Bayesian Analysis in a Knowledge-Intensive CBR System

Hoda Nikpour, Agnar Aamodt and Kerstin Bach

Norwegian University of Science and Technology, Department of Computer Science
Sem Sælandsvei 9 Gløshaugen, Trondheim, Norway
{Hoda.Nikpour, Agnar.Aamodt, Kerstin.bach}@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 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. The system is evaluated against a simplified version of a corresponding system named TrollCreek and the results of running two examples from two different application domains, i.e., a "food domain" and a "drilling process domain" are compared with a human expert prediction. 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 criteria, leading to a capability of explanation generation. A challenge with that method was 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].

1.1 Related works

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 strategies. 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 feature 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 networks 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 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 [12], in which the first step is a computationally cheap, syntactic matching pro-
cess, 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 aims to improve the accuracy of the retrieval by taking advantage of both BN and CBR. BN-Creek provides a formal basis for causal inference based on Bayesian probability theory. 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 relational 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 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 a semantic network, a Bayesian network and case-base which together create the knowledge model of the system with a three-layer structure. The semantic layer consists of the ontology nodes 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 by causal probability 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 and provides scalability for the system. 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 which is 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" and "Bayesian network" modules form the "General domain knowledge model"; and the "General domain knowledge model", "Case Base" 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

The retrieving process in BN-Creek has three phases, i.e., the raw input case pre-processing, 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.

For more clarification, a run-through example from a food domain is given in the following. 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 raw input case pre-processing phase is triggered by the entering of a raw case (knowledge about a concrete problem situation which consists of a set of feature names and features values [2]). A raw case description consists of assumed features like "enough salt" and symptoms like "smelly food". In the running example the chef enters the dish ingredients and the reported observations, rep-
Algorithm 1: Retrieve in BN-Creek

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Utilize the symptoms of the input case from its case description.</td>
</tr>
<tr>
<td>2</td>
<td>Compute the Bayesian layer posterior distribution given the symptoms.</td>
</tr>
<tr>
<td>3</td>
<td>Extract the causal chains that cause the case symptoms.</td>
</tr>
<tr>
<td>4</td>
<td>Modify the raw input case by adding the causal chain concepts to the case description.</td>
</tr>
<tr>
<td>5</td>
<td>Adjust the updated Bayesian beliefs to the knowledge model causal strengths.</td>
</tr>
<tr>
<td>6</td>
<td>While not all the case base is tested do</td>
</tr>
<tr>
<td>7</td>
<td>Consider one case from case base.</td>
</tr>
<tr>
<td>8</td>
<td>Compute the explanation strength between any pair of input and retrieved case features.</td>
</tr>
<tr>
<td>9</td>
<td>Compute the similarity between input and retrieved case.</td>
</tr>
<tr>
<td>10</td>
<td>End</td>
</tr>
<tr>
<td>11</td>
<td>List the matched cases.</td>
</tr>
<tr>
<td>12</td>
<td>Generate a graphical causal explanation for the input case.</td>
</tr>
</tbody>
</table>

The system utilizes 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).

\[ p(\theta|\text{symptoms}) \propto p(\text{symptoms}|\theta) \times p(\theta) \quad (1) \]

The posterior distribution \( p(\theta|\text{symptoms}) \) is obtained by Eq. 1. \( \theta, p(\theta) \) and \( p(\text{symptoms}|\theta) \) 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 symptoms. In the given example, the causal chains which lead to the observed symptoms are as follows: "little oil" causes "dried and juice-
less 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".

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

The case description is modified based on the extracted causal chain concepts and forms what is referred to as a pre-processed case description. The pre-processed case description consists of assumed features like "enough salt", inferred features like "little oil" and symptoms like "smelly food". So, "ok chicken 0.5" 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. For example, the belief of the concept "little onion" is changed from 60% to 60.68% of causal probability (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 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.

BN-Creek considers each of the case base members at the time and utilizes Dijkstra’s Algorithm [13] 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 "LC shrimp" as a feature of the case 7 (see the case description in Fig. 6) as a retrieved case and the feature "LC chicken" from case 6 (input case). The extracted paths between the two features are displayed in Fig. 5. The different causal strengths reveal the
effect of Bayesian analysis which, in contrast to the fixed strength the previous TrollCreek system, computes the posterior beliefs (causal strengths) based on the prior beliefs (from the expert) and the observed symptoms of the particular input case.

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. Consider "LC chicken" as \(f_i\), "LC shrimp" as \(f_j\) and Fig. 5 for possible paths between them. The \(1 - \text{pathstrength}\) for the first path in Fig. 5 is \(1 - (0.9 \times 0.9 \times 0.9 \times 0.9)\) which is 0.35 and for rest of the paths will be equal to 0.47, 0.71, 0.51, 0.71, 0.71 which 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.

\[
\text{explanation strength}(f_i, f_j) = 1 - \prod_{p=1}^{P} (1 - \prod_{r=1}^{R} \text{relation strength}_r) \\
\text{sim}(C_{IN}, C_{RE}) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \text{explanation strength}(f_i, f_j) \times \text{relevance factor}_{f_j}}{\sum_{i=1}^{n} \sum_{j=1}^{m} \beta(\text{explanation strength}(f_i, f_j)) \times \text{relevance factor}_{f_j}}
\]

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 \(\beta\) (explanation strength\((f_i, f_j)\)). The function named \(\beta\) (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 features in input and retrieved cases are shown by ‘\(m\)’ and ‘\(n\)’. See Eq. 3.
Here we compute the total similarity between case6 and case7. For computing the numerator of Eq. 3, the explanation strength between any coupled features from input and retrieved cases (e.g. "LC chicken" and "LC shrimp") multiplies 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 $1 \times 0.5 + 0.96 \times 0.9 + 1 \times 0.5 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9$ which is 7.26. See the relevance factors of case7 in Fig. 6. In the denominator, for each feature from the input case the relevance factors of the retrieved case will be multiplied to the binary function of $\beta$ and add together. $\beta$ for explanation strength of "LC chicken" and "LC shrimp" is 1 and the relevance factor of "LC shrimp" is 0.9. Then the denominator for all coupled features is $1 \times 0.5 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9 + 1 \times 0.9$ which is 8.2. Finally, the total similarity between case6 and case7 is 7.26/8.2 which is 0.88.

The system computes the similarity between the input case and all the cases from the case base (Algorithm 1 line 6 to 10).

Fig. 6. The six food cases description.

### 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 related to the "explanation strength" between two concepts in the model. A graphical causal explanation is generated to show the extracted causal chains behind the observed symptoms for the benefit of the user.

Fig. 7 demonstrates a graphical causal explanation structure 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. 7 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. 7 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, play a significant role in computing the similarity (by providing explanation paths) and clarifies the proposed solution for the expert.

3 Experiments and results

To evaluate BN-Creek, we set up two different experiments from "food failure domain" and "drilling process domain" that both aim at detecting the failures based on observations. In addition to the general system evaluation which is investigated in both experiments, the ability of the system to determine the cases similarity based on their failures and detect the similar cases with few common features is discussed. The BN-Creek results for both experiments are compared against a simplified version of TrollCreek and evaluated by domain expert’s prediction.
3.1 The food failure domain

The main type of application domains for the presented system is uncertain domains. Using this system for more certain domains such as "food domain" in some cases doesn’t make sense. However, on account of the simple nature of "food 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 made to fit the ontology to the BN-Creek 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) and eleven failure cases are created.

![Fig. 8. The six drilling cases description.](image)

**Expert similarity prediction methodology** Due to the simplicity of the domain, the expert’s result prediction strategy is presented. The similarities are determined by weighting the ingredients and the symptoms for any stored case in compare with the input case. The weighting strategy is to give number 4 to each common symptoms, number 3 to each common inferred features, number 2 to any inferred feature that is not the same but is in the same category with the input case feature, number 2 to each common assumed feature. Then the similarity of the retrieved case is calculated by summing the weights up and computing the case score out of its best possible score. Here, the inferred features are derived by the expert’s measures. Consider the "chicken fried steak and cream gravy (case 6)" as an input case. Cases 7 has two symptoms i.e. "dried and juiceless" and "smelly food" in common with case 6 then score 8 is assigned for the symptoms. It has "few oil", "few milk", "few garlic" and "much flour" as inferred features in common with case 6, so the score 12 is assigned for the inferred features. Case 7 has "enough salt" and "enough pepper" as assumed features in common with case 6 then score 4 is assigned for them. Finally, it has "LC shrimp" which is in the same category with "LC chicken" and gets score 2. Totally, case 7 gets score 26 out of 29 (if it had all features in common with case 6) which is
89% of similarity. The expert’s similarity prediction for any coupled cases has followed the same strategy.

**System evaluation** To evaluate the system similarity assessment in compare with the TrollCreek and the expert prediction, the system is run for all the cases in leave one out manner. Fig. 8 illustrates the results of BN-Creek, simplified TrollCreek and the expert, in its left, middle and the right sides, respectively. To investigate the correlation between the retrieved cases ranks, the Kendall’s Tau measure is employed. Fig. 9 demonstrates 0.610 correlation with a P-value of <0.01 between expert and BNCreek results for 74 sample cases. To compare the similarity degrees between the 3 systems, the average similarity number for each case is considered. Fig. 10 compares the results of average similarity between cases.

<table>
<thead>
<tr>
<th>Correlations</th>
<th>ExpertFood</th>
<th>BN-Creek</th>
<th>TrollCreek</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kendall’s tau_b</td>
<td>Correlation Coefficient</td>
<td>1.000</td>
<td>.610**</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>N</td>
<td>74</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>BN-Creek</td>
<td>Correlation Coefficient</td>
<td>.610**</td>
<td>1.000</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>N</td>
<td>74</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>TrollCreek</td>
<td>Correlation Coefficient</td>
<td>.569**</td>
<td>.548**</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>N</td>
<td>74</td>
<td>74</td>
<td>74</td>
</tr>
</tbody>
</table>

**. Correlation is significant at the 0.01 level (2-tailed).

![Fig. 9. Kendall’s Tau correlation](image1)

![Fig. 10. similarity degree comparison](image2)
3.2 The drilling process domain

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. 11 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 1.

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 1.

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 1.

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 1.

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.

Fig. 11. The six drilling cases description.
Table 1. 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.

<table>
<thead>
<tr>
<th>BN-Creek</th>
<th>TrollCreek</th>
<th>Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Matched</td>
<td>Input</td>
</tr>
<tr>
<td>case6</td>
<td>79%</td>
<td>case6</td>
</tr>
<tr>
<td>case6</td>
<td>78%</td>
<td>case4</td>
</tr>
<tr>
<td>case6</td>
<td>68%</td>
<td>case7</td>
</tr>
<tr>
<td>case6</td>
<td>64%</td>
<td>case5</td>
</tr>
<tr>
<td>case6</td>
<td>63%</td>
<td>case1</td>
</tr>
<tr>
<td>case6</td>
<td>61%</td>
<td>case2</td>
</tr>
</tbody>
</table>

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 with case 6 as the input case 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 ranked as a medium similar case by the simplified TrollCreek while, based on the expert's prediction, it is a weak similar case to case 6. This problem stems from the similarity assessment mechanism in the 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 are 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 the "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 for its feature relevance factors 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 relation strengths based on the BN posterior distribution dynamically which leads to capturing the similarity between instances like Cases 6 and 3 correctly. Fig. 12 demonstrates the differences between the matching degrees of three sampled features from Cases 6 and 3.

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
Fig. 12. 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.

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, based 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 steps for this research would be in line of the main goal of the study, i.e. retrieving the more similar cases. In this regard, Machine learning methods may employ to learn the network Bayesian parameters to decrease the prior distribution error. Different similarity assessment methods may investigate and apply to improve the accuracy of the similarity assessment.

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.