Bayesian Analysis in a Knowledge-Intensive CBR System

Hoda Nikpour and Agnar Aamodt

Norwegian University of Science and Technology, Department of Computer Science
Sem Sælandsvei 9 Gløshaugen, Trondheim, Norway
{Hoda.Nikpour,Agnar.Aamodt
}@ntnu.no
https://www.idi.ntnu.no

Abstract. This study presents a case-based reasoning system that makes use of general domain knowledge - referred to as a knowledge-intensive CBR system. The system applies Bayesian analysis method aimed at increasing the accuracy of similarity assessment. The idea is to employ the Bayesian posterior distribution for each case symptom to modify the case descriptions and the causal strengths of the ontology. To test the system, two examples from two different application domains, i.e., a "food failure domain" and a "drilling process domain" have been chosen. To evaluate the results, a simplified version of the system is implemented, and the same examples are tested with that system as well. The results from both of the systems are compared with results from a human expert. The obtained results reveal the capability of Bayesian analysis to increase the accuracy of the similarity assessment.

Keywords: Bayesian Analysis, Case-Based Reasoning, Causal Explanations, Knowledge Intensive System

1 Introduction

Knowledge-intensive case-based reasoning (CBR) enables cases to be matched based on semantic rather than purely syntactic criteria. It captures and reuses the human experiences for complex problem solving domains [1], and generates targeted explanations for the user as well as for its own internal reasoning process.

Although pure Case-based reasoning is an efficient method for complex domains problem solving, it is not able to generate an explanation for the proposed solution. Aamodt [2] combined CBR with a semantic network of multi-relational domain knowledge which allows the matching process to compute the similarity based on semantic rather than purely syntactic criteria, leading to capability of explanation generation. A problem with that method is the lack of a formal basis for the semantic network, which made the inference processes within the network difficult to develop and less powerful than desired. The need for a more formal treatment of uncertainty has led to some initial investigations into how a Bayesian Network (BN) model could be incorporated [3], [4]. A Bayesian framework includes an inference engine, builds probabilistic models without introducing unrealistic assumptions of independencies, enables the conditioning over any of the variables and supports any direction of reasoning [6], [7], [5]. Moreover, the nature of Bayesian reasoning enables the explanations generation [5].

Been et al. [8] integrated BN and CBR to model the underlying root causes and explanations with the aim of bridging the gap between the machine learning methods and human decision-making strategy. They used case-based classifiers and BN as two interpretable models to identify the most representative cases and important features. Bruland et al. [9] studied reasoning under uncertainty. They advocated the use of Bayesian networks to model aleatory uncertainty, which works by assigning a probability to a particular state given a known distribution, and case-based reasoning to handle epistemic uncertainty, which refers to cognitive mechanisms of processing knowledge. Houeland et al. [7] presented an automatic reasoning architecture that employs meta reasoning to detect the robustness and performance of systems, which combined case-based reasoning and Bayesian network. Tran et al. [10] used a distributed CBR system to assist operators in finding solutions for faults by determining the cases that share common symptoms. Aamodt et al. [3] focused on retrieval and reuse of past cases. They proposed a BN-powered sub-model as a calculation method that works in parallel with general domain knowledge. Kofod-Petersen et al. [4] investigated weaknesses of Bayesian network in structural and parametric changes by adding case based reasoning functionality to the Bayesian network. Lacave [5] reviewed accomplished studies in Bayesian networks explanation and addressed the remaining challenges in this regard. Koton [11] presented a system called CASEY in which the CBR and a probabilistic causal model are combined to retrieve a qualified case. It takes advantage of the causal model, as a second attempt, after trying a pure CBR to solve the problem.

Aamodt [2] presented a knowledge intensive system called TrollCreek which is an implementation based on the Creek architecture for knowledge-intensive case-based problem solving and learning targeted at addressing problems in open and weak-theory domains. In TrollCreek, case-based reasoning is supported by a model-based reasoning component that utilizes general domain knowledge. The model of general knowledge constitutes a combined frame system and semantic network where each node and each link in the network are explicitly defined in their own frame object. Each node in the network corresponds to a concept in the knowledge model, and each link corresponds to a relation between concepts. A concept may be a general definitional, prototypical concept or a heuristic rule and describes knowledge of domain objects as well as problem solving methods and strategies. A frame represents a node in the network, i.e., a concept in the knowledge model. Each concept is defined by its relations to other concepts represented by the set of slots in the concept's frame definition. A case is also viewed as a concept (a situation-specific concept), and hence it is a node in the network linked into the rest of the network by its case features. The case retrieval process in TrollCreek is a two-step process, in line with the two-step MAC- FAC model

[12], in which the first step is a computationally cheap, syntactic matching process, and the second step is a knowledge-based inference process that attempts to create correspondences between structured representations in the semantic network. In the first step, cases are matched based on a weighed number of identical features, while in the second step paths in the semantic network are identified that represent relation sequences between unidentical features. Based on a method for calculating the closeness between two features at the end of such a sequence, the two features are given a local similarity score. Some of the aforementioned research apply BN in different segments of CBR. The research presented here has been inspired by TrollCreek and is partly based on it; however, it tries to improve the accuracy of retrieval by taking advantage of both BN and CBR. BN-Creek provides a formal basis for causal inference in weak theory domains. The main idea behind BN-Creek is to inject the Bayesian analysis into a semantic network (domain ontology) to assist the retrieve phase of a knowledge-intensive CBR system. BN-Creek and TrollCreek conceptually work on the same ontology and the difference between them stems from the relation strengths, which in Trollcreek are static whereas in BN-Creek change dynamically. This paper investigates the effects of Bayesian analysis within the Creek architecture as a specific knowledge intensive CBR system. In Section 2, the structure of BN-Creek and its retrieve process are presented. Section 3 investigates two examples from two application domains: a food failure domain and a drilling process domain. Section 4 discusses and concludes the paper.

2 BN-Creek

BN-Creek is a knowledge-intensive system to address problems in uncertain domains. The knowledge representation in BN-Creek is a combination of semantic network, Bayesian network and case-based reasoning modules which all together create the knowledge model of the system with a three-layer structure. The semantic layer consists of the ontology nodes (each node refers to a concept in the knowledge domain) which are connected by structural relations, i.e., "subclass-of", "part-of", etc. This layer enables the system to conduct semantic inference through various forms of inheritance. The Bayesian layer consists of the nodes that are connected with the causal relations. The Bayesian layer is strongly integrated with the semantic layer in the form of several separated Bayesian networks. This layer assists the retrieve process to find the potential causes and the most similar cases in addition to generating the causal explanations. There is an individual module named Mirror Bayesian network which interacts with the Bayesian layer and is responsible for the Bayesian inference computational issues. The Mirror Bayesian network is created to keep the implementation complexity low. It gathers a copy of all the small Bayesian networks that are integrated with the semantic network in a computational module. The case base layer is connected to the upper layers through the cases features (features are nodes of the Bayesian or the semantic networks) each possessing a relevance factor (a number that shows the importance of a feature for a stored case [2]).

Fig. 1 illustrates the graphical representation of the system structure. Each box presents one module of the BN-Creek, and the inner boxes make up the outer ones. A set of minor modules form a major module, i.e., "semantic network", "Bayesian network" and "case base" modules form the "general domain knowledge model"; and the "general domain knowledge model" and mirror Bayesian network form the BN-Creek system. The solid arrows show the direction of connecting nodes of each module and the dotted arrow indicates the information flow between the "Bayesian network" and the "mirror Bayesian network".

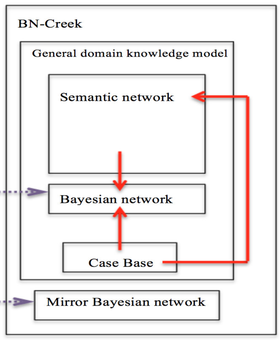
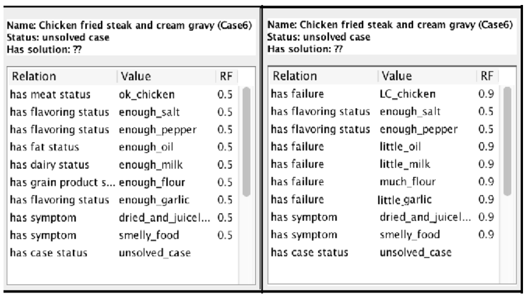


Fig. 1. The graphical representation of BN-Creek.

2.1 The retrieve process

In this section, the retrieve process of BN-Creek system is described. The retrieving process in BN-Creek has three phases, i.e., the standard case preparation, the relation strength adjustments and the similarity assessment. The first two phases are the preprocess for the similarity assessment phase. Algorithm 1 describes the retrieve process in a stepwise manner.



Name: Chicken fried steak and cream gravy (Case6)
Status: unsolved case
Has solution: ??

Relation	Value	RF
has meat status	ok_chicken	0.5
has flavoring status	enough_salt	0.5
has flavoring status	enough_pepper	0.5
has fat status	enough_oil	0.5
has dairy status	enough_milk	0.5
has grain product s...	enough_flour	0.5
has flavoring status	enough_garlic	0.5
has symptom	dried_and_juicel...	0.5
has symptom	smelly_food	0.5
has case status	unsolved_case	

Name: Chicken fried steak and cream gravy (Case6)
Status: unsolved case
Has solution: ??

Relation	Value	RF
has failure	LC_chicken	0.9
has flavoring status	enough_salt	0.5
has flavoring status	enough_pepper	0.5
has failure	little_oil	0.9
has failure	little_milk	0.9
has failure	much_flour	0.9
has failure	little_garlic	0.9
has symptom	dried_and_juicel...	0.9
has symptom	smelly_food	0.9
has case status	unsolved_case	

Fig. 2. The left side represents a raw case description and the right side represents the standard case description.

For more clarification, a run-through example from a food failure domain is given. The domain description and details can be found in the "Experiments and results" section. Suppose a customer ordered a "chicken fried steak and cream gravy" dish. He receives his order and finds it "dried and juiceless" and "smelly" and reports the problem to the chef. The chef employs BN-Creek to find the problems which led to this failed dish to prevent repeating the same mistake in the future.

The standard case preparation phase of the retrieve process is triggered by the creation of a raw case (knowledge about a concrete problem situation which consists of a set of relations and features [2]). In the running example, the chef enters the dish ingredients and the reported observations (represented as symptoms) as a raw input case description and calls it "chicken fried steak and cream gravy (case 6)", illustrated on the left side of Fig. 2.

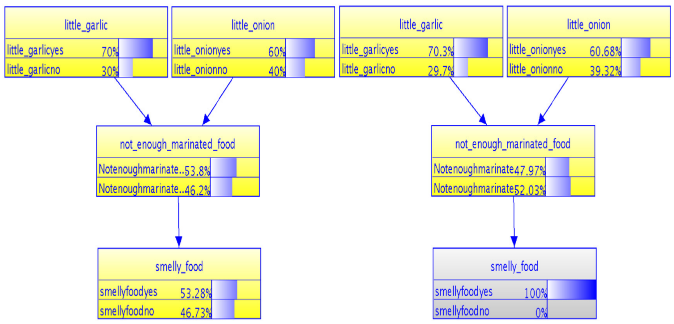


Fig. 3. Part of the Bayesian beliefs before and after applying the symptoms to the network.

The system extracts the symptoms from the raw case description, i.e., "dried and juiceless" and "smelly food" and applies them to the Bayesian network. Afterwards, it updates the Bayesian network beliefs to obtain the posterior distribution (Algorithm 1, lines 1 and 2).

$$p(\theta|symptoms) \propto p(symptoms|\theta) \times p(\theta) \tag{1}$$

The posterior distribution ($p(\theta|symptoms)$) is obtained by Eq. 1. θ, $p(\theta)$ and $p(symptoms|\theta)$ stand for the parameter of distribution, prior distribution and the likelihood of the observations, respectively. The left and the right sides of Fig. 3 show parts of the prior and posterior beliefs of the Bayesian network, respectively.

BN-Creek considers the network posterior distribution and extracts the causal chain behind any of the applied symptoms. In the given example, the causal chains which lead to the observed symptoms are as follows: "little oil" causes "dried and juiceless dish"; "long cooked chicken" causes "dried and juiceless dish"; "little milk" causes "dried and juiceless dish"; "much flour" causes "dried and juiceless dish" and "little garlic" causes "smelly food".

The case description is modified based on the extracted causal chain concepts and forms what is referred to as a standard case description, so, "ok chicken 0.5"

Algorithm 1: Retrieve in BN-Creek

Input : An input raw case.
Output: A sorted list of retrieved failure cases and graphical causal explanations
1 Extract the symptoms of the input case from its case description.
2 Compute the Bayesian layer posterior distribution given the extracted symptoms.
3 Extract the causal chains that cause the case symptoms.
4 Modify the raw input case by adding the causal chain concepts to the case description.
5 Adjust the updated Bayesian beliefs to the knowledge model causal strengths.
6 **while** *not all the case base is tested* **do**
7   Extract one case from case base.
8   Compute the explanation strength between any pair of input and retrieved case findings.
9   Compute the similarity between input and retrieved case.
10 **end**
11 List the matched cases.
12 Generate a graphical causal explanation for the input case.

becomes "LC chicken 0.9"; LC stands for long cooked (see the right side of Fig. 2 and Algorithm 1, lines 3 and 4).

The strength adjustment phase extracts the causal relations strength utilizing the posterior distribution of the Bayesian module. The causal strengths, as opposed to the others which are fixed, are adjusted dynamically corresponding to any new case. Fig. 3 shows the Bayesian beliefs before and after applying the "smelly food" as a symptom to the network. The beliefs of the concepts such as "little garlic" are changed from 0.7 to 0.71 (Algorithm 1, line 5).

The similarity assessment phase follows an "explanation engine" (Fig. 4) with an Activate-Explain-Focus cycle [2]. Activate finds the directly matched findings between input and retrieved cases then the Explain tries to account for the not directly matched findings of the input and retrieved cases. Focus applies the preferences or external constraints to adjust the ranking of the cases.

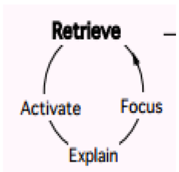


Fig. 4. The retrieve explanation cycle.

BN-Creek considers each of the case base members at the time and utilizes the Dijkstra's Algorithm to extract all possible paths in the knowledge model that represent relation sequences between any findings from the testing case (f_i) and all the findings from the retrieved case (f_j). Consider case 7 (see the case description in Fig. 7) as a retrieved case and the findings "LC chicken" and "LC shrimp" from case 6 (input case) and case 7. The extracted paths between the two findings are displayed in Fig. 5. The different causal strengths reveal the effect of Bayesian analysis which, in contrast to the fixed strength in previous system, computes the posterior beliefs (causal strengths) based on the prior beliefs(from the expert) and the observed symptoms of any particular input case.

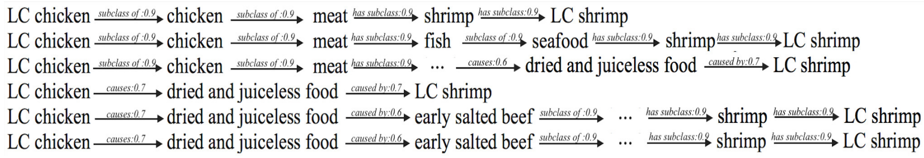


Fig. 5. All possible Paths between two findings from case 6 and case 7 namely, "LC chicken" and "LC shrimp".

To explain the similarity strength between any coupled features, Eq. 2 is employed. To compute the *explanation strength*(f_i, f_j), the strength of any path between (f_i) and (f_j) is computed by multiplying its R relation strengths, then all the P path strengths are multiplied, so, the explanation strengths between "LC chicken" and "LC shrimp" is approximately 0.96 (see Fig. 5 for possible paths between "LC chicken" and "LC shrimp"). For the situations where the paired features are the same (exact matched features), the explanation strength is considered as 1.

$$explanation\ strength(f_i, f_j) = 1 - \prod_{p=1}^{P}(1 - \prod_{r=1}^{R} relationstrength_{rp}) \tag{2}$$

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

The similarity between input case (C_{IN}) and the retrieved case (C_{RE}) is computed by summing up all the multiplication of explanation strength of (f_i, f_j) with relevance factor of f_j divided by the summation of relevance factor of f_j multiplied by β (explanation strength(f_i, f_j)). The function named β (explanation strength(f_i, f_j)) is a binary function which is equal to one when explanation strength(f_i, f_j) is not zero. Number of findings in input and retrieved cases are shown by 'm' and 'n'.

The system computes the similarity between the input case and all the cases from the case base and lists the matching results, displayed on the left side of Table 1 (Algorithm 1 line 6 to 10).

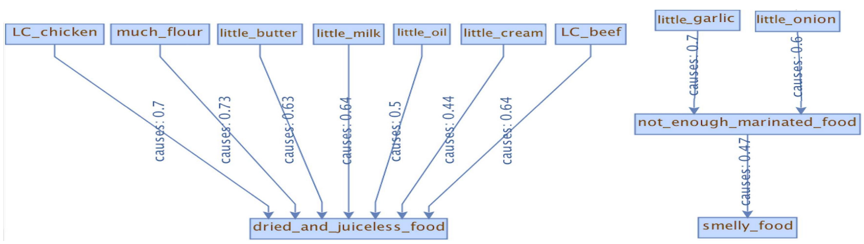
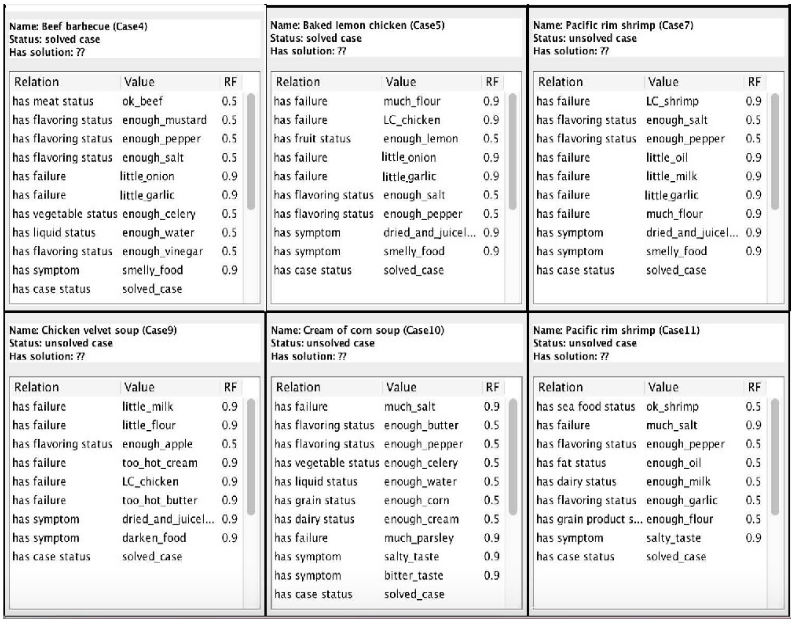


Fig. 6. An example of the graphical causal explanation in the food failure domain.

2.2 The explanations

There are two uses of explanations in knowledge-based systems. One is as the explanation that a system may produce for the benefit of the user, e.g., to explain its reasoning steps or to justify why a particular conclusion was drawn. The other is as the internal explanation that a system may construct for itself during problem solving. BN-Creek provides internal explanations for solving the problems which are called an "explanation strength". A graphical causal explanation is generated to show the extracted causal chains behind the observed symptoms for the benefit of the user. Fig. 6 demonstrates the graphical causal explanation for "chicken fried steak and cream gravy (case 6)". The explanation is the result of Bayesian analysis given the two observations, i.e., "dried and juiceless" and "smelly food". BN-Creek considers the case features and browses into the network to find the related causal chain. The left part of Fig. 6 explains the seven possible causes for "dried and juiceless food" in which the "LC chicken", "little oil", "little milk" and "much flour" are related to the case 6 with causal strengths of 0.7, 0.5,0.64 and 0.73, respectively. The causal strengths demonstrate that "LC chicken" and "much flour" have most effect on causing the "dried and juiceless food". The right part of Fig. 6 shows two causal chains for "smelly food", i.e., "little garlic" causes "not enough marinated food" causes "smelly food" and "little onion" causes "not enough marinated food" causes "smelly food" with causal strengths of 0.32 and 0.28, respectively. The generated Explanation in more uncertain domains like oil well drilling process, play a significant role in computing the similarity (by providing explanation paths) and clarifies the proposed solution for the expert.



Name: Beef barbecue (Case4)
Status: solved case
Has solution: ??

Relation	Value	RF
has meat status	ok_beef	0.5
has flavoring status	enough_mustard	0.5
has flavoring status	enough_pepper	0.5
has flavoring status	enough_salt	0.5
has failure	little_onion	0.9
has failure	little_garlic	0.9
has vegetable status	enough_celery	0.5
has liquid status	enough_water	0.5
has flavoring status	enough_vinegar	0.5
has symptom	smelly_food	0.9
has case status	solved_case	

Name: Baked lemon chicken (Case5)
Status: solved case
Has solution: ??

Relation	Value	RF
has failure	much_flour	0.9
has failure	LC_chicken	0.9
has fruit status	enough_lemon	0.5
has failure	little_onion	0.9
has failure	little_garlic	0.9
has flavoring status	enough_salt	0.5
has flavoring status	enough_pepper	0.5
has symptom	dried_and_juicel...	0.9
has symptom	smelly_food	0.9
has case status	solved_case	

Name: Pacific rim shrimp (Case7)
Status: unsolved case
Has solution: ??

Relation	Value	RF
has failure	LC_shrimp	0.9
has flavoring status	enough_salt	0.5
has flavoring status	enough_pepper	0.5
has failure	little_oil	0.9
has failure	little_milk	0.9
has failure	little_garlic	0.9
has failure	much_flour	0.9
has symptom	dried_and_juicel...	0.9
has symptom	smelly_food	0.9
has case status	solved_case	

Name: Chicken velvet soup (Case9)
Status: unsolved case
Has solution: ??

Relation	Value	RF
has failure	little_milk	0.9
has failure	little_flour	0.9
has flavoring status	enough_apple	0.5
has failure	too_hot_cream	0.9
has failure	LC_chicken	0.9
has failure	too_hot_butter	0.9
has symptom	dried_and_juicel...	0.9
has symptom	darken_food	0.9
has case status	solved_case	

Name: Cream of corn soup (Case10)
Status: unsolved case
Has solution: ??

Relation	Value	RF
has failure	much_salt	0.9
has flavoring status	enough_butter	0.5
has flavoring status	enough_pepper	0.5
has vegetable status	enough_celery	0.5
has liquid status	enough_water	0.5
has grain status	enough_corn	0.5
has dairy status	enough_cream	0.5
has failure	much_parsley	0.9
has symptom	salty_taste	0.9
has symptom	bitter_taste	0.9
has case status	solved_case	

Name: Pacific rim shrimp (Case11)
Status: unsolved case
Has solution: ??

Relation	Value	RF
has sea food status	ok_shrimp	0.5
has failure	much_salt	0.9
has flavoring status	enough_pepper	0.5
has fat status	enough_oil	0.5
has dairy status	enough_milk	0.5
has flavoring status	enough_garlic	0.5
has grain product s...	enough_flour	0.5
has symptom	salty_taste	0.9
has case status	solved_case	

Fig. 7. The six food failure cases description.

3 Experiments and results

In this section first the employed evaluation methodology is introduced then the system is tested by studying two examples from two different application domains: "food failure domain" and "drilling process domain".

3.1 System evaluation methodology

To evaluate the effects of the Bayesian analysis in similarity assessment, a test system is implemented. This system is based on the BN-Creek architecture, but with all the capabilities that stem from the Bayesian analysis removed. The resultant system is a simplified version of TrollCreek[2].

Without Bayesian inference, the simplified TrollCreek is not able to extract the potential failure causal chains of an input case. Then, the raw input case from the user is considered as the finalized case description. Moreover, the system is not able to adjust the causal strength dynamically and uses the predefined causal strength for all parts of the knowledge model.

In each application domain, the given example results are also manually predicted by a domain expert, and the results of BN-Creek and TrollCreek are evaluated by comparing with the expert. The system with closer results to the expert's prediction is considered as the more efficient system.

BN-Creek			TrollCreek			Expert		
Input		Matched	Input		Matched	Input		Matched
case6	88%	case7	case6	82%	case7	case6	strong	case7,5,3,2
case6	56%	case5	case6	58%	case5	case6	medium	case9,4
case6	47%	case3	case6	29%	case9	case6	weak	case1,8,10,11
case6	37%	case2	case6	23%	case11			
case6	22%	case9	case6	20%	case3			
case6	15%	case8	case6	15%	case2			
case6	15%	case11	case6	15%	case4			
case6	14%	case4	case6	10%	case1			
case6	13%	case1	case6	4%	case8			
case6	11%	case10	case6	4%	case10			

Table 1. The left side illustrates the results of BN-Creek, the middle is the results of the simplified TrollCreek and the right side is the expert's predicted results.

3.2 Food failure domain

The main type of application domains for the presented system is uncertain domains, and using this system for more certain domains such as "food failure domain" in some cases doesn't make sense. However, on account of the simple nature of "food failure domain", which leads to a better understanding of the system process, an example from this domain is presented.

A food recipe failure ontology inspired by Taaable[14] is utilized with some modifications that are made to fit the ontology to BN-Creek structure, i.e., adding causal relations. The causal relations present the failures of using an appropriate amount of ingredients. Fifteen recipes are examined and simplified to the basic elements (e.g., Gouda cheese simplified to cheese) and eleven failure cases are created.

Evaluating the results To analyze the BN-Creek efficiency the results of one given example on the three different systems,i.e., BN-Creek, simplified Troll-Creek and the expert are considered and presented in Table 1.

The results of running BN-Creek with "chicken fried steak and cream gravy (case 6)" as an input case (the run through example from section 2) is displayed on the left side of Table 1.

The simplified TrollCreek is run with the "chicken fried steak and cream gravy (case 6)" as its input case as well. The obtained results are listed in the middle part of Table 1.

Case name: Wellbore Clean 03 (Case5)		
has activity	Activity Of Tripping In(s)	0.9
has activity	Activity Of Directional Drilling(s)	0.9
has activity	Overpull(s)	0.9
has geology	Shale Swelling Invisible(i)	1.0
has observation	Wellbore Enlarged(i)	1.0
has observation	LC(f)	1.0
has observation	Well Openhole Long(ss)	1.0
has observation	Fm Soft(s)	1.0
has observation	Stuck Pipe Mechanically(f)	1.0
has observation	Torque High(s)	1.0
has observation	Losses Serious(s)	1.0
has observation	Accumulated Cavings(err)	1.0
has observation	Cavings Produced(i)	1.0
has outcome	Time Long(i)	0.9
has solution	ECD - Collapse D Low(s)	1.0
has case status	Solved Case	1.0

Case name: Wellbore Clean 01 (Case3)		
has activity	Activity Of Drilling(s)	0.9
has observation	Build/Drop Section Inside Openhole(ss)	1.0
has observation	Torque Erratic(s)	1.0
has observation	Pressure Spike(s)	1.0
has observation	Took Weight(s)	1.0
has observation	Fm Hard Stringer(s)	1.0
has observation	Fm Laminated(s)	1.0
has observation	Well Length High(ss)	1.0
has outcome	Fm Soft(s)	0.9
has solution	Activity Of Reaming(s)	1.0
has state	Cuttings Concentration High(i)	1.0
has case status	Solved Case	1.0

Case name: Wellbore Clean 02 (Case4)		
has activity	Activity Of Tripping In(s)	0.9
has alarm	Cuttings Concentration High(i)	1.0
has alarm	Overpull(s)	1.0
has failure	LC(f)	0.9
has geology	Build/Drop Section Inside Openhole(ss)	0.01
has observation	Well Length High(ss)	1.0
has observation	Fm Laminated(s)	1.0
has observation	Pressure Spike(s)	1.0
has observation	Wellbore Enlarged(i)	1.0
has observation	Took Weight(s)	1.0
has observation	Torque Erratic(s)	1.0
has operation	Activity Of Directional Drilling(s)	1.0
has repair activity	Activity Of Reaming(s)	0.01
has case status	Solved Case	1.0

Case name: kick2 (Case2)		
has activity	Activity Of Drilling(s)	0.5
has parameter	ECD - Pore D Low(s)	0.5
has geology	Shallow Gas Zone Expected(ss)	1.0
has observation	Well Depth Shallow(ss)	1.0
has observation	Kick(f)	1.0
has observation	Losses Seepage(s)	1.0
has observation	Mud Gas Content High(i)	1.0
has case status	Solved Case	1.0

Case name: Wellbore Clean 04 (Case6)		
has finding	Cavings Produced(i)	0.95
has finding	Mud Water Activity Low(ss)	0.8
has finding	Cavings On Shaker(i)	0.95
has finding	WBM(ss)	0.8
has finding	Mud YP High(ss)	0.5
has finding	Mud Weight High(ss)	0.01
has finding	Mud Water Activity High(ss)	0.8
has case status	UnSolved Case	1.0

Case name: Wellbore Clean 05 (Case7)		
has finding	Losses Serious(s)	0.8
has finding	WBM(ss)	0.5
has finding	Stuck Pipe Mechanically(f)	0.5
has finding	Shale Swelling Invisible(i)	0.5
has finding	Cuttings On Shaker(i)	0.8
has finding	Mud Weight High(ss)	0.5
has finding	Mud Water Activity Low(ss)	0.5
has activity	ECD - Collapse D Low(s)	0.01
has case status	Solved Case	1.0

Fig. 8. The six drilling cases description.

Due to the simplicity of the domain, the analysis of the results predicted by the expert is as follows. Consider the "chicken fried steak and cream gravy (case 6)" as an input case. Cases 7, 5, 3 and 2 have the same observations (see Fig. 7) as case 6, and are expected to be the most similar cases. Case 7 has almost the same case description as case 6 and the only difference is "long cooked chicken" that is replaced by "long cooked shrimp", so it is the most similar case to it. Cases 5, 3 and 2 are sorted with the same logic. Cases 4 and 9 hold one shared observation with case 6, and considering their causal chain they are expected to be the second best similar cases. Cases 1, 8, 10, 11 don't carry any observation in common with case 6 and are expected to be the weakest similar cases. Then, the similarity degree between case 6 and the retrieved cases can be distributed in 3 levels. Very similar cases which fill the first four ranks, medium similar cases which fill the 5th and the 6th ranks and the weak similar cases which fill the rest of the positions. The right side of Table 1 displays the expected retrieved cases.

By comparing the BN-Creek and the Simplified TrollCreek results with the expected results, BN-Creek detected the very similar cases correctly whereas the simplified TrollCreek detected cases 9 and 11 erroneously. For the medium and weak similar cases, BN-Creek categorized cases 4 and 8 incorrectly and the simplified TrollCreek categorized case 4, 11 and 2 wrong.

3.3 Drilling process

The oil and gas domain is an uncertain domain with a weak theory in which implementing ad hoc solutions frequently leads to a reemergence of the problem and repetition of the cycle. These types of domains are the main application domains addressed by BN-Creek.

A drilling process ontology created by Prof Paal Skalle [15] together with seven failure cases, is utilized. Fig. 8 shows the cases descriptions. A case named "Wellbore clean4 (case 6)" is randomly considered as an input case. The retrieved cases are listed on the left side of Table 2.

Evaluating the results The results of running BN-Creek with "Wellbore clean4 (case 6)" as an input case is displayed on the left side of Table 2.

The simplified TrollCreek is run with the same input case, i.e., "Wellbore clean4 (case 6)" and the retrieved cases are listed in the middle part of Table 2.

The results of the two systems (BN-Creek and simplified TrollCreek) are compared with the predicted results by expert [15] and listed on the right side of Table 2.

Based on the expert's prediction, cases 3, 7, 4, 5, 1 and 2 are the most similar ones to "Wellbore clean4 (case 6)", respectively. By comparing the results of BN-Creek and simplified TrollCreek with the expected results, BN-Creek revealed case 4 stronger than case 7, which is wrong, but the rest of the similarity order is captured correctly. The simplified TrollCreek recognized case 4 and 5 to be stronger than cases 3 and 7 which is also wrong.

BN-Creek			TrollCreek			Expert	
Input		Matched	Input		Matched	Input	Matched
case6	79%	case3	case6	62%	case4	case6	case3
case6	78%	case4	case6	54%	case3	case6	case7
case6	68%	case7	case6	53%	case5	case6	case4
case6	64%	case5	case6	48%	case7	case6	case5
case6	63%	case1	case6	48%	case1	case6	case1
case6	61%	case2	case6	46%	case2	case6	case2

Table 2. The left side shows the results of BN-Creek, the middle is the results of the simplified TrollCreek and the right side is the expert predicted results.

4 Discussion and conclusion

In this section, the obtained results from the two utilized examples are analyzed and the advantages and weaknesses of the BN-Creek are addressed.

Case 11, from the food domain example, has almost the same ingredients as the input case (case 6) and their differences originated from "LC chicken" which is replaced by "ok shrimp" in case 11 and their symptoms which are not the same. Case 11 is categorised as a very similar case by the simplified TrollCreek system while, based on the expert's prediction, it is the least similar case to case 6. This problem stems from similarity assessment mechanism in simplified TrollCreek which incorporates the raw case descriptions without considering the effect of different symptoms on cases (e.g. a peppery sandwich is more similar to a peppery steak than to a salty sandwich) which leads to a wrong categorising of the cases such as case 11. Whereas BN-Creek, in its three phases, injects the effect of Bayesian analysis into the case description and similarity assessment process. So it is eligible to incorporate the effect of symptoms in the similarity assessment in such a way that after passing the first two phases, a modified case description, with the adjusted relevance factors is produced as an input for the third phase. The third phase computes the similarity based on the modified causal strength of the ontology which leads to a correct categorisation of the instances such as case 11.

Case 3 from "drilling process domain" doesn't have any feature in common with the input case, i.e., "Wellbore clean4 (case 6)". Case 3 is categorised as the second best case by the simplified TrollCreek system while, based on the expert's prediction, it is the most similar case to case 6. The problem with the simplified TrollCreek is originated in its similarity assessment method that uses the static relation strengths to compute the similarity which leads to a wrong categorising of the cases such as case 3. Whereas BN-Creek, in its third phase, adjusts the relations strengths based on the BN posterior distribution dynamically which leads to capturing the similarity between instances like Cases 6 and 3 correctly. Fig. 9 demonstrates the differences between the matching degrees of three sampled features from Cases 6 and 3.

BN-Creek		
Input		Retrieved
Cavings On Shaker(i)	0.97	Took Weight (s)
Cavings On Shaker(i)	0.68	Fm Soft (s)
Cavings On Shaker(i)	0.87	Torque Erratic (s)

TrollCreek		
Input		Retrieved
Cavings On Shaker(i)	0.61	Took Weight (s)
Cavings On Shaker(i)	0.29	Fm Soft (s)
Cavings On Shaker(i)	0.48	Torque Erratic (s)

Fig. 9. The left side displays the indirectly matched degrees between two features from the input and retrieved cases resulted from BN-Creek and the right side is the same results from the TrollCreek system.

BN-Creek in both of the examples didn't manage to list all the similarity orders correctly and the problem mostly relates to the medium similar cases. It is speculated that this weakness stems from the imprecise prior distribution of the Bayesian beliefs which spreads to the modified relevance factors as well and decreases the accuracy of the system.

BN-Creek showed a higher performance than the simplified TrollCreek, base on two application domains test results. This indicates the Bayesian analysis efficiency for similarity assessment, independent from the application domain.

5 Future studies

The next two future steps for this research will focus on learning methods and temporal reasoning. To increase the accuracy of the similarity assessment, machine learning methods would be employed to minimise the error of Bayesian prior distribution and relevance factors. In application domains with large knowledge models, the causal chains length are too long which leads to a high computational complexity. Some primary studies are conducting to employ the temporal reasoning and use the time sequence of the causal relations to provide a threshold for the causal chains length[13].

References

1. Odd Erik Gundersen, Frode Sørmo, Agnar Aamodt, Pål Skalle: A real-time decision support system for high cost oil-well drilling operations. AI Magazine, Volume 34, Number 1, Spring 2013. ISSN-0738-4602. pgs. 21-32.
2. Aamodt, Agnar. "Knowledge-intensive case-based reasoning in Creek." In Advances in Case-Based Reasoning, pp. 1-15. Springer Berlin Heidelberg, 2004.
3. Agnar Aamodt, Helge Langseth: Integrating Bayesian Networks into knowledgeintensive CBR. In Case-based reasoning integrations; Papers from theAAAI workshop. David Aha, Jody J. Daniels (eds.).Technical Report WS-98-15. AAAI Press, Menlo Park, 1998. ISBN 1-57735-068-5. pp1-6.
4. Anders Kofod-Petersen, Helge Langseth, Agnar Aamodt: Explanations in Bayesian Networks using provenance trough case-based reasoning. ICCBR 2010, Alessandria, Italy. Workshop Proceedings. School of Electrical Engineering, Ohio University, 2010. pp. 79-86.
5. Lacave, Carmen, and Francisco J. Diez. "A review of explanation methods for Bayesian Networks." The Knowledge Engineering Review 17, no. 02 (2002): 107-127.
6. Velasco, Fco Javier Molinero. "A Bayesian Network approach to diagnosing the root cause of failure from Trouble Tickets." Artificial Intelligence Research1, no. 2 (2012): p75.
7. HOUELAND, Tor Gunnar, Tore BRULAND, Agnar AAMODT, and Helge LANGSETH. "A hybrid metareasoning architecture combining case-based reasoning and Bayesian Networks."
8. Kim, Been, Cynthia Rudin, and Julie A. Shah. "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.
9. Bruland, Tore, Agnar Aamodt, and Helge Langseth. "Architectures integrating case-based reasoning and Bayesian Networks for clinical decision support." In Intelligent Information Processing V, pp. 82-91. Springer Berlin Heidelberg, 2010.
10. Tran, Ha Manh, and Jurgen Schonwalder. "Fault resolution in case-based reasoning." In PRICAI 2008: Trends in Artificial Intelligence, pp. 417-429. Springer Berlin Heidelberg, 2008.
11. Koton, Phyllis A. "Using experience in learning and problem solving." PhD diss., Massachusetts Institute of Technology, 1988.
12. Forbus, K., Gentner, D. and Law, K.: MAC/FAC: A model of Similarity-based Retrieval. Cognitive Science, 19(2), April-June, 1995, pgs. 141-205.
13. Long, William. "Temporal reasoning for diagnosis in a causal probabilistic knowledge base." Artificial intelligence in medicine 8, no. 3 (1996): 193-215.
14. Badra, Fadi, Julien Cojan, Amelie Cordier, Jean Lieber, Thomas Meilender, Alain Mille, Pascal Molli et al. "Knowledge acquisition and discovery for the textual case-based cooking system WIKITAAABLE." In 8th International Conference on Case-Based Reasoning-ICCBR 2009, Workshop Proceedings, pp. 249-258. 2009.
15. Paal Skalle, Agnar Aamodt, Isak Swahn. Detection of failures and interpretation of causes during drilling operations. ADIPEC, Abu Dhabi, Nov. 2016. Society of Petroleum Engineers, SPE-183 022-MS.