Applied Artificial Intelligence, 15:707–722, 2001

0883-9514/01 $12.00 +.00

LEARNING SPATIO-TEMPORAL RELATIONAL STRUCTURES

WALTER F. BISCHOF and TERRY CAELLI
Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada

We introduce a rule-based approach for learning and recognition of complex actions in terms of spatio-temporal attributes of primitive event sequences. During learning, spatio-temporal decision trees are generated which satisfy relational constraints of the training data. The resulting rules are used to classify new dynamic pattern fragments, and general heuristic rules are used to combine classification evidences of different pattern fragments.

The most current techniques for the encoding and recognition of actions use numerical machine-learning models that are not relational. They typically induce rules over unstructured sets of numerical attributes which are not indexed or linked via an underlying data relational structure. For example, recurrent neural networks (Caelli, Guan, & Wen, 1999) or hidden Markov models (Rabiner & Juang, 1993) assume that the input dynamical variables that compose complex actions are identified with respect to specific classes of actions. They, therefore, fail to function efficiently when dealing with the classification of specific actions embedded in animations consisting of multiple concurrent classes of actions. That is, the models have to assume that the correspondence between candidate and model features is known *before* rule generation (learning) or rule evaluation (matching) occurs. This assumption is inappropriate when complex models have to be learned, for example, when actions involving movements of multiple limb segments have to be learned.

Well-known symbolic relational learners, on the other hand, such as inductive logic programming (ILP) are not designed to deal efficiently with numerical data. Although they induce over relational structures (e.g., Horn clauses), they typically generalize or specialize over the symbolic variables and not so much over numerical attributes. It is thus rare that symbolic representations *explicitly* constrain the types of permissible numerical generalizations obtained from training data.

Address correspondence to Walter F. Bischof, Department of Computing Science, University of Alberta, Edmonton, Alberta, T6G 2E8, Canada. E-mail: wfb@ualberta.ca

Over the past few years we have explored methods for combining the strengths of both sources of model structures (Bischof & Caelli, 1994, 1997; Caelli & Bischof, 1997) by combining the expressiveness of ILP with the generalization models of numerical machine learning. This led to a class of numerical relational learners, which induce over numerical attributes in ways that are constrained by relational pattern models. Our approach, conditional rule generation (CRG) generates rules that are numerical decision trees which are linked together to satisfy relational constraints of the training data (see Figure 1). Relational constraints are introduced adaptively, i.e., they are added if they are required to make the classification rules apply efficiently.

Since CRG induces over a relational structure it requires general model assumptions, the most important being that the models are defined by a labeled graph, where relational attributes are defined only with respect to neighboring vertices. Such assumptions constrain the types of unary and binary features which can be used to resolve uncertainties (Figure 1).

In this article, we describe CRG_{ST}, a spatio-temporal extension of CRG for learning dynamic patterns and its application to animated scenes. We discuss representational issues, rule generation, and rule application. The inclusion of time makes modeling and algorithmic issues more challenging and requires the addition of further assumptions to make the problem tractable.

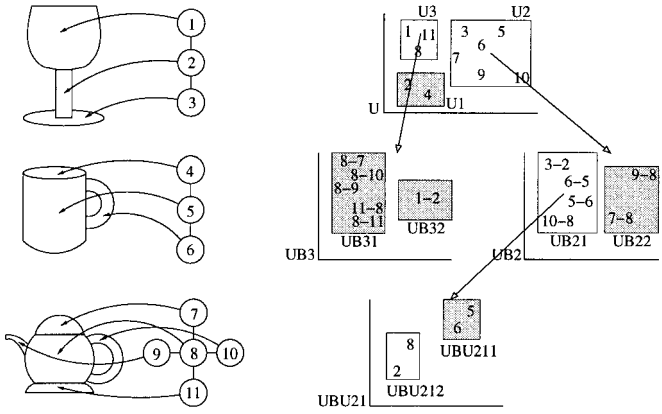

FIGURE 1. Example of input data and conditional cluster tree generated by CRG method. The left panel shows input data and the attributed relational structures generated for these data, where each vertex is described by a unary feature vector $\vec{b}$. In this example, we assume that there are two unary and two binary features. We further assume that there are two pattern classes, class 1 consisting of the drinking glass and the mug, and class 2 consisting of the teapot. The right panel shows a cluster tree generated for the data on the left. Numbers refer to the vertices in the relational structures, rectangles indicate generated clusters (defined by lower and upper bounds on each feature), grey ones are unique, and white ones contain elements of multiple classes. Classification rules are derived directly from this tree.

CONDITIONAL RULE GENERATION

In conditional rule generation (Bischof & Caelli, 1994), classification rules for patterns or pattern fragments are generated which include structural pattern information to the extent that is required for correctly classifying a set of training patterns. Conditional rule generation analyzes unary and binary features of connected pattern components and creates a tree of hierarchically organized rules for classifying new patterns. A generation of a rule tree proceeds in the following manner (see Figure 1).

First, the unary features of all parts of all patterns are collected into a unary feature space U in which each point represents a single pattern part. The feature space U is partitioned into a number of clusters U_i. In the example in Figure 1, we assume that there are two unary features and that each cluster is defined by lower and upper bounds for each of these features. Some of these clusters may be unique with respect to class membership (e.g., cluster U_1) and provide a classification rule: If a pattern contains a part p_r whose unary features $\vec{u}(p_r)$ satisfy the bounds of a unique cluster U_i then the pattern can be assigned a unique classification. The nonunique clusters contain parts from multiple pattern classes and have to be analyzed further. For every part of a nonunique cluster, we collect the binary features of this part with all adjacent parts in the pattern to form a (conditional) binary feature space UB_i. The binary feature space is clustered into a number of clusters UB_{ij}. In the example in Figure 1, we assume that there are two binary features and, as before, that each cluster is defined by lower and upper bounds for each of these features. Again, some clusters may be unique (e.g., clusters UB_{22} and UB_{31}) and provide a classification rule: If a pattern contains a part p_r whose unary features satisfy the bounds of cluster U_i, and there is another part p_s, such that the binary features $\vec{b}(p_r, p_s)$ of the pair $\langle p_r, p_s \rangle$ satisfy the bounds of a unique cluster UB_{ij} then the pattern can be assigned a unique classification. For nonunique clusters, the unary features of the second part p_s are used to construct another unary feature space UBU_{ij} which is again clustered to produce clusters UBU_{ijk}. This expansion of the cluster tree continues until all classification rules are resolved or a maximum rule length has been reached.

If unresolved rules remain at the end of the expansion procedure (which is normally the case), the rules are split into more discriminating rules using an entropy-based splitting procedure. Consider the cluster tree in Figure 1 with the nonunique cluster UBU_{212}. One way to proceed would be to recluster feature space UBU_{21} into a larger number of clusters. Alternatively, one can simply split cluster UBU_{212} along one of the feature dimensions. The latter method is used here.

Consider splitting the elements of an unresolved cluster C along a (unary or binary) feature dimension F. The elements of C are first sorted by their

feature (attribute) value $f(c)$, and then all possible cut points T midway between successive feature values in the sorted sequence are evaluated. For each cut point T, the elements of C are partitioned into two sets, $P_1 = \{c|f(c) \leq T\}$ with n_1 elements and $P_2 = \{c|f(c) > T\}$ with n_2 elements. We define the normalized partition entropy $H_P(T)$ as

$$H_P(T) = (n_1 H(P_1) + n_2 H(P_2))/(n_1 + n_2). \tag{1}$$

The cut point T_F that minimizes $H_P(T_F)$ is considered the best point for splitting cluster C along feature dimension F. The best split of cluster C is considered the one along the feature dimension F that minimizes $H_P(T_F)$. Furthermore, rather than splitting an unresolved leaf cluster C_L, one can split any cluster C_i in the parent chain of C_L. For each cluster C_i, the optimal split T_F is computed, and the cluster C_i that minimizes T_F is considered the optimal level for refining the cluster tree.

Rule splitting continues until all classification rules are unique or some termination criterion has been reached. This results in a tree of conditional feature spaces (Figure 1), and within each feature space, rules for cluster membership are developed in the form of a decision tree. Hence, CRG generates a tree of decision trees.

Classification rules are derived directly from the final cluster tree. For the cluster tree shown in Figure 1, the classification rule derived for cluster UB_{32}, for example, can be described as follows: If there is a part p_i with unary features $\vec{U}(p_i)$ within the bounds defined by cluster U_3, and part p_i is connected to some other part p_j such that the binary features of their relation, $\vec{b}(p_i, p_j)$, are within the bounds defined by cluster UB_{32}, then the pattern fragment $p_i - p_j$ belongs to class 1.

From the empirical class frequencies of all training patterns, one can derive an expected classification (or evidence vector) $\vec{E}$ associated with each rule (e.g., $\vec{E}(UBU_{212}) = [0.5, 0.5]$), given that it contains one element of each class. Similarly, one can compute evidence vectors for partial rule instantiations, again from empirical class frequencies of nonterminal clusters (e.g., $\vec{E}(UB_{21}) = [0.75, 0.25]$). Hence, an evidence vector $\vec{E}$ is available for every partial or complete rule instantiation.

Again, for the cluster tree in Figure 1, the classification rule derived from cluster UBU_{212} can be described as follows: If there is a part p_i with unary features $\vec{u}(p_i)$ within the bounds defined by cluster U_2, and part p_i is connected to some other part p_j such that the binary features of their relation, $\vec{b}(p_i, p_j)$, are within the bounds defined by cluster UB_{21}, and part p_j has unary features $\vec{u}(p_j)$ within the bounds defined by cluster UBU_{212}, then the pattern fragment $p_i - p_j$ belongs to classes 1 and 2 with probabilities [0.5, 0.5].

Conditional rule generation generates classification rules for pattern fragments in the form of symbolic, possibly fuzzy, Horn clauses. When the

classification rules are applied to some new pattern, one obtains one or more (classification) evidence vectors for each pattern fragment, and the evidence vectors have to be combined into a single evidence vector for the whole pattern. The combination rules can be learned (Wolpert, 1992), they can be knowledge-guided (Dillon & Caelli, 1997), or they can be based on general compatibility heuristics (Bischof & Caelli, 1997). In the latter approach, sets of instantiated classification rules are analyzed with respect to their compatibilities and rule instantiations that lead to incompatible interpretations are removed. This is particularly important in scenes composed of multiple patterns where it is unclear whether a chain $p_i - p_j - \cdots - p_n$ of pattern parts belongs to the same pattern or whether it is "crossing the boundary" between different patterns. Our heuristic approach is presented in detail below, after we have introduced the extension of CRG to spatio-temporal patterns.

$\mathbf{CRG_{ST}}$

We now turn to $\mathrm{CRG_{ST}}$, a generalization of CRG from a purely spatial domain into a spatio-temporal domain. Here, data consist typically of time-indexed pattern descriptions, where pattern parts are described by unary features, spatial part relations by (spatial) binary features, and changes of pattern parts by (temporal) binary features. In the following sections, we discuss representational issues, rule generation models, learning paradigms, and applications of the $\mathrm{CRG_{ST}}$ approach. In contrast to more popular temporal learners like hidden Markov models (Rabiner & Juang, 1993) and recurrent neural networks (Caelli et al., 1999), the rules generated from $\mathrm{CRG_{ST}}$ are not limited to first-order time differences, but can utilize more distant (lagged) temporal relations as a function of the data model and uncertainty resolution strategies. At the same time, $\mathrm{CRG_{ST}}$ allows for the generation of nonstationary rules, unlike stationary models like multivariate time series, which also accommodate correlations beyond first-order time differences but do not allow for the use of different rules at different time periods.

In the following, we illustrate each component of $\mathrm{CRG_{ST}}$ with an example where three different variations of grasp movements were learned: one where the hand moved in a straight path to the object, another where an obstacle in the direct path was avoided by moving over it, and a third where the obstacle was avoided by moving around it.

The movements were recorded using a Polhemus system (Raab, Blood, Steiner, & Jones, 1979) running at 120Hz for three sensors, one on the upper arm, one on the forearm, and one on the hand (see Figure 2). From the position data $(x(t), y(t), z(t))$ of these sensors, three-dimensional velocity $v(t)$, acceleration $a(t)$, curvature $k(t)$, and torsion $\tau(t)$ were extracted.

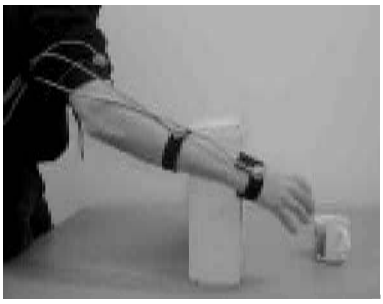

FIGURE 2. Grasping movement around an obstacle. The movement sensors were placed on the upper arm, the forearm, and the hand.

Sample time-plots of these measurements are shown in Figure 3. Each of the three movement types was recorded five times.

Representation of Spatio-Temporal Patterns

A spatio-temporal pattern is defined by a set of labeled time-indexed attributed features. A pattern P_i is thus defined in terms of $P_i = \{p_{i1}(\vec{a} : t_{ij}), \ldots, p_{in}(\vec{a} : t_{in})\}$, where $p_{ij}(\vec{a} : t_{ij})$ corresponds to part j of pattern

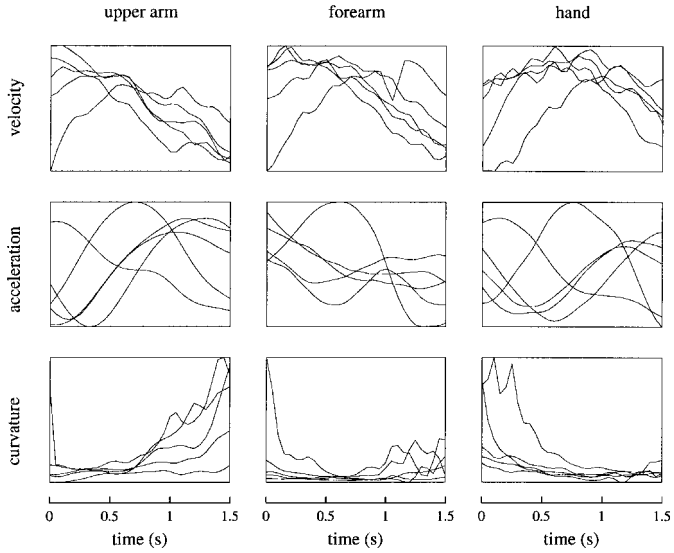

FIGURE 3. Sample time-plots of the movement sequences illustrated in Figure 2. The left column shows traces for the upper arm, the middle column for the forearm, and the right column for the hand. The first row shows time-plot for velocity (for a straight grasp movement), the second for acceleration (for a grasp movement over an obstacle), and the third for curvature (for a grasp movement around an obstacle). Each graph shows five samples for each action type. All measurements have been normalized for display purposes.

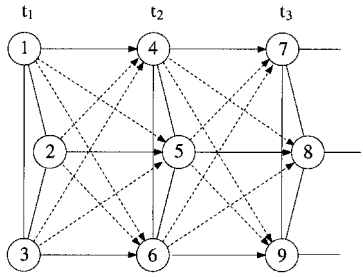

FIGURE 4. A spatio-temporal pattern consisting of three parts over three time-points. Undirected arcs indicate spatial binary connections, solid directed arcs indicate temporal binary connections between the same part at different time-points, and dashed directed arcs indicate temporal binary connections between different parts at different time-points.

i with attributes $\vec{a}$ that are true at time t_{ij}. The attributes $\vec{a}$ are defined with respect to specific labeled features, and are either unary (single-feature attributes) or binary (relational-feature attributes, either over space or over space-time), that is, $\vec{a} = \{\vec{u}, \vec{b}_s, \vec{b}_t\}$ (see Figure 4). Examples of unary attributes $\vec{u}$ include area, brightness, position; spatial binary attributes $\vec{b}_s$ include distance, relative size; and temporal binary attributes $\vec{b}_t$ include changes in unary attributes over time, such as size, orientation change, or long-range position change.

For the "grasp" example, the definition of the spatio-temporal patterns is straightforward. At every time-point, the patterns consist of three parts, one for each sensor, each part being described by unary attributes three-dimentional position, velocity, acceleration, curvature, and torsion, i.e., $\vec{u}(p_{i,t}) = [x, y, z, v, a, k, \tau]$. Binary attributes were defined by simple differences, i.e., the spatial attributes were defined as $\vec{b}_s(p_{i,t}, p_{j,t}) = \vec{u}(p_{j,t}) - \vec{u}(p_{i,t})$, and the temporal attributes were defined as $\vec{b}_t(p_{i,t}, p_{j,t+1}) = \vec{u}(p_{j,t+1}) - \vec{u}(p_{i,t})$.

Our data model, and consequently our rules, are subject to spatial and temporal adjacency (in the nearest neighbor sense) and temporal monotonicity, i.e., features are only connected in space and time if they are spatially or temporally adjacent, and the temporal indices for time must be monotonically increasing (in the "predictive" model) or decreasing (in the "causal" model). Although this limits the expressive power of our representation, it is still more general than strict first-order discrete time dynamical models such as hidden Markov models or Kalman filters.

For $\mathrm{CRG}_{\mathrm{ST}}$ finding an "interpretation" involves determining sets of linked lists of attributed and labeled features, which are causally indexed (i.e., the temporal indices must be monotonic) and maximally index a given pattern.

Rule Learning

CRG_{ST} generates classification rules for spatio-temporal patterns involving a small number of pattern parts subject to the following constraints: First, the pattern fragments involve only pattern parts that are adjacent in space and time. Second, the pattern fragments involve only noncyclic chains of parts. Third, temporal links are followed in the forward direction only to produce causal classification rules that can be used in classification and in prediction mode.

Rule learning proceeds in the following way: First, the unary features of all parts (of all patterns at all time points), $\vec{u}(p_{i,t})$, $i = 1, \ldots, n$, $t = 1, \ldots, T$, are collected into a unary feature space U in which each point represents the feature vector of one part at one time-point. From this point onward, cluster tree generation proceeds exactly as described in the second section, except that expansion into a binary space can now follow either spatial binary relations $\vec{b}_s$ or temporal binary relations $\vec{b}_t$. Furthermore, temporal binary relations $\vec{b}_t$ can be followed only in strictly forward direction, analyzing recursively temporal changes of either the same part, $\vec{b}_t(p_{i,t}, p_{i,t+1})$ (solid arrows in Figure 4), or of different pattern parts, $\vec{b}_t(p_{i,t}, p_{j,t+1})$ (dashed arrows in Figure 4) at subsequent time-points t and $t+1$. Again, the decision about whether to follow spatial or temporal relations is simply determined by entropy-based criteria, consistent with the usual minimum description length (MDL) criterion for decision trees (Quinlan, 1995).

For the "grasp" example, an example of a classification rule generated by CRG_{ST} is the following rule, which happens to be of the form $U - B_t - U - B_t - U$, with v = velocity; a = acceleration; Δx = displacement (over time) in x; Δy = displacement (over time) in y:

if $\vec{u}(p_{i,t})$	with $-1.34 \leq v \leq 7.9$ and $-2.93 \leq a \leq 1.54$
and $\vec{b}_t(p_{i,t}, p_{j,t+1})$	with $-0.16 \leq \Delta x \leq 0.07$ and $-6.51 \leq \Delta y \leq 5.37$
and $\vec{u}(p_{j,t+1})$	with any value
and $\vec{b}_t(p_{j,t+1}, p_{k,t+2})$	with $-5.39 \leq \Delta x \leq 0.08$ and $-6.51 \leq \Delta y \leq 5.37$
and $\vec{u}(p_{k,t+2})$	with $4.74 \leq v \leq 5.04$ and $-.78 \leq a \leq -0.06$
then	pattern fragment $p_{i,t} - p_{j,t+1} - p_{k,t+2}$ is part of a grasping action moving over an obstacle.

In plain language, this classification rule reads as follows: If there is any sensor at any time-point t with instantaneous velocity in the range $[-1.34, 7.9]$ and acceleration in the range $[-2.93, 1.54]$, and the sensor position changes by Δx in the range $[-0.16, 0.07]$ and by Δy in the range $[-6.51, 5.37]$ to the next time step, etc., then this is part of a grasping action

over an obstacle. Note that the rules do not refer to particular sensors, even though this would be possible, and indeed helpful in this particular example. In general, however, identification of particular pattern parts may not be possible. As discussed in the introduction, it is one of the fundamental assumptions of $\mathrm{CRG_{ST}}$ that the correspondence between pattern parts and model parts is *not* known a priori, and this is what sets it apart from approaches such as recurrent neural networks or bidden Markov models.

In general, $\mathrm{CRG_{ST}}$ produces classification rules of the form $U_i - B_{ij} - U_j - B_{jk} - \ldots$, with B involving spatial ($\vec{b}_s$) and/or temporal ($\vec{b}_t$) binary relations, and the resultant Horn clause rules are of the form:

$$\begin{aligned} \text{class} \Leftarrow \quad & \text{part } p_{i,t} \text{ at time t with attributes } (p_{i,t}) \text{ AND} \\ & \text{part relation } \langle p_{i,t}, p_{j,t'} \rangle \text{ at times } t \text{ and } t' \\ & \text{with attributes } \vec{b}(p_{i,t}, p_{j,t'}) \text{ AND} \\ & \text{part } p_{it'} \text{ at time } t' \text{ with attributes } (p_{j,t'}) \text{ AND } \ldots \end{aligned}$$

Rule Application

A set of classification rules is applied to a spatio-temporal pattern in the following way. Starting from each pattern part (at any time point), all possible sequences (chains) of parts are generated subject to the constraints the only adjacent parts are involved and no loops are generated. (Note that the same spatio-temporal adjacency constraints and temporal monotonicity constraints were used for rule generation.) Each generated chain $S_i = \langle p_{i1}, p_{i2}, \ldots, p_{in} \rangle$ is then classified using the classification rules, and the evidence vectors of all rules instantiated by S_i are averaged to obtain the evidence vector $\vec{E}(S_i)$ of the chain S_i. Furthermore, the set S_p of all chains that start at p is used to obtain an initial evidence vector for part p:

$$\vec{E}(p) = \frac{1}{|\mathcal{S}_p|} \sum_{S \in \mathcal{S}_p} \vec{E}(S), \tag{2}$$

where $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$. Evidence combination based on eq. (2) is adequate if it is known that a single pattern is to be recognized. In the "grasp" example, this would be the case if it is known that a single movement is being presented.

However, $\mathrm{CRG_{ST}}$ is designed to work in more general situations where different pattern types overlap in space and time. This is the case, for example, when two or more different movement patterns are presented at the same time, or when the movement type changes over time. In this case, the simple scheme based on eq. (2) can produce incorrect results because

some chains of parts may not be contained completely within a single pattern but "cross" different pattern types. Such "crossing" chains are likely to be classified in a arbitrary way, and to the extent that they can be detected and eliminated, the part classification based on eq. (2) can be improved.

We use general heuristics for detecting rule instantiations involving parts belonging to different patterns. One such heuristic is based on the following idea. If a chain $S_i = \langle s_{i1}, s_{i2}, \ldots, s_{in} \rangle$ does not cross boundaries of objects, then the evidence vectors $\vec{E}(s_{i1}), \vec{E}(s_{i2}), \ldots, \vec{E}(s_{in})$ are likely to be similar, and dissimilarity of the evidence vectors suggests that S_i may be a "crossing" chain. This similarity can be captured in the following way (McCane & Caelli, 1997): For a chain $S_i = \langle p_{i1}, p_{i2}, \ldots, p_{in} \rangle$, we compute the compatibility vector

$$\vec{w}(S_i) = \frac{1}{n} \sum_{k=1}^{n} \vec{E}(p_{ik}), \tag{3}$$

where $\vec{E}(p_{ik})$ is the evidence vector of part p_{ik}. This compatibility measure can be used directly in an iterative relaxation scheme for updating the part evidence vectors:

$$\vec{E}^{t+1}(p) = \Phi\left(\frac{1}{Z} \sum_{S \in S_p} \vec{w}^{(t)}(S) \otimes \vec{E}(S) \right), \tag{4}$$

where Φ is the logistic function $\Phi(z) = (1 + \exp[-20(z - 0.5)])^{-1}$. Z a normalizing factor, and the binary operator $\otimes$ is defined as a component-wise vector multiplication $[a\ b]^T \otimes [c\ d]^T = [ac\ bc]^T$. Convergence of the relaxation scheme eq. (4) is typically obtained in about 10–20 iterations, and the updated part evidence vectors then reflect the partitioning of the test pattern into distinct subparts.

In summary, application of $\mathrm{CRG_{ST}}$ rules to a spatio-temporal pattern P_i (as shown in Figure 4) proceeds as follows:

1. For all pattern parts $\mathrm{p_i}$, extract all part chains $\mathrm{p_i - p_j - p_k - \ldots}$ starting at $\mathrm{p_i}$, up to a specified maximal length.
2. Classify all part-chains S_i using the $\mathrm{CRG_{ST}}$ rules, each producing an evidence vector $\vec{E}(S_i)$.
3. Compute initial compatibility vectors $\vec{w}(S_i)$ using Eq. (3).
4. Run the relaxation scheme Eq. (4) until convergence. This produces a final evidence vector $\vec{E}(p_i)$ for each part.

Performance of $\mathrm{CRG_{ST}}$

Performance of $\mathrm{CRG_{ST}}$ was tested with the "grasp" example in a leave-one-out paradigm, i.e., in each run, movement classes were learned using all but one

TABLE 1 Performance of CRG_{ST} for learning three different types of grasping actions. The first three columns indicate what attributes were used for unary, spatial binary and temporal binary relations, and the last column indicates the percentage of test pattern points that was classified correctly.

$\vec{u}$	$\vec{b}_s$	$\vec{b}_t$	correct
xyz	xyz	xyz	95.4%
–	xyz	xyz	96.3%
–	–	xyz	43.1%
va	va	va	52.2%
–	va	va	46.6%
–	–	va	28.3%
kτ	kτ	kτ	34.6%
–	kτ	kτ	40.7%
–	–	kτ	28.9%
xyzva	xyzva	xyzva	90.8%
–	xyzva	xyzva	96.5%
–	–	xyzva	33.1%

Dashes indicate that no feature was used. xyz = position in 3D; v = velocity: a = acceleration; k = curvature; τ = torsion.

pattern, and the resulting rule system was used to classify the remaining pattern. Results of these tests are shown in Table 1 for different attribute combinations for unary, spatial binary, and temporal binary relations. The last column indicates what percentage of pattern parts was classified correctly on average. Although each test pattern consisted of a single movement, this was not assumed by the classification algorithm in order to show the basic classification performance. Using the "single-movement" assumption, e.g., in a winner-take-all scheme, would lead to somewhat higher classification percentages.

The results show that classification performance varies, not unexpectedly, with the choice of attribute sets. For the simple movement patterns used here, position information, possibly enhanced by velocity and acceleration information, was clearly sufficient for encoding and learning the movement patterns. Curvature and torsion information alone was insufficient, which is not surprising given that the movements were fairly linear.

The results show that CRG_{ST} is a promising technique for the learning of motion patterns. Obviously, the movement patterns used here were very simple, but work is currently in progress on the encoding and learning of much more complex movement sequences, as well as on extensions of temporal coding to allow temporal interval modeling.

INCORPORATING DOMAIN MODEL CONSTRAINTS INTO RULE GENERATION

The definition of spatio-temporal patterns introduced in the third section is very general and applies to situations where no domain knowledge is

available. Learning of patterns may be made more efficient through introduction of relational constraints based on domain knowledge. For example, for the recognition of human body movements, the spatial relation between hand and elbow may be much more diagnostic than the relation between hand and knee, or, more generally, intralimb spatial relations are more diagnostic than interlimb spatial relations. For these reasons, arbitrary model-based constraints can be introduced into the underlying relational structure, thus covering the range from fully connected nondirected relational models to specific directed relational models. Obviously, in situations where no domain knowledge is available, the most general model should be used, and learning is consequently slower and suboptimal. Conversely, when sufficient domain knowledge is available, strong constraints can be imposed on the relational model, and learning is consequently more efficient.

The model-based $\mathrm{CRG}_{\mathrm{ST}}$ approach is illustrated in an example where the classification of four different variations of lifting movements were learned, two where a heavy object was lifted, and two where a light object was lifted. Both objects were either lifted with a knees bent and a straight back (''good lifting''), or with knees straight and the back bent (''bad lifting''). Thus, there were four movement classes:

1. good lifting of heavy object,
2. good lifting of light object,
3. bad lifting of a heavy object,
4. bad lifting of a light object.

The movements are quite difficult to discriminate, even for human observers. This was done in order to test the limits of the movement learning system.

The movements were recorded using a Polhemus system (Raab et al., 1979) running at 120Hz for six sensors, located on the hip, above the knee, above the foot, on the upper arm, on the forearm, and on the hand of the left side of the body (see Figure 5). Each movement type recorded five times. From the position data $(x(i), y(t), z(t))$ of these sensors, three-dimensional velocity $v(t)$ and acceleration $a(t)$ were extracted, both w.r.t. arc length $ds(t) = (dx^2(t) + dy^2(t) + dz^2(t))^{1/2}$, i.e., $v(t) = ds(t)/dt$ and $a(t) = (d^2s(t)/dt^2$ (Mokhtarian, 1997). Sample time-plots of these measurements are shown in Figure 6.

The spatio-temporal patterns were defined in the following way: at every time point, the patterns consisted of six parts, one for each sensor, each part being described by unary attributes $\vec{u} = [x, y, z, v, a]$. Binary attributes were defined by simple differences, i.e., the spatial attributes were defined as $\vec{b}_s(p_{it}, p_{jt}) = \vec{u}(p_{it})$, and the temporal attributes were defined as $\vec{b}_t(p_{it}, p_{jt+1}) = u(p_{jt+1}) - \vec{u}(p_{it})$.

FIGURE 5. Lifting a heavy object. The movement sensors were placed on the hip, above the knee, above the foot, on the upper arm, on the forearm, and on the hand of the left body side.

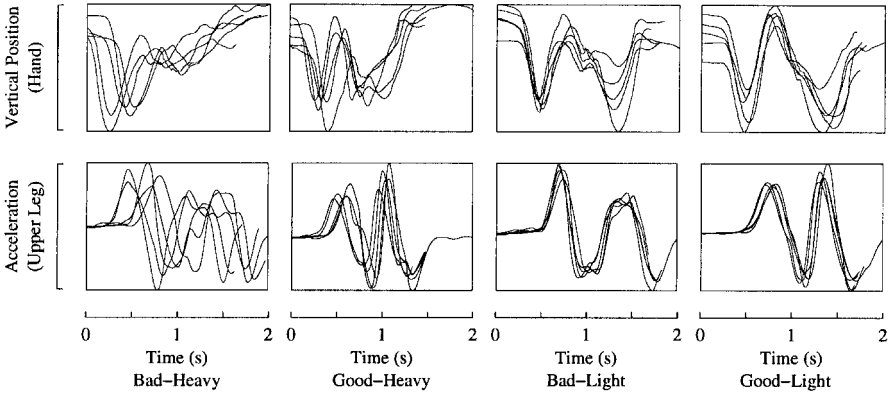

FIGURE 6. Sample time-plots of the movement sequences illustrated in Figure 5. The first row shows time-plots for the vertical position of the sensor placed on the hand, and the second row the acceleration of the sensor placed above the knee. The four columns show traces the four movement classes (see text for further details).

Performance of $\mathrm{CRG}_{\mathrm{ST}}$ was tested with a leave-one-out paradigm, i.e., in each test run, movement classes were learned using all but one sample, and the resulting rule system was used to classify the remaining pattern, as described in the third section. The system was tested with three attribute combinations and four pattern models. The three attributes combinations were:

1. $\vec{\mathrm{u}} = [\mathrm{x}, \mathrm{y}, \mathrm{z}]$,
2. $\vec{u} = [v, a]$,
3. $\vec{\mathrm{u}} = [\mathrm{x}, \mathrm{y}, \mathrm{z}, \mathrm{v}, \mathrm{a}]$.

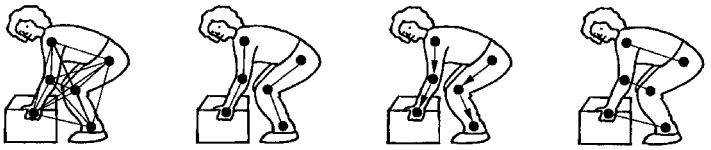

FIGURE 7. Sketch of the four pattern models used for the recognition of lifting movements. From left to right, the sketches show the fully connected relational model, the nondirectional intralimb model, the directional intralimb model, and an interlimb model. See text for further explanations.

The four pattern models were (see Figure 7):

1. a fully connected relational model (i.e., binary relations were defined between all six sensors),
2. a nondirectional intralimb model, i.e., binary relations were defined between hip-knee, knee-foot, upper arm-forearm, and forearm-hand,
3. a directional intralimb model (i.e., binary relations were defined as in 2 but only in one direction),
4. an interlimb model (i.e., binary relations were defined between hip-upper arm, knee-forearm, and foot-hand).

Results of these tests are shown in Table 2, for the attribute subsets and the pattern models just described. The results show that performance is fairly high, in spite of the fact that the movement patterns are not easy to discriminate for human observers. Best performance is reached for the intralimb directional model (see Figure 7) and the full feature combination *xyzva*. Even though performance for feature combination *va* is very low, the two features improve, not unexpectedly, performance for the *xyz* feature combination (Kittler, Hatef, Duin, & Matas, 1998).

An example of a classification rule produced with the directional intralimb model is the following with $V =$ velocity, $A =$ acceleration, $\Delta V =$ velocity difference between different sensors or for the same sensor over different time points, $\Delta A =$ acceleration difference between different sensors or for the same sensor over different time points.

if $\vec{u}(p_{i,t})$	with any value
and $\vec{b}_s(p_{i,t}, p_{j,t})$	with $-57 \leq \Delta V \leq 114$ and $-580 \leq \Delta A \leq 550$
and $\vec{u}(p_{j,t})$	with $A \leq 180$
and $\vec{b}_t(p_{j,t}, p_{k,t+1})$	with $-249 \leq \Delta V \leq 73$ and $181 \leq \Delta A \leq 2210$
and $\vec{u}(p_{k,t+1})$	with $17 \leq V \leq 24$ and $132 \leq A \leq 301$
then	pattern fragment $p_{i,t} - p_{j,t} - p_{k,t+1}$ is part of a "good lifting" of a heavy object

TABLE 2 Performance of CRG_{ST} for learning four different types of lifting actions. The first column indicates what relational model was used, and the three remaining columns give the average performance for three different attributes combinations (xyz= position in 3D; v= velocity: a= acceleration). Each cell gives raw percentage correct for a model+ - feature set combination. The number in parentheses gives classification performance under the assumption that a single movement pattern is present and is obtained from the former using a simple winner take-all criterion.

Model	*xyz*	*va*	*xyzva*
Fully connected	48.7 (85)	24.6 (30)	45.7 (75)
Intralimb nondirectional	46.2 (75)	32.4 (32)	46.2 (75)
Intralimb directional	52.7 (85)	24.1 (20)	63.3 (90)
Interlimb nondirectional	41.4 (60)	22.1 (5)	42.1 (70)

In plain language, this rule says the following: If the relative velocity between the upper and lower limb is in the range $[-57, 114]$ and that of the relative acceleration in the range $[-580, 550]$, and the lower limb has an acceleration less than 180, and to the next time-step, velocity change of the lower limb is in the range $[-249, 73]$ and that of acceleration change is in the range [181, 2210], and at the next time-point velocity of the lower limb is in the range [17, 24] and that of acceleration in the range [132, 301], then this is part of a good lifting of a heavy object. As explained in the third section, CRG_{ST} rules do not refer to specific sensors. However, given the (directional intralimb) model constraints, parts $p_{i,t}$ and $p_{j,t}$ can only refer to one of the following combinations: upper arm + forearm, forearm + hand, hip + knee, knee + foot.

CONCLUSIONS

Most current learners are based upon rules defined iteratively in terms of expected states and/or observations at time $t+1$ given those at time t. Examples include hidden Markov models and recurrent neural networks. Although these methods are capable of encoding the variations that occur in signals over time and can indirectly index past events of varying lags, they do not have the explicit expressiveness of CRG_{ST} for relational time-varying structures. Relational learners like CRG_{ST} can also index events and states hierarchically and allow for explicit rules capable of including dependencies between *labeled* intervals. This allows for rules which include terms such as "while," "before," "after," etc. With this in mind, our future work involves combining aspects of interval temporal logic with the capacity for induction over intervals as done in CRG_{ST}. In all, there is much more research to be done in the area of spatio-temporal learning, and the exploration of spatio-temporal data structures, which are best suited to the encoding and efficient recognition of complex spatio-temporal events.

REFERENCES

Bischof, W. F., and T. Caelli. 1994. Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation. *Pattern Recognition* 27:1231–1248.

Bischof, W. F., and T. Caelli. 1997. Scene understanding by rule evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12:1284–1288.

Caelli, T., and W. F. Bischof, eds. 1997. *Machine Learning and Image Interpretation*. New York: Plenum.

Caelli, T., L. Guan, and W. Wen. 1999. Modularity in neural computing. *Proceedings of the IEEE* 87:1497–1518.

Dillon, C., and T. Caelli. 1997. Cite – scene understanding and object recognition. In *Machine Learning and Image Interpretation*, eds. T. Caelli and W. F. Bischof, 119–187. New York: Plenum.

Kittler, J., M. Hatef, R. P. W. Duin, and J. Matas. 1996. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20:226–239.

McCane, B., and T. Caelli. 1997. Fuzzy conditional rule generation for the learning and recognition of 3d objects from 2d images. In *Machine Learning and Image Interpretation*, eds T. Caelli and W. F. Bischof, 17–66. New York: Plenum.

Mokhtarian, F. 1997. A theory of multiscale, torsion-based shape representation for space curves. *Computer Vision and Image Understanding* 68:1–17.

Quinlan, J. R. 1995. MDL and categorical theories (continued). In *Proceedings of the 12th International Conference on Machine Learning*, pp. 464–470, Tahoe City, CA.

Raab, F. H., E. B. Blood, T. O. Stiener, and H. R. Jones. 1979. Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace and Electronic Systems* AES-15:709–720.

Rabiner, L., and B.-H. Juang. 1993. *Fundamentals of Speech Recognition*. New York: Prentice-Hall.

Wolpert, D. H. 1992. Stacked generalization. *Neural Networks* 5:241–259.