Inductive Learning for Case-Based Diagnosis with Multiple Faults

Joachim Baumeister, Martin Atzmueller and Frank Puppe

University of Würzburg, 97074 Würzburg, Germany
Department of Computer Science
Phone: +49 931 888-6740, Fax: +49 931 888-6732
email: {baumeister, atzmueller, puppe}@informatik.uni-wuerzburg.de

Abstract. We present adapted inductive methods for learning similarities, parameter weights and diagnostic profiles for case-based reasoning. All of these methods can be refined incrementally by applying different types of background knowledge. Diagnostic profiles are used for extending the conventional CBR to solve cases with multiple faults. The context of our work is to supplement a medical documentation and consultation system by CBR techniques, and we present an evaluation with a real-world case base.

1 Introduction

The main advantage of case-based reasoning (CBR) systems is its quite natural knowledge acquisition process, because cases often must be documented for various purposes and can then be exploited for decision support. However, cases are not sufficient for CBR, which needs four knowledge containers [1]: vocabulary, case base, similarity measure and adaptation knowledge. In a structured documentation system high quality cases are available with a predefined vocabulary. In this paper, we show how to extract the knowledge for the other two containers (similarity measure, adaptation knowledge) from the cases (semi-)automatically, in order to augment a structured documentation system by CBR. In particular, we discuss, which background knowledge is helpful in learning the content for the two containers and measure its effects in tests. The main difficulty is how to deal with multiple faults in cases, which makes it a rare event, that cases have exactly the same solution. For solving this difficulty we learn diagnostic profiles, i.e. typical observations for each diagnosis, infer set-covering knowledge from the profiles, and use the capabilities of set-covering inference for multiple faults. The results are tested by combining appropriate cases.

Our context is a structured documentation system in medicine, being used for documenting the results of specialized examinations. The cases are detailed descriptions

© In: S. Craw, A. Preece (Eds.): Advances in Case-Based Reasoning (ECCBR 2002), LNAI 2416, pp 28-42, Springer Verlag
Proceedings of the 6th European Conference on Case Based Reasoning
Revised version of paper (August 18, 2003)

of symptoms and findings of the examination(s), together with the inferred diagnoses (faults), i.e. a case consists of a list of attribute-value pairs (observations) together with a list of solution elements. Both observations and diagnoses may be ordered hierarchically due to the structured data gathering strategy, i.e. findings are usually first specified in general terms and then further detailed with follow-up questions, and diagnoses have a specialization hierarchy as well. This setting yields – in contrast to many other CBR projects – a high quality of the case base with detailed and usually correct case descriptions.

Our implementation and evaluation is based on the knowledge-based documentation and consultation system for sonography SONOCONSULT (an advanced and isolated part of HEPATOCONSULT [2]) being in routine use in the DRK-hospital in Berlin/Köpenick based on the diagnostic shell kit D3 [3]. In addition to an documentation system, SONOCONSULT also infers diagnoses with heuristic expert knowledge, but this capability is not important for our approach. SONOCONSULT documents an average of 300 cases per month and generates a conventional physician's letter with a rule-based template from the structured input entered in hierarchical questionnaires. Included are the inferred diagnoses, which can be corrected manually, but are usually correct due to first evaluations of SONOCONSULT.

The goals for adding a CBR component to SONOCONSULT are to validate the solution for the current case and to provide additional information, e.g. explanations based on the presented similarity to former cases and the possibility to look up information in the corresponding patient records concerning therapy, complications, prognosis or the treating physician as contact person for special questions. In general, we envision a hybrid way of building intelligent documentation systems, by defining the data gathering, data abstraction and basic diagnostic inference knowledge in a rule-based representation and using case-based reasoning for fine-tuning and maintenance.

The rest of the paper is organized as follows: In Section 2 we define case-based diagnosis and diagnostic profiles. We introduce our basic concept of case-based reasoning with multiple faults. In Section 3 we describe methods for learning partial similarities, weights and diagnostic profiles from cases. These knowledge extensions are applied when retrieving a new case. In Section 4 we present the usage of the learned knowledge in a dynamic retrieve process, which is appropriate for handling multiple faults. An evaluation with a real-world case base is given in Section 5. We will conclude the paper in Section 6 with a discussion of the presented work and we show promising directions for future work.

2 Case-Based Diagnosis with Diagnostic Profiles

In this section we give the basic definitions needed for the learning methods presented in Section 3 and the reasoning task in Section 4. We introduce a similarity measure to compare cases and we define diagnostic profiles, which support the case-based reasoning process. Furthermore we present a concept for handling cases with (possibly) multiple faults in the retrieve step.

2.1 Basic Definitions

Let Ω_D be the set of all diagnoses and Ω_P the set of all parameters (attributes). To each parameter $p \in \Omega_P$ a range $dom(p)$ of values is assigned. Further we assume Ω_F to be the (universal) set of findings $(p = v)$, where $p \in \Omega_P$ is a parameter and $v \in dom(p)$ is a possible value. Let CB be the case base containing all available cases that have been solved previously. A case $c \in CB$ is defined as a tuple

$$c = (F_c, D_c, I_c) \tag{1}$$

where $F_c \subseteq \Omega_F$ is the set of findings observed in the case c. These findings are commonly called *problem description*. The set $D_c \subseteq \Omega_D$ is the set of diagnoses describing the *solution* for this case. We see, that the solution D_c for a case c can consist of multiple diagnoses (faults). The case can also contain additional information I_c like therapy advices or prognostic hints.

To compare the similarity of a new case c with another case c' we apply the commonly used weighted similarity measure given in Equation 2, which is an adaptation of the *Hamming distance* with weights and partial similarities between values of parameters p (e.g. see [4], page 183f.) :

$$sim(c, c') = \frac{\sum\limits_{p \in \Omega_P} w_a(p) \cdot sim_p\big(v_c(p), v_{c'}(p)\big)}{\sum\limits_{p \in \Omega_P} w_a(p)} \tag{2}$$

where $v_c(p)$ is a function, which returns the value of the parameter p in case c, and $w_a(p)$ is the weight of parameter p. Additionally, the weight of a parameter can be supplemented by an abnormality degree specified for each parameter value (cf. Section 3.2). If weights for parameters are not available, then we set $w_a(p) = 1$ for all $p \in \Omega_P$.

Now we will introduce diagnostic profiles, which describe a compact case representation for each diagnosis, since they contain the findings that occur most frequently with the diagnosis.

Definition 1 (Frequency Profile). *A frequency profile F_{P_d} for a diagnosis $d \in \Omega_D$ is defined as the set of tuples*

$$F_{P_d} = \big\{ (f, \mathit{freq}_{f,d}) \,\big|\, f \in \Omega_F \wedge \mathit{freq}_{f,d} \in [0,1] \big\} \tag{3}$$

where f is a finding and $\mathit{freq}_{f,d} \in [0,1]$ represents the frequency the finding f occurs in conjunction with d, i.e.

$$\mathit{freq}_{f,d} = \frac{\big|\{\, c \in CB \mid f \in F_c \wedge d \in D_c \,\}\big|}{\big|\{\, c \in CB \mid d \in D_c \}\,\big|} \,. \tag{4}$$

Since we consider cases with multiple faults, it can be quite helpful to know the set of diagnoses a given diagnosis usually occurs with. For this reason we augment frequency profiles with this information in the next step.

Definition 2 (Diagnostic Profile). *A diagnostic profile P_d for a diagnosis $d \in \Omega_D$ is defined as the tuple*

$$P_d = (F_{P_d}, D^{corr}_{P_d}) \tag{5}$$

where F_{P_d} is a frequency profile for diagnosis d. The set

$$D^{corr}_{P_d} \in \left\{ (d', \mathit{freq}_{d',d}) \,\middle|\, d' \in \Omega_D \wedge \mathit{freq}_{d',d} \in [0,1] \right\}$$

contains all diagnoses d' that appear together with d in the solution part of the cases in CB. The number $\mathit{freq}_{d',d}$ represents the frequency the diagnosis d' co-occurs with d.

2.2 Basic Concept for Handling Multiple Faults

We give a brief overview of the basic concept we developed for handling multiple faults (diagnoses) in case-based reasoning. In Sections 3 and 4 we focus on the methods in more detail.

To improve the quality of the case-based reasoner and to handle multiple diagnoses in an appropriate way, we apply the following steps:

1. Use the available cases for learning knowledge about partial similarities and weights for parameters.
2. Construct diagnostic profiles utilizing the learned knowledge and infer basic set-covering knowledge [5,6] from the profiles.
3. Apply learned knowledge for case-based reasoning as described in Equation 2.
4. If no case is sufficiently similar, then combine cases guided by the set-covering knowledge.

For the first two steps we provide the opportunity of a manual adaptation of the learned knowledge (similarities, weights, diagnostic profiles) for refinement. In the third step we apply the learned knowledge in the retrieve step of the case-based reasoner. For a new case we firstly try to find a sufficiently similar case in the case base. If a sufficiently similar case has been solved before, then we simply reuse the solution of this case. We say, that a case is *sufficiently similar* to another case, if the similarity between these two cases exceeds a given (and usually high) threshold. Since we consider a domain with cases containing multiple faults, such matches might be rare.
If no sufficiently similar case has been found, we apply an abductive reasoning step using the diagnostic profiles to find diagnoses, that can explain the current problem description. On the basis of this explanation we construct prototypical candidate cases containing cases of the case base. These candidates are presented to the user as possible solutions for the current case.

Related Work The combination of case-based reasoning with other knowledge representations has already been investigated in many approaches. The systems CASEY [7] and ADAPtER [8] are prominent examples for approaches that use case-based reasoning in a first step for selecting solved cases, which match the current observation best. In a second step, abductive knowledge is applied to adapt the old cases with respect to the current observation and give a verbose explanation for the adaptation. In our work we use the reverse approach, when using abductive reasoning for guiding the search of how to combine cases.

Another aspect of the work presented here is the problem of learning abductive models from cases containing multiple faults. Work in this field has been done by Thompson and Mooney [9] with an inductive learning algorithm, that generates set-covering relations from cases containing multiple faults. For this, a simple hill-climbing strategy is applied which adds more specific set-covering rules until the classification accuracy decreases. In contrast to our approach no additional knowledge like partial similarities is used to increase the diagnostic quality. Wang et al. [10] presented a connectionist approach when learning fuzzy set-covering models from cases generated by a simulation environment. They use an adapted back-propagation method that learns fuzzy set-covering relations by adjusting connective weights. But, besides the fuzzy covering relations, no additional knowledge like feature weights is applied in this method. Schmidt et al. [11] considered a simple generation process of prototypes from cases with the ICONS project. This case-based system has been developed for selecting an appropriate antibiotics therapy for patients in ICU domains. For this purpose, prototypes are generated from previous cases to supply retrieval and adaptation of a newly entered case. The construction of the prototypes is quite simple, since they are formed out of cases containing equal findings and therapy advices. When a new case is entered, the system adapts the most similar prototypes with respect to contra-indications for the given therapy advices. Furthermore new prototypes are generated, if new cases do not fit in existing ones.

3 Inductive Learning of Similarities, Weights and Diagnostic Profiles

In this section we consider the problem of inductively learning partial similarities for parameter values and weights for parameters. We further show how to build diagnostic profiles for single diagnoses.

3.1 Preprocessing Heterogenous Data

The algorithms presented in the further subsections are designed to handle discrete value domains of parameters. Nevertheless the available case base also contains some continuous data as well. Therefore we will transform continuous parameters into parameters with discrete partitions in a preprocessing step. The discretization is only done for the learning task and will not change the case base in principle. A lot of work has been done on the discretization of continuous parameters and there exists a wide range of methods (cf. [12,13] for empirical comparisons).

Automatic Partitioning of Parameter Domains The simplest method applicable to our problem is the *Equal Width Interval Binning*, which divides the domain $dom(p)$ of a parameter p into equal sized bins. A more promising approach seems to be the usage of clustering methods (cf. [14] for a survey), that groups partitions relative to the frequency the findings occur in the single partitions. Due to the limited space we will omit a more detailed description of appropriate clustering methods.

Predefined Partitions For some continuous parameters the expert already defined reasonable partitions. In the case, that there are predefined partitions available, we use these instead of the automatic binning methods mentioned above.

3.2 Similarity Knowledge for Parameter Values

The use of similarities between finding values can improve learning methods and reasoning capabilities dramatically. For example, the construction of diagnostic profiles benefits from similarity knowledge, because similar values of a parameter can work together in a diagnostic profile, rather than to compete against each other. Thus we consider learning similarities before building diagnostic profiles.

In the following we will use the term *distance function*, but it is obvious, that a distance function d directly corresponds to a similarity function sim. For two findings ($p = x$) and ($p = y$) we define their similarity by

$$sim_p(x, y) = 1 - d_p(x, y) \; . \tag{6}$$

Common Distance Functions One of the most commonly known distance function is the *City-Block* or *Manhattan* distance function, which is defined as follows:

$$d_p^m(x, y) = \frac{|x - y|}{\alpha} \tag{7}$$

where x and y are values for parameter p and $\alpha = x_{max} - x_{min}$. Obviously the Manhattan distance function is only appropriate for continuous or scaled parameters p.
For discrete parameters we implemented the *Value Difference Metric* (VDM) as proposed in [15] and improved in [16]. Given two findings $f_1 = (p = x)$ and $f_2 = (p = y)$ the VDM defines the distance between the two values x and y of parameter p:

$$vdm_p(x, y) = \frac{1}{|\Omega_D|} \cdot \sum_{d \in \Omega_D} \left| \frac{N(p = x \,|\, d)}{N(p = x)} - \frac{N(p = y \,|\, d)}{N(p = y)} \right| \tag{8}$$

where $N(p = x)$ is the number of cases in CB, for which parameter p is assigned to value x, i.e. $(p = x) \in F_c$. $N(p = x \,|\, d)$ is the number of cases c in CB with diagnosis $d \in D_c$, and parameter p is assigned to value x, i.e. $(p = x) \in F_c$.

With this measure, two values are considered to be more similar, if they have more similar correlations with the diagnoses they occur with. Thus we obtain the following distance function d for a parameter $p \in \Omega_P$ with values $x, y \in dom(p)$:

$$d_p(x, y) = \begin{cases} 1 & \text{if } x \text{ or } y \text{ is unknown,} \\ vdm_p(x, y) & \text{otherwise.} \end{cases} \tag{9}$$

Distance Metrics using additional Knowledge Since the underlying knowledge base is highly structured, we were able to utilize helpful information to augment the distances between parameter values.

Abnormalities. During the knowledge-acquisition process discrete and nominal values were marked to describe, whether they represent a normal or an abnormal state of their corresponding parameter (e.g. *pain=none* is normal, whereas *pain=high* is abnormal). Abnormal states can be sub-categorized into five degrees of abnormality (i.e. $A1$, $A2$, $A3$, $A4$, $A5$). We can utilize this information to divide the value range into an abnormal and a normal partition. To obtain the distance between a normal value y and an abnormal value x we use the following matrix

$d_p(x,y)$	*abn(x)=A1*	*abn(x)=A2*	*abn(x)=A3*	*abn(x)=A4*	*abn(x)=A5*
abn(y)=A0	0.6	0.7	0.8	0.9	1

where $abn(x)$ is a function returning the abnormality for the given value and $A0$ defines a normal value. So we get a maximum distance between a normal and a totally abnormal value, e.g., $d_p(x,y) = 1$ for $abn(x) = A5$ and $abn(y) = A0$.
After that, we compute the similarities for the remaining values by applying the VDM method (see Equation 8) for the values contained in the "normal values"–partition and for the values contained in the "abnormal values"–partition.

Scalability Knowledge. Beyond abnormalities the expert may mark some of the parameters as *scaled* to characterize, that values, that are closer to each other, are more similar. For example, $dom(pain) = \{none, little, medium, high\}$ is scaled, whereas $dom(color) = \{green, black, red\}$ is not scaled. We can utilize this flag, by applying the VDM method not for all distinct pairs of values within each partition, but only for adjacent values. Then, we interpolate the remaining distances by the following equation

$$d_p(v_i, v_{i+k}) = d_p(v_i, v_{i+k-1}) + d_p(v_{i+k-1}, v_{i+k}) \qquad (10)$$

where $k \geq 2$. After interpolating the remaining distances we have to normalize the whole distance matrix for parameter p, so that for all values $v, v' \in dom(p)$ it holds that $0 \leq d(v, v') \leq 1$.

3.3 Learning Diagnostic Profiles from Cases

The available cases usually contain more than one diagnosis (multiple faults). This characteristics makes it difficult to generate exact profiles for each single diagnosis, because it is not obvious, which findings are caused by the single diagnoses. Since we had a sufficient number of cases containing each diagnosis but rare repetitions of combinations of the diagnoses, we applied a statistical method for learning the most frequent covered symptoms of a diagnosis.

In the following we present an algorithm for building diagnostic profiles describing single diagnoses. Each profile contains at least all relevant findings for the specified diagnosis. We divide the algorithm into two parts: In Algorithm LCP we learn coarse profiles from the cases given by the case base. In Algorithm BDP we build diagnostic profiles from the coarse profiles learned before.

Learning Coarse Profiles In Algorithm LCP we will consider the cases contained in the training set CB. For each case $c \in CB$ we will update the diagnostic profiles of the diagnoses contained in the solution part of c. So, for each diagnosis $d \in D_c$ we will add the findings of the case to the corresponding diagnostic profile P_d, respectively increase their frequencies. Additionally we will update P_d by increasing the frequencies of the diagnoses d' co-occurring with d. Diagnoses $d' \in D_c \setminus \{d\}$ with a very high frequency, i.e. co-occurring very often with diagnosis d, tend to depend on d. Therefore the profiles for both diagnoses may have equal subsets of findings, which are only caused by one diagnosis. Thus, removing findings from the diagnostic profile, that are caused by the other diagnosis, will increase the quality of the profiles. Due to the limited space of this paper we will omit a detailed consideration of learning dependency between diagnoses (e.g. [17] introduces learning dependencies).

Algorithm 1. LCP: LEARNING COARSE PROFILES

```
Require: Cases $c$ contained in $CB$
for all cases $c \in CB$ do
    for all diagnoses $d \in D_c$ do
        /* Update profile $P_d = (F_{P_d}, D_{P_d}^{corr})$ */
        for all findings $f \in F_c$ do
            Increment frequency of $f$ in $F_{P_d}$
        end for
        for all diagnoses $d' \in D_c \setminus \{d\}$ do
            Increment frequency of $d'$ in $D_{P_d}^{corr}$
        end for
    end for
end for
Ensure: Coarse Diagnostic Profile $P_d = (F_{P_d}, D_{P_d}^{corr})$ for each diagnosis $d$
```

Build Diagnostic Profiles The diagnostic profiles learned in Algorithm LCP will also contain rare findings. In a second step we will remove these unfrequent findings from the profile. Before that, we will consider similarities between findings in the profile. For example, if a coarse profile includes the finding *pain=high* (*p=h*) with frequency 0.4 and the finding *pain=very high* (*p=vh*) with frequency 0.4 then both findings might be too rare to remain in the profile (e.g. with a threshold $\mathcal{T}_{DP} = 0.5$). But, since both findings are very similar to each other, an adapted frequency may be sufficiently frequent to remain in the profile. For example, if $sim_p(h, vh) = 0.8$, then an adapted frequency $freq'_{x,d}$, concerning similar findings, will be

$$\begin{aligned} freq'_{p=h,d} &= freq_{p=h,d} + \big(sim_p(h, vh) \cdot freq'_{p=vh,d}\big) \\ &= 0.4 + (0.8 \cdot 0.4) = 0.72 \,. \end{aligned}$$

We will adapt this idea, when we firstly compute a combined frequency $freq'_f$ for each finding $f \in F_{P_d}$, regarding similarities between values of the same parameter. After

Algorithm 2. BDP: BUILD DIAGNOSTIC PROFILES

```
Require: Coarse profile $P_d$ is available for each diagnosis $d$,
  defined threshold $\mathcal{T}_{DP}$ for pruning unfrequent findings
for all diagnostic profiles $P_d = (F_{P_d}, D^{corr}_{P_d})$ do
  Generate finding sets $F^m_{P_d}$ such that each finding contained
  in set $F^m_{P_d}$ is assigned to the same parameter $m$.
  for all finding sets $F^m_{P_d}$ do
    /* compute adapted frequencies of findings regarding their similarities */
    for all findings $f \in F^m_{P_d}$ do
      $\mathit{freq}'_{f,d} = \mathit{freq}_{f,d} + \sum_{f' \in F^m_{P_d} \setminus \{f\}} \mathit{freq}_{f',d} \cdot sim(f, f')$
    end for
  end for
  /* Remove findings with frequency less than threshold $\mathcal{T}_{DP}$ */
  $F_{P_d} = \{\, f \in F_{P_d} \,|\, \mathit{freq}'_f \geq \mathcal{T}_{DP} \,\}$
end for
Ensure: Created diagnostic profile $P_d$ for each diagnosis.
```

adapting the frequencies, we will remove all findings from the profile, which are still too unfrequent with respect to a given threshold $\mathcal{T}_{DP}$. We point out, that a diagnostic profile can contain more than one value of the same parameter, if their adapted frequencies exceed the threshold $\mathcal{T}_{DP}$.

It is worth mentioning, that the threshold $\mathcal{T}_{DP}$ directly corresponds to the resulting size of the diagnostic profiles. For a large threshold we will compute sparse profiles, which may be too special to cover all common findings for the diagnosis. Small thresholds will result in too general profiles, which will cover too many findings. This can yield a bad diagnostic quality.

The result of the algorithm is a set of a diagnostic profiles, where each diagnostic profile directly corresponds to a set-covering model [5,6] defining the frequently observed findings of the diagnosis. A set-covering model contains set-covering relations, which describe relations like: *"A diagnosis d predicts, that the finding f is observed in $\mathit{freq}_{f,d}$ percent of all known cases."* We denote set-covering relations by

$$r = d \rightarrow f \ [\mathit{freq}_{f,d}]\,. \tag{11}$$

Further, we define $cause(r) = d$ and $\mathit{effect}(r) = f$. A set-covering model SCM_d for a diagnosis d is defined as a set of covering relations

$$SCM_d = \{\, r \in \mathcal{R} \,|\, cause(r) = d \,\} \tag{12}$$

where $\mathcal{R}$ is the set of covering relations included in the knowledge base. As shown in [5], set-covering models are able to process similarities, weights and frequencies.

To transform a given diagnostic profile $P_d = (F_{P_d}, D^{corr}_{P_d})$ into a set-covering model SCM_d, we simply have to perform the following step: For all findings $f \in F_{P_d}$, create a new covering relation $r = d \rightarrow f \ [\mathit{freq}_{f,d}]$ and add the relation to the set-covering knowledge base.

If we have a small count of cases for the learning task, then the resulting profiles can be poor. For this reason we provide an editor for visual presentation and adaptation of the learned diagnostic profiles. Thus, the expert is able to inspect the single profiles in order to justify the threshold parameter $\mathcal{T}_{DP}$ or to possibly refine the profile by manually inserting additional findings or deleting unimportant ones.

3.4 Learning Weights of Parameters from Cases

Weights for parameters are another common knowledge extension for case-based reasoning systems. After the construction of diagnostic profiles, we will now describe how to learn weights of parameters. In general, the weight $w(p)$ of a parameter p specifies the importance of the parameter.

Learning Weights without additional Knowledge Our approach is inspired by a procedure mentioned in [18], when using the VDM method to discriminate the importance of attributes. However, our interpretation also considers additional information like abnormalities and structural knowledge.
A parameter p is defined to be important, if p has a high *selectivity* over the solutions contained in the case base CB. The degree of selectivity directly corresponds to the importance (weight) of the parameter. So, if different values of a parameter p indicate different diagnoses, then the parameter is considered to be *selective* for the diagnostic process.

We define the *partial selectivity* of a parameter p combined with a diagnosis d by the following equation

$$sel(p,d) = \frac{\sum\limits_{x,y \in dom'(p)} \left| \frac{N(p{=}x \mid d)}{N(p{=}x)} - \frac{N(p{=}y \mid d)}{N(p{=}y)} \right|}{\binom{|dom'(p)|}{2}} \tag{13}$$

where $x \neq y$ and $dom'(p) \subseteq dom(p)$ contains only values, that occur in cases $c \in CB$.
To compute the global *selectivity* of a parameter p, we average the partial selectivities $sel(p,d)$

$$sel(p) = \frac{\sum\limits_{d \in D^p_{rel}} sel(p,d)}{|D^p_{rel}|} \tag{14}$$

where $D^p_{rel} = \{\, d \in \Omega_D \mid \exists\, p \in \Omega_P, x \in dom(p) : \frac{N(p=x|d)}{|CB|} > \mathcal{T}_w \,\}$. So we only investigate the selectivities between parameters and diagnoses, whose combined frequency is larger than a given threshold $\mathcal{T}_w$.

Since $sel(p,d) \in [0,1]$ for all diagnoses $d \in D^p_{rel}$ and all parameters $p \in \Omega_P$, we see that $sel(p) \in [0,1]$ for all parameters $p \in \Omega_P$. The lower bound 0 is obtained, if parameter p has no selectivity over the diagnoses contained in Ω_D; the upper bound 1 is obtained, if p has a perfect selectivity over the diagnoses contained in Ω_D, i.e. each value $x \in dom(p)$ occurs either always or never with the diagnosis.

After determining the selectivity of the parameter, we use the following logarithmic conversion table to transform the numerical selectivity into a symbolic weight.

sel(p)		**w(p)**	**sel(p)**		**w(p)**
0	$\rightharpoonup$	G0	(0.08, 0.16]	$\rightharpoonup$	G4
(0, 0.02]	$\rightharpoonup$	G1	(0.16, 0.32]	$\rightharpoonup$	G5
(0.02, 0.04]	$\rightharpoonup$	G2	(0.32, 0.64]	$\rightharpoonup$	G6
(0.04, 0.08]	$\rightharpoonup$	G3	(0.64, 1.00]	$\rightharpoonup$	G7

We accept the loss of information to facilitate a user-friendly adaptation of the learned weights by the expert in a later step. So, similar to the diagnostic profiles, the weights can be adapted manually by the expert to refine the learned knowledge.

Utilizing Abnormalities for Learning Weights If there are abnormalities available for a given parameter p, then we can use this information to improve the learning algorithm. In this case we will adapt Equation 13 to consider only the selectivity between normal and abnormal parameter values.

$$sel(p,d) = \frac{\sum\limits_{x \in abnormal(p) \,\wedge\, y \in normal(p)} \left| \frac{N(p=x \mid d)}{N(p=x)} - \frac{N(p=y \mid d)}{N(p=y)} \right|}{\mid abnormal(p) \mid \cdot \mid normal(p) \mid} \tag{15}$$

where $abnormal(p) = \{\, x \in dom(p) \mid abn(x) \neq A0 \,\}$ is the set of values $x \in dom(p)$ representing an abnormal state, and $normal(p) = dom(p) \setminus abnormal(p)$.

Optimizing Parameter Weights by Structural Knowledge As mentioned in the introduction of this paper, we operate on a highly structured knowledge base, where parameters (questions) are arranged in sets called *examinations* (corresponding to the diagnostic tests during the clinical use).
Examinations contain an *examination weight* to mark their significance in the overall examination process. These weights help us to adjust the weights of the parameter contained in the examination. So parameters contained in dense examinations (i.e. containing many parameters) will receive a decreased weight, whereas parameters contained in sparse examinations with fewer parameters will obtain an increased weight. Thus, for a parameter p contained in examination E we will obtain an adjusted weight $w'(p)$ defined by Equation 16.

$$w'(p) = \frac{w(p)}{\sum\limits_{p' \in E} w(p')} \cdot w(E) \tag{16}$$

where $w(E)$ is the *examination weight* for examination E. The heuristic given in Equation 16 is motivated by the fact, that in the applied domain single phenomena are structured in single examinations. Thus, if the examination contains many parameters describing the phenomenon, then this examination is likely to contribute more weights than an examination with fewer parameters. Nevertheless, each examination only describes one phenomenon. It is worth mentioning, that this method is not reasonable in general, but can be used, when highly structured case bases are available.

3.5 Additional Knowledge used by the Learning Task

Now we will shortly summarize the processing facilities of additional knowledge (e.g. abnormalities, scalability, examination structure) during the learning task. We remark, that similarities, profiles and weights depend on each other, since we apply similarity knowledge for learning diagnostic profiles and utilize diagnostic profiles, when determining weights of parameters.

	no knowledge	***abnormalities***	***scalability * / examination structure ****
similarities	We apply the VDM method to compute similarities between each distinct value pair $v, v' \in dom(p)$.	We partition the value range $dom(p)$ into abnormal and normal values, and use the VDM method only within the partitions.	* Use VDM method only for adjacent values, normalize interpolated distance matrix. If abnormalities given, divide the value range into 2 partitions before comp. distances.
profiles	Similarities are required	Improve similarities for computing the diagnostic profile	* Improve similarities for computing the diagnostic profile
weight	Use a modified VDM method, that determines the weight of a parameter by its selectivity between the given diagnoses.	Adapt the modified VDM method, so that we only consider selectivities between normal and abnormal parameter values.	** Normalize weight of parameters w.r.t. the weight of their corresponding examination.

The table gives a brief overview of the adaptations we made to the presented learning methods in this section.

4 Dynamic Retrieval of Cases with Multiple Faults

In this section we describe an adapted retrieve process, following the notions of Aamodt and Plaza. In [19], they defined the case-based reasoning process as a cycle containing the following four sub-processes: *Retrieve*, *Reuse*, *Revise*, *Retain*. For handling multiple faults we need to adapt the *Retrieve* step in the following way:

Typically starting with a (partial) problem description of observed findings the retrieve process ends with a previously solved case best matching the given problem description.

In a first step we try to find a sufficient similar case, i.e. containing a sufficient similar problem description. We use a high threshold $\mathcal{T}_{CBR}$ of the required similarity to define *sufficient similarity*. If we have found a sufficient similar case, then we propose this case as a possible solution in the *Reuse*-step. If no sufficient similar case has been found, then we apply the following steps:

1. Use the transformed set-covering models to compute the k best hypotheses. A hypothesis is a set of diagnoses, that can explain the problem description of the new case, i.e. the observed findings.
2. Given the hypotheses we generate a set of *candidate cases*: A candidate case contains several cases, whose combined solutions yield the diagnoses of one of the hypotheses generated in the step above.
3. Compare each candidate with the new case c using Equation 2 and remove all candidates c' with $sim(c, c') < \mathcal{T}_{CBR}$.
4. Propose the three most similar cases as retrieved solution.

We combine single cases to a candidate case by 1) joining sets of the solutions contained in the single cases, and 2) joining problem descriptions of the cases. If the cases contain a parameter with different values, then we take the value with the highest abnormality. We motivate this approach with the following heuristic: If a patient has two (independent) diagnoses, then it seems to be reasonable, that the more severe finding will be observed. If there are no abnormalities defined, then we either try to take the value contained in the new case, if included in one problem description, or we take the value, which is most similar to the value listed in the new case.

5 Evaluation and Discussion

As mentioned in the introduction, we applied a real-world case base to our learning algorithms. The SonoConsult case base currently contains 744 cases, with a mean of diagnoses per case $M_d = 6.71 \pm 04.4$ and a mean of relevant findings $M_f = 48.93 \pm 17.9$ per case. For the evaluation of our experiments we adopted the *intersection accuracy* measure proposed in [9]. Let c be a new case and c' the retrieved case, that is most similar to c. Then the *intersection accuracy* $\mathcal{IA}$ is defined as follows

$$\mathcal{IA}(c, c') = \left(\frac{|D_c \cap D_{c'}|}{|D_c|} + \frac{|D_c \cap D_{c'}|}{|D_{c'}|} \right) / \, 2 \qquad (17)$$

where D_c is defined as the set of diagnoses contained in case c.

In the following table we present results of the experiments E0–E4 (all implementing leave-one-out cross-validation). For each experiment we incrementally applied additional background knowledge:

E0 Plain case comparison, no knowledge used.
E1 Predefined abnormalities in the knowledge base (*Abn*)
E2 Learned partial similarities (*PaSim*)
E3 Learned weights (*Weight*)
E4 For unsolved cases, dynamic candidate case generation based on learned diagnostic profiles (*CCG*).

Additional knowledge enables us to decrease the case similarity threshold $\mathcal{T}_{CBR}$ without receiving a dramatically decreased intersection accuracy of the solved cases. Cases below this threshold were withdrawn and marked as "*not solvable*", because no sufficiently similar case was found. For the experiments we applied two versions of the case base. The first one (CB_{raw}) contains only the raw data, whereas the second one (CB_{abs}) additionally contains findings gained by data abstraction. The abstracted data is inferred with rules based on expert knowledge, which is not available in typical case-based applications. As expected, it shows a significant increase in the number of solved cases and accuracy, but still is clearly insufficient to deal with the multiple fault problem.

	used knowledge / method					CB_{raw}		CB_{abs}	
	Abn	*PaSim*	*Weight*	*CCG*	*threshold* $\mathcal{T}_{CBR}$	*solved cases*	*mean accuracy*	*solved cases*	*mean accuracy*
E0	–	–	–	–	0.60	8	0.96	12	0.74
E1	+	–	–	–	0.55	14	0.81	19	0.77
E2	+	+	–	–	0.55	26	0.71	43	0.72
E3	+	+	+	–	0.50	29	0.75	74	0.71
E4	+	+	+	+	0.26	429	0.70	–	–

The small number of solved cases in the evaluations E0–E3 is justified by the special characteristic of our case base, which shows a high count of diagnosis combinations per case and rare repetitions of diagnosis combinations in the case base. The high variance of diagnosis combinations on the other hand causes a high variance of possible problem descriptions. The numbers in the table above show, that standard CBR-methods are performing poor for cases with multiple faults even when additional knowledge like similarities and weights is applied.

This conclusion strongly motivates the usage of set-covering techniques in the case-based process, which was evaluated in E4. Here we can see, that a dynamic candidate case generation can present similar cases for 429 cases. Figure 1 clearly shows the trend, that additional knowledge increase the number of solved cases. In Figure 2 we can see the intersection accuracy for the 40 most similar cases in E0–E4, which suggests the trend, that learned knowledge improves the quality of the solved cases. The results presented above are quite promising. Nevertheless, we see enhancements for the number of solved cases and intersection accuracy, when applying a more detailed integration of diagnosis hierarchies into the data preprocessing step and when assessing the intersection accuracy. In general, a well-elaborated data preprocessing step will increase the quality of the learned similarities, profiles and weights.

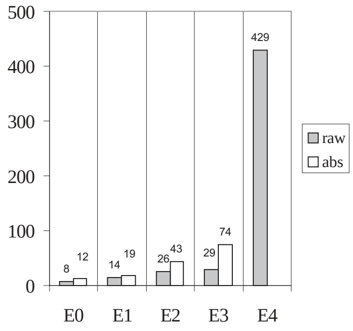

Fig. 1. Number of solved cases in experiments E0–E4

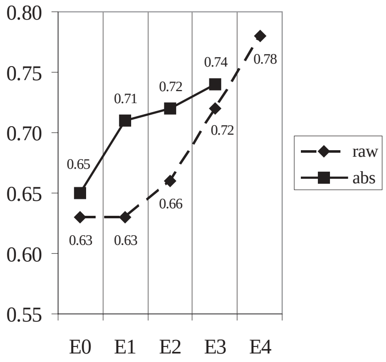

Fig. 2. Mean accuracy for the 40 most similar cases retrieved in E0–E4.

6 Summary and Future Work

In this paper we presented a new approach for handling multiple faults in case-based reasoning, describing inductive methods for learning similarities, weights and diagnostic profiles. We found diagnostic profiles to be very useful for handling multiple faults, since they can be combined to explain a new problem description, that had not been emerged before. We integrated this idea in a dynamic retrieval process, that does not leave the paradigm of case-based reasoning, since it always explains its presented solutions in parts of cases. For the inductive learning methods we pointed out, that additional knowledge can improve the resulting quality of the learned similarities, weights and profiles. In this case, the knowledge can be applied incrementally depending on its availability. Experiments have shown, that the dynamic candidate generation method using diagnostic profiles significantly improved the number of solved cases.

In the future, we are planning to consider more detailed adaptations of the presented learning methods. For example, preprocessing heterogeneous data or learning parameter weights still needs improvements. Besides the diagnostic profile generation process, we are currently working on an enhanced approach taking advantage of causal independencies between groups of diagnoses. Furthermore the case base is still growing due to the routine usage of the system in a clinical environment. An evaluation of the presented methods with a larger number of cases should yield better results.

References

1. Michael Richter. The Knowledge contained in Similarity Measures. Invited talk at ICCBR-95, http://www.cbr-web.org/documents/Richtericcbr95remarks.html, 1995.
2. Hans-Peter Buscher, Ch. Engler, A. Führer, S. Kirschke, and F. Puppe. HepatoConsult: A Knowledge-Based Second Opinion and Documentation System. *Artificial Intelligence in Medicine*, 24(3):205–216, 2002.
3. Frank Puppe. Knowledge Reuse among Diagnostic Problem-Solving Methods in the Shell-Kit D3. *Int. J. Human-Computer Studies*, 49:627–649, 1998.
4. Christoph Beierle and Gabriele Kern-Isberner. *Methoden wissensbasierter Systeme. Grundlage, Algorithmen, Anwendungen*. Vieweg, 2000.
5. Joachim Baumeister, Dietmar Seipel, and Frank Puppe. Incremental Development of Diagnostic Set–Covering Models with Therapy Effects. In *Proc. of the KI-2001 Workshop on Uncertainty in Artificial Intelligence*, Vienna, Austria, 2001.
6. Joachim Baumeister and Dietmar Seipel. Diagnostic Reasoning with Multilevel Set–Covering Models. In *Proc. of the 13th International Workshop on Principles of Diagnosis (DX-02)*, Semmering, Austria, 2002.
7. Phyllis Koton. Reasoning about Evidence in Causal Explanations. In *Proc. of the Seventh National Conference on Artificial Intelligence*, pages 256–261, 1988.
8. Luigi Portinale and Pietro Torasso. ADAPtER: An Integrated Diagnostic System Combining Case-Based and Abductive Reasoning. In *Proc. of the ICCBR 1995*, pages 277–288, 1995.
9. Cynthia A. Thompson and Raymond J. Mooney. Inductive Learning for Abductive Diagnosis. In *Proc. of the AAAI-94, Vol. 1*, pages 664–669, 1994.
10. Xue Z. Wang, M.L. Lu, and C. McGreavy. Learning Dynamic Fault Models based on a Fuzzy Set Covering Method. *Computers in Chemical Engineering*, 21:621–630, 1997.
11. Rainer Schmidt and Bernhard Pollwein and Lothar Gierl. Case-Based Reasoning for Antibiotics Therapy Advice. In *Proc. of the ICCBR 1999*, pages 550–559, 1999.
12. James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and Unsupervised Discretization of Continuous Features. In *Proc. of the International Conference on Machine Learning*, pages 194–202, 1995.

13. Dan Ventura and Tony R. Martinez. An Empirical Comparison of Discretization Methods. In *Proc. of the 10th Int. Symp. on Computer and Information Sciences*, pages 443–450, 1995.
14. Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Mateo, California, 2000.
15. Craig Stanfill and David Waltz. Toward Memory-Based Reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.
16. D. Randall Wilson and Tony R. Martinez. Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.
17. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, California, 1988.
18. Dietrich Wettschereck and David W. Aha. Weighting Features. In Manuela Veloso and Agnar Aamodt, editors, *Case-Based Reasoning, Research and Development, First International Conference*, pages 347–358, Berlin, 1995. Springer Verlag.
19. Agnar Aamodt and Enric Plaza. Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1), 1994.