Gaining Insight through Case-Based Explanation

Conor Nugent, Dónal Doyle and Pádraig Cunningham
Computer Science, Trinity College, Dublin 2, Ireland

December 2, 2004

Abstract

Because CBR is an interpretable process, it is a reasoning mechanism that supports explanation. This can be done explicitly by the system designers incorporating explanation patterns in cases. This can be termed knowledge-intensive explanation in CBR. However, of more interest here is case-based explanation that works by allowing users to consider the relation between different cases. The recommendation of a decision support system can be explained by presenting similar cases that motivate the recommendation. Users can derive insight from similar cases that have different outcomes. The differences in outcome are due to the differences in the un-matching features (provided the effect is not due to noisy data). This is a more knowledge-light approach to case-based explanation. This is appropriate for weak-theory domains where the details of the causal interactions in the domain are not well understood; experts would however be able to express the *direction* of causal interactions. In this paper we present such a knowledge-light framework for Case-Based Explanation.

1 Introduction

It is useful to divide work on explanation in CBR into two categories (Cunningham et al., 2003). There is the knowledge intensive approach where explanations are produced using explanation patterns that have been encoded in cases (Kass and Leake, 1988; Armengol et al., 2001). And there is the knowledge light approach where explanation is achieved by revealing the case descriptions to the user. A useful analysis of CBR from a knowledge perspective is Richters knowledge containers model (Richter, 1998). This model identifies that knowledge in a CBR system can reside in one of four containers:

- **Vocabulary:** the attributes and predicates that can be used to describe the cases.
- **Similarity Knowledge:** the knowledge that is used during case retrieval to identify similar cases.
- **Adaptation Knowledge:** the knowledge that is used to transform retrieved cases (if required) to conform to the problem under investigation.
- **Case-Base:** the cases themselves that describe examples or episodes.

The fact that CBR systems have this structure is important from an explanation perspective because it allows the user to consider the relation between cases (Roth-Berghofer, 2004). The CBR idea is based on examples and a means of assessing the similarity or differences between examples. Thus the recommendation of a decision support system can be explained by presenting similar cases that motivate the recommendation. In addition, users can derive insight from similar cases that have different outcomes.

The extent to which a CBR system is useful for insight and knowledge discovery depends to some extent on the approach to case acquisition. There are two models for case acquisition in CBR:

- **Gold standard cases:** cases are carefully selected/crafted (often by a committee) to ensure good coverage of the problem domain; normally a small number of cases can cover the commonly occurring problems.
- **Naturally occurring cases:** there is no manual case-selection process; instead, the addition of cases to the case-base is managed by a case-base maintenance/learning process without human intervention.

In the first scenario cases are produced through knowledge engineering, in the second by a data mining process.

The final aspect we wish to emphasise is the type of behaviour a case captures. The focus may be on capturing an expert decision making process or on capturing the behaviour of a complex system. On the one hand, a CBR system may wish to capture underwriting decisions in a bank or therapy decisions in a hospital. On the other hand it may wish to predict outcomes associated with underwriting decisions or the prognosis for a set of symptoms. Explanations with these alternatives would be as follows:

- **Capturing expert decision making:** The system recommends that Patient X should be discharged because Patient Y with similar symptoms was discharged.
- **Capturing the behaviour of a complex system:** The system recommends that Patient X should be admitted because Patient Y with similar symptoms to the current case was discharged and required readmission within 2 days.

Clearly the potential to gain insight is greater with the second rather than the first. In this second scenario the CBR is discovering knowledge in data and explaining it. In this paper we discuss CBR systems that are:

- Knowledge-light,
- Built on naturally occurring cases, and
- Capture the behaviour of a complex system.

This is an approach that is appropriate for *weak-theory* domains where the details of the causal interactions are not well understood, where all experts can provide is a sense of the *direction* of causal interactions. The structure of the paper is as follows. In the next section we provide an overview of case-based

explanation and outline the advantages. In section 3 we present an explanation-oriented framework for case-retrieval. In section 4 we show how the details of these cases can be assessed and selected for highlighting in explanation. This paper concludes with a summary in section 5.

2 Case-Based Explanation

Machine learning techniques have been successfully used in knowledge discovery tasks but many of these models fail to present the detected patterns (knowledge) to the user in an interpretable manner. Although these systems have proved to be successful in terms of predictive accuracy, the lack of interpretability and transparency in the way they operate has been a major stumbling block in their application to real world tasks (Andrews et al., 1995). People are understandably reluctant to accept without question machine derived predictions, particularly when there is no insight on how the prediction was produced or might be justified. These systems also fail from a knowledge discovery perspective in that the knowledge is inaccessible to the end user.

By providing explanatory feedback it is hoped that confidence in the system's prediction can be given to the end user. Explanations also provide the user with an insight into the problem domain since, by their nature, they seek to impart some form of knowledge to the user (Sormo and Cassens, 2004). However many state of the art machine learning techniques are inherently un-interpretable and lack transparency in the way they operate. Ensembles, Support Vector Machines (SVMs) and Neural Networks are typical examples of such systems and all offer no prospect of direct transparent user feedback. Rule and tree based systems are often regarded as being inherently interpretable. The relevant rules or a tree structure can be presented to the end user as explanatory feedback and justification of its prediction. However in reality such systems often fail to offer convincing explanations. When applied to complex problems the complexity of the rules or tree structure also increases at the cost of their interpretability leading to complex and unconvincing explanatory feedback. This has been the experience of the Knowledge-Based Systems community (Majchrzak and Gasser, 1991).

Conversely CBR systems have an inherent transparency that has particular advantages for explanations as Leake (1996) points out:

> *"...neural network systems cannot provide explanations of their decisions and rule-based systems must explain their decisions by reference to their rules, which the user may not fully understand or accept. On the other hand, the results of CBR systems are based on actual prior cases that can be presented to the user to provide compelling support for the system's conclusions."*

The type of explanations given by knowledge-light CBR systems and the insight they offer to the end user differ considerably from those found in rule-based and other approaches in a number of ways:

- **Natural Form of Explanation:** Research in cognitive science and other areas suggests that explanation by analogy is a natural form of explanation in some domains and one people can quickly relate to (Gentner et al., 2003; Cunningham et al., 2003).

- **Use of Real Evidence:** In CBR the user is presented with actual cases that represent past experiences. In most applications these cases are undoubtedly true and so their validity isn't in question, this is the great strength of case-based explanations. Users who are unfamiliar or suspicious of a system are more likely to be convinced by explanations that contain factual evidence than by unsupported rules.
- **Fixed and Simple form of Explanation:** CBR explanations avoid the interpretability versus fidelity trade off that can plague some other techniques. The type of explanation presented to the user is independent of the complexity of the problem.

The major challenge with case-based explanations lies in ensuring the perceived appropriateness of the presented cases to the validity of the prediction. The task of ensuring that the cases are deemed appropriate and convincing can be broken down into two distinct stages:

- **The selection of cases to present to the user:** The major driving force for the provision of explanations is to offer a justification for a prediction. The goal of providing a convincing argument may not always be best served by supplying the user with the nearest neighbors. Convincing explanations are domain and user dependent (Sormo and Cassens, 2004), and this should be reflected in the case retrieval process. Taking the domain and user details into account, the retrieval process should be adjusted to select the cases that form the most convincing argument.
- **Explaining the details and relevance of retrieved cases:** This is an issue that has recently received a lot of attention in the CBR community. In CBR explanations, the ability of the user to make meaningful comparisons between feature values in the query and the retrieved explanation cases is of critical importance to the success of the explanation. CBR systems are not wholly transparent and much domain knowledge can be contained within the similarity metrics used in the system. It is implicitly assumed in simple CBR explanations systems that the user has this same domain knowledge and so the appropriateness of the explanation case is clear. However, this may not be the case and the relevance of the retrieved case may be lost on novice users. By providing users with extra explanatory feedback further insights into the problem are given in addition to further reassuring the user.

In the following sections we will discuss each of these tasks in turn focusing on recent research that has been done in each area.

3 Case Retrieval

The obvious way to provide explanations in CBR is to display the case most similar to the target case to the user. Some of our recent work has shown that this type of explanation is considered more convincing than that provided by rule based systems (Cunningham et al., 2003). More recently we have discovered that the nearest neighbour may in fact not be the most convincing case to use as an explanation for a classification (Doyle et al., 2004). In this section we will

look at the idea of *a fortiori* arguments and consider how this principle might help select more convincing cases to use in explanation. We will then look at how we can use explanation utility measures based on this principle to produce more convincing explanations.

3.1 A Fortiori Arguments

Most parents are familiar with the use of *a fortiori* arguments by children[1]. *A fortiori* arguments are used to argue a case beyond reasonable doubt. Let us consider an example of a child using an *a fortiori* argument to plead their case to see the latest Harry Potter movie.

Figure 1 shows an example of a child called Mark (the triangle) who wants to see the latest Harry Potter movie. The circles represent the children who have seen the movie and the squares the children who have not. Mark knows that Kate is the closest in age to him and she has seen the movie. But Mark knows that the older you are the more likely you are to be allowed to see the movie. If Mark were to use Kate as an argument to convince his parents to let him go to the movie, there is a possibility that Mark's parents can argue that Mark is still a little too young to go. However Mark knows that if he uses John who is younger than him as his argument to see the movie, he has a stronger case.

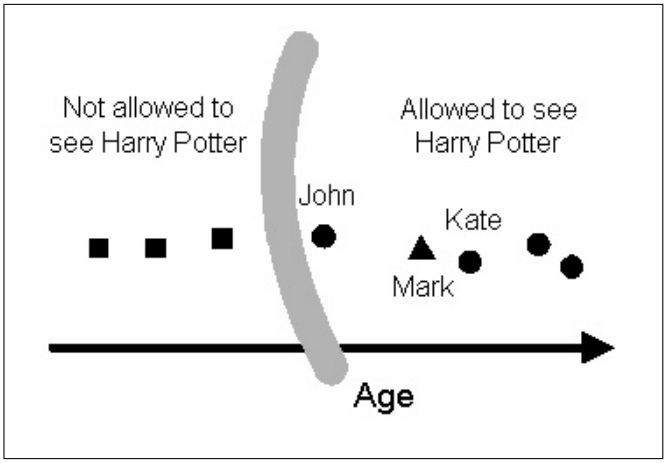

Figure 1: Using *a fortiori* arguments

The above example shows that the most similar situation is not always the most convincing situation to support an argument. In the above situation, picking a child between himself and the perceived decision boundary (Mark doesn't know the exact cutoff age, but knows the younger you are the less likely you will not be allowed go) Mark was able to strengthen his case. Similar to the above example, we believe that when supporting a classification in CBR, the most similar neighbour to a target problem case is not necessarily the most convincing case to support the classification. For instance, if a decision is being

[1] "*a fortiori* - adv. for similar but more convincing reasons: if Britain cannot afford a space program, then, a fortiori, neither can India." - Collins English Dictionary

made on whether to discharge a sick 12 week old baby from hospital, a similar example with a 9 week old baby that was discharged is more compelling than one with an 13 week old baby (based on the notion that younger babies are more likely to be kept in).

3.2 Explanation Oriented Retrieval

We have developed a system that attempts to select a more convincing case than the most similar neighbour. A major step in this process is to use explanation utility measures (Doyle et al., 2004). These measures are dependent on the classification of the case being explained. Figure 2 shows the explanation utility measure for the feature Age when the classification is Allowed to See Harry Potter. When calculating the utility between a case x and a query case q, if the age of x is older than q, the utility measure for age is in the range of 0-1. However if q is older than x the utility is 1. Therefore the utility for the 'Allow' argument works by favouring younger cases than the query case. Alternatively when trying to argue that someone should not be allowed to the cinema an alternative graph that favours older cases would be used.

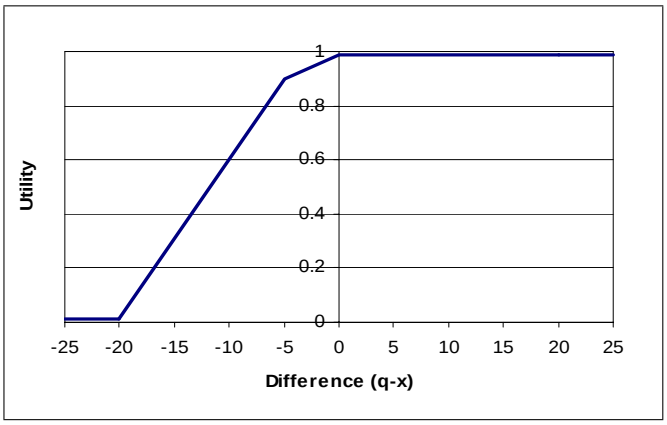

Figure 2: Explanation Utility graph for the Age feature and the 'Allowed'.

Once the utility measures for each feature and classification combination has been defined, the most convincing case to support a particular classification is selected using the following process:

1. Get Nearest Neighbours
2. Perform Classification using Nearest Neighbours
3. Select explanation utility measures to use based on classification
4. Reorder Nearest Neighbours of the same class using selected explanation utility measures

It should be noted, that different sets of explanation utility measures can be defined for different people. It may be that some people may prefer a case that is close to the Nearest Neighbour, while others may prefer a case that is closer to

the perceived decision bounday. More information on using utility neighbours can be found in (Doyle et al., 2004).

3.3 Nearest Unlike Neighbour

An important issue in decision support systems is the idea of providing some measure of confidence in the prediction of the system (Cheetham and Price, 2004). This can help in promoting user-acceptance of the system and it can also alert the user to situations when the prediction may be wrong. An idea we are currently looking at is to display the nearest unlike neighbour(NUN) to help in this direction. If the NUN is quite different to the query case it should give confidence that the prediction is robust. If the NUN is very similar to the query case, that should alert the user to the fact that the query case is close to the border. In addition, since the decision surface is between the NUN and the explanation case, it offers some insight into the features that influence classification in that part of the problem space.

3.4 Example

We have done some research on using case-based techniques to predict blood-alcohol levels (Cunningham et al., 2003). This task has many of the characteristics of typical medical decision support problems and it is possible to gather a significant number of cases using our own resources. In this domain there are two tasks, there is the regression task of predicting the blood-alcohol level and the classification problem of predicting if a person is under or over the legal drink driving limit. Table 1 shows an example from this domain.

Table 1: Example from the Breathalyser Domain

Features	Target Case	Nearest Neighbour	Explanation Neighbour	NUN
Weight	76	76	73	63
Duration	60	60	60	120
Gender	Male	Male	Male	Male
Meal	Full	Full	Full	Full
Units	2.9	2.6	5.2	7.2
BAC	Under	Under	Under	Over

In this situation the Nearest Neighbour is very similar to the query case. However as the Nearest Neighbour has consumed less alcohol than the query case, there is the possibility that the query case could be over the limit, i.e. that the decision surface is between the Nearest Neighbour and the query case.

However the case selected using the Explanation Utility measures, the Explanation Case in the table, has consumed more alcohol than the query case and is still under the limit. This adds support to the prediction that the query case is under the limit. In this situation the Weight of the Explanation Case is also less than the query case. This also supports the argument given that the lighter a person is(all other factors being constant) the higher their alcohol level.

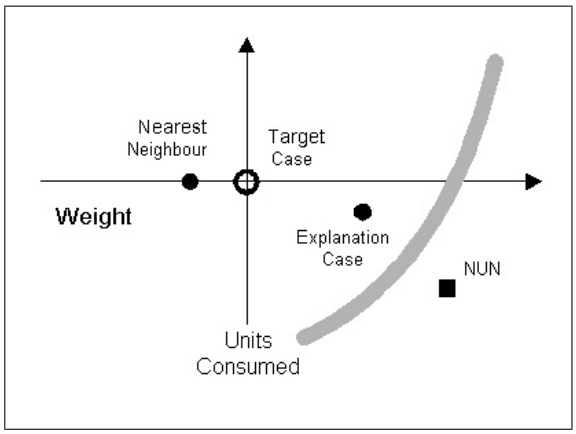

Figure 3: Explanation Utility Graph for Age Feature

In this example the cases only differ in two dimensions, Units Consumed and Weight. This allows us to represent the situation graphically as shown in figure 3. From our understanding of the way in which features influence blood-alcohol level we expect that the decision boundary is somewhere 'north west' of the NUN. Without knowing about the Explanation Case it is just possible that it could lie between the query case and the Nearest Neighbour. Knowing of the existence of the Explanation Case gives use the assurance that the decision boundary must be on the far side of it (from the query case).

This is a knowledge-light approach to explanation that works in weak theory domains. The knowledge engineering required involves expressing the direction of causal interaction as utility measures as shown in figure 2. The return on this is that the system can show Explanation Cases and NUN to the user as described in table 1.

4 Explaining Cases

As has been made clear in section 1 much of the knowledge in a CBR system is contained within the vocabulary that descibes the cases and the similarity measures. However it is also clear from section 2 that the knowledge captured in each of these containers isn't always transparent to the user. Recently much work has been done in overcoming these problems. In this section we discuss two different approaches to tackling this problem. We begin in section 4.1 with some fundamental work done in the area, we then focus on a more general approach in section 4.2.

4.1 Explaining Feature-value Effects

The issue of transparency is one that McSherry has addressed in his ProCon System (McSherry, 2003). McSherry has focused on making the relationship between the feature values within a case and its predicted value explicit. He argues that simply presenting the feature values in the most similar cases may be misleading. The relationship between feature values and the predicted value

may not always be a positive one; the presence of some feature values may in fact be evidence against the prediction. Simply supplying the user with a case may lead them to incorrectly infer the relationship between feature-values and the prediction. To combat this McSherry provides the user with extra relational information about the case feature-values and the predicted class-value. To infer the feature-values to class-value relationships a Naïve Bayes model is built on the entire training set and from this the relational information is derived. Using the Naïve Bayes model it is possible to infer the effect of different feature-values and so inform the user whether a particular feature-value is a *supporter* or *opposer* of a given prediction. In table 2 we can see an example of the ProCon-2 system's output as seen in (McSherry, 2003).

It is clear that this approach is very effective in informing the end user on the relationships between feature-values and the predicted outcome. However the Naïve Bayes approach induces a global model of interactions and this may not always be appropriate as the influence of features may vary over the problem space. Indeed, it might be argued that global models are not in the spirit of CBR which is a local approach to learning.

4.2 Explanations for Complex Problems

We can see in McSherry's work that the provision of additional information describing the relationship between feature-values and outcome has the potential to greatly enhance the information conveyed in CBR explanations. However, CBR systems provide localised solutions derived from cases retrieved specific to the query case. This means that the use of a global model may not always be suitable. The local approach taken in CBR makes it applicable to problems that are non-linear in nature. In non-linear problems the relationship between features and outcome may vary across the feature space. The use of global model to describe the interactions of features may well fail to capture this non-linear behaviour and so may not reveal what is actually going on. Some features may be important in some areas of the feature space and not at all relevant in other areas.

We have developed a general framework for producing explanations that are built upon feature-value relationships derived locally and specific to each query case. The framework was developed to provide case-based explanations for black-box systems because of the strengths of this form of explanation (Nugent and Cunningham, 2004). However the framework is equally applicable to CBR systems and demonstrates the strength of CBR for Knowledge Discovery tasks.

In providing an explanation for a user we would like to provide the user with a sense of the role the feature-values had in producing a prediction. It would also be useful to explain the effects of any differences in the explanation and query that there might be. This would provide the user with a sense of the relationship of the feature-values to the prediction for the presented case that they can then critically assess. This information will also focus the users attention on the more important features of a case. However, as stated earlier and specifically in the domains that we have considered (Blood Alcohol Content (Cunningham et al., 2003) and Bronchiolitis (Walsh et al., 2004)), it seems that simple global models are not suitable for such tasks. To overcome this problem we have designed a two stage explanation system that takes advantage of our ability to use the black-box, or in this case the CBR system, as an oracle from

Table 2: **Sample ProCon-2 Output from (McSherry, 2003)**

ProCon-2:	Please describe the target problem.
User:	weight=79, duration=90, sex=male, meal=full, units=10.1
ProCon-2:	The 3 most similar cases are:
	Case 13: 79, 240, male, full, 9.6, over-limit(0.97) Case 26: 73, 120, male, full, 9, not-over-limit(0.96) Case 56: 79, 120, male, full, 7.2, not-over-limit(0.96)
	The predicted outcome is: over-limit.
User:	**why**
ProCon-2:	Features of the target problem that SUPPORT the prediction:
	units=10.1
	Features of the target problem that OPPOSE the prediction:
	weight=79 duration=90 sex=male meal=full
	The most similar case is: Case 13:79, 240, male, full, 9.6, over-limit
	The outcome in this case was over-limit in spite of the oposing evidence that:
	weight=79 sex=male meal=full

which we can extract information about its behaviour in particular regions of the feature space. This information is then used to inform the user of the role of the various features and also to drive the retrieval process. We will now talk about each of the two stages in turn.

4.3 Derive Local Feature Information

The provision of feature role information provides the user with a sense of how each of the feature-values contributed to the particular prediction. It is important too that these rankings should reflect the locality of the presented case on which the prediction is made. In order to provide such feature rankings, two distinct steps are taken. First, the CBR system is treated as an oracle and an artificial data set is constructed around the point of inquiry, then a model is built on this data. The model is then used to transform the information captured in the locally derived case-base into a useful form.

Treating the CBR system in this way allows us to get a prediction for any set of feature-values we care to imagine. We can present the CBR system with feature-value sets similar to those of the query case and so we can build up a local case-base around the original query point. This is done by perturbing, in a controlled manner, the feature values of the case we are providing an explanation for and using the CBR system to attach a prediction to the artificial case. In this way we can implicitly garner some of the knowledge that is encapsulated in the similarity measures.

The choice of model used to capture the local behaviour is primarily driven by the need for a model that can provide us with information about each feature-value's role. We would also like to be able to describe what effect any differences between the query case and the explanation case might have.

4.4 Case Retrieval Mechanism

The objective is to present the user with cases that reinforce the prediction from the system. However, as we have already discussed in section 3 the nearest neighbour may not always make the most convincing argument. The information captured in the localised model can be used to find cases that are more convincing to the user.

4.5 Flow Diagram of the Framework

The flow of execution and the relationships and dependencies between individual processes in the framework outlined above is described in Figure 4. In our explanation framework each explanation is tailored to the particular set of inputs and the prediction made. First, the query case is used to seed the generation of an artificial case-base. Once we have this data we then describe it using some model as discussed in Section 4.3. We now have a description of the behaviour of the CBR system in the region of interest and this information is then used in two further stages of the explanation process; case retrieval and the explanation stage as can be seen below. In the case retrieval process the feature-value information is used to select the best case from the original case-base to use in the explanation. This is then passed on to the final explanation stage where it, along with the feature-value information, is used to generate

the final explanation presented to the user. The exact form of the explanation and what information is presented to the user is very much both task and domain dependent (Sormo and Cassens, 2004; Cassens, 2004). In Section 4.7 we present one possibility in which a localised logistic regression system is used in a classification task.

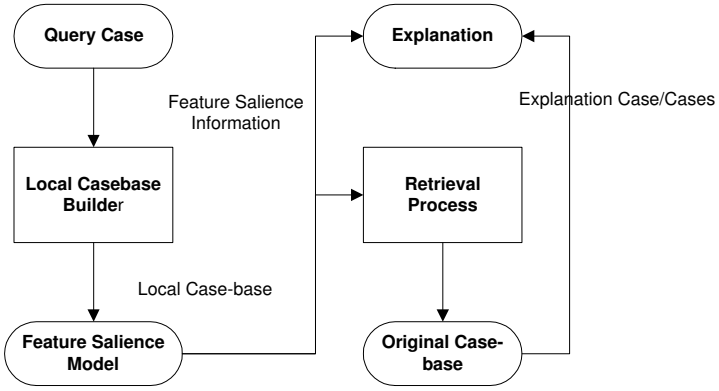

Figure 4: A Flow Diagram of the Explanation Process

4.6 Sample Implementation

As an example of how the framework outlined above can be used to provide convincing explanations, we applied it to the task of explaining the predictions of Nearest Neighbour Classifier (NN) with $k = 3$. The NN was built on the Blood Alcohol Content (BAC) case-base. The task involves using information about peoples' weight, gender, number of units of alcohol consumed, etc. to predict whether the concentration of alcohol in their blood stream has reached a level where they would be over the drink-driving limit. The training data was taken from the data that had previously been collected and used by Cunningham et al. (2003). The local model used was a logistic regression model. The Logistic Regression model has a number of characteristics that make it an ideal candidate for our purposes. First we will briefly discuss logistic regression models, in particular focusing on how they can be used to derive feature-value information. We then present some sample explanations produced using the logistic regression in conjunction within the explanation framework.

4.6.1 Logisitic Regression Models

Hosmer and Lemeshaw have written an excellent and comprehensive book on the subject of Logistic Regression (Hosmer and Lemeshow (2000)). In the preface to the second edition they point out the huge increase in the use of the modelling technique from its original use within epidemiologic research to use within fields as diverse as *"biomedical research, business and fiance, criminology, ecology, engineering, health policy, linguistics and wildlife biology"*. Logistic regression is a data analysis technique that offers an insight into the relationship between input variables and a target, or class variable. It is specifically

designed for binary classification problems and the increase in popularity of the modelling technique is understandable as it offers powerful insights while maintaining model simplicity.

Logistic regression, like linear regression, produces a set of coefficients from which the relationship of an input variable to the target class variable can be deduced. However unlike linear regression, logistic regression coefficients don't directly correspond to slope values in the same way. In logistic regression tasks, the two possible class values are coded as being either 0 or 1. Because the value predicted by the model, the conditional mean, is no longer an unbounded value as in linear regression but a value between 0 and 1, the data is fitted to a distribution that ensures the outputted value always meets this bounding criteria. To do this the logistic distribution is applied as can be seen below (1).

$$Y(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}} \tag{1}$$

Here $Y(x)$ is the conditional mean for a particular value of x while β_0 and β_1 are the model parameters. The distribution produces the conditional mean, a value between 0 and 1, for any given inputted value of x. Importantly, for binary problems the conditional mean is in fact the probability of class 1 given x.

At first glance this model looks quite intimidating and seems to offer no hope of offering an insight into the relationship between x and our class variable. However, the logistic distribution is chosen because it can be easily transformed into another form which has many of the desirable properties of a linear regression model. By applying the logit transform, equation 2, we end up with a simple and interpretable model, the logit (3).

$$g(x) = \ln \frac{Y(x)}{1 - Y(x)} \tag{2}$$

$$g(x) = \beta_0 + \beta_1 x \tag{3}$$

The parameters of the logit model can easily be converted into odds ratios. The odds ratio of an event is the odds of that event occurring over the odds of it not happening. For instance if someone were to state the odds ratio of smokers to non-smokers getting cancer is 2 then this would mean smokers are twice as likely to develop cancer as non-smokers. Alternatively, if we looked at the relationship the other way round, non-smokers to smokers, we would get a odds ratio of 0.5. This means that non-smokers are half as likely to get cancer. In general an odds ratio greater then one for possibility A over possibility B means A makes the event more likely than the alternative while and odds ratio of less then one means it makes it less likely. The logistic regression model makes the calculation of odds ratios quite easy and this is extremely useful and informative. It is this simple relationship between the model coefficients and the odds ratio and their natural interpretation that has made logistic regression such a popular tool. We will first discuss in a very general sense how this is done as it will be of use in section 4.6.2 and then focus on a particular example that highlights why logistic regression has proved so popular.

In order to extract the odds ratio, two steps are taken. First the logit difference is found. Imagine we are interested in the odds ratio of two different

events, $x = c$ and $x = d$. the logit difference can be calculated as in equation 4. The logit difference, ld, is simply the difference in the logit function for the two values of x we are interested in. Once this value has been obtained it can then be converted into odds ratio, see equation 5.

$$LogitDifference(x = c, x = d) = g(c) - g(d) = ld \tag{4}$$

$$OddsRatio(x = c, x = d) = e^{ld} \tag{5}$$

The trick with the logistic regression model is that in many cases it isn't necessary to calculate the logit difference. If the model variables have been properly coded then the desired information usually can be got by simply looking at the model coefficients. As an example consider a hypothetical situation where we have developed a model that relates smoking to the development of cancer. Our hypothetical model might look something like that shown in equation 6.

$$g(Smoker) = 0.3 + 0.69Smoker \tag{6}$$

If we code our smoking variable as being equal to 1 if someone smokes and 0 if they don't then the calculation of the logit difference is simply equal to the $Smoker$ coefficient (7).

$$\begin{aligned} g(Smoker = 1) - g(Smoker = 0) &= 0.3 + 0.69(1) - (0.3 + 0.69(0)) \\ &= 0.69 \\ OddsRatio(Smoker) &= e^{0.69} \\ OddsRatio(Smoker) &= 2 \end{aligned} \tag{7}$$

(8)

As can be seen above in 6 we need not have bothered calculating the logit difference and instead just used the model coefficient. This is also true for continuous and multi-value nominal variables if they are coded correctly (chapter 4, Hosmer and Lemeshow (2000)). Once we have the odds ratio the relationship between input variable and the class variable is clear. We have focused most of our discussion on examples with only a single input variable for simplicity sake but the above observations are also true in multi-variable problems. In the next section we discus how exactly information derived from the logistic regression model can be used to provide convincing explanations.

4.6.2 Using Logistic Regression in the Framework

Information from the local model is used in two different sections of the framework; in the retrieval of the explanation case and in giving feature-value information. The logistic regression model can be used very effectively to generate feature-value information as it allows us to calculate the effects of differences between the query case and the explanation case (equations 4 and 5). With this information we can then generate a dialog and explain these effects to the end user.

The logistic regression model can also be used to generate pseudo-*a fortiori* arguments as discussed in section 3. Logistic regression models allow us to

generate a probability for a given set of inputs. In the explanation case retrieval process we can then use this to find an explanation case that is nearer the decision boundary and so a more convincing argument. By passing each of our candidate explanation cases through our local logistic model we can generate a probability for each. A case that is nearer the decision boundary will have a more marginal probability and so this should be the case we select.

The logistic regression model is also useful in one further aspect as it allows us generate a confidence measure of the prediction from the CBR system. If we pass the query case through the locally derived logistic model we will then get a probability measure of it being of a certain class. This can then be used to calculate our confidence in that prediction. Generating confidence measures in system predictions is an important issue in their application to real world problems (Cheetham and Price, 2004). If we can alert the user to when we are not confident of a prediction it is more likely the system will be trusted in the long term. To make each of these processes clearer we will first step through how one particular explanation was generated.

After the Query Case has been classified we can then build our logistic model on our local data. In table 3 we can see the Query Case, its predicted classification and the three Nearest Neighbours used to classify it. In order to select a case to use in our explanation we first run each of cases including the Query Case through our local logistic regression model. This gives us the set of probabilities that can also be seen in table 3. We can see that Nearest Neighbour 2 has the lowest probability and so is the case nearest the decision boundary. This is an alternative to the explanation utility framework described in section 3.2 for selecting the case to present to the user to make the most convincing argument. Although Nearest Neighbour 2 had consumed more units of alcohol and weighed less, they were under the limit so it seems reasonable that our Query Case should be too.

Table 3: Explanation Case Retrieval Process

Features	Query Case	Nearest Neighbour 1	Nearest Neighbour 2	Nearest Neighbour 3
Weight	88	82	79	76
Duration	120	120	120	120
Gender	Male	Male	Male	Male
Meal	Full	Full	Full	Full
Units	5.2	5.0	7.2	4.6
BAC	Under	Under	Under	Under
Probability	0.98	0.97	0.89	0.96

We can make this argument more explicit to the end user by explaining the effects of the feature differences between the Query Case and Explanation Case using our local logistic regression model and equations 4 and 5. We can substitute each of the feature differences into the equations individually and get the odds ratio for each. Using the odds ratio we can then determine the effect of the change. The kind of dialog that can be produced can be seen in Table 4. In this sample dialog we can see the advantage of using our local model to classify

the Query Case as this gives us a measure of confidence in the prediction. In the next section we discus some further sample dialogs.

4.7 Sample Explanations on the BAC case-base

Two further sample dialogs are shown in Tables 5 and 6. The benefit of the explanation dialog can be seen in both examples. It is clear that the feature information is in line with our intuitive understanding of the problem and that they add value to the overall explanation. The dialog is useful in appreciating the difference between the explanation and query case as well as offering an insight into the nature of the problem being studied.

In Table 5 the user is presented with both the Query Case and the Explanation Case. The Explanation Case forms an *a fortiori* argument as the number of units is lower but the case is still over the limit. However there is a difference in the `Weight` value that means that the Explanation Case is more likely to be Over the Limit given the same amount of alcohol as the Query Case. Using the local logistic model this information can be derived without expert knowledge and presented to the user as can be seen in the generated dialog.

In Table 6 a strong argument in favour of the prediction is made.In the Explanation Case more `Units` have been consumed and the subject was lighter yet they were found to be Under the Limit. Again this information is made explicit through the generated dialog. In this case the confidence measure is 87% which means we can still be reasonably confident of our prediction. By providing the confidence measure the user can be alerted to situations where the prediction might be wrong.

5 Conclusions

We have shown how this knowledge-light approach to explanation can provide insight into a problem domain by presenting similar matching and un-matching cases to the user.

The first aspect of this approach is to select explanation cases based on an *a fortiori* principle. The argument is of the type, "if this case from the case-base is of a particular class, then there is even more reason to believe that the query case is of the class". These explanation cases are selected based on knowledge about the *direction* of the effect of features on outcomes. This knowledge is expressed in the explanation utility framework that was described in section 3.

The second aspect of the approach is to highlight case features and indicate how they influence outcomes as described in section 4. Features are selected based on a local logistic regression model from the region of the query case.

References

Andrews, R., Diederich, J., and Tickle, A. (1995). A survey and critique of techniques for extracting rules from trained artificial neural networks. knowledge based systems. *Knowledge Based Systems*, pages 187–202.

Armengol, E., Palaudàries, A., and Plaza, E. (2001). Individual prognosis of diabetes long-term risks: A CBR approach. *Methods of Information in Medicine*, 40:46–51.

Cassens, J. (2004). Knowing what to explain and when. In *1st Workshop on Case-Based Explanation (Proceedings of the ECCBR 2004 workshops)*, pages 97–104.

Cheetham, W. and Price, J. (2004). Measures of solution accuracy in case-based reasoning systems. In Funk, P. and Calero, P. A. G., editors, *Advances in Case-Based Reasoning, 7th. European Conference on Case-Based Reasoning (ECCBR 2004)*, volume 3155 of *Lecture Notes in Computer Science*, pages 106–118. Springer.

Cunningham, P., Doyle, D., and Loughrey, J. (2003). An evaluation of the usefulness of case-based explanation. In Ashley, K. D. and Bridge, D. G., editors, *Case-Based Reasoning Research and Development, 5th International Conference on Case-Based Reasoning (ICCBR 2003)*, volume 2689 of *Lecture Notes in Computer Science*, pages 122–130. Springer.

Doyle, D., Cunningham, P., Bridge, D., and Rahman, Y. (2004). Explanation oreinted retrieval. In Funk, P. and Calero, P. A. G., editors, *Advances in Case-Based Reasoning, 7th. European Conference on Case-Based Reasoning (ECCBR 2004)*, volume 3155 of *Lecture Notes in Computer Science*, pages 157–168. Springer.

Gentner, D., Loewenstein, J., and Thompson, L. (2003). Learning and transfer: A general role for analogical encoding. *Journal of Educational Psychology*, 95(2):393–408.

Hosmer, D. and Lemeshow, S. (2000). *Applied Logistic Regression*. Wiley, 2nd edition.

Kass, A. and Leake, D. (1988). Case-based reasoning applied to constructing explanations. In Kolodner, J., editor, *Proceedings of 1988 Workshop on Case-Based Reasoning*, pages 190–208. Morgan Kaufmann.

Leake, D. (1996). *Case-Based Reasoning: Experiences, Lessons and Future Directions*. AAAI/MIT Press.

Majchrzak, A. and Gasser, L. (1991). On using artificial intelligence to integrate the design of organixational and process change in us manufacturing. *AI and Society*, 5:321–338.

McSherry, D. (2003). Explanation in case-based reasoning: an evidential approach. In *8th UK Workshop on Case-Based Reasoning*, pages 47–55.

Nugent, C. and Cunningham, P. (2004). A case-based explanation system for 'black-box' systems. In *1st Workshop on Case-Based Explanation (Proceedings of the ECCBR 2004 workshops)*, pages 155–164.

Richter, M. (1998). Introduction. In Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., and Wess, S., editors, *Case-Based Reasoning Technology, From Foundations to Applications*, pages 1–16. Springer.

Roth-Berghofer, T. (2004). Explanations and case-based reasoning: Foundational issues. In Funk, P. and Calero, P. A. G., editors, *Advances in Case-Based Reasoning*, pages 389–403. Springer-Verlag.

Sormo, F. and Cassens, J. (2004). Explanation goals in case-based reasoning. In *1st Workshop on Case-Based Explanation (Proceedings of the ECCBR 2004 workshops)*, pages 165–174.

Walsh, P., Cunningham, P., Rothenberg, S. J., O'Doherty, S., Hoey, H., and Healy, R. (2004). An artificial neural network ensemble to predict disposition and length of stay in children presenting with bronchiolitis. *European Journal of Emergency Medicine*, 11(5):259–264.

Table 4: Sample Dialog

	Query Case	**Explanation Case**
Weight (kgs)	88	79
Duration (mins)	120	120
Gender	Male	Male
Meal	Full	Full
Amount (Units)	5.2	7.2
BAC	Predicted Under	Under

Explanation:

The differences in the feature `Amount` and `Weight` mean that the Query case is more likely to be under the limit than the Explanation Case.

The estimated probability of this prediction being correct is 98%

Table 5: Example A

	Query Case	**Explanation Case**
Weight (kgs)	69	66
Duration (mins)	180	180
Gender	Male	Male
Meal	Lunch	Lunch
Amount (Units)	15.6	10.2
BAC	Predicted Over	Over

Explanation:

The differences in the feature `Amount` mean that the Query case is more likely to be over the limit than the Explanation Case.

The differences in the features `Weight` mean that the Explanation case is more likely to be over the limit than the Query Case.

The estimated probablilty of this prediction being correct is 94 %

Table 6: Example B

	Query Case	**Explanation Case**
Weight (kgs)	76	73
Duration (mins)	120	120
Gender	Male	Male
Meal	Full	Full
Amount (Units)	7.2	9.0
BAC	Predicted Under	Under

Explanation:

The differences in the feature `Weight` and `Units` mean that the Query case is more likely to be under the limit than the Explanation Case.

The estimated probablilty of this prediction being correct is 87 %