Bayesian Feature Construction for Case-Based Reasoning: Generating Good Checklists

Eirik Lund Flogard[1,2], Ole Jakob Mengshoel[1], and Kerstin Bach[1]

[1] Norwegian University of Science and Technology (NTNU),
Sem Sælands Vei 9, Trondheim, Norway
{eirik.l.flogard, ole.j.mengshoel, kerstin.bach}@ntnu.no
[2] Norwegian Labour Inspection Authority, Prinsensgt. 1, Trondheim, Norway
eirik.flogard@arbeidstilsynet.no

Abstract. Checklists are used to aid the fulfillment of safety critical activities in a variety of different applications, such as aviation, health care or labour inspections. However, optimizing a checklist for a specific purpose can be challenging. Checklists also need to be trustworthy and user friendly to promote user compliance. With labour inspections as a starting point, we introduce the Checklist Construction Problem. To address the problem, we seek to optimize the content of labour inspection checklists in order to improve the working conditions in every organisation targeted for inspections. To do so, we introduce a hybrid framework called BCBR to construct trustworthy checklists. BCBR is based on case-based reasoning (CBR) and Bayesian inference (BI) and constructs new checklists based on past cases. A key novelty of BCBR is the use of BI for constructing new features in past cases. The augmented past cases are retrieved via CBR to construct new checklists, which ensures justification for the content of the checklists and promotes trust. Experiments suggest that BCBR is more effective than any other baseline we tested, in terms of constructing trustworthy checklists.

Keywords: Bayesian CBR · Feature construction · Checklist.

1 Introduction

Context. Every year more than three million workers are victims of serious accidents causing more then 4000 deaths due to poor working conditions in EU alone.[3] Worldwide, it has been estimated that there are at least 9.8 million people in forced labour (2005) [2]. The most important measure to prevent poor working conditions is regulations. Regulations are usually enforced through labour

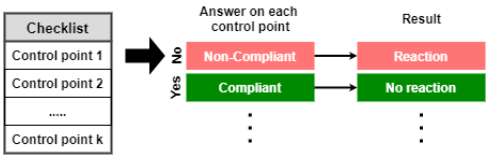

Fig. 1. Conceptual view of NLIA's procedure

[3] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52014DC0332

inspections, which make them a vital part of the strategy employed by many countries to ensure good health, safety, decent work conditions and well-being for workers (see UN's SDGs 3, 8 and 16[4]). Hence it is important to carry out labour inspections efficiently at large scale.

To identify poor working conditions, labour inspection agencies use surveys to check individual organisations for non-compliance [24]. Such procedures vary between different countries and we will use the Norwegian Labour Inspection Authority (NLIA) as an example. NLIA's inspection procedure is shown in Figure 1. It consists of a checklist which is a set of control points that are answered during the inspection. Every control point is a question that corresponds to a specific regulation. The answer to each question indicates whether the inspected organisation is compliant or not. These answers provide a basis for reactions if non-compliance is found. Checklists for ensuring health and safety are also used in other domains such a surgery or flight procedures to ensure high accuracy of due diligence, and success often relies on correctly applying checklists [5].

Challenges with Checklists. Currently, labour inspection agencies operate with a limited, fixed number of static procedures or checklists targeting specific industries that organisations belong to. The inspectors select the checklist they subjectively believe is most relevant to the organisation they are visiting. A drawback with this approach is that the selected checklist can be poorly optimized for its target, while also being limited in terms of scope. This may prevent the inspections from fulfilling their purpose of addressing high risks to the workers' health, environment and safety. Checklists used for other applications such as aviation and health care may have similar problems where poorly optimized checklists can suffer from compatibility issues with users or contexts [5, 7]. This can have a negative effect on the users' motivation to use the checklists.

Contributions. We introduce the Checklist Construction Problem (CCP): Suppose that we have N unique questions with yes/no answers, where the answer to each question has an unknown probability distribution. Given the questions, construct a checklist for a target entity by selecting K unique questions that maximize the likelihood for obtaining no-answers to every selected question.

This problem could be applied to any domain where checklist optimization is an issue, such as healthcare or aviation. In these domains, the N unique questions may be designed to accomplish a specific task such as surgery or flight check and the target entity may be a patient or an aircraft. Any question with a likely no-answer should then be on the surgery or flight checklist so that yes-answers are obtained instead. However, this work focuses on solving CCP for labour inspections and introduces a new data set as a starting point to do so.

To solve CCP, we introduce BCBR, which is a framework based on Bayesian inference (BI) and case-based reasoning (CBR) for constructing new checklists optimized for a target organisation (entity). BCBR uses CBR to retrieve questions from checklists which have been used in past cases to survey organisations similar to the target organisation. BI is used to construct features in past cases which ensures that the retrieved questions have high probabilities for non-

[4] https://sdgs.un.org/

compliance. The approach starts with a data set of cases containing organisations and questions from previously used checklists. New features are then constructed by means of BI and added to each row in the data set to create augmented cases. The augmented cases are added to a case base which is queried using similarity based retrieval. The query contains a target probability and organisation, which is used to retrieve cases containing the questions for a new checklist (solution).

From a technical perspective, the use of augmented cases is a key novelty of BCBR that can be viewed as a data-driven approach that uses feature construction to embed solution knowledge in cases for case retrieval in CBR [8, 15, 18]. The use of BI to estimate probability ensures transparency because the estimates are made by counting cases in the data set. The use of similarity based retrieval also promotes trustworthiness and ensures justification of the BI estimates because they are related to past cases. Trustworthiness is important to ensure user compliance with the checklists. The core contributions of this paper are:

- We introduce a formal definition of the Checklist Construction Problem and a new data set of previously used questions (control points) collected from NLIA's labour inspections between 2012 and 2019.
- We present the details for BCBR, which is designed for constructing checklists based on CBR and Bayesian inference.
- We establish an approach for evaluating the checklists constructed by BCBR. The framework is then empirically compared to baselines. The results show that BCBR constructs more efficient checklists than the baselines.

2 Related Work

Hybrid Frameworks Based on CBR and BI. There are multiple examples of frameworks with combinations of CBR and BI to address uncertainty for applications where some prior belief or information is available. Such frameworks also provide explanations, where CBR has been used to achieve explanation goals [22] (such as transparency and justification) or generate explanations [19]. Nikpour et al. [18] use Bayesian posterior distributions to modify or add features to input case descriptions to increase accuracy of similarity assessments in case retrieval. They also use the same approach to provide explanations for case failures in different domains [17]. This approach is similar to BCBR, but BCBR constructs new features which are also added to the case base-cases rather than modifying input cases. Kenny et al. [12] also use a combination of BI and CBR to exclude outlier cases from case retrieval and to provide explanations by examples. The purpose of the framework is to predict grass growth for sustainable dairy farming. Gogineni et al. [9] combines CBR and BI to retrieve and down-select explanatory cases for underwater mine clearance.

Similarity Based Retrieval for Trustworthiness. Lee et al [13] replaced the output layer of a neural network with k-nearest neighbour (kNN) to generate voted predictions and find the nearest neighbour cases to explain the predictions. This also guarantees that every prediction can be justified by a relevant past explanatory case. The justification via explanatory cases increases the reliability

of the neural network predictions and promotes trustworthiness. BCBR is also based on the same principle where BI predictions are justified by being embedded in past cases as features.

Trustworthy Case-Based Recommender Systems. BCBR aims to select a subset of all possible questions for a new checklist. Similarily, in recommender systems, a user is recommended a subset of items from the space of all possible items. Such systems can be divided into two classes: collaborative and case-based (content or user-based) recommender systems [3], where the latter approach could relate to our work. The case-based approach has been used to predict running-paces for different stages in ultra races, based on cases from similar runners in past cases [16]. CBR has also been used to provide explanatory cases for black-box recommender systems to achieve justification [4, 10]. Explanations for such systems can also be created through relations between features (concepts) [11]. However, the quality of explanations for black-box systems in terms of transparency, interpretability and trustworthiness can still be questionable [20]. Some authors also suggest to avoid explainable black box models in cases where they are not needed [21] and to use transparent, interpretable models for high-stakes decision making [20].

3 Case and Problem Definition

In this section we introduce the formal case and problem definition used for the rest of the paper.

Data Set and Cases. A data set $\mathcal{D}$ for variables $\mathbf{Z}$ is a finite length tuple where a case $\mathbf{d}_j \in \mathcal{D}$ is an instantiation of $\mathbf{Z}$ [6]. A case is a tuple $\mathbf{d} = (e, \mathbf{x}, l)$ where e denotes a question from a checklist, $\mathbf{x}$ is an entity and $l \in \{0, 1\}$ denotes the answer of the question. A case in the data set is a past experience where a question e has been applied to $\mathbf{x}$ to obtain the answer l. A case description is shown in Table 1.

Table 1. Description of a case in the data set

Name	*Description*	*Type*
x_{isc}	Industry subgroup code	Ordinal
x_{igc}	Industry group code	Ordinal
x_{ic}	Industry code	Ordinal
x_{iac}	Industry area code	Ordinal
x_{imac}	Industry main area code	Nominal
x_{mnr}	Municipality number	Ordinal
x_{fyl}	Fylke (county)	Nominal
e	Question	Nominal
l	Non-compliance	Binary

Entity. Every case d in the data set contains an entity description in the form of an organisation $\mathbf{x}$, defined by its location and industry. The features are organised according to Figure 2. An organisation can be implicitly defined as $\mathbf{x} = (x_{mnr}, x_{isc})$, since the other features of $\mathbf{x}$ are located higher in the hierarchies.

Question. Each case in the data set contains a question (control point) e with a yes/no an-

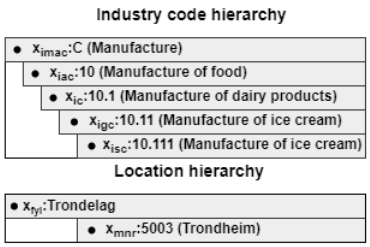

Fig. 2. Industry and location hierarchies of an organisation

swer. The question is used to survey the entity
$\mathbf{x}$ in the case. A specific question can appear in multiple checklists.
Checklist. A checklist $\mathbf{y}$ is defined as a set of yes/no questions constituted by cases in the data set, so that $\mathbf{y} = (e_1 \in \mathbf{d}_1, e_2 \in \mathbf{d}_2 ... e_{nd} \in \mathbf{d}_{nd})$. A question can only appear once per checklist such that $e_i \neq e_j$ for every $e_i \wedge e_j \in \mathbf{y}$.
Answer. The label l of a case is the observed answer from applying the question e to the entity $\mathbf{x}$. The answer $l = 1$ means that non-compliance has been found, while $l = 0$ means that $\mathbf{x}$ is compliant.
The Checklist Construction Problem. The problem is shown on Figure 3. Let there be a set of N unique questions and a new target entity $\mathbf{x}^{cnd}$. Each question has an unobserved answer l about $\mathbf{x}^{cnd}$ that belongs to an unknown distribution. Given the N questions, a model $\mathcal{M}$ first needs to correctly estimate the probability for observing $l = 1$ for each question. $\mathcal{M}$ then needs to select K unique questions $(e_1, e_2, ..., e_K)$, with the highest estimated probability, for a candidate checklist $\mathbf{y}^{cnd}$. The goal is to observe as many $l = 1$ answers as possible when applying $\mathbf{y}^{cnd}$ to $\mathbf{x}^{cnd}$.

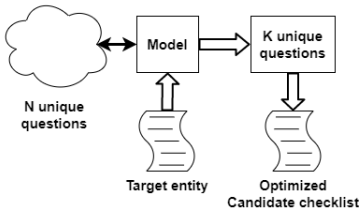

Fig. 3. An overview of CCP.

4 BCBR Framework

An overview of the BCBR framework is shown in Figure 4. The motivation for the framework is to solve the CCP problem while also ensuring that every question $e_i \in \mathbf{y}^{cnd}$ can be justified by a relevant past experience (see Section 5.3). The framework can be described by the following three steps: (1) A naive Bayesian inference method is used to generate two probability estimates ($\theta^{be}_{x_{isc}}$ and $\theta^{be}_{x_{mnr}}$) for every case $\mathbf{d}_j \in \mathcal{D}$. The estimates are generated by counting the cases in the data set with the same question and entity description as $\mathbf{d}_j$. This is done because many of the cases in the data set contains identical questions and/or identical target entities. Using Bayesian inference also ensures transparency for the estimates. (2) A case base $\mathcal{CB}$ of augmented CBR cases $\mathbf{c}_j$ is created. Each case $\mathbf{c}_j \in \mathcal{CB}$ is created by adding both estimates as features to each $\mathbf{d}_j \in \mathcal{D}$. (3) A query $\mathbf{q}$ is defined, which contains a target entity $\mathbf{x}^{cnd}$ and target values for the probability estimates. The query is used to retrieve a selection of K cases from $\mathcal{CB}$. Each case contains a question e_i for the candidate checklist $\mathbf{y}^{cnd}$.

4.1 Bayesian Inference

We use empirical distributions of the data set $\mathcal{D}$ to estimate the probability for observing $l = 1$, to achieve transparency for the BCBR framework. When prior knowledge or belief about l is available, BI can be used instead of the standard maximum likelihood method. An advantage with BI is that it (to some extent) can be used to address inaccurate empirical estimates caused low or zero case counts ("Zero count problem") [6]. The problem may have a negative impact

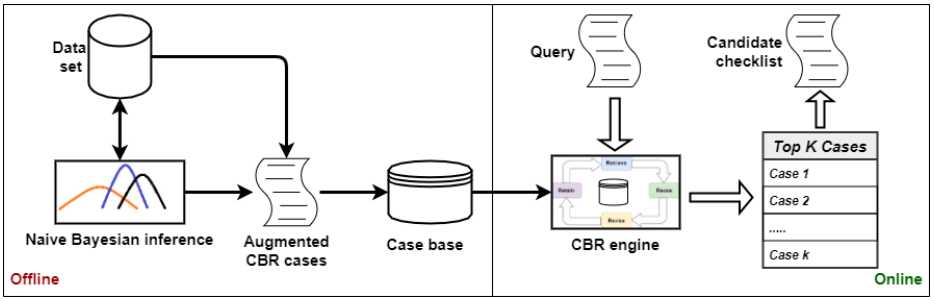

Fig. 4. An overview of the BCBR framework. The creation of augmented cases and the case base happens offline. The case base is used for the construction of checklists in the online-part.

on the quality of the K answers selected by BCBR. To further deal with this problem we use Naive Bayesian inference (NBI) which generates two probability estimates instead of just one. A derivation for this follows below.

Estimating the Empirical Probability for Non-compliance (l). By using the definitions from Section 3, the empirical distribution of the data set $\mathcal{D}$ can be defined as:

$$\theta_D(\alpha) = \frac{\mathcal{D}\#(\alpha)}{\mathcal{N}} \tag{1}$$

where $\mathcal{D}\#(\alpha)$ is the number of cases in the data set $\mathcal{D}$ which satisfy the event α and $\mathcal{N}$ is the number of cases in $\mathcal{D}$ [6]. We denote the event $L = 1$ as observing the outcome $l = 1$ and $L = 0$ for $l = 0$. From the expression above, the probability for $L = 1$ can then be calculated given $\mathbf{x}$ and e:

$$\theta_D(L = 1|\alpha) = \frac{\theta_D(L = 1 \wedge \alpha)}{\theta_D(\alpha)} = \frac{\mathcal{D}\#(L = 1 \wedge \mathbf{X} = \mathbf{x} \wedge E = e)}{\mathcal{D}\#(\mathbf{X} = \mathbf{x} \wedge E = e)} \tag{2}$$

where $\alpha = (\mathbf{X} = \mathbf{x}) \wedge (E = e)$. That is, the event where the entity description is given as $\mathbf{x}$ and the question is given as e.

Naive Bayesian Inference for Estimating Empirical Probability (l). The posterior probability for an event $L = 1|\alpha$ can be expressed as the mean of a Beta distribution [6]:

$$\theta^{be}(L = 1|\alpha) = \frac{\mathcal{D}\#(L = 1 \wedge \alpha) + \psi_{L=1|a}}{\mathcal{D}\#(L = 1 \wedge \alpha) + \psi_{L=1|a} + \mathcal{D}\#(L = 0 \wedge \alpha) + \psi_{L=0|\alpha}} \tag{3}$$

where ψ is a set of prior belief parameters and where $(\mathcal{D}\#(L = 1 \wedge \alpha) + \psi_{L=1|a})$ and $(\mathcal{D}\#(L = 0 \wedge \alpha) + \psi_{L=0|a})$ are the parameters for a Beta distribution.

From the components x_{isc} and x_{mnr} of $\mathbf{x}$, two NBI probability estimates $\theta^{be}_{x_{isc}}$ and $\theta^{be}_{x_{mnr}}$ can be obtained from Equation 3 by substituting α: $\theta^{be}_{x_{isc}} = \theta^{be}(L = 1|(X_{isc} = x_{isc} \wedge E = e))$ and $\theta^{be}_{x_{mnr}} = \theta^{be}(L = 1|(X_{mnr} = x_{mnr} \wedge E = e))$. Using two probability estimates instead of one is an effective measure against low case counts because $\mathcal{D}\#(X_{isc} = x_{isc} \wedge E = e) \geq \mathcal{D}\#(\mathbf{X} = \mathbf{x} \wedge E = e)$ and $\mathcal{D}\#(X_{mnr} = x_{mnr} \wedge E = e) \geq \mathcal{D}\#(\mathbf{X} = \mathbf{x} \wedge E = e)$. The approach is "naive" since it assumes that x_{mnr} and x_{isc} are independent given l and e.

4.2 Case Base Creation and CBR Engine

This section defines the details for the augmented CBR cases, case base and similarity based retrieval from Figure 4.

Augmented CBR Case and Case Base. Algorithm 1 shows the creation of a case base $\mathcal{CB}$ with augmented cases $\mathbf{c}$. The algorithm includes two additional features: $\kappa_{x_{mnr}}$ and $\kappa_{x_{isc}}$. The features are included to adjust for the case counts of the probability estimates when retrieving cases. The values for the θ^{be} and the κ-features are estimated from $\mathcal{D}$, given $x_{mnr,j}$, $x_{isc,j}$ and e_j from $\mathbf{d}_j \in \mathcal{D}$. The features are added to $\mathbf{d}_j$ to form a case $\mathbf{c}_j$ for $\mathcal{CB}$. An example showing the specific features of the augmented cases can be found in Section 4.3.

Algorithm 1 Creation of a case base $\mathcal{CB}$ with cases $\mathbf{c}_j$

Input: $\mathcal{D}$;
Output: $\mathcal{CB} \leftarrow ()$;
for each $\mathbf{d}_j \in \mathcal{D}$ **do**
  $//(x_{isc,j}, x_{mnr,j}, e_j) \in \mathbf{d}_j$
  $\theta^{be}_{x_{isc}} \leftarrow \theta^{be}(L = 1|(x_{isc,j}, e_j))$;
  $\theta^{be}_{x_{mnr}} \leftarrow \theta^{be}(L = 1|(x_{mnr,j}, e_j))$;
  $\kappa_{x_{mnr}} \leftarrow \mathcal{D}\#(L = 1 \wedge X_{mnr} = x_{mnr,j} \wedge E = e_j)$;
  $\kappa_{x_{isc}} \leftarrow \mathcal{D}\#(L = 1 \wedge X_{isc} = x_{isc,j} \wedge E = e_j)$;
  $\mathbf{c}_j \leftarrow Join(\mathbf{d}_j, \theta^{be}_{x_{mnr}}, \theta^{be}_{x_{isc}}, \kappa_{x_{mnr}}, \kappa_{x_{isc}})$;
  $\mathcal{CB} \leftarrow Join(\mathcal{CB}, \mathbf{c}_j)$;
end for
return $\mathcal{CB}$;

Case Retrieval and Similarity Function. To retrieve questions e_i for the candidate checklist $\mathbf{y}^{cnd}$, a query case $\mathbf{q}$ and similarity function is used. The query consists of the target entity $\mathbf{x}^{cnd}$ and the desired values for both the probability estimates and the case count features. A similarity function assigns a score $Sim(\cdot, \cdot) \in [0, 1]$ to every pair $(\mathbf{q}, \mathbf{c}_j \in \mathcal{CB})$. A set of unique e_i for $\mathbf{y}^{cnd}$ is then retrieved from the K cases with the highest similarity score. The similarity function is defined according to the equation below:

$$Sim(\mathbf{q}, \mathbf{c}_j) = \frac{1}{\sum w_i} \sum_i w_i \cdot sim_i(\mathbf{q}, \mathbf{c}_j). \tag{4}$$

Where w_i is a weight, sim_i is a local similarity function and i denotes a feature common to the query and the case. Each local similarity function in Equation (4), yields a score $[0, 1]$ for each feature (i) according to the similarity $sim_i(\mathbf{q}, \mathbf{c}_j)$ between the cases $\mathbf{q}$ and $\mathbf{c}_j$. The local similarity functions and the weights are defined by a domain expert for the purpose of this work (see Section 5.1).

4.3 Example: NBI Estimates, Case Retrieval and CBR Case

NBI Estimates. Let $x_{isc} = 22.230$, $x_{mnr} = 1507$ be features of an entity description $\mathbf{x}$ and $e =$"Did the employer make sure to equip all employees who carry out work at the construction site with a HSE card?" be a question of a case $\mathbf{d} \in \mathcal{D}$. The prior parameters are $\psi_{L=1|\alpha} = 1$ and $\psi_{L=0|\alpha} = 5$ because $l = 1$ is observed in approximately 1 of 6 cases. Given this information, $\theta^{be}_{x_{isc}}$ is estimated by counting cases $\mathbf{d}$ in data set $\mathcal{D}$ which satisfy $X_{isc} = x_{isc}$ and $E = e$. Applying $\alpha = (X_{isc} = x_{isc} \wedge E = e)$ to Equation 3 yields: $\theta^{be}_{x_{isc}} = \frac{1+1}{1+2+6} \approx 22\%$.

This estimate is more accurate than the empirical probability estimate, which is $\theta_{x_{isc}} = \frac{1}{1+2} \approx 33\%$ (Eq. 2). The difference can be explained by low case count, which affect the quality of both the Bayesian and empirical estimates.

The same procedure is used to calculate: $\theta^{be}_{x_{mnr}} = \frac{89+1}{89+186+6} \approx 32\%$. In this case the Bayesian estimate is approximately the same as the empirical probability estimate, since the case count is high. The estimates are used to create an augmented CBR case $\mathbf{c}$.

Table 2. Description of case features, similarity weights, query and retrieved case for the example.

Feature	**w**	***Query 1***	***Case 1***	***Query 2***	***Case 2***
x_{isc}	1	22.230	22.230	22.230	22.230
x_{igc}	2	22.23	22.23	22.23	22.23
x_{ic}	2	22.2	22.2	22.2	22.2
x_{iac}	2	22	22	22	22
x_{imac}	2	C	C	C	C
x_{mnr}	2	1507	1507	1507	1507
x_{fyl}	2	MoM	MoM	MoM	MoM
l	0	-	0	-	0
e	0	-	e_1	-	e_2
$\theta^{be}_{x_{isc}}$	9	100%	22%	-	7%
$\theta^{be}_{x_{mnr}}$	4	100%	32%	-	7%
$\kappa_{x_{isc}}$	1	70	1	-	0
$\kappa_{x_{mnr}}$	1	70	89	-	30
Sim		-	0.546	-	0.448

Case Retrieval and Augmented CBR Case. For this example we assume that a case base of CBR cases has been created and that $K = 1$, for the sake of brevity. The case retrieval starts by defining a query case (Query 1), shown in Table 2. $\theta^{be}_{x_{isc}}$ and $\theta^{be}_{x_{mnr}}$ are set to 100%, which is the target value for the retrieved cases. Both $\kappa_{x_{isc}}$ and $\kappa_{x_{mnr}}$ are set to 70 so that case counts of 70 or higher yield full similarity scores, according to Figure 5.

After applying the similarity function to every pair ($\mathbf{q}, \mathbf{c} \in \mathcal{CB}$), the top $K = 1$ case with highest similarity (Case 1) is retrieved for the candidate checklist $\mathbf{y}^{cnd}$.

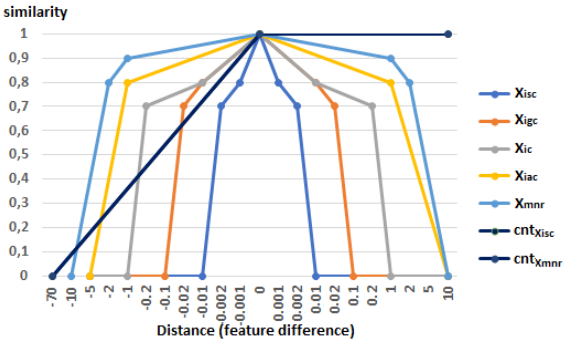

Fig. 5. Local similarity functions.

For comparison, we also define Query 2 in Table 2 where $\theta^{be}_{x_{isc}}$, $\theta^{be}_{x_{mnr}}$, $\kappa_{x_{isc}}$ and $\kappa_{x_{mnr}}$ are undefined. The $K = 1$ case returned from $\mathcal{CB}$ is Case 2. Case 2 fully matches Query 2 in terms of $\mathbf{x}$, but $\theta^{be}_{x_{isc}}$ and $\theta^{be}_{x_{mnr}}$ suggest that it is unlikely to observe $l = 1$ when e_2 is applied to $\mathbf{x}$. This is expected because we removed the part of the query that maximizes the probability for observing $l = 1$.

5 Experiments

In this section three experiments are presented. In the first experiment a simple label classification problem is introduced to establish a starting point for

comparing ML methods as baselines for the labour inspection CCP. The second experiment aims to measure the justification of checklists constructed by BCBR and the two best-performing baselines from the first experiment. The third experiment aims to measure the performance of BCBR against the baselines from the second experiment.

5.1 Experimental Setup

Measure of Justification. We introduce Equation 5 to measure the justification ($J \in [0, 100\%]$) of a checklist $\mathbf{y}$ for a given entity $\mathbf{x}$, according to the proportion of questions $e_i \in \mathbf{y}$ which also exist in past cases $(e_i, \mathbf{x}, \cdot) \in \mathcal{D}$.

$$J(\mathbf{y}, \mathbf{x}, \mathcal{D}) = \frac{|\{e_i \in \mathbf{y} : (e_i, \mathbf{x}, \cdot) \in \mathcal{D}\}|}{|\{e_i \in \mathbf{y}\}|} \tag{5}$$

The expression can be seen as an adaptation of Massie alignment score [14] that measures the percentage of questions $e_i \in \mathbf{y}$ with full alignment to the nearest neighbour case in $\mathcal{D}$.

BCBR Configuration. For the experiments, BCBR uses the same configuration as in Section 4.3. The only difference is that $K = 15$ is used instead of $K = 1$, so that the constructed checklists consist of 15 questions.

The weights and local similarity functions are set based on domain knowledge and are shown in Table 2 and Figure 5 respectively. The weights are set according to the importance of each feature, while the similarity functions are defined to model the similarity according to the hierarchical relationship between the ordinal features of the entity $\mathbf{x}$ (see Section 3). For the other features not shown in Figure 5, the default option in the myCBR tool is used to define the local similarity functions.

Baselines for the Experiments. The baseline methods used for the experiments are: CBR (CBR-BL), Logistic Regression(LR), Decision tree (DT) and Naive Bayes classifier (NBC), Conditional probability estimates (CP), Bayesian inference (BI), Naive conditional probability (NCP) and NBI.

CBR-BL generates predictions from the label of the closest neighbour case in the training data. CP generates predictions for any pair $(e, \mathbf{x})$ according to Equation 2. BI uses Equation 3 with $\psi_{L=1|\alpha} = 1$, $\psi_{L=0|\alpha} = 5$ and $\alpha = (\mathbf{X} = \mathbf{x} \wedge E = e)$. NCP is based on Equation 2 and is defined as: $\theta\,(L = 1|e, \mathbf{x}) = \frac{\theta_{x_{isc}} + \theta_{x_{mnr}}}{2}$. The baseline NBI estimates are calculated using $\psi_{L=1|\alpha} = 1$ and $\psi_{L=0|\alpha} = 5$ according to: $\theta\,(L = 1|e, \mathbf{x}) = \frac{\theta^{be}_{x_{isc}} + \theta^{be}_{x_{mnr}}}{2}$.

Environment. A Dell XPS 9570 with Intel i9 8950hk, 32GB RAM and Windows 10 were used for the experiments. Every experiment is conducted in a Python environment using Jupyter Notebook. NBI for BCBR, NBI, BI, CP and NCP are implemented as MSSQL17 queries via PYODBC. The similarity based retrieval for BCBR and CBR-BL are implemented via MyCBR [1]. The rest of the methods are implemented via Scikit-learn 0.24.

Data Set. For the experiments we introduce a new data set of questions used in previous inspections conducted by NLIA.[5] The data set is denoted as $\mathcal{D}$ for the rest of this section and consists of 1,111,502 entries from inspections conducted between 01/01/2012 and 01/06/2019. Embedded in these entries are $N = 1,967$ unique questions from checklists used in 59,988 inspections. Each entry (case) in $\mathcal{D}$ is also associated with an id[6] which maps to a checklist $\mathbf{y}$ (past solution) used to survey the organisation $\mathbf{x}$ in one of the 59,988 inspections within $\mathcal{D}$.

5.2 Experiment 1: Answer Classification Performance (Baselines)

The goal of this experiment is to compare ML methods and select two of the best as baselines for the labour inspection CCP. Because CCP is a complex problem, we here study a new, simple classification problem as a stepping stone.

The Answer Classification Problem. Let each $\mathbf{d}_j \in \mathcal{D}$ be a case with a two-class ground truth label l_j. A model $\mathcal{M}$ is trained on the cases in $\mathcal{D}$. For any new case $\mathbf{d} = (e, \mathbf{x}, l)$ where $l = 0$ (compliance) or $l = 1$ (non-compliance), the problem goal is for $\mathcal{M}$ to correctly classify the value of l based on $(e, \mathbf{x})$.

Method. Each model is validated on the data set $\mathcal{D}$, using 8-fold cross validation with the same partitioning of data for every model. Each model $\mathcal{M}$ outputs a class prediction score for every $(e, \mathbf{x})$. Thus, the classification threshold is set to the median of $\mathcal{M}$'s scores for each validation fold. The results are measured in terms of accuracy, precision and recall which are calculated for per validation fold: $Acc = \frac{TP+TN}{TP+FP+TN+FN}$, $Prec = \frac{TP}{TP+FP}$ and $Rec = \frac{TP}{TP+FN}$.

Results and Discussion. The results are shown in Table 3 where the baselines are sorted according to Avg, which is the average score of the preceding columns. In terms of the Avg-score NBI performs better then standard ML methods such as LR, DT and NBC. NBI also has the best recall and an average runtime of 10.4 seconds per validation fold, which is significantly less than NBC, DT, LR and CBR-BL. BI has the best performance in terms of accuracy and precision, but it also has poor recall which results in a low average score. The worst performing method was CBR-BL where the size of the training data was reduced to 100,000 cases due to long running time.

Table 3. Results from the experiment. Time is measured in seconds per validation fold.

Method	***Acc***	***Prec***	***Rec***	***Avg***	***Time***
CBR-BL	0.677	0.178	0.246	0.367	60238
Random	0.500	0.161	0.500	0.387	-
CP	0.680	0.210	0.357	0.416	**3.84**
BI	**0.760**	**0.270**	0.288	0.439	3.89
DT	0.644	0.233	0.529	0.469	122.6
NCP	0.592	0.250	0.761	0.534	9.0
NBC	0.588	0.251	0.778	0.539	67.33
LR	0.591	0.252	0.782	0.542	68.4
NBI	0.605	0.261	**0.790**	**0.552**	10.4

The results indicate that NBI yields the best average performance, which motivates us to combine NBI with CBR. LR, NBC and NCP also perform well, but we select NBI and LR as baselines for the next experiments. A limitation for this experiment is that it cannot be used to evaluate BCBR, as BCBR is designed for CCP and not ACP.

[5] The data set is available at https://dx.doi.org/10.21227/m1t7-hg51

[6] The id is a "key" for identifying a past checklist/organisation pair (value) in $\mathcal{D}$.

5.3 Experiment 2: Trustworthiness of Constructed Checklists

The goal of this experiment is to measure justification of constructed checklists $\mathbf{y}^{cnd}$ for the CCP. This is done by measuring the average proportion of questions $e_i \in \mathbf{y}^{cnd}$ which are justified by past cases. The experiment is based on Lee et al.'s use of past cases to justify predictions and promote trust [13]. The experiment is conducted on checklists constructed by BCBR and two of the baselines from Section 5.2, NBI and LR.

Method. Each model $\mathcal{M}$ is trained on the data set $\mathcal{D}$ containing 1,111,502 entries. An evaluation data set $\mathcal{D}_V$ of 59,988 tuples $(\mathbf{x}^{cnd}, \mathbf{y})$ of past entity/checklist pairs is created using every unique id from $\mathcal{D}$. For each $\mathbf{x}^{cnd} \in \mathcal{D}_V$, $\mathcal{M}$ constructs a checklist $\mathbf{y}^{cnd}$ for $\mathbf{x}^{cnd}$ as following depending on the model in question. For $\mathcal{M} = NBI$ or $\mathcal{M} = LR$: $\mathcal{M}$ generates a prediction score for every unique $e_j \in \mathcal{D}$. The $K = 15$ questions with the highest prediction scores are selected as the candidate checklist $\mathbf{y}^{cnd}$ for $\mathbf{x}^{cnd}$. For $\mathcal{M} = BCBR$: a query containing $\mathbf{x}^{cnd}$ is defined to retrieve past cases, containing $K = 15$ unique questions for $\mathbf{y}^{cnd}$.

Each $\mathbf{y}^{cnd}$ constructed by one of the models $\mathcal{M}$ then forms an evaluation pair $(\mathbf{y}^{cnd}, \mathbf{x}^{cnd})$ with each corresponding $\mathbf{x}^{cnd}$ from $\mathcal{D}_V$. Based on Equation 5, the average justification $(J_{\mathcal{M}})$ for every pair $(\mathbf{y}^{cnd}, \mathbf{x}^{cnd})$ given $\mathcal{M}$ is:

$$J_{\mathcal{M}}(\mathcal{D}, \mathcal{D}_V) = \frac{\sum_{(\mathbf{y}^{cnd}, \mathbf{x}^{cnd})} J(\mathbf{y}^{cnd}, \mathbf{x}^{cnd}, \mathcal{D})}{|\mathcal{D}_V|} \tag{6}$$

$J_{\mathcal{M}}$ measures the average percentage of questions $e_i \in \mathbf{y}^{cnd}$ where at least one corresponding explanatory case $(e_i, \mathbf{x}^{cnd}, \cdot)$ exists in $\mathcal{D}$. The purpose of the $J_{\mathcal{M}}$ score is to enable a fair comparison between the three models. A higher relative score means higher justification of the checklists constructed by $\mathcal{M}$.

Results and Discussion. The results are: $J_{NBI} = 0.6\%$, $J_{LR} = 4.8\%$ and $J_{BCBR} = 64\%$. This suggests that both LR and NBI perform poorly in terms of justification of their constructed checklists. Qualitative assessments of some of the checklists also reveal that many of their questions $(e_i \in \mathbf{y}^{cnd})$ are unrelated to and incompatible with the target entities. Because of the incompatibility issues and that less than 5% of the items on the checklists are justified, LR and NBI are not trustworthy. BCBR scored 64% which is significantly higher. Incompatible questions also seam to appear less frequently in BCBR's checklists.

5.4 Experiment 3: Evaluation of Constructed Checklists

The goal of this experiment is to evaluate the performance of the BCBR framework against LR, NBI and the original past checklists from the data set. Since BCBR uses similarity based retrieval, NBI and LR serve as non-similarity based baselines to compare with. Due to the results in Section 5.3, a filter is applied to both LR and NBI to ensure that every checklist can be justified by past cases. This is necessary for the evaluation procedure, as it assumes that the questions on the checklists can be justified by past similar cases.

Method. The evaluation approach is done on the data set $\mathcal{D}$ which contains 1,111,502 entries. The approach can be summarized as following: The data set

$\mathcal{D}$ is partitioned into a training fold ($\mathcal{D}_T$) and validation fold ($\mathcal{D}_{\mathcal{CB}}$), where the training fold is used to calculate probability estimates for the validation cases. The validation fold is used as the case base and for performance evaluation. A model $\mathcal{M}$ is trained on $\mathcal{D}_T$ and the evaluation is done on every checklist $\mathbf{y}^{cnd}$ constructed by $\mathcal{M}$.

A problem with the validation is that since every $\mathbf{y}^{cnd}$ is a new checklist, the ground truths l needed to evaluate $\mathbf{y}^{cnd}$ can be missing. A common solution to this problem is to collect the ground truth empirically [23], but this is not an option for us. To get a meaningful validation result, the performance statistics for the evaluation need to be estimated. To accomplish this, the following assumption is made: Let $\mathbf{d}^{cnd} = (-, \mathbf{x}^{cnd}, -)$ be a case without question component or observed ground truth answer and $\mathbf{d} = (e, \mathbf{x}, l)$ be any validation case with ground truth. If $\mathbf{x}^{cnd}$ and $\mathbf{x}$ are content-wise equal or similar, we assume that the unobserved ground truth answer l^{cnd} from applying e to $\mathbf{x}^{cnd}$ is correctly estimated from an empirical distribution of l, conditioned on $\mathbf{x}, e$ and the validation data fold. This is based on the assumption that similar problems have similar solutions [15].

Based on the assumption, we introduce the following procedure to estimate accuracy (Acc), precision (Prec)[7] and recall (Rec) for every model $\mathcal{M}$.

1. Let $\mathcal{D}_T$ be the training fold and $\mathcal{D}_{\mathcal{CB}}$ be both the validation fold and case base(for BCBR). Let $\mathcal{D}_V$ be a set of past entity/checklist pairs $(\mathbf{x}^{cnd}, \mathbf{y})$ from $\mathcal{D}_{\mathcal{CB}}$, created using every unique id in $\mathcal{D}_{\mathcal{CB}}$. A model $\mathcal{M}$ is trained on $\mathcal{D}_T$.
2. For every $\mathbf{x}^{cnd} \in \mathcal{D}_V$, $\mathcal{M}$ selects K unique questions (e_i) for a checklist $\mathbf{y}^{cnd}$ to form a validation pair $(\mathbf{x}^{cnd}, \mathbf{y}^{cnd})$. The questions are selected from $\mathcal{D}_{\mathcal{CB}}$.
3. For each pair $(\mathbf{x}^{cnd}, \mathbf{y}^{cnd})$ the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) are estimated by evaluating each $e_i \in \mathbf{y}^{cnd}$(predicted positives) and $e_j \notin \mathbf{y}^{cnd}$(predicted negatives).
4. For every question $e_i \in \mathbf{y}^{cnd}$, both TP_{e_i} and FP_{e_i} are estimated using the following function: $f(l, \mathbf{x}_0, e_i) = \frac{\mathcal{D}_{\mathcal{CB}}\#(L=l \wedge \mathbf{X}=\mathbf{x}_0 \wedge E=e_i)}{\mathcal{D}_{\mathcal{CB}}\#(\mathbf{X}=\mathbf{x}_0 \wedge E=e_i)}$, so that $TP_{e_i} = f(1, \mathbf{x}_0, e_i)$ and $FP_{e_i} = f(0, \mathbf{x}_0, e_i)$. If $\mathcal{D}_{\mathcal{CB}}\#(\mathbf{X} = \mathbf{x}^{cnd} \wedge E = e_i) > 0$, then $\mathbf{x}_0 = \mathbf{x}^{cnd}$ is applied to f. If $\mathcal{D}_{\mathcal{CB}}\#(\mathbf{X} = \mathbf{x}^{cnd} \wedge E = e_i) = 0$, then $\mathbf{x}_0 = \mathbf{x}_i$ from the case $(e_i, \mathbf{x}_i, l_i)$, retrieved by BCBR[8] for $\mathbf{y}^{cnd}$, is used because there is no data to evaluate $(e_i, \mathbf{x}^{cnd})$. Each TP_{e_i} and FP_{e_i} is assigned a value $v \in [0, 1]$ via f so that $TP_{e_i} = 1 - FP_{e_i}$.
5. For every unique question $e_j \notin \mathbf{y}^{cnd}$ in $\mathcal{D}_{\mathcal{CB}}$, both TN_{e_j} and FN_{e_j} are estimated using the function: $g(l, e_j \notin \mathbf{y}^{cnd}) = \frac{\mathcal{D}_{\mathcal{CB}}\#(L=l \wedge \mathbf{X}=\mathbf{x}^{cnd} \wedge E=e_j)}{\mathcal{D}_{\mathcal{CB}}\#(\mathbf{X}=\mathbf{x}^{cnd} \wedge E=e_j)}$. The function is used to obtain $TN_{e_j} = g(0, e_j)$ and $FN_{e_j} = g(1, e_j)$, so that each TN_{e_j} and FN_{e_j} receives a value of $v \in [0, 1]$ and that $TN_{e_j} = 1 - FN_{e_j}$.
6. TP, FP, FN and TN for each candidate checklist $\mathbf{y}^{cnd} \in (\mathbf{x}^{cnd}, \mathbf{y}^{cnd})$ are calculated as following: $TP = \sum_{e_i} TP_{e_i}$, $FP = \sum_{e_i} FP_{e_i}$, $TN = \sum_{e_j} TN_{e_j}$ and $FN = \sum_{e_j} FN_{e_j}$ for every unique $e_i \in \mathbf{y}^{cnd}$ and $e_j \notin \mathbf{y}^{cnd}$ from $\mathcal{D}_{\mathcal{CB}}$.

[7] An additional statistic Prec(gt) is included, which is precision calculated (step 4-8) using only $e_i \in \{\mathbf{y}^{cnd} \cap \mathbf{y}\}$ from cases containing the original ground truth labels.

[8] The condition $\mathcal{D}_{\mathcal{CB}}\#(\mathbf{X} = \mathbf{x}^{cnd} \wedge E = e_i) = 0$ only occurs if BCBR is used.

7. Statistics are then calculated for each $\mathbf{y}^{cnd}$: $Acc_{\mathbf{y}^{cnd}} = \frac{TP+TN}{TP+FP+TN+FN}$, $Prec_{\mathbf{y}^{cnd}} = \frac{TP}{TP+FP}$ and $Rec_{\mathbf{y}^{cnd}} = \frac{TP}{TP+FN}$. Repeat from Step 2 until every pair $(\mathbf{x}^{cnd}, \mathbf{y}^{cnd})$ is evaluated.
8. The average Acc, Prec and Rec of every checklist $\mathbf{y}^{cnd}$ constructed by $\mathcal{M}$ is: $Acc = \frac{\sum_{\mathbf{y}^{cnd}} Acc_{\mathbf{y}^{cnd}}}{|\mathcal{D}_V|}$, $Prec = \frac{\sum_{\mathbf{y}^{cnd}} Prec_{\mathbf{y}^{cnd}}}{|\mathcal{D}_V|}$ and $Rec = \frac{\sum_{\mathbf{y}^{cnd}} Rec_{\mathbf{y}^{cnd}}}{|\mathcal{D}_V|}$.

The procedure is used to evaluate BCBR and the other baselines. To evaluate the original checklists, the procedure is applied to the past checklists in the validation fold so that $\mathbf{y}^{cnd} = \mathbf{y}$ for $\mathbf{y} \in \mathcal{D}_V$ in Step 2. Step 2 for NBI and LR is done by generating predictions for every unique question (see Sect. 5.3). Then a filter is applied after prediction and before the selection of the questions for $\mathbf{y}^{cnd}$. The filter excludes any question (e) from selection if $(e, \mathbf{x}^{cnd}, \cdot) \notin \mathcal{D}_{\mathcal{CB}}$. This means that every $e_i \in y^{cnd}$ is justified by a past case so that J_{NBI} and J_{LR} is 100% (Eq. 6). The filter is necessary for the evaluation to ensure that NBI and LR construct checklists that satisfy the assumption above. The models use $K = 15$ and are validated using 4,8 and 16-fold cross validation.

Results and Discussion. The results are shown in in Table 4. The *Avg* column shows the average of the four preceding columns, where the results suggest that the checklists constructed by NBI, LR and BCBR are more effective than the original checklists. BCBR scores 0.474 which is significantly higher than the original checklists and also higher than NBI and LR. Figure 6 shows the results for different numbers of validation folds. The figure suggests that BCBR consistently outperforms NBI and LR in accuracy and precision. Also, both accuracy and precision statistics tend to increase with the size of the validation data sets. We believe this is caused by the fact that TP and TN increases compared to FP and FN as the quality of the retrieved questions increases when more cases are available. Recall also decreases with the size of the validation data sets as the number of predicted positives is fixed ($K = 15$), which entails that FN increases more than TP when the size of the validation set increases. The experiment suggests that BCBR is more effective for constructing checklists than LR or NBI.

Table 4. 8 fold cross validation results of the constructed vs. the original checklists (Org. CL).

Method	Acc	Prec (gt)	Prec	Rec	*Avg*
Org. CL	0.337	0.170	0.181	0.622	0.328
LR	0.484	0.226	0.267	0.694	0.418
NBI	0.486	0.229	0.270	0.698	0.421
BCBR	**0.574**	**0.259**	**0.343**	**0.718**	**0.474**

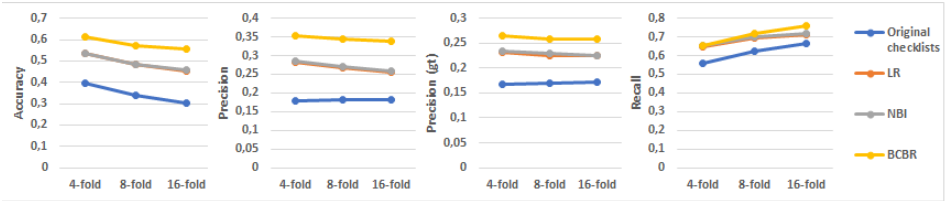

Fig. 6. Crossvalidation results for different validation fold sizes

A limitation of this experiment is that the results are based on estimates of *Acc*, *Prec* and *Rec*. For CBR frameworks, the validity of the evaluation re-

sults partially depends on high similarity between the **x**-part of the query and retrieved cases. This could be problematic when evaluating and comparing multiple CBR-based frameworks and should be investigated in future work.

6 Conclusion

In this paper we studied the problem of constructing checklists for safety critical applications, in particular labour inspections where constructing good high-performance checklists manually is difficult. Thus, we proposed the CCP where we consider the automatic construction of good, justifiable checklists. To address the CCP we introduced BCBR, which uses naive BI to construct features in CBR cases for retrieving questions for the checklists. We conducted three experiments on a data set of past labour inspections, which we introduced for the paper. Because CCP is a fairly complex problem, we conducted our first experiment on a simple answer classification problem. The goal of the experiment was to select two baselines for CCP, which was NBI and LR. In the second experiment we measured the justification of the checklist constructed by BCBR, NBI and LR, where we found that only BCBR constructs checklists which are justified by past cases. Another conclusion from the experiment is that questions selected for the constructed checklists should be justified in terms of prior use in similar entities, because some questions may be closely related to the entities that they originally were designed for. The results from the last experiment also suggest that BCBR is the most effective method for constructing checklists to address poor working conditions in inspected organisations. The checklists constructed by BCBR also perform significantly better than the original checklists.

One of the things that could be addressed in future work is solution adaptation, such as adapting questions after they have been retrieved for a checklist. Another option is to explore data-driven approaches to derive the weights and local functions for BCBR. It could also be interesting to see how BCBR perform in other CCPs such as surgery or preflight checklists.

References

1. Bach, K., Mathisen, B.M., Jaiswal, A.: Demonstrating the mycbr rest api. In: ICCBR Workshops. pp. 144–155 (2019)
2. Belser, P.: Forced labour and human trafficking: Estimating the profits (2005)
3. Bridge, D., Goker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. The Knowledge Engineering Review **20**(3), 315–320 (2005)
4. Caro-Martinez, M., Recio-Garcia, J.A., Jimenez-Diaz, G.: An algorithm independent case-based explanation approach for recommender systems using interaction graphs. In: Bach, K., Marling, C. (eds.) Case-Based Reasoning Research and Development. pp. 17–32 (2019). https://doi.org/10.1007/978-3-030-29249-2_2
5. Catchpole, K., Russ, S.: The problem with checklists. BMJ quality & safety **24**(9), 545–549 (2015)
6. Darwiche, A.: Modeling and reasoning with Bayesian networks. Cambridge University Press (2009). https://doi.org/10.1017/CBO9780511811357

7. Degani, A., Wiener, E.L.: Human factors of flight-deck checklists: the normal checklist. Ames Research Center (1990)
8. Gabel, T., Godehardt, E.: Top-down induction of similarity measures using similarity clouds. ICCBR 2015: Case-Based Reasoning Research and Development pp. 149–164 (2015). https://doi.org/10.1007/978-3-319-24586-7_11
9. Gogineni, V.R., Kondrakunta, S., Brown, D., Molineaux, M., Cox, M.T.: Probabilistic selection of case-based explanations in an underwater mine clearance domain. In: Bach, K., Marling, C. (eds.) Case-Based Reasoning Research and Development. pp. 110–124 (2019). https://doi.org/10.1007/978-3-030-29249-2_8
10. Jorro-Aragoneses, J., Caro-Martinez, M., Recio-Garcia, J.A., Diaz-Agudo, B., Jimenez-Diaz, G.: Personalized case-based explanation of matrix factorization recommendations. In: Bach, K., Marling, C. (eds.) Case-Based Reasoning Research and Development. pp. 140–154 (2019)
11. Jorro-Aragoneses, J.L., Caro-Martínez, M., Díaz-Agudo, B., Recio-García, J.A.: A user-centric evaluation to generate case-based explanations using formal concept analysis. In: International Conference on CBR. pp. 195–210 (2020)
12. Kenny, E.M., Ruelle, E., Geoghegan, A., Shalloo, L., O'Leary, M., O'Donovan, M., Keane, M.T.: Predicting grass growth for sustainable dairy farming: A cbr system using bayesian case-exclusion and post-hoc, personalized explanation-by-example (xai). In: Bach, K., Marling, C. (eds.) Case-Based Reasoning Research and Development. pp. 172–187 (2019)
13. Lee, R., Clarke, J., Agogino, A., Giannakopoulou, D.: Improving trust in deep neural networks with nearest neighbors. In: AIAA Scitech 2020 Forum. p. 2098
14. Massie, S., Wiratunga, N., Craw, S., Donati, A., Vicari, E.: From anomaly reports to cases. In: International Conference on Case-Based Reasoning. pp. 359–373 (2007)
15. Mathisen, B.M., Aamodt, A., Bach, K., Langseth, H.: Learning similarity measures from data. Progress in Artificial Intelligence (2020)
16. McConnell, C., Smyth, B.: Going further with cases: Using case-based reasoning to recommend pacing strategies for ultra-marathon runners. In: Bach, K., Marling, C. (eds.) Case-Based Reasoning Research and Development. pp. 358–372 (2019)
17. Nikpour, H., Aamodt, A.: Fault diagnosis under uncertain situations within a bayesian knowledge-intensive cbr system. Progress in Artificial Intelligence pp. 1–14 (2021). https://doi.org/10.1007/s13748-020-00227-x
18. Nikpour, H., Aamodt, A., Bach, K.: Bayesian-supported retrieval in bncreek: A knowledge-intensive case-based reasoning system. In: Case-Based Reasoning Research and Development. pp. 323–338 (2018)
19. Recio-García, J.A., Díaz-Agudo, B., Pino-Castilla, V.: Cbr-lime: A case-based reasoning approach to provide specific local interpretable model-agnostic explanations. In: International Conference on Case-Based Reasoning. pp. 179–194 (2020)
20. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019). https://doi.org/10.1038/s42256-019-0048-x
21. Rudin, C., Radin, J.: Why are we using black box models in ai when we don't need to? Harvard Data Science Review **1**(2) (2019)
22. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in case-based reasoning–perspectives and goals. Artificial Intelligence Review **24**(2), 109–143 (2005)
23. Wang, C., Agrawal, A., Li, X., Makkad, T., Veljee, E., Mengshoel, O., Jude, A.: Content-based top-n recommendations with perceived similarity. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
24. Weil, D.: If osha is so bad, why is compliance so good? RAND Journal of Economics **27**(3), 620 (1996)