A similarity measure for case based reasoning modeling with temporal abstraction based on cross-correlation

Florian Hartge[a,*], Thomas Wetter[a], Walter E. Haefeli[b]

[a] Institute for Medical Biometry and Informatics, Department Medical Informatics, Im Neuenheimer Feld 400, D-69120 Heidelberg, Germany

[b] Department Internal Medicine VI, Clinical Pharmacology and Pharmacoepidemiology, University of Heidelberg, Im Neuenheimer Feld 410, D-69120 Heidelberg, Germany

[*]Corresponding author:

Florian Hartge, MSc
Department Medical Informatics
Im Neuenheimer Feld 400
D-69120 Heidelberg
Germany

Phone: +49 6221 56-7461
Fax: +49 6221 56-4997
E-mail: florian.hartge@med.uni-heidelberg.de

Abstract

Adverse drug events (ADEs) are a major limitation of drug safety. They are often caused by inappropriate selection of dose and the concurrent use of drugs modulating each other (drug interaction). Risk assessment and prevention strategies must therefore consider co-administered drugs, individual doses, and their timing. In a new approach we evaluated the performance of cross correlation, commonly used in signal processing, to determine similarities in patient treatments. To achieve this, patient treatments were modeled as groups of vectors representing discrete time intervals. These vectors were cross-correlated and the results evaluated to find clusters in time courses indicating similarity in treatment of different patients.

To evaluate our algorithm, we then created a number of test cases. The focus of this article is on each treatment and its pattern in time and dosage. The algorithm successfully produces a relatively low similarity score for cases that are completely different with respect to their pattern of time and dosage but high scores when they are equal (score of 0.699) or similar (score of 0.528) in their therapies and thus succeeds in having a relatively high specificity (27/30). Such an approach might help to considerably reduce the problem of false alarms which hampers most existing alerting systems for medication errors or impending ADEs.

Key words – Case based reasoning, temporal abstraction, similarity measure, adverse drug events

1. Introduction

Adverse drug events (ADEs) are a major factor limiting safety and effectiveness of drug therapy in modern health care systems. For example in the USA an estimated 770,000 people are subject to an ADE during their stay in a hospital and about 140,000 of them die because of it [1]. The estimated annual costs are about $4 billion, of which about half may be avoidable [2].
Although not studied thus far there is no evidence to suggest that the problem is much different or substantially smaller in Germany. However, there have been some efforts in the last years to overcome this drawback [3 - 5]. Computerized physician order entry systems (CPOE), barcode identification, and automated dosing systems have been introduced to reduce the errors inherent to the process of administering medication (from the prescription process to dispensing) like misspelling, misreading, and confusion of medication or patient data. Another effective option is to support the physician during the prescription process with pertinent information and decision support on medical therapy [6]. This is the focus of this paper.

It is a common approach to warn the practitioner while he is entering the intended medication for a patient if there is an assumed adverse interaction with another drug or with the patient's state of health. This alert process is normally aided by the use of databases containing lists of interactions or other known risk situations [7]. Standard practice entails the computer to issue warnings to the user if the patient has drug X or condition Y (e.g. allergy) in the presence of which drug Z is known to likely cause an ADE. However, the specificity of such alert systems is limited because potentially harmful drug combinations do not result in an ADE in all instances.
This leads conventional warning systems to issue many false alarms. These warning systems have a rather good sensitivity but their poor specificity prompts practitioners to ignore them [8, 9] or to override alerts [10,11].
The aim of this ongoing research is to develop an electronic drug prescription system, which more selectively identifies risks associated with drug therapy, with more specific alerts and therefore better acceptance than the current systems. The proposed algorithm may be a key element in such a system since it will directly influence its user-friendliness.

2. Background

In most medical specialties the occurrence of ADEs is a complex process and the circumstances leading to a particular event are not always known in detail. While it is often possible to retrospectively identify an ADE it is much more difficult to point out exactly what leads to it, especially considering individual variability of important pharmacologic processes between different patients [12,13]. In such a relatively uncertain domain it is challenging to form sets of clear rules that identify ADEs. Therefore, case similarity suggests itself as an approach that may circumvent the problems of characterizing ADEs through sets of rules.

2.1Introduction to case based reasoning (CBR)

CBR is a sub-discipline of artificial intelligence that is based on the assumption of analogy. The main theorem of CBR is that if two problems are similar their solutions also are. The mechanics are therefore that a pool of problems and their respective solutions within a certain domain is collected. If then a new problem arises the pool is searched for the most similar problem. The solution of this already solved problem is then applied to the new problem at hand. The solution of the old problem may have to be slightly modified to the new needs (Figure 1).
CBR has some intriguing advantages. First it works well in domains where the knowledge is relatively rudimentary. Second, in order to build a system it is not necessary to query experts about their way of reasoning. Rather records of old cases such as progress notes or standardized charts can be used. Third it is easy to add new knowledge in the form of new cases or to adapt existing ones and, therefore, fourth, it is simple to manage and maintain an up-to-date base of knowledge [14].
In our system we want to collect a reference base of detailed medical cases with verified ADEs. If a new case arises in practice with striking similarity to a case in the database, then the system should alert the practitioner.
One problem with this approach is the complexity of modeling time elapsed and temporal interrelationships between exposure with different drugs and concurrently developing events. Most drug effects do not only depend on pharmacodynamics but are time dependent and often related to pharmacokinetics. Therefore, the temporal relationships between drug exposure of a patient and events are essential to assume similarity in different cases.
An earlier approach using CBR and temporal abstraction [15] made it possible to cover the feature space of a problem with a fixed set of predictable states this problem can reach. Therefore a case modeling of problems was developed which consists of a distribution of different states over time. This is only possible if a predefined set of attributes with conceivable value ranges can directly be mapped to the set of medical states. Obviously it is not possible to predefine every possibly occurring medical state during treatment. Therefore we aimed to find a different case modeling and similarity measure.

2.2 Introduction to cross correlation

We took a new approach in using cross-correlation to determine similarities in treatments. In signal processing, the cross-correlation function is commonly used to determine similarities between two time dependent signals. To determine cross-correlation between two signals we first have to define the correlation between data in general. Correlation between two groups of data implies that they move or change with respect to each other in a structured way. The correlation coefficient is an indicator for the strength and sense or direction of correlation. For N pairs of data (x, y) the correlation coefficient is calculated thus

$$r_{xy} = \frac{\frac{1}{N-1}\sum_{n=1}^{N}(x(n)-\bar{x})(y(n)-\bar{y})}{((\frac{1}{N-1}\sum_{n=1}^{N}(x(n)-\bar{x})^2)(\frac{1}{N-1}\sum_{n=1}^{N}(y(n)-\bar{y})^2))^{\frac{1}{2}}} \qquad (1)$$

The ‘active part’ of equation (1) is the enumerator summation which tends to zero if there is little common movement between x and y and approaches high positive and negative values depending on whether x and y tend to move together or in opposite senses. The denominator terms merely have a normalizing effect which delimits the range of the correlation coefficient to [-1,1] [16].

The correlation coefficient can be used to determine correlation between two sets of data. When measuring the correlation between two signals from two time series of discrete numbers it is possible that two signals may have common components but different timing. The cross-correlation function (equation (2)) is that function which is formed from successive values of the correlation coefficient taken at time shifts k = 1, 2, … n data sampling intervals. Different approaches are available to insert y values where the formula would reach beyond the interval covered through recorded values.

$$r_{xy}(k) = \frac{\frac{1}{N-1}\sum_{n=1}^{N}(x(n)-\bar{x})(y(n+k)-\bar{y})}{((\frac{1}{N-1}\sum_{n=1}^{N}(x(n)-\bar{x})^2)(\frac{1}{N-1}\sum_{n=1}^{N}(y(n)-\bar{y})^2))^{\frac{1}{2}}} \qquad (2)$$

The time shift at which the correlation coefficient reaches its highest value indicates the relative position in time where both signals are closest correlated. Therefore the cross-correlation function can be used to identify simple time delays between events in two signals. [17]

3. Design considerations and system description

3.1 Definition of terms

For the development of the model and the similarity algorithm the following terminology has been applied

(1) A *treatment* is the information about a distinct therapy with an active compound of a drug administered to a patient during his hospitalization (e.g. the fact of giving him aspirin).

(2) A *time and dosage pattern* is the chronological information about dose and dosing interval of a treatment (e.g. every day one 10mg tablet in the morning and two 10mg tablets in the evening).

(3) A *course of treatment* is the information of a *treatment* (1) combined with the corresponding *time and dosage pattern* (2).

(4) A *case* is the set of all *courses of treatment* (3) one patient receives during his stay. This information is used equivalent to a CBR case.

3.2 Case modeling

Our modeling of *cases* is simple. For each *treatment*, the model includes an identifier for the *treatment*, as well as a vector for the *time and dosage pattern* which consist of slots for discrete points in time where a treatment could have taken place and the particular dose that was given then (Figure 2). Per day, there are five time slots for drug administration (morning, noon, evening, bedtime, and night) in the present version of the model.

3.3 Traditional approach

A common approach to compare two different *cases* of two different patients would be to count the number of medically equivalent *treatments* (e.g. both patients received some sort of aspirin product). This is done by interleaving two list searches as described earlier [18]. Depending on the number of equivalent and different *treatments* in both *cases*, a similarity score can be determined according to Algorithm 1 (Figure 3). Because this algorithm omits the chronological information when those *treatments* occurred (e.g. two tablets in the morning and two in the evening) we call it a ‘no time’ algorithm.

3.4 New algorithm

In medicine, chronological information is an important element for judging the equivalence of different *cases*. For this reason we enhanced the ‘no time’ algorithm by correlating the *time and dosage pattern* of a *treatment* in one *case* with the *time and dosage pattern* of the equivalent *treatment* in another *case*. The presented time sensitive similarity algorithm consists of four sub-algorithms which are explained below.

In the first steps our algorithm does not differ from the ‘no time’ algorithm. We have two patient encounters and want to determine how similar they are in respect to their *cases*, which is achieved by comparison of the lists of *treatments* in both cases. Every pair of medically equivalent *treatments* from each *case* is marked and put aside in a separate list. The number of equivalent *treatments* is divided by the overall number of *treatments*. This number determines the first half of the similarity score (Algorithm 1 / Figure 3).

In this first part of the algorithm we want to ensure that the model’s similarity score accounts for both similarities and discrepancies. Cases that have matching pairs in two of thirty *treatments* should not be rated too similar even if these two have a very similar pattern in time because we aim to assess the general similarity of two cases in respect to their whole courses of drug therapy.

Next, the *time and dosage pattern* comparison begins. The temporal part of each *course of treatment* in each pair is cross-correlated. The result is a number that is subsequently normalized to a score between 0 and 1. As we are not interested in negative correlation 0 is assigned if formula (2) delivers a negative value. The correlation is calculated for every possible step of the time scale (Figure 4). The highest score indicates the relative distance in time where both *time and dosage patterns* have the highest similarity. This score is stored together with the corresponding relative time shift. The resulting process creates a list of *treatments*, their corresponding similarity scores, and the corresponding time lag at which the similarity occurred (Algorithm 2).

In the next step the algorithm parses this list for the relative position in time around which most of the high scores gather. It steps through the list building up a histogram over all possible relative positions in time. Thereafter the best point of co-occurrence is identified in this histogram by building a triangular weighted sum of surrounding values. This sum is shifted over the whole histogram. The time lag with the highest sum is taken as optimal (Algorithm 3).

From the *courses of treatment* which contribute to such a cluster, the correlation values are summed up and the whole sum is normalized. This normalized sum creates the second half of our similarity score (Algorithm 4).

Both halves of the score, one from the occurrences of treatments and the other from their temporal correlation, are summed together to create the overall similarity score.

3. Status Report

To evaluate our algorithm, we thoroughly created 62 abstract test cases. Our main focus at this time is on *treatments* and their corresponding *pattern in time and dosage.* Therefore, we just modeled the *courses of treatment* omitting other data such as diagnoses or demographic patient data. Those data may also determine the similarity of a medical condition and will thus be topic of future research.

Starting from one standard *case* consisting of 8 different drugs given at different times and dosages for 10 days we developed various ways of how other cases could be different or similar with respect to treatment characteristics. We modeled for example a set of 3 interdependent drugs applied in a short period of time as it is done e.g. before cardiac catheter examinations. Then this set was altered systematically in time, dosage, and drugs in several degrees of alteration. By this we evaluated the algorithms’ response to defined levels of difference. We identified twelve different categories of how cases can be similar or dissimilar (Table 1). These categories emerge from relating new and known case to each other in a plane defined by the following two axes. One of these axes is similarity in *treatment.* Cases similar on this axis have a similar list of *treatments* that were administered to the patient. Similarity on this axis ignores chronological information. The other axis represents similarity in *time and dosage patterns*. Cases similar on this axis have similar patterns of administration in time and dosage. Similarity here completely omits the information to which *treatments* the *time and dosage patterns* belong.

Each of the twelve categories contains a value corresponding to a medical assessment. Ideally an algorithm assigning a numerical similarity value to different cases in relation to a standard case should rate cases in accordance with the respective qualitative particular statements in Table 1. We designed a set of five to six test cases per category with 56 having 8, 4 having 7, and 2 having 6 courses of treatment each. The algorithm was applied to these test sets. The resulting mean values of overall similarity score are shown in Table 2. In this table, a number close to 1 means high similarity, a number closer to 0 means low.

We also applied the algorithm to a case collection of 882 detailed real medical cases of patients hospitalized in internal medicine treated with an average of 10.8 (0, 72) different drugs and hospitalized for a mean length of stay of 10.2 (1, 127) days. These cases were collected in an earlier project and were used to confirm that the algorithm suitably scales up to typical size sets of real data. However, because these cases were not classified as specified in Table 1 we did not use them to calculate any ratings. For each case the similarity to all other cases in the case base was calculated. Several iterations with different thresholds for similarity were made. At each iteration, random sample checks were made to evaluate whether the classification of cases with similar ratings appeared plausible. According to this evaluation of real cases a similarity threshold of 0.7 turned out to be an adequate score to determine whether two cases are sufficiently similar or not. Setting this same threshold for similarity at 0.7 the algorithm's sensitivity on the synthetic test cases was 32/32 and the specificity was 27/30.

Our algorithm needs an average of 576 milliseconds for the calculation of the similarity of one real medical case against the case base of 882 other cases. That is an average of 0.653 milliseconds per single case comparison. The computer on which this test was executed was equipped with an Intel Pentium 4 2.4 GHz processor and with 512 Mb RAM. The algorithm is implemented in Java 1.4 and run on a 1.4 runtime engine and a MS Windows 2000 operating system. The case base was preloaded from the database into Java managed objects before execution.

4. Lessons learned and future plans

As shown in Table 2, our algorithm successfully produces a relatively low similarity score for cases that are completely different with respect to their *pattern of time and dosage* but high scores when cases are equal (score of 0.699) or similar (score of 0.528) in *treatments*. This proves to be as intended, although the first score is suboptimal since the mean does not show that three of five test cases are rated above our threshold of 0.7. It therefore seems that our algorithm may be open for further weight optimization on the tradeoff between *treatment* and *time and dosage pattern*.

An application related problem that might arise is how to deal with standard therapies. During the course of treatment for a normal hospital stay for a specific diagnosis, many of the treatments are part of a standardized practice guideline or order set. Though the standardization in treatment plans will produce high similarity scores due to their predefined patterns of application, these are probably not associated with ADEs.

Another application related problem that might arise is whether our model is capable of representing pharmacokinetics appropriately. In our test we just modeled dosing and application intervals but for a better pharmacological conclusion it would be important to model the concentration of active compounds in the body. An approximation of this could be done by calculating estimated body concentrations from the time and dosage pattern combined with some rudimentary pharmacokinetic elimination function. This data could possibly be derived from pharmacological databases enhanced by calculation of altered degradation due to impaired kidney or liver function and co-medication (drug interactions).

While these are no problems inherent to the modeling itself and thus not the main concern this time they should be considered in future research to make such a system useful in a clinical setting.

Our aim was to develop a case modeling and an associated similarity measurement algorithm for a CBR system that is capable of rating time-dependent data with high sensitivity and specificity. The focus of our CBR test cases was on drug therapies in which timing is an essential part of any treatment plan and which determines the ultimate (adverse) effect of pharmacotherapy. In this first step we have developed a successful model and similarity algorithm for time-value patterns with a relatively high specificity as an essential component of a CBR based alerting system for drug related events.

5. References

[1] D.C. Classen, S.L. Pestonik, R.S. Evans, J.F. Loyd, et al. Adverse drug events in hospitalized patients, *JAMA*. 277 (1997) 301-306.

[2] D.W. Bates, N. Spell, D.J. Cullen, E. Burdick, et al. The costs of adverse drug events in hospitalized patients, *JAMA*. 277 (1997) 307-311.

[3] K. Chung, Y.B. Choi, S. Moon, Toward efficient medication error reduction: Error-reducing information management systems, *J Med Syst*. 27 (2003) 553-560.

[4] G.T. Schumock, V.P. Nair, J.M. Finley, R.K. Lewis, Penetration of mediation safety technology in community hospitals, *J Med Syst*. 27 (2003) 531-541.

[5] R.M. Mullner, Patient safety and medication errors, *J Med Syst*. 27 (2003) 499-501.

[6] D.F. Doolan, D.W. Bates, Computerized physician order entry systems in hospitals: mandates and incentives, *Health Aff*. 21 (2002) 180-188.

[7] G. Del Fiol, B.H.S.C. Rocha, G.J. Kuperman, D.W. Bates, P. Nohama, Comparison of two knowledge bases on the detection of drug-drug interactions, *JAMIA*. Annual Symposium (2000) 171-175.

[8] D. Magnus, S. Rodgers, A.J. Avery, GP's view on computerized drug interaction alerts: questionnaire survey, *J Clin Pharm Ther*. 27 (2002) 377-382.

[9] P. Glassman, B. Simon, P. Belperio, A. Lanto, Improving recognition of drug interactions: Benefits and barriers to using automated drug alerts, *Med Care*. 40 (2002) 1161-1171.

[10] S.N. Weingart, M. Toth, D.Z. Sands, M.D. Aronson, R.B. Davis, R.S. Phillips, Physicians' decisions to override computerized drug alerts in primary care, *Arch Intern Med*. 163 (2003) 2625-2631.

[11] T. C. Hsieh, G. J. Kuperman, T. Jaggi, P. Hojnowski-Diaz, J. Fiskio, D. H. Williams, D. W. Bates, T. K. Gandhi, Characteristics and consequences of drug allergy alert overrides in a computerized physician order entry system, *JAMIA*. 11 (2004) 482-491.

[12] G. Levy, W.F. Ebling, A. Forrest, Concentration- or effect-controlled clinical trials with sparse data, *Clin Pharmacol Ther*. 56 (1994) 1-8.

[13] K. Sugimoto, M. Ohmori, S. Tsuruoka, K. Nishiki, A. Kawaguchi, K. Harada, M. Arakawa, K. Sakamoto, M. Masada, I. Myamori, A. Fujimura, Different effects of St John's Wort on the pharmacokinetics of simvastatin and pravastatin, *Clin Pharmacol Ther*. 70 (2001) 518-524.

[14] L. Gierl, D. Steffen, D. Ihracky, R. Schmidt, Methods, architecture, evaluation and usability of case-based antibiotics advisor, *Comp Meth Prog Biomed*. 72 (2003) 139-154.

[15] R. Schmidt, L. Gierl, Case-based reasoning prognosis for temporal courses. Computational intelligence techniques in medical diagnosis and prognosis, *Studies in Fuzziness and Soft Computing, Vol. 96*, (Springer-Verlag, Berlin, 2001) 101-128.

[16] D. Freedman, R. Pisani, R. Purves, *Statistics third edition* (W.W. Norton & Company Inc., New York, 1998).

[17] R.E. Challis, R.I. Kitney, Biomedical signal processing (in four parts) Part 1 Time-domain methods, *Med Biol Eng Comput*. 28 (1990) 509-524.

[18] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, 1999).

[19] A. Aamodt, E. Plaza, Case-based reasoning: Foundational issues, methodological variations, and system approaches, *AICOM*. 7 (1994) 39-59.

Tables

Table 1: Categories of cases and how they should be classified

	Time and dosage pattern			
Treatments	equivalent	equivalent but shifted in time	similar	different
equivalent	same	same	similar	different
similar	similar	similar	similar	different
different	different	different	different	different

Same: The cases should be rated very similar and shifts in the timing of drug therapies should not affect the classification. Courses of treatments consisting of equivalent treatments applied in the same temporal constellation to each other belong to this class.

Similar: The rating should indicate a similarity but it should reflect that the cases do have differences. Courses of treatments that have some groups of treatments and their respective temporal constellation in common belong to this class.

Different: The rating should indicate that both cases differ in critical aspects. Courses of treatments that differ substantially either in their type of treatment or in their respective temporal constellation belong to this class.

Table 2: Assigned similarity in the mean by new algorithm

Time and dosage pattern

Treatments	equivalent	equivalent but shifted in time	similar	different
equivalent	0.923	0.998	0.959	0.699
similar	0.875	0.910	0.877	0.528
different	0.475	0.396	0.464	0.195

Figure 1

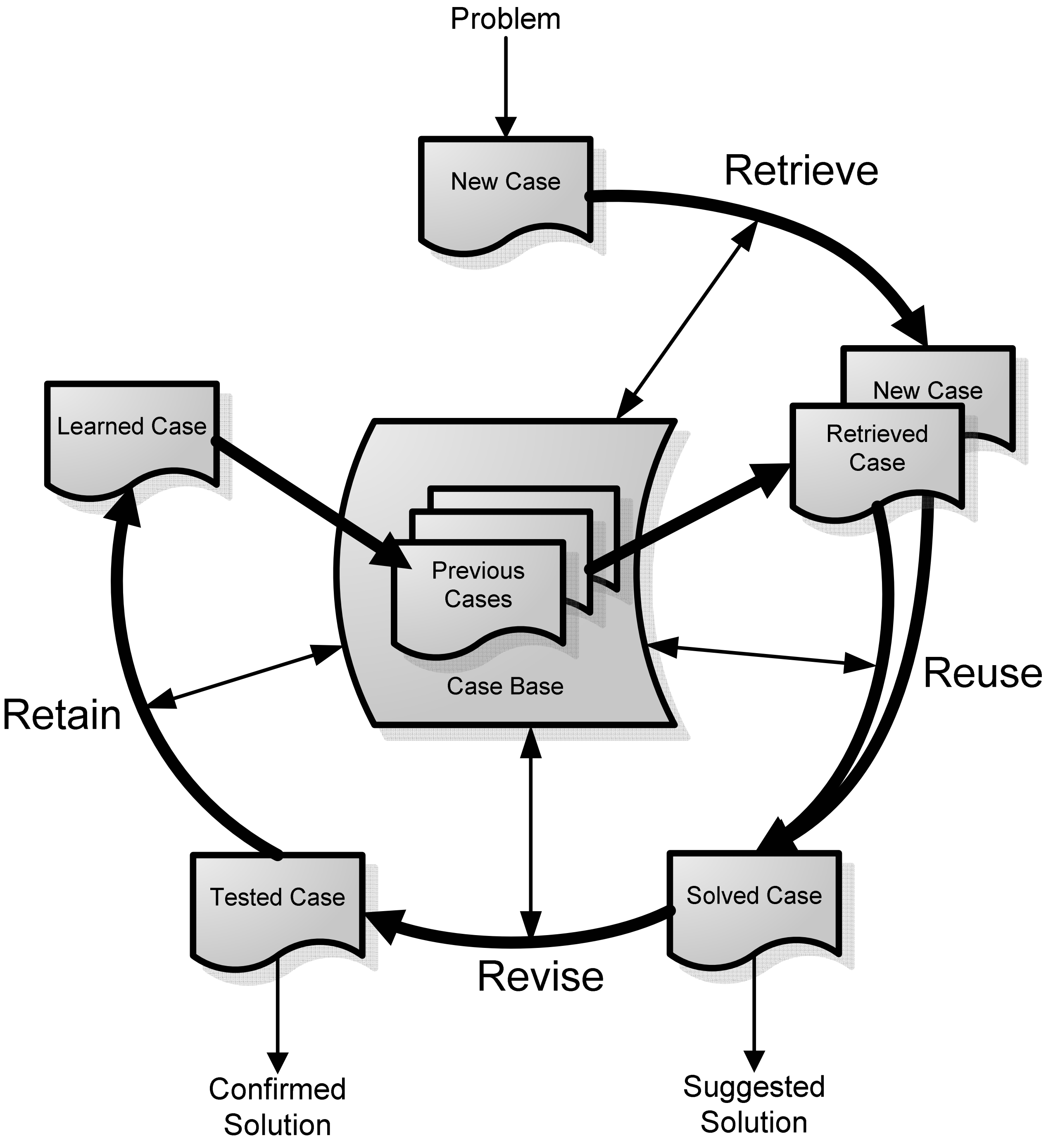

Problem
New Case
Retrieve
New Case
Retrieved Case
Learned Case
Previous Cases
Case Base
Reuse
Retain
Tested Case
Solved Case
Revise
Confirmed Solution
Suggested Solution

Figure 2

	day 1					day 2					day 3			
aspirin		1					2					1		
fluoxetine	1		2			1		2			1		2	
	mo	no	ev	be	ni	mo	no	ev	be	ni	mo	no	ev	be

Figure 3

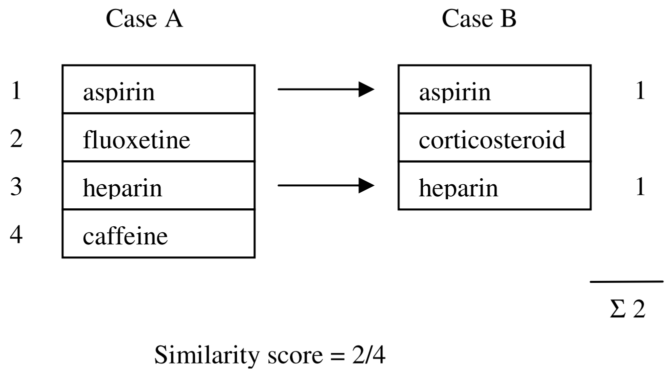
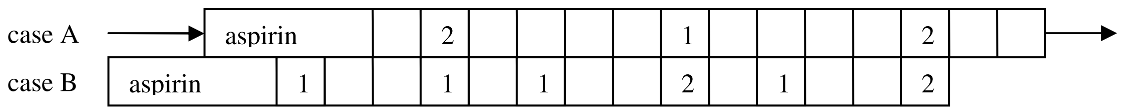

Case A
Case B
1
2
3
4
aspirin
fluoxetine
heparin
caffeine
aspirin
corticosteroid
heparin
1
1
Σ 2
Similarity score = 2/4

Figure 4



case A		aspirin			2					1					2		
case B	aspirin	1			1		1			2		1			2		

Figure captions

Figure 1:
Case based reasoning (CBR) cycle [modified after 19]. A new problem arises and is compared to a collection of historic problems whose solutions are already known. The most similar historic problem and the respective solution are retrieved. By adapting the historic solution to the new problem the historic knowledge is reused to suggest a new solution. The suggested solution is revised for its appropriateness and finally retained for usage as new historic knowledge to help solving future problems.

Figure 2:

Course of treatments: One course of treatments in one case of one patient is modeled by vectors of applied doses for each respective treatment. The fields of these vectors model a temporal abstraction of preselected granularity over the duration of the case. The figures inside the boxes represent the given doses at the respective point of time (mo: morning, no: noon, ev: evening, be: bedtime, ni: night).

Figure 3:
'No time' algorithm, basic algorithm that calculates similarity in treatments in cases without taking time into account. Courses of treatments of two different hospitalizations are compared by counting the equivalent treatments in each and dividing the resulting number by the maximum number of treatments of both courses.

Figure 4:
Two time and dosage patterns of two equivalent treatments from two different cases are cross-correlated to determine the temporal deviation at which their statistical similarity is highest.

Algorithms

```
Algorithm 1
Input
   listA: a list of all courses of treatment occurring in case A
   listB: a list of all courses of treatment occurring in case B
Output
   res: a similarity score
   // The two following are not an output of the normal algorithm. They are needed for further
   // use when this algorithm is used as first step of our new approach.
   slistA: a list of all courses of treatment occurring in both cases with the time patterns of case A
   slistB: a list of all courses of treatment occurring in both cases with the time patterns of case B
Begin
   // The class Treatment carries all information of a course of treatment
   Treatment treA, treB;
   // building up both lists
   for (i = 0; i < listA.size(); i++) {
       treA = (Treatment) listA.get(i);
       for (j = 0; j < listB.size(); j++) {
           treB = (Treatment) listB.get(j);
           if (treA.name.equals(treB.name)) {
               // Adding to the lists
               slistA.add(treA);
               slistB.add(treB);
           }
       }
   }
   // determining the longer list of treatments
   llen = (listA.size() < listB.size() ? listA.size() : listB.size());
   // since both ‘tl’ lists have the same length it does not matter which we take for numerator
   tllen = tlA.size();
   // calculating the similarity score
   res = (tllen / (2 * llen));
End
```

```
Algorithm 2
Input
  slistA: a list of all treatments occurring in both cases with the time pattern of case A
  slistB: a list of all treatments occurring in both cases with the time pattern of case B
Output
  best: list of best correlations and their respective relative position in time
Begin
  Treatment treA, treB;
  // checking for the longer length of time pattern
  treA = (Treatment) slistA.get(0);
  treB = (Treatment) slistB.get(0);
  tplen = (treA.timePattern.length < treA.timePattern.length ? treB.timePattern.length :
          treA.timePattern.length);
  // best is a two dimensional array holding the correlation score in the first axis
  // and the position in the second
  for (i = 0; i < slistA.size(); i++) {
    treA = (Treatment) slistA.get(i);
    treB = (Treatment) slistA.get(i);
    // Correlating both time pattern over the full length
    corr  = crossCorrelation(treA.timePattern, treB.timePattern, tplen);
    // checking if the actual correlation score is the highest by now
    // and if it is then replacing the old one
    for (j = 0; j < corr.length; j++) {
    if (best[i][0] < corr[j]) {
    best[i][0] = corr[j];
    best[i][1] = j;
  }
End
```

```
Algorithm 3
Input
    best: list of best correlations and their respective relative position in time
Output
    maxhist: central position of the maximal cluster within the histogram
Begin
    // building histogram of best correlations
    for (i = 0; i < best.length; i++) {
        histo[best[i][1]]++;
    }
    // searching for the maximal cluster within the histogram
    maxhist = 0;
    currsum = 0;
    // stepping along the histogram
    for (i = 0; i < histo.length; i++) {
        tmpsum = 0;
        // summing up the area of the cluster
        for (j = -2; j < 3; j++) {
            actpos = i + j;
            // checking if the current focus is within bounds
            if (actpos < 0) {
                tmpval = 0;
            } else if (actpos > (histo.length - 1)) {
                tmpval = 0;
            } else {
                // if it is summing it up
                // histoweights has the structure of histoweights = {0.4, 0.8, 1.0, 0.8, 0.4}
                tmpval = histo[actpos] * histoweights[j + 2];
            }
            tmpsum = tmpsum + tmpval;
        }
        if (tmpsum > currsum) {
        maxhist = i;
        currsum = tmpsum;
    }
End
```

```
Algorithm 4
Input
   best: list of best correlations and their respective relative position in time
   maxhist: central position of the maximal cluster within the histogram
Output
   res: second half of the similarity score
Begin
   tmpsum = 0;
   count = 0;
   for (i = 0; i < best.length; i++) {
      if ((maxhist - 2 <= best[i][1]) && (best[i][1] <= maxhist + 2)) {
         tmpsum = tmpsum + best[i][0];
         count++;
      }
   }
   res = tmpsum / (2 * count);
End
```