Procedia Computer Science
Volume 29, 2014, Pages 1324–1333
ICCS 2014. 14th International Conference on Computational Science

Context-aware Dynamic Data-driven Pattern Classification*

Shashi Phoha[1], Nurali Virani[2], Pritthi Chattopadhyay[2], Soumalya Sarkar[2], Brian Smith[1], and Asok Ray[2]

[1] Applied Research Laboratory, The Pennsylvania State University
University Park, PA 16802, USA.
sxp26@arl.psu.edu, bms24@arl.psu.edu
[2] Department of Mechanical and Nuclear Engineering, The Pennsylvania State University
University Park, PA 16802, USA
nnv105@psu.edu, pxc271@psu.edu, svs5464@psu.edu, axr2@psu.edu

Abstract

This work aims to mathematically formalize the notion of context, with the purpose of allowing contextual decision-making in order to improve performance in dynamic data driven classification systems. We present definitions for both intrinsic context, i.e. factors which directly affect sensor measurements for a given event, as well as extrinsic context, i.e. factors which do not affect the sensor measurements directly, but do affect the interpretation of collected data. Supervised and unsupervised modeling techniques to derive context and context labels from sensor data are formulated. Here, supervised modeling incorporates the a priori known factors affecting the sensing modalities, while unsupervised modeling autonomously discovers the structure of those factors in sensor data. Context-aware event classification algorithms are developed by adapting the classification boundaries, dependent on the current operational context. Improvements in context-aware classification have been quantified and validated in an unattended sensor-fence application for US Border Monitoring. Field data, collected with seismic sensors on different ground types, are analyzed in order to classify two types of walking across the border, namely, normal and stealthy. The classification is shown to be strongly dependent on the context (specifically, soil type: gravel or moist soil).

Keywords: Context-aware sensing; US Border Control; sensor-fence; DDDAS; machine learning

1 Introduction

It is widely recognized that the interpretation of sensor data and the associated performance of any application which uses that data depend strongly on the operational context in which the

*The work reported in this paper has been supported in part by U.S. Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-12-1-0270 and by U.S. Army Research Office (ARO) under Grant No. W911NF-13-1-0461. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2014

doi:10.1016/j.procs.2014.05.119

data are generated. Context-aware sensor operations remain a weak link in military applications, despite significant relevant research over the past decade in diverse fields (e.g., physics-based environmental modeling, machine learning, image processing, natural language processing, ubiquitous computing, human-machine interaction, and cognitive neuroscience). In order to implement context-awareness for automated systems, it is first necessary to precisely define the concept of "context". One possible definition, provided by Dey *et al.* [7], is as follows: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." Their work focused specifically on the implementation of context-aware features for design applications in hand-held devices, which could be effectively customized to the user's current situation by using implicitly sensed context. Blasch *et al.* [3] presents an extensive survey of contextual tracking for information fusion. Shi *et al.* [18] performed vehicle detection from wide area motion imagery by extracting contextual information about roads from vehicle trajectories and fed back this information to reduce false alarms. Context due to the target itself may refer to pose change, deformation, etc. and due to external factors to changes in illumination, viewpoint, occlusions, etc. Oliva *et al.* [14] did an extensive review of work done on effects of context on object recognition by humans, methods of learning of contextual information by the human brain and the mechanism of contextual analysis by humans. Elnahrawy [8] has defined context in terms of values of neighboring sensing nodes in sensor networks and the past history of measurements of the sensor, which can be used to predict the next sensor reading. The notion of context in the DDDAS framework presented here differs from those mentioned above in two specific ways:

1. The context is required to be machine-understandable in order to allow machines to autonomously extract it from sensor data and then use it to improve decisions and to adjust the sensing mechanisms.
2. Two different types of contexts are identified based on their influence on the sensor data. While intuitive descriptions are presented here, mathematical definitions of intrinsic and extrinsic context are given later in this paper.

   - *Intrinsic Context*: Factors that directly influence the sensor measurements for a particular event are called the intrinsic context. For e.g., changes in soil stiffness due to precipitation affect seismic sensor response for a border crossing event [12]; changes in lighting conditions affect camera pictures of a person for identity verification.
   - *Extrinsic Context*: Factors that do not affect the sensor measurements for any particular event, but influence the interpretation of sensor data are called the extrinsic context. For e.g., intelligence information about a possible smuggling-related border crossing; changes in commander's input (e.g., switch to alert from normal mode).

The theories of context-aware sensing are succinctly presented along with an application of the proposed techniques for detection and classification relevant to border security needs. Data from unattended ground sensors, specifically seismic sensors, were collected for events which consist of multiple subjects walking. The subjects perform both normal walking and an alternate gait, which we term *stealthy walking*. We see that the data obtained from the ground sensors are sensitive to the type of soil in which they are located. Data sets were collected for both moist soil and gravel soil, which define the operational contexts. We present and validate both supervised and unsupervised techniques to learn the context and the contextually-aware decision boundaries. Both methods improve classification performance on the footstep seismic data, and the unsupervised methods are able to extract the context (the soil type) without prior knowledge of what these contexts might have been. Results are compared with context-ignorant classification methods. Future research directions are also presented.

2 Context Definition and Modeling

In the this section, the notion of context is mathematically formalized. We also describe two data-driven techniques to model context from labeled and unlabeled sensor data, respectively, with the aim of enabling machines to use context for improved classification performance.

2.1 Mathematical Definition of Context

Let S be a nonempty finite set of sensing modalities and X be a random variable, which takes values in a finite hypothesis set H of random events. Let $s \in S$ be a sensing modality, that, is used to observe a random event $x \in H$ and let $Y(s)$ be a random vector associated with the measurement of sensor s, which is used to classify the event of interest x.

Definition 2.1. (Context Elements) Let $\mathcal{L}(s)$ be a non-empty finite set of labels, and each element of $\mathcal{L}(s)$ is called a *context element*. Every context element is a physical phenomena (natural or man-made), which is relevant to the sensing modality used to observe an event of interest. It is assumed, that the elements of the set $\mathcal{L}(s)$ have been listed in such a way that no two elements can occur simultaneously.

The assumption in construction of $\mathcal{L}(s)$ is not restrictive. If it is possible for few context elements (say, l, m and n) to occur together, then, a new context element(say k) representing l, m, and n occurring together is added to $\mathcal{L}(s)$ and the extension of $\mathcal{L}(s)$ is obtained. The further computation would be done using this extension. Let $P(Y|X, l)$ be the contextual observation density of sensor measurements of modality $s \in S$ for event X, under a given context $l \in \mathcal{L}(s)$. These observation densities models the likelihood of a particular measurement for an event with context element l. Classification of events would be performed using some features extracted from the measurements, so it is convenient and practical to construct observation densities with low dimensional features, instead of the actual measurements.

Definition 2.2. (Extrinsic and Intrinsic Subsets of Contexts) A nonempty set $\tilde{\mathcal{C}} \subseteq \mathcal{L}(s)$ is called extrinsic relative to an event X and the associated measurement Y, if

$$\forall\, l, \tilde{l} \in \tilde{\mathcal{C}}, \quad P(Y|X, l) = P(Y|X, \tilde{l}).$$

A nonempty set $\tilde{\mathcal{C}} \subseteq \mathcal{L}(s)$ is called intrinsic relative to an event X and the associated measurement Y, if

$$\exists\, l, \tilde{l} \in \tilde{\mathcal{C}} \ \text{ such that } \ P(Y|X, \tilde{l}) \neq P(Y|X, l).$$

The context elements in an intrinsic subset may probabilistically affect the measurement Y associated with an event X, whereas an extrinsic context does not affect the conditional distribution $P(Y|X)$. This paper focuses on modeling of intrinsic context as it can be autonomously extracted by the sensors using the data-driven approaches presented in this paper. In the remaining sections, we will use the word "context" to refer in general to "intrinsic context".

2.2 Context Learning

The aim of modeling context has been addressed by the research community using physics-based modeling [21] as well as data-driven modeling, to understand the environment and its impact on sensor data. In this work, two different data-driven techniques to obtain the labels for context elements and intrinsic context subsets are proposed. Context construction can be addressed through either of the following:

1. **Supervised Modeling**: In some application areas, all relevant factors which significantly affect the data are known and these factors can be individually applied to the system (probably in a controlled environment). It is desired to use this knowledge in training and improve system performance. This technique of modeling context with knowledge of the relevant factors known *a priori* is a form of supervised modeling.
2. **Unsupervised Modeling**: In most application areas, the factors affecting the data are either not known *a priori* or cannot be individually applied in a controlled environment. Such systems need to learn the relevant factors, during in-situ training by using unsupervised learning algorithms. This technique of modeling context is a form of unsupervised modeling.

Both these approaches are used to obtain a set of labels for distinguishable factors affecting the sensor data. The set of labels obtained through these data-driven methods (i.e. an approximation to the list of intrinsic context subsets) will be termed the context alphabet.

2.3 Supervised Modeling of Context

Herein, we introduce a supervised modeling technique for context alphabet construction. Following Definition 2.1, observation densities $P(Y(s)|x,l)\ \forall x \in H,\ \forall\, l \in \mathcal{L}(s)$, can be constructed, if several measurements Y are collected for an event $x \in H$ under the same context element $l \in \mathcal{L}(s)$. If the observation densities are overlapping and very close, then, those context elements would have nearly the same effect on the sensor data. Sets of context elements which are approximately indistinguishable can be constructed for a given threshold parameter $\varepsilon > 0$ and a metric $d(\cdot,\cdot)$ on the space of observation densities using the following definition.

Definition 2.3. (Context Alphabet and Context Symbol) Let $C(s,x)$ be a set cover of $\mathcal{L}(s)$ for each modality s and event x. Then, $C(s,x)$ is called a *context alphabet* and a (non-empty) set $c(s,x) \in C(s,x)$ is called *context symbol* provided that the following properties hold:

1. $c(s,x) = \big\{l, m \in \mathcal{L}(s) :\ d(P(Y|x,l), P(Y|x,m)) < \varepsilon\big\}$. Then, the observation density $P(Y|x,l)\ \forall\, l \in c$ is denoted as $P(Y|x,c)$.
2. The set $c(s,x)$ is maximal, i.e., it cannot be augmented by including another element $l \in \mathcal{L}(s)$.

The construction of a context alphabet from the training data set can be reduced to the standard maximal clique listing problem in graph theory. A maximal clique is a complete subgraph, that, cannot be extended by including one more adjacent vertex. In the maximal clique listing problem, the input is an undirected graph, and the output is a list of all its maximal cliques. At first, a complete weighted graph G with context elements as the vertices and the distance between any two contextual observation densities as the weight of the respective edge is constructed using all context elements in $\mathcal{L}(s)$ and a suitable metric $d(\cdot,\cdot)$. Then, all edges whose weight is larger than the threshold parameter ε are deleted to obtain undirected graph

G'. Finally, the maximal clique listing problem for the undirected graph G' is solved using algorithm in Remark 2.1. If a symbol is assigned to each maximal clique, the set of all symbols associated with the maximal cliques is the Context Alphabet, because a maximal clique in G' represents all possible context elements whose observation densities are mutually close together or the observations of the chosen modality are mutually indistinguishable. This procedure offers to select the threshold parameter ε, which can be used to trade-off between robustness and modeling accuracy/performance. If the threshold parameter ε is too large, there would be just one context, the complete graph would be the only maximal clique. On the other hand, if ε is too small, during thresholding all the edges will be deleted, and all singleton contexts would be obtained, i.e., the isolated vertices in the graph. The exact mechanism to choose ε is application specific and needs to be investigated.

Remark 2.1. (Algorithms for Listing all Maximal Cliques) Enumeration of maximal cliques in an undirected graph has been widely researched topic in the domain of graph theory and combinatorics [4, 2, 6]. The popular Bron-Kerbosch algorithm [4] is efficient in the worst-case sense with a running time of $O(3^{n/3})$. The problem is NP Complete, so any known exact algorithms have exponential time complexity. The application targeted in this paper will usually be reduced to a small graph (< 100 vertices), so time complexity is not a major concern. This work uses a variant of original Bron-Kerbosch algorithm as mentioned in [19].

2.4 Unsupervised Modeling of Context

In general, the context under which the data is collected is unknown *a priori* and in many cases, multiple factors affect the data together. Hence, some context information needs to be extracted autonomously from the sensor data. For a given event, the feature vectors extracted from repeated experiments, under the same context are expected to be close to each other in the feature space. With this intuition, clustering techniques can be used to identify the different clusters and thus, symbolize the feature space. These symbols together form a machine-understandable context alphabet.

In this work, a graph theoretic approach was used to identify clusters and create the context alphabet. A graph was constructed with a feature vector extracted from each time series as a node and the weight on the edge connecting any two nodes was proportional to the similarity between the two feature vectors. The similarity score was considered to be the inverse of the Euclidean norm between the feature vectors. Clustering using these graphs is same as community detection, where a community in a graph is a cluster of nodes with more intra-cluster edges than inter-cluster edges. The majority of methods developed until now for community detection in graphs can be reviewed in [9]. Community detection in graphs aims to identify the modules and possibly, their hierarchical organization in graphs, by only using the information encoded in the graph topology. Here, modularity [10] was used as the quality measure for clustering which had to be maximized. Modularity compares the actual density of edges in a subgraph to the edge density one would expect to have in the subgraph, if the edges were added randomly. It can be written as follows:

$$Modularity = \frac{1}{2m} \sum_{vw} [A_{vw} - \frac{k_v k_w}{2m}] \delta(c_v c_w) \tag{1}$$

Where, k_v is the degree of node v, c_v is the community to which node v belongs,and m is the total number of edges. A_{vw} is 1 if there is an edge between vertices v and w. In case of a weighted graph, A_{vw} is the weight of $v - w$ edge. The second term is the probability of there being a edge between vertices v and w, if the edges were added at random, preserving the degree of each vertex. For good community structure, the term in the square brackets should be positive and larger when both vertices are in same community.

In this work, the *fast community detection algorithm* [13] was used. Unlike most other clustering methods, this approach has the advantage of providing both an optimal number of communities and the members of each community. It is an agglomerative hierarchical clustering method, which starts with one vertex in each community. At each stage, the two communities which result in the greatest increase of modularity with respect to the previous configuration are merged. The largest value of modularity in this subset of partitions corresponds to the optimal partitioning of the dataset. Hence, this procedure gives us the desired context alphabet by exploiting the structure and distribution of features in the feature space.

2.5 Contextual Classification

Context and events both affect the sensor measurements. If the effect of context on the sensors measurements for the same event is much smaller than the effect of different events under the same context, then, practically there might not be any classification performance improvement by using the knowledge of context. There are many realistic problems in which this is not the case, so the effect of different contexts is either comparable or larger than that of the different events. In such cases, it is beneficial to extract the knowledge of context and use it to improve event classification performance. The technique developed (to extract context from sensor data) and used in this work involves, first classifying the features extracted from time series to identify context and then, using the classifier which improves classification performance in the known context. Not only the decision boundary/classifier, but also, the feature extraction scheme can be adapted according to the identified context. Traditionally, a classifier is defined as a function which maps a feature space into a finite set. In [20], the term context-based classifier was introduced as follows:

Definition 2.4. (Context-based Classifier) Let $\mathcal{C}$ be the set of contexts, Ψ be the set of features, and X be the (finite) set of classes. Then, a function $A : \mathcal{C} \times \Psi \to X$ is said to be a context-based classifier, if $\forall\ c \in \mathcal{C},\ \forall\ P \in \Psi,\ A(c, P) \in X$. In other words, $A(c, \cdot)$ is a classifier $\forall\ c \in \mathcal{C}$.

During the training of a context-based classifier, once the feature space is partitioned, every feature is assigned the context symbol associated with their partition and a classifier is trained using all features which have the same context symbol. These classifiers take input of feature and context and give output as the event class of the feature under the given context. The context-based classifiers essentially implements a dynamic classifier selection framework [17] in a systematic way.

3 Experiments and Results

This section describes the experiment and results to validate the proposed data-driven context extraction and contextual classification technique with the data collected in field experiments.

3.1 Scenario and Data Collection

A series of experiments were designed to validate the proposed technique for context-aware event classification. Three-axis geophones were deployed to identify two different types of walking: (i) normal walking and (ii) stealthy walking. The seismic response from geophones, used in the analysis, were collected on two different types of test fields: namely a gravel road and a moist soil road. As the characteristics of the seismic response vary significantly with the change of soil properties [12], the two ground types are considered to be two different physical contexts. Eighty experiments (40 for normal walking and 40 for stealthy walking), constituting of two different human subjects, were performed for each context. The seismic sensors (geophones) were buried approximately 15 cm deep underneath the soil surface. Human subjects passed by the sensor sites at a distance of approximately 2 m. The geophone signal was acquired at a sampling frequency of 4 kHz for 10 seconds for each experiment. The main task of the context-aware event classifier is to discriminate between normal and stealthy walking over different soil types with high accuracy.

3.2 Data preprocessing and Feature extraction

The feature extraction method used in this work is built upon the concept of symbolic dynamic filtering (SDF) [16], where (finite-length) time series data are partitioned to produce symbol strings that, in turn, generate a special class of probabilistic finite state automata (PFSA). As described in [16], there are two pertinent steps in SDF feature extraction: (i) Symbolization (also known as quantization) of the time series based on a selected alphabet Σ of symbols $\{\sigma | \sigma \in \Sigma\}$, and (ii) Construction of probabilistic finite state automata (PFSA).

For data symbolization, the data sets are partitioned by maximum entropy partitioning (MEP) [15]. MEP maximizes the entropy of the generated symbols and therefore, each generated symbol is chosen so as to appear approximately equally often in the training set.

The PFSA we construct, called D-Markov machines, have a deterministic algebraic structure and their states are represented by symbol blocks of length D or less. The generalized D-Markov machines [1] are constructed in two steps: (i) State splitting : The states are split based on their information contents, and (ii) State merging : Two or more states (of possibly different lengths) are merged together to form a new state without any significant loss of the embedded information. After the D-Markov machine is constructed, the stationary state probability vector is computed and used as the low-dimensional feature representing the associated time series for future classification purpose.

In the signal preprocessing step of analysis of our experimental data, the DC component of a seismic signal was first eliminated, resulting in a zero mean signal. Then, the signal was partitioned using the maximum entropy partitioning approach with a symbol size of 7. The maximum number of allowable states of the D-Markov machine was varied, and the classification performance on a validation test set was found in each case. This process was repeated 3 times to obtain average error. The number of states was then chosen to be 10, as it resulted in the best performance on the validation test set (Fig. 4). The features thus obtained from a time-series data set, i.e. the stationary state probability vector of the D-Markov machine thus represented, were determined and used for classification.

3.3 Context Alphabet Construction

The entire data set was randomly divided into training (60%) and test (40%) sets ten times and the entire analysis was repeated for each combination of training and testing sets. Graph-theoretic techniques were then applied on the training data to get the context alphabet. The results were as follows:

1. ***Supervised Modeling:*** The features with the same event labels and same type of soil are used to model the contextual observation densities. The k-nearest neighbor (k=6)

Figure 1: Gravel Soil

Figure 2: Moist Soil

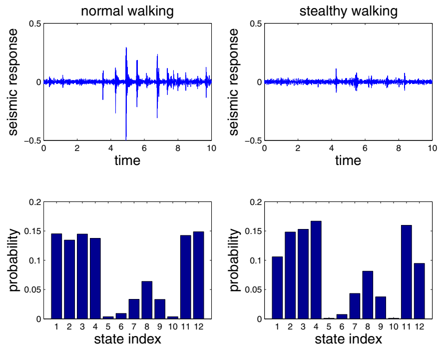

Figure 3: Signals and Features

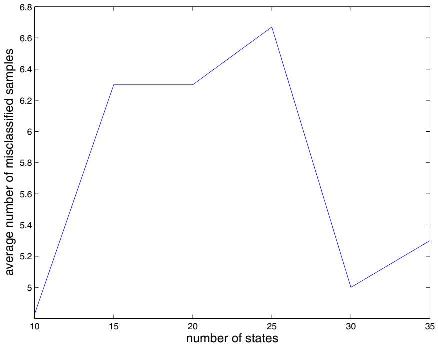

Figure 4: Selection of Number of States

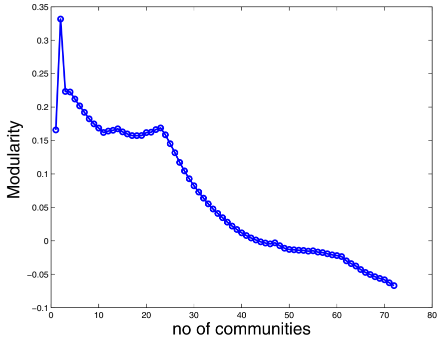

Figure 5: Modularity Score

algorithm was used as a nonparametric density modeling technique. Symmetric KL divergence [11] was used as a metric of distance between the observation densities. Two maximal cliques were obtained one for each soil type; hence, the context alphabet size was 2. Note the human-understandable factors can be associated with machine-understandable context in supervised modeling. If a single maximal clique was obtained, it would imply that the features obtained from each context element are similar and contextual classification would not significantly improve performance.

2. ***Unsupervised Modeling:*** Figure 5 shows a modularity plot on a sample dataset of normal walking. The modularity of the graph configuration is highest when there are two communities. Hence, observing the modularity plot gives the number of communities present, essentially yielding the context alphabet. Since the ground truth context labels were available in the experiment, it was seen that the two communities detected by the algorithm actually corresponded to the two different ground types.

3.4 Classification Performance

The proposed analysis was performed for three different cases and the corresponding results are reported. The classifier used was a linear support vector machine [5]. The following cases have been considered:

1. **No context knowledge:** The training set had only the event labels. The knowledge that the data had been collected under two different physical contexts (Sect. 3.1) was not used in training; hence, just a single classifier was trained. This classifier was then used to classify the event of the test set. The context knowledge was not available, yet the error was found to be $14.53 \pm 3.82\%$ (Table. 1), which shows the robustness of the chosen feature extraction method.

2. **Perfect context knowledge:** The training set was labeled with the event and context symbols. A separate classifier was trained for each context symbol corresponding to a ground type. A k-nearest neighbor classifier was trained and used to assign the context symbol to the test set. The assigned symbol and the feature were used to classify the event of this sample. Minimum classification error was obtained in this case ($8 \pm 2.4\%$ (Table. 1)), as expected.

3. **With context identification:** The training set had only the event labels. Both unsupervised and supervised modeling yielded two context symbols. A k-nearest neighbor classifier was used to assign the context symbol to each test sample. The assigned symbol and the feature were then used to classify the event of this sample. The average performance was 8.1%, almost same as case 2, albeit with a larger standard deviation (2.81%). One reason for this could be that the extracted context might not always coincide with the ground truth, so the resulting classifiers might not be optimal under the true context.

Table 1: Experimental Results: Error Percentage in the three cases

Case	Mean	Standard Deviation
1	14.53	3.82
2	8	2.4
3	8.1	2.81

This experiment shows that, if a sensor has been trained with data collected from several sites, then during operation, it can first learn about the soil type and use that knowledge for better target classification. Also, in this experiment, the benefit of context-aware adaptation of classifiers is clearly visible as the performance improves by choosing the appropriate classifier for the estimated context.

4 Summary, Conclusions, and Future Research

This paper proposes a dynamic data-driven approach to extract context from sensor data for context-aware pattern recognition. A formal definition of machine-understandable context is presented. Contexts are extracted from Generalized D-Markov features, which are derived from sensor data via graph-theoretic methods (maximal clique finding, modularity based clustering) using both *supervised* and *unsupervised* approaches. Finally, contextual decision adaptation is performed via modification of the classifier. The proposed approach was validated in field experiments (seismic responses of human walking) relevant to border control problem. Context-aware event classification shows a significant improvement over non-contextual classification in discriminating different types of walking on different soil types. Intended topics of future research include machine learning of new context states and context evolution modeling for real-time classifier adaptation, as well as dynamic data-driven contextual adaptation of heterogeneous sensor networks for cross-sensory event classification.

References

[1] P. Adenis, K. Mukherjee, and A. Ray. State splitting and state merging in probabilistic finite state automata. *Proc. American Control Conference, San Francisco, CA, USA*, pages 5145–5150, 2011.

[2] E.A. Akkoyunlu. The enumeration of maximal cliques of large graphs. *SIAM Journal on Computing*, 2:1–6, 1973.

[3] E. Blasch, J. Herrero, L. Snidaro, J. Llinas, G. Seetharaman, and K. Palaniappan. Overview of contextual tracking approaches in information fusion. In *Proceedings of SPIE*, 2013.

[4] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communcations of ACM*, 16(9):575–577, 1973.

[5] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[6] F. Cazals and C. Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1):564–568, 2008.

[7] A. Dey and G. Abowd. Towards a better understanding of context and context awareness. In *In HUC'99 :Proceedings of the 1st international symposium on Handheld and Ubiquitous computing*, pages 304–307. Springer-Verlag, 1999.

[8] E. Elnahrawy and B. Nath. Context-aware sensors. In *EWSN*, 2004.

[9] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75 – 174, 2010.

[10] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99:7821–7826, 2002.

[11] S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[12] J.R. McKenna and M.H. McKenna. Effects of local meteorological variability on surface and subsurface seismic-acoustic signals. Technical report, DTIC Document, 2006.

[13] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, September 2003.

[14] A. Olivaa and A. Torralba. The role of context in object recognition. *Trends in cognitive sciences*, pages 520–527, 2007.

[15] V. Rajagopalan and A. Ray. Symbolic time series analysis via wavelet-based partitioning. *Signal Processing*, 86(11):3309–3320, 2006.

[16] A. Ray. Symbolic dynamic analysis of complex systems for anomaly detection. *Signal Processing*, 84(7):1115–1130, July 2004.

[17] D. Ruta and B. Gabrys. An overview of classifier fusion methods. *Computing and Information systems*, 7(1):1–10, 2000.

[18] X. Shi, H. Linga, E. Blasch, and W. Hu. Context driven moving vehicle detection in wide area motion imagery. In *ICPR*, pages 2512–2515, 2012.

[19] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):2842, 2006.

[20] N. Virani, S. Marcks, S. Sarkar, K. Mukherjee, A. Ray, and S. Phoha. Dynamic data driven sensor array fusion for target detection and classification. *Procedia Computer Science*, 18:2046–2055, 2013.

[21] D.K. Wilson, D. Marlin, and S. Mackay. Acoustic/seismic signal propagation and sensor performance modeling. In *SPIE*, volume 6562, 2007.