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
{ole.j.mengshoel, kerstin.bach}@ntnu.no
[2] Norwegian Labour Inspection Authority, Prinsensgt. 1, Trondheim, Norway
eirik.flogard@arbeidstilsynet.no

Abstract. Checklists are used 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. With labour inspections as a starting point, we introduce the Checklist Construction Problem. To address the problem, we seek to optimize 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 to promote trustworthiness of the BI estimates. The augmented past cases are retrieved via CBR to construct checklists, which ensures justification for the content of the checklists. Experiments show that BCBR outperforms 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 inspections, which makes them a vital part of the strategy employed by many countries to ensure good health, safety, decent work conditions and well-being for

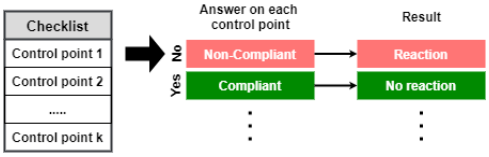

Fig. 1. Conceptual view of NLIA's procedure

[3] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52014DC033

workers (see UN's SDGs 3, 8 and 16[4]). Hence it is important that governmental agencies carry out inspections efficiently at large scale.

To identify poor working conditions, the 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 quality assurance are also used in other domains such a health (e.g. surgery) or flight procedures to ensure high accuracy of due diligence, and success often relies on 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 probability for obtaining no-answers.

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.

As a starting point for solving 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 entity. For this work, BCBR is used to address the CCP for labour inspections. BCBR uses CBR to retrieve control points 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 control points have high probabilities for non-compliance. The approach starts with a data set of cases containing organisations and control points from previously used checklists. New features

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

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 control points (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 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. The key motivations behind this are trustworthiness and interpretability.
- 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 for explanations. 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] 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. 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 neighbor (kNN) to generate voted predictions and find the nearest neighbor 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 promotes trustworthi-

ness, according to the authors. 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 control points 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 could be described as a relevant approach to solve our problem. 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].

Summary. Although there are methods and frameworks aiming to achieve goals similar to ours, these are designed for other problems or purposes that do not have much in common with the Checklist Construction Problem. However, some of the principles and ideas are used for our work.

3 Data set and problem definition

We introduce a new data set of control points used in previous inspections conducted by NLIA. The data set consists of 1,075,126 entries from inspections conducted between 01/01/2012 and 01/06/2019. Embedded in these entries are $N = 1967$ unique control points from checklists used in 59,989 inspections. The entire data set will be made available after an eventual acceptance for publication. The following definitions can be derived from 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	Categorical
x_{mnr}	Municipality number	Ordinal
x_{fyl}	Fylke (county)	Categorical
id	Past checklist id	Integer
l	Non-compliance	Binary
e	Control point	Categorical

Table 1. Description of a case in the data set.

Data set and cases. A data set D for variables $\mathbf{Z}$ is a tuple $(\mathbf{d}_1, ..., \mathbf{d}_N)$ where a case $\mathbf{d}_j \in D$ is an instantiation of $\mathbf{Z}$ [6]. A case can be defined as a tuple $\mathbf{d} = (e, \mathbf{x}, l)$ where e denotes a control point which is a question of a checklist, $\mathbf{x}$ is an organisation and $l \in \{0, 1\}$ denotes non-compliance. A case in the data set can be viewed as 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. Cases in the data set may be identical (share the same e, $\mathbf{x}$ and l) if they have different ids.

Entity. Every case 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 in the form of a control point e, which consists of a text formulated as a yes/no question used to survey $\mathbf{x}$.

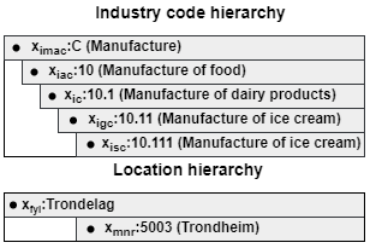

Fig. 2. Industry and location hierarchies of an organisation, including an example.

Checklist. Each case in the data set has an id which maps to a checklist $\mathbf{y}$ (past solution) used to survey the organisation $\mathbf{x}$ in a past inspection[5]. A checklist is defined as a set of control points such that $\mathbf{y} = (e_1 \in \mathbf{d}_1, e_2 \in \mathbf{d}_2 ... e_n \in \mathbf{d}_n)$. A control point can only occur once per checklist such that $e_i \neq e_j$ for every $e_i \wedge e_j \in \mathbf{y}$. The checklists contain around 15 unique control points on average and the size of the checklists varies according to the location and industry.

Answer. The label l of each case in the data set is the recorded answer from applying the control point e to the entity $\mathbf{x}$. This represent the answer of the control point where $l = 1$ 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. Given a target entity $\mathbf{x}^{cnd}$ (organisation), a model M needs to select K unique questions $(e_1, e_2, ..., e_k)$ (control points) for a candidate checklist $\mathbf{y}^{cnd}$. Each question $e_i \in \mathbf{y}^{cnd}$ needs to be selected so that it maximizes the probability for observing the answer $l_i = 1$(non-compliance) when applied to the target entity $\mathbf{x}^{cnd}$.

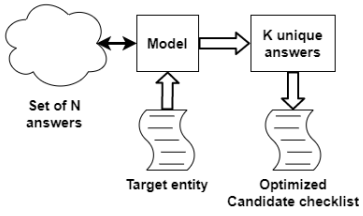

Fig. 3. An overview of CCP.

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 and to ensure that every $e_i \in \mathbf{y}^{cnd}$ 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 $\mathbf{d}_j \in D$. The estimates are generated by counting cases with the same question and entity description as $\mathbf{d}_j$. This ensures transparency for the estimates. (2) A case base CB of augmented CBR cases $\mathbf{c}_j$ is created. Each case $\mathbf{c}_j \in CB$ is created by adding both estimates as features to each $\mathbf{d}_j \in D$. (3) A query $\mathbf{q}$ is

[5] id is used to index a pair $(\mathbf{y}, \mathbf{x})_{id}$ of the data set.

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 CB. Each case contains a question e_i for the candidate checklist $\mathbf{y}^{cnd}$.

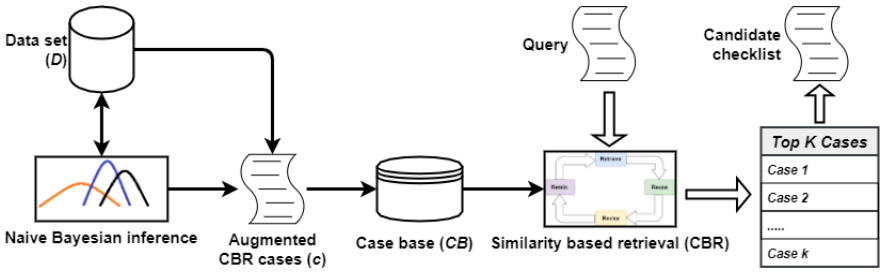

Fig. 4. An overview of the BCBR framework

4.1 Naive Bayesian inference

When prior knowledge or belief is available, it is possible to use BI to estimate empirical distributions from a data set by counting cases. BI replaces the standard maximum likelihood method for doing so and addresses inaccurate empirical estimates caused low or zero case counts ("Zero count problem") [6]. The problem may have a negative impact 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 l. By using the definitions from Section 3, the empirical distribution of a data set D can be defined as:

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

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

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

where alpha is rewritten as $\alpha = (X = \mathbf{x}) \wedge (E = e)$. 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 the probability for l. The posterior probability for an event $L = 1|\alpha$ can be expressed as the mean of a Beta distribution according to the formula below [6]:

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

where ψ is a set of prior belief parameters and where $(D\#(L = 1 \wedge \alpha) + \psi_{L=1|a})$ and $(D\#(L = 0 \wedge \alpha) + \psi_{L=0|a})$ 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 $D\#(X_{isc} = x_{isc} \wedge E = e) \geq D\#(X = \mathbf{x} \wedge E = e)$ and $D\#(X_{mnr} = x_{mnr} \wedge E = e) \geq D\#(X = \mathbf{x} \wedge E = e)$. The approach is "naive" since it assumes that x_{mnr} and x_{isc} are independent given l and e.

4.2 Case base and retrieval

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 CB with augmented cases $\mathbf{c}$. The algorithm includes two additional features: $cnt_{x_{mnr}}$ and $cnt_{x_{isc}}$. The features are included to adjust for the case counts of the probability estimates when retrieving cases. The values for the probability and *cnt*-features are estimated from D, given $x_{(mnr,j)}$, $x_{(isc,j)}$ and e_j from $\mathbf{d}_j \in D$. The features are added to $\mathbf{d}_j$ to form a case $\mathbf{c}$ for 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 CB with cases $\mathbf{c}$

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

Case retrieval and similarity function. The similarity based retrieval is implemented by using the myCBR tool [1]. 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 $(\mathbf{q}, \mathbf{c}_j \in CB)$. 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. They can be found in 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 $\mathbf{d} \in D$. The prior parameters are $\psi_{L=1|\alpha} = 1$ and $\psi_{L=0|\alpha} = 5$ because $l = 1$ is observed in approximately 1 of (1+5) cases. Given this information, $\theta^{be}_{x_{isc}}$ is estimated by counting cases $\mathbf{d}$ in data set D that satisfy $X_{isc} = x_{isc}$ and $E = e$. Applying $\alpha = (X = x_{isc} \wedge E = e)$ 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 $\mathbf{c}$, which happens to be Case 1 in Table 2.

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%
$cnt_{x_{isc}}$	1	70	1	-	0
$cnt_{x_{mnr}}$	1	70	89	-	30
Sim		-	0.546	-	0.448

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

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 $cnt_{x_{isc}}$ and $cnt_{x_{mnr}}$ 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 $(\mathbf{q}, \mathbf{c} \in CB)$, the top $K = 1$ case with highest similarity (Case 1) is retrieved for the candidate checklist $\mathbf{y}^{cnd}$.

For comparison, we also defined Query 2 in Table 2 where $\theta^{be}_{x_{isc}}$, $\theta^{be}_{x_{mnr}}$, $cnt_{x_{isc}}$ and $cnt_{x_{mnr}}$ are undefined. The $K = 1$ case returned from 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 ex-

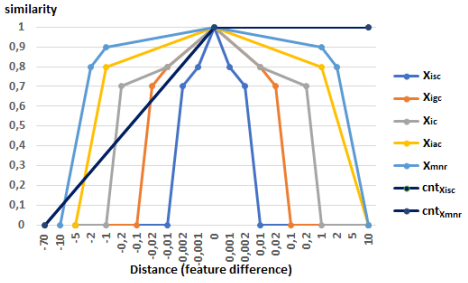

Fig. 5. Local similarity functions.

pected 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 the two best performing baselines from the first experiments, to establish the motivation for the BCBR framework. 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 an entity $\mathbf{x}$, according to the proportion of questions $e_i \in \mathbf{y}$ which also exist in past cases $(e_i, \mathbf{x}, \cdot) \in D$.

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

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 neighbor case in D.

BCBR configuration. The NBI estimates $\theta^{be}_{x_{isc}}$ and $\theta^{be}_{x_{mnr}}$ are calculated from Equation 3 using fixed priors $\psi_{L=1|\alpha} = 1$ and $\psi_{L=0|\alpha} = 5$. The priors are based on the belief that $l = 1$ can be observed in 1 out of 6 cases. In the future, more domain knowledge could be incorporated into the estimates by varying the priors according to $\mathbf{x}$ and e.

The weights used for the similarity based retrieval are the same as in the example above and can be found in Table 2. The weights for the probability estimates are calculated by summing the weights assigned for $\mathbf{x}$. The weight for θ^{be}_{mnr} is set to 4 because x_{mnr} and x_{fyl} belongs to the same hierarchy (see Fig. 2) and because $w_{x_{mnr}} + w_{x_{fyl}} = 4$. The weight for $\theta^{be}_{x_{isc}}$ is set to 9 because $w_{x_{imac}} + w_{x_{iac}} + w_{x_{ic}} + w_{x_{igc}} + w_{x_{isc}} = 9$. Each of the other weights in Table 2 are assigned one of three possible values $\{0, 1, 2\}$ by a domain expert, according to the importance of the feature it is associated with.

The local similarity functions are defined according to Figure 5. The functions are defined to adjust the similarity according to the hierarchical relationship between the ordinal features of the entity $\mathbf{x}$ (see Section 3). For any other features the default option in the myCBR tool is used, which is identity functions for categorical features and linear difference for everything else.

For every query $\mathbf{q}$ executed by BCBR, the target $\theta^{be}_{x_{isc}}$ and $\theta^{be}_{x_{mnr}}$ are set to 100%. The case count targets $cnt_{x_{isc}}$ and $cnt_{x_{mnr}}$ are set to 70.

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 neighbor 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 $\alpha = (X = \mathbf{x} \wedge E = e)$. NCP is based on Equation 2 and is defined as following:

$$\theta\left(L = 1|e, \mathbf{x}\right) = \frac{\theta_{x_{isc}} + \theta_{x_{mnr}}}{2} \tag{6}$$

NBI estimates are calculated using $\psi_{L=1|\alpha} = 1$ and $\psi_{L=0|\alpha} = 5$:

$$\theta\left(L = 1|e, \mathbf{x}\right) = \frac{\theta^{be}_{x_{isc}} + \theta^{be}_{x_{mnr}}}{2} \tag{7}$$

Environment. A Dell XPS 9570 with Intel i9 8950hk, 32GB RAM and Windows 10 was used as hardware for the experiments. Every experiment is conducted in a Python environment using Jupyter Notebook. NBI (both baseline and BCBR), BI, CP and NCP are implemented as MSSQL17 queries executed by PYODBC. CBR-BL is implemented via myCBR. The rest of the methods are implemented via Scikit-learn 0.24.

5.2 Experiment 1: Answer classification performance test of basic ML methods

The goal of this experiment is to compare ML methods and select the two best as baselines for the labour inspection CCP. Because CCP is a complex problem, the experiment is conducted on a new, simple classification problem:

The Answer Classification Problem. Let each case $\mathbf{d}_j \in D$ be a case with a ground truth label l_j. A model M is trained on the cases in D such that for any new case $\mathbf{d} = (e, \mathbf{x}, l)$, M correctly classifies the value of l based on $(e, \mathbf{x})$.

Method. Each model is validated on the data set D (from Section 3), using 8-fold cross validation with the same partitioning of data for every model. Each model M outputs a class prediction score for every $(e, \mathbf{x})$. Thus, the classification threshold is set to the median of M's scores for each validation fold.

Results and discussion. The results are shown in Table 3. In terms of accuracy, precision and recall NBI performs better then standard ML methods such as LR, DT and NBC. BI had the best performance in terms of accuracy and precision, but had poor performance in terms of recall due to a high number

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

Table 3. Result from baseline experiments. The time is measured in seconds per validation fold.

of cases with zero-value predictions. The worst performing method was CBR-BL where the size of the training data was reduced to 100000 cases due to long running time.

The results suggest that NBI yields the best performance in average, which is one of the motivations for combining NBI with CBR. Another advantage with NBI is the average runtime of 10.4 seconds per validation fold, which is significantly less than NBC, DT, LR and CBR-BL. A limitation for this experiment is that it cannot be used to evaluate BCBR, as BCBR is designed for CCP.

5.3 Experiment 2: Trustworthiness of constructed checklists

The goal of this experiment is to measure justification of the 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 concept where the existence of a past case justifies a prediction and promotes trust [13]. The experiment is conducted on checklists constructed by BCBR and the two best baselines in the previous experiment, NBI and LR.
Method. Each model M is trained on the data set D. The models are evaluated on every past entity/checklist $(\mathbf{y}, \mathbf{x})_{id}$ given by an unique $id \in D$ (see Sect. 3). For each $\mathbf{x}^{cnd} \in \{(\mathbf{x}, \mathbf{y})_{id} : id \in D\}$ the construction of a checklist $\mathbf{y}^{cnd}$ is done based on M. For $M = NBI$ or $M = LR$: M generates a prediction score for every unique $e_j \in D$. The top $K = 15$ questions with the highest prediction scores are selected as the candidate checklist $\mathbf{y}^{cnd}$ for each $\mathbf{x}^{cnd}$. For $M = BCBR$: a query containing each $\mathbf{x}^{cnd}$ is defined to retrieve $\mathbf{y}^{cnd}$. Each $\mathbf{y}^{cnd}$ constructed by M forms an evaluation pair $(\mathbf{y}^{cnd}, \mathbf{x}^{cnd})$ with the corresponding $\mathbf{x}^{cnd}$. Based on Eq. 5, the average justification (J_M) for every pair $(\mathbf{y}^{cnd}, \mathbf{x}^{cnd})$ given M is:

$$J_M(D) = \frac{\sum_{(\mathbf{y}^{cnd}, \mathbf{x}^{cnd})} J(\mathbf{y}^{cnd}, \mathbf{x}^{cnd}, D)}{|\{id \in D\}|} \tag{8}$$

J_M measures the average percentage of $e_i \in \mathbf{y}^{cnd}$ where at least one explanatory case $(e_i, \mathbf{x}^{cnd}, \cdot)$ exists in D. A high J_M score means high justification.
Results and discussion. The results are: $J_{NBI} = 0.6\%$, $J_{LR} = 4.8\%$ and $J_{BCBR} = 64\%$, which indicates that both LR and NBI perform poorly in terms of justification. Qualitative assessments of some of the checklists also reveals that many of their questions $(e_i \in \mathbf{y}^{cnd})$ are unrelated to the target entities. Thus, LR and NBI are not trustworthy because their checklists are unreliable and unjustified. The checklists constructed by BCBR is more reliable because similarity based retrieval is used.

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 past checklists. 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 can be summarized as following: The data set D is divided into a training and validation fold, 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. The evaluation is done on every checklist $\mathbf{y}^{cnd}$ constructed for every entity $\mathbf{x}$ in the validation fold.

A problem with the validation is that since every $\mathbf{y}^{cnd}$ is a new checklist, the ground truths l needed to evaluate $\mathbf{y}^{cnd}$ does not necessarily exist. 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)[6] and recall (Rec) for every model M.

1. The procedure is done on every pair $(\mathbf{x}, \mathbf{y})_{id} \in CB$ given by every unique $id \in CB$. CB denotes both the validation fold and case base (for BCBR).
2. For every $\mathbf{x}^{cnd} \in \{(\mathbf{x}, \mathbf{y})_{id} : id \in CB\}$, K unique questions (e_i) are selected for $\mathbf{y}^{cnd}$. The questions are selected from CB by a model M. The model is trained on the training data fold from D.
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: $f(l, \mathbf{x}_0, e_i) = \frac{CB\#(L=l \wedge X=\mathbf{x}_0 \wedge E=e_i)}{CB\#(X=\mathbf{x}_0 \wedge E=e_i)}$, so that $TP_{e_i} = f(1, \mathbf{x}_0, e_i)$ and $FP_{e_i} = f(0, \mathbf{x}_0, e_i)$. If $CB\#(X = \mathbf{x}^{cnd} \wedge E = e_i) > 0$, then $\mathbf{x}_0 = \mathbf{x}^{cnd}$ is applied to f. If $CB\#(X = \mathbf{x}^{cnd} \wedge E = e_i) = 0$, then $\mathbf{x}_0 = \mathbf{x}_i$ from the case $(e_i, \mathbf{x}_i, l_i)$ retrieved by BCBR[7] 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 $[0, 1]$ via f so that $TP_{e_i} = 1 - FP_{e_i}$.
5. For every unique question $e_j \notin \mathbf{y}^{cnd}$ in CB, both TN_{e_j} and FN_{e_j} are estimated using the following function: $g(l, e_j \notin \mathbf{y}^{cnd}) = \frac{CB\#(L=l \wedge X=\mathbf{x}^{cnd} \wedge E=e_j)}{CB\#(X=\mathbf{x}^{cnd} \wedge E=e_j)}$. 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 $[0, 1]$ 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 CB.

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

[7] The condition $CB\#(X = \mathbf{x}^{cnd} \wedge E = e_i) = 0$ 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 for every $\mathbf{y}^{cnd}$ constructed by M is found by: $Acc = \frac{\sum_{\mathbf{y}^{cnd}} Acc_{\mathbf{y}^{cnd}}}{|\{id \in CB\}|}$, $Prec = \frac{\sum_{\mathbf{y}^{cnd}} Prec_{\mathbf{y}^{cnd}}}{|\{id \in CB\}|}$ and $Rec = \frac{\sum_{\mathbf{y}^{cnd}} Rec_{\mathbf{y}^{cnd}}}{|\{id \in CB\}|}$.

The procedure is used to evaluate BCBR and the other baselines. For the original checklists, the procedure is applied by using the past checklists so that $\mathbf{y}^{cnd} = \mathbf{y}$ 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 $(e, \mathbf{x}^{cnd}, \cdot) \notin CB$. 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. 8). The filter is necessary for the evaluation to ensure that NBI and LR construct checklists that satisfy the assumption above. The models uses $K = 15$ and are validated using 4,8 and 16-fold cross validation.

Results and discussion. The results in Table 4 shows that the constructed checklists are more effective than the original checklists. In average, each checklist constructed by BCBR contains 5.14 true positives against 2.86 for the original checklists. Figure 6 shows the results for different numbers of validation folds and that BCBR has the best performance in every setup. The variation across the validation folds can be explained by the fact that folds are used as case bases (BCBR) or filters (NBI/LR). Figure 6 shows that both accuracy and precision statistics tend to increase with the size of the validation data sets. This is mainly 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. In overall, BCBR is more effective for constructing checklists than LR or NBI.

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**

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

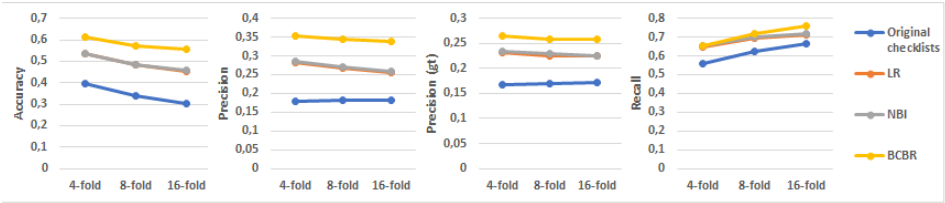

Fig. 6. Crossvalidation results for different validation fold sizes

A drawback with this experiment is that the results are based on estimated statistics. For CBR frameworks, the validity of the evaluation results 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 further in future work.

6 Conclusion

In this paper we introduced BCBR for constructing checklist to address CCP. BCBR 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 indicate that BCBR is the most effective method for constructing checklists to address poor working conditions in inspected organisations. Compared to the original checklists, the checklists constructed by BCBR yield a 79% increase in the average number of violations (true positives) that can be expected to be found at the labour inspections.

One of the things that could be addressed in future work is solution adaptation, such as optimizing the order of the questions in checklists. 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)
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)
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)
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)
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. Cham (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 Case-Based Reasoning. 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 (2020)
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)
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)
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)