Relational Temporal Data Mining for Wireless Sensor Networks

Teresa M.A. Basile, Nicola Di Mauro, Stefano Ferilli, and Floriana Esposito

Università degli Studi di Bari, Dipartimento di Informatica, 70125 Bari, Italy
{basile,ndm,ferilli,esposito@di.uniba.it}

Abstract. Wireless sensor networks (WSNs) represent a typical domain where there are complex temporal sequences of events. In this paper we propose a relational framework to model and analyse the data observed by sensor nodes of a wireless sensor network. In particular, we extend a general purpose relational sequence mining algorithm to tackle into account temporal interval-based relations. Real-valued time series are discretized into similar subsequences and described by using a relational language. Preliminary experimental results prove the applicability of the relational learning framework to complex real world temporal data.

Key words: Wireless Sensor Networks, Relational Sequence Mining, Temporal Patterns

1 Introduction

Wireless sensor networks (WSNs) represent a recent technology able to monitor the physical world such as health, micro-climate and habitat, or earthquake and building health [1–4]. A WSN represents a typical domain where there are complex temporal sequences of events, such as computer security and planning. In this paper we aim at applying a relational framework to model and analyse the data observed by nodes involved in a sensor network. The main contribution of this work regards the proposal of a powerful and expressive description language able to represents the spatio-temporal relations appearing in a sensor network, and a general purpose system able to elicit hidden frequent temporal correlations between sensor nodes. In particular the objective of this paper is twofold: **O1**) exploiting a relational language to describe the temporal evolution of a sensor network, and **O2**) using relational learning techniques to discover interesting and more human readable patterns relating spatio-temporal correlations. Furthermore, we can combine temporal and content-based relations into a heterogeneous language providing a general framework applicable to other domains characterized by temporal and content relational features. Indeed, the data generated by sensor nodes involved in a sensor network are type-related (the humidity depends on the temperature), time-related (the temperature may change over time) and spatio-related. All these relations could be easily represented by using a relational language such that proposed in this paper, trying to shift the basic time-series description language to an higher one. Algorithms

proposed for sequential pattern mining generally consider events occurring in a time instant, while in some applications, like in sensor networks, events may occur in a time interval. The paper proposes an efficient algorithm able to mine frequent relational patterns representing arrangements of related intervals.

2 Relational pattern mining

The algorithm we present in this paper is based on the work described in [5] where the authors presented a framework for mining complex patterns, expressed in first-order language, in which events may occur along different dimensions. Specifically, multi-dimensional patterns were defined as a set of atomic first-order formulae in which events are explicitly represented by a variable and the relations between events were represented by a set of dimensional predicates. Here, that framework has been extended in order to take into account interval-based temporal data. We used Datalog [6] as representation language for the domain knowledge and patterns. Sequences and patterns are represented by a set of logical atoms. An atom $p(t_1, \ldots, t_n)$ is a predicate symbol p of arity n applied to n terms t_i (constants or variables).

Definition 1 (Subsumption). *A* substitution θ *is defined as a set of bindings* $\{X_1 \leftarrow a_1, \ldots, X_n \leftarrow a_n\}$ *where* $X_i, 1 \leq i \leq n$ *is a variable and* $a_i, 1 \leq i \leq n$ *is a term. A substitution* θ *is applicable to an expression* e*, obtaining the expression* $e\theta$*, by replacing all variables* X_i *with their corresponding terms* a_i*.*

Definition 2 (Relational sequence). *A relational sequence may be defined as an ordered list of atoms separated by the operator* $<$*:* $l_1 < l_2 < \cdots < l_n$*.*

In order to make the framework more general, the concept of *fluents* has been considered. Assuming a sequence as an ordered succession of events, a fluent is used to indicate that an atom holds for a given event, and hence distinguishing *dimensional* and *non-dimensional* atoms. In order to describe multi-dimensional relational patterns, some dimensional operators for describing general event relationships have been introduced: a) $<_i$, *next step on dimension* i; b) $\lhd_i$, *after some steps on dimension* i; and c) $\bigcirc_i^n$, *exactly after* n *steps on dimension* i*.*

Definition 3 (Subsequence [7]). *Given a sequence* $\sigma = (e_1 e_2 \cdots e_m)$ *of* m *elements, a sequence* $\sigma' = (e'_1 e'_2 \cdots e'_k)$ *of length* k *is a subsequence (or pattern) of the sequence* σ *if*

1. $1 \leq k \leq m$
2. $\forall i, 1 \leq i \leq k, \exists j, 1 \leq j \leq m : e'_i = e_j$
3. $\forall i, j, 1 \leq i < j \leq k, \exists h, l, 1 \leq h < l \leq m : e'_i = e_h$ *and* $e'_j = e_l$*.*

The frequency of a subsequence in a sequence is the number of different mappings from elements of σ' *into the elements of* σ *such that the previous conditions hold.*

Note that this is a general definition of subsequence, in our case the *gaps* represented by the third condition are modelled by the $\lhd_i$ and $\bigcirc_i^n$ operators as reported in the following definition.

Definition 4 (Multi-dimensional relational pattern). *A multi-dimensional relational pattern is a set of atoms, involving k events and regarding n dimensions, in which there are non-dimensional atoms and each event may be related to another event by means of the operators* $<_i$, $\lhd_i$ *and* $\bigcirc_i^n$ *operators,* $1 \leq i \leq n$.

We are interested in mining maximal frequent patterns.

Definition 5 (Maximal pattern). *A pattern* σ' *of a sequence* σ *is maximal if there is no pattern* σ'' *of* σ *more frequent than* σ' *and such that* σ' *is a subsequence of* σ''.

In order to calculate the frequency of a pattern over a sequence it is important to define the concept of sequence subsumption.

Definition 6 (Pattern Subsumption). *Given P a multi-dimensional relational pattern and S a multi-dimensional relational sequence. The pattern P* subsumes *the sequence S, written as* $P \subseteq S$, *iff there exists an* SLD_{OI}*-deduction of P from S.*

An $\mathrm{SLD_{OI}}$-deduction is an SLD-deduction under Object Identity. In the Object Identity framework, within a clause, terms that are denoted with different symbols must be distinct, i.e. they must represent different objects of the domain.

2.1 The algorithm

The algorithm for frequent multi-dimensional relational pattern mining is based on the same idea of the generic level-wise search method, known in data mining from the Apriori algorithm [8]. The generation of the frequent patterns is based on a top-down approach. Specifically, it starts with the most general patterns of length 1 generated by adding to the empty pattern a non-dimensional atom. Then, at each step it *specializes* all the frequent patterns, discarding the non-frequent patterns and storing the ones whose length is lesser than the parameter *maxsize*. Furthermore, for each new refined pattern, semantically equivalent patterns are detected, by using the θ_{OI}-subsumption relation, and discarded.

In the specialization phase, the refinement of patterns is obtained by using a refinement operator ρ that maps each pattern to a set of specializations of the pattern, i.e. $\rho(p) \subset \{p' | p \preceq p'\}$ where $p \preceq p'$ means that p is more general of p' or that p subsumes p'.

The algorithm uses a background knowledge $\mathcal{B}$ (a set of Datalog clauses) containing the sequence and a set of constraints that must be satisfied by the generated patterns. In particular $\mathcal{B}$ contains the following predicates:

- *maxsize(M)*: maximal pattern length (i.e., the maximum number of non-dimensional predicates that may appear in the pattern);
- *minfreq(m)*: this constraint indicates that the frequency of the patterns must be larger than m;
- *dimension(next_i)*: this kind of atom indicates that the sequence contains events on the dimension i. One can have more that one of such atoms, each of which denoting a different dimension. In particular, the number of these atoms represents the number of the dimensions.

Constraints Furthermore the background knowledge contains some constraints that useful to avoid the generation of unwanted patterns. Specifically they are:

- *negconstraint([$p_1, p_2, \ldots, p_n$])*: specifies a constraint that the patterns must not fulfill, i.e. if the clause $\{p_1, p_2, \ldots, p_n\}$ subsumes the pattern then it must be discarded. For instance, negconstraint([p(X,Y),q(Y)]) discards all the patterns subsumed by the clause {p(X,Y),q(Y)};
- *posconstraint([$p_1, p_2, \ldots, p_n$])*: specifies a constraint that the patterns must fulfill. It discards all the patterns that are not subsumed by the clause $\{p_1, p_2, \ldots, p_n\}$;
- *atmostone([$p_1, p_2, \ldots, p_n$])*: this constraint discards all the patterns that make true more than one predicate among $p_1, p_2, \ldots, p_n$. For instance, atmostone([red(X),blue(X),green(X)]) indicates that each constant in the pattern can assume at most one of red, blue or green value.

Hence, the solution space is pruned by using some positive and negative constraints specified by the *negconstraint* and *posconstraint* literals. The last pruning choice is defined by the *atmostone* literals. This last constraint is able to describe that some predicates are of the same type.

Efficiency Issues In order to avoid the generation of patterns containing not linked variables we used the classical types and modes declaration:

- *type(p)*: denotes the type of the predicate's arguments p;
- *mode(p)*: denotes the input output mode of the predicate's arguments p.

In this way we improve the efficiency of the algorithm, since it does not generate patterns containing unrelated atoms. These classical mode and type declarations specify a language bias indicating which predicates can be used in the patterns and to formulate constraints on the binding of variables.

Finally, the background knowledge contains the predicate *key([$p_1, p_2, \ldots, p_n$])* specifying that each pattern must have one of the predicates $p_1, p_2, \ldots p_n$ as a starting literal. Since each pattern a) must start with a non-dimensional predicate, or with a predefined key, and b) its frequency must be less than the sequence length, the frequency of a pattern can be defined as follows.

Definition 7 (Pattern Frequency and Support). *Given a relational pattern $P = (p_1, p_2, \ldots, p_n)$ and S a relational sequence, the* frequency *of the pattern P is equal to the number of different ground literals used in all the possible SLD_{OI}-deductions of P from S that make true the literal p_1. The* support *of P on S is equal to the frequency of the pattern $\{p_1\}$ over the frequency of the pattern P.*

Mining from more than one sequence the support is calculated as the number of covered sequences over the total number of sequences.

In order to improve the efficiency of the algorithm, for each pattern $P = (p_1, p_2, \ldots, p_n)$ the set Θ of the substitutions defined over the variables in p_1 that make true the pattern P are recorded. In this way, the support of a specializations P' of P is computed by first applying a $\theta \in \Theta$ to P'. It is like to remember all the keys of a table that make true a query.

3 Interval-based Temporal Sequences

Here we describe the problem of subdividing real-valued time series into similar subsequences and the extension of the framework to the case of interval-based sequences.

3.1 Abstracting Time Series

The aim is to segment a signal by looking for a sequence of measurements over which a property holds, such as below a given threshold, and to label this segment.

A method to segment a sequence is to iteratively merge two similar segments based on the squared error minimization criteria. Another approach is using clustering, by firstly finding the set of subsequences with length w, by sliding a window of width w, and then clustering the set of all subsequences. A different symbol is associated with each cluster. Other approaches are based on using self-organizing maps.

Here, we concentrate on the abstraction process that translate the initial sequence (with real-valued elements) to a discretized sequence composed of symbols taken from an given alphabet. In particular, our segmentation method is a supervised process that assigns labels to a portion of a time series by using a set of predefined attributes. Future extensions of proposed approach include the use of more powerful techniques to partition time series, like that proposed in [9].

Given a real-valued time series $(t_i, x_i)_{1 \leq i \leq n}$, $x_i \in \mathbb{R}$, the goal is to transform it into a discrete series $(t_i, c_i)_{1 \leq i \leq n}$, $c_i \in \{1, \ldots, C\}$. In the case of a sensor network, made up of n nodes, each node i, located in the environment at the position p_i, sense a set of properties $\mathcal{P}$ at every time instance t. Our approach is to define some abstraction rules useful to shift the basic sensor description language into a more general one. In particular each sensor produces a time series, describing its reading over time, that are then divided into intervals.

Let $\mathcal{C}$ denote the set of possible properties or descriptive labels, such as "temperature is high". Having a time series $(t_i, x_i)_{1 \leq i \leq n}$, denoted by $(t, x)_{1\_n}$, an abstraction rule is a function $\phi_a((t, x)_{1\_n})$ returning a set of m intervals. In particular,

$$\phi_a((t, x)_{1\_n}) = \{\delta_a(l, t_i, t_{i+h}, c_k) | t_j \in \mathcal{D}_k^a, i \leq j \leq i + h \wedge c_k \in \mathcal{C}\}_{1 \leq l \leq m}$$

where $\delta(k, t_i, t_{i+h}, c_k)$ denotes an interval starting from t_i and ending to t_{i+h}, and $\mathcal{D}_k^a$ represents the domain of values for the function ϕ_a associated to the label $c_k \in \mathcal{C}$. For instance for the temperature time series in the wireless sensor network domain we defined the abstraction function
$\phi_t((t, x)_{1\_n}) = \{\delta_t(l, t_i, t_{i+h}, c_k) | t_j \in \mathcal{D}_k^t, c_k \in \mathcal{C}_t\}$ where

$$\mathcal{D}_t^{vl} = \{x | x < 13\}, \quad \mathcal{D}_t^{l} = \{x | 13 \leq x < 22\}, \mathcal{D}_t^{m} = \{x | 22 \leq x < 31\}$$
$$\mathcal{D}_t^{h} = \{x | 31 \leq x < 40\}, \mathcal{D}_t^{vh} = \{x | x \geq 40\}$$

and $\mathcal{C}_t = \{$ *very_low*, *low*, *medium*, *high*, *very_high* $\}$.

Now that we have discretized the time series into intervals, we can extend the definitions to the case of interval-based relational sequences.

Definition 8 (Relational Interval Sequence). *Given a set $\mathcal{T}$ of time series and the sets $\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{T}|}$ of descriptive labels, a* relational interval sequence *is a sequence of relational atoms*

$$\delta_{a_1}(id_1, b_1, e_1, v_1), \delta_{a_2}(id_2, b_2, e_2, v_2), \ldots, \delta_{a_n}(id_n, b_n, e_n, v_n)$$

where $v_j \in \mathcal{C}_i$ is a descriptive label, b_j and e_j represent, respectively, the starting and ending time, $id_j \in \mathbb{N}$ represents the interval identifier, and δ_{a_j} is the corresponding name of the time series $a_j \in \mathcal{T}$. (The interval $\delta(id, b, e, v)$ can be written also by means of three literals as $\delta(id, v)$, begin*(id,b),* end*(id,e)).*

In particular a relational interval sequence can describe several labeled interval sequences into a single one, enabling one to take into account the multivariate analysis in case of different time series. Relations between time intervals are described adopting the Allen's temporal interval logic [10], as reported in Figure 1.

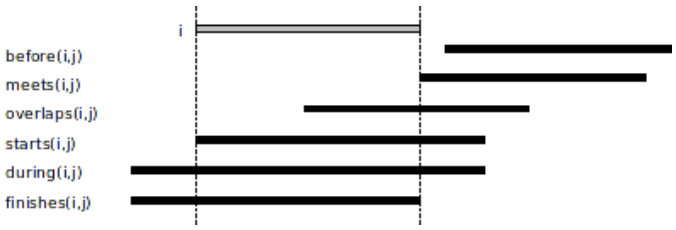

Fig. 1. Allen's temporal intervals [10]

Definition 9 (Relational Interval Pattern). *Given $\mathcal{S}$, the set of interval relation symbols, a* relational temporal pattern *is a set of relational atoms*

$$P = I \cup R = \{\delta_i(id_i, b_i, e_i, v_i)\}_{i=1\ldots n} \cup \{rel_j(id_j^1, id_j^2)\}_{j=1\ldots m}$$

where $rel_j \in \mathcal{S}$, and $\forall rel_j(id_j^1, id_j^2) \in R$ $\exists \delta_h(id_h, b_h, e_h, v_h), \delta_k(id_k, b_k, e_k, v_k) \in I$ such that $id_j^1 = id_h$ and $id_j^2 = id_k$.

4 Experiments

In order to evaluate our approach, we used the data, freely available from [11], collected from a wireless sensor network made up of 54 Mica2Dot sensors deployed in the Intel Berkeley Research Lab and arranged in the laboratory as shown in Figure 2.

A sensor network node is a small autonomous unit, often running on batteries, with hardware to sense environmental characteristics, such as temperature,

humidity and light. Such nodes usually communicates using a wireless network. A sensor network is composed of a large number of sensors deployed in a natural environment. The sensors gather environmental data and transfer the information to the central base station with external power supply. The 54 sensors have been monitored from February 28th to April 5th 2004, and the data, about 2.3 million readings, was collected using the TinyDB in-network query processing system, built on the TinyOS platform. Each sensor collected topology information, along with humidity, temperature, light and voltage values once every 31 seconds.

We selected the measurements (temperature, humidity, light and voltage) from the sensors 41, 42, and 24, for the time period from 2004-03-18 to 2004-03-21 corresponding to 23178 log rows. The aim is to discover some correlations between sensors and/or measurements useful for anomaly detection. For instance, there is a strong correlation between the temperature and humidity, as we can see from the Figure 3 that reports the corresponding graphs for the sensor 41. The first task is to discretize the time series corresponding to each information in order to obtain a interval-based temporal sequence like that reported in Figure 4 where each interval is labeled with a specific name. The discretization step has been executed exploiting the functions ϕ_t, ϕ_h, ϕ_l, and ϕ_v with the corresponding domains $\mathcal{D}_i^j$ where i is the time series name (temperature, humidity, light and voltage) and j is the descriptive label (very low, low, medium, high and very high) associated to the interval:

$$
\begin{array}{llll}
\mathcal{D}_t^{vl} = \{x|x < 13\}, & \mathcal{D}_t^{l} = \{x|13 \leq x < 22\}, & \mathcal{D}_t^{m} = \{x|22 \leq x < 31\}, & \\
\mathcal{D}_t^{h} = \{x|31 \leq x < 40\}, & \mathcal{D}_t^{vh} = \{x|x \geq 40\}, & & \\
\mathcal{D}_h^{vl} = \{x|x < 10\}, & \mathcal{D}_h^{l} = \{x|10 \leq x < 25\}, & \mathcal{D}_h^{m} = \{x|25 \leq x < 40\}, & \\
\mathcal{D}_h^{h} = \{x|40 \leq x < 55\}, & \mathcal{D}_h^{vh} = \{x|x \geq 55\}, & & \\
\mathcal{D}_l^{vl} = \{x|x < 50\}, & \mathcal{D}_l^{l} = \{x|50 \leq x < 200\}, & \mathcal{D}_l^{m} = \{x|200 \leq x < 400\}, & \\
\mathcal{D}_l^{h} = \{x|400 \leq x < 600\}, & \mathcal{D}_l^{vh} = \{x|x \geq 600\}, & & \\
\mathcal{D}_v^{l} = \{x|x < 2\}, & \mathcal{D}_v^{m} = \{x|2 \leq x < 2.4\}, & \mathcal{D}_v^{h} = \{x|2.4 \leq x < 2.75\}, & \\
\mathcal{D}_v^{vh} = \{x|x \geq 2.75\}. & & &
\end{array}
$$

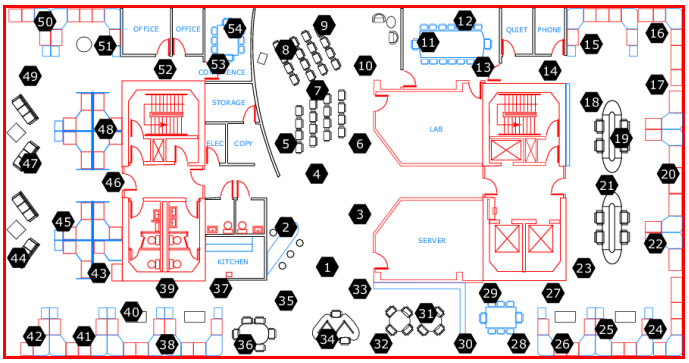

Fig. 2. Sensors in the Intel Berkeley Research lab

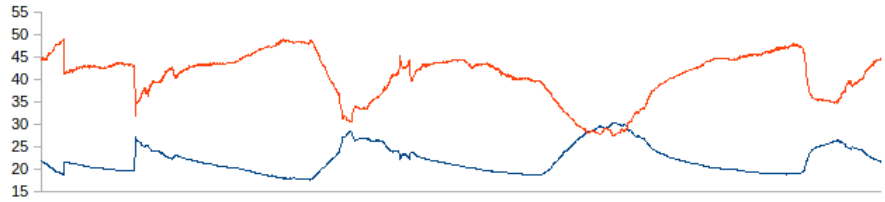

Fig. 3. Correlation between temperature (bottom) and humidity (top) time series.

Fig. 4. Intervals of the temperature (bottom) and humidity (top) time series.

Adopting these functions we obtained a temporal sequence made up of 816 intervals (81 for temperature, 94 for humidity, 255 for light and 386 for voltage). Then we added all the Allen's temporal relations between the intervals (332402 before, 1052 meets, 8872 overlaps, 163 starts, 7438 during, 131 finishes and 42 matches atoms) obtaining a relational sequence of about 350000 literals. The following literals represent a fragment of a sequence describing the relational representation of some time series, where each interval is described by three predicates

α(sensor, interval, label), begin(interval, s), end(interval, e)
where $\alpha \in$ {temperature, humidity, light, voltage}.

```
near(41,42). far(41,24).  far(42,24).
temperature(24,i1,high). begin(i1,0). end(i1,20).
light(41,i30,very_low). begin(i30,2). end(i30,16). ...
starts(i1,i2), before(i12,i34), ...
```

Table 1 reports the results of the algorithm when applied on the sequence previously described and using two different values for the minimum support. The fourth column reports the number of patters belonging to all the possible specializations whose support is greater than MinSupport. The fifth column reports the number of maximal patterns fulfilling all the constraints obtained by the algorithm. Some interval-based patterns discovered by the algorithm and expressing the time correlation and the information correlation are:

temperature(A,B,low), before(B,D), temperature(A,D,medium) [$s = 32.1\%$],
temperature(A,B,low), meets(B,D), humidity(A,D,high) [$s = 13\%$],
temperature(A,B,medium),overlaps(B,D),humidity(24,D,high),far(24,A) [$s = 24\%$].

5 Conclusion

Previous work on mining temporal patterns for interval-based sequences, as opposed to point-based events, include [12–14] but without using a logical descrip-

MinSupport	Level	Specializations	Candidates	Maximals	Time (secs)
10%	1	16	11	585	337.805
	2	174	42		
	3	691	213		
	4	3447	765		
	5	10672	1060		
	6	12378	408		
	7	3992	0		
20%	1	16	8	327	222.313
	2	128	30		
	3	518	160		
	4	2609	473		
	5	6636	381		
	6	4639	46		
	7	452	0		

Table 1. Detailed results of two experiments.

tion language. Other work in the research area of spatial data mining try to tackle into account complex representations with a logical language, however without considering temporal-based relations [15, 16].

The work presented in this paper can be related to that proposed in [17, 18]. However, we apply the proposed framework to a real world wireless sensor network data, and we used a powerful and general purpose multi-dimensional relational pattern mining system [5]. In [17] the authors represents a single sequence as a set of predicates and temporal relations. Each predicate is assumed to be hold in a given temporal interval, while the temporal relations are predicates expressing the Allen's temporal correlation between two predicates. Furthermore, each predicate is associated to a unique symbolic identifier indicating a specific temporal intervals, and temporal relations are expressed between those identifiers. Hence, every we use a predicate in a sequence we are implicitly assuming that it corresponds to a fluent predicate without having the possibility to introduce predicate that only express a structural relation between objects. Furthermore, as reported in [17], the algorithm they presented is not applicable to real world problems due to its high complexity. Indeed, they specialize a pattern by adding a literal, or by variable unification, or by introducing k^n (where k is the number of different Allen's relation and n corresponds to the number of possible predicate pairs) temporal restrictions between predicate pairs leading to an exponential time complexity.

The framework we presented in this paper can be used to solve complex temporal data mining tasks by using a relational interval-based description. The algorithm that we proposed is an extension of the efficient learning system presented in [5] with new dimensional operators. Preliminary experimental results prove that the framework can be applied to real world domains such as wireless sensor networks.

Acknowledgment

This work is partially supported by the Italian project PS121 "Infrastrutture di Telecomunicazione e Reti Wireless di Sensori nella Gestione di Situazioni di Emergenza".

References

1. Estrin, D., Culler, D., Pister, K., Sukhatme, G.: Connecting the physical world with pervasive networks. IEEE Pervasive Computing **1**(1) (2002) 59–69
2. Akyildiz, I., Su, W., Sankarasubramanian, Y., Cayirci, E.: A survey on sensor networks. IEEE Communication Magazine **40**(8) (2002) 102–114
3. Akyildiz, I., Su, W., Sankarasubramanian, Y., Cayirci, E.: Wireless sensor networks: a survey. IEEE Transactions Systems, Man and Cybernetics **38**(4) (2002) 393–422
4. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless sensor networks for habitat monitoring. In: International Workshop on Wireless Sensor Networks and Applications, ACM (2002) 88–97
5. Esposito, F., Di Mauro, N., Basile, T., Ferilli, S.: Multi-dimensional relational sequence mining. Fundamenta Informaticae **89**(1) (2008) 23–43
6. Ullman, J.: Principles of Database and Knowledge-Base Systems. Volume I. Computer Science Press (1988)
7. Jacobs, N., Blockeel, H.: From shell logs to shell scripts. In Rouveirol, C., Sebag, M., eds.: Proceedings of the 11th International Conference on Inductive Logic Programming. Volume 2157. Springer (2001) 80–90
8. Agrawal, R., Manilla, H., Srikant, R., Toivonen, H., Verkamo, A.: Fast discovery of association rules. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds.: Advances in Knowledge Discovery and Data Mining. AAAI Press (1996) 307–328
9. Hoppner, F.: Knowledge Discovery from Sequential Data. PhD thesis (2003)
10. Allen, J.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11) (1983) 832–843
11. Intel Berkeley Research Lab: Intel lab data. http://db.csail.mit.edu/labdata/labdata.html (2004)
12. shan Kam, P., chee Fu, A.W.: Discovering temporal patterns for interval-based events. In: Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery. LNCS, Springer (2000) 317–326
13. Hoppner, F.: Learning dependencies in multivariate time series. In: Proceedings of the 15th ECAI Workshop on Knowledge Discovery in (Spatio-) Temporal Data. (2002) 25–31
14. Laxman, S., Unnikrishnan, K., Sastry, P.: Generalized frequent episodes in event sequences. In: 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Workshop on Temporal Data Mining. (2002)
15. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: In Proceedings of the 4th International Symposium on Advances in Spatial Databases. (1995) 47–66
16. Malerba, D., Lisi, F.A.: An ilp method for spatial association rule mining. In: Working notes of the First Workshop on Multi-Relational Data Mining. (2001) 18–29

17. Lattner, A., Herzog, O.: Unsupervised learning of sequential patterns. In: ICDM 2004 Workshop on Temporal Data Mining: Algorithms, Theory and Applications. (2004)
18. Lattner, A., Herzog, O.: Mining temporal patterns from relational data. In: Lernen Wissensentdeckung Adaptivitt. (2005) 184–189