CHAPTER 1

DATA DRIVEN THERAPY DECISION SUPPORT SYSTEM

Witold Jacak, Karin Proell
Upper Austria University of Applied Sciences
Faculty of Informatics, Communications and Media Hagenberg
4232 Hagenberg, Austria

Therapy modeling and planning are important components for optimal and cost-effective patient care. Therapeutic response of the individual patient not only relies on the selection of effective drugs but is also heavily influenced by appropriate drug dosage. A variety of intelligent techniques have been initiated to support physicians in deciding an optimal treatment for an individual patient.

In recent years, decision analysis techniques are increasingly being applied to model and analyze dynamic decision problems in medicine [1, 2]. Dynamic decision analysis and modeling frameworks are based on structural and semantically extensions of conventional decision models e.g. decision trees and influence diagrams, with the mathematical definitions of finite state Markov Stochastic Processes [3, 4]. Most approaches use Markov Decision Processes (MDP) to describe and solve decision problems in which the optimal choice has to be revised periodically in accordance with the evolution of the patient's conditions [1, 3, 4]. Unfortunately the adoption of MDP to model complex systems as medical decision problems is hampered by the difficulty in knowledge elicitation from a specific domain. In particular the traditional formulation of a MDP [4, 5] through its transition matrix imposes to specify a great number of parameters, whose meaning is not always understood promptly, and it does not allow us to represent explicitly the structured knowledge underlying the model.

Therefore a clinician relies on his knowledge of fundamental physiologic and pathologic processes to develop diagnostic methods and procedures and to investigate the effects of drugs and new treatments on real clinical cases. The clinician must have a precise representation of the clinical state of the patient and of relevant physiologic processes ongoing in the patient's body.

Representations of those cognitive structures are based on clinical observations, case records and available empirical data.

A further commonly used formalism for knowledge representation is a *semantic network* with a *graph-grammar* approach to manage the complex graph transformations driven by information entries (patient data) and medical problem solving (Such a system is applied in Children's University Hospital Mainz, Germany, for therapy planning in pediatric oncology) [6]. Similarly to MPD approaches the complexity of semantic network systems grows with the increase of available patient data.

This chapter focuses on a multilevel approach for constructing a data evidence based model for classification of different therapies and their effectiveness for clinical treatment. We present methods for aligning patient records, a mapping to clusters based on preprocessed sequences of critical events of patient treatment and algorithms for therapy planning support. Having clusters of patient records medication profiles can be derived. A method for therapy decision support is based on such profiles to determine a sequence of ordered steps of possible medication for a given patient state. The decision system is described in section 1.3.

Therapy planning can also be seen as a learning problem. In [7, 8] an extension to the Q-Learning algorithm is used to incorporate existing clinical expertise into the process of acquiring an appropriate administration strategy of rHuEPO to patients with anemia.

In order to synthesize the Q-Learning agent we need a patient state generalization function to provide generalized patient states as categories of patient observation vectors, which is presented in Section 1.2. For classification of medications we introduce a medication generalization function based on similarity classes of medications and a similarity function between two drug dosages. Both generalization functions are used for generalizing patient trials and the life long quality function. In section 1.4 the Q-Learning Agent is presented.

In the first phase, a sequence of events called *patient trial* will be extracted from computer patient records (CPR). These events describe only one flow of therapy of a concrete disease. Each event is represented as a pair *(state, time)*. The *state* does not only contain standard numeric parameters but can be extended with images (MR, RT, or photo) and text based linguistic descriptions. Based on such *state* we introduce the measure between states of different patients. We assume that each patient's state is represented in the global state space. Based on the measure the system calculates the best alignment between different patient trials. The alignment measure (score) calculates the distance between two sequences of patient states, which represents the similarity of flow of therapy [14].

This procedure is applied to each pair of patient trials stored in the Hospital Information Systems (HIS) concerning similar diseases. Based on the value of similarity a semantic network is constructed and divided into full-connected partitions. Each of these partitions represents a class of similar therapy and can be used for computer-aided decision-making. The clustering can be extended by integrating biomedical information such as gene expression data of microarray date for those patient sets.

1.1 Patient Record and Patient State Generalization

Patient Record Data based Patient State Distance

On the patient level of knowledge base the data from patient records should be preprocessed to obtain a compact representation of course of disease. Normally we have different sources of medical information concerning a couple of numerical data representing laboratory test results, RT images or linguistic text describing diagnosis and therapy. The general patient record can be represented as *critical event set* described by the *observation protocols set*

$$Patient\ Record = \{Critical\ Event_j \mid j=1,..,n\} \qquad (1.1)$$

The information in form of *critical events* is the notification that assists in ensuring compliance with practice guidelines and medication protocols. The clinical documentation contains a great number of medical parameters from which structured knowledge about patient state must be derived.

The clinician must have a precise representation of the clinical state of the patient and of the relevant physiologic processes ongoing in the patient. Each state represents a variable obtained from the factorization of the parameter space. Different methods for factorization (based on fuzzy sets) can be found in [9].

Each *critical event e* is represented as a pair *(state, time).* The state does not only contain standard numeric laboratory test parameters *v* but also text based linguistic description of observations (observation protocol).

$$Critical\ Event = e = (state, time) = (v, Observation\ Protocol), \tau)\} \qquad (1.2)$$

where v is vector of test parameters, and τ is the time-interval between initial moment t_0 and current event's moment t, ($\tau = t - t_0$).

Representation of such a cognitive structure as general patient state is based on clinical observations, case records and available empirical data. The classic method of state construction is the structured interview. Such interviews contain clinical findings and disease states. The interview results are used to get a formal observation protocol. The diagnostic checklists – structured interview - provide an automated method for assuring documentation of key symptoms and behavioral issues. Critical events allow clinicians to document the reason for deviating from standard treatment [10, 11, 12, 13, 15]. In psychopharmacology, for example, there are few procedures and laboratory tests that definitively establish a psychiatric observation. The International Classification of Diseases (ICD) and DMS provide standard criteria on which to base the observation protocol. The role of theoretical representations has been verified experimentally by several studies [12, 13].

In pediatric oncology for example, protocols typically cover time periods including initial diagnostics and assignment to different risk groups and therapy branches, intensive inpatient chemo- and radiotherapy phases for tumor remission, consolidation therapy, and outpatient long-term therapy to avoid tumor reoccurrence [6]. In this domain the observation protocol represents time-tables of chemo- and radiotherapy, long-term sequences and stratifications of therapy phases, and temporal constraints concerning the duration of therapy-intensive and therapy-free intervals.

In the first phase the patient record should be transformed into a sequence of critical events *e* representing the time course of a patient healing process called patient trials.

$$Trial=(m, (e_1,\ldots,e_n)) =(medication, (state_1, time_1),..,(state_n, time_n)) \} \qquad (1.3)$$

Each change of treatment or medications *m* establishes the new patient's trial of disease. Critical events describe only one flow of treatment for a concrete treatment of disease and medication. In order to compare different trials it is necessary to introduce a formal description of states to allow the calculation of distance or similarity between two states. It is obvious that a vector of numerical data $(v_j \mid j=1,..,n)$ is easy to compare. To make it possible to measure the similarity between images it is sometimes useful to map images to 3D models of the human body which can be used to obtain additional numerical data $(w_j \mid j=1,..,k)$ derived on various image contents. Such methods can be useful in treatment of dermatological diseases. Based on 3D models we cannot only calculate various geometrical parameters as for example field, contour length or shape but we also can automatically perform a classification of infected anatomical regions and a coding of

disease (Fig. 1.1.). Additionally we assume that diagnosis will be transformed into standard code for example ICD-10.

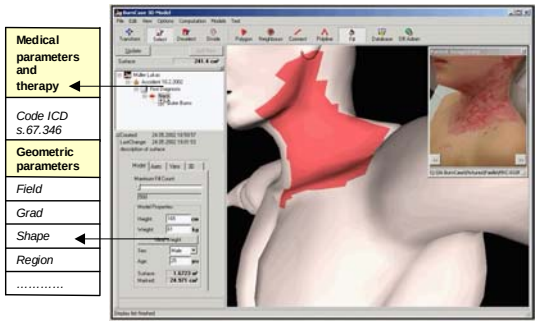

Fig.1.1 *Case-dependent Patient State containing numerical data and 3D-model data*

After these preprocessing steps we can represent the state as a vector of numerical parameters and codes.

$$e^i = (state^i, time^i) = (((v^i_j \,|j=1,..,n), (w^i_j \,|j=1,..,k), diag\_code^i), time^i) \} \qquad (1.4)$$

Based on such *state* we introduce the measure of distance $\rho : S \times S \rightarrow R^+$ between states of different patient (see Fig.1.2).

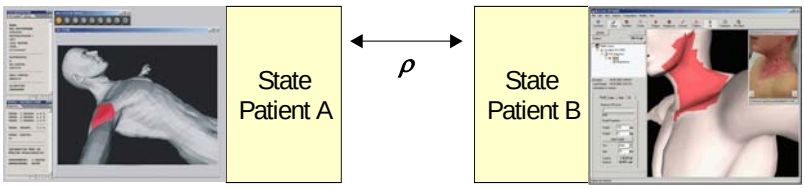

Fig.1.2 *The local measure between states of different patient.*

We assume that each patient's state is represented in this global state space *S*. This distance represents the similarity of two states form different trials of the course of disease. An example of a patient trial for a dermatological disease is presented in Fig. 1.3.

Fig. 1.3 *Example of a patient trial*

The images above are extracted from a concrete patient trial showing the healing process of a dermatological case in the neck region. Each state is based on a 3D model of the affected part of the body and a set of numeric and coding parameters representing results of assessments or treatment protocols. The 3D Model and the parameters are used to build the state vector at each point of time, which are used to calculate the measure between states of treatment in different patient trials.

Each change of treatment or medications establishes the new patient's trial of disease T. It means that there exists the function

$$M: Trials \rightarrow Medications \qquad (1.5)$$

such that $M(trial)=m$. The patient trials are shown in Fig. 1.4.

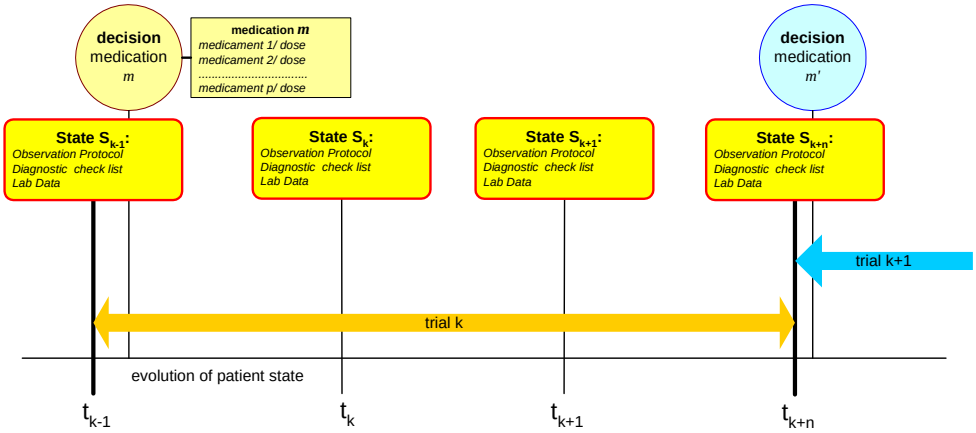

Fig. 1.4*. Patient trials*

The distance function $\rho : S \times S \rightarrow R^{+}$ between states of different patient can be used to analyze similarity of patient state and for computing the distance function between different trials.

Other method for generalization of different patient states uses the self-mapping neural network to construct the *similar (generalized) state* from different patient states [16].

Patient State Generalization with Self-Mapping Neural Network

One of the main components of the therapy planning system is the module for building a function that gives generalized states from clinical patient records like patient data, clinical guidelines, outcome laboratory measures etc. Representation of such a cognitive structure as *generalized patient state* is based on clinical observations, case records and available empirical data.
We establish the generalized state function

$$\boldsymbol{gen_{State}}\textbf{:}\ \textit{Patient States} \rightarrow \textit{Generalized States Space} \qquad (1.6)$$

The only information available is the *k*-dimensional vector *v* of test parameters and factorized observations protocols $\boldsymbol{v} = (v, \textit{Observation Protocol})$. Our goal is thus to construct a state generalization function to provide generalized patient states as categories of patient observation vectors. This leads to clusters of observation vectors with vectors in the same cluster being mapped to the same generalized patient state by the function ***gen***. The function $\boldsymbol{gen_{State}}$ that produces generalized states from observation vectors can be implemented as a neural network, so that new states can be incorporated to the implementation of $\boldsymbol{gen_{State}}$ by an unsupervised learning process [16]. The problem of obtaining generalized states from patient data to preserve proximity information among the observation vectors can be solved by a applying a neural network [17, 18, 19, 20] that forms clusters in its input space and produces a good representative of these clusters as output.
The network operates in combining the Kohonen clustering algorithm and the class creation and pruning methods incorporated in the fuzzy-ART and fuzzy-ARTMAP algorithms. The topology of the network consists of an input layer and an output layer with full connection between these two layers. The input layer has dimensionality *k*, and the output layer grows and shrinks as new category neurons (each representing a conceptual state) are added and deleted.
When an input vector is presented to the network, all the output neurons calculate their distance (i.e., the distance of their weight vector to the input) in parallel. The neuron with the smallest distance is the winner neuron.

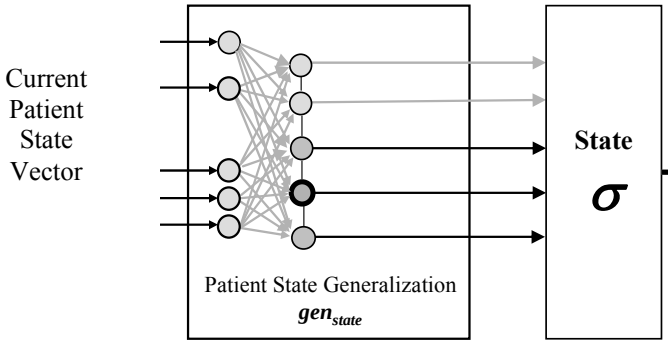

Fig 1.5 *Neural Network for patient state generalization*

At this point, we use an idea from the fuzzy-ART algorithm and check whether the winning neuron is close enough to be able to represent the input vector, or whether the input vector is so dissimilar to the winning neuron's weights (and thus to all the other categories as well) that it has to be placed into a new category. For this, we define a *similarity radius* as the maximum range an observation vector can be distant from the winning neuron's weights to be still considered close enough to fall into that category.

Category neuron training: If the observation vector v is within the similarity radius of the winning neuron's weights w than the weights of the winning neuron and the neurons in its neighbourhood are adjusted to reflect the new entry in this category by moving them into the direction of v.

Category neuron creation: If the observation vector is not within the similarity radius of the winning neuron, it is not similar enough to be included in one of the current categories, then a new category has to be created.
In this algorithm, this corresponds to the creation of a new output neuron that is fully connected to the input layer, and whose weight vector is equal to the sensor vector. The network starts with one output neuron that is created upon presentation of the first input pattern. Output neurons are created when an observation vector is found not to be within the similarity radius of any of the output neurons. It is clear that the smaller the similarity radius is, the more output neurons will be created because the criterion for similarity is stronger the smaller this radius is.

Category neuron pruning: To prevent a proliferation of output neurons, we include a pruning step that cuts output neurons. The pruning can be done according to two different criteria:

- a neuron is cut when it has not been the winning neuron for a given period of time, or

- a neuron is cut when it encodes too small a fraction of input vectors compared with the output neuron that encodes the most observation vectors.

The output associated with the clustering algorithm is not the output of the network, as these are only the similarity numbers for the various categories. Instead, the output of the algorithm is the weight vector of the winning output neuron. In other words, the output of the algorithm is the representative of the category that is most similar to the input presented to the network. If the observation vector is within the similarity radius of one of the output neurons, the output is the weight vector of this neuron; otherwise, it is the observation vector itself (which is also being incorporated in the network as a new category neuron).

The network presented above produces conceptual states from observation data. These generalized states are then used as inputs for the Q-Learning approach that models the effects of medication in generalized states space. The neural network that implements this function is presented in Section 4.

1.2 Therapy Planning Based On Patient Trials

Trial Level: Alignment of Trials

To store a patient's data and course of treatment, an electronic patient record as a source for retrospective inspection and medical decision-making is provided. There are many different approaches to store and generalize raw patient data. The generalization process depends on the kind of representation of knowledge for the medical decision-support system. One of most used formalism for knowledge representation is a *semantic network* with a *graph-grammar* approach to manage the complex graph transformations driven by the information entry (patient data) and medical problem solving [6]. For such knowledge representation it is necessary to use a similar representation of patient trials. The patient trial is constructed as a temporal subnet for the representation of the course of treatment. This temporal net represents medical events such as the administration of a drug through labeled, attributed nodes, while temporal relationships are modeled with labeled edges [6]. New patient data is represented as sub-graph and special *graph production rules* of the patient graph-grammar are used for mapping the new sub-graph into semantic patient graph. This production maps an inconsistency between two (diagnostic) data items, (such as a new laboratory value being not consistent to the current diagnosis) to the patient graph. After having found the matching region in the patient graph, the production generates a

new node and connects it to the existing nodes via inconsistency edges and tests the inconsistency related data. [6]. The complex network transformation processes driven by knowledge acquisition, problem solving and data entry are controlled by a graph-grammar.

Semantic graphs of individual patients are not efficient for the analysis of similarities between different patient trials. We present here one of other methods, which can be used to analyze patient trials.

The most basic analysis task of trials is to ask if two trials can be related. There are many positions at which two corresponding patient states (entries) can be compared. However, in the case when the one of the trial has extra entries, then in a couple of places gaps have to be inserted to the second trial to maintain an alignment across such regions. When we compare sequences of patient states, the basic processes are considered as substitution, which changes entries in the sequence, and insertion or deletion, which adds or removes entries. Insertions and deletions are referred to as gaps. An example for state alignment can be seen in Fig. 1.6. Based on the distance ρ between two generalized patient states (see Section 1.2), the system calculates the alignment distance (score) ρ^*, which describes the similarity (distance) between different patient trials.

$$\rho^* : S^* \times S^* \rightarrow R^+ \qquad (1.7)$$

where S^* is the set of ordered sequences of the states from the patient state space S. Each sequence $x = (x_1,\ldots, x_n) \in S^*$ has a respective sequence of events describing the concrete patient trial $(e_1,\ldots e_n)$.

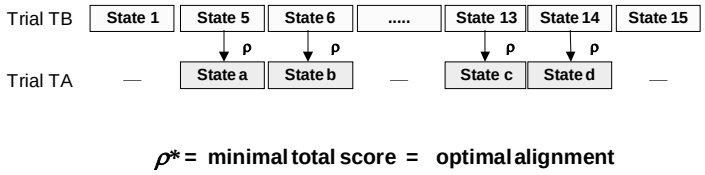

Fig.1.6 *Alignment between two patient trials*

The measure $\rho^*(x, y)$ calculates the distance between these two sequences, which describes the similarity of the flow of the therapy. We apply a special sequence-matching algorithm, well known in Bioinformatics for biological sequence alignment [21].

The global measurement (score) we assign to an alignment is the sum of terms for each aligned (similar) pair of states (entries), plus terms for gaps (see Fig. 1.7). We will consider a pair of trials (patient states sequences) *x* and *y* of lengths *n* and *m*, respectively.

Let x_i be the *i*-th state in *x* and y_j be the *j*-th symbol of *y*. Given a pair of trials, we want to assign a score to the alignment that gives a measure of the relative likelihood that the trials are related as opposed to being unrelated. For each two states x_i and y_j we can use ρ as measure of similarity of these residues. Let $n > m$ then we should add gaps g in the second trial to find the best alignment.

Gap penalties

We expect to penalize gaps. Each state x_i in the sequence *x* has an additional parameter τ_i which represents the time interval between the state x_{i-1} and x_i. The first state x_1 has $\tau_1=0$. For finding the penalty value of a gap at the *i*-th position of the trial *y* we use the knowledge about time intervals associated with each state. Let the last ungapped substitution with state y_l be on *(i-k)*-th position in the trial *x*.

x_{i-k}	x	x	x_{i-1}	x_i
y_l	g	g	g	g

Fig.1.7*. Gaps Insertion*

The standard cost associated with a gap is given by

$$\rho(x_i, g) = K \exp(-(|\tau^x_i - \tau^y_l|)) = pen(x_i, y_l) \qquad (1.8)$$

where τ^x_i is the time interval associated with state x_i and τ^y_l is the time interval associated with state y_l from trial *y* which was aligned with the state x_{i-k} from the trial *x*. The long insertions and deletions with different intervals of time are penalized less as those where the intervals of the time is quite the same.

Alignment algorithm

The problem we consider is that of obtaining an optimal alignment between two patient trials, allowing gaps. We can use the well known dynamic programming algorithm, which has many applications in biological sequences analysis [21]. The idea is to build up an optimal alignment using previous solutions for optimal alignments of smaller subsequences. The problem can be defined as follows:

Find the best alignment between sequences x and y* (x*, y* represent sequences x and y extended by necessary gaps) such that global score*

$$\rho^*(x,y) = \Sigma(\rho(x^*_i, y^*_i)) = min \qquad (1.9)$$

where $x^*_i = x_i$ *or gap g and if* x_i *is aligned to* y_l *and* x_{i+k} *is aligned to* y_u *then* $l<u$.

We can use the known dynamic programming algorithm, which has many applications in the biological sequences analysis [21]. The idea is to build up an optimal alignment using previous solutions for optimal alignments of smaller subsequences. We construct a matrix *F* indexed by *i* and *j*, one index for each trial, where value *F(i,j)* is the score of the best alignment between the initial segment $x_1 \ldots x_i$ of *x* up to x_i and the initial segment $y_1 \ldots y_j$ of *y* up to y_j. We can build *F(i,j)* recursively. For details see [22]. Let us assume that we are only interested in matches scoring ρ less than some threshold *T*, it means that similarity between two states of the patients is very high. *F(i,0)* is the minimal (best) sum of completed match scores to the subsequence $x_1, \ldots x_i$ assuming that x_i is in an unmatched region.

It is obvious that we expect that one trial contains the other or that they overlap. It means that we want a match to start on the top or left border of the matrix and finish on the right or bottom border. The initialization equations are that $F(i,0) = 0$ for $i = 1,\ldots, n$ and $F(0,j) = 0$ for $j = 1,..,m$. Now we calculate recursively the matrix value as

$$F(i,0) = \min \begin{cases} F(i-1,0) \\ F(i-1,m)+T \end{cases} \qquad (1.10)$$

$$F(i,j) = \min \begin{cases} F(i-1,j-1) + \rho(x_i, y_j) \\ F(i-1,j) + pen(x_{i-1}, y_j) \\ F(i,j-1) + pen(x_i, y_{j-1}) \end{cases} \qquad (1.11)$$

The calculation steps are presented in Fig.3.3. Let F_{min} be the minimal value on the right border *(i,m)* for $i = 1,..n$, and the bottom border *(n,j)* $j = 1,..,m$. This minimal score is the measure of the similarity between the complete two trials x and y. i.e.

$$\rho^*(x,y) = F_{min} \qquad (1.12)$$

To find the alignment itself we must find the path of choices that led to the minimal value. The procedure for doing this is known as a trace back. The trace back starts from the minimal point and continues until the top or left edge is reached.

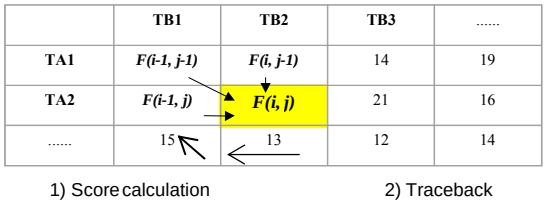

Fig.1.8 *Matrix for alignment calculation*

This procedure is applied to each pair of patient trials.

Clustering of the Trial Space

Based on $\rho*$ we can build the distance matrix between each trial and use standard clustering methods [14] to obtain the similarity classes (cluster) of trials. The similarity cluster c is defined as follows:

- $c \subset$ *Set of Trials*
- $(\forall\ x, y \in c)(\rho(x,y) < \varepsilon)\ldots$ (1.13)
- *card* $(c) \rightarrow max$

where ε is the threshold value discriminating the similarity of two trials. For construction of similarity classes we build the graph which nodes are trials and its arcs is the relation ρ*. The threshold value cuts different arcs. The similarity class is the maximal sub-graph of the graph for which its nodes are full connected (see Fig. 1.9).

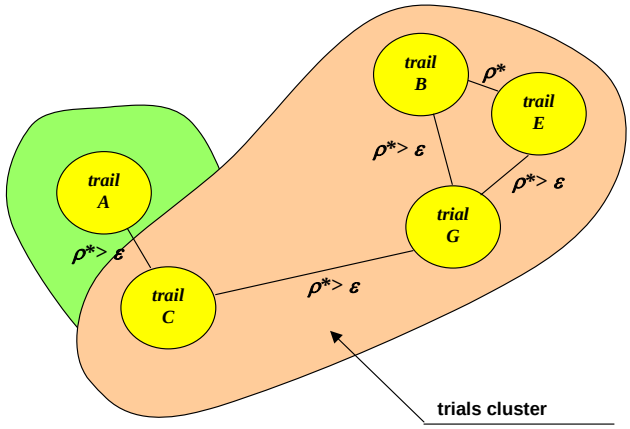

Fig.1.9 *Trials clustering*

Profiling of Patient Trials Families

In the previous sections we have already created a set of trials belonging to a particular cluster. Such a cluster is called a trial family. Trials in a family are diverged from each other in the primary sequence of the course of a disease. For each cluster *c* the profile trial in form of *pair generalized initial state* and *generalized final state*

$$profile(c) = (s^{c}_{initial}, s^{c}_{final}) \qquad (1.14)$$

is established.

The *generalized initial/final state* is generated as most probably *factorized values vector* of laboratory parameters from initial/final states and text mining created *common context* of observation protocols from initial/final states of each trial from cluster *c* respectively.
Additionally, for each trial *trial* from cluster *c* the medications *m* i.e. treatment of drug dosages, therapeutic serum levels and indications are assigned in a unique way. The different medications *m* can be applied to obtain the generalized final state from generalized initial state.
Let $\mathbf{M}(c) = \{M(trial) | \; trial \in c\}$ be the set of medications applied in trials from cluster *c*. For each medication m_i from $\mathbf{M}(c)$ we can calculate the application probability as

$$p(m_i) = \mu(m_i) / \Sigma \{\mu(m_j) \mid m_j \in \mathbf{M}(c)\} \qquad (1.15)$$

where $\mu(m_i)$ is the frequency of medication m_i application in cluster *c*. The profile of cluster c is shown in Fig. 1.10).

The distance measure ρ between the clinical states of patients may be used to introduce the new order relation between the clusters from cluster set *C*.
Let s_{fd} designate the state of a cured patient (destination state). We establish the preferable relation *pref* between the clusters

$$pref \subset C \times C$$

$$c_1 \; pref \; c_2 \Leftrightarrow \rho(s^{c1}_{final}; s_{fd}) + K\tau^{c1}_{final} < \rho(s^{c2}_{final}; s_{fd}) + K\tau^{c2}_{final} \qquad (1.16)$$

where $(s^{c1}_{init}, s^{c1}_{final}) = profile(c_1)$, $(s^{c2}_{init}, s^{c2}_{final}) = profile(c_2)$ and τ^{c1}_{final} , τ^{c2}_{final} are the time intervals associated with final events of both cluster respectively.

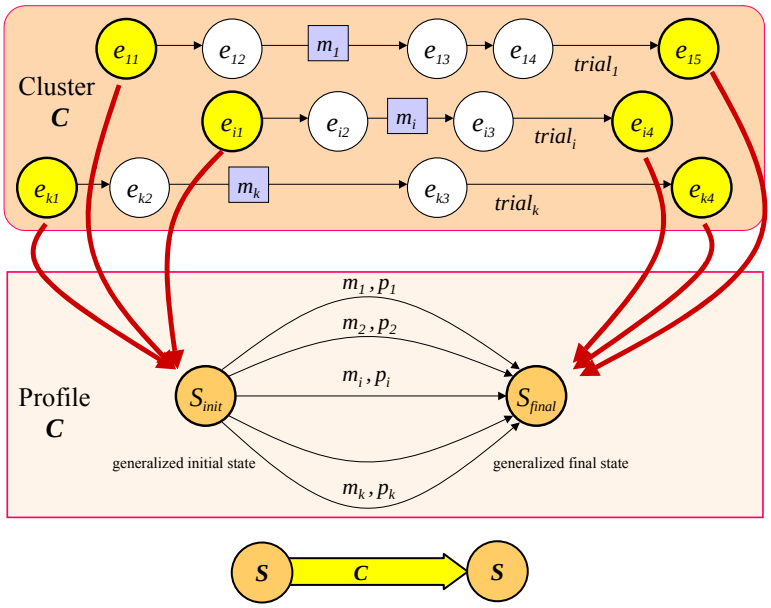

Fig.1.10 *Cluster and profile of cluster*

This relation *pref* establishes a linear order on the cluster set *C*. The ordered cluster set *C* is shown in Fig.1.11.

We assume that for each patient there exists generalized additional information such as age, gender, weight etc., and if available biomedical parameters. These static parameters are factorized and represented as fuzzy linguistic variables with appropriate membership functions. The set of fuzzy description of patient parameters can be clustered with help of fuzzy similarity relation [9]. As an additional support for diagnostics gene expression data can be used. Using micro-arrays, we can measure the expression levels of more genes simultaneously. These expression levels can be determined for samples taken at different time points during a biological process or for samples taken under different conditions. For each gene the arrangement of these measurements into a vector leads to what is generally called an *expression profile*. The *expression vectors* can be regarded as data points in high dimensional space. Cluster analysis in a collection of gene expression vectors aims at identifying subgroups - *clusters of co-expressed genes*, which have a higher probability of participating in the same pathway. The clusters can be used to validate or combine the cluster to prior medical knowledge. Many clustering methods are available, which can be divided into two groups: first and second generation algorithms. The first generation algorithms are represented by hierarchical clustering algorithms, K-means clustering algorithms

or self-organizing maps. These algorithms are complicate in use and often require the predefinition of more parameters that are hard to estimate in biomedical praxis. Another problem is that first generation clustering algorithms often force every data to point into a cluster. It can lead to lack of co-expression with other genes. Recently new clustering algorithms have started to tackle some of limitations of earlier methods. To this generation of algorithms belong: Self-organizing tree algorithms, quality based clustering and model-based clustering [18, 20, 23].

Self-organizing tree algorithms combine both: self-organizing maps and hierarchical clustering. The gene expression vectors are sequentially presented to terminal nodes of a dynamic binary tree. The greatest advantage is that the number of clusters does not have to be known in advance. In quality based clustering clusters are produced that have a quality guarantee, which ensures that all members of a cluster should be co-expressed with all other members of these clusters. The quality guarantee itself is defined as a fixed threshold for a maximal distance between two points between clusters. Based on these methods it is possible to generate clusters on the gene expression states.

Let *CPR* be a gene expression cluster, which contains data obtained from micro-arrays, in the patient record set. Each micro-array is connected with one or more patient trials. The cluster *CPR* can be mapped to the patient trial space *C*. This results in two patterns, which can be used for classifying each trial. On the one side the pattern based on trial alignment - on the other side the pattern based on gene expression and statistic patient information can be used. Both patterns can be combined for creation fine classes containing very similar trials with high co-expression of genes and statistic patient information (see Fig.1.12) [20].

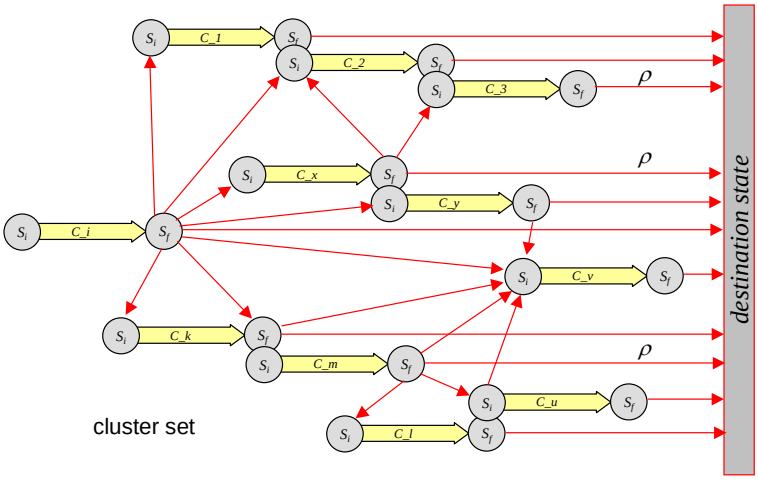

Fig. 1.11 *Preferable relation between clusters*

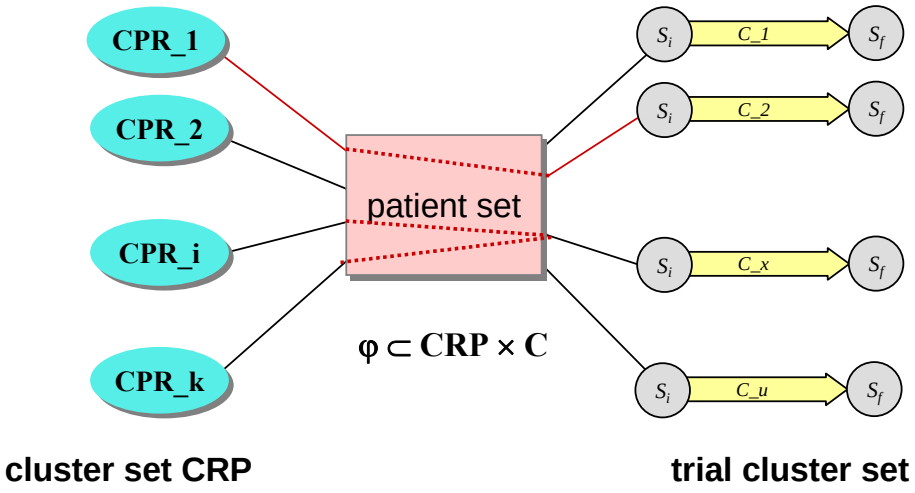

Fig. 1.12 *Mapping CPR clusters and trial clusters*

Based on these data we introduce the set of medications **M(***c***)** and set of patient information cluster **CPR**(*c*) = $\varphi^{-1}(c)$ for each cluster *c*. Finally the therapy model can be represented as

$$T\text{ - }Model = (C, \{profile(c) | c \in C\}, \{\mathbf{M}(c) | c \in C\}, \{\mathbf{CPR}(c) \mid c \in C\}, pref) \qquad (1.17)$$

The general structure of the therapy model is sketched in the Fig. 1.13.

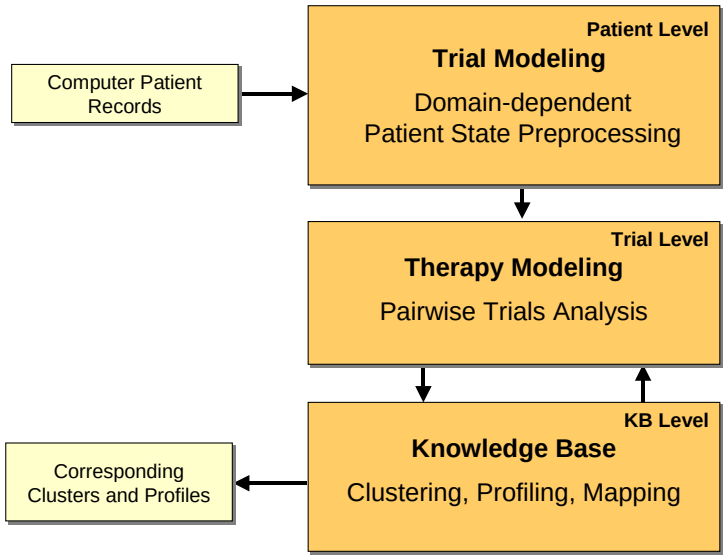

Fig.1.13 *Therapy model structure*

Therapy Planning Support

There are many different computer based support systems for diagnosis and therapy planning using different knowledge representation [1, 3, 6, 10, 11]. One of them is for example CEMS [10] or TheMPO: A Knowledge-Based System for Therapy Planning in Pediatric Oncology [6]. CEMS - Clinical Evaluation and Monitoring System - was developed to provide comprehensive support for clinical services in a psychiatric hospital. It represents a psychopharmacology monitoring system. The system provides decision support and automated monitoring for each key component of care i.e. assessment diagnosis and treatment and consist of four modules; treatment standards – pharmacotherapy guidelines, diagnosis checklists - DCL, and information alerts and outcome assessment.

Based on the first observation and CEMS procedures including laboratory tests we can establish the treatment and standard medication m_x for a new patient $\boldsymbol{x}$. Additionally the clinical state of patient s_x and statistic patient information cpr_x can be established.
We can use our therapy model to evaluate the proposed treatment m_x or to find a more preferable treatment for this disease case.

Algorithm for evaluation and monitoring of clinical treatment:
The evaluation procedure is realized in following constraints satisfaction steps (Fig 1.14). At first we look for the cluster in which the static data of patient $\boldsymbol{x}$ fits and we connect it to the trial cluster set. Based on this trial cluster we look for the path C_x in trial cluster graph that minimizes the sum of distances beginning in current patient state s_x to destination state s_{fd} . This path will be ordered with the relation *pref.* If current medication belongs to the first medication set in ordered path, then the decision about therapy is most preferable. If the current medication belongs to a different cluster the system proposes the set of other probably most preferable treatments.

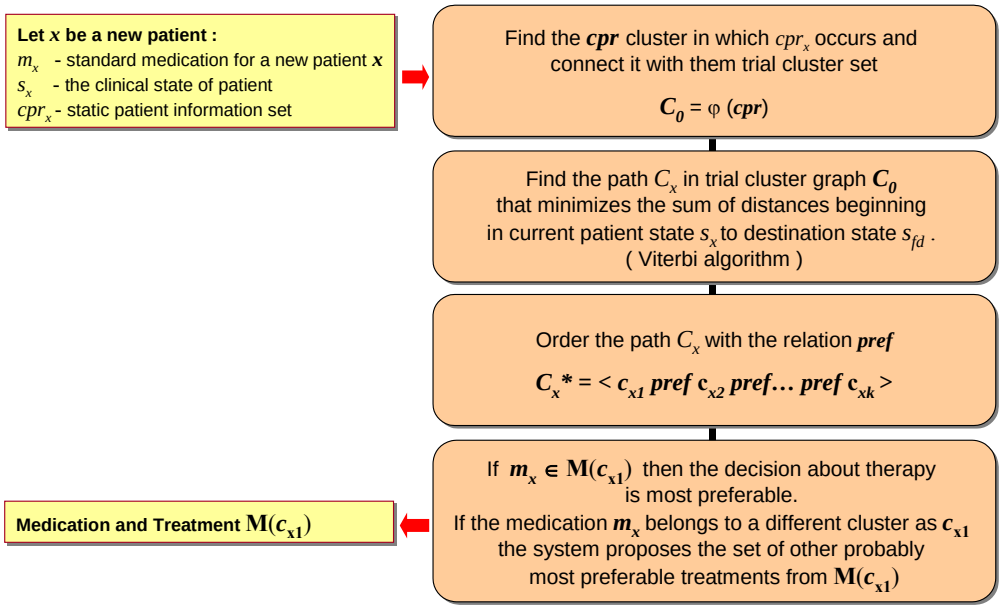

Fig 1.14 *Algorithm for evaluation and monitoring of clinical treatment*

This algorithm can be used to support the decision process for effective therapy planning in a clinical evaluation and monitoring system. The accuracy of the evaluation obviously depends on the number of trials used to construct the trial clusters. Model uncertainty about the trials sets in the clusters should be taken into account by adding posterior probabilities for the effectiveness of a therapy suggestion.

The concatenation of various trials in combination with extended computational methods could be used to gain deeper insight into the course of specific diseases depending on variations of medication dosages. The profile of cluster could be used to monitor a new patient's therapy course and produce alerts when deviations from the expected therapy path are observed.

1.3 Reinforcement Learning for Therapy Planning

Therapy planning for a specific disease is heavily affected by various levels of uncertainty concerning a patient's response to a therapeutic treatment. Underlying PD/PK models are often too complex to use in clinical practice, sometimes individual patients respond to drugs in an unexpected manner.

Therefore data driven approaches based on Reinforcement Learning like Q-learning strategies get more and more popular in planning individualized drug therapy. These strategies do not rely on

PD/PK models for therapy planning but use clinical observations, case records and available empirical data for drug selection and dosage.

In [7, 8] the application of Reinforcement-Learning methods are used to incorporate existing clinical expertise into the process of acquiring an appropriate administration strategy of rHuEPO to patients with anemia. In [24] a reinforcement learning approach is used to extract optimal structured treatment interruption strategies directly from clinical data without using an accurate mathematical model of HIV infection dynamics. A framework for the application of Q-learning in therapy planning is presented in the next sections.

Generalization of medications and dosages

For each critical event e_i of patient trial (see Section 1.1), the medications m_i i.e. treatment of drug dosages, therapeutic serum levels and indications are assigned in a unique way. The medication can be represented as set $m_i = \{(c_{ij}, \alpha_{ij}) | j = 1,..,k\}$ where c_{ij} is a chemical component of medicament and α_{ij} is its dosage.

For classification of medications it is necessary to introduce a similarity function between two medications m_k and m_j , s : M x M → $\Re$;

Let m'_i be a set of components of medication, $m'_i = \{c_{ij}, | j = 1,..,k\}$ and $I_{ik} = m'_k \cap m'_i$ then the similarity function can be described as:

$$s(m_k, m_i) = 1 - \beta_{ik}\, \gamma_{ik} \qquad (1.18)$$

where $\beta_{ik} = |I_{ik}| / \max\{|m'_i|, |m'_k|\}$ and $\gamma_{ik} = \min\{\Sigma \alpha_{cj}^{i} | c_{ij} \in I_{ik}, \Sigma \alpha_{cj}^{i} | c_{ij} \in I_{ik}\} / \max\{\Sigma \alpha_{cj}^{i} | c_{ij} \in I_{ik}, \Sigma \alpha_{cj}^{i} | c_{ij} \in I_{ik}\}$.

It is easy to observe that $s(m_k, m_k) = 0$ and $s(m_k, m_i) = s(m_i, m_k)$. The similarity function realizes the tolerance relation on the set of M. This tolerance relation can be used to construct the similarity classes of medications as: The $M_\varepsilon \subset M$ is the tolerance class of medication if and only if

$$(\forall m_k, m_i) \in M_\varepsilon)\, (s(m_k, m_i) < \varepsilon) \qquad (1.19)$$

The similarity classes – clusters of medication and similarity function are used to build the medication generalization function $\boldsymbol{gen_{Med}}$ ($\boldsymbol{gen_{Med}}$: M → M_ε) as

$$\boldsymbol{gen_{Med}}(m) = \arg(\min_{M\varepsilon}(\max\{s(m, m_k) | m_k \in M_\varepsilon\})) \qquad (1.20)$$

Q-Learning based Decision Support System for Therapy Planning

For each patient trial we can calculate a generalized trial as

$$\text{TRIAL} = ((\boldsymbol{gen}_{\boldsymbol{Med}}(m_i), \boldsymbol{gen}_{\boldsymbol{State}}(s_i)) \mid i=1,...,n) = ((\mu_i, \sigma_i)) \mid i=1,...,n) \quad (1.21)$$

and $\delta\tau_i < \delta\tau_{i+1}$ and $\boldsymbol{gen}_{\boldsymbol{State}}$

The decision support system is based on Q-Learning [5] which is a popular learning method to select actions from delayed and sparse reward. The goal of Q-learning is to learn strategies for generating whole action sequences, which maximize an externally given reward function. The reward may be delayed and/or sparse, i.e. reward is only received upon reaching the goal of the task or upon total failure.

Let S be the set of all possible patient states and $\boldsymbol{gen}_{\boldsymbol{State}}$ be the generalization function mapping the current patient state *s* into the generalized state of patients. We use now the restrictive Markov assumption, i.e. we assume that at any discrete point of time, the system obtains the complete generalized state. Additionally, let $\mathbf{M_\varepsilon}$(medication clusters) be the action set of the system. Based on the adequate state $\boldsymbol{gen}_{\boldsymbol{State}}(s)$ the system picks a generalized medication $\mu_\varepsilon \in \mathbf{M_\varepsilon}$, where $\mathbf{M_\varepsilon}$ is the set of medication clusters. As the result, the patient state changes. The trainer receives a scalar reward value, denoted by $\boldsymbol{r}(\boldsymbol{gen}_{\boldsymbol{State}}(s),\mu_\varepsilon)$, which measures the action performance. Such a reward can be exclusively received upon reaching a designated goal or upon total failure, respectively.

The Q-Learning algorithm should find an action strategy

$$\pi : S \rightarrow \mathbf{M_\varepsilon} \quad (1.22)$$

mapping from patient-states S to actions M_ε, which, when applied to action selection, maximize the so called *cumulative discounted future reward*. For fast finding the best action in current state s the key of Q-Learning is to learn a value function Q for picking the actions:

$$Q: S \times \mathbf{M_\varepsilon} \rightarrow \Re \quad (1.23)$$

maps percept by agent conceptual states s and actions μ_ε to scalar utility values. In the ideal case $Q(\boldsymbol{gen}_{\boldsymbol{State}}(s),\mu_\varepsilon)$ is, after learning, the maximum cumulative reward one can expect upon executing action μ_ε in state *s*. The function Q schedules actions according to their reward. The

value $Q(\boldsymbol{gen}_{State}(s),\mu_\varepsilon)$ grows with the expected cumulative reward for applying action μ_ε in the current state *s*. The value function Q, after learning, allows generating optimal actions by picking the action which maximizes Q for current state s, i.e.

$$\pi(s) = \arg(\max\{Q(\boldsymbol{gen}_{State}(s),\mu_\varepsilon)|\mu_\varepsilon \in \mathbf{M}_\varepsilon\}) \qquad (1.24)$$

The values of Q have to be learned over the whole lifetime of the system acting in the same disease. The function Q can be realized as the complex neural network. Initially, all values $Q(\sigma,\mu)$ are set to zero. During learning values are incrementally updated, using the following standard recursive procedure. Suppose the agent just executed a whole action sequence which, starting at some initial state σ_0 and led to a final state σ_F which reward $\boldsymbol{r}$ (σ_E,μ_F). For all steps *i* within this episode, $Q(\sigma_i,\mu_i)$ is updated through mixture of the values of subsequent state-action pairs, up to the final state. This standard procedure has the following form [26].

$$Q^{new}(\sigma_i,\mu_i) = \begin{cases} +\mathbf{N} \text{ if } \mu_i & \text{final action, positive result of treatment} \\ -\mathbf{N} \text{ if } \mu_i & \text{final action, negative result of treatment} \\ \psi[(1-\lambda)(\max_{\mu} Q(\sigma_{i+1},\mu)) + \lambda\, Q^{new}(\sigma_{i+1},\mu_{i+1})] & \end{cases} \qquad (1.25)$$

where $\psi \leq 1$ is the discount factor.

Such Q-learning learns individual strategies independently, ignoring the opportunity for the transfer of knowledge across different sequences of medications. The architecture of the Q-Learning decision support system is presented in Figure 1.15.

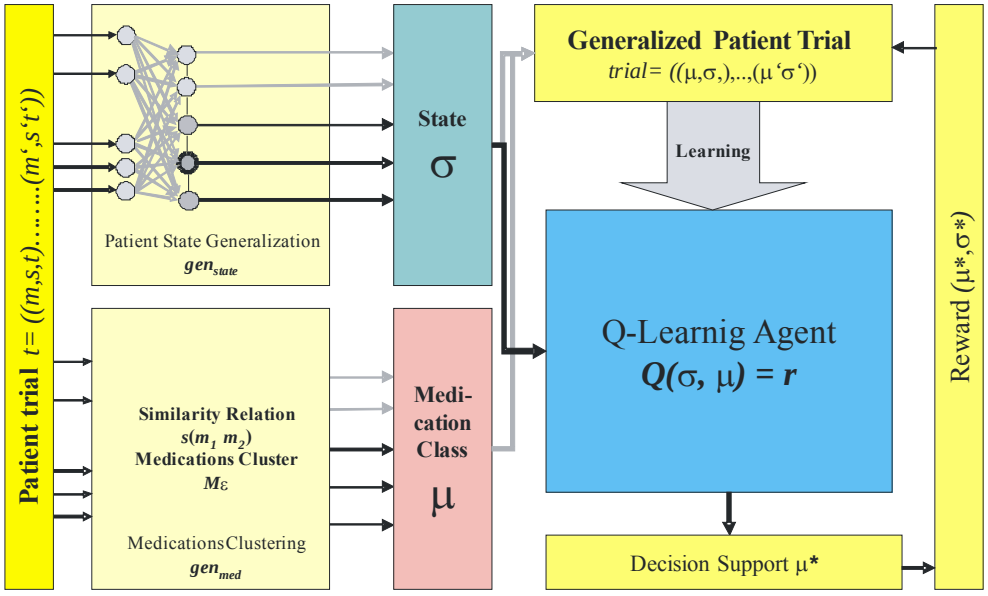

Fig. 1.15 *Q-Learning based Decision Support System*

Our approach considers various dosages of one specific drug and accounts for the fact that drug prescription can change during therapy by switching to medications with different active components.

1.5 Final Remarks

Evidence-based Systems gain a more important role in decision support for medical practice. This chapter presents two approaches for a multilevel knowledge base system for therapy decision support combined with bioinformatics components. In the first level a sequence of events called patient trial is extracted from computer patient records. These events describe one flow of therapy for a concrete disease. Each event is represented by state and time. We introduce a measure between states, which is used to calculate the best alignment between different patient trials. The alignment measure calculates the distance between two sequences of patient states, which represents the similarity of the course of disease. On the second level based on similarity-value classes are introduced by using specific clustering methods. These classes can be treated more exactly by combining the information about gene expression data on microarrays and other patient data. This leads to finer clustering containing similar trials - called trial families.

The trial clusters can be used to support the decision process for effective therapy planning in a clinical evaluation and monitoring system. The accuracy of the evaluation obviously depends on

the number of trials used to construct the trial clusters. Model uncertainty about the trials sets in the clusters should be taken into account by adding posterior probabilities for the effectiveness of a therapy suggestion. Furthermore the general patient data could be extended by adding information about the genotype of the patient such as relevant gene expression patterns observed in specific disease cases to obtain more diversification on cluster level. The second approach presented uses Reinforcement learning techniques (Q-Learning) for finding strategies for optimal drug selection and dosage.

References for Chapter 1

[1] C. Cao, T. Leong, A. Leong, "Dynamic Decision Analysis in Medicine: A Data Driven Approach". Int. Journal of Medical Informatics, 1999

[2] W. Jacak, K. Proell, "Data Driven Therapy Modeling", Proceedings of the International Multiconference (I3M 2006), Barcelona, Spain, 2006

[3] P. Magni, R. Bellazzi, F. Locatelli, "Using Uncertainty Management Techniques in medical Therapy Planning: a Decision-Theoretic approach", Applications of Uncertainty Formalisms A. Hunter and S. Parsons, (Eds). LNCS, Springer,1998

[4] S. Miksch, K. Cheng; and B. Hayes-Roth, "An Intelligent Assistant For Patient Health Care", Proceedings of the First International Conference on Autonomous Agents (Agents'97), ACM Press, 1997

[5] N. Lavrac, "Machine learning for data mining in medicine", Plenary invited talk at AIMDM'99, Aalborg, 20-24 June 1999. Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, pages 47-62, Springer Verlag, 1999

[6] R. Müller, M. Sergl, U. Nauerth, D. Schoppe, K. Pommerening, H.-M. Dittrich, "TheMPO: A Knowledge-Based System for Therapy Planning in Pediatric Oncology", Computers in Biology and Medicine, vol. 27(3), 177-200, 1997

[7] A.E. Gaweda, M.K. Muezzinoglu, G.R. Aronoff, A.A Jacobs, J.M. Zurada, M.E Brier, "Incorporating prior knowledge into Q-learning for drug delivery individualization", Machine Learning and Applications, 2005

[8] A.E. Gaweda, M.K. Muezzinoglu, G.R. Aronoff, A.A Jacobs, J.M. Zurada, M.E Brier, "Using clinical information in goal-oriented learning", Engineering in Medicine and Biology Magazine, IEEE, March-April 2007

[9] H. Teodorescu, A. Kandel, L. Jain, "Fuzzy and Neuro-Fuzzy Systems in Medicine", CRC Press, 1999

[10] J. Bronzino, "Expert system in psychiatry: A review". Journal of Intel. Systems, 1993

[11] R. A. Dunstan, R. Devenish, R. Kevill, "A Computer-Guided Diagnosis System for Transfusion Medicine Laboratories", School of Biomedical Sciences and Academic Computing Services, Curtin University of Technology, 1997

[12] C. Safran and G. Herrmann, "Computer based support for clinical decision-making", Med. Decision. Making, 1999

[13] S. Smith, J. Park, "Therapy Planning as Constraints Satisfaction Problem: A Computer based Antiretroviral Therapy Advisor for the Management of HIV", Medinfo, 2000

[14] Y. Moreau, F. De Smet, G. Thijs, K. Marchal, B. De Moor, "Adaptive quality-based clustering of gene expression profiles", Bioinformatics, Vol.18, no.5, 2002

[15] F. Sonnenberg, C. Hagerty, C. Kulikowski, "An architecture for knowledge based construction of decision models", Med. Decision Making, 1994

[16] T. Kohonen, Self-organizing Maps, Springer-Verlag, 1997

[17] S. Thrun, T. Mitchell, "Integrating inductive neural network learning and explanation-based learning", Proc. Of IJCAI'93, Chamberry, France, 1993

[18] M.E. Brier, J.M. Zurada, G.R. Aronoff, "Neural network predicted peak and trough gentamicin concentrations", Pharmaceutical Research 1995 Mar; pp. 406–412, 1995

[19] J.D.M. Guerrero, E.S. Olivas, G. Camps Valls, A.J. Serrano Lopez, J.J. Perez Ruixo, N.V. Jimenez Torres, "Use of neural networks for dosage individualisation of erythropoetin in patients with secondary anemia to chronic renal failure", Computers in Biology and Medicine, 2003

[20] A.A. Jacobs, P. Lada, J.M. Zurada, M.E. Brier, G.R. Aronoff, "Predictors of hematocrit in hemodialysis patients as determined by artificial neural networks", Journal of American Society of Nephrology 12, 2001

[21] R. Durbin., S. Eddy, A. Krogh, G. Mitchison, Biological sequence analysis, Cambridge University Press, 1998

[22] R. Merkl, S. Waack, Bioinformatics Interactive, Wiley, 2001

[23] V. Aris, M. Recce, "A method to improve detection of disease using selective expressed genes in microarray data", Methods of Microarray Data Analysis, Kluwer Academic, pp. 69-80, 2002

[24] D. Ernst, G. –B. Stan, J. Gongalves, L. Wehenkel, "Clinical data based optimal STI strategies for HIV: a reinforcement learning approach", Decision and Control, 2006 45th IEEE Conference on, Volume, Issue, 13-15 Dec. Page(s): 667 – 672, 2006