



Inference and reasoning in a Bayesian knowledge-intensive CBR system

Hoda Nikpour¹ · Agnar Aamodt¹

Received: 6 February 2020 / Accepted: 7 November 2020 / Published online: 7 January 2021
© The Author(s) 2021

Abstract

This paper presents the inference and reasoning methods in a Bayesian supported knowledge-intensive case-based reasoning (CBR) system called BNCreek. The inference and reasoning process in this system is a combination of three methods. The semantic network inference methods and the CBR method are employed to handle the difficulties of inferencing and reasoning in uncertain domains. The Bayesian network inference methods are employed to make the process more accurate. An experiment from oil well drilling as a complex and uncertain application domain is conducted. The system is evaluated against expert estimations and compared with seven other corresponding systems. The normalized discounted cumulative gain (NDCG) as a rank-based metric, the weighted error (WE), and root-square error (RSE) as the statistical metrics are employed to evaluate different aspects of the system capabilities. The results show the efficiency of the developed inference and reasoning methods.

Keywords Bayesian network · Case-based reasoning · Knowledge-intensive system · Uncertain domains

1 Introduction

The main role of the inference and reasoning process in AI systems is interpreting raw data to generate new information from the domain knowledge. However, it is not possible to create a complete model from an uncertain domain. Therefore, it is necessary to employ a method to make rational decisions when there is not enough information to prove their validity.

Looking into the literature, different inference and reasoning methods are utilized for working in uncertain domains, which are weak-theory domains in the sense that relationships between concepts are uncertain. Hence, statements derived from within domain models are uncertain. The contrast would be perfect-theory domains, in which relations are certain, and statements can be proved true or false. This lack of certainty means that in order to build a representative knowledge model, more knowledge is needed to support

a statement than what would be needed in a proof-oriented system. A stronger model is also facilitated by an integration of various knowledge types, with their corresponding inference and reasoning methods. The system presented here, BNCreek, serves this purpose by combining case-based reasoning (CBR), semantic networks, and Bayesian networks (BN). CBR has proved its capabilities to work in such areas [1]. It often employs simple similarity inference methods like *k*-nearest neighbors (KNN) to reason about the cases' similarity. Aamodt et al. [1] integrated a semantic network structure to the CBR method and developed the Creek knowledge-intensive CBR system. The semantic network provides the underlying value propagation for the system. However, in TrollCreek, the semantic network inference and reasoning methods are implicit, hidden in the code, and mostly not formally defined. Therefore, formal analysis and comparison with other inference methods are challenging [2–5].

BNCreek utilizes the Creek advantages as a background and presents a novel method that makes the inference and reasoning in uncertain domains more accurate. The need for a clearly defined semantic and a more formal treatment of uncertainty led to the incorporation of the Bayesian network model. Bayesian networks as graphical models include a formally defined inference engine that provides the ability of

✉ Hoda Nikpour
hoda.nikpour@ntnu.no

Agnar Aamodt
agnar@ntnu.no

¹ Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

reasoning under uncertain conditions by estimating the probabilities for values of the variables that are not observable.

The first attempts to investigate effects of Bayesian analysis in cooperation with the CBR and the semantic network inference methods can be found in [6,7].

The study presented in this paper digs deeper into the issue and specifically focuses on the BNCreek inference and reasoning methods. The methods address the problem of failure diagnosis in uncertain domains.

The integration of CBR, semantic network, and Bayesian networks facilitate the combination of different kinds of uncertainty handling, improves the similarity assessment, and prepares a formal and flexible basis for reasoning based on probabilities.

The paper is organized into ten parts. Section 2 briefly describes related studies. Sections 3, 4, and 5 explain the details of the inference and reasoning in the semantic network, the Bayesian network, and CBR. Section 6 describes the combined reasoning steps in the system. In Sects. 7 and 8, the oil well drilling knowledge model is presented, and the conducted experiment and its results are discussed. Finally, Sects. 9 and 10 discuss and conclude the detailed advantages and weaknesses of the system.

2 Related work

In BNCreek, the CBR method is employed to cover the lack of generalized knowledge. The semantic network supports similarity assessment. It plays an essential role in increasing the accuracy of inference processes. The Bayesian analysis improves the similarity assessment quality and adds a formally defined probability-based inference method to the system. CBR is integrated with the semantic network and the Bayesian network to make the most out of the domain's well-defined parts.

We make a short review of the similarity assessment methods that are utilized in some CBR-based systems. Then, we focus on summarizing some related work that utilized the CBR, Bayesian network, and semantic network methods on their own or in integration.

Tversky [12] designed a representational similarity model called the contrast model. The model satisfies five assumptions: 1. matching, 2. monotonicity, 3. independence, 4. solvability, and 5. invariance. It preserves the observed similarity order and expresses similarity as a contrast of the measures of the common and the distinctive features: $a * common / (a * common + b * different)$. a and b are the parameters of importance for the common and different features, respectively.

Several similarity models are characterized by different values of the parameters in Tversky's similarity model.

Eisler [13] considered common features over a summation of case features and investigated a quantitative mechanism of the subjective similarity for the dimension of a pitch. Gregson [14] reduced the contrast model into the ratio of common features to overall features, while Bush [15] considered the ratio of common features over the input case features. Their models provide two possible frameworks for analyzing problems in stimulus generalization and discrimination.

Richter et al. [16] designed an expert system called Patdex. They defined a similarity measure to find similar cases for the fault diagnosis of complex machines. Patdex is another system with a similarity model from the contrast model family. It considers the similarity of attribute values as criteria for common and different features. In another version of Patdex, a weighting mechanism is defined as a local similarity measure that determines the similarity between possible symptom values.

myCBR [8] is a case retrieval system targeting at developing customized knowledge models. It provides distance-based knowledge-intensive similarity measures at two levels of local and global similarities for attributes and concepts. Each attribute could have more than one value and several similarity measures. The numeric, textual, and symbolic attributes are covered by data types like Int, Float, Double or String, and taxonomies, respectively.

Cain et al. [11] combined CBR and explanation-based learning and proposed a parametrized similarity model. They utilize the domain model to determine the relevance of a feature to a case. The partial matching of cases and the reasoning within the domain model are lacking in their system.

Been et al. [9] integrated BN and CBR to model the underlying root causes and explanations and introduced a model called Bayesian Case Model (BCM). They used case-based classifiers and BN as two interpretable models to identify the most representative cases and important features. These cases are the exemplars that are the most representative and not necessarily the most similar samples to the current situation. The BCM serves as exemplars for prototype clustering and subspace learning. Their study is motivated by removing the gap between the machine learning methods and human reasoning/learning based on decision-making strategies.

Yuan et al. [10] proposed a probabilistic retrieval method that avoids the traditional similarity measurement. The effect of big data technology in CBR analysis motivates their study. A big case base leads to a heavy and complex computational similarity assessment task. In order to improve the CBR efficiency in dealing with the big data analysis, a Bayesian network is integrated into the CBR system to conduct case retrieval. The retrieval is performed by obtaining the joint probability distribution of all problems from the case base and predicting the solution of a new problem by matching and identifying all problem distributions by the new case.

Aha et al. [17] proposed a decision aids tool based on an integrated BN and CBR problem-solving architecture for solving multiagent planning tasks. In their system, the task models are represented with Bayesian networks and experiences with cases. They used their architecture in Navy force level planning. They focused on multiagent teamwork tasks, which were characterized by incomplete domain models and frequent feedback within the decision making process. Each agent plays a distinct role, and solving a task often relies on teamwork. Their architecture in a selection–execution loop uses Bayesian networks to select the actions. It utilizes CBR to pick a case, implement an action, conduct a learning task for the network, and update the case base until the goal is reached.

Generally, it is not easy to distinguish between inference and reasoning; it would be impossible in some cases. In our study, the underlying detailed processes are considered as inference methods. These methods apply some rules to a knowledge base and derive new information used for the reasoning process. Any new observation about the domain could trigger the inference process and lead to new information. Our system’s primary similarity assessment method is partially inspired by Tversky [12].

In addition, it utilizes a knowledge model to generate explanations for a more accurate retrieval, with some similarities to Cain et al. [11] and Been et al. [9]. BNCreek utilizes Bayesian analysis as a formal basis to improve the overall accuracy of the inference and reasoning process, while [9] employed the BN for classification purposes.

3 Inference and reasoning in the semantic network

The semantic network structure is represented as an edge-labeled directed graph to capture and model a real domain as detailed as possible. In this structure, the nodes are domain concepts that are denoted by the upper-case letter (C) and their instances are denoted by lower-case letters (c). The edges demonstrate the relationships (R) between the concepts. Each relationship is a quadruple of (concept, relation type, relation strength, concept). The relation type (RT) is a label that denotes the kind of relation between two concepts, e.g., has subclass, has instance, causes, has status and their reverse like subclass of. The instances of RT are denoted by (rt). The relation strength (RS) reflects relationship strength by a real number between 0 to 1. The instances of RS are denoted by (rs).

A run-through toy example from a cooking domain will be used as an introductory example. The knowledge model is built for investigating the ingredient and failures in making a good dish. The example is used to clarify the system’s processes and methods. Figure 1 shows a sample of the

knowledge model that has captured and modeled some cooking domain details. It is a toy knowledge model with 46 concepts and 43 edges. It consists of a semantic network, a Bayesian network, and partial descriptions of three cases that are connected to the networks (dashed edges). The relation types and strengths of the relationships are written along the edges. HS and HSt stand for Has subclass and Has status, respectively. Each relationship has a reverse that is not displayed in the figure. In this example, for the simplicity of calculation, all relationships’ strengths and their reverses are set to 0.9. This simplification goes to all the relation types, but the causal ones. The causal relations present the failures of using an inappropriate amount of ingredients.

It should be noted that the causal relations and their relevant concepts in addition to forming the Bayesian network are part of the semantic network as well. This overlap enables interaction between the two networks.

The concept names in the fourth layer are abbreviated. The *ShCE*, *BCE*, and *FCE* stand for *shrimp cooked enough*, *beef cooked enough*, and *fish cooked enough*, respectively. The UC, OC, L, M, and E stand for undercooked, overcooked, little, much, and enough, respectively, and followed by the first letter of its parent. For example, *UCSh* stands for *undercooked shrimp* and *MP* stand for *much pepper*.

Two of the semantic network inference methods are developed in this system, i.e., inheritance and path following. Both of the methods are defined based on Touretzky’s [18] definitions.

The inheritance mechanism is a basic inference method that assigns a semantic interpretation to the relationships of a semantic network knowledge representation. In a simplified definition, the inheritance mechanism interprets how the semantic network relationships inherit over subclass-of relations, see Definition 1. In Fig. 1, *flavoring* is a concept with a property of seasoning food. The *pepper* concept is a subclass of *flavoring*, so it is assumed that it can season the food. In general, a concept is described by its typical properties, which are inherited by more specialized concepts and instances. The inheritance is not absolute; however, an inherited property may be overridden by a specific, local property [19].

Definition 1 (*simplified inheritance mechanism*) If the domain concept X is a subclass of domain concept Y , it inherits its properties as long as there is no exceptional information associated directly with the concept X .

For example, consider the domain concept Y as a bird and the domain concept X , its subclass, as an eagle. Therefore, based on Definition 1, it is assumed that the eagle can fly, which is correct. In another example, consider the domain concept X to be a penguin. It is assumed that X inherits Y ’s properties, but the penguin cannot fly, although it is a bird. Therefore considering the exceptional information, the pen-

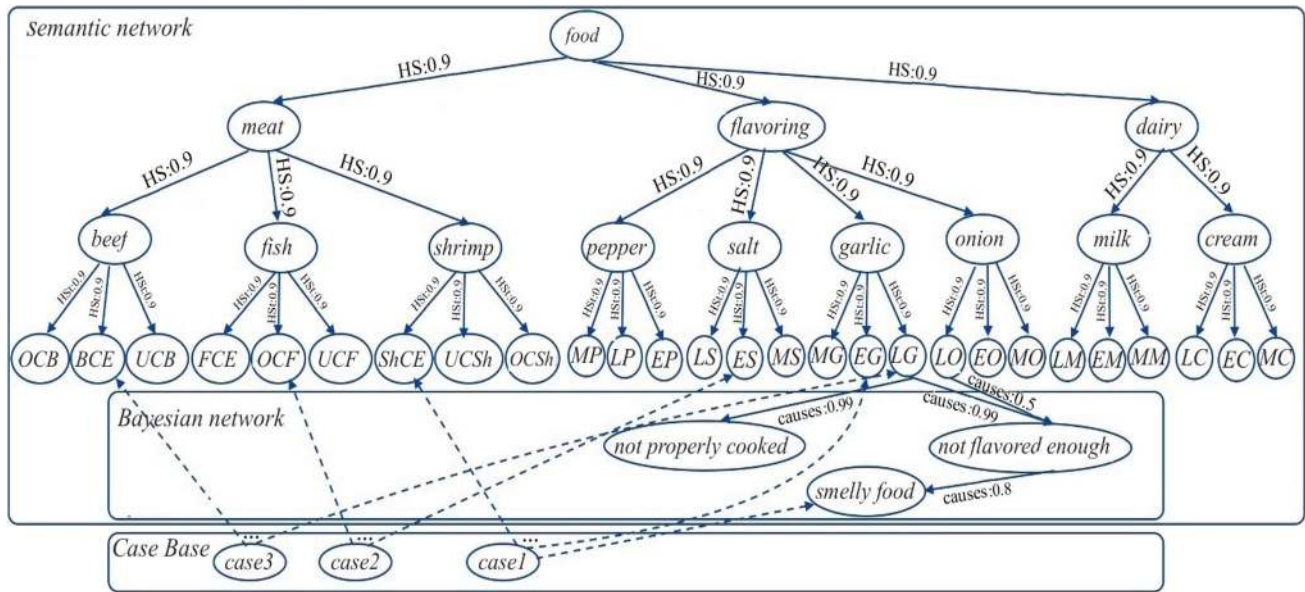


Fig. 1 Figure illustrates parts of the cooking area knowledge model. The relation types and strengths are written along the edges. Reverse relationships are not included. HS and HSt stand for Has subclass, and Has status, respectively. BCE, FCE, and ShCE stand for beef cooked

enough, fish cooked enough, and shrimp cooked enough, respectively. The OC, UC, L, M, and E stand for overcooked, undercooked, little, much, and enough, respectively, and are followed by the first letter of its parent. For example, LS stands for little salt

guin is still a subclass of birds but it does not inherit the flying property. Definition 1 is utilized in generating the knowledge model. It is the basis for the path following method.

Path following is an inference method founded on the inheritance mechanism and helps to navigate through the semantic network by generating a path between two concepts. A path is a set of chained relationships where the starting concept of the first relationship is the starting point of the path, the end concept of any relationship is the start of the next relationship, and the end concept of the last relationship is the ending point of the path. In Fig. 1, the chain of $\{(fish, \text{subclass of}, 0.9, meat)(meat, \text{has subclass}, 0.9, shrimp)\}$ is an example of a path between the fish and shrimp concepts. Definition 2 formally defines a path based on Touretzky's definition.

Definition 2 (Paths) A path P from c_1 to c_n is an ordered, loop-free set of chained relationships.

$$P = (c_1, rt_1, rs_1, c_2)(c_2, rt_2, rs_2, c_3) \dots (c_{n-1}, rt_{n-1}, rs_{n-1}, c_n). \text{ Such that: } rt_i \text{ and } rs_i \text{ are the instances of the relation type and the relation strength.}$$

The paths which are generated for reasoning in the semantic network are called explanation paths (Epath). They are generated by following the sequence of relationships between two concepts. There are two purposes for generating an Epath in BNCreek. The first one is the explanation that the system generates for the user's benefit, explaining the reasoning steps and justifying why a particular conclusion was drawn. The other is the internal explanation that the system con-

structs for itself during the problem-solving process, which has two types: The general explanations explain the similarity of any two concepts, and the causal explanations explain the relevance of an evidence to a failure.

Considering Fig. 1, the system explains the partial similarity between little onion (LO) and little garlic (LG) by pointing to not flavored enough as their common concept and generates the $\{(little\ onion\ (LO), \text{causes}, 0.5, \text{not\ flavored\ enough})(\text{not\ flavored\ enough}, \text{caused by}, 0.99, \text{little\ garlic\ (LG)})\}$ as an Epath that can be used as a general explanation or the user explanation. Consider case1, which has a smelly food symptom. The $\{(smelly\ food, \text{caused by: } 0.8, \text{not\ flavored\ enough})\}$ is an example of a causal explanation.

The internal explanations mostly help to improve the problem-solving quality by working on the cases. A case is a previously experienced situation that consists of a set of features. A case feature (F) is a triple of (concept, relation type, relevance factor). An instance of F is denoted by (f). The relevance factor represents the importance of a feature for a stored case [20]. Cases are considered as domain situation-specific concepts that are connected to the semantic network by their features.

There are three types of case descriptions in this system. A raw case is an unsolved case that is entered into the system by the user based on his primary observations. The pre-processed case is an unsolved case that is modified and extended during the system process. The modified features are referred to as inferred features. The third type is the solved

case, which has a finalized description and a solution. Figure 2 exemplifies a complete description of case1 for each of the three aforementioned types. The figure is explained in detail further in the paper.

An Epath is an explanation path through the knowledge model. In a comparison of two cases, the Epath is an ordered set of relationships that starts with a feature of one case, has some relationships in between, and ends with a feature of another case, i.e., $(f_{1i}, t_1, s_1, c_2) \dots (c_n, t_n, s_n, f_{2j})$. Such that f_{1i} is a feature number i from the first case, and f_{2j} is a feature number j from the second case.

All the relation strengths involved in an explanation chain are combined to form a single explanation strength for an Epath. The similarity of the two cases may be explained by several Epaths. In this case, the explanation strengths of all single Epaths are combined to form the total explanation strength utilizing Eq. 1, [19,21]. According to this equation, the similarity of the identical features is explained by the highest similarity degree, and the similarity of nonidentical features is weakened as more relationships are followed. The $EXPS(f_i, f_j)$ stands for the explanation strength between feature f_i and f_j . RS stands for the relation strength in a path. P stands for the maximum number of the accepted Epaths between two features f_i, f_j , and p_{f_i, f_j} is an instance of it. R and r stand for the number of relationships in a path and an instance of it, respectively.

$$EXPS(f_i, f_j) = 1 - \prod_{p_{f_i, f_j}=1}^P (1 - \prod_{r=1}^R RS_{rp}) \quad (1)$$

The explanations are generated by propagating into the knowledge model along with the relationships between the two features. Dijkstra's algorithm [22], as a well-known algorithm for finding the shortest-path with the specific lengths in a weighted graph, is utilized for this purpose. Any path between two features potentially could be an Epath. A path can be accepted as an Epath if it meets some pre-determined path strength criteria, which are defined by the domain expert. For example, the paths with a strength higher than 0.5 are accepted as the Epath in the food domain.

Consider case2 and case3 shown in the knowledge model of Fig. 1. Let us compute the partial similarity between feature *enough garlic* (EG) from case2 and feature *little garlic* (LG) from case3. Suppose the strength of an acceptable Epath must be higher than 0.6. The Dijkstra algorithm extracts the paths between the two features. The ones with path strengths > 0.6 are approved and listed below as the Epaths.

Ep1: $LG \rightarrow garlic \rightarrow EG$. (path strength: 0.81)

Ep2: $LG \rightarrow garlic \rightarrow flavoring \rightarrow garlic \rightarrow EG$. (path strength: 0.64)

Utilizing Eq. 1, the total explanation strength between LG and EG is: $1 - ((1 - 0.81) * (1 - 0.64)) = 0.93$. In a bigger knowledge model with more details of the domain, the partial similarity between two concepts like LG and EG could be a smaller value.

4 Inference and reasoning in the Bayesian networks

Bayesian inference is the process of updating an uncertain belief within a domain using Bayes' theorem when more information becomes available. Mathematically, Bayesian inference derives the posterior probability distribution by renormalizing the product of the prior probability distribution and the likelihood according to the Bayes' theorem. The prior probability distribution is an estimation of the domain beliefs based on the statistical hypothesis made by an expert. The likelihood expresses the plausibility of values that are assigned to the parameters based on the given information. The posterior probability distribution is a probability distribution of the domain beliefs conditioned on the new information obtained from an experiment or an observation.

$$p(\theta|E) = \frac{p(E|\theta) \times p(\theta)}{p(E)} \quad (2)$$

Equation 2 is the Bayes' theorem in which $p(\theta)$ is the prior probability of the parameter θ , the $p(E|\theta)$ is the likelihood of the evidence and the $p(E)$ is the probability of the events that renormalizes the updated beliefs.

Having access to the Bayesian network's joint distribution tables and given unlimited time, inference in a Bayesian network could easily be done by calculating the posterior probabilities ($p(\theta|E)$) via enumeration. In this way, the whole joint distributions will join up and then the hidden variables will sum out. Increasing the network size exponentially increases the joint distribution that results in the repeated multiplications and makes the process slow. Therefore, more efficient methods are employed to calculate the posterior probabilities.

Bayesian network inference employs two classes of exact and approximate algorithms. The exact BN inference algorithms analytically compute the conditional probability distribution over the variables of interest. They guarantee the correct answer to the query. The approximate BN inference algorithms use an estimation of posterior probabilities for the Bayesian analysis. Both the exact and approximate inference algorithms are NP-hard [23,24]. The exact inference could be applied to a large range of problems, while in more complex and bigger domains, the approximate algorithms may be used, although they do not guarantee the correct answer.

Name: Roasted Shrimp (case1) (raw)			Name: Roasted Shrimp (case1) (pre processed)			Name: Roasted Shrimp (case1) (solved)		
Relation	Value	RF	Relation	Value	RF	Relation	Value	RF
has normal st.	shrimp cooked enough (SHCE)	0.7	has normal st.	shrimp cooked enough (ShCE)	0.7	has normal st.	Shrimp cooked enough (ShCE)	0.7
has normal st.	enough flour (EF)	0.5	has normal st.	enough flour (EF)	0.5	has normal st.	enough flour (EF)	0.5
has normal st.	enough garlic (EG)	0.5	has failure	little garlic (LG)	0.68	has failure	little garlic (LG)	0.68
has normal st.	enough salt (ES)	0.5	has normal st.	enough salt (ES)	0.5	has normal st.	enough salt (ES)	0.5
has normal st.	enough pepper (EP)	0.5	has normal st.	enough pepper (EP)	0.5	has normal st.	enough pepper (EP)	0.5
has normal st.	enough butter (EB)	0.5	has normal st.	enough butter (EB)	0.5	has normal st.	enough butter (EB)	0.5
has symptom	smelly food	0.9	has failure	not flavored enough	0.55	has failure	not flavored enough	0.55
has case st.	Unsolved case		has symptom	smelly food	0.9	has symptom	smelly food	0.9
			has case st.	Unsolved case		has case st.	Solved case	
						has solution	little garlic (LG), not flavored enough	

Fig. 2 Figure illustrates the three changing steps for a case consisting of a raw new case (a), a pre-processed case (b) and finally a solved case (c)

In the current version of BNCreek, the exact BN inference is applied.

A common exact BN inference algorithm is called Variable Elimination. The algorithm implicitly constructs the joint probability distribution given the evidence. Then, it sums out the unwanted variables and constructs a marginal distribution over the variables of interest. This process avoids dealing with bigger joint probability distributions and repetitive computations. However, Variable Elimination is query sensitive. It means for each query, the entire algorithm must run again [25].

One of the most popular exact Bayesian network inference algorithms is the Junction tree algorithm from Lauritzen and Spiegelhalter [26]. Junction tree inference is not query-oriented and can be applied to general graphs. It executes two runs of the Variable Elimination to hold a set of pre-computed factors and then answers any marginal queries quickly. There is no computationally cheaper approach to conduct the exact inference on a general arbitrary graph that shows the algorithm's efficiency. It generalizes Variable Elimination to the simultaneous execution of a large class of queries with the same basic form of computation in all directions at once. These properties made the junction tree inference algorithm a powerful candidate as an inference method in our system [27,28].

In order to implement the Bayesian inference methods, UnBBayes framework is employed. The UnBBayes utilizes the Junction tree algorithm for inferencing in a Bayesian network. It is an open-source Java-based Bayesian package for modeling, learning, and reasoning upon probabilistic networks [29].

As mentioned before, the inference process considers the values of the evidence (symptoms) and computes the posterior probability distribution of the beliefs by conditioning upon the evidence. The Bayesian inferences will be used to reason in the knowledge model via a flow of information through the network in any direction in three types [30].

1. Reasoning from symptoms to cause (diagnostic reasoning). This reasoning occurs in the opposite direction of the network arcs. It updates the system beliefs about specific failures (causes) by observing new symptoms. In Fig. 1, if a case has *smelly food* as a symptom, the system beliefs will update the *LG* and *LOn* as related causes.

2. Reasoning from the causes to the symptoms (predictive reasoning). This reasoning occurs in the direction of the network arcs when new information regarding a possible cause updates the system's beliefs. In Fig. 1, if a case has *LG* as an inferred feature, the system beliefs will be updated for *not properly cooked*. One benefit of predictive reasoning is increasing the accuracy of the similarity assessment. It should be noted that the current system announces a set of failures with their probability degrees as a case solution. There are two ways for an unsolved pre-processed case to have a feature with a failure relation type: 1) when some inferred features are added to the case by the system as a prediction and 2) when the failure comes from the expert to the case as a prediction.

3. Reasoning about the mutual causes of a common effect (intercausal reasoning). This reasoning occurs in the network arc direction when two independent causes have a common effect. If new information about one of the causes is acquired, the other cause's probability will be updated, and the other cause will be explained away. In Fig. 1, *LOn* and *MOn* cause *Notflavored enough*. If a case has *LOn* as one possible cause (inferred by the system or predicted by an expert), the system beliefs will explain away the *LG* as another possible cause.

5 Inference and reasoning methods in CBR

Inference and reasoning quality in CBR is based on its built-in similarity model. The similarity measure in the retrieve process is used to quantify the degree of resemblance between a pair of cases [31]. Two types of similarity models in CBR sys-

tems are considered here. The first type is the distance-based similarity models, and the second is the representational similarity models. The distance-based models calculate the similarity between cases by computing the distance between the constituting objects of the cases. The representational approaches index the input case similar to the structure of the case base or connect the input case into the graphical structure of the case base and compare the cases. Some systems utilize a combination of the two types of similarity models [31]. Similarity assessment plays an essential role in the retrieval process of the CBR cycle [20].

5.1 Similarity assessment

One of the interpretations of an explanation is a method for reasoning. With this interpretation, generating an explanation is inferring a chain that is a good explanation of the similarity of the two features.

Therefore, the similarity of $case_i = (f_{i1}, f_{i2}, \dots, f_{in})$ and $case_j = (f_{j1}, f_{j2}, \dots, f_{jm})$ depends on the number of common features and the explanations' strengths for non identical ones.

The total similarity measure in BNCreek is a ratio model, which is normalized between 0 and 1. The model is based on the Creek similarity model. The underlying assumptions are:

1. The similarity is a reflective relation, i.e., $sim(x, x)$ is equal to one.
2. The similarity is not necessarily a symmetric relation, i.e., $sim(x, y)$ is not necessarily equal to $sim(y, x)$.

A mathematical model measures the similarity between the two cases. It sums up the multiplication of local similarities by the relevance factors of the retrieved features. To keep the total similarities comparable, the degrees are normalized by summing up the relevance factors of the retrieved case.

Equation 3 shows the similarity model of the input case C_{IN} and the retrieved case C_{RE} . RF stands for the relevance factor. It is a positive value for each feature that is set by the expert and measures the importance of any feature for that specific case. The $EXPS(f_i, f_j)$ stands for the explanation strength between coupled features f_i from C_{IN} and f_j from C_{RE} . n and m are the number of features in the input and retrieved cases, respectively [1].

The $\beta(EXPS(f_i, f_j))$ function is defined from input domain $EXPS(f_i, f_j)$, such that $\beta(EXPS(f_i, f_j))$ will be equal to one when $EXPS(f_i, f_j)$ is not zero. For any coupled explained features, the β function keeps the result normalized. The α coefficient is a real number that is multiplied to the β function and controls its intervention amount. α controls the denominator extent which results in controlling the total degrees of similarities. In this version of BNCreek, the

most efficient number for the α coefficient is determined by testing various numbers for any new studying domain or a new case base.

$$sim(C_{IN}, C_{RE}) = \frac{\sum_{i=1}^n \sum_{j=1}^m EXPS(f_i, f_j) * RF_{f_j}}{\alpha * (\sum_{i=1}^n \sum_{j=1}^m \beta(EXPS(f_i, f_j)) * RF_{f_j}) + \sum_{j=1}^m RF_{f_j}} \tag{3}$$

Consider the pre-processed $case_1$ as a new case and the solved $case_2$ as a retrieved case illustrated in Figs. 2 and 4, respectively. We want to calculate their total similarity. Let us suppose, an acceptable path strength should be higher than 0.5 and consider the α coefficient equal to 1.

The explanation strengths ($EXPS(f_i, f_j)$) of the coupled features are multiplied to the relevance factor of the retrieved case features, i.e. f_j , based on Eq. 3 as follows: $((ES, ES) = 1 * 0.5 + (ShCE, OCF) = 0.91 * 0.55 + (LG, EL) = 0.98 * 0.5 + (EP, EL) = 0.94 * 0.5 + (not\ flavored\ enough, EL) = 0.95 * 0.5 + (smelly\ food, EL) = 0.76 * 0.5 + (LG, EG) = 0.93 * 0.5 + (EP, EG) = 0.94 * 0.5 + (not\ flavored\ enough, EG) = 0.89 * 0.5 + (smelly\ food, EG) = 0.63 * 0.5 + (EB, EO) = 0.82 * 0.5 + (ShCE, overcooked) = 0.66 * 0.9) = 5.6$.

The final value for $\alpha * (\sum_{i=1}^n \sum_{j=1}^m \beta(EXPS(f_i, f_j)) * RF_{f_j})$, which is the first part of the denominator is as follows: $1 * (1 * 0.5 + 1 * 0.55 + 1 * 0.5 + 1 * 0.5 + 1 * 0.5 + 1 * 0.5 + 1 * 0.5 + 1 * 0.5 + 1 * 0.5 + 1 * 0.5 + 1 * 0.5 + 1 * 0.9) = 6.45$. The second part of the denominator, i.e., $\sum_{j=1}^m RF_{f_j}$ is equal to 3.45. Therefore, the total similarity of the $case_1$ and $case_2$ is $(5.6 / (6.45 + 3.45))$, which is equal to 56%.

5.2 Retrieve

To perform the retrieval in BNCreek, we are given a set of solved failure cases c_1, \dots, c_n in a case base (CB), a similarity measure (SIM) and a new case (NC). The goal is to retrieve the (c_i) that maximizes the SIM measure for the NC .

The retrieve process conducts the Bayesian reasoning and computes the similarity in association with the structural reasoning in five steps: 1. conduct a Bayesian inference, 2. update case descriptions, 3. update the knowledge model strengths, 4. generate explanations, and 5. compute the similarities.

Conduct a Bayesian inference: The system enters the raw case symptoms to the Bayesian network module and triggers the Bayesian inference. It results in the network posterior distribution that is dynamic in nature, i.e., the probabilities of the dependencies change for any new entered case. Figure 3 shows a small part of the Bayesian network beliefs before and after propagating the symptom. The *smelly food, not flavored*

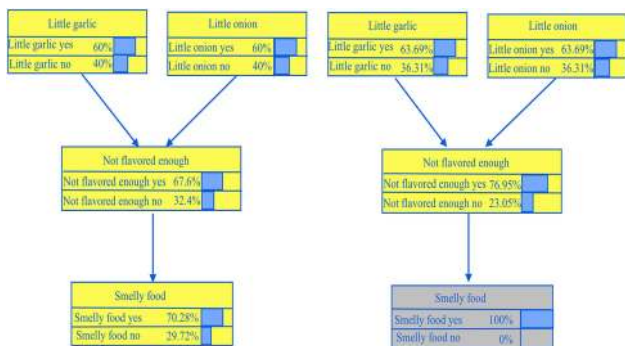


Fig. 3 Part of the Bayesian beliefs before (prior, to the left) and after (posterior, to the right) applying the symptoms into the network. The smelly food is the evidence node that is shown in blue. The example is adapted from [6]

enough, little garlic, and little onion probabilities on the left side are 70%, 67%, 60%, and 60%, respectively. While after propagating the evidence on the right side, they are 100% (shown in blue as the evidence node), 76%, 63%, and 63%, respectively. The observed changes in the network beliefs reflect the effect of customizing the network’s beliefs for any new case.

Update case descriptions: This step extracts informative knowledge from the knowledge model and adds it to the case description.

It extracts the causes of the case symptoms utilizing the posterior distribution. Several causes could be extracted for any symptom. A threshold for the numbers of extracted causes will be determined by the expert based on the knowledge model size. The system modifies the case description by adding the causes as inferred features. The updated case is referred to as a pre-processed case. Figure 2 shows the case1 as a raw and then modified case, referred to as a pre-processed case.

Update the knowledge model strengths: Causal strengths of the semantic network are adjusted dynamically based on the Bayesian posterior beliefs. The other relationships do not change.

Generate explanations: This step utilizes the semantic network reasoning and explains the partial similarities between the case features.

Compute the similarities: This step utilizes the adjusted causal strengths, the pre-processed case description, and generated explanations and computes the total similarity between the input case and cases in the case base. In our example, the total similarity between case1 and case2 is equal to 56% and the total similarity between case1 and case3 is equal to 71%.

Name: Baked fish (case2) (solved)		
Relation	Value	RF
has failure	overcooked fish (OCF)	0.55
has normal st.	enough lemon (EL)	0.5
has normal st.	enough garlic (EG)	0.5
has normal st.	enough salt (ES)	0.5
has normal st.	enough oil (EO)	0.5
has symptom	overcooked	0.9
has case st.	Solved case	
has solution	overcooked fish (OCF)	

Name: Beef steak (case3) (solved)		
Relation	Value	RF
has normal st.	beef cooked enough (BCE)	0.5
has failure	little onion (LO)	0.6
has failure	little garlic (LG)	0.68
has normal st.	enough salt (ES)	0.5
has normal st.	enough pepper (EP)	0.5
has normal st.	enough flour	0.5
has normal st.	enough butter	0.5
has failure	not flavored enough	0.55
has symptom	smelly food	0.9
has case st.	Solved case	
has solution	little garlic (LG), little onion (LO)	

Fig. 4 Tables illustrate the solved case2 and solved case3

5.3 Reuse

The reuse section in this system is a model-based adaptation process. The reuse process as the second part of the Aamodt and plaza 4R CBR cycle [20] considers the retrieved cases’ solutions and generates an appropriate solution for the new case. In this system, the adaptation term refers to all modifications made on the retrieved cases’ solutions to tailor them to be as efficient as the new case’s solution. For simplicity, the adaptation term is even used in rare situations that no changes are necessary for a solution to be presented as a new case solution.

The system adapts previous solutions to a new case in two main phases: 1. adding potential candidates and 2. removing false candidates.

5.3.1 Adding potential candidates

The system performs the adding candidates phase by applying two strategies for two circumstances:

Transfer failures: This strategy considers the solutions of the first *h* best similar cases and modifies their features’ relevance factors using the corresponding causal strengths from the knowledge model. The causal strengths in the knowledge model are dynamically changed by propagating any input cases’ symptoms, leading to the unicity of the causal

strength for any new case. For different domains, a different number of cases are needed. In this version, different values should be examined to determine the best number. Second, the *transfer failures* strategy adds the modified solutions into the initial solution list.

Inferred failures: This strategy is invoked right after the transfer failures strategy. First, it considers each feature of the new case and generates causal explanations with maximum length m to explain the nearest failures' relevance. The expert determines the number of m . Second, the *inferred failures* strategy automatically sets the extracted failures' strengths based on the corresponding causal explanations and adds them into the initial solution list.

5.3.2 Removing false candidates

This phase modifies the potential solution list. It applies two strategies in two circumstances:

The expert distinction: This strategy modifies the potential solution list by applying some pre-implemented rules. The rules are derived to capture the expert principles regarding the domain specifications. For example, some failures have an impact on the occurrence probability of the other failures. These type of dependencies between the failures identifies by domain experts. The system considers each failure from the first phase result and removes the ones which break the rules.

Removing weaker candidates: This strategy considers the adjusted probabilities of each failure in the potential solution list and removes the less probable failures. An expert determines a threshold for deciding if a failure is weak or not.

Finally, the modified solution from the second phase is presented to the user as the input case solution.

Consider pre-processed case1, solved case2, and solved case3, from Figs. 2 and 4, respectively. Let us find a solution for case1. First, we form the potential list by transferring the solutions of case3 and case2 as the first two most similar cases, {*overcooked meat*, *little garlic*, *little onion*}. To make the case solution comparable to similar cases, the system renames the *overcooked fish* to *overcooked meat*. We assign the causal strength from the new case posterior beliefs to the list elements, {*overcooked meat*: 0.4, *little garlic*: 0.63, *little onion*: 0.63}. Then, the system infers the failures which are close enough to the symptoms of case1 from the knowledge model and adds them to them the list as inferred failures. For example, consider {*smelly food* as a symptom of case1. The causal paths from it to the close enough failures are *smelly food*→...→*overcooked shrimp*: (path strength=0.12), *smelly food*→...→*little garlic*: (path strength=0.47), *smelly food*→...→*not flavored enough*: (path strength=0.76). The inferred failures are added to the list, {*overcooked meat*: 0.4, *little garlic*: 0.63, *little onion*:

0.63, *not flavored enough*: 0.76}. The failures that are already in the list keep their strengths. According to the expert's rules, the system removes the items that are entered on the list while they are not relevant to the new case. *little onion*: 0.63 removes as onion is not part of ingredient, {*overcooked meat*: 0.4, *little garlic*: 0.63, *not flavored enough*: 0.76}. *overcooked meat*: 0.4 has a very small possibility and will be removed. The final solution is {*little garlic*: 0.63, *not flavored enough*: 0.76}.

6 Inference and reasoning method in BNCreek

Figure 5 illustrates the combined inference and reasoning process of the system. The process aiming to derive a proper solution is triggered by the symptom list of a new case. The whole process could be divided into two sections of retrieve and reuse from the CBR method. The Bayesian network reasoning and the semantic network reasoning are the subprocesses of it.

As the first step of reasoning in the retrieval process, the Bayesian network reasoning methods update the system beliefs by propagating the events (new case symptoms). Then the semantic network reasoning methods generate the necessary explanations and compute the similarity between the new case and the case base. The final result is a descending order list of the retrieved cases.

The reasoning in the reuse process gets the retrieved cases as input. The role of the Bayesian network reasoning subprocess is to update the potential solutions' beliefs based on the new case specifications. The semantic network reasoning methods, as the next subprocess, generate causal explanations between the new case symptoms and the candidate failures. The result of the reuse process is a list of failures that will be presented to the user as a solution.

7 Oil well drilling knowledge model

Oil well drilling problems frequently occur because of the complexity and variety of the geological formations. Each well may experience both similar and new problems during the drilling operation. While drilling is an expensive operation, access to the experts to solve problems and knowledge acquisition in offshore and onshore rigs is limited, so each failure causes a lot of expense [32].

To extract the oil and gas, a drilling rig that rotates a drill string with a bit attached cuts into the rock and bring petroleum oil hydrocarbons to the surface. This process is called wellbore drilling. To prevent destabilization of the rock in the wellbore walls and lifting rock cuttings to the surface, drilling fluid and mud, that is essentially a mixture of water

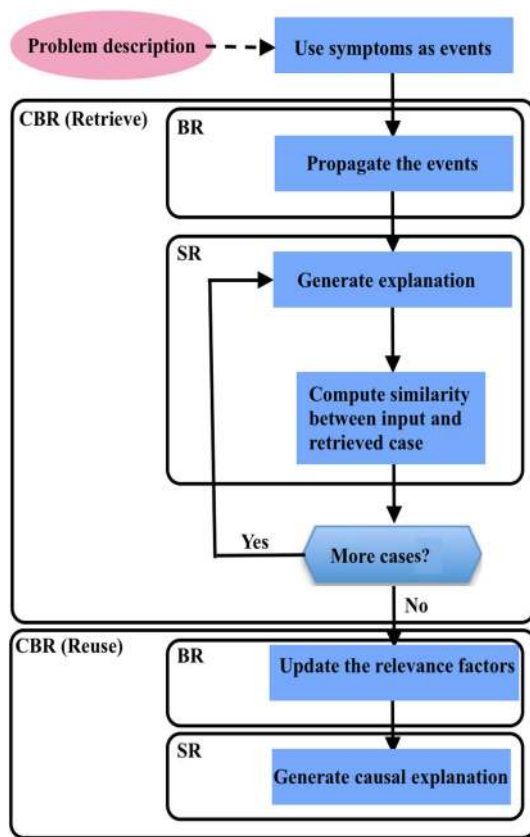


Fig. 5 Figure illustrates the system reasoning process flow chart

and clay is pumped down the inside of the drill pipe and exits at the drill bit and circulates back to the surface outside the drill pipe.

The drilling domain model constitutes the knowledge fundamental to the domain. It describes the drilling process concepts, properties, and the relationships between concepts such as hierarchical structures, functional relations, and causalities. The model gives detailed knowledge and understanding to the system that helps an efficient similarity assessment.

The drilling operation is a widespread process containing approximately 300 properties described by observable or measurable descriptors. Some of the concepts describe simple internal properties (e.g., Cuttings On Shaker (i)). The others represent the non-normal situations, i.e., symptoms (e.g., Cuttings Initial Concentration High (s)) and their causes (e.g., Accumulated Cuttings (f)). There are about 20 or so significant single causes for about 100 non-normal drilling operation situations and many relevant combinations. Diagnosing the failures is a complicated problem because the values of drilling properties are interdependent. Besides, one symptom may have more than one cause that led to the diagnosis of more than one failure. This situation introduces a level of complexity that is difficult to handle with traditional

methods. Due to the problem's complexity, it is not possible to cover all scenarios without intervening the probability theory.

The knowledge model consists of 350 drilling domain concepts and more than 1000 relationships between them, which makes it a very detailed ontology. Forty-five drilling failure cases are generated by an expert. The cases are utilized as queries (input cases) to evaluate the system.

8 System evaluation

We evaluated the presented inference and reasoning method using an experiment from the oil well drilling domain. The system is evaluated in two aspects. First, a quality assessment is performed that evaluates the system's ability to pick out the most important cases. A quantity assessment is then done to evaluate the system's ability to measure the correct similarity degree between two cases. In this regard, the BNCreek reasoning method is compared by simplified versions of seven relevant methods or systems. They can be classified into three types: The systems with distance-based similarity measure, the systems that follow Tversky's [13] contrast similarity model, and the systems that compute the similarity assisted by a knowledge model.

8.1 Experimental set-up

We utilized myCBR [8] (Version: 3.1betaDFKIGmbH). The case base, including 45 cases, is constructed. Three attribute collections named failures, internal concepts, and symptoms, which are subclasses of a concept collection representing drilling concepts, are defined. Each case, in turn, is considered as a query. To create a CBR application with myCBR, the weighted sum similarity measure is utilized as a local–global approach. For the local part, the attributes are divided into a set of weighted local similarity measures. The three types of attributes, i.e., failures, internal concepts, and symptoms, are set to the weights as 0.9, 0.7, 0.7, respectively. Equation 4 is utilized as the global similarity measure for calculating the final similarity value.

$$sim(q, c) = \sum_{i=1}^n w_i * sim_i(q_i, c_i) \quad (4)$$

For each case consisting of n attributes, $sim(q, c)$ is the similarity between a query q and a case c . sim_i and w_i denote the local similarity measure and the weight of attribute i , and Sim represents the global similarity measure [33].

We utilized TrollCreek [1] (version: 0.96devbuild). TrollCreek is an implementation of Creek. It is the background system from which BNCreek is further developed. So their

basic setups share many similarities, with the main difference that BNCreek adds the Bayesian network module. BNCreek is compared with TrollCreek to investigate the effect of the Bayesian network module.

We implemented Tversky [12] contrast similarity model. Eisler [13], Gregson [14], Bush [15], and Patdex [16] are instances of the Tversky model, such that the common features factor (a) and the different features factor (b) for each of the systems are different. The Eisler similarity model has $a = b = 1/2$, the Gregson similarity model has $a = b = 1$, the Bush similarity model has $a = 1, b = 0$ and the Patdex similarity model $a = 1, b = -2$. We focused on the similarity measure of each system and implemented a very simplified version of them; some other possible specifications of them were ignored. As the BNCreek similarity measure is inspired by the Tversky similarity model, it is compared with the systems with Tversky-based similarity measures to investigate the effect of utilizing a knowledge model in the similarity assessment process.

We also implemented a simplified version of the Cain [11] system. Its similarity assessment is a combination of the nearest neighbor algorithm by the effect of case features' relevant degrees achieved from the domain knowledge. The relevant degrees are considered equal to the relevance factors of the solved cases in the BNCreek system. BNCreek is compared with the Cain system as they both employ a knowledge domain and a Bayesian network in their case retrieval process.

Forty-five drilling failure cases are run on all the systems using leave-one-out cross validation. The case base split into 45 subsets, each containing only one case. In each evaluation cycle, the test case is taken as a description of a new problem.

For the quantitative assessment, to keep the consistency of the similarity degrees, the predicted scores are normalized between 0 to 1.

8.2 The qualitative assessment

We have examined the retrieved cases rank that illustrates the system's ability to retrieve the most important cases in the correct order.

8.2.1 Evaluation metrics

We utilized normalized discounted cumulative gain (NDCG) metric in four different cutoffs { cut@5, cut@10, cut@15, cut@20 } that demonstrates the quality of the generated ranked list. The higher values for NDCG reveal the higher performance of the retrieve process.

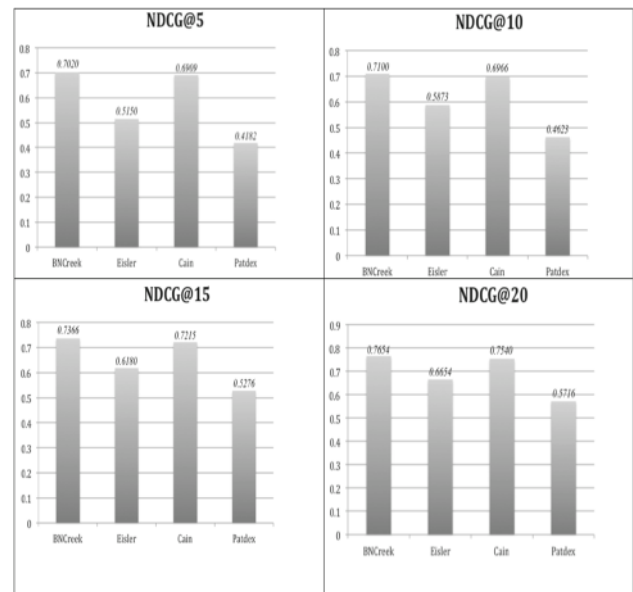


Fig. 6 Four diagrams show { cut@5, cut@10, cut@15, cut@20 } of NDCG rank for BNCreek, Cain, Eisler and Patdex systems

8.2.2 Results

In Fig. 6, we report on NDCG at four ranks { cut@5, cut@10, cut@15, cut@20 }. The horizontal axis shows the studies systems and the vertical axis shows the values for the correct ranks. BNCreek and Cain were the two best systems with the ranking scores { 0.7020, 0.7100, 0.7366, 0.7654 } and { 0.6909, 0.6966, 0.7215, 0.7540 }, respectively. The second best systems were, Eisler and Patdex with the ranking scores { 0.5150, 0.5873, 0.6180, 0.6654 } and { 0.4182, 0.4623, 0.5276, 0.5716 }, respectively. The BNCreek NDCG score was the highest in all cuts which reveals the efficiency of the Bayesian analysis.

8.3 The quantitative assessment

The above-ranking measures only evaluate the correct positions of the retrieved cases relative to each other and do not value the system's accuracy in capturing the correct degree of similarity. Here, we have a quantitative focus on the systems similarity model that investigates the correctness of the similarity degrees.

8.3.1 Evaluation metrics

Two statistical metrics: root square error (RSE) and weighted error (WE) were applied to measure the accuracy of the similarity degrees against the ground truth.

The RSE metric measures the general fitness of the estimated similarity degrees against the expert predictions. It calculates the error based on the differences between the real

and predicted values. It indicates the summation of the divergence between each of the actual points from the predicted values. The smaller error illustrates a better overall prediction of the ground truth. It is calculated as $\sqrt{\sum(i - j)^2}$. Such that, i and j are expert predictions and studied systems estimations, respectively.

In the case retrieval studies, correct retrieval of the higher ranked cases is very important. Higher similarity degrees in most systems are more critical as they belong to the more similar cases. For example, approximating the similarity degree of 80 with an absolute error of 8 is much worse than approximating the similarity degree of 10 with the absolute error of 8. This argument is based on the fact that a case with similarity degree of 10 may not be an important case for the experts while deciding between two cases with 80 or 87 as similarity degrees could be important, and therefore, more accuracy for assessing more similar cases is required. To implement this fact, we have multiplied the absolute errors by the expert predictions as a weight. The utilized weighted error considers a more significant error for the incorrect estimations of the more similar cases. The formula is $\sum|i - j| * i$, such that, i and j are expert predictions and the studied systems' estimations, respectively.

8.3.2 Results

We used RSE as a measure for accuracy. The system with the lowest RSE would be the most accurate one. The WE is utilized to measure the local accuracy of the system in recognizing the first similar cases correctly. The system with the lowest WE has higher local accuracy.

Table 1 shows the RSE and WE values for the six studied systems that are compared with BNCreek, TrollCreek, Eisler, Patdex, myCBR, and Cain in five cuts { 1 best case, 5 best cases, 10 best cases, 35 best cases, 45 best cases }. The first two systems with the fewest errors are BNCreek and Eisler. The RSE and WE for BNCreek are { 1.366, 3.497, 4.543, 8.860, 10.393 }, { 5.875, 31.561, 50.296, 92.242, 101.928 } and for Eisler are { 1.441, 4.479, 6.198, 7.943, 8.304 }, { 7.058, 43.629, 73.635, 107.874, 111.769 }, respectively. The system with the highest errors is myCBR with RSE and WE of { 3.403, 8.220, 10.811, no data, no data }, { 14.904, 79.798, 130.44, no data, no data }, respectively. The other three systems have different scores for different cuts. The RSE and WE for Patdex are { 2.904, 6.110, 7.197, 10.047, 11.364 }, { 13.638, 59.531, 84.831, 125.071, 134.597 } and for Cain are { 2.653, 6.376, 8.324, 10.390, 10.851 }, { 12.638, 64.593, 101.829, 148.36, 153.29 } and for TrollCreek are { 2.881, 6.684, 8.823, no data, no data }, { 13.386, 67.216, 109.73, no data, no data }, respectively.

In addition to the general and local accuracy evaluation, we analyze the system's accuracy to estimate the similarity degrees of the first ten best cases individually. We calculated

the squared errors (SE) for all retrieved cases that are set in the first to tenth positions separately.

Figure 7 shows the SE values' plot versus the first to tenth rank positions for the six systems we are studying. The horizontal axis shows the position of the first ten best cases and the vertical axis is the error value for each system in all ten positions. The BNCreek and TrollCreek diagrams, marked with the blue circles and red triangles, have the lowest and the highest errors according to the SE measure, respectively. The other six systems show medium-level errors.

9 Discussion

In the BNCreek system, the Bayesian and semantic network inference methods are integrated and form a rather complex inference approach. The combined underlying inference method is developed to perform a Bayesian supported knowledge-intensive, case-based reasoning that utilizes model generated explanations in its Retrieve and Reuse phases.

In Fig. 6, for all four cuts of NDCG, BNCreek showed high performance for the correct ranking of the retrieved cases. The Cain system was the best after the BNCreek, with remarkable differences by Eisler and Patdex. The similarity measure of all four systems is inspired by Tversky [12]. In addition, BNCreek and Cain utilize Bayesian analysis for solving problems. We consider their success as a credit for utilizing the Bayesian analysis for problem solving in uncertain domains.

In Table 1 for the RSE values, BNCreek has almost the lowest error in all cuts in comparison with the other systems, which indicates the influence of the knowledge model and Bayesian inference on similarity assessment efficiency. The Eisler system showed a better performance in the middle and weak cuts, which shows BNCreek's weakness in capturing the similarity degrees of the medium similar cases.

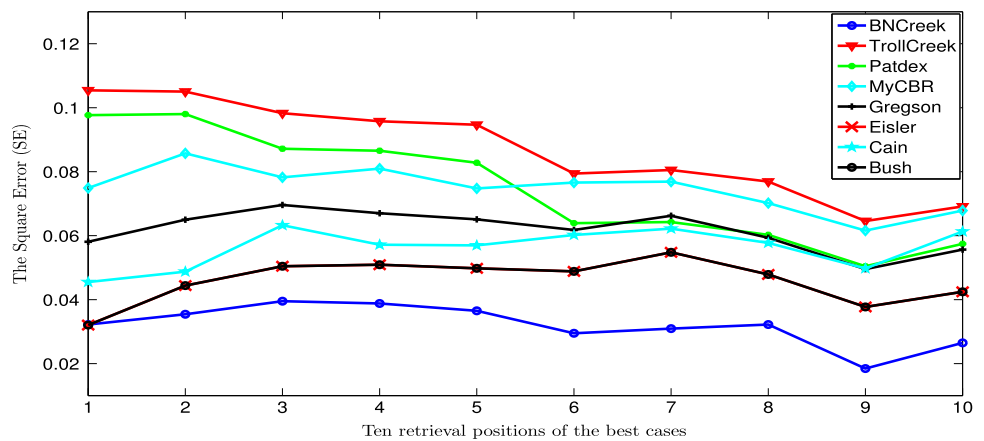
According to the obtained RSE and WE values, BNCreek and Eisler showed the lowest general or local errors while myCBR had the highest error. In comparing BNCreek and myCBR, the remarkable difference is the knowledge model approach that is not employed in myCBR. However, TrollCreek employed the knowledge model approach while it does not have the Bayesian analysis and showed an intermediate performance for error measures. We argue that both the knowledge model and Bayesian analysis were essential and beneficial for BNCreek's lower errors.

BNCreek's general domain model is a combination of the causal and associative models. They are connected with the causal and structural relations, respectively. Similar to Cain and TrollCreek, BNCreek's similarity assessment takes advantage of employing domain knowledge. It uses the domain knowledge to calculate partial similarity degrees for

Table 1 RSE and WE for BNCreek, TrollCreek, Eisler, Patdex, myCBR, and Cain in five cuts { 1 best case, 5 best cases, 10 best cases, 35 best cases, 45 best cases }

01 best case	BNCreek	TrollCreek	Eisler	Patdex	myCBR	Cain
RSE	1.366	2.881	1.441	2.904	3.403	2.653
WE	5.875	13.386	7.058	13.638	14.904	12.638
0 5 best cases	BNCreek	TrollCreek	Eisler	Patdex	myCBR	Cain
RSE	3.497	6.684	4.479	6.110	8.220	6.376
WE	31.561	67.216	43.629	59.531	79.798	64.593
10 best cases	BNCreek	TrollCreek	Eisler	Patdex	myCBR	Cain
RSE	4.543	8.823	6.198	7.197	10.811	8.324
WE	50.296	109.73	73.635	84.831	130.44	101.828
35 best cases	BNCreek	TrollCreek	Eisler	Patdex	myCBR	Cain
RSE	8.860	No data	7.943	10.047	No data	10.390
WE	92.242	No data	107.874	125.071	No data	148.36
45 best cases	BNCreek	TrollCreek	Eisler	Patdex	myCBR	Cain
RSE	10.393	No data	8.304	11.364	No data	10.851
WE	101.928	No data	111.769	134.597	No data	153.29

Fig. 7 Square error (SE) of the first ten ranked cases for the eight systems. The results are for the drilling experiment. The x-axis illustrates the ten retrieval positions of the best cases



nonidentical features while Cain uses the domain knowledge to determine the relevance of features. Moreover, BNCreek utilizes the dynamic causal strengths that are the results of the Bayesian analysis. The model partly acquires the dynamic inputs from the Bayesian inference. The dynamic structure is treated as a posterior distribution of the causal relation strengths. We rely on the expert predictions to estimate the rest of the relation strengths. The Cain system, unlike its high performance in case ranking, was one of the systems with a middle score for error measures.

In Fig. 7, the SE values trend for each system can be related to the system’s specific features. For example, Bush, Eisler, and Gregson, which are direct instances of Tversky’s similarity model, showed a similar trend in all ten ranks. In the drilling experiment, Bush and Eisler overlapped. Although the BNCreek similarity model is inspired by the Creek system, its trend in the first two ranks is not similar to TrollCreek.

The small SE values of BNCreek in the first and the last ranks that belong to the most and the least similar cases are acceptable, but the system performance for the medium similar cases is not good. This demonstrates its weakness in assessing the medium similar cases.

10 Conclusions and future work

In order to make inferencing on the data, acquiring knowledge, representing the obtained knowledge, and reasoning on it, semantic network and Bayesian network analysis are employed. They are formally defined underlying inference and reasoning, and knowledge representation methods. They both effectively contribute to BNCreek system’s reuse of situation-specific knowledge components by case based reasoning.

Our system is examined by an experiment in oil well drilling as an example of a real-world, weak-theory, uncertain domain. The system's performance in problem-solving in this domain has been measured by the normalized discounted cumulative gain (NDCG) as a rank-based metric. The obtained results show the positive effect of employing a knowledge model and a Bayesian analysis to increase the system's general performance.

The local performance of the system regarding its accuracy in assessing the similarities is measured by the root square error (RSE), weighted error (WE), and squared error (SE) as the statistical metrics. The obtained results have verified the effect of causal explanations in improving the similarity assessment accuracy.

Improving the system accuracy in assessing the similarity of the medium level cases will be the first step in our future work.

Acknowledgements The authors would like to thank Prof. Pål Skalle for providing the drilling network model, and preparing the drilling cases, and Prof. Helge Langseth and Dr. Frode Sørmo for their useful suggestions.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aamodt, A.: Knowledge-intensive case-based reasoning in Creek. In: European Conference on Case-Based Reasoning, pp. 1–15, Springer (2004)
- Lacave, C., Díez, F.J.: A review of explanation methods for bayesian networks. *Knowl. Eng. Rev.* **17**(2), 107–127 (2002)
- Velasco, F.J.M.: A bayesian network approach to diagnosing the root cause of failure from trouble tickets. *Artif. Intell. Res.* **1**(2), 75–85 (2012)
- Jensen, F.V., et al.: *An Introduction to Bayesian Networks*, vol. 210. UCL Press, London (1996)
- Sørmo, F.: *Plausible Inheritance; Semantic Network Inference for Case-based Reasoning*. Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim (2000)
- Nikpour, H., Aamodt, A., Bach, K.: Bayesian-supported retrieval in BNCreek: a knowledge-intensive case-based reasoning system. In: International Conference on Case-Based Reasoning, pp. 323–338, Springer (2018)
- Nikpour, H., Aamodt, A., Skalle, P.: Diagnosing root causes and generating graphical explanations by integrating temporal causal reasoning and CBR. In: CEUR Workshop Proceedings (2017)
- Bach, K., Sauer, C., Althoff, K.-D., Roth-Berghofer, T.: Knowledge modelling with the open source tool myCBR. In: CEUR Workshop Proceedings (2014)
- Kim, B., Rudin, C., Shah, J. A.: The Bayesian case model: a generative approach for case-based reasoning and prototype classification. In: *Advances in Neural Information Processing Systems*, pp. 1952–1960, (2014)
- Guo, Y., Chen, W., Zhu, Y.-X., Guo, Y.-Q.: Research on the integrated system of case-based reasoning and Bayesian network. *ISA Trans.* **90**, 213–225 (2019)
- Cain, T., Pazzani, M. J., Silverstein, G.: Using domain knowledge to influence similarity judgements. In: *Proceedings of the Case-Based Reasoning Workshop*, pp. 191–198 (1991)
- Tversky, A.: Features of similarity. *Psychol. Rev.* **84**(4), 327 (1977)
- Eisler, H., Ekman, G.: A mechanism of subjective similarity. *Nordisk Psykologi* **11**(1), 1–10 (1959)
- Gregson, R.A.M.: *Psychometrics of Similarity*. Academic Press, Cambridge (1975)
- Bush, R.R., Mosteller, F.: A model for stimulus generalization and discrimination. *Psychol. Rev.* **58**(6), 413 (1951)
- Richter, M.M., Wess, S.: Similarity, uncertainty and case-based reasoning in patdex. In: Boyer, R.S. (ed.) *Automated Reasoning*, pp. 249–265. Springer, Dordrecht (1991)
- Aha, D.W., Chang, L.W.: *Cooperative Bayesian and case-based reasoning for solving multiagent planning tasks*. Navy Center for, Citeseer (1996)
- Touretzky, D.S.: *The Mathematics of Inheritance Systems*, vol. 8. Morgan Kaufmann, Burlington (1986)
- Aamodt, A.: A knowledge-intensive, integrated approach to problem solving and sustained learning. *Knowledge Engineering and Image Processing Group*. University of Trondheim, pp. 27–85 (1991)
- Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun.* **7**(1), 39–59 (1994)
- Swartout, W.R.: *Rule-based expert systems: The mycin experiments of the stanford heuristic programming project*: BG Buchanan and EH Shortliffe, (Addison-Wesley, Reading, MA, 1984); 702 pages, \$40.50 (1985)
- Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. *Numer. Math.* **1**(1), 269–271 (1959)
- Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. *Artif. Intell.* **42**(2–3), 393–405 (1990)
- Dagum, P., Luby, M.: Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artif. Intell.* **60**(1), 141–153 (1993)
- Paskin, M.: A short course on graphical models: 3. The junction tree algorithms. Short Course notes <http://ai.stanford.edu/paskin/gm-short-course/lec3.pdf> (May 30, 2010) (2003)
- Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **50**(2), 157–194 (1988)
- Barber, D.: *Probabilistic modelling and reasoning: the junction tree algorithm*. Course Notes (2004)
- Bishop, C.M.: *Pattern Recognition and Machine Learning* by Christopher M. Bishop. Springer, Berlin (2006)
- Carvalho, R., Laskey, K., Costa, P., Ladeira, M., Santos, L., Matsumoto, S.: Unbbayes: modeling uncertainty for plausible reasoning in the semantic web. In: Wu, G. (ed.) *Semantic Web*, pp. 953–978. Rijeka, InTech (2010)

30. Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence. CRC Press, Boca Raton (2010)
31. Liao, T.W., Zhang, Z., Mount, C.R.: Similarity measures for retrieval in case-based reasoning systems. *Appl. Artif. Intell.* **12**(4), 267–288 (1998)
32. Shokouhi, S.V., Skalle, P., Aamodt, A.: An overview of case-based reasoning applications in drilling engineering. *Artif. Intell. Rev.* **41**(3), 317–329 (2014)
33. Stahl, A., Roth-Berghofer, T.R.: Rapid prototyping of CBR applications with the open source tool myCBR. In: European conference on case-based reasoning, pp. 615–629, Springer (2008)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.