ELSEVIER

Knowledge-Based Systems 15 (2002) 391–398

Knowledge-Based SYSTEMS

www.elsevier.com/locate/knosys

Formalising optimal feature weight setting in case based diagnosis as linear programming problems

Lu Zhang*, Frans Coenen, Paul Leng

Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK

Received 25 July 2001; accepted 7 January 2002

Abstract

Many approaches to case based reasoning (CBR) exploit feature weight setting algorithms to reduce the sensitivity to distance functions. In this paper, we demonstrate that optimal feature weight setting in a special kind of CBR problems can be formalised as linear programming problems. Therefore, the optimal weight settings can be calculated in polynomial time instead of searching in exponential weight space using heuristics to get sub-optimal settings. We also demonstrate that our approach can be used to solve classification problems.

Keywords: Case based reasoning; Feature weight; Linear programming

1. Introduction

Case based reasoning (CBR) is a multi-disciplinary subject that focuses on the reuse of experiences [2]. Typically, a CBR approach retains a fairly large number of previous experiences (which are usually called cases) in a database (which is usually called the case base). When a new problem occurs, it will be represented as a new case and compared to the cases in the case base. Thus, the cases similar to the new cases will be used to suggest to users a solution for the new problem. Usually, the solved new case will also be added into the case base. The underlying assumption for CBR is that similar cases will have similar solutions. Many CBR approaches have been reported for solving problems in various domains, such as planning [11], design [14], software engineering [4,42], image processing [21,40], and diagnosis [8,31,49].

Cases can be represented in various forms. In this paper, we focus on the well-structured form. Other forms of case representation include *directed acyclic graphs* [30] and *examplars* [9,41]. A well-structured case is defined as a vector of features: $x = \{x_1, x_2, \ldots x_q\}$, where q is the number of features. Therefore, an algorithm such as k-nearest neighbour (k-NN) [18,22] can be exploited through calculating the distance between case $x = \{x_1, x_2, \ldots x_q\}$ and case $y = \{y_1, y_2, \ldots y_q\}$ using a distance function. However, a universal distance function may not be suitable for solving problems in different domains. One solution is to investigate multiple distance functions [55,56]. The other is to parameterise the distance function with feature weights [3,53]. For a feature weighting approach, a main research topic is to determine feature weight settings (see e.g. [16,32,34,48,54]).

In particular, CBR approaches can also be applied to diagnosis problems. In a diagnosis problem, the solution to a case is a set of faults, which is usually easy to be compared with the set of faults of another case. For two cases whose solutions have been found, the similarity in each feature, the calculated overall similarity, and the real similarity between the two sets of faults can be obtained. However, when diagnosing a new case, only the similarity in each feature and the calculated overall similarity between the new case and an existing case can be obtained. The calculated overall similarity is used to predict the fault similarity.

Linear programming (LP) [17] has been a rapidly developing mathematical discipline since it started in 1947. Theoretically, LP problems have been proved to be polynomial time solvable [26,27]. Practically, the simplex algorithm [17] and its derivatives and the interior point algorithm [26] and its derivatives are fast enough to be used in real world applications. Therefore, many realistic problems that have been expressed as LP problems have been well solved, and there is also a lot of commercial or free software for LP problems available from vendors and/or on the Internet.

* Corresponding author. Tel.: +44-151-794-3792; fax: +44-151-794-3715.

E-mail addresses: lzhang@csc.liv.ac.uk (L. Zhang), frans@csc.liv.ac.uk (F. Coenen), phl@csc.liv.ac.uk (P. Leng).

0950-7051/02/$ - see front matter
PII: S0950-7051(02)00023-0

In this paper, we demonstrate that optimal feature weight setting in a general form of case based diagnosis can be formalised as LP problems. Therefore, available LP software can be used to solve case based diagnosis problems.

The organisation of the remainder of the paper is as follows. Section 2 gives a brief overview of LP problems, and presents a general form of case based diagnosis. Section 3 demonstrates that calculation of optimal initial weights can be formalised as an LP problem. Section 4 demonstrates that calculation of new case weights can also be formalised as an LP problem. Section 5 provides some preliminary empirical results. Section 6 further discusses some related works of our approach. Section 7 concludes this paper.

2. Preliminaries

2.1. Linear programming problem

In general, an LP problem is to calculate the maximum or minimum value of a linear combination of a set of variables subject to a set of linear equations and/or linear inequalities as constraints. Therefore, an LP problem can be represented in the following form [12]:

maximise or minimise

$$\sum_{j=1}^{n} c_j x_j$$

subject to Constraint_i $(i = 1, 2, \ldots m)$

Constraint_i is of one of the three forms:

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \text{ or} \tag{1}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \text{ or}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

$$x_j \geq 0 (j = 1, 2, \ldots n)$$

In Eq. (1), $x_1, x_2, \ldots x_n$ are the *decision variables*; $c_1, c_2 \ldots c_n$ are the *cost coefficients*; $\sum_{j=1}^{n} c_j x_j$ is the *objective function*; $b_1, b_2, \ldots b_m$ are the *right-hand-side constants*; n is the number of decision variables; and m is the number of *constraints*.

There have been many efficient algorithms for LP problems reported in the literature. Algorithms that can solve the problem in Eq. (1) in polynomial time of $m + n$ have been reported in the literature (see e.g. [26,27]). Although the *simplex algorithm* [17] has been proved to be exponential in the worst case [29], there is strong empirical evidence suggesting that it typically takes $O(m + n)$ time to solve the problem in Eq. (1) [12]. In Ref. [37], it is also demonstrated that some interior point algorithms are even faster than the simplex algorithm when dealing with problems of large sizes. However, a thorough theoretical analysis of the exact complexity of interior point algorithms is still on its way. Other material concerning the complexity of algorithms for LP problems can be found in Refs. [5,35].

2.2. A general form of case based diagnosis

A feature weighting approach to the case based diagnosis problem can be summarised as follows. Every case is fully structured as a vector of features. Given a case x in the case base, we can obtain the similarity between x and any other case in any feature. Therefore, we can calculate the overall similarity between x and the other case as a linear combination of the similarity in each feature. Supposing there are q features in a case, the overall similarity between x and the other case i can be calculated as formula (2) [51].

$$\sum_{k=1}^{q} S_{xik} W_{xk} \tag{2}$$

In Eq. (2), S_{xik} is the similarity between case x and case i in feature k and W_{xk} is the weight for case x in feature k. Each weight for one case represents the contribution of the similarity of the corresponding feature to the overall similarity. Informally, we would say that W_{ik} represents the influence of feature k in using case i to diagnose another case. When diagnosing a new case t, we will calculate every overall similarity between any case x in the case base and t. The most similar l cases will be used to suggest the faults for t. After diagnosis, case t may be also added into the case base to diagnose future cases.

We call this form of case based diagnosis a general form; because we assign each feature in each case a weight, rather than assigning each feature a weight for all the cases. This distinction was previously referred as *local weighting* (i.e. different feature weights for different cases) and *global weighting* (i.e. same feature weights for all cases) [23,54]. In this paper, it is not our aim to discuss which form is preferable. We use the general form only because global weighting is a simplification of local weighting for our method.

In the general form, the calculated overall similarities rely heavily on the weights. Therefore, to calculate the initial weights for the training cases and to calculate the weights for the new case to be added into the case base are two main tasks, which will be discussed in Sections 3 and 4.

3. Calculation of initial set of weights

3.1. The problem

There are n training cases in the case base, each having a value in each of q features. For any two cases, the similarity between the two cases in each feature and the real similarity between the two cases can both be obtained. The overall