Bayesian-Supported Retrieval in BNCreek: A Knowledge-Intensive Case-Based Reasoning System

Hoda Nikpour, Agnar Aamodt, and Kerstin Bach

Norwegian University of Science and Technology, Department of Computer Science, Trondheim, Norway
https://www.ntnu.edu/idi

Abstract. This study presents a case-based reasoning (CBR) system that makes use of general domain knowledge - referred to as a knowledge-intensive CBR system. The system applies a Bayesian analysis aimed at increasing the accuracy of the similarity assessment. The idea is to employ the Bayesian posterior distribution for each case symptom to modify the case descriptions and the dependencies in the model. To evaluate the system, referred to as BNCreek, two experiment sets are set up from a "food" and an "oil well drilling" application domain. In both of the experiments, the BNCreek is evaluated against two corresponding systems named TrollCreek and myCBR with Normalized Discounted Cumulative Gain (NDCG) and interpolated average Precision-Recall as the evaluation measures. The obtained results reveal the capability of Bayesian analysis to increase the accuracy of the similarity assessment.

Keywords: CBR · Bayesian Analysis · Similarity assessment.

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 human experiences for complex problem-solving domains [1], and generates targeted explanations for the user as well as for its 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, beyond the cases themselves. 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 criteria, leading to a capability of explanation generation. A challenge with that method was the lack of a formal basis for the semantic network. It makes the inference processes within the network difficult to develop and less powerful than desired. The procedural semantic inherent in that type of network allows for a large degree of freedom in specifying the network semantics underlying the inferences that can be made, such as the value propagation and various forms of inheritance [3].

The disadvantage is that inference methods are implicit and hidden in the code, hence difficult to interpret and compare to other inference methods. In the past network a prototype-based semantic was implemented, as a way to handle uncertainly. The need for a clearly defined semantic and a more formal treatment of uncertainty led to some initial investigations into how a Bayesian Network (BN) model could be incorporated [4], [5]. A Bayesian framework includes an inference engine and builds probabilistic models without introducing unrealistic assumptions of independencies. It enables the conditioning over any of the variables and supports any direction of reasoning [7], [8], [6].

BNCreek, as a knowledge intensive system, provides a formal basis for the causal inference within the knowledge model, based on Bayesian probability theory.

2 Related work

Been et al. [9] integrated BN and CBR to model the underlying root causes and explanations to bridge the gap between the machine learning methods and human decision-making strategies. They used case-based classifiers and BN as two interpretable models to identify the most representative cases and important features. Bruland et al. [10] studied reasoning under uncertainty. They employed 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. [8] 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. [11] used a distributed CBR system to assist operators in feature solutions for faults by determining the cases shareing common symptoms. Aamodt et al. [4] 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. [5] investigated weaknesses of Bayesian networks in structural and parametric changes by adding case based reasoning functionality to the Bayesian network. Lacave [6] reviewed accomplished studies in Bayesian networks explanation and addressed the remaining challenges in this regard. Koton [12] presented a system called CASEY in which, 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. 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 [14]. The first step is a computationally cheap, syntactic matching process, and the second step is a knowledge-based inference process 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 represent relation sequences between unidentical features, are identified. Based on a specific 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 aims to improve the accuracy of the retrieval by taking advantage of both BN and CBR. The main idea behind BNCreek is to inject the Bayesian analysis into a domain ontology (semantic network) to assist the retrieve phase of a knowledge-intensive CBR system. BNCreek and TrollCreek conceptually work on the same ontology and the difference between them stems from the relational strengths, which in Trollcreek are static whereas in BNCreek change dynamically. In the present paper, we investigate the effects of Bayesian inference within the Creek architecture as a specific knowledge intensive CBR system. In section 3, the structure of BNCreek and its retrieve process are presented. Section 4 evaluates our approach by NDCG and interpolated average precision/recall measures. Section 5 discusses the obtained results and section 6 concludes the paper and names the future steps.

3 The BNCreek methodology

BNCreek is a knowledge-intensive system to address problems in uncertain domains. The knowledge representation in BNCreek is a combination of a semantic network, a Bayesian network, and a case-base, which together constitutes the knowledge model of the system as a three-module structure. The semantic module consists of the ontology nodes, which are connected by structural, causal, etc relations (e.g., "subclass-of", "part-of", etc.). This module enables the system to conduct semantic inference through various forms of inheritance. The Bayesian module is a sub-model of the semantic module and consists of the nodes that are connected by causal relations. That module enables the system to do the Bayesian inferences within the knowledge model in order to extract extra knowledge from the causes behind the observed symptoms and to utilize it for the case similarity assessment process. There is an individual module named Mirror Bayesian network, which interacts with the Bayesian module and is responsible for the Bayesian inference computational issues. The Mirror Bayesian network is created to keep the implementation complexity low and provides scalability for the system. The case base layer is connected to the upper layers through the case features (features are nodes of the Bayesian or the semantic networks) each possessing a relevance factor, which is a number that shows the importance of a feature for a stored case [2].

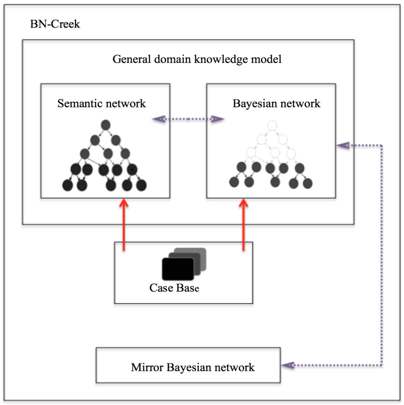


Fig. 1. The graphical representation of BNCreek.

Fig. 1 illustrates the graphical representation of the system structure. Each box illustrates one module of the BNCreek, and the inner boxes make up the outer ones, i.e., "semantic network" and "Bayesian network" modules form the "General domain knowledge model"; and the "General domain knowledge model", "Case Base" and "Mirror Bayesian network" form the BNCreek system. The solid arrows show the direction of connecting nodes inside and between modules and the dotted arrow indicates the information flow between the "semantic network", "Bayesian network" and the "Mirror Bayesian network".

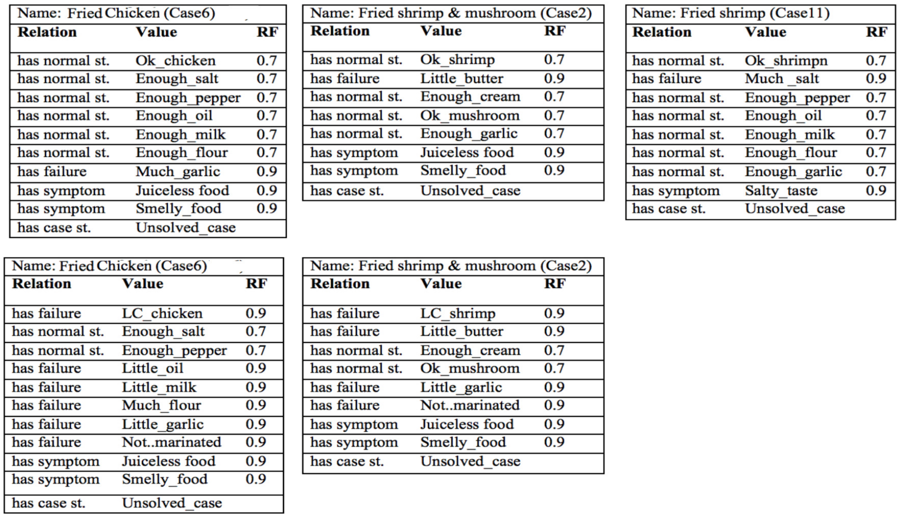
3.1 The retrieve process

The retrieve process in the current BNCreek system is the mature version of the process earlier outlined for BNCreek1 [13]. In the version presented here, the Bayesian analysis is integrated into the retrieve procedure and is an essential part of it, in contrast with BNCreek1 in which the semantic and Bayesian analysis were working in parallel and their results were combined.

The retrieve process in BNCreek is a master-slave process, which is triggered by observing a new raw case, i.e. knowledge about a concrete problem situation consisting of a set of feature names and features values [2]. In BNCreek the features are of two types:

1. Observed features that are entered into the case by the user (the raw case features).
2. Inferred features that are entered into the case by the system.

Each of the Observed and Inferred features could have three types, i.e., the symptom features (symptoms), the status features (status) and the failure features (failures). Below the retrieve process is presented utilizing a run-through example from the "food domain". The domain description and details can be found in the "System evaluation" section.



Name: Fried Chicken (Case6)

Relation	Value	RF
has normal st.	Ok_chicken	0.7
has normal st.	Enough_salt	0.7
has normal st.	Enough_pepper	0.7
has normal st.	Enough_oil	0.7
has normal st.	Enough_milk	0.7
has normal st.	Enough_flour	0.7
has failure	Much_garlic	0.9
has symptom	Juiceless food	0.9
has symptom	Smelly_food	0.9
has case st.	Unsolved_case	

Name: Fried shrimp & mushroom (Case2)

Relation	Value	RF
has normal st.	Ok_shrimp	0.7
has failure	Little_butter	0.9
has normal st.	Enough_cream	0.7
has normal st.	Ok_mushroom	0.7
has normal st.	Enough_garlic	0.7
has symptom	Juiceless food	0.9
has symptom	Smelly_food	0.9
has case st.	Unsolved_case	

Name: Fried shrimp (Case11)

Relation	Value	RF
has normal st.	Ok_shrimpn	0.7
has failure	Much _salt	0.9
has normal st.	Enough_pepper	0.7
has normal st.	Enough_oil	0.7
has normal st.	Enough_milk	0.7
has normal st.	Enough_flour	0.7
has normal st.	Enough_garlic	0.7
has symptom	Salty_taste	0.9
has case st.	Unsolved_case	

Name: Fried Chicken (Case6)

Relation	Value	RF
has failure	LC_chicken	0.9
has normal st.	Enough_salt	0.7
has normal st.	Enough_pepper	0.7
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 failure	Not..marinated	0.9
has symptom	Juiceless food	0.9
has symptom	Smelly_food	0.9
has case st.	Unsolved_case	

Name: Fried shrimp & mushroom (Case2)

Relation	Value	RF
has failure	LC_shrimp	0.9
has failure	Little_butter	0.9
has normal st.	Enough_cream	0.7
has normal st.	Ok_mushroom	0.7
has failure	Little_garlic	0.9
has failure	Not..marinated	0.9
has symptom	Juiceless food	0.9
has symptom	Smelly_food	0.9
has case st.	Unsolved_case	

Fig. 2. The upper cases are three sample raw cases from the food domain, and the two lower cases are the corresponding pre-processed cases descriptions. "st." and "LC" stand for status and long cooked, respectively.

As the run-through example, consider a "Fried chicken" dish with reported "Juiceless food" and "Smelly food" symptoms as a raw input case. The case is entered by a chef into BNCreek, to find the failures behind the symptoms. The raw case description consists of the dish ingredients like "Enough salt" as a status feature and the reported symptoms, illustrated on the upper left side of Fig. 2.

The master phase is based on inferencing in the Bayesian module. It has three steps.

The first step: The system utilizes the symptoms from the raw case description, i.e., "Juiceless food" and "Smelly food" and applies them to the Bayesian network module. The Bayesian inference results in the network posterior distribution (Algorithm 1, lines 1 and 2) utilizing the Eq. 1, in which " θ", "$p(\theta)$" and "$p(symptoms|\theta)$" stand for the parameter of distribution, prior distribution and the likelihood of the observations, respectively. The Bayesian module posterior distribution is dynamic in nature, i.e., the probabilities of the dependencies change as a new raw case is entered.

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

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

BNCreek considers the network posterior distribution and extracts the causes behind any of the case description symptoms. Due to the nature of the Bayesian networks, the parent nodes cause the children. So there would be several causes for any symptom that could be extracted. A threshold for the numbers of extracted causes will determine by the expert based on the knowledge model size. In the given example, the symptoms' causes are as follows: "Little oil" causes "Juiceless food"; "LC chicken" causes "Juiceless food"; "Little milk" causes "Juiceless food"; "Much flour" causes "Juiceless food" and "Not enough marinated" causes "Smelly food".

Algorithm 1: Retrieve in BNCreek

```
   Input  : An input raw case.
   Output: A sorted list of retrieved failure cases and graphical causal explanations
 1 Utilize the symptoms of the input raw case from its case description.
 2 In the Bayesian module compute the Bayesian layer posterior beliefs given the symptoms.
 3 Extract the cause of the applied symptoms.
 4 Modify the raw input case description by adding the extracted causes or adjusting the
   features.
 5 Pass the posterior beliefs from the Bayesian network module into the Semantic network
   module.
 6 Adjust the semantic network module causal strengths utilizing the posterior distribution
   from the Bayesian network module.
 7 while not all the case base is tested do
 8     Consider one case from case base.
 9     Compute the explanation strength between any pair of input and retrieved case features.
10     Compute the similarity between input and retrieved case.
11 end
12 List the matched cases.
13 Generate a graphical causal explanation for the input case.
```

Then the case description is modified based on the extracted causes and forms what is referred to as a pre-processed case description. The pre-processed case consists of Observed and Inferred features, e.g., "Enough salt" and "Not enough marinated", respectively. Which the "Not enough marinated" is added and the "Little oil" is adjusted from "Enough oil" in the modification process (see the bottom left side of Fig. 2 and Algorithm 1, lines 3 and 4).

The third step: The obtained posterior distribution from the Bayesian network module is passed to the semantic network module (Algorithm 1, line 5).

The slave phase is based on inferencing in the semantic network module. This phase has two steps.

The first step: The semantic network causal strengths are adjusted dynamically corresponding to the Bayesian posterior beliefs, in contrast to the other

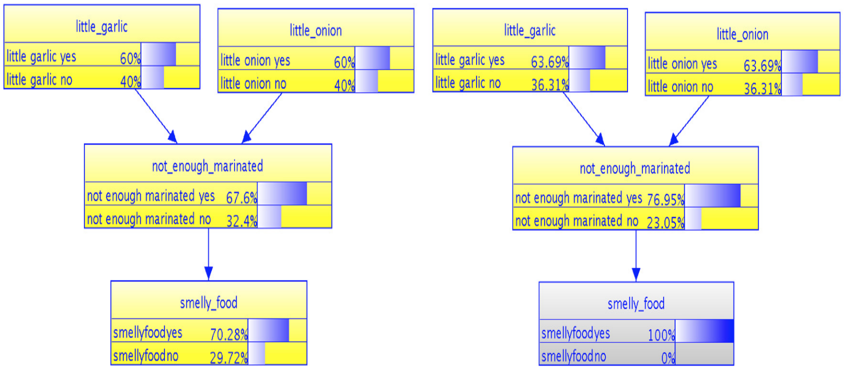


Fig. 3. Part of the Bayesian beliefs before (prior) and after (posterior) applying the symptoms into the network.

relations in the semantic network module, which are fixed (Algorithm 1, line 6). Fig. 3 illustrates part of the Bayesian network prior and posterior distribution, respectively. In which the posterior beliefs will be utilized to adjust the semantic network strengths.

The second step: This step utilizes the adjusted causal strengths and the preprocessed case description and computes the similarity between the input case and the case base.

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

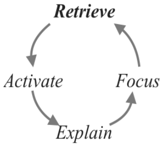


Fig. 4. The retrieve explanation cycle.

BNCreek considers each of the case base members at the time and utilizes Dijkstra's Algorithm [15] to extract all possible paths in the knowledge model that represent relation sequences between any features in the input case (f_i) and all the features in the retrieved case (f_j).

Consider case 2 as a retrieved case. Here, the partial similarity degree calculation between "LC shrimp", a feature from the retrieved case, and the "LC chicken", a feature from the input case, are presented. See Fig. 2 for cases descrip-

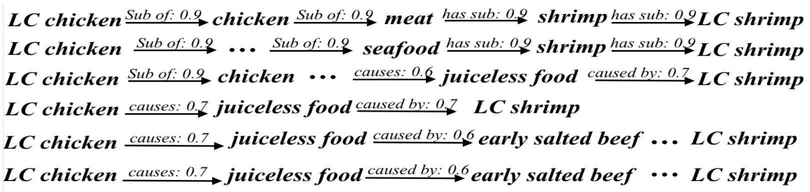

LC chicken Sub of: 0.9 chicken Sub of: 0.9 meat has sub: 0.9 shrimp has sub: 0.9 LC shrimp
LC chicken Sub of: 0.9 ... Sub of: 0.9 seafood has sub: 0.9 shrimp has sub: 0.9 LC shrimp
LC chicken Sub of: 0.9 chicken ... causes: 0.6 juiceless food caused by: 0.7 LC shrimp
LC chicken causes: 0.7 juiceless food caused by: 0.7 LC shrimp
LC chicken causes: 0.7 juiceless food caused by: 0.6 early salted beef ... LC shrimp
LC chicken causes: 0.7 juiceless food caused by: 0.6 early salted beef ... LC shrimp

Fig. 5. All possible Paths between "LC chicken" and "LC shrimp" features from the input and retrieved cases, respectively.

tions and Fig. 5 for the extracted paths between the two features. The various causal strengths reveal the effect of Bayesian analysis.

To explain the similarity strength between any coupled features, Eq. 2 is applied. In Eq. 2 the $EXPS(f_i, f_j)$ and RS stand for the explanation strength of $feature_i$ and $feature_j$ and the relation strengths, respectively. R and P demonstrate the number of relations and paths between f_i and f_j.

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

To compute the $EXPS(f_i, f_j)$, first, the strengths of each path relations is multiplied to achieve the path strength. Then the path strengths are multiplied to achieve the total $EXPS(f_i, f_j)$.

Consider "LC chicken" as f_i, "LC shrimp" as f_j and Fig. 5 for possible paths between them. The $1 - pathstrength$ for the first path in Fig. 5 is $1 - (0.9 * 0.9 * 0.9 * 0.9)$, which is 0.35 and for rest of the paths will be equal to 0.47, 0.71, 0.51, 0.71, 0.71. The multiplication of them is approximately 0.04. Finally the strengths between considered f_i and f_j is $1 - 0.4$, which is 0.96. For the situations where the paired features are the same (exact matched features), the explanation strength is considered as 1.

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

In Eq. 3, the (C_{IN}) and the (C_{RE}) stand for the input and the retrieved cases. The explanation strength of the $feature_i$ and $feature_j$ is demonstrated by $EXPS(f_i, f_j)$ and the relevance factor of the $feature_j$ is demonstrated by RF_{f_j}. The number of the features in the input and the retrieved cases are shown by 'm' and 'n', respectively.

The similarity between the input case (C_{IN}) and the retrieved case (C_{RE}) is computed by summing up the the explanation strength between f_i and f_j multiplied by the f_j relevance factor. The result is divided into the summation of the f_j relevance factor multiplied by $\beta(EXPS(f_i, f_j))$. The function of β is a binary function, which is equal to one when $EXPS(f_i, f_j)$ is not zero. See Eq. 3.

The calculation of the total similarity between case6 and case2 are presented here. For the numerator of Eq. 3, the explanation strength between any coupled features from the input and retrieved cases (e.g. "LC chicken" and "LC shrimp") is multiplied to the relevance factor of the retrieved case feature (i.e., "LC shrimp"), which is 0.96*0.9 and is equal to 0.86. Then the numerator is $1*0.9+1*0.9+1*0.9+0.9*0.89+0.9*0.59+0.9*0.85+0.9*0.47+0.9*0.56+0.9*0.96+0.9*0.67+0.9*0.73+0.7*0.84+0.7*0.65+0.7*0.89+0.7*0.89$, which is rounded to 10. In the denominator, for each feature from the input case, the relevance factors of the retrieved case will be multiplied by the binary function β and add together. β for any explained coupled feature is 1. For the directly matched couples, the relevance factors of the retrieved case will be summed up once. Due to the cases descriptions, cases 6 and 2 have 3 direct matched and 6 explained features. Then the denominator will be $(1*0.9+1*0.7+1*0.7+1*0.9)*6$ for the explained coupled features plus $1*0.9+1*0.9+1*0.7+1*0.9+1*0.7+1*0.9+1*0.9$ for the direct matched coupled features, which is 25. Finally, the total similarity between case6 and case2 is $10/25$ that is 0.4.

The system computes the similarity between the input case and all the cases from the case base (Algorithm 1 lines 7 to 11).

3.2 Explanations in BNCreek

There are two uses of explanations in the knowledge-based systems. One is the explanation that a system may produce for the user's benefit, e.g., to explain its reasoning steps or to justify why a particular conclusion was drawn. The other one is the internal explanation a system may construct for itself during problem-solving. BNCreek provides internal explanations for solving the problems, which are related to the "explanation strength" between two concepts in the model. A graphical causal explanation is generated to show the causal chains behind the observed symptoms for the benefit of the user.

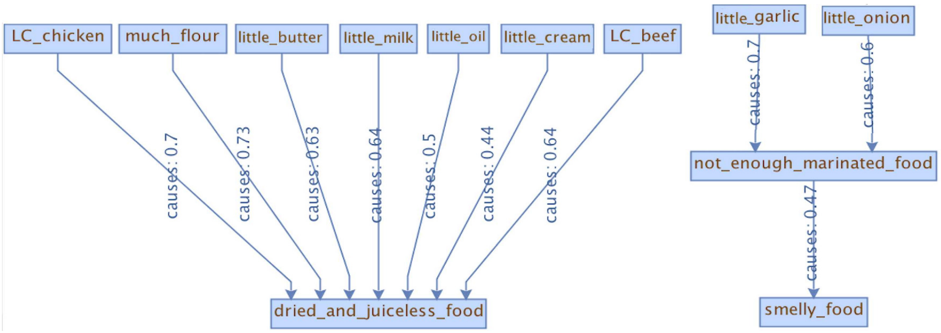


Fig. 6. An explanation structure from which a causal explanation in the food domain can be derived.

Fig. 6 demonstrates a graphical causal explanation structure for "Fried chicken (case 6)". The explanation is the result of Bayesian analysis given the two observations, i.e., "Juiceless food" and "Smelly food". BNCreek 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 "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 the most effect on causing the "Juiceless food". The right part of Fig. 6 shows two causal chains for "Smelly food", i.e., "Little garlic" causes "Not enough marinated" causes "Smelly food" and "Little onion" causes "Not enough marinated" causes "Smelly food" with causal strengths of 0.32 and 0.28, respectively (Algorithm 1 line 13).

The generated explanation in more uncertain domains like oil well drilling, plays a significant role in computing the similarity (by providing explanation paths) and clarifies the proposed solution for the expert.

4 System evaluation

To evaluate our approach, we set up two sets of experiments. One from the "food domain", a kind of initial toy-domain, and the other one from the "oil well drilling domain" the main investigated domain. Both of the experiments aim to measure the capability of the system to prevent the failures utilizing the observations. The application domains are tested by TrollCreek (version: 0.96devbuild), myCBR (Version: 3.1betaDFKIGmbH) and our implementation of BNCreek. The results from the systems are evaluated against the "ground-truth": domain expert predictions. The evaluation measures in this study are NDCG and Interpolated Average Precision-Recall.

Normalized Discounted Cumulative Gain (NDCG) as a ranked based information retrieval measure, values the highly relevant items appearing earlier in the result list. Usually, NDCG reports at rank cuts of 5, 10 or 20, i.e., nDCG@5, nDCG@10 or nDCG@20. The higher NDCG value shows the better performance of the system [16].

NDCG does not penalize for missing and not relevant cases in the results, so we utilized the Interpolated Average Precision-Recall measure to evaluate the relevance of the retrieved cases at 11 recall levels.

4.1 Food domain experiment

The main type of application domains for the presented system is complex and uncertain domains. Using our system for the smaller and more certain domains such as the utilized version of the "food domain" wouldn't be justifiable. However, Due to the simple nature of the "food domain", which leads to a better understanding of the system process, a run-through example from this domain is utilized and a set of evaluating experiments is set up.

Setup The food domain knowledge model is inspired by "Taaable ontology". Taaable is a CBR system that uses a recipe book as a case base to answer cooking queries[17]. Some modifications are made to fit the ontology to the BNCreek structure, i.e., adding causal relations. The causal relations present the failures of using an inappropriate amount of ingredients. Fifteen recipes are examined and simplified to their basic elements (e.g., Gouda cheese simplified to cheese). The resulted knowledge model consists of 130 food domain concepts and more than 250 relations between them. Eleven failure cases are created and utilized as the queries of the experiment. Each query applies to the case base in leave one out evaluation manner, which results in 11 query sets. The upper side of Fig. 2 demonstrates three samples of raw food failure cases descriptions.

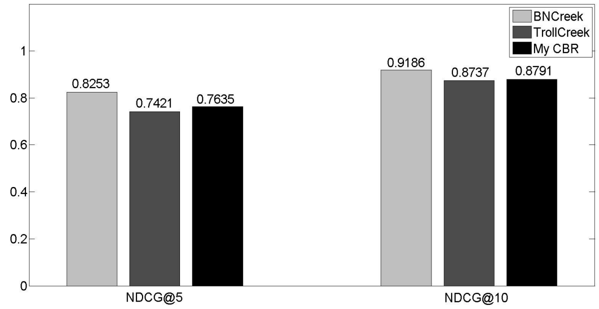


Fig. 7. NDCG values at cuts of 5 and 10 for food domain experiment

Food experiment results Utilizing the similarity scores of BNCreek, myCBR and TrollCreek, the NDCG values are computed against the expert predictions and reported at NDCG@5 and 10 in Fig. 7.

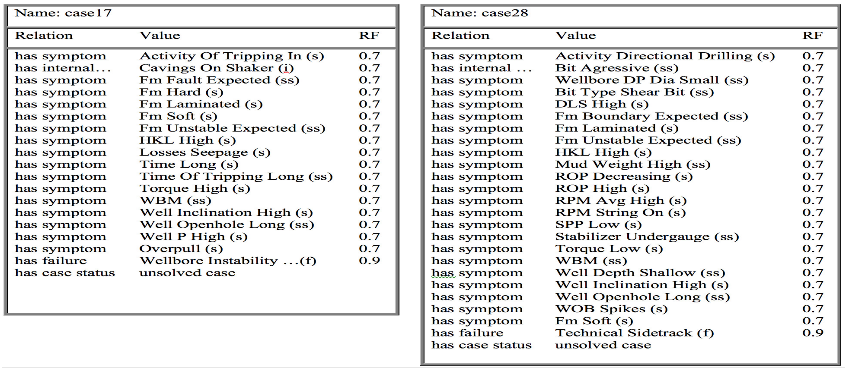
Fig. 7 illustrates that BNCreek with 0.8253 and 0.9186 values at NDCG@5 and 10 ranked the retrieved cases closer to the expert prediction in comparison to the TrollCreek and myCBR with 0.7421, 0.8737 and 0.7635, 0.8791 values.

The overall performance of the three systems is high and not so different, which can be explained by the fact that this experiment is set up on a small case base. Besides that, the BNCreek showed a somewhat better performance than the others in both cuts.

4.2 Oil well drilling domain experiment

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 BNCreek.

Setup In this experiment, we utilized an oil well drilling process knowledge model created by Prof Paal Skalle [18]. The knowledge model consists of 350



Name: case17

Relation	Value	RF
has symptom	Activity Of Tripping In (s)	0.7
has internal...	Cavings On Shaker (i)	0.7
has symptom	Fm Fault Expected (ss)	0.7
has symptom	Fm Hard (s)	0.7
has symptom	Fm Laminated (s)	0.7
has symptom	Fm Soft (s)	0.7
has symptom	Fm Unstable Expected (ss)	0.7
has symptom	HKL High (s)	0.7
has symptom	Losses Seepage (s)	0.7
has symptom	Time Long (s)	0.7
has symptom	Time Of Tripping Long (ss)	0.7
has symptom	Torque High (s)	0.7
has symptom	WBM (ss)	0.7
has symptom	Well Inclination High (s)	0.7
has symptom	Well Openhole Long (ss)	0.7
has symptom	Well P High (s)	0.7
has symptom	Overpull (s)	0.7
has failure	Wellbore Instability ...(f)	0.9
has case status	unsolved case	

Name: case28

Relation	Value	RF
has symptom	Activity Directional Drilling (s)	0.7
has internal ...	Bit Agressive (ss)	0.7
has symptom	Wellbore DP Dia Small (ss)	0.7
has symptom	Bit Type Shear Bit (ss)	0.7
has symptom	DLS High (s)	0.7
has symptom	Fm Boundary Expected (ss)	0.7
has symptom	Fm Laminated (s)	0.7
has symptom	Fm Unstable Expected (ss)	0.7
has symptom	HKL High (s)	0.7
has symptom	Mud Weight High (ss)	0.7
has symptom	ROP Decreasing (s)	0.7
has symptom	ROP High (s)	0.7
has symptom	RPM Avg High (s)	0.7
has symptom	RPM String On (s)	0.7
has symptom	SPP Low (s)	0.7
has symptom	Stabilizer Undergauge (ss)	0.7
has symptom	Torque Low (s)	0.7
has symptom	WBM (ss)	0.7
has symptom	Well Depth Shallow (ss)	0.7
has symptom	Well Inclination High (s)	0.7
has symptom	Well Openhole Long (ss)	0.7
has symptom	WOB Spikes (s)	0.7
has symptom	Fm Soft (s)	0.7
has failure	Technical Sidetrack (f)	0.9
has case status	unsolved case	

Fig. 8. Two samples of drilling cases description. RF stands for relevance factor.

drilling domain concepts and more than 1000 relationships between them, which makes it a very detailed ontology. Forty five drilling failure cases are utilized as the queries (input cases) in a leave-one-out evaluation to retrieve the matched similar ones among the rest of 44 failure cases. Each of the failure cases in average has 20 symptoms and one failure as the case solution that has been removed for the query case. Fig. 8 shows two examples of drilling cases.

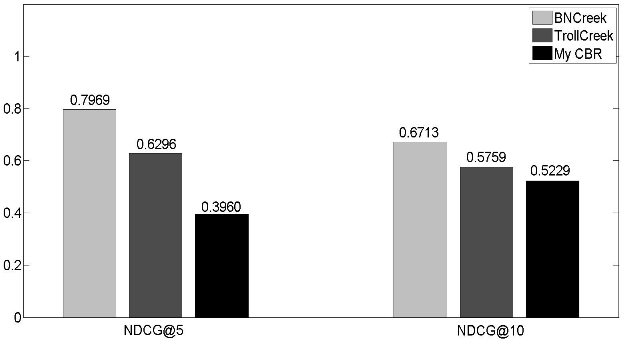


Fig. 9. NDCG values at cuts of 5 and 10

Oil well drilling experiment results We report on NDCG at ranks 5 and 10 in Fig. 9. The BNCreek NDCG@5 and 10 are reported as 0.7969 and 0.6713, which significantly outperform TrollCreek with 0.6296, 0.5759 and myCBR with 0.3960, 0.5229, respectively. The ordering produced by BNCreek yields a better NDCG value than the ordering produced by TrollCreek and myCBR at both cuts. This shows the efficiency of the Bayesian inference in case ranking in comparison with the other systems, which do not utilize the Bayesian inference.

Fig. 10 demonstrates the three systems interpolated average Precision at 11 Recall levels. In all recall levels, BNCreek has higher precision, which demonstrates the efficiency of the system in retrieving the more similar cases comparing to the other systems.

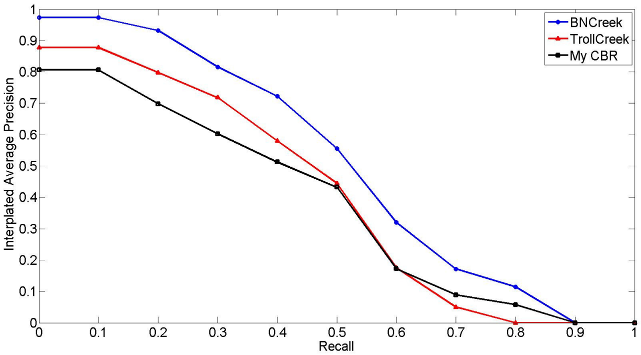


Fig. 10. Interpolated precision/recall graph for the results of BNCreek and TrollCreek

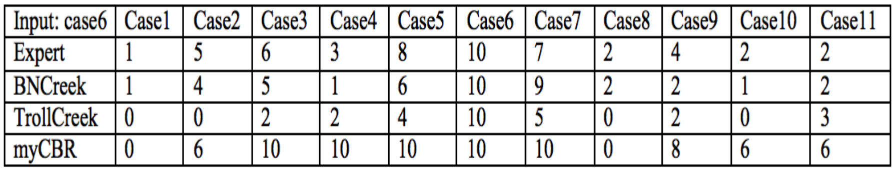
5 Discussion

The higher NDCG values in the two experiments, show the overall ability of BNCreek in ranking the retrieved cases correctly in comparison to TrollCreek and myCBR. The Interpolated Precision-Recall graph for the first experiment would be 1 for all recall levels because we retrieve all 10 cases of the case base. The Interpolated Precision-Recall graph for the second experiment illustrates the higher performance of BNCreek to retrieve the relevant cases in all 11 recall levels. Here we discuss the BNCreek ability to rank the cases in detail by an example.

According to the system goal (finding the failure behind the reported/observed symptoms), the most similar cases would be the cases carrying common symptoms. Then, the other features of the cases are irrelevant as long as they are not failures of the symptoms. Then there are two types of challenging cases. The first type has a similar overall case description but not the same symptoms, and the second one is the cases with the same symptoms but not similar case description, comparing to the input case. The first type should be categorized as not similar and the second type should be categorized as a very similar case.

Fig. 11 demonstrates the expert, BNCreek, TrollCreek and myCBR similarity assessment results for case 6 (the input case of the given run-through example) and the case base of the food application domain.

Case 11, has a similar ingredient with the case 6 and their differences originated in "Ok chicken" being replaced by "Ok shrimp", "Enough salt" being



Input: case6	Case1	Case2	Case3	Case4	Case5	Case6	Case7	Case8	Case9	Case10	Case11
Expert	1	5	6	3	8	10	7	2	4	2	2
BNCreek	1	4	5	1	6	10	9	2	2	1	2
TrollCreek	0	0	2	2	4	10	5	0	2	0	3
myCBR	0	6	10	10	10	10	10	0	8	6	6

Fig. 11. The expert, BNCreek, TrollCreek and myCBR similarity assessment results for case 6 and the case base of the food application domain.

replaced by "Much salt" and their symptoms, which are not the same. Case 11 is categorized as the third most similar case by TrollCreek while, based on the expert's prediction, it is almost the least similar case to case 6. This problem stems from similarity assessment mechanism in 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 categorizing of the cases such as case 11. BNCreek ranked case 11 as the sixth similar case, which is a better ranking. BNCreek in its master 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 into the similarity assessment.

Case 2 symptoms are the same with the input case. It is categorised as a not similar case by TrollCreek while, based on the expert's prediction, it is the fourth similar case. The problem with TrollCreek is originated in its similarity assessment method that uses the static relation strengths to compute the similarity which leads to a wrong categorizing the cases such as case 2. While BNCreek, utilizes a dynamically adjusted ontology relations strengths based on the BN posterior distribution given any input case.

myCBR ranked both of the case 2 and 11 as the sixth similar case. It ranks the retrieved cases based on their common features and a fixed similarity table which is determined by the expert. The similarity table could work like a pre computed similarity explanations between the not identical coupled features, if the expert knows all the coupled features similarity degrees, which rarely happens in uncertain domains with several features. This explains the decreased performance of myCBR from the small experiment of the food domain to the drilling domain experiment.

6 Conclusion and Future work

We studied the effect of Bayesian analysis in the similarity assessment within a knowledge-intensive system. We have developed a knowledge-intensive CBR system, BNCreek, which employs the Bayesian inference method to retrieve similar cases. The Bayesian analysis is incorporated to provide a formal and clear defined inference method for reasoning in the knowledge model.

To evaluate the effectiveness of our approach, we set up two experiments and employed the NDCG and Precision-Recall measures. Over two sets of experiments, we demonstrated that our approach has a better performance in ranking the retrieved cases against the expert prediction compared with the results of TrollCreek and myCBR. This indicates the Bayesian analysis efficiency for similarity assessment, across several application domains.

Although BNCreek showed a better performance in comparison with the other systems, in both of the examples it didn't manage to rank the cases same as the expert. Moreover, comparing the NDCG values in Fig. 7 and Fig. 9 shows the decreased performance of the BNCreek by increasing the case base size. A possible further step for this study is designing new metrics that help to rank the cases more accurately in bigger case bases.

7 Acknowledgement

The authors would like to thank Prof. Paal Skalle for preparing drilling cases and Prof. Helge Langseth and Dr. Frode Sørmo for their useful suggestions.

References

1. Gundersen, Odd Erik, Frode Sørmo, Agnar Aamodt, and Pål Skalle. "A real-time decision support system for high cost oil-well drilling operations." AI Magazine 34, no. 1 (2012): 21.
2. Aamodt, Agnar. "Knowledge-intensive case-based reasoning in creek." In European Conference on Case-Based Reasoning, pp. 1-15. Springer, Berlin, Heidelberg, 2004.
3. Sørmo, Frode. "Plausible Inheritance; semantic Network Inference for Case-Based Reasoning." Department of Computer and Information Science. Trondheim: Norwegian University of Science and Technology 102 (2000).
4. Aamodt, Agnar, and Helge Langseth. "Integrating Bayesian networks into knowledge-intensive CBR." In AAAI Workshop on Case-Based Reasoning Integrations, pp. 1-6. 1998.
5. Kofod-Petersen, Anders, Helge Langseth, and Agnar Aamodt. "Explanations in Bayesian networks using Provenance through Case-based Reasoning." In Workshop Proceedings, p. 79. 2010.
6. Lacave, Carmen, and Francisco J. Díez. "A review of explanation methods for Bayesian networks." The Knowledge Engineering Review 17, no. 2 (2002): 107-127.
7. Velasco, Fco Javier Molinero. "A Bayesian Network approach to diagnosing the root cause of failure from Trouble Tickets." Artificial Intelligence Research 1, no. 2 (2012): 75.
8. Houeland, T. G., T. Bruland, A. Aamodt, and H. Langseth. "A hybrid metareasoning architecture combining case-based reasoning and Bayesian networks (extended version)." IDI. NTNU 2011.
9. 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.
10. Bruland, Tore, Agnar Aamodt, and Helge Langseth. "Architectures integrating case-based reasoning and bayesian networks for clinical decision support." In International Conference on Intelligent Information Processing, pp. 82-91. Springer, Berlin, Heidelberg, 2010.

11. Tran, Ha Manh, and Jürgen Schönwälder. "Fault resolution in case-based reasoning." In Pacific Rim International Conference on Artificial Intelligence, pp. 417-429. Springer, Berlin, Heidelberg, 2008.
12. Koton, Phyllis A. "Using experience in learning and problem solving." PhD diss., Massachusetts Institute of Technology, 1988.
13. Nikpour, Hoda, Agnar Aamodt, and Pål Skalle. "Diagnosing root causes and generating graphical explanations by integrating temporal causal reasoning and CBR." CEUR Workshop Proceedings, 2017.
14. Forbus, Kenneth D., Dedre Gentner, and Keith Law. "MAC/FAC: A model of similarity-based retrieval." Cognitive science 19, no. 2 (1995): 141-205.
15. Dijkstra, Edsger W. "A note on two problems in connexion with graphs." Numerische mathematik 1, no. 1 (1959): 269-271.
16. Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst., 20(4):422?446, 2002.
17. Badra, Fadi, Julien Cojan, Amélie 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.
18. Pål 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.