Spatio-Temporal Event Detection Using Dynamic Conditional Random Fields

Jie Yin
Information Engineering Laboratory
CSIRO ICT Centre
Australia
jie.yin@csiro.au

Derek Hao Hu and **Qiang Yang**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
{derekhh, qyang@cse.ust.hk}

Abstract

Event detection is a critical task in sensor networks for a variety of real-world applications. Many real-world events often exhibit complex spatio-temporal patterns whereby they manifest themselves via observations over time and space proximities. These spatio-temporal events cannot be handled well by many of the previous approaches. In this paper, we propose a new Spatio-Temporal Event Detection (STED) algorithm in sensor networks based on a dynamic conditional random field (DCRF) model. Our STED method handles the uncertainty of sensor data explicitly and permits neighborhood interactions in both observations and event labels. Experiments on both real data and synthetic data demonstrate that our STED method can provide accurate event detection in near real time even for large-scale sensor networks.

1 Introduction

The advent of wireless sensor networks has fostered growing interest in many real-world applications, such as coal mine surveillance [Xue *et al.*, 2006], disaster monitoring [Yao and Gehrke, 2003], and object tracking [Hellerstein *et al.*, 2003]. In such monitoring applications, automatic event detection is an essential task, which aims at identifying emergent physical phenomena and make real-time decisions about physical environments. When a particular event under our monitor or an abnormal event is detected, the monitoring system sounds an alarm for immediate attention, so that prompt actions can be taken to minimize adverse impact of abnormal events.

Previous approaches to event detection can be classified into three broad categories: (1) threshold-based approaches [Abadi *et al.*, 2005], in which an event is regarded to occur when sensor readings exceed some predefined thresholds; (2) pattern-based approaches [Xue *et al.*, 2006], in which an event is represented as spatio-temporal patterns and event detection is performed using pattern matching techniques; and (3) learning-based approaches [Wang and Yu, 2005; Wang *et al.*, 2008], which model spatio-temporal dependencies of sensor data and make probabilistic inference about events. Among these techniques, learning-based approaches are very promising because spatio-temporal correlations can be explicitly modeled to deal with the inherent uncertainty of sensor data. Therefore, the false alarm rates can be effectively reduced for event detection. In this category, dynamic Bayesian networks (DBNs) and Markov random fields (MRFs) have been widely employed to formulate spatial and temporal constraints in the estimation process.

Despite much progress in this area, most of the existing works on event detection have been restricted to inference in the spatial or the temporal dimensions separately, while the challenge of integrating spatial and temporal constraints has not been addressed. In real-world situations, since sensor nodes are deployed in a physical space (*spatial relationship*) and sensor readings are collected over a period of time (*temporal relationship*), the changes in sensor readings caused by an event usually exhibit strong spatio-temporal correlations. Such spatial and temporal relationships are very critical to perform accurate event detection. Figure 1 gives an example of spatio-temporal events. At time t, an event is identified to occur at two adjacent sensor nodes in the field. As time moves on, the event spreads to affect the neighboring sensor nodes at time $t + 1$. This observation indicates that, a same event is likely to happen at adjacent sensor nodes at different time slices, and one sensor node is likely to be influenced by the same event in consecutive time slices. Because of the uncertainty in sensor data, such co-occurrences often exhibit long-term dependencies at multiple spatial and temporal scales. For example, an event may occur causing a group of geographically close-by (but not immediately next) sensor nodes to be affected for an interval of time slices. Therefore, detecting such complex events must take a global view of both space and time in an integrated way.

In this paper, we propose a new Spatio-Temporal Event Detection (STED) algorithm using a dynamic conditional random field (DCRF) model. The DCRF model extends CRFs by incorporating temporal constraints among contiguous spatial fields [Sutton *et al.*, 2004]. Our STED method handles the uncertainty of sensor data explicitly and permits neighborhood interactions in both observations and event labels. Compared to generative models, including DBNs and MRFs, DCRFs relax the strong independence assumption among observations and capture spatio-temporal dependencies among observations and events in a unified probabilistic framework. In order to achieve near real-time event detection, we also derive an approximate inference method to efficiently estimate

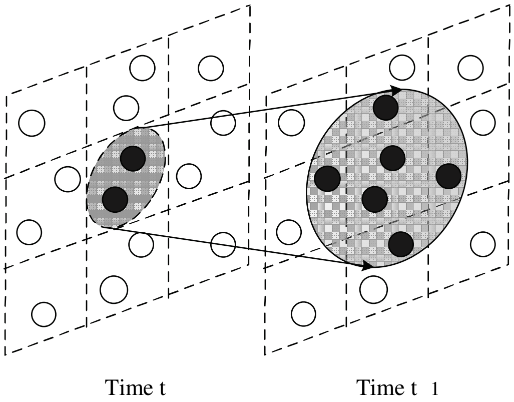


Figure 1: Illustration of Spatio-Temporal Events

the occurrences of events from the history of observed data. We evaluate the performance of our STED method on both real-world data and synthetic data, and show that our STED method can provide accurate event detection in near real time even for large-scale sensor networks.

2 Related Work

In recent years, research on event detection has received much interest in the sensor network area. In general, we classify existing solutions to event detection into three main categories: *threshold-based approaches* [Abadi *et al.*, 2005], *pattern-based approaches* [Solis and Obraczka, 2005; Xue *et al.*, 2006; Li *et al.*, 2007] and *learning-based approaches* [Wang and Yu, 2005; Wang *et al.*, 2008].

For threshold-based approaches, an event is considered to occur when sensor readings exceed a pre-defined threshold value. For example, REED [Abadi *et al.*, 2005] extends the capability of TinyDB for supporting join operations to achieve event detection. Threshold-based approaches are simple to implement. However, threshold values alone are inaccurate and incapable of capturing spatio-temporal characteristics of events, which would incur high false alarm rates for monitoring applications of sensor networks.

Pattern-based approaches, on the other hand, represent events as spatio-temporal patterns in sensor readings and perform event detection using efficient pattern matching techniques. Existing techniques typically employ contour maps [Xue *et al.*, 2006], isolines [Solis and Obraczka, 2005], and gradient maps [Li *et al.*, 2007] to model patterns of events. An event is detected when a user-specified pattern matches recent snapshots of sensor data. The major limitation of these approaches is that they require event patterns to be precisely predefined *a priori* so that exact pattern matching techniques can be applied for event detection.

More recently, learning-based approaches have been proposed to model spatio-temporal dependencies among sensor data and apply probabilistic inference for event detection. In [Wang *et al.*, 2008], dynamic Bayesian networks (DBNs) are applied to detect abnormal events in underground coal mines. Markov random fields (MRFs) are adopted to model spatial relationship at neighboring sensor nodes and perform inference about events [Wang and Yu, 2005]. However, these approaches rely on stringent independence assumptions among observation in order to ensure computational tractability. In contrast, conditional random fields (CRFs) relax this assumption by directly modeling the conditional distribution over hidden states given the observations, which has been demonstrated to be able to capture complex human motions from video sequences [Sminchisescu *et al.*, 2005].

To model complex spatio-temporal events, we would like to design a model that satisfies the following properties: (1) the model should be able to capture long-range dependencies among observations at different spatial and temporal scales; (2) the model should be probabilistic and should be learnable from the given training data; (3) the model should be able to make near real-time inference on event occurrences. In the next section, we will present such a model.

3 Spatio-Temporal Event Detection

In this section, we first review preliminaries on conditional random fields and then provide a detailed description about our proposed approach for spatio-temporal event detection.

3.1 Preliminaries on Conditional Random Fields

Conditional random fields (CRFs) [Lafferty *et al.*, 2001] are undirected graphs that encode a conditional probability distribution using a given set of features. Formally, a conditional random field models the conditional probability of a state sequence $\mathbf{y}$ given the observed sequence $\mathbf{x}$ as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Phi(\mathbf{y}_c, \mathbf{x}_c). \qquad (1)$$

Above (1), $\mathbf{y} = \{y_t\}, \mathbf{x} = \{x_t\}$ for $t = 1, \ldots, T$ and $\mathbf{y}$ is a labeling of the observed sequence $\mathbf{x}$. C is the set of cliques in $\mathcal{G}$, where $\mathcal{G}$ denotes the graph representation of the CRF model. Φ is a potential function defined on the cliques, and $Z(\mathbf{x})$ is the normalizing partition function. The potential functions $\Phi(\mathbf{y}_c, \mathbf{x}_c)$ are usually written in the form of the factorization of a set of feature functions $\{f_k\}$:

$$\Phi(y_c, x_c) = \exp\left(\sum_k \lambda_k f_k(\mathbf{y}_c, \mathbf{x}_c)\right). \qquad (2)$$

Such a formulation is a *linear-chain CRF* which imposes a first-order Markov assumption on the hidden variables. Figure 2 shows its corresponding graphical model.

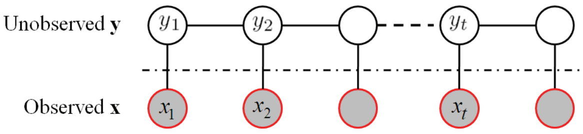


Figure 2: Linear-Chain Conditional Random Field

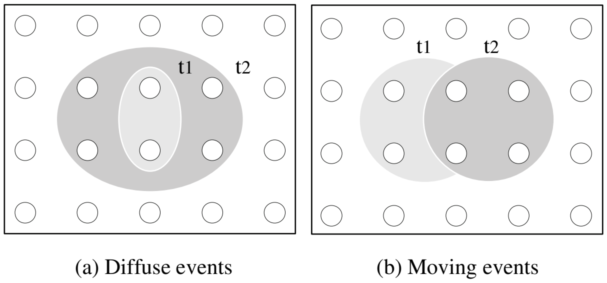
A linear-chain CRF can be used to encode temporal relationship by defining feature functions over the edges of neighboring y's. Therefore, it has been widely applied for labelling and segmenting in natural language processing. However, linear-chain CRFs have difficulty in effectively modeling both spatial and temporal constraints in our problem.

3.2 Our Proposed Approach

This section details our STED (Spatio-Temporal Event Detection) algorithm based on a dynamic conditional random field (DCRF) model.

Spatio-Temporal Events

We first define an event space as a collection of spatio-temporal patterns. These patterns can be the movement of an object, or any pattern that constitutes a finite space of entities. Each entity E can entail a set of observations in sensor readings, $x(E, s, t)$, in a probabilistic manner, where t is the time slice and s is spatial coordinates of sensor nodes. Our objective is to detect whether an event E occurs at specific locations s at time t given a collection of observations $x(E, s, t)$.



(a) Diffuse events (b) Moving events

Figure 3: Two Common Types of Spatio-Temporal Events

Figure 3 illustrates two common types of spatio-temporal events, which we call diffuse events and moving events, respectively. A diffuse event originates from a point source and spread in all directions in the spatial space. A moving event occurs at a region and moves in one direction over time. Examples of real-world spatio-temporal events includes spreading fires in forests, gas leakage in underground mines, and moving warm currents in oceans.

Our DCRF Model

To model such spatio-temporal events, we need a probabilistic model to capture both the spatial relationship of sensor nodes at each time slice and the temporal relationship between neighboring sensor nodes across different time slices. To achieve this, we apply a dynamic conditional random field (DCRF) [Sutton *et al.*, 2004] to model such spatio-temporal correlations in an integrated way.

DCRFs extend linear-chain CRFs by incorporating temporal constraints among successive spatial fields. As with a DBN, a DCRF model can be specified by a template that gives the graphical structure, features, and weights for two time slices. Formally, let C be a set of clique indices and $F = \{f_k(\mathbf{y}_{t,c}, \mathbf{x}, t)\}$ be a set of feature functions and $\Lambda = \{\lambda_k\}$ be a set of real-valued weights. The distribution p is defined as a dynamic conditional random field if and only if:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_t \prod_{c \in C} \exp\left(\sum_k \lambda_k f_k(\mathbf{y}_{t,c}, \mathbf{x}, t) \right). \quad (3)$$

Above (3), $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$ is a sequence of random vectors $\mathbf{y}_i = (y_{i1}, \ldots, y_{im})$, where $\mathbf{y}_i$ is the state vector at time i and y_{ij} is the value of variable j at time i. Interested readers please refer to [Sutton *et al.*, 2004] for detailed specifications of DCRFs.

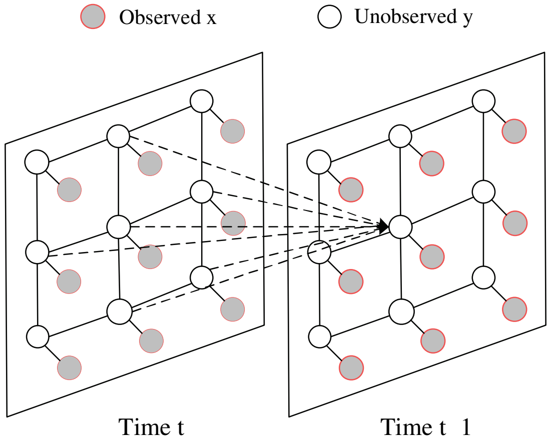


Figure 4: Our Dynamic Conditional Random Field Model

Figure 4 gives an example of the graphical representation of our DCRF model. In the figure, shaded nodes indicate observed measurements $\mathbf{x}$, and non-shaded nodes indicate the hidden labelings $\mathbf{y}$ about events, with the values to be inferred from observations $\mathbf{x}$. In the model structure, solid lines represent spatial relationships between sensor nodes at the same time slice, and dashed lines represent temporal constraints between sensor readings across different time slices. Such relationships can be encoded by defining domain-specific feature functions for the DCRF model. For example, we can define a feature function as $f(y_{i,t}, y_{j,t+1}, x_{i,t}, x_{j,t+1}) = \delta(\| x_{i,t} - x_{j,t+1} \|^2 \leq \epsilon)$, where $\delta(w)$ is an indicator function which equals 1 if the condition w holds and 0 otherwise. This feature function measures the consistency between the hidden nodes $y_{i,t}$ and $y_{j,t+1}$ at two adjacent sensor nodes s_i and s_j. In other words, if an event is detected at sensor node s_i at time t, it is likely to spread to its neighboring node s_j at the next time if their corresponding sensor readings $x_{i,t}$ and $x_{j,t+1}$ are similar at two consecutive time slices.

Parameter Estimation in DCRFs

The goal of parameter estimation is to determine the weights $\Lambda = \{\lambda_k\}$ of the feature functions in a DCRF given a training data set $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$. The weights are estimated discriminatively by maximizing the conditional log-likelihood of labeled training data:

$$\mathcal{L}(\Lambda) = \sum_i \log_\Lambda p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}). \quad (4)$$

The derivative of the log-likelihood $\mathcal{L}$ with respect to λ_k is:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \lambda_k} &= \sum_i \sum_t f_k(\mathbf{y}_{t,c}^{(i)}, \mathbf{x}^{(i)}, t) \quad (5) \\ &\quad - \sum_i \sum_t \sum_{\mathbf{y}_{t,c}} p_\Lambda(\mathbf{y}_{t,c}|\mathbf{x}^{(i)}) f_k(\mathbf{y}_{t,c}, \mathbf{x}^{(i)}, t). \end{aligned}$$

It is easy to observe that the conditional log-likelihood function is convex with respect to the weights λ_k, and thus, it can be optimized using numerical gradient algorithms. Unfortunately, this optimization runs an inference algorithm at each iteration, which can be intractably inefficient in large and dense networks. We therefore resort to maximizing the *pseudo-likelihood* of the training data, which has been shown as an effective method for fast, approximate parameter estimation in CRFs [Besag, 1975]. The essential idea is to break the model down into a collection of independent nodes by conditioning each node on the values of its direct neighbors (also known as the Markov Blanket of the node). The key advantage of maximizing pseudo-likelihood rather than the likelihood is that its gradient can be computed extremely efficiently, without running an inference algorithm.

Inference in DCRFs

Given an observation sequence $\mathbf{x}$ and the estimated parameter Λ, inference in DCRFs involves solving two inference problems: the first is to compute the marginal probabilities $p(\mathbf{y}_{t,c}|\mathbf{x},\Lambda)$ of the states over all the cliques $\mathbf{y}_{t,c}$, and the second is to compute the optimal labels as:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x};\Lambda). \tag{6}$$

In our problem, since we aim at estimating the states of event occurrences for all the sensor nodes at each time slice, the number of combinations over such a large domain is exponential. This makes exact inference intractable for event detection, especially in large-scale sensor networks. Therefore, we apply loopy belief propagation (LBP), an approximate inference algorithm, to solve the two inference problems.

Belief propagation aims to iteratively update a vector $\mathbf{m} = (m_u(x_v))$ of messages between vertices x_u and x_v. The update from x_u to x_v is given by:

$$m_u(x_v) \leftarrow \sum_{x_u} \Phi(x_u, x_v) \prod_{x_t \neq x_v} m_t(x_u), \tag{7}$$

where $\Phi(x_u, x_v)$ is the potential function on the edge (x_u, x_v). Given a message vector m available, approximate marginals are computed as:

$$p(x_u, x_v) \leftarrow \kappa\Phi(x_u, x_v) \prod_{x_t \neq x_v} m_t(x_u) \prod_{x_w \neq x_u} m_w(x_v), \tag{8}$$

where κ is a normalization constant. We can compute the approximate marginal probability using such a belief propagation scheme. Similarly, we can also use a belief propagation scheme to compute the optimal labels $\mathbf{y}^*$ based on the Viterbi algorithm. The corresponding update is defined as:

$$m_u(x_v) \leftarrow \max_{x_u} \Phi(x_u, x_v) \prod_{x_t \neq x_v} m_t(x_u). \tag{9}$$

In belief propagation, each round of belief propagation is called a *message sending* procedure. Since different schedules for message sending can affect the convergence of inference, we adopt three schedules for belief propagation: (1) a random schedule, which simply sends messages across edges in a fully random order. (2) a tree-based schedule [Wainwright *et al.*, 2001], which propagates messages along a set of cross-cutting spanning trees of the original graph. (3) a Residual Belief Propagation (RBP) schedule, which was recently proposed by [Elidan *et al.*, 2006]. For a graph with V nodes and E edges, a tree-based schedule needs to send $O(V)$ messages at each iteration while a random schedule needs to send $O(E)$ messages. The advantage of RBP is that it is a *dynamic schedule*, in which the message values during inference are used to determine which update to perform next. Such dynamic schedules are shown to converge much faster than previous static schedules like tree-based or random schedules. In RBP, an upper bound on the error of a message can be computed and then used as a priority for scheduling future message updates.

In our experiments, we will analyze how different belief propagation schemes would affect the overall computational complexity of our DCRF model, particularly for event detection in large-scale sensor networks.

The STED Algorithm

We now give a detailed description of our STED (Spatio-Temporal Event Detection) algorithm. The STED algorithm operates in two phases:

- Offline training phase: In the offline phase, we collect a training data set $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}$, where $\mathbf{y}^{(i)}$ represents the label for event occurrences. Based on the training data, we estimate the optimal parameters $\Lambda = \{\lambda_k\}$ for our DCRF model.
- Online detection phase: In the online phase, given a sequence of observations, $\mathbf{x}_{t-k+1}, \ldots, \mathbf{x}_t$, obtained at a time interval k, where $k \leq t$, we can estimate an optimal label sequence $\mathbf{y}^*$ about event occurrences by applying collective inference, as follows:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}_{t-k+1}, \ldots, \mathbf{y}_t|\mathbf{x}_{t-k+1}, \ldots, \mathbf{x}_t; \Lambda),$$

where the optimal label sequence $\mathbf{y}_{t-k+1}, \ldots, \mathbf{y}_t$ are estimated jointly. Accordingly, $\mathbf{y}_t$ is the estimated labels about event occurrences at time t for the sensor nodes.

4 Experimental Evaluation

To evaluate the performance of our proposed algorithm, we performed experiments on both real-world data and synthetic data. All the experiments were run on a Dual 2.13GHz Intel Core2 6400 PC with 2GB RAM.

4.1 Baselines and Metrics

We use three different algorithms as the baselines for comparison. The first baseline is an MRF model that has been used in [Wang and Yu, 2005]. The second one is a DBN model in which Markov chain Monte Carlo (MCMC) is used for structure learning. We also compare the performance of our algorithm with a 2D-CRF model. The structure of this model is similar to what we build in our DCRF model at each time slice. However, spatio-temporal relationships are not modeled across consecutive time slices because the 2D-CRF model implicitly assumes that different time slices are conditionally independent. This baseline is used to demonstrate the capability of our DCRF model in incorporating spatio-temporal constraints effectively. In our experiments, we used exact inference for MRF, DBN and 2D-CRF, and chose the L-BFGS method [Nocedal and Wright, 1999] to optimize the parameters for our DCRF model.

The evaluation metrics used in our experiments include the precision, recall and F1-score measure. Let us define the number of sensor nodes that we predict to have an event as C, the number of sensor nodes that actually have an event in the set we predict as A, and the number of sensor nodes that actually have an event as B. The precision, recall and F1-score are defined as follow:

$$\text{Precision} = \frac{A}{C}, \quad \text{Recall} = \frac{A}{B},$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

4.2 Experiments on Real Data

Experiments were first carried out on real data collected from a sensor network. We deployed 30 off-the-shelf motes (TMote Sky) with the light sensors (Hamamatsu S1087 PAR) to measure the light strength in our office area. All the motes were programmed to collect two samples of light strength every minute. The samples were packaged to a sink node at 2405MHz radio frequency and then sent to a computer via a USB serial port. We chose light as the sensing modality for our experiments because it is relatively easy to control the light intensity in an indoor setting, and introduce spatio-temporal events by covering light sensors with paper cups.

For the task of event detection, we stimulated a diffuse event over time by covering a group of sensor nodes with colorful paper cups. Figure 5 shows our experimental setup for data collection. As shown in the figure, the simulated event originates from an inside region at time t_1, and it then spreads outwards at time t_2 and t_3, respectively. Note that, after the event spreads to the outside, those sensor nodes that are affected by an event at a previous time became unaffected. In the end, we obtained a complex spatio-temporal pattern that is difficult to be handled by previous methods, where two sensor nodes that are not physically close by either in time or space may actually belong to the same event. Our proposed approach can detect these events successfully.

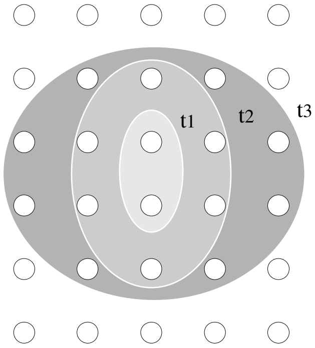


Figure 5: Experimental Setup for Data Collection

We first performed experiments to compare the accuracy of our STED algorithm against the three baselines (MRF, DBN and 2D-CRF). The results based on three cross-validation are summarized in Table 1. We can see that, our STED algorithm outperforms the three baseline methods in all three performance measures. This indicates that, by utilizing a DCRF model, our STED algorithm can effectively model spatial and temporal relationship among observation and events, which makes online event detection more accurately.

Algorithms	Precision	Recall	F1-Score
STED (DCRF)	**88.2%**	**93.8%**	**87.6%**
MRF	72.2%	81.2%	76.4%
DBN	47.4%	56.3%	51.5%
2D-CRF	70.0%	87.5%	77.8%

Table 1: Performance Comparison with Baselines

We also compared the inference efficiency of our STED algorithm using three different BP schedules. Our experiments show that the inference time per time slice of our STED algorithm is around 0.15 second for all the BP schedules. We can conclude that, for this small-scale data set, different BP schedules do not significantly influence the inference time of our STED algorithm.

So far we have demonstrated our STED algorithm can perform accurate event detection in real time based on this real data set. However, one important requirement still remains to be verified, that is, whether our algorithm would be able to provide near real-time event detection in large-scale sensor networks. Therefore, in the following, we investigate the scalability of our algorithm on a large-scale synthetic data set.

4.3 Experiments on Synthetic Data

To test the scalability of our STED algorithm, experiments were also performed on a large-scale synthetic data set. We simulated a sensor network at a square field of 100 × 100 meters. The number of sensor nodes was set at 500, and the locations of sensor nodes were randomly generated in the field. Similar to [Dogandzic and Zhang, 2006], we simulated two moving events in the sensor network using the Gaussian measurement-error model. For sensor nodes with the presence of events, the Gaussian means were set as $\mu_1 = 2$ and $\mu_2 = 8$, respectively, and the noise variance was set as $\sigma^2 = 0.5$. For the rest of sensor nodes, the Gaussian mean was set as $\mu = 5$ and the noise variance was set as $\sigma^2 = 0.5$. Based on this setting, we generated another independent data set for testing. Both the training data set and the testing data set consist of 500 time series at the length of 1000.

Similarly, we would like to first compare the accuracy of our STED algorithm against other baseline methods. However, note here that the scale of this data set is large, so it is very computationally expensive to train a DBN and a 2D-CRF. Therefore, we only compared our STED algorithm against MRF. We report that our STED algorithm achieves an F1-score of 92.30% while the F1-score of MRF is 90.52%.

Table 2 shows the inference time per time slice for our STED algorithm with different BP schedules. We can see that, the random schedule is far less efficient than the tree-based schedule and RBP because $O(E)$ message updates need to be performed at each time. In contrast, the inference times per time slice are 8.9 and 2.8 seconds for the tree-based schedule and RBP, respectively. This indicates that, the tree-based schedule and RBP can provide more efficient inference than the random schedule when applied to a larger data set. Noteworthily, the inference time for RBP is only 2.8 seconds per time slice, making it extremely useful for near real-time event detection in large-scale sensor networks.

Algorithms	Inference Time Per Time Slice (seconds)
Random	37.2s
Tree-based	8.9s
RBP	2.8s

Table 2: Inference Time with Different BP Schedules

5 Conclusions and Future Work

In this paper, we proposed a new probabilistic approach for detecting spatio-temporal events in sensor networks. Our STED algorithm utilizes a DCRF model to effectively capture spatio-temporal dependencies among observations and events in a unified framework. We validated the effectiveness and efficiency of our algorithm through experiments on real and synthetic data. Experimental results show that our STED algorithm can significantly improve the accuracy of previous approaches and provide near real-time event detection.

We plan to extend our work in several directions: one future study is to look into the feasibility of using an online CRF model that could better satisfy the requirement of real-time event detection in sensor networks. Another possible direction is to explore how we can automatically learn feature functions from a collection of time series data for multivariate spatio-temporal event detection.

References

[Abadi *et al.*, 2005] D. Abadi, S. Madden, and W. Lindner. REED: Robust, efficient filtering and event detection in sensor networks. In *Proceedings of the 31st International Conference in Very Large Database (VLDB)*, pages 769–780, Trondheim, Norway, August 2005.

[Besag, 1975] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24, 1975.

[Dogandzic and Zhang, 2006] A. Dogandzic and B. Zhang. Distributed estimation and detection for sensor networks using hidden Markov random field models. *IEEE Transactions on Signal Processing*, 54(8):3200–3215, 2006.

[Elidan *et al.*, 2006] G. Elidan, I. McGraw, and D. Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI)*, Cambridge, MA, USA, July 2006.

[Hellerstein *et al.*, 2003] Joseph M. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Towards sophisticated sensing with queries. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN)*, pages 63–79, Palo Alto, CA, USA, April 2003.

[Lafferty *et al.*, 2001] J.D. Lafferty, A. McCallum, and F.C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289, Williamstown, MA, USA, June–July 2001.

[Li *et al.*, 2007] M. Li, Y. Liu, and L. Chen. Non-threshold based event detection for 3d environment monitoring in sensor networks. In *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS)*, pages 9–16, Toronto, Ontario, Canada, June 2007.

[Nocedal and Wright, 1999] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[Sminchisescu *et al.*, 2005] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV)*, pages 1808–1815, Beijing, China, October 2005.

[Solis and Obraczka, 2005] I. Solis and K. Obraczka. Efficient continuous mapping in sensor networks using isolines. In *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, pages 325–332, San Diego, CA, USA, July 2005.

[Sutton *et al.*, 2004] C. A. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, Banff, Alberta, Canada, July 2004.

[Wainwright *et al.*, 2001] M. Wainwright, T. Jaakkola, and Alan S. Willsky. Tree-based reparameterization for approximate estimation on loopy graphs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1001–1008, Vancouver, BC, Canada, December 2001.

[Wang and Yu, 2005] T.-Y. Wang and C.-T. Yu. Collaborative event region detection in wireless sensor networks using markov random fields. In *Proceedings of the 2nd International Symposium on Wireless Communication Systems (ISWCS)*, pages 493–497, Siena, Italy, September 2005.

[Wang *et al.*, 2008] X.R. Wang, J.T. Lizier, O. Obst, M. Prokopenko, and P. Wang. Spatiotemporal anomaly detection in gas monitoring sensor networks. In *Proceedings of the European Conference on Sensor Networks (EWSN)*, pages 90–105, Cork, Ireland, January–Feburary 2008.

[Xue *et al.*, 2006] W. Xue, Q. Luo, L. Chen, and Y. Liu. Contour map matching for event detection in sensor networks. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 145–156, Chicago, IL, USA, June 2006.

[Yao and Gehrke, 2003] Y. Yao and J. Gehrke. Query processing in sensor networks. In *Proceedings of the First Conference on Innovative Data System Research (CIDR)*, Asilomar, CA, USA, January 2003.