Mathware & Soft Computing 11 (2004) 197-210

A Neuro-Fuzzy System for Sequence Alignment on Two Levels

T. Weyde[1] and K. Dalinghaus[2]
[1] Research Department of Music and Media Technology
[2] Institute of Cognitive Science
University of Osnabrück
e-mail: tweyde@uos.de, kdaling@uos.de

Abstract

The similarity judgement of two sequences is often decomposed in similarity judgements of the sequence events with an alignment process. However, in some domains like speech or music, sequences have an internal structure which is important for intelligent processing like similarity judgements. In an alignment task, this structure can be reflected more appropriately by using two levels instead of aligning event by event. This idea is related to the structural alignment framework by Markman and Gentner. Our aim is to align sequences by modelling the segmenting and matching of groups in an input sequence in relation to a target sequence, detecting variations or errors. This is realised as an integrated process, using a neuro-fuzzy system. The selection of segmentations and alignments is based on fuzzy rules which allow the integration of expert knowledge via feature definitions, rule structure, and rule weights. The rule weights can be optimised effectively with an algorithm adapted from neural networks. Thus, the results from the optimisation process are still interpretable. The system has been implemented and tested successfully in a sample application for the recognition of musical rhythm patterns.

1 Introduction

Former applications on alignment problems often use the dynamic programming approach (DP) to calculate a minimal edit distance (see [9]). In speech recognition, hidden Markov models (HMM) are applied often to alignment tasks. HMM and DP approaches are equivalent [6] and are both based on the assumption that the events within the sequences are context independent. But in many fields this is not the case. Hence, Dedre Gentner and Arthur Markman set up the structural alignment framework [14] for calculating similarity judgements for objects with an internal structure.

In speech and music there are sequences of events, e.g. phonemes, letters, notes, with a strong internal structure: phonemes or letters form words, musical notes form motifs. In order to align such sequences, it is important to recognise relevant groups of events,

i.e. words or motifs need to be separated in a segmentation. The event groups are used as structural units in the alignment process. This implies working on two levels, aligning groups on the *structure level* and aligning events on the *group level.* The approach used here decomposes the task into two parts: a method for generating and selecting alignments of sequences and a rating system to choose the best alignment. The space of possible sequence alignments depends on the segmentation and the alignments on both levels combinatorially, and it grows exponentially with the length of the sequences. To limit this space, we use an algorithmic technique and additional domain specific constraints on the segmentation and the alignments. Depending on the domain, the definition of an appropriate rating function can be quite difficult. Thus, data-based evaluation and optimisation can be very helpful. For this reason, the rating function is realised as a neuro-fuzzy-System, based on fuzzy rules describing the quality of a given alignment. The fuzzy rule set is transformed into a neural network, which can be trained by examples. This neuro-fuzzy system combines the capabilities of fuzzy systems to model human knowledge with the learning abilities of neural networks. The results of this learning process are interpretable as truth values of the fuzzy rules.

The system introduced here has been implemented in an example application for the recognition and comparison of pattern structure in musical rhythms and melodies, the *Integrated Segmentation and Similarity Model* (see [21] and [22]). This domain is a good test case for a combination of expert knowledge and learning from data. Although there is a large body of research on temporal auditory perception, both general and music-specific (e.g. [2] and [13]), a coherent theoretical framework suitable for supporting computer modelling has not yet been established.

2 Related work

The alignment of sequences is important in many domains to judge similarity, e.g. in bioinformatics, linguistics, and music, but often the internal structure of the sequences is not taken into account. Since in linguistics and music the need of more structural information in similarity judgements is stated in the literature, some approaches shall be discussed here.

In the literature on Music Information Retrieval (MIR) it is a common statement that the structure of music is important for the recognition or similarity judgements. The segmentation of a rhythm pattern or a melody is essential to generate its structure. In the following, some approaches from MIR are described.

In [20] Rolland and Ganascia present an approach for the extraction of musical patterns in a melody database. For this task, they derive additional musical descriptions from the basic data. In their approach, the user has to adjust the weights for this additional information by hand. This contrasts with the approach presented in this paper, where the weights are learned by the system. The similarity judgement of the segments is done by dynamic programming. Rolland and Ganascia indicate the segmentation problem as remaining open, it is not clear how this problem is solved in their approach.

In Lartillot's approach [12], patterns are stored as *minimal interval representation* in Abstract Pattern Trie's (APT): Patterns with the same prefix are stored in the same path. The APT is built up by a chronological walk through the piece of music, comparing a new

interval with old ones in the associative memory and old ones already stored in the APT, and thus doing pattern initiating, pattern extending, and pattern confirming.

Widmer looks in *The Musical Expression Project* [25] for explainable and quantifiable principles which govern an expressive performance. He emphasises the need for computational methods, since in his project the data was segmented and aligned by hand.

There are some models for segmentation: The *Local Boundary Detection Model* (LBDM) of Cambouropoulos [3] calculates a boundary profile for a melody using Gestalt-based identity-change and proximity-difference rules; the *Melodic Density Segmentation Model* (MDSM) by Ferrand, Nelson, and Wiggins [7] calculates the accumulated melodic cohesion as the weighted sum of the contributions of all intervals occurring over a period of time. These models operate on one pattern only and are not incorporated in a similarity judgement environment.

Linguistics are another domain where sequence alignment is important. There is the field of phonetic alignment which is required e.g. for comparing dialects, cognate word-forms, or underlying and surface word-forms. Contrary to the music domain, in linguistics the need of structural information on the sequences is not often found in literature. Nerbonne and Heeringa [18] set up a model for calculating the distance between several Dutch dialects. They use the dynamic programming approach in their work, and assume context independence of the phonemes. Kondrak [11] supports this view and emphasises the definition of good cost functions for the comparison of phonemes. In contrast, Hahn [10] argues that the contextual structure within words is essential for comparing word sounds. Hahn criticises a lack of empirical studies in this field and presents her own experiment. There, she gives examples for which the dynamic programming approach is not sufficient.

3 Representation of Sequence Alignments

The representation of a structural alignment of two sequences as mentioned above involves three structures: Segmentations of the sequences, an alignment of the *Segments*[1], and alignments of events or *Elements*[2] in aligned segments. We call the combination of segmentations and alignments on both levels an *Interpretation* of the sequences. Features are calculated from an interpretation, which rate qualities of the segmentations and the alignments. These qualities result from modelling domain knowledge and are based on several basic features of the sequence and the sequence elements. Thus, these features are called *complex features*. The neuro-fuzzy system used for rating the interpretations operates on the complex feature values and not on the sequences or the interpretation itself.

An interpretation is based on the sequences $P = (x_1,\ldots,x_n)$ and $Q = (y_1,\ldots,y_m)$ consisting of elements x_i, y_j. The segmentations $P_{seg} = (P_1,\ldots,P_k)$ and $Q_{seg} = (Q_1,\ldots,Q_l)$ divide P and Q into contiguous segments so that every element belongs exactly to one segment. From the elements x_i, y_j of the sequences P and Q and the segmentations P_{seg} and Q_{seg}, the complex features, which model aspects of the segmentation, are calculated using a *Complex Feature Function for Segmentation* C_s, as is illustrated in figure 1. The vector $\tilde{P}_i$ consists of all complex features for the group P_i and has a fixed length. Consequently, it

[1]Note that the term segment denotes a group of elements in contrast with phonology where a segment is a indivisible part of a sequence.

[2]In the following more formal description the term element is used for the indivisible part of a sequence

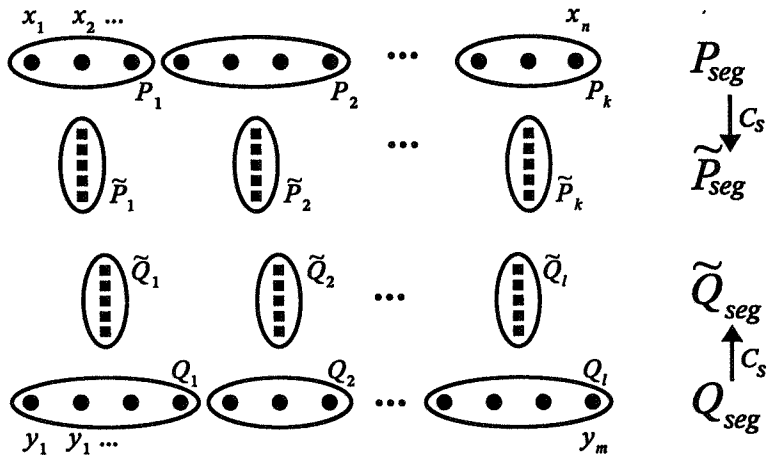

Figure 1: Segmentation features.

is independent of the length of the group P_i, which is important for the rating system. The collections $\tilde{P}_{seg}$ and $\tilde{Q}_{seg}$ consist of equal-dimensional vectors of the complex features for every group of the corresponding sequence.

An *Alignment* is a relation $A \in \wp(I \times J)$ where I and J are the index-sets of the elements of the two sequences to be aligned. We refer to the alignment of the segmentations P_{seg} and Q_{seg} as the alignment on the structure level A_s with $I = \{1,\ldots,k\}$ and $J = \{1,\ldots,l\}$, i.e. the sequences P_{seg} and Q_{seg} consist of the segmentations of P and Q: If $(i,j) \in A_s$, then group P_i is aligned to group Q_j. For every pair P_i, Q_j with $(i,j) \in A_s$, we refer to the alignment on the group level as $A_g^{(i,j)} \in \wp(I_i \times J_j)$ where I_i and J_j are the index-sets of the groups P_i and Q_j.[3]

Corresponding to the segmentation features, there are features calculated for the alignments on group and on structure level. Figure 2 schematically shows alignments and features on the group level: The relation $A_g^{(k,l)}$ describes the alignment of the elements of the groups P_k and Q_l and the *Complex Feature Function for Groups* C_g calculates the complex features for this alignment. The resulting vector has a unit length which is again independent from the number of elements in the aligned groups. For more detailed information see [4].

4 Feature Definitions

The definition of features is, of course, domain dependent and represents the meaning of similarity in the domain. Here is a clear distinction between the general algorithm and our exemplary special application, which is the comparison of musical rhythms.

The approach of features being calculated from the interpretation rather than from the sequences alone offers the benefit of yielding information on the alignment which can

[3]Note that the index-sets I_i and J_j correspond to elements of the sequences whereas the index-sets I and J correspond to segments of the sequences

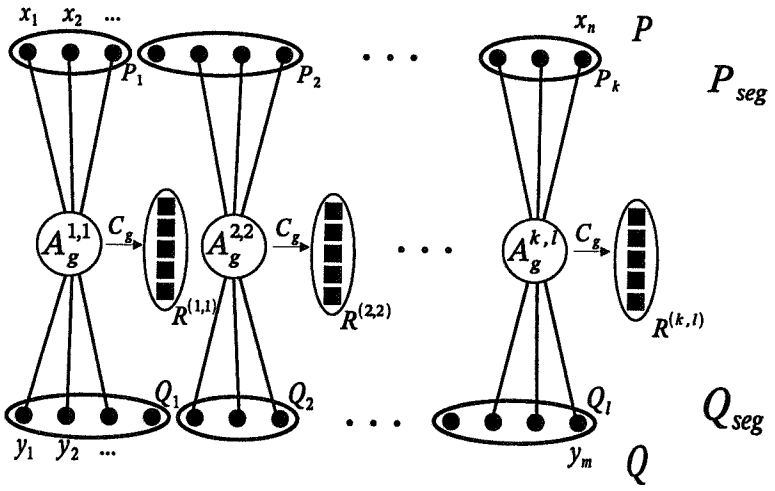

Figure 2: Alignment features on group level.

be used in applications. Such information includes for instance differences in intensity, temporal position, and duration, which can be used as basis for user a feedback, e.g. in a tutorial application. In the case of musical rhythms, the sequences consist of note events based on MIDI[4] data. Three values of the note events are used as data for the system: onset time, duration, and loudness. The features calculated from aligned rhythm sequences are related to the quality of the segmentations, to the similarity of the assigned groups, and to the structural relations of these groups such as changes in the tempo or the relative position of groups. We cannot describe all features here in detail, but only give some examples; for a full description see [23] and [21]).

The feature definitions are based on results from music theory and music psychology. These are mostly scalar values, which need to be fuzzyfied before feeding them into the system. For example, the fuzzy comparison of numbers x and m is realised using a Gaussian function with a parameter s:

$$gauss(x,m,s) = e^{-\frac{(x-m)^2}{2s^2}}. \tag{1}$$

The function *gauss* is not normalised in respect to the integral, because for $x = m$ the result should be 1. For the tempo of an input group in relation to the aligned model group, we need to rate the tempo ratio, not the tempo difference. Thus, a fuzzy set for an equality relation, *equal*, is defined, which uses *gauss* as membership function to give a symmetric rating on a logarithmic scale by using the average of the arguments and an additional factor q for the scaling on the x-axis:

$$equal(x,y) = gauss(x,y,q \cdot \frac{x+y}{2}). \tag{2}$$

equal is symmetric on a log-scale as illustrated in figure 3 and commutative as it can be expected from an equality measure (see [23]). It is also possible to transform the values

[4]Musical Instrument Digital Interface

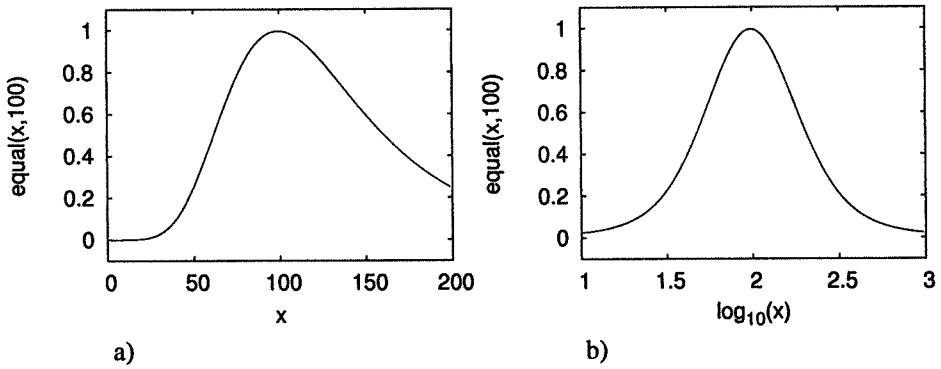

Figure 3: Fuzzy set $equal(x,y)$ modelling the equality relation for fixed $y = 100$ plotted with a) linear and b) logarithmic scaling of the x-axis.

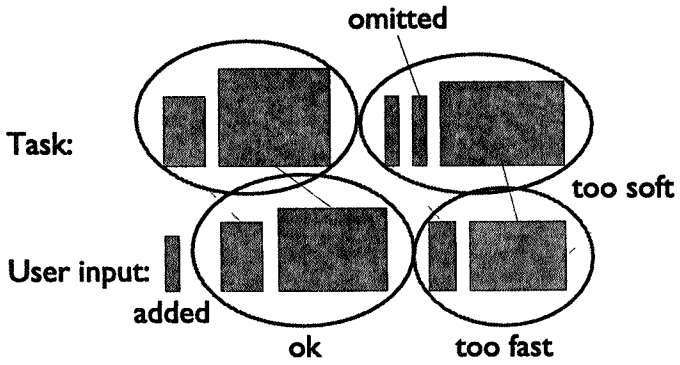

Figure 4: Alignment of a musical rhythm.

to a logarithmic scale and to apply a normal Gussian, but this solution is more efficient. Another feature measures the difference in onset time of aligned notes. This value is measured relative to an offset and a scaling factor, which is calculated for a complete group, to distinguish between the global differences in position and tempo and the local distortions of the segment's structure.

The values of the defined features are not only used for finding the best interpretation, but they also provide useful information about the musical relations between the two sequences. These musical relations are useful in several applications, such as the following:

- Music tutorials, where detailed information about the differences of a user's performance from the given task can be used for interaction
- Search for plagiarism in copyright conflicts, where finding similarities and differences of melodies can be automated
- Analysis of music performances in relation to written scores

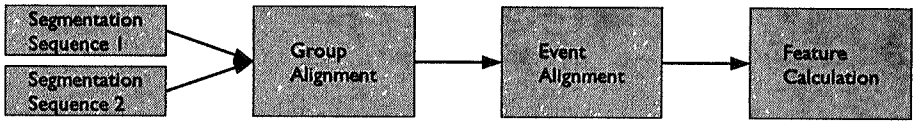

Figure 5: Preprocessing.

In a tutorial application, the result can be presented to the user as shown in figure 4. Here, information on the matching of the groups is displayed graphically, with added information on the omitted notes and additional information like tempo (too slow or too fast) and loudness (too loud or too soft) on the group and the note level. In figure 4, each box represents one note of a sequence with its length (width) and loudness (height). The ellipses represent the segmentation of the sequences, and the connections between the boxes represent the alignment of the notes within aligned groups.

5 Preprocessing

The preprocessing is carried out in four steps as sketched in figure 5. First, the segmentations for both sequences are created. Then, the alignment on the structure level is created, i.e. the groups are aligned. In the next step, the alignment on the group level is created, i.e. the events within the groups are being aligned. Based on the complete alignment, complex features are calculated which can be used for rating. These steps are performed for each possible combination of segmentations and alignments on structure and group level. Since the combination of all segmentations and alignments leads to a large number of alternatives, it is necessary to restrict the generation of segmentations and alignments. This is achieved by using a branch-and-bound approach and domain specific constraints.

The system efficiency can be enhanced by incremental generation of alignments and avoiding unnecessary evaluations using a branch-and-bound scheme. The effect of branch-and-bound largely depends on the order of evaluation. In our application the segmentations are calculated first and they are sorted according to their quality, beginning the incremental generation of alignments with the best segmentations. This turned out to be a good heuristic for the branch-and-bound scheme. The average calculation time on the test set decreased from 1,021.09 to 8.34 seconds for the usage of the branch-and-bound-scheme.

Additionally, constraints are used on each level of generating interpretations, defined as *Filter Functions F*, to prevent undesired processing. The recognition of rhythmic pattern structure can serve as an example: Music theory and music psychology have studied some properties of the perception and cognition of musical rhythm, on which the constraints and features are based. There are constraints applied on the segmentation which restrict the number of events in a group to a value of 5 (see [15]). The temporal duration of groups can be restricted to 2 seconds (see [8]). Also, group boundaries must not grossly contradict grouping by proximity (see [5]). Another constraint restricts the occurrence of groups containing only one event (see [13]). On the structure level, the alignment of patterns with different numbers of events is restricted. The application of these constraints as filter functions results in a drastic reduction of calculation time. The average calculation time on a test set decreased from 1,021.09 to 1.41 seconds due to the usage of the filters.

When using branch-and-bound and filters together, the calculation time decreased to 0.32 seconds. For more detailed information on the reduction of the complexity see [23] and [24]. These constraints need to be designed very carefully, because they do not only have a great impact on the efficiency of the whole process but they can also potentially prevent good interpretations from being evaluated. Hence, they have to be as restrictive as possible but as permissive as necessary not to reject appropriate solutions.

6 Neuro-Fuzzy Modelling of a Rating Function

Based on the calculated features, expert knowledge about how to rate an interpretation can be expressed in a fuzzy rule set. In the case of musical rhythms, typical expert statements name factors which contribute to the perception and cognition of certain musical phenomena, e.g. a loud note tends to start a new group. These rules are usually not quantified, but they can have degrees of strength, e.g. a long temporal interval between note onsets strongly indicates a group boundary. This kind of expert knowledge can be modelled appropriately in fuzzy logic. The following rules are a subset of those used in our application:

$$
\begin{aligned}
\mathrm{SQual}(ip) &\leftarrow \mathrm{STpoQual}(ip) \wedge \mathrm{SPrcsn}(ip) \wedge \\
&\quad \mathrm{SCorrect}(ip) \wedge \mathrm{SPosition}(ip) \\
\mathrm{STpoQual}(ip) &\leftarrow \mathrm{GTpoQual}(ga_1) \wedge \ldots \wedge \mathrm{GTpoQual}(ga_n), \\
&\quad \text{where } ip = [ga_1, \ldots, ga_n] \\
\mathrm{GTpoQual}(ga_i) &\leftarrow \mathrm{GTpoStbl}(ga_i) \wedge \mathrm{GTpoPlsbl}(ga_i) \\
\mathrm{SPrcsn}(ip) &\leftarrow \mathrm{GPrcsn}(ga_1) \wedge \ldots \wedge \mathrm{GPrcsn}(ga_n), \\
&\quad \text{where } ip = [ga_1, \ldots, ga_n] \\
\mathrm{SCorrect}(ip) &\leftarrow \mathrm{GCorrect}(ga_1) \wedge \ldots \wedge \mathrm{GCorrect}(ga_n), \\
&\quad \text{where } ip = [ga_1, \ldots, ga_n]
\end{aligned}
$$

The first rule states that the quality of an interpretation (*ip*) on the structure level (*SQual*) depends on the tempo quality (*STpoQual*), the precision (*SPrcsn*), the correctness (*SCorrect*), and the positioning of *ip* (*SPosition*). The second, fourth, and fifth rule deal with a variable number of premises, we call them *multi-rules*. This is necessary for the transition from the the structure level (*ip*) to the group level (*ga*), because the number of groups in a segmentation varies. The second rule states that the quality of the tempo of a sequence alignment (*ip*) on the structure level (*STpoQual*) depends on the quality of the tempo of every pair of aligned groups (ga_i) within the sequence alignment (*GTpoQual*). The third rule states that the tempo quality of an aligned pair of groups (*GTpoQual*(ga_i)) depends on the tempo stability of the group alignment (*GTpoStbl*(ga_i), change of the tempo) and the plausibility of the tempo (*GTpoPlsbl*(ga_i)). The fourth rule describes the transition from the structure level to the group level for the precision (*SPrcsn*, *GPrcsn*) which rates whether the notes were played at the right time with the right loudness. The

fifth rule describes the transition from the structure level to the group level for the correctness (*SCorrect*, *GCorrect*) which means whether notes were marked as added or omitted in the interpretation.

Often t-norms and t-conorms like $\top_{min}$ and $\bot_{min}$ are used to calculate the truth value of logical expressions in fuzzy logic (see [16]). But they do not allow compensation of their arguments, i.e. that a low value in one argument can be compensated with a high value in another, which is required for plausible modelling of rhythmic perception. Also, they are not very suitable to work with a variable number of arguments. Basically, the result of an operator μ should be independent from the number of arguments, which is the motivation for two multi-operator-criteria (MOC). If multiple arguments have the same value, the result should be independent of the number of arguments:
MOC 1:

$$\mu_{\otimes} \underbrace{(\nu,\ldots,\nu)}_{n-times} = \mu_{\otimes} \underbrace{(\nu,\ldots,\nu)}_{m-times}.$$

If a pattern is repeated, the number of repetitions also should not affect the result.
MOC 2:

$$\mu_{\otimes}\underbrace{(\nu_1,\ldots,\nu_k,\cdots,\nu_1,\ldots,\nu_k)}_{n-times} = \mu_{\otimes}\underbrace{(\nu_1,\ldots,\nu_k,\cdots,\nu_1,\ldots,\nu_k)}_{m-times}.$$

Following this criteria, a new operator, the *q-operator*, is defined for this system:

$$\mu_{\otimes,q}(\alpha_1,\ldots,\alpha_n) = \left(\frac{1}{n}\sum_{i=1}^{n}\alpha_i^q\right)^{\frac{1}{q}} \qquad q>0.$$

The q-operator satisfies both, MOC 1 and MOC 2 (see [23]), and it is continuous, hence the MOCs are not singularities but ensure some stability among changing numbers of arguments, and it allows compensation between the operands. In our sample application, values of $q=\frac{1}{2}$ for the conjunction and $q=2$ for the disjunction proved to be useful.

The q-operator is related to the Yager operator (see [26]), it converges to $\top_{min}$ in the limit of $q\to 0$ and to $\bot_{min}$ in the limit of $q\to\infty$. It also calculates the average over the arguments for $q\to 1$. It is not a t-norm because it violates the requirement of identity and associativity. Nevertheless, it can be used in our neuro-fuzzy system where only differentiability of the operator is necessary.

Based on a method presented by Nauck et al. in [17], a fuzzy-logical program can be transformed into a neural network under certain conditions. This is achieved by generating a neuron for every atomic expression. For every rule, connections from the premises to the conclusion are inserted into the network. The weights created in the network correspond to the truth values of the fuzzy logic program and are therefore interpretable after training. The rules for moving from the structure level to the group level are handled as special case. For the premises in this rule, a so-called *multi-neuron* is generated which has as many instances as group alignments exist. Each instance of this multi-neuron is connected to the neuron corresponding to the conclusion, and all connections share one weight. This means that the network can change its topology according to the length and segmentation

of the sequences. The neurons do not calculate their input by using a weighted sum but by using the fuzzy operator of the corresponding rules instead.

For the training of the network, the standard backpropagation algorithm has to be modified, using the derivative of the input functions. The modified backpropagation rule, which is obtained as gradient descent on the sum of the squared error on a given training set, can be written as

$$\triangle w_{ij} = \eta \sum_{p \in P} o_i^{(p)} \cdot drvt_{ij}^{(p)} \cdot \delta_j^{(p)} \tag{3}$$

with

$$\delta_j^{(p)} = \begin{cases} f'(net_j^{(p)})(t_j^{(p)} - o_j^{(p)}), & \text{if } j \text{ is outp. neur.,} \\ f'(net_j^{(p)}) \sum_{s=1}^{m} \delta_{k_s}^{(p)} \cdot w_{jk_s} \cdot drvt_{jk_s}^{(p)}, & \text{else,} \end{cases} \tag{4}$$

where $drvt_{ij}^{(p)}$ is the derivative of the used input function for neuron j given input pattern p, $o_i^{(p)}$ is the output of neuron i for pattern p, $t_j^{(p)}$ is the target value for pattern p and neuron j, w_{ij} denotes the weight from i to j in the network, and f denotes the activation function of the neuron. The weights at the connections coming from the instances of one multi-neuron use weight-sharing. The network is trained with a more efficient modification of a simple gradient descent, the *Resilient Backpropagation* algorithm (see [19]).

7 Learning by Examples

The training process requires examples, i.e. interpretations: alignments of pairs of sequences together with a given rating. It is very difficult for an expert to choose an exact value for interpretations in a consistent manner over many examples. But if we have two given interpretations, it is easy to determine which one should be rated higher. Thus, a training approach is used which is based on relative examples shown in figure 6 (see [1]). Here a training example for the neural network consists of an ordered pair of two differ-

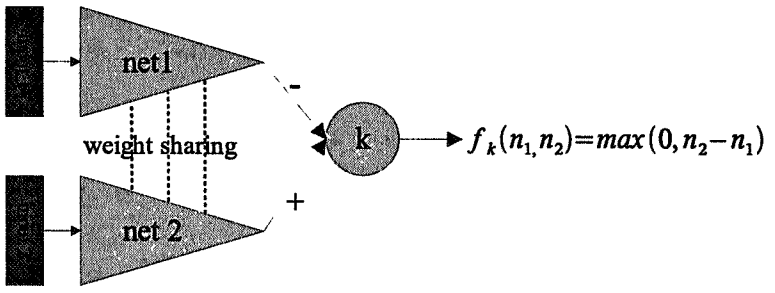

Figure 6: Training with relative examples.

ent interpretations of the same two sequences, where the first one is rated higher by the expert. To handle this, the network is duplicated using weight-sharing, and a comparator neuron k connects the output of the two networks. The interpretation rated higher is always

used as *input 1*, and therefore the target value for the comparator neuron output is 0. The semi-linear activation function of the comparator neuron can be replaced by a sigmoid.

Comparing is easier for experts than absolute rating, but due to combinatorial growth it is not feasible to let an expert rate every possible interpretation. This problem is solved by using *Iterative Training* (see [23]) which dynamically generates relative examples using a single interpretation provided by an expert. First the expert selects the best interpretation for every example, then the interpretation currently rated best by the system is calculated. If it differs from the interpretation provided by the expert, a relative training example is created. Then the network is trained. This process is iterated, checking the highest rated interpretations and training the network again. This is necessary because even if the training has been completely successful, there can be other interpretations, now rated highest by the system, but again differing from the expert's interpretation.

8 Results

For training the system, 100 samples were used consisting of 50 performances by students, which were used in original form and with noise added. The expert interpretations were defined by graduate students. The samples were divided randomly into a training set and a test set of 50 samples each. After training, 80% of the samples were processed correctly on the test set. Here, correct means that the system chooses the same interpretation as the expert. But even if the system output is not the expected interpretation, it can nevertheless be musically acceptable. This was the case for another 17% of the samples. Overall, 97% of the interpretations chosen by the system trained with the neuro-fuzzy approach are musically acceptable, which is a good result.

The fuzzy truth values for the trained system show some tendencies which were consistent through the use of different training parameters and error criteria in the backpropagation (see [23]). The values show consistently higher weights for group similarity ratings than for segmentation, which seems plausible for an alignment task. Another tendency is that the structural differences receive smaller weights than temporal precision. This was not clear in advance, since removing or inserting events is not a small change in musical structure. Training results show higher weights for timing than for loudness concerning segmentation which agrees with Deutsch's statement (see [5]) that timing is more influential than loudness for segmentation. A problem in interpreting these training results is that the weights, resp. the fuzzy truth values, resulting from training depend on the scaling of the feature input which can be of different dimensions, e.g. the truth values depend on whether time is measured in seconds or milliseconds. For drawing conclusions, the scaling needs to be appropriate to the domain. Also, the system is non-linear, which changes the influence of individual values on the output, depending on the context. This adds calculatory power to the system, but nevertheless it has to be taken into account when the results of the training are interpreted.

9 Conclusions

A method has been proposed for calculating alignments and similarity rating of sequences by analyzing them on two levels, utilising the internal structure of the sequence. By using fuzzy modelling of knowledge in combination with machine learning, the task can be solved without a complete or exact model of relevant processes, e.g. perception and cognition of speech and music. For the recognition of musical rhythm patterns, a system was implemented which showed encouraging results.

The system can be clearly divided into a general part of generating the interpretations and a domain specific part modelling the features and the filters from the expert knowledge. Thus, the system is sufficiently general to be transferred to other fields of application, e.g. the alignment of phoneme sequences. Potential areas of application are those with a clear segmentation and alignment problem in similarity judgements where the incorporation of structural knowledge about the segments is useful. Subjects of future research include improvements in the general method and new applications. This includes especially experiments with different learning methods. improved interpretation of the results, and the automation of rule acquisition. Potential areas of application are phoneme/grapheme alignment, bioinformatics, and temporal analysis of technical processes.

References

[1] H. Braun, J. Feulner and V. Ulrich. Learning strategies for solving the planning problem using backpropagation. In *Proceedings of NEURO-Nimes 91, 4th International Confenrence on Neural Networks and their Applications*, 1991.

[2] A. S. Bregman. *Auditory scene analysis: The perceptual organization of sound.* The MIT Press, Cambridge, Mass., 1990.

[3] E. Cambouropoulos. The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference (ICMC'2001)*, Havana, Cuba, 2001.

[4] K. Dalinghaus and T. Weyde. Structure recognition on sequences with a neuro-fuzzy system. In *Proceedings of the 3rd International Conference in Fuzzy Logic and Technology (EUSFLAT 2003)*, pages 386–391. European Scociety for Fuzzy-Logic and Technology, 2003.

[5] D. Deutsch. Grouping mechanisms in music. In Diana Deutsch, editor, *The Psychology of Music*, chapter 4, pages 99–134. Academic Press, New York, 1982.

[6] R. Durbin, S. R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

[7] M. Ferrand, P. Ǹelson and G. Wiggins. Memory and melodic density: A model for melody segmentation. In F. Giomi N. Bernardini and N. Giosmin, editors, *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, pages 95–98, Florence, Italy, 2003.

[8] P. Fraisse. Time and rhythm perception. In E. C. Carterette and M. P. Friedman, editors, *Perceptual Coding*, Handbook of Perception, pages 203–247. Academic Press, New York, 1978.

[9] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Univ. Press, Camebridge, 1999.

[10] U. Hahn and T. M. Bailey. What makes words sound similar. *Cognition*, 2004. to appear.

[11] G. Kondrak. Phonetic alignment and similarity. *Computers and the Humanities*, 37(3):273–291, 2003.

[12] O. Lartillot. Generalized musical pattern discovery by analogy from local viewpoints. In Steffen Lange, Ken Satoh, and Carl H. Smith, editors, *Proceedings of Discovery Science 2002*, number 2534 in LNAI, pages 382–389, Heidelberg, 2002. Springer-Verlag.

[13] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Camebridge, Mass., 1983.

[14] A. B. Markman. Structural alignment, similarity, and the internal structure of category representations. In Ulrike Hahn and Michael Ramscar, editors, *Similarity and Categorization*, chapter 7, pages 109–130. Oxford University Press, Oxford, UK, 2001.

[15] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.

[16] D. Nauck. Neuro–fuzzy systems: Review and prospects. In *Proceedings of Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97)*, pages 1044–1053, 1997.

[17] D. Nauck, F. Klawonn and R. Kruse. *Foundations of Neuro-Fuzzy Systems*. Wiley, Chichester, 1997.

[18] J. Nerbonne and W. Heeringa. Computational comparison and classification of dialects. *Dialectologia et Geolinguistica*, 9:69–83, 2001.

[19] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *ICNN-93: IEEE International Conference of Neural Networks*, pages 586–591, San Francisco, CA, 1993.

[20] P.-Y. Rolland and J.-G. Ganascia. Pattern detection and discovery: The case of music data mining. In David J. Hand, Niall M. Adams, and Richard J. Bolton, editors, *Proceedings of the ESF Explanatory Workshop Pattern Discovery and Detection, London*, volume 2447 of *Lecture Notes in Artificial Intelligence*, pages 190–198. Springer-Verlag, September 2002.

[21] T. Weyde und K. Dalinghaus. Recognition of musical rhythm patterns based on a neuro-fuzzy-system. In Chihan H. Dagli et. al., editor, *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Complex Systems*, volume 11, pages 679–684, New York, NY, 2001. ASME press.

[22] T. Weyde. Integrating segmentation and similarity in melodic analysis. In K. Stevens, D. Burnham, G. McPherson, E. Schubert, and J. Renwick, editors, *Proceedings of the International Conference on Music Perception and Cognition 2002*, pages 240–243, Sydney, Australia, 2002. University of New South Wales.

[23] T. Weyde. *Lern- und wissensbasierte Analyse von Rhythmen.* epOs, Osnabrück, 2003.

[24] T. Weyde and K. Dalinghaus. Design and Optimization of Neuro-Fuzzy-Based Recognition of Musical Rhythm Patterns. *International Journal of Smart Engineering System Design*, 5(2):67–79, 2003.

[25] G. Widmer. The musical expression project: A challenge for machine learning and knowledge discovery. In Luc de Raedt and Peter Flach, editors, *Proceedings of the 12th European Conference on Machine Learning (EMCL 2001), Freiburg, Germany*, volume 2167 of *Lecture Notes in Computer Science*, pages 603–614, Heidelberg, 2001. Springer-Verlag.

[26] R. R. Yager. On a general class of fuzzy connectives. *Fuzzy Sets and Systems*, 4:235–242, 1980.