# CBR Confidence as a Basis for Confidence in Black Box Systems

Lawrence Gates(✉), Caleb Kisby, and David Leake

School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408, USA
{gatesla,cckisby,leake}@indiana.edu

**Abstract.** Determining when to trust black box systems is a well-known challenge. An important factor affecting users' trust is confidence in system solutions. Previous case-based reasoning (CBR) research has developed criteria for assigning confidence to the solutions of a CBR system. This paper investigates whether such analysis, coupled with factors such as CBR system competence, can be used to predict confidence in the outputs of a black box system, when the black box and CBR systems are provided with the same training data. The paper presents initial strategies for using CBR confidence to predict black box system confidence. An evaluation explores the ability of the strategies to provide useful information and suggests future questions.

**Keywords:** Case base competence · Case-based reasoning · Neural network · Explainable artificial intelligence · Confidence · Competence

## 1 Introduction

Advances in machine learning, and in particular in deep networks, have led to widespread applications of AI systems with powerful performance achieved through methods that are largely opaque to their human users. Such systems, often referred to as *black box* systems, accept an input and propose an output without an account of how the output was generated. This can be especially troubling when the black box systems, despite overall strong performance, sometimes perform unexpectedly. For example, it is well known that deep networks may exhibit unexpected behaviors on *adversarial examples*; two images that a human sees as identical may receive different classifications [20,29]. Such behavior and the inability to explain the performance of black box systems has been widely acknowledged as a concern for confidence in their conclusions, which can limit the domains to which they are applied. This in turn has led to an outpouring of research on explainable AI (e.g., [2]), including in the context of case-based reasoning [1].

Explanation of black box systems has a long history of combining the black box systems with more interpretable methods. For example, one approach is


K. Bach and C. Marling (Eds.): ICCBR 2019, LNAI 11680, pp. 95–109, 2019.
https://doi.org/10.1007/978-3-030-29249-2_7

to use interpretable ML methods, such as decision trees, to build a model of the black box system reasoning that can then be used to explain predictions. However, as rules become more complex they become less interpretable, and it may be difficult to capture the black box system's behavior with sufficient fidelity (e.g., [10]). From the early days of case-based reasoning, the ability to explain CBR system reasoning by reference to prior cases has been seen as an important benefit [15]. This makes it appealing to combine case-based reasoning with black box systems, to increase explainability of black box system behavior. For example, Shin et al. [27] propose a CBR/neural network hybrid in which neural-network-generated features are used to retrieve relevant cases, with the goal of explainability. Nugent and Cunningham propose a general framework for case-based explanation of behavior of black box systems [22]. In their approach an artificial case base, seeded with cases generated by the black box system, is used to determine local feature salience, which is used in turn to guide retrieval of real cases as the basis for explanations to increase user confidence in black box system conclusions. Keane and Kenny provide an extensive survey of research on "twinning" CBR and neural network systems to provide explanations [14].

This paper brings together CBR and black box systems in a different way, for a task complementary to the explanation task per se: to assess confidence in black box solutions. In the presented approach, COBB (Case-based cOnfidence for Black Box), both the CBR system and black box system have access to the same training data (or subsets of each other's data); each functions in parallel. However, the goal of the CBR system processing is not to provide the solution, but instead, to ascribe confidence to the black box system output. That confidence judgment can be directly provided to a user, as a unitary confidence judgment, and the confidence (not the solution itself) can explained in terms of characteristics derived from the CBR system. Thus in contrast with, e.g., Nugent and Cunningham: The role of the CBR system is not to replicate the black box system performance, but to provide an independent view, based on the same data, as a "second opinion" based on a more intelligible process that can be examined to assess its conclusions. The confidence information can then be used, for example, to decide when to expend scarce resources on evaluating solutions (e.g., in a financial system, presenting the problem to a human expert, or presenting the case retrieved by COBB as the basis of independent assessment).

An important question for such paired systems is how much their value depends on the relative performance of the CBR and black box systems. The primary use case for the COBB approach is situations in which in general, black box system solutions have higher confidence. Were that not the case, the CBR system, not the black box, should be the primary reasoning system. We discuss this question in more detail in Sect. 6.1.

Given that the CBR system and black box system are independent, with the CBR system potentially having lower accuracy, a natural question is the extent to which the CBR system can ascribe confidence to the black box system results. The answer is twofold. First, for assessing confidence, independence of the two systems can be a benefit to give a true second opinion. On the other

hand, a premise of the approach is that the world basically conforms to the CBR hypothesis that "similar problems have similar solutions," so black box system behavior that conflicts with that premise—as manifested by the CBR system—should be ascribed lower confidence.

This paper proposes and evaluates three potential methods for predicting confidence of a black box system based on a CBR system. The first method proposed is based on a naïve analysis of the relationship between CBR confidence, black box confidence, and the distance between the two solutions. The second method combines several predictors in order to determine the confidence in the black box solution. The third method builds on the extensive work of Cheetham on CBR confidence by applying his confidence indicators approach to the black box system outputs.

Experimental results show that the method with the best overall quality was the second method. It generally had better overall quality than the other two, and for large testing sets had very good quality. The paper closes with directions for extending this work.

## 2 Previous Work

*CBR Confidence Models:* In seminal work, Cheetham proposed the development of confidence models for case based reasoning. His goal was twofold: to provide information to help predict whether a solution had low error, and to determine whether the output of the CBR system should be used for a given task. His approach [5,7] explores incorporating a measure of quantitative values for confidence and an error factor into each score. Reilly et al. [26] developed an explicit model of confidence for case-based conversational recommender systems.

*Neural Network Confidence Models:* Previous work has explored using confidence intervals to determine prediction intervals for Neural Networks (e.g., [4]). We note, however, that use of confidence intervals is different from determining the confidence in a system in the sense pursued by Cheetham. Confidence intervals "are enclosed in prediction intervals and are concerned with the accuracy of our estimates" [4], whereas confidence in Cheetham's (and our) sense is "the degree of belief in the correctness of the result of a CBR system" [7]. Additional approaches for confidence measures of neural networks with confidence intervals have emphasized the use of maximum likelihood error [24] and confidence intervals in classifier models [30].

*CBR Integrations:* There is a long history of CBR integrations with other types of systems [18], including for black box systems such as neural networks, in which the two systems contribute jointly to problem-solving. For example, in the medical domain to classify skin lesions, a convolutional neural network was used to get features, where those features were passed into the CBR to get retrieved case and output [21]. The proposed integration differs, however, in that the goal is for CBR to contribute to assessment of the other system rather than to the

problem-solving process itself. This work is an instance of twinning of CBR and black box systems, such as ANNs [14].

## 3 Black Box Confidence

### 3.1 The Notion of Confidence of a Black Box System

Developing an approach to assessing black box system confidence by CBR depends first on understanding what "confidence" should represent. The term *confidence* has been used in CBR to refer to the "degree of belief" that the CBR system's solution is correct [5]. This is a fuzzy notion for which values in the range $[0, 1]$ indicate "percentage belief" in the CBR solution. We distinguish this notion from that of trust; confidence is a technical property of our system, whereas trust is a psychological property of humans using a system [17]. This notion of confidence is well understood for CBR systems [7], and can augment the assessments that could be done by examining the CBR system's internal process of retrievals and adaptation.

Black box systems are widely used, with applications for high-stake scenarios. Unfortunately, it is impossible to examine the internals of a black box system in order for a user to develop a level of trust in it (by definition). Thus it would be useful to be able to evaluate a level of confidence in a black box's solution to a problem.

A simple first approach for black box confidence would be to use a global measure of the black box system's performance, such as its accuracy, to ascribe a level of confidence in the system's solutions. This approach is not satisfactory, however, because the global accuracy provides no per-case information. Based on it, equal confidence would be ascribed to all solutions—which would provide no guidance on which solutions to verify or perhaps reject.

Given the ability to ascribe confidence to CBR solutions, it is appealing to use confidence in a CBR system to assist with determining confidence in the black box. We can twin a CBR system with our black box, training the two on the same set of training cases, for each to provide solutions to each problem. To account for differences in system characteristics, the confidence in the CBR system's solution can then be combined with information from other properties of the CBR and black box systems in order to calculate the confidence in the black box's solution.

In the following sections, we discuss potential predictors for black box confidence. Using these predictors, we present three methods for determining confidence in a black box system's solution.

### 3.2 Distance from CBR System Solution

A simple indicator for confidence in a black box solution is the distance of that solution to the solution provided by the CBR system itself. We assume that the solution space has some distance metric normalized to the interval $[0, 1]$. Applying this involves complexities discussed in Sect. 3.6.

### 3.3 CBR Confidence

As mentioned previously, confidence in a CBR system's solution is well understood. We follow Cheetham's approach for calculating the CBR confidence [5–7]. His method involves constructing fuzzy preference functions which map the CBR system's solutions to confidence in those solutions. His approach first sets a confidence scale mapping intervals of error in the CBR solution to confidence intervals. We use his scale, where the Fuzzy Linguistic Term *good* has a confidence interval of 1–0.75 and an error of less than 5%, the term *questionable* has a confidence interval of 0.75–0.5 and an error between 5% and 10%, and the term *poor* has a confidence interval 0.5–0.0 and an error greater than 10%. (Here, "confidence interval" refers to the range of values our "fuzzy" confidence can have. This is distinct from the statistical notion of "confidence interval").

The next step is to pick a few statistical indicators of confidence [6]. We select the following indicators (Cheetham proposes both positive and negative indicators, but we consider only positive):

- Similarity between the given problem case and the most similar retrieved case with the best solution
- Sum of all the similarity scores between the problem case and the $k$-closest retrieved cases with the best solution
- Number of cases with the best solution out of the $k$ closest retrieved cases
- Percentage of the $k$ closest retrieved cases that have the best solution
- Average similarity score between the problem case and the $k$ closest cases with the best solution

As suggested by Cheetham, we then use the *C4.5* algorithm to construct a decision tree for predicting whether a solution will be correct, based on the values of the indicators. We select the indicators highest in the tree as the most important indicators. The goal is to choose the indicators best at predicting a correct CBR solution, because "the more likely the solution is to be correct the higher our confidence should be" [6].

For each selected indicator, we construct a fuzzy preference function mapping that indicator value to confidence in the CBR system's solution. To do this, we treat each case in the training set as a test case (temporarily removing each in turn from the case base). For each training case, we calculate the value of the indicator and the error in the CBR system's proposed solution. Similarly to Cheetham, we plot the indicator values against the error, and fit a cubic regression to the resulting plot using NumPy [23]. We then construct a piecewise linear function from this regression by obtaining the straight lines that meet at the extrema and inflection points.

We next compose the piecewise function (mapping indicator values to error) with the confidence scale (mapping error to confidence) to construct the fuzzy preference function (mapping indicator values to confidence). The details are spelled out by Cheetham in [5], and involve identifying key indicator values at which we are in a different error interval, and hence in a different confidence interval.

We then can determine the confidence in a CBR solution by calculating the selected indicators for a particular solution, for each fuzzy preference function, taking the confidence value for the value of its respective indicator, and taking the mean of these confidences as the final confidence value.

### 3.4 CBR Competence

The CBR system's *competence*, the "the range of target problems that a given system can solve" [28], may be another predictor of black box confidence. If the training data is insufficient for CBR coverage of the problem space, it is plausible that it could be insufficient for the black box as well. We follow Smyth and McKenna's model of CBR competence in which the competence of a CBR system depends on the density and distribution of cases in the case base and the strength of the CBR system's retrieval and adaptation.

### 3.5 Black Box Accuracy

We also expect higher confidence in the black box when the black box itself globally performs well. As a global measure of our black box's performance we use its *accuracy*, i.e. the percent of its own training cases for which the fully-trained black box can successfully provide a solution (within an acceptable threshold). This can be estimated, for example, by leave-one-out testing.

### 3.6 Proposed Methods for Estimating Black Box Confidence

Given the predictors for black box confidence (black box *accuracy*, CBR *competence*, *confidence* in the CBR solution, and *distance* between the CBR solution and black box solution), we propose three ways to combine them to determine confidence in the black box solution. As emphasized before, each method produces a *fuzzy* confidence value within $[0, 1]$ which represents the degree of belief that the black box's solution is correct.

*Naïve Method:* Our first approach is based on the insight that if we are very confident in the CBR system's solution and the distance between the two solutions is small, we should also be very confident in the black box system's solution. Similarly, when we are very confident in the CBR system's solution and the distance between the two solutions is large, we should doubt the black box's solution. When we doubt the CBR system's solution and the distance between the two solutions is small, we expect again to doubt the black box's solution. Following this reasoning, we might infer that the confidence of the black box's solution is given by a formula such as:

$$conf_{BB} = |conf_{CBR} - distance| \tag{1}$$

That is, confidence in the black box's solution is the distance between CBR confidence and solution distance, both scaled to $[0, 1]$.

However, there is a problem with this confidence formulation: If we have low confidence in the CBR system and there is a large distance between the two solutions, this method predicts that we will have high confidence in the black box. This is not necessarily the case, because the distant black box solution could still easily be far from the actual solution. In addition, this formula does not make use of the black box accuracy or CBR competence, both of which should affect the confidence in our black box's solution. Because of these issues, we do not expect this method to predict black box confidence well, but we will test it as a simple baseline.

*Cheetham's Fuzzy Preference Method:* This method provides the most natural extension of CBR confidence to black box confidence. We can simply apply Cheetham's method for determining confidence in a CBR solution to the black box's solution. We use the same confidence scale as for our CBR confidence. For our indicators of black box confidence, we pick the confidence in the CBR system's solution for the same problem and the distance between the black box and CBR solutions. We again construct fuzzy preference functions mapping these indicators to black box confidence (using the training set), and then average the outputs of these fuzzy preference functions for a given black box solution.

Note that we cannot use *accuracy* or *competence* as indicators here, because Cheetham's method requires that indicator values vary per problem case (whereas accuracy and competence are system-global properties). So, like the Naïve Method, this method also does not make use of the black box accuracy or the CBR competence. Rejecting *accuracy* and *competence* as indicators on their own is also justified pragmatically by the fact that they provide no comparative information: They give no indication of which solutions might require further verification.

*Weighted Average Method:* Unlike the previous two methods, this approach attempts to make use of *all* of our confidence predictors. Each of our predictors is on the same interval $[0, 1]$, and we can propose a weighted average:

$$conf_{BB} = \frac{w_1 \times (1 - distance) + w_2 \times conf_{CBR} + w_3 \times comp_{CBR} + w_4 \times acc_{BB}}{w_1 + w_2 + w_3 + w_4} \quad (2)$$

where *distance* is the distance between the two solutions, $conf_{CBR}$ is our confidence in the CBR solution, $comp_{CBR}$ is our CBR competence, and $acc_{BB}$ is our black box accuracy. Note that we use $1 - distance$ because the relationship between the CBR and black box confidences strengthens as the distance *decreases*.

For any domain, weights may be set by hill climbing (see Sect. 4). For concreteness, in our evaluation, hill climbing resulted in the following weights, which define what we henceforth refer to as the Weighted Average method:

$w_1 = 3.0$ (for $1 - distance$)
$w_2 = 0.25$ (for $conf_{CBR}$)

$w_3 = 1.5$ (for $comp_{CBR}$)
$w_4 = 3.0$ (for $acc_{BB}$)

Although the $w_2$ value is comparatively small, its inclusion as a nonzero value improves overall performance, showing that $conf_{CBR}$ provides useful information.

## 4 Evaluating Methods for Black Box Confidence

### 4.1 Assessing Quality of Confidence Predictions

In order to evaluate how well the proposed methods predict black box confidence, we propose a measure of confidence function *quality*. This measure is founded on the principle that ideally, confidence should be high if and only if error in the black box solution is low. Hence, a method for determining black box confidence is "good" if it assigns high confidence whenever there is low error in the solution, and low confidence whenever there is high error in the solution. We propose that the quality of a black box confidence predictor *is the degree to which the black box confidence prediction decreases monotonically with black box system error*. We consider the ability of the measure to properly rank cases by confidence as more important than the particular score it assigns, which could be normalized or scaled to fit domain expectations. The primary goal is to be able to assess which solutions should be ascribed more confidence than others, to identify those which might deserve more scrutiny. Spearman's rank correlation coefficient provides a method to assess the ability of the measure to properly order solutions, i.e., to determine the correlation of the measure's assessment with the actual ordering by accuracy [9].

We evaluate the quality of a confidence method as follows. First, we use it to compute the confidence in the black box solution for each problem in the testing set, and also determine the error in the black box system's solution for each problem in the testing set. We then rank the test problems from lowest confidence to highest confidence and rank the problems again from highest error to lowest error. We then computes the Spearman correlation coefficient $\rho$ for these rankings.

A $\rho$ value of 1 implies that, for that confidence method, black box confidence increases monotonically with reverse-ranked error. That is, black box confidence decreases monotonically with error. Similarly, a $\rho$ value of $-1$ implies that black box confidence increases monotonically with error. We interpret the *strength* of this correlation using the table suggested by Akoglu [3] for Spearman coefficients. Using this table, one may say that the confidence method is "good" whenever we have a $\rho$ value that corresponds to a strong correlation.

### 4.2 Experimental Questions

Given our measure of black box confidence quality, we can begin to experimentally evaluate the methods for black box confidence prediction. Three key questions are:

1. How effectively do the methods predict confidence in the black box system's solution? How do they compare with baselines of using CBR confidence alone or random confidence assignment?
2. Are the methods able to ascribe confidence successfully even when the black box's accuracy is very low?
3. When the black box outperforms the CBR (as is likely to be the case if the black box system is used instead of relying on CBR alone), is CBR confidence a better or worse predictor of black box confidence?

We perform an evaluation addressing questions 1 and 2 in this paper. We do not answer question 3, but we include a discussion of how this could be done in Sect. 6.1.

The first two questions directly deal with the quality of our black box confidence methods. In order to evaluate Question 1, we first establish baseline methods against which to compare. The baselines are:

- *CBR Confidence:* This baseline simply returns the confidence in the CBR system's solution in lieu of confidence in the black box.
- *Random Confidence:* This baseline returns a randomly generated confidence value on the interval $[0, 1]$.

## 5 Testing Confidence Methods with COBB

### 5.1 Overview of COBB System Design

Our testbed system, COBB (Case-based cOnfidence for Black Box), pairs a CBR system and a black box system to predict confidence in the black box system using the methods outlined in Sect. 3.6.

Our particular CBR system is a simple domain-independent retrieval system; it returns the closest case as a solution, with no adaptation. Feature weights were determined by hand, with no attempt to fine-tune weight values. A case is considered to solve a problem $c$ if its solution is within a certain threshold of the solution for $c$ (this threshold is used in competence calculations).

Our black box system is a multi-layer perceptron regressor provided by the SciPy 'scikit-learn' package [13,25]. In order to handle non-numerical attribute values during training, any non-numerical value from a training case is converted using one-hot encoding. If a non-numerical value is encountered during testing that was not seen in the training set, our one-hot encoding codes it as a sequence of zeroes. The COBB system does not rely on any particular properties of this regressor, so is fully general for other black box systems.

### 5.2 Test Domains

We test COBB with four regression datasets from the UCI Machine Learning Repository [11]: Computer Hardware (7 numerical attributes, 2 text attributes, 209 total cases), Student Portuguese Performance (SPP) [8] (16 numerical

attributes, 17 text attributes, 649 total cases), Airfoil (5 numerical attributes, 0 text attributes, 1503 total cases), and SML (23 numerical attributes, 2 text attributes, 4137 total cases). For each domain, we perform 10-fold cross-validation on the domain dataset. For each fold, we train both the CBR and the black box on that fold.

To these domains we apply black box systems of varying accuracies, ranging from very high to very low. The average accuracy (across 10-fold cross-validation) of each black box's domain is as follows: Computer Hardware at 23.7%, SPP at 97.6%, Airfoil as 9.4%, and SML at 70.2%. We treat the Computer Hardware and Airfoil domains as examples for which black box accuracy is low, and similarly the SPP and SML domains as examples for which accuracy is high.

### 5.3 Results for Quality of Confidence Methods

For each domain and each confidence method, we compute the Spearman correlation function[1] to obtain the $\rho$ value per fold. We then take the mean of the $\rho$ values across the folds, and calculate a 95% confidence interval for the mean of the $\rho$ values.

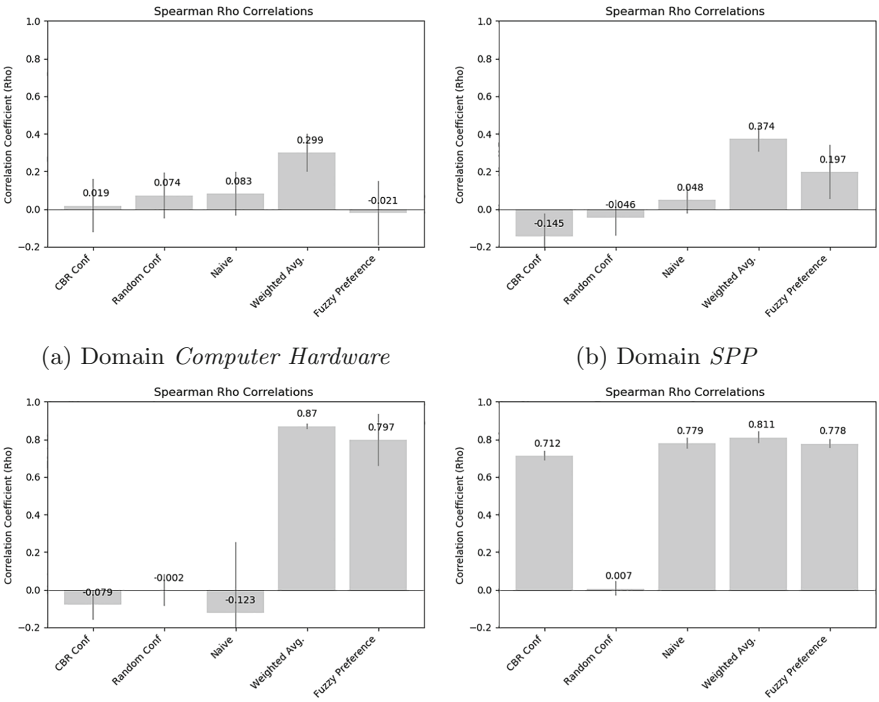
Figure 1 shows the mean Spearman $\rho$ values (across 10-fold cross-validation) along with their confidence intervals for each of our domains.[2] We also include the mean Spearman $\rho$ values and confidence intervals for our two baseline methods, CBR confidence in lieu of black box confidence and random confidence.

To assess the results, first, we compare our confidence methods to the baselines. As shown, the Naïve Method has positive correlation but low quality on the Computer Hardware and SPP domains. The Fuzzy preference method, on the other hand, has higher quality than the base methods on the Airfoil and SPP domains. The Weighted Average method performed consistently well across domains, maintaining a higher quality than both CBR Confidence and Random Confidence. Interestingly, in the SML domain all three methods have higher quality than Random Confidence, but match the quality of just using CBR confidence.

Next, we compare the quality of the confidence methods with each other. For the Computer Hardware and SPP domains, the weighted average method outperforms the Naïve and Fuzzy Preference confidence methods. Within the Airfoil domain, on the other hand, the Weighted Average and Fuzzy Preference confidence methods have roughly the same quality, and this quality is far higher than that for the Naïve method. Surprisingly, in the SML domain all three methods have roughly the same quality.

For the SML domain, the mean $\rho$ values for all three methods are in the "very strong" range (using [3]). In addition, this domain is the only one in which CBR Confidence (on its own) has very strong quality. In the Airfoil domain, only the Weighted Average and Fuzzy Preference methods have mean $\rho$ values in the "very strong" range, whereas the Naïve method has weak negative correlation.

[1] Using SciPy [13].
[2] Plotted using the Matplotlib package [12]).




(a) Domain *Computer Hardware* (b) Domain *SPP*

(c) Domain *Airfoil* (d) Domain *SML*

**Fig. 1.** Spearman $\rho$ values for each confidence method, for each domain.

For the other two domains, we obtain mean $\rho$ values in the weak and moderate range for all three confidence methods. We suspect that this discrepancy is due to how well the black box and CBR systems are paired for each domain, but future work is needed to evaluate this.

## 6 Discussion

### 6.1 Answering the Experimental Questions

The previous results suggest preliminary answers to questions 1 and 2 from Sect. 4.2.

*Question* 1 asks how successfully the three methods predict confidence in the black box solution. The experiments suggest that the Weighted Average method and Fuzzy preference method can give high quality predictions. Compared to the baselines of CBR confidence or random assignment, the Weighted Average method has consistently higher quality, whereas the Fuzzy Preference method only has higher quality in certain domains. The Naïve method has poor quality across our domains and fails to compete with the Weighted Average method in any domain except SML.

*Question* 2 asks whether the confidence methods can still have high quality even when the black box system has low accuracy. As mentioned in Sect. 5.2, the Airfoil domain black box system has very poor accuracy. The Weighted Average and Fuzzy Preference methods give good results in the Airfoil domain despite low black box accuracy.

*Question* 3 asks whether the CBR confidence has better or worse quality (as a black box confidence method) when the black box outperforms the CBR system. Answering this question requires considering instances in which CBR accuracy is low and black box accuracy is high. Tests on the current datasets did not produce any such situations. We intend to address this and further analyze results for the prior questions using additional datasets in future work.

### 6.2 Reflecting on Assumptions Made in COBB

An initial hypothesis for this paper was that CBR confidence could be a good predictor of black box confidence. Surprisingly, in our experiments, CBR confidence by itself had almost no monotonic correlation with error, except in the SML domain. However, when combined with other predictors (as in the Weighted Average method), CBR confidence is a useful predictor. So we must revise the initial hypothesis: The individual indicators *combined* provide a good prediction of black box confidence.

We previously mentioned the potential problem of the Naïve method that it does not apply when our confidence in the CBR is low and there is a large distance between the two solutions. Because the quality of our Naïve method is consistently low across all domains except SML, we conclude that the Naïve method is not a useful approach.

## 7 Explaining Confidence with COBB

The confidence judgments of COBB can be treated as standalone confidence judgments to aid a user determining trust in black box system conclusions, in the tradition of the confidence literature. However, the information developed by COBB can also be used to provide users with useful explanations of the confidence judgment, in two ways:

- Direct explanation from cases: When COBB retrieves a case for a similar problem, and low confidence suggests that additional scrutiny is needed, that case may be presented to the user either as substantiation (if its solution is in agreement) or as a conflict for the user to examine. Depending on the domain, presentation of the case could be paired with traditional information sources in explainable CBR to help the user assess the proposed conflicting solution (e.g., visualizations of attributes [19]). *Bracketing cases*, the most similar cases with and without the same solution [16], could be presented as well. The key added benefit from COBB compared to normal presentation of a retrieved case is that the user's attention need only be drawn to problems likely to be worthy of scrutiny.

– Explanations based on confidence indicators: The values for the specific confidence indicators from Sect. 3.3 can be presented to the user as additional data for assessing the overall confidence judgment.

## 8 Conclusion

We have proposed three methods for determining the confidence of a black box system using a paired CBR system. These methods make use of various predictors of black box confidence (i.e. distance between systems' solutions, CBR confidence, CBR competence, and black box accuracy).

We have also provided a test for quality of a black box confidence method. Applying this test to COBB, the black box confidence method with the best quality in general was the Weighted Average method. In certain domains, the Fuzzy Preference method has nearly as high quality as the Weighted Average method. As expected, the Naïve method has low quality in almost all domains (although in one domain it performs just as well as the former methods). We also noted that there is one domain in which both the Weighted Average and Fuzzy Preference methods are high-quality confidence methods, despite poor black box accuracy.

We see multiple future steps. In addition to performing evaluations on additional domains, we intend to incorporate both negative and positive indicators into the CBR confidence calculation. Some neural network systems output a value characterizing strength of a prediction; the quality of this self-assessment could be compared with that of the methods here, and could also be used as an additional input to the calculations of the weighted method. A more substantial extension would involve systematic study of the methods applied to different CBR and black box systems with varying competences and accuracies. This would enable experimentally answering questions such as Question 4.2. A fundamental question is how closely paired the CBR and black box systems must be for the approach to be useful. For practical application, we intend to explore the feasibility of using an initial calibration phase to determine domain suitability for the COBB approach.

COBB can explain its confidence assessment in terms of confidence indicators, as well as presenting cases for user examination when the CBR confidence assessor detects potential problems. A future topic is analyzing the value of explanations aimed directly at confidence.

The COBB approach was envisioned for situations in which a black box system is more accurate, motivating its use as the primary system but also raising the need for confidence assessment and explanation. An interesting question is whether, when the CBR system is more accurate, the black box system could help in assessing the CBR system confidence. The COBB approach could also be applied to two CBR systems that are independent (e.g., due to different similarity metrics) in parallel, with each assessing confidence in the other, explaining its confidence assessment, and presenting both conclusions to the user or to be combined in an overarching system. This could provide the basis for an approach to system- or user-mediated ensemble reasoning in CBR.

**Acknowledgment.** Testing was performed on a machine supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

## References

1. Aha, D., Agudo, B.D., Garcia, J.R. (eds.): Proceedings of XCBR-2018: First Workshop on Case-Based Reasoning for the Explanation of Intelligent Systems (2018)
2. Aha, D., Darrell, T., Pazzani, M., Reid, D., Sammut, C., Stone, P. (eds.): Proceedings of the IJCAI-17: Workshop on Explainable AI (XAI) (2017)
3. Akoglu, H.: User's guide to correlation coefficients. Turkish J. Emerg. Med. **18**(3), 91–93 (2018)
4. Carney, J.G., Cunningham, P., Bhagwan, U.: Confidence and prediction intervals for neural network ensembles. In: International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339), IJCNN 1999, vol. 2, pp. 1215–1218, July 1999
5. Cheetham, W.: Case-based reasoning with confidence. In: Blanzieri, E., Portinale, L. (eds.) EWCBR 2000. LNCS, vol. 1898, pp. 15–25. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44527-7_3
6. Cheetham, W., Price, J.: Measures of solution accuracy in case-based reasoning systems. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 106–118. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28631-8_9
7. Cheetham, W.E.: Case-based reasoning with confidence. Ph.D. thesis, Rensselaer Polytechnic Institute (1996)
8. Cortez, P., Silva, A.: Using data mining to predict secondary school student performance. In: Brito, A., Teixeira, J. (eds.) Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008), pp. 5–12. EUROSIS (2008)
9. Dodge, Y.: The Concise Encyclopedia of Statistics, chap. 379, pp. 502–505. Springer, New York (2008). https://doi.org/10.1007/978-0-387-32833-1_379
10. Domingos, P.: Knowledge discovery via multiple models. Intell. Data Anal. **2**(3), 187–202 (1998)
11. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
12. Hunter, J.D.: Matplotlib: a 2D graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007)
13. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001). http://www.scipy.org/
14. Keane, M., Kenny, E.: How case based reasoning explained neural networks: An XAI survey of post-hoc explanation-by-example in ANN-CBR twins. arXiv preprint arXiv:1905.07186 (2019)
15. Leake, D.: CBR in context: the present and future. In: Leake, D. (ed.) Case-Based Reasoning: Experiences, Lessons, and Future Directions, pp. 3–30. AAAI Press, Menlo Park (1996). http://www.cs.indiana.edu/~leake/papers/a-96-01.html
16. Leake, D., Birnbaum, L., Hammond, K., Marlow, C., Yang, H.: An integrated interface for proactive, experience-based design support. In: Proceedings of the 2001 International Conference on Intelligent User Interfaces, pp. 101–108 (2001)
17. Madsen, M., Gregor, S.: Measuring human-computer trust. In: Proceedings of the Eleventh Australasian Conference on Information Systems, pp. 6–8 (2000)

18. Marling, C., Sqalli, M., Rissland, E., Munoz-Avila, H., Aha, D.: Case-based reasoning integrations. AI Mag. **23**(1), 69–86 (2002)
19. Massie, S., Craw, S., Wiratunga, N.: A visualisation tool to explain case-base reasoning solutions for tablet formulation. In: Macintosh, A., Ellis, R., Allen, T. (eds.) SGAI 2004. Springer, Berlin (2004). https://doi.org/10.1007/1-84628-103-2_16
20. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
21. Nasiri, S., Helsper, J., Jung, M., Fathi, M.: Enriching a CBR recommender system by classification of skin lesions using deep neural networks. In: ICCBR 2018, p. 86, July 2018
22. Nugent, C., Cunningham, P.: A case-based recommender for black-box systems. Artif. Intell. Rev. **24**(2), 163–178 (2005)
23. Oliphant, T.: NumPy: A Guide to NumPy. Trelgol Publishing, USA (2006). http://www.numpy.org/
24. Papadopoulos, G., Edwards, P.J., Murray, A.F.: Confidence estimation methods for neural networks: a practical comparison. IEEE Trans. Neural Netw. **12**(6), 1278–1287 (2001)
25. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
26. Reilly, J., Smyth, B., McGinty, L., McCarthy, K.: Critiquing with confidence. In: ICCBR, pp. 436–450 (2005)
27. Shin, C., Yun, U.T., Kim, H.K., Park, S.: A hybrid approach of neural network and memory-based learning to data mining. IEEE Trans. Neural Netw. Learning Syst. **11**(3), 637–646 (2000)
28. Smyth, B., McKenna, E.: Modelling the competence of case-bases. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS, vol. 1488, pp. 208–220. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0056334
29. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
30. Zaragoza, H., Buc, D.: Confidence measures for neural network classifiers. In: Proceedings of the Seventh Int. Conf. Information Processing and Management of Uncertainty in Knowlegde-Based Systems (IPMU), vol. 1, pp. 886–893. Editions EDK (Paris), January 1998