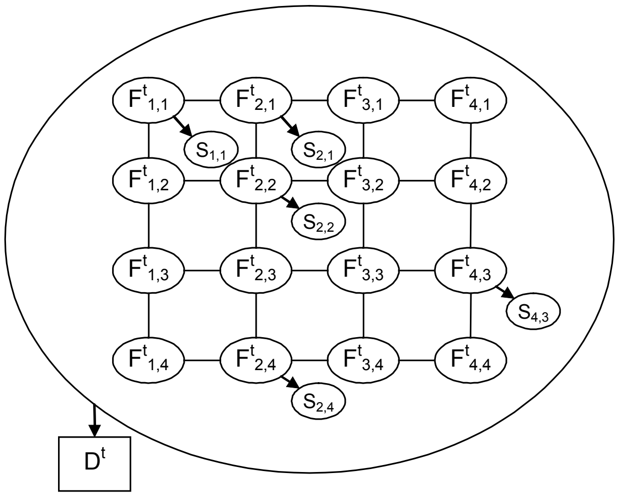


Figure 2: Dynamic Markov Random Field with 5 sensor measurements across a 4 by 4 grid.

$$P(Ft+1p,q) = P(Ftp,q) + (1 - P(Ftp,q)) * [bp,q + (1 - bp,q) * tp,q * (Ftp+1,q, Ftp-1,q, Ftp,q+1, Ftp,q-1) / 4] \qquad (2)$$

where, tp,q is the rate at which the fire spreads. We used the values tp,q = 0 if T = W, 0.3 if T = M, and 0.8 if T = F. The variable bp,q is the probability of a fire starting at random, which is also conditioned on Tp,q. We used the values tp,q = 0 if T = W, 0.002 if T = M, and 0.004 if T = F. The second set of edges connects the variables Ftp,q with the variables Sp,q. The potential for these edges follows the problem definition. The third set of edges links the variables Ftp,q with the Ftp+1,q, Ftp-1,q, Ftp,q+1, Ftp,q-1. For each of the four variables in the set, the potential is ξ match if Ftp,q equals its neighbor and ξ not-match if Ftp,q does not equal its neighbor. We used the values 0.9 and 0.2 respectively for ξ match and ξnot-match. The fourth set of edges links the variables Ftp,q with the variable Dt. This potential is a strict definition that follows from the queries we have asked. That is, given a particular world with certainty, the query is either true or not true. The inference algorithms will consider the subtleties and different probabilities, but the edge potential for D need not consider that.

Inference Algorithms To Answer Queries

While straightforward exact inference algorithms, such as variable elimination or junction tree, exist for discrete MRFs such as the one posited above, these techniques fail to scale to the size of the MRFs we are considering. Thus, we turn to alternative methods, namely, the Markov Chain Monte Carlo (MCMC) sampling in conjunction with value iteration.

Our first query deals with how sensor accuracy influences the accuracy of results obtained from our probabilistic model. MCMC deals with this problem by starting with a random configuration of the world and evolving it according to its internal Markov chain in accordance with the MRF likelihood values. A sub-algorithm determines if a significant fire exists before each iteration, and the majority of the tally of all the results determines the final answer.

Our second query deals with the posterior probability of fire at each location given sensor data. MCMC uses an identical Markov chain, but the query variable D changes to recording whether or not a particular F value was 0 or 1 after each iteration. The average of each tally is used to produce the posterior map.

Our third query deals with the growth of a fire. We start with the posterior generated by the second query and run the MRF forward in time with value iteration. Specifically, we calculate the value of each F variable in expectation at time step t+1, given the probabilities of the neighboring F variables at previous time steps in accordance with edge potentials. This can be done indefinitely, and we can peer as far into the future as we desire. The developed algorithm has been implemented with built-in matrix architecture into MATLAB.

Experimental Results

In this section, we will show the results from some of the queries applied to the inference algorithm. First, we have a specific network to show the posterior probability of a big fire with a varying threshold. Second, we demonstrate the robustness of our inference algorithm through comparison with traditional methods of detection as sensor reliability deteriorates. Third, we use two sets of specific networks to illustrate the posterior estimations of the location of fire. Fourth, we use a specific network to demonstrate the prediction through the value iteration function.

Inference of Probability of a Large Fire

The first and foremost question that is posed concerning our environment is whether there exists a potentially malignant fire.

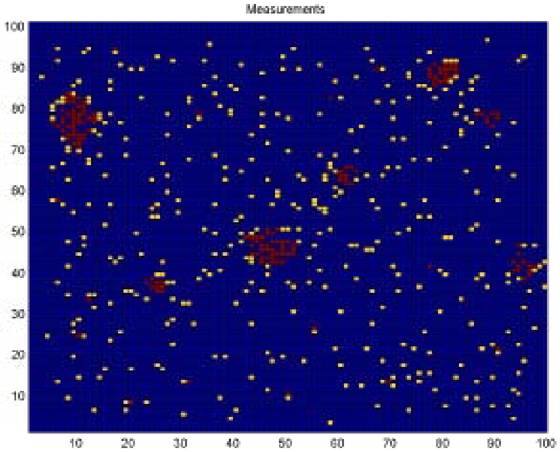
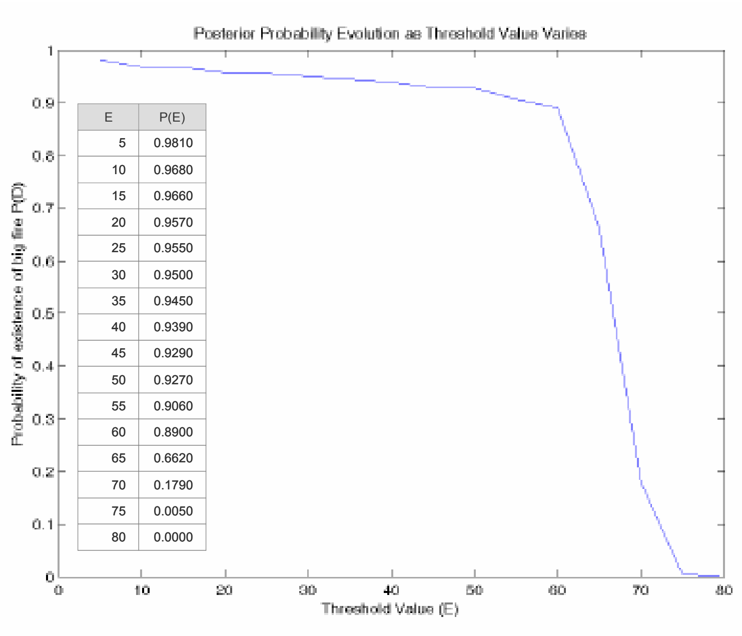


Figure 3. An example measurement return from a sensor network. Red corresponds to sensors that have detected fire, yellow corresponds to sensors that have not detected fire, blue represents no information available.

Figure 3 is a representation of a sensor network. In this example, the environment is enumerated to 100 by 100 grid cells with 300 fire cells and 600 sensors. The algorithm is run with $\xi = 0.01$ with 1E6 iterations; none of the sensors failed in the simulation.

Figure 4 shows the posterior probabilities returned by the inference algorithm. A threshold value of 0 would automatically return 1 as long as at least a single fire cell was present. As the value is increased, the likelihood of MCMC returning a posterior probability with a large fire is decreased. At a threshold value of around 60 fire cells, a drastic drop is observed, demonstrating the ability of the algorithm to determine the number of alternate contiguous configurations.



E	P(E)
5	0.9810
10	0.9680
15	0.9660
20	0.9570
25	0.9550
30	0.9500
35	0.9450
40	0.9390
45	0.9290
50	0.9270
55	0.9060
60	0.8900
65	0.6620
70	0.1790
75	0.0050
80	0.0000

Figure 4: Posterior probability as a function of threshold value for a large fire.

As MCMC is able to successfully determine the likelihood according to a variable threshold, the inquirer may run multiple queries to gather the most likely size and configuration of the fire. This is one of the redeeming features of the probabilistic approach of MCMC - instead of simply providing an inflexible Boolean answer, it can provide a relative reference to each individual situation.

Comparative Inference Given Deterioration of Sensor Accuracy

We now compare the statistical benefits of implementing MCMC as opposed to a traditional algorithm which interprets the sensor results directly, rather than generating any inference models. As the accuracy of the sensors drops, it is inherently true that any algorithm will fail eventually. The key lies in the ability of an algorithm to retain its reliability even when presented with inaccurate data. The ξ value is fluctuated to simulate the progressive malfunctioning of the sensor network.

Figure 5 shows the aggregate of two sets; one inferred from MCMC and the other interpreted from the traditional algorithm. When the sensors are perfectly accurate, both MCMC and the traditional algorithm reflect complete correctness. It is only after the sensor accuracy falls below 90% that the two algorithms deviate. As the sensor accuracy decreases, the MCMC algorithm is able to maintain a reliability level of 75%, while the traditional algorithm dips below 50%.

The ability of MCMC to account for erroneous data becomes invaluable when it is necessary to make life and death decisions with faulty sensors. Whether there is overexposure to heat, or damage to the circuitry due to physical impact, the environment is naturally unpredictable. Instead of citing a false alarm or mistakenly overlooking a critical fire, MCMC provides a way for the sensors to do a consistency check among themselves to determine a possible error.

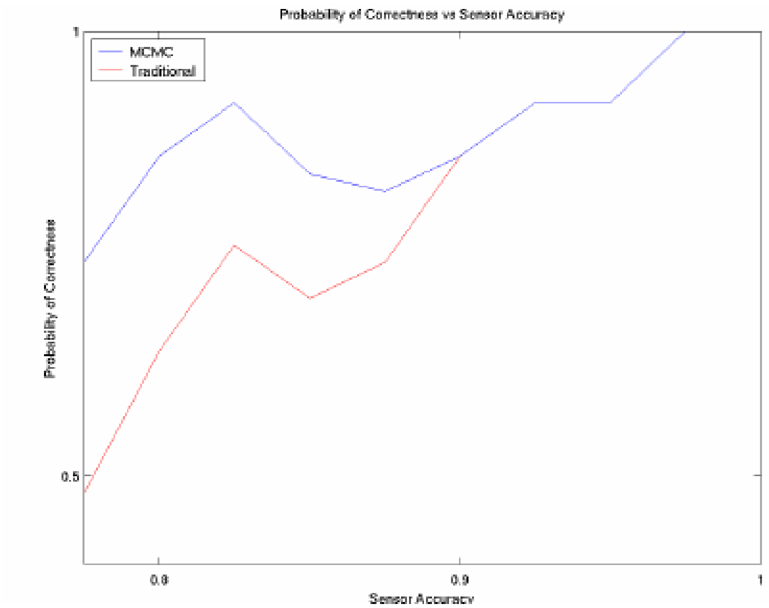


Figure 5. Probability of correctness of the net posterior as sensor accuracy varies.

Determining the Posterior Fire Configuration

We now move on to the determination of the current situation through the weighted average of posteriors. Figure 6 shows a world configuration and the measurements sample from the sensor network. In this example, the environment is enumerated to 10 by 10 grid cells with 14 fire cells and 10 sensors. The algorithm is run with $\xi = 0.01$ with 1E6 iterations.

Figure 7 shows the posterior fire configuration as calculated by our algorithm. The varying gradient represents the likelihood of fire actually being present in the particular grid cell. Whereas the traditional approach would fail to detect a mass of fire as the measurements were not contiguous, our algorithm is able to determine the posterior probabilities of the fire actually being present in the unobserved regions.

Predicting the Future Location of a Fire

It is also important to obtain a prediction of a future model of the fire. Figure 8 is a sample data set returned from a sensor network. In this example, the environment is enumerated to 100 by 100 grid cells with 300 fire cells and 600 sensors. The algorithm is run with $= 0.01$ with 1E6 iterations.

Figure 9 is the visualization of the environmental model. In this specific scenario, this is a rendition of a large lake with surrounding foliage. The model is provided to simulate the

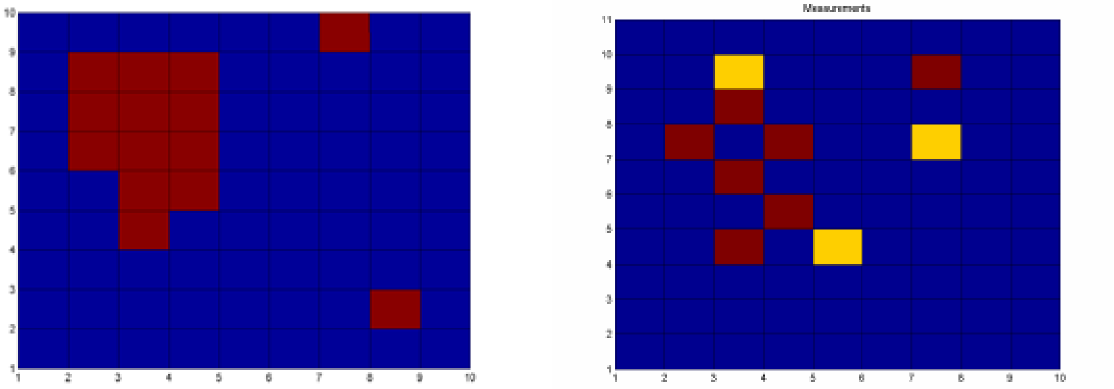


Figure 6. World and sensor network configurations. (left) Red represents fire, blue represents no fire. (right) Red represents sensor detecting fire, yellow represents sensor not detecting fire, blue represents no information.

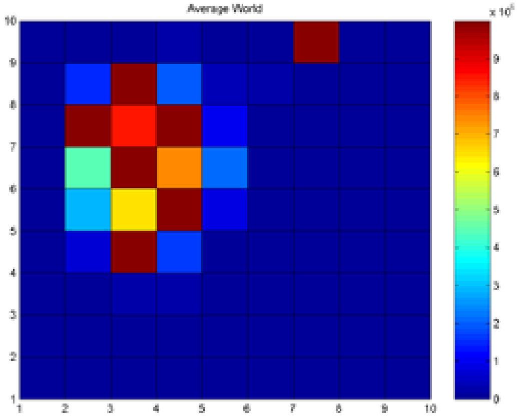


Figure 7. Posterior fire configuration as calculated by our algorithm.

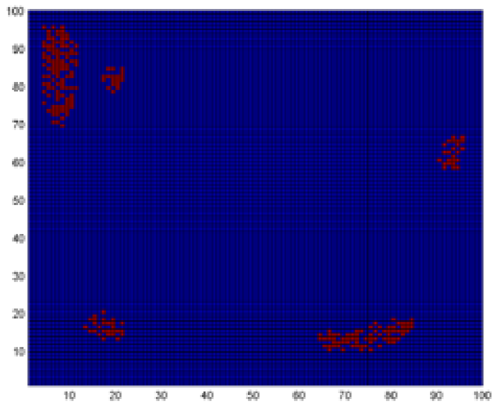


Figure 8. A sample dataset returned from a sensor network. Red represents the cells where fire was detected, and the blue represents the cells where no fire was detected, or no sensor was present.

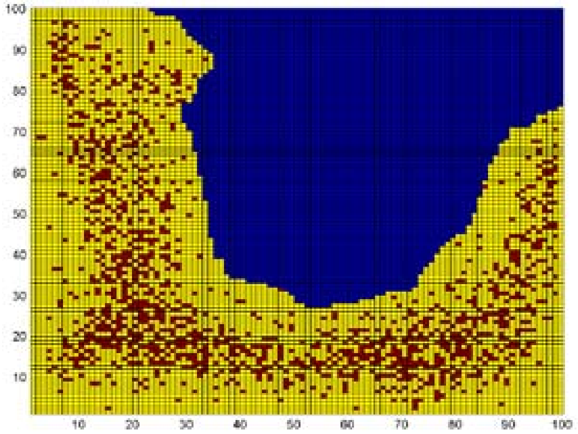


Figure 9: An example vegetation model map. Red represents forests, yellow represents grassland, blue represents water. Collected environmental information, fire, leads to the iterations in Figure 10.

greater likelihood of fire spreading with a greater density of vegetation.

Figure 10 displays the progression of the value of iteration as the fire growth is monitored. It was initiated from measurement data collected from environment in Figure 9. Comparison between Figure 8 and Figure 10 indicates that the first iteration facilitates the growth that exists in the cells that are unobserved. As each iteration progresses, a distinct pattern is observed as the effects of the vegetation model begin to take effect. By the 10th iteration, the certain oblong shape of the fire in the upper left corner is demonstrative of the effects of the differing landscapes.

Whether it is to determine which areas are most critically at risk, or which areas should be targeted for preventive measures, the ability to probabilistically generate a growth model is integral to any form of strategic planning. The key lies in the fact these procedures are to be carried out with the raw data directly provided by the sensors.

Future Work and Discussion

There are three major ways in which this research can be built upon in the future. First, it would be useful to implement this algorithm with actual, physical sensors. Second, in conjunction with the first, is the expansion of the sensor node to be able to factor in other environmental variables such as temperature, humidity, and wind conditions to render a more accurate model. Third, we want the sensor network to control itself in monitoring to make the process as cost-efficient as possible, balancing information gain with power consumption.

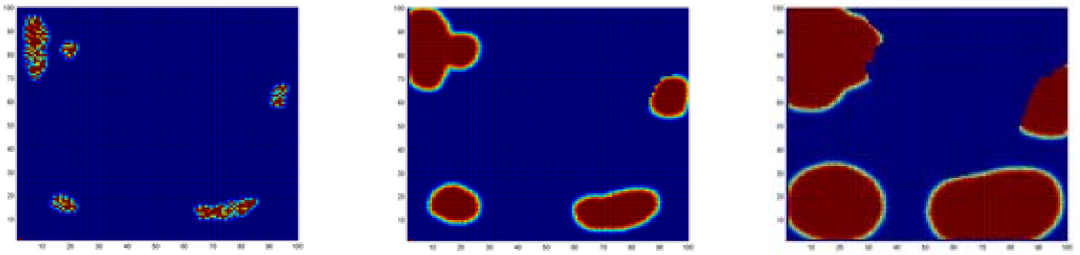
We plan to use this algorithm in an outdoor setting with sensors fitted with motion detectors. The sensor network would be responsible for detecting any movement deviating from the animal population in the area. It becomes altogether critical for sensors to cross-reference and account for unobserved locales as it tracks the movement of animals. As the algorithm is designed to work with any form of sensor data, the provided generality allows for this extension. While our algorithm can effectively make inferences regarding the configuration of the environment, the accuracy of such models can be increased greatly through the implementation of additional sensor modules on the sensor nodes. Once the individual sensors can detect temperature, humidity, and wind conditions, the sensor network as a whole is able to predict both the current and future behavior of the fire with great accuracy.

Finally, although the sensor network provides an efficient framework to extract data from limited information, there is a large advantage that has yet to be fully appreciated through the extended application of this algorithm. As network longevity is a key feature in this application, the frequency of the use of the sensors is critical. Inference algorithm and sensor use

Iteration 1

Iteration 10

Iteration 20



Iteration 30

Iteration 40

Iteration 50

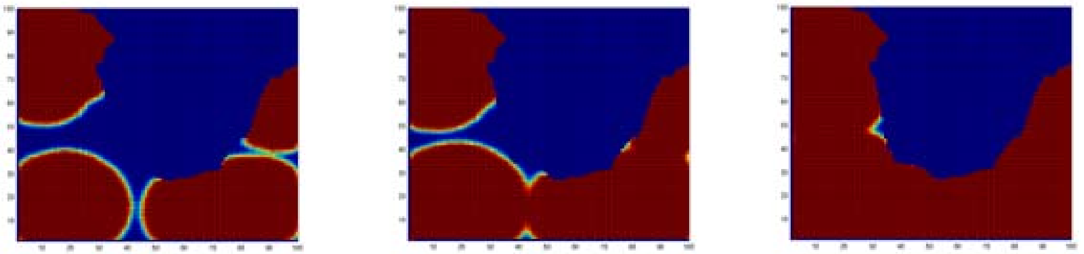


Figure 10: Value iteration applied to sensor measurements taken from Figure 9. Red represents a high probability of the presence of fire at the certain time step, and blue represents low probability of fire.

are inherently linked, as one sensor may seek information from or provide information to another. This will be a major focus of the study using cost-based analysis to effectively manage power consumption for any future algorithms to implement a distributed sensor network.

Conclusions

We have presented a robust inference algorithm that can enhance sensor networks to answer specific queries. Using probabilistic techniques, this approach is able to account for limited and stochastic information and physical issues with sensor malfunction, providing information for the world state - even in locales not directly observed. In the scenarios presented, the results suggest that the algorithm is highly accurate in comparison with traditional methods. The approach is able to provide the probability of a significant fire, allowing the assessment of risk rapidly and efficiently. Furthermore, the posterior configuration of the fire presents a visual representation of a model accounting for probabilistic concerns, and the value iteration prediction model to account for future behavior of the fire. Moreover, in the accuracy studies of diminishing sensor accuracy, our algorithm is able to better retain its than the traditional algorithm. Hopefully, through better fire detection technology, wildfire disasters may be averted, protecting the property and lives of many.

References

[1] National Fire Information Center. "Wildland Fire Statistics."

http://www.nifc.gov/stats/wildlandfirestats.html, August 3 2004. Accessed August 10, 2004.

[2] W. Glks, S. Richardson, and D. Spiegelhalter. Markov Chain Monte Carlo in Practice. Chapman and Hall CRC, 1996.

[3] U. Lerner. Hybrid Bayesian Networks for Reasoning about Complex Systems Ph.D. Thesis, Stanford University, October 2002.

[4] J. Hershberger and S. Suri. "Convex Hulls and Related Problems in Data Streams", Proc. ACM SIGMOD/PODS Workshop on Management and Processing of Data Stream, 2003.

[5] G. Battista, R. Tamassa, and I. Tollis. Constrained Visibility Representation of Graphics Proc. IPL, 1992.

[6] M. Berg, M. Kreveld, M. Kreveld, M. Overmars, and O. Schwarzkopf. Computational Geometry: Algorithms and Applications. Springer-Verlag, 2000.

[7] R. Biswas, S. Thurn, and L. Guibas. "A Probabilistic Approach to Inference with Limited Information in Sensor Networks" IPSN, 2004.

[8] J. Byers and G. Nasser. "Utility-based decision-making in wireless sensor networks" Proc. IEEE MobiHOC Symposium, 2000.

[9] CNN. "Firefighters gaining ground, halting blazes"

http://www.cnn.com/2003/US/West/10/31/california.wildfire, November 1, 2003. Accessed August 10, 2004.

[10] T. Cover and J. Thomas Elements of Information Theory John Wiley and Sons, Inc., 1991.

[11] L. Guibas, and J. Stolfi. "Primitives for the manipulation of general subdivisions and the computation of Voronio diagrams". ACM Trans. Graph. 4, 74-123. 1985.

[12] L. Guibas. "Sensing, Tracking and Reasoning with Relations". IEEE Signal Processing Magazine. Volume: 19 Issue: 2, March 2002.

[13] J. Kahn, R. Katz, and K. Poister. "Mobile networking for smart dust" Proc. ACM/IEEE International Conference on Mobile Computing and Networking, 1999.

[14] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., 1988.

[15] G. Pottie and W. Kaiser. "Wireless integrated network sensors." Proc. ACM Communications, 2000.