University of Auckland

CS760 Datamining and Machine Learning

Assignment 1

Case-based reasoning - Travel Case

Author:
Joel Glemne

Student-id:
145076454

September 9, 2016

Abstract

Your abstract.

1 Introduction to CBR

This section is divided into three sections; one giving an everyday life example with the purpose of giving an intuitive understanding of case-based reasoning (**CBR**), one section with a more theoretically general description followed by a mathematical description of the global similarity metrics.

1.1 Intuitive approach

In your everyday life you meet many small problems that you probably would consider quite simple. Let us imagine that you are at your boyfriend's house for the first time, having dinner with his family. The mother asks you to go fetch one more plate. What do you do?

If it was me, I would first go to the kitchen. Then, I would start to look around if I see any *obvious* place for the plates to be (be sure to notice the word *obvious* for later). After that, I would start looking in the cabins. What is then funny, is that the possibility of me finding the right cabin on the first try would be high. How is that possible?

The reason why, is that we often base our decisions on previous *experiences* and the outcome of those experiences. I have been to a lot of kitchens before and it is common that different families - at least families from the same culture - organize their kitchens in similar ways. What I did, was simply evaluating (**reasoning**) my **base** of previous experiences (**cases**) in order to find a similar solution for the given problem.

1.2 General approach

CBR is mainly a technique for solving a large varieties of problems. Though, to fully understand the concept of CBR it is important to explain the meanings of *case-base* and *reasoning*.

A *case-base* is a collection of experienced solutions to similar problems. *Reasoning* is the procedure of drawing conclusions based on previous cases in order to solve a given problem. It is, however, important to note that it is not necessary to find an exact replica of the considered problem in order to

find a solution, but only a *similar* problem. What is then important in the CBR methodology is how to decide what *similar* means.

In the example in section 1.1 the word *obvious* was used. What obvious means is basically that the given case problem is very similar to a, or several, previous experienced case(s). For humans, it might seem easy to compare different cases and identify all key *features* in an instant, but for a computer it is crucial to pre-identify key features, define and assign similarity metrics to each feature and then organize everything into a reusable model for comparing different cases and predict outcome.

1.3 Mathematical approach

The similarity between two cases is calculated according to the k-nearest neighbors algorithm explained here below.

Assume a case-base of n cases y_i where $i = 1, 2, ..., n$ and each case having the same m pre-defined features with different values. For each feature a *local* similarity function sim_j is defined where $j = 1, 2, ..., m$. The functions takes two arbitrary cases a, b as arguments so that $0 \leq sim_j(a, b) \leq 1$. Every feature is also assigned a weight ω_j which corresponds to the *local* simililarity's importance to the *global* similarity.

When comparing an arbitrary **target case** x with all the **source cases** in the assumed case-base, n *global* similarities S_i are produced where $0 \leq S_i \leq 1$. S_i is defined as

$$S_i = \frac{s_i}{W}$$

where

$$s_i = \sum_{1}^{j=m} (\omega_j * sim_j(x, y_i))$$

and

$$W = \sum_{1}^{j=m} \omega_j$$

In order to find the most similar cases in the case-base, the cases y_i corresponding to the highest global similarities S_i are chosen.

One of the most interesting things in this algorithm is how the local similarity functions sim_j are defined individually. This is what is handled in the next section of this report.

2 Features

In this section, all the local similarity functions for the given case-base are presented and motivated.

As described in section 1.3, every case has a couple of pre-defined *features* or *variables*. In this application the cases considered are different options for travel plans and they all have the following features:

1. Accommodation
2. Case name
3. Duration
4. Holiday type
5. Hotel
6. Journey code
7. Number of persons
8. Price
9. Region
10. Season
11. Transportation

Every feature has its own local similarity function, producing values within a range of $[0, 1]$, and an assigned weight within the range $[1, \infty[$. If a feature is considered important for the global similarity calculation, it is assigned a high value and vice versa.

2.1 Accommodation

The possible values for the accommodation feature were:

1. HolidayFlat
2. OneStar

3. TwoStars
4. ThreeStars
5. FourStars
6. FiveStars

2.1.1 Local similarity function

To calculate the local similarity of accommodation between two arbitrary cases, some sort of ranking was here needed. In this case, HolidayFlat was considered the least exclusive kind of accommodation and thereafter rising with number of stars. All of the feature values were therefore assigned an integer value I_{acc} corresponding to above given table numbers, e.g. for an arbitrary case a which has the feature value TwoStars, $I_{acc}(a) = 3$.

In order to produce a similarity sim_{acc} so that $0 \leq sim_{acc} \leq 1$, it was here assumed a *more-is-perfect* approach with a linear calculation of the similarity. This means, that if $I_{acc}(target\,case) \leq I_{acc}(source\,case)$, then $sim_{acc} = 1$. Otherwise, the similarity is calculated as

$$sim_{acc} = \frac{range - (I_{acc}(target\,case) - I_{acc}(source\,case))}{range}$$

where $range = 6 - 1 = 5$.

2.1.2 Feature weight

When considering a vacation option, the accommodation type is normally considered a feature which gives an extra bonus to the experience rather than being a feature vital for the selection. Therefore, the accommodation feature is given a relatively low weight of 3.

2.2 Case name

Since the case name by definition is unique for each case, e.g. *Journey984*, and really does not give any information about the case, this feature is not considered in the similarity metrics. It is however interesting to save it in the database since it is then very easy to search for the given case.

2.3 Duration

Duration of a vacation is most of the time adjusted according to the number of vacation days you are given by your employer, which makes this feature quite important for the selection of vacation. The more interesting discussion is rather how adaptable this feature is, a discussion more thoroughly dealt with in an own section here below; *Adaptation discussion*.

2.3.1 Local similarity function

If the duration in the given target case, $D(target\,case)$, is exactly the same as the one in the source case, $D(source\,case)$, the similarity sim_{dur} is set equal to 1. For other source cases, where the difference between the compared cases durations are less than or equal to 5, the similarity is calculated as:

$$sim_{dur} = \frac{5 - (D(target\,case) - D(source\,case))}{5}$$

For cases where the difference is larger than 5 days, $sim_{dur} = 0$.

2.3.2 Feature weight

Since this feature is considered to be fairly adaptable, the feature is given a correspondingly low weight of 2.

2.3.3 Adaptation discussion

In this application, it is supposed that the duration could be adapted by the accommodation provider but that is rather reflected in how the price similarity is calculated. For more info, see the feature section *Price*.

2.4 Holiday type

The holiday type has a couple of different possible values, all given in one of the lecture slides. In this application all these different types are entered into a taxonomy tree according to Figure 1 here below.

2.4.1 Local similarity function

What the tree says, is that if you for example are comparing *Diving* with *Surfing*, the similarity will be equal to 0.5, but if you are comparing *Diving*

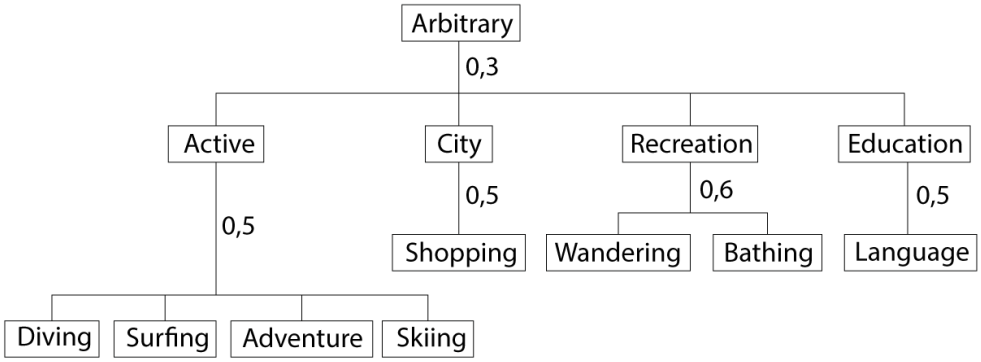

Figure 1: Taxonomy tree for holiday types

with for example *Shopping* or any other feature value outside of the same group, the similarity will be 0.3.

2.4.2 Feature weight

With the exception of hotel name and journey code, holiday type is in this program considered to be the most important feature and is given the high value of 10. The reasoning behind this is that one normally first consider what the purpose or main activity during the trip will be and then consider place and everything else. These assumptions are based on personal common knowledge and very informal interviews with friends.

2.5 Hotel

This feature is simply the name of the hotel given in the case.

2.5.1 Local similarity function

Since the hotel names really does not say anything about the standard of the hotel or any other features, this is considered a fairly unimportant feature for one who considers different options for a vacation and does not have a certain hotel in mind. That is why if the exact hotel name is entered, then the similiarity will be equal to 1, otherwise it will be equal to 0.

2.5.2 Feature weight

Would the case be that one has heard that a certain hotel is considered to be really good or if one has stayed at a certain hotel before, than perhaps it is likely that the hotel name will be requested. Would that be the case, it certainly should be prioritized when actually entering the name of the hotel. It has therefore been given the high weight of 20.

2.6 Journey code

The journey code is only an integer, mostly used as an id of the case. Would the case be that one is searching for a specific case, the similarity will be 1 if the query is the same, otherwise it will be 0. Since this only would be used to find a single certain case, the weight is assigned the extreme value of 200, just to be sure that the certain case is found.

2.7 Number of persons

The feature *Number of persons* is exactly what it sounds like. This is most of the time something that is not really adjustable when considering different vacation options, but it is at many hotels adaptable so that it may suit the customer. As said before, the thing one really is looking for is the price per person per day. This will be covered in the feature section *Price*.

2.7.1 Local similarity function

The similarity for number of persons is calculated in the exact same way as the duration.

2.7.2 Feature weight

With the same reasoning as with duration, weight also here is set to 2.

2.8 Sections

Use section and subsection commands to organize your document. LaTeX handles all the formatting and numbering automatically. Use ref and label commands for cross-references.

2.9 Comments

Comments can be added to the margins of the document using the todo command, as shown in the example on the right. You can also add inline comments too:

This is an inline comment.

Here's a comment in the margin!

2.10 Tables and Figures

Use the table and tabular commands for basic tables — see Table **??**, for example.

2.11 Mathematics

LaTeX is great at typesetting mathematics. Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with $\mathrm{E}[X_i] = \mu$ and $\mathrm{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i}^{n} X_i$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

2.12 Lists

You can make lists with automatic numbering ...

1. Like this,
2. and like this.

... or bullet points ...

- Like this,
- and like this.

We hope you find writeLaTeX useful, and please let us know if you have any feedback using the help menu above.

References

[1] Michael M. Richter & Rosina O. Weber, *Case-Based Reasoning*, Springer-Verlag Berlin Heidelberg, 2013.