Learning Actions: Induction over Spatio-Temporal Relational Structures - CRG$_{\mathrm{ST}}$

Walter F. Bischof and Terry Caelli [1]

Abstract. 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 that satisfy relational constraints of the training data. The resulting rules, in form of Horn clause descriptions, are used to classify new dynamic pattern fragments, and general heuristic rules are used to combine classification evidences of different pattern fragments.

1 Introduction

Most current techniques for the encoding and recognition of actions use numerical machine learning models which are not relational in so far as they typically induce rules over numerical attributes which are not indexed or linked via an underlying data structure (e.g. a relational structure description or a directed acyclic graph, DAG). Therefore most learning models assume that the correspondence between candidate and model features is known *before* rule generation (learning) or rule evaluation (matching) occurs. This assumption is dangerous when large models or test data are involved, as is the case in complex actions involving, for example, the tracking of multiple limb segments of human operators. On the other hand well known symbolic relational learners like Inductive Logic Programming (ILP) are not designed to apply efficiently to numerical data. So, although they are suited to induction over relational structures (e.g. Horn clauses), they typically generalize or specialize over the symbolic variables and not so much over numerical attributes. More specifically, it is very rare that symbolic representation *explicitly* constrains the types of permissible numerical learning or generalizations obtained from training data.

Over the past six years we have explored methods for combining the strengths of both sources of model structures [1, 2, 3] by combining the expressiveness of ILP with the generalization models of numerical machine learning to produce a class of numerical relational learning which induce over numerical attributes in ways which are constrained by relational pattern or shape models. Our approach, Conditional Rule Generation (CRG), generates rules that consist of attributed linked lists of pattern (shape) features which, together, completely cover the training data but the generated rules are ordered with respect to their discriminatory power with respect to both attributes and features (see Figure 1).

Since CRG induces over a relational structure it requires general model assumptions, the most important being that the models (shapes) 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).

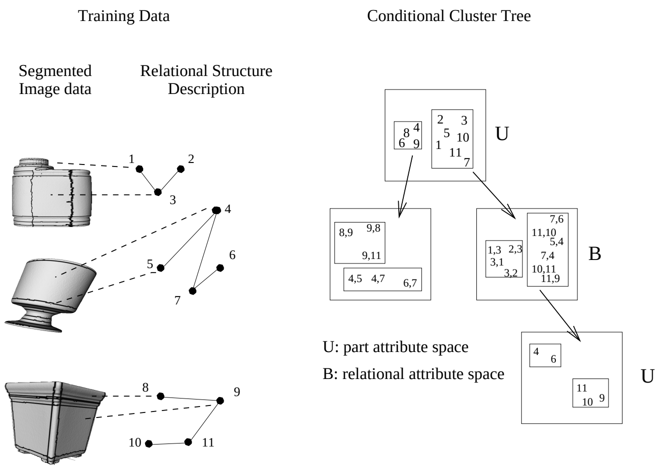

Figure 1. Example of input data and conditional cluster tree generated by CRG method. The left panel shows segmented input data with a sketch of the relational structure descriptions generated for these data. The right panel shows a cluster tree generated for the data on the left. Classification rules of the form $U_i - B_{ij} - U_j \ldots$ are derived directly from this tree [6].

In this paper, we describe CRG$_{\mathrm{ST}}$, a spatio-temporal extension of CRG for learning dynamic patterns and its application to animated scenes. We discuss representational issues, rule generation models 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.

2 Conditional Rule Generation

In Conditional Rule Generation [1], classification rules for patterns or pattern fragments are generated that include structural pattern information to the extent that is required for classifying correctly a set of training patterns. CRG analyzes unary and binary features of connected pattern components and creates a tree of hierarchically organized rules for classifying new patterns. 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. Some of these clusters may be unique with respect to class mem-

[1] Department of Computing Science, University of Alberta, Edmonton, T6G 2H1, Canada, Email: (wfb,tcaelli)@ualberta.ca

bership 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 non-unique clusters contain parts from multiple pattern classes and have to be analyzed further. For every part of a non-unique 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}. Again, some clusters may be unique 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 an other 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 non-unique clusters, the unary features of the second part p_s are used to construct another unary feature space UBU_{ij} that 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 there remain unresolved rules at the end of the expansion procedure (which is normally the case), the generated rules are split into more discriminating rules using an entropy-based splitting procedure where the elements of a cluster are split along a feature dimension such that the normalized partition entropy $H_P(T) = (n_1 H(P_1) + n_2 H(P_2))/(n_1 + n_2)$ is minimized, where H is entropy. 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.

CRG 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 [12], they can be knowledge-guided [7], or they can be based on general compatibility heuristics [2]. 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 $p_i - p_j - \ldots - p_n$ of pattern parts belongs to the same pattern or whether it is "crossing the boundary" between different patterns. Through application of these compatibility heuristics, we solve two problems at the same time, namely classification of pattern parts and segmentation of different patterns, eliminating the requirement of having to group the image into regions corresponding to single objects before the image regions have been classified.

3 CRG$_{\mathrm{ST}}$

We now turn to CRG$_{\mathrm{ST}}$, 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 CRG$_{\mathrm{ST}}$ approach. In contrast to more popular temporal learners like hidden Markov models [11] and recurrent neural networks [4], the rules generated from CRG$_{\mathrm{ST}}$ 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, CRG$_{\mathrm{ST}}$ allows for the generation of non-stationary 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.

3.1 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 $P_i = \{p_{i1}(\vec{a} : t_{i1}), \ldots, p_{in}(\vec{a} : t_{in})\}$ where $p_{ij}(\vec{a} : t_{ij})$ corresponds to part j of pattern i with attributes $\vec{a}$ that are true at time j. The attributes $\vec{a}$ are defined with respect to specific labeled features, and are restricted to arity 1 (unary, i.e. single feature attributes) or 2 (binary, i.e. relational feature attributes), that is, $\vec{a} = \{\vec{u}, \vec{b}_s, \vec{b}_t\}$ (see Figure 2). 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, long range position change, etc. Our data model and consequent 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 that the temporal indices for time must be monotonically increasing ("predictive" model) or decreasing ("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, for example, hidden Markov models or Kalman filters.

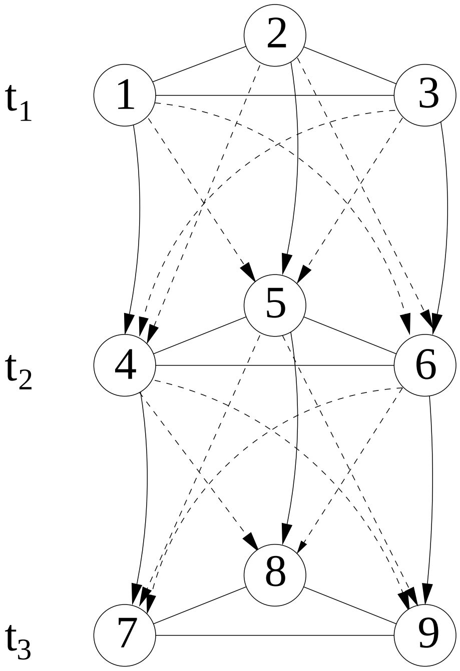

Figure 2. Illustration of a spatio-temporal pattern consisting of three parts over three time-points. Undirected arcs indicate spatial binary connections, solid directed indicate temporal binary connections between the same part at different time-points, and dashed directed arcs indicate temporal binary connections between different parts at different time-points.

For CRG$_{\rm ST}$ an "interpretation" then involves determining the smallest set of linked lists of attributed and labeled features, causally indexed (i.e. the starting times must be monotonically indexed) over time, which maximally index a given pattern, and it is defined by directed paths within the directed acyclic graph (DAG) which covers all examples and classes in the training set as,illustrated in Figure 2.

3.2 Rule Learning

CRG$_{\rm ST}$ generates classification rules for spatio-temporal patterns involving a small number of pattern parts subject to the following constraints: 1) The pattern fragments involve only pattern parts that are adjacent in space and time, 2) the pattern fragments involve only noncyclic chains of parts, 3) 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), $\vec{u}(p_{it})$, $i = 1, \ldots, n$, $t = 1, \ldots, T$, are collected into a unary feature space U in which each point represents a single pattern part at any time point $t = 1, \ldots, T$. From this unary feature space, cluster tree expansion can proceed in two directions, in the spatial domain and in the temporal domain. In the spatial domain cluster tree generation proceeds exactly as described in Section 2 following spatial binary relations, etc. In the temporal domain, binary relations can be followed only in strictly forward (predictive) or backward (causal) directions, analyzing recursively temporal changes of either the same part, $\vec{b}_t(p_{it}, p_{it+1})$ (solid arrows in Figure 2), or of different pattern parts, $\vec{b}_t(p_{it}, p_{jt+1})$ (dashed arrows in Figure 2) at subsequent timepoints. This leads to a conditional cluster tree as shown in Figure 1, except that the relational attribute spaces B can be either spatial or temporal, in accordance with the usual Minimum Description Length (MDL) criterion for Decision Trees[9].

CRG$_{\rm ST}$ produces classification rules of the form $U_i - B_{ij} - U_j - B_{jk} - \ldots$ involving spatial and/or temporal binary relations. The resultant Horn clause rules are of the form:

class ⇐ part(i at time j with attributes k) AND
part relations(ij at time j+n with attributes u) AND
part(l at time j+m: with attributes s) AND
...

These rules cover all training examples and define a path in the DAG discussed above.

From the empirical class frequencies of all training patterns one can derive an expected classification vector, or evidence vector $\vec{E}$ associated with each rule. We can also compute evidence vectors for partial rule instantiations, again from empirical class frequencies of non-terminal cluster spaces. Hence, an evidence (classification) vector $\vec{E}$ is available for every partial or complete rule instantiation, as discussed in the next section.

3.3 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 using parallel, iterative deepening, subject to the constraints the only adjacent parts are involved and no loops are generated. Note, again, that spatio-temporal adjacency and temporal monotonicity constraints are used for rule generation. Each chain is classified using the classification rules. Expansion of each chain $S_i = < p_{i1}, p_{i2}, \ldots, p_{in} >$ terminates if one of the following conditions occurs: 1) the chain cannot be expanded without creating a cycle, 2) all rules instantiated by S_i are completely resolved, or 3) the binary features $\vec{b}_s(p_{ij}, p_{ij+1})$ or $\vec{b}_t(p_{ij}, p_{ij+1})$ do not satisfy the features bounds of any rule.

If a chain S cannot be expanded, the evidence vectors of all rules instantiated by S are averaged to obtain the evidence vector $\vec{E}(S)$ of the chain S. Further, the set $\mathcal{S}_p$ of all chains that start at p is used to obtain an initial evidence vector for part p:

$$\vec{E}(p) = \frac{1}{\#(\mathcal{S}_p)} \sum_{S \in \mathcal{S}_p} \vec{E}(S). \tag{1}$$

where $\#(\mathcal{S})$ denotes the cardinality of the set $\mathcal{S}$. Evidence combination based on (1) is adequate if it is known that a single pattern is to be recognized. However, if the test pattern consists of multiple patterns then this simple scheme can easily produce incorrect results because some some part chains may not be contained completely within a single pattern but "cross" spatio-temporal boundaries between patterns. This occurs when actions corresponding to different types cross can intersect in time and/or space. These chains are likely to be classified in a arbitrary way. To the extent that they can be detected and eliminated, the part classification based on (1) can be improved.

We use general heuristics for detecting rule instantiations involving parts belonging to different patterns. They are based on measuring the compatibility of part evidence vectors and chain evidence vectors. More formally, the compatibility measure can be characterized as follows. For a chain $S_i =< p_{i1}, p_{i2}, ..., p_{in} >$,

$$\vec{w}(S_i) = \frac{1}{n} \sum_{k=1}^{n} \vec{E}(p_{ik}) \tag{2}$$

where $\vec{E}(p_{ik})$ refers to the evidence vector of part p_{ik}. Initially, this can be found by averaging the evidence vectors of the chains which begin with part p_{ik}. Then the compatibility measure is used for updating the part evidence vectors using an iterative relaxation scheme [8]:

$$\vec{E}^{(t+1)}(p) = \Phi\left(\frac{1}{Z} \sum_{S \in S_p} \vec{w}^{(t)}(S) \otimes \vec{E}(S)\right), \tag{3}$$

where Φ is the logistic function, Z a normalizing factor $Z = \sum_{S \in S_p} w^{(t)}(S)$, and the binary operator $\otimes$ is defined as a component-wise vector multiplication $[a\ b]^T \otimes [c\ d]^T = [ac\ bc]^T$. The updated part evidence vectors then reflect the partitioning of the test pattern into distinct subparts.

4 Example

The CRG$_{\rm ST}$ approach is illustrated in an example where three different variations of grasp movements were learned: 1) where the hand moved in a straight path to the object, 2) where an obstacle in the direct path was avoided by moving over it, and 3) where the obstacle was avoided by moving around it.

The movements were recorded using a Polhemus system [10] running at 120Hz for three sensors, one on the upper arm, one on the forearm, and one on the hand (see Figure 3). Each movement type was recorded five times. From the position data $(x(t), y(t), z(t))$ of these sensors, 3-D velocity $v(t)$, acceleration $a(t)$, curvature $k(t)$, and torsion $\tau(t)$ were extracted. Sample time-plots of these measurements are shown in Figure 4.

For these data, the definition of the spatio-temporal patterns is straightforward. At every time point, the patterns consist of three

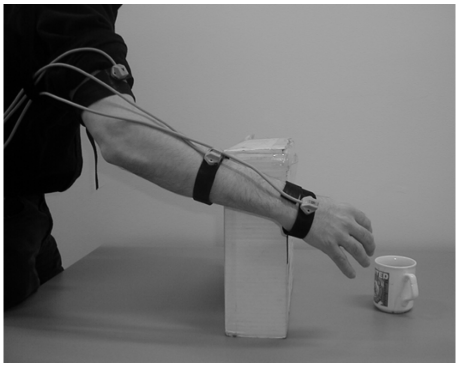

Figure 3. Grasping movement around an obstacle. The movement sensors were placed on the upper arm, the forearm, and the hand.

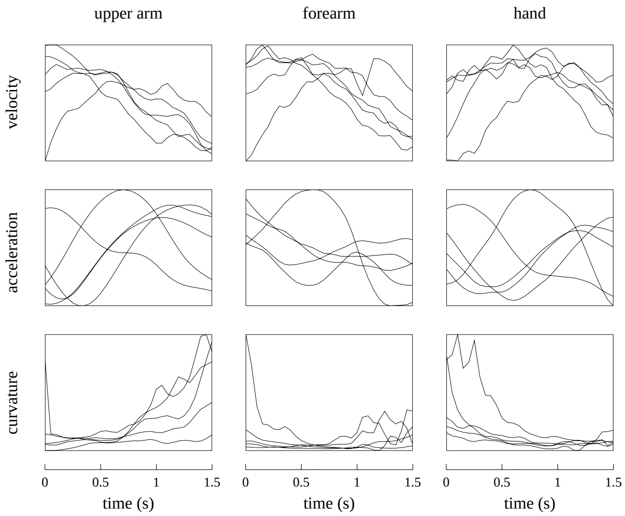

Figure 4. Sample timeplots of the movement sequences illustrated in Figure 3. The left column shows traces for the upper arm, the middle column for the forearm and the right column for the hand. The first row shows time-plot for velocity (for a straight grasp movement), the second for acceleration (for 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.

parts, one for each sensor, each part being described by unary attributes $\vec{u} = [x, y, z, v, a, k, \tau]$. Binary attributes were defined by simple differences, i.e. the spatial attributes were defined as $\vec{b}_s(p_{it}, p_{jt}) = \vec{u}(p_{it}) - \vec{u}(p_{jt})$, and the temporal attributes were defined as $\vec{b}_t(p_{it}, p_{jt+1}) = \vec{u}(p_{jt+1}) - \vec{u}(p_{it})$.

Performance of CRG$_{\mathrm{ST}}$ was tested with a leave-one-out paradigm, i.e. in each run, movement classes were learned using all but one patterns, and the resulting rule system was used to classify the remaining pattern. Pattern learning and pattern classification proceeded exactly as described in Sections 3.2 and 3.3. 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 points 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.

$\vec{u}$	$\vec{b}_s$	$\vec{b}_t$	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%

Table 1. Performance of CRG$_{\mathrm{ST}}$ 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. Dashes indicate that no feature was used. xyz: position in 3D; v: velocity: a: acceleration; k: curvature; τ: torsion.

An example of a classification rule generated by CRG$_{\mathrm{ST}}$ 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 U(t)	$-1.34 \leq V \leq 7.9$ and
	$-2.93 \leq A \leq 1.54$
and T(t,t+1)	$-0.16 \leq \Delta X \leq 0.07$ and
	$-6.51 \leq \Delta Y \leq 5.37$
and U(t+1)	any value
and T(t+1,t+2)	$-5.39 \leq \Delta X \leq 0.08$
	and $-6.51 \leq \Delta Y \leq 5.37$
and U(t+2)	$4.74 \leq V \leq 5.04$ and
	$-.78 \leq A \leq -0.06$
then	this is part of a grasping action
	moving over an obstacle

The results show that $\mathrm{CRG}_{\mathrm{ST}}$ 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.

5 Conclusions

We have considered an extension of a spatial relational learning model to learning of spatio-temporal patterns such as complex human actions and gestures. What differentiates our $\mathrm{CRG}_{\mathrm{ST}}$ approach from models like hidden Markov models is that the rules are capable of generalizing over higher-order spatial and temporal relations. Further, the resultant rule forms are Horn clauses whose structures and lengths are constrained by the general topology of the underlying models and by a Minimum Description Length criterion.

REFERENCES

[1] W. F. Bischof and T. Caelli, ‘Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation’, *Pattern Recognition*, **27**, 1231–1248, (1994).

[2] W. F. Bischof and T. Caelli, ‘Scene understanding by rule evaluation’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 1284–1288, (1997).

[3] *Machine Learning and Image Interpretation*, eds., T. Caelli and W. F. Bischof, Plenum, New York, NY, 1997.

[4] T. Caelli, L. Guan, and W. Wen, ‘Modularity in neural computing’, *Proceedings of the IEEE*, **87**, 1497–1518, (1999).

[5] T. Caelli, A. McCabe, and G. Binsted, ‘On the 3D measurement and representations of human actions’, (2000).

[6] T. Caelli, G. West, M. Robey, and E. Osman, ‘A relational learning method for pattern and object recognition’, *Image and Vision Computing*, **17**, 391–401, (1999).

[7] C. Dillon and T. Caelli, ‘Cite – scene understanding and object recognition’, in *Machine Learning and Image Interpretation*, eds., T. Caelli and W. F. Bischof, 119–187, Plenum, New York, NY, (1997).

[8] B. McCane and T. Caelli, ‘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, Plenum, New York, NY, (1997).

[9] J. R. Quinlan, ‘Mdl and categorical theories (continued)’, in *Proceedings of the 12th International Conference on Machine Learning*, pp. 464–470, (1995).

[10] F. H. Raab, E. B. Blood, T. O. Steiner, and H. R. Jones, ‘Magnetic position and orientation tracking system’, *IEEE Transactions on Aerospace and Electronic Systems*, **AES-15**, 709–, (1979).

[11] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, New York, NY, 1993.

[12] D. H. Wolpert, ‘Stacked generalization’, *Neural Networks*, **5**, 241–259, (1992).